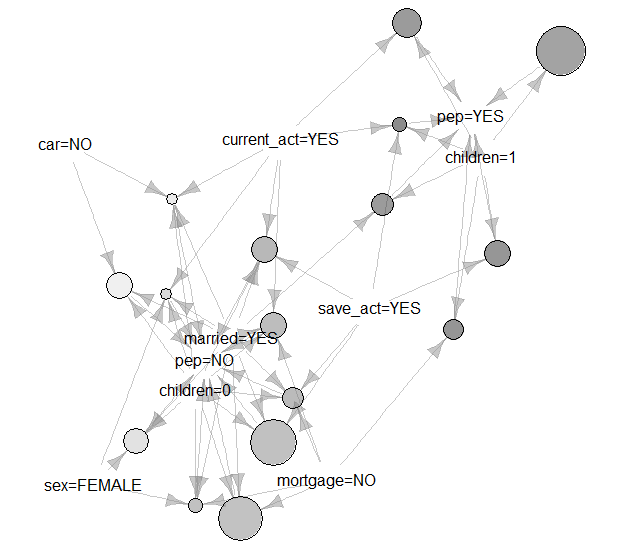
|  |
| --- |
| UMUC |
| Association Rule Mining with R |
| Week 3 Exercise |
|  |
| **DBST 667** |
|  |



|  |
| --- |
| In this exercise, you will use R Studio interface to run the Apriori Rules algorithm to find the frequent patterns, called rules, in the bank dataset. To determine which rules apply to most data, you will analyze the metrics for ranking rules. Since the algorithm works only on categorical data, you will revisit the discretization data preprocessing filter. |

Contents

[Bank Dataset 2](#_Toc450666862)

[Launch the Program 3](#_Toc450666863)

[Load the Data 4](#_Toc450666864)

[Data Preprocessing 5](#_Toc450666865)

[Remove the id attribute 5](#_Toc450666866)

[Discretization 6](#_Toc450666867)

[Factor Function 6](#_Toc450666868)

[Running Association Rules Method 7](#_Toc450666869)

[Apriori Rules Metrics 7](#_Toc450666870)

[Method Arguments 9](#_Toc450666871)

[Run the Method with Default Argument Values 9](#_Toc450666872)

[Run the Method with Specified Minimum Support and Minimum Confidence 11](#_Toc450666873)

[Eliminating Blank Itemsets 11](#_Toc450666874)

[Showing Additional Metrics 12](#_Toc450666875)

[Generating Rules for Specified Itemsets 13](#_Toc450666876)

[Pruning the Redundant Rules 15](#_Toc450666877)

[Apriori Rules Visualization 18](#_Toc450666878)

Association Rules for Bank Data

This exercise illustrates running an Apriori algorithm on bank dataset. An algorithm finds the relationships among the variables by identifying the frequent combinations in data, called rules.

Unlike the classification algorithms that you will learn later in the course, Apriori algorithm finds general rules without building a model to predict the class. Hence, any attribute or group of attributes can be on the right hand side of a rule. However, the algorithm has an option to extract the rules subset with the specified attribute at the right hand side.

This dataset is available for public download at http://people.inf.elte.hu/sila/lab\_ODM/data\_class/

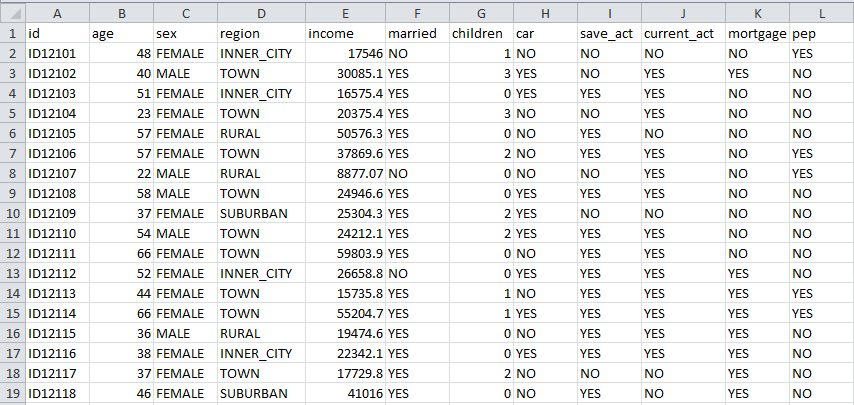
# Bank Dataset

The bank dataset tracks the customer demographics, income, car ownership, accounts ownership, and personal equity loan purchase decision (PEP). Each customer has a unique identification number.

Figure 1 shows the partial content of the bank data file. The column headings in the first row of the file are the banking attribute names called variables. The remaining 600 rows are the data, where each row is a single banking record.

Has the customer purchased the personal equity loan plan (PEP)

Customer unique identifier



Data

(each row is a bank record)

Column headers – attribute names / variables

Figure 1: Bank Data Preview

# Launch the Program

Launch the R studio program to open an interface on Figure 2.

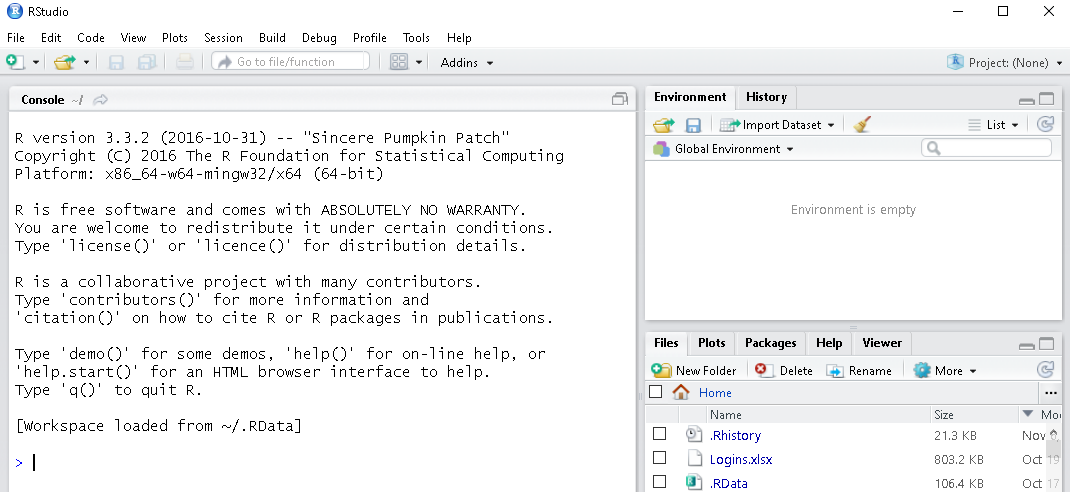


Figure 2: R Studio Interface

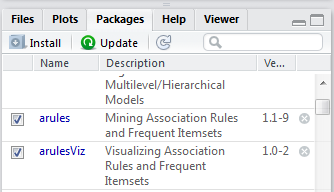
To run the Apriori rules, you need to install the arules package if you have not installed it before. Enter the following command into an application console and hit enter.

install.packages("arules")

To plot the apriori rules, you need to install the arulesViz package if you have not installed it before. Enter the following command into an application console and hit enter. If prompted to restart the R session, select yes.

You may noticed that the command installs/updates additional packages such as zoo, munsell, and labeling. The package arulesViz is dependent on those packages.

install.packages("arulesViz")



Select the Packages tab at the bottom right window of an interface.

Check the checkbox next to arules and next to arulesViz on Figure 3 to load both packages into memory.

Figure 3: Select the Packages to Load

# Load the Data

Suppose that the Bank.csv file we want to load is in the E:/Datasets folder. To set the working directory to E:/Datasets, enter the following setw command in the console window and hit the enter key. The directory path is specified in parentheses enclosed in double quotes.

setwd("E:/Datasets")

To verify that the working directory is set correctly, run the dir() command to display the files in the current working directory on Figure 4.

The dot in double quotes stands for current directory.

dir(".")



We will use Bank.csv file

Figure 4: Files in the Working Directory

Use **read.csv** command to read the bank file content into a data frame variable called bank. The first input parameter for the read.csv function is the data file name enclosed in double quotes. The second parameter, head=TRUE, specifies that the first row in the file contains the column headers. The sep parameter is the columns delimiter enclosed in double quotes. For example, sep=“,” means that the values in each data row are comma delimited.

The values delimiter

Command to Read from CSV file

bank<-read.csv(file="Bank.csv", head=TRUE, sep=",")

Read the column headings from the first row

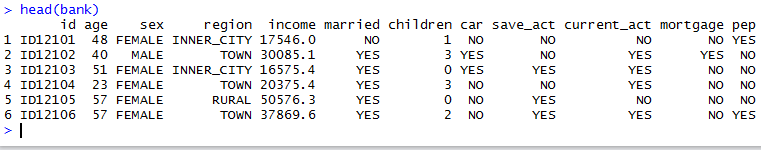
File Name

Data frame name – stores data from the first sheet in CSV file

Run the head command to preview the first 6 bank data rows on Figure 5. The head command takes the dataset name as a required input parameter and the number or rows as an optional input parameter. When the number of rows is unspecified, 6 rows are returned by default.

Make sure that the column headings in the first row match the column headings in the CSV file we just loaded. The first column contains the row names, and the column heading is empty. The remaining columns are the bank variables’ values.

head(bank)



First six bank records

Data Row Labels

Variable Names

Figure 5: First 6 Bank Records

Run the str command to display the dataset structure on Figure 6. The dataset contains 600 observations (data rows) and 12 variables. Variables age, income, and children are numeric, and the remaining variables are factors.

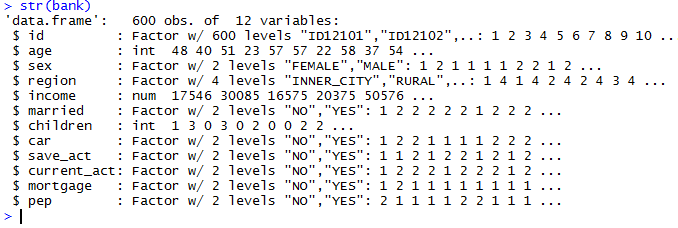
str(bank)

The Factor w/600 levels property for an id attribute indicates that an attribute is the unique identifier. The number of levels, or distinct values, is equal to the number of rows in the dataset. The unique identifiers may affect the algorithm results if they are not removed at the data pre-processing stage.

Unique Identifier – number of levels=number of observations

Number of variables

Number of Observations



Variable Names

Figure 6: Bank Data Structure

# Data Preprocessing

## Remove the id attribute

The unique identifier values are irrelevant to the analysis. For example, we are not going to analyze the relationship between the id and the purchase decision. To remove the id variable, we set it equal to NULL

NULL needs to be in the upper case.

bank$id<-NULL

To preview the first 6 rows after removing the ID attribute run the head command on Figure 7. The column with ID heading is no longer is the dataset. The number of variables has changed to 11.

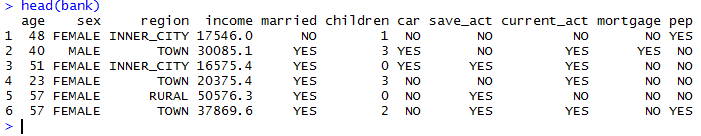


Figure 7: Bank Data Preview after Removing ID

## Discretization

The Apriori rules method requires all variables in the dataset to be discrete, or factor. However, variables age, income, and children are numeric.

To convert the age and income to factor variables, we run the unsupervised discretization filter with equal frequency binning. Numeric variable children has only 4 possible values in the dataset (0, 1, 2, 3). We use a factor function discussed in the next section.

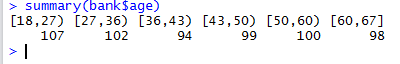
Run the following command to discretize an age variable. Then run the summary command to display the variable statistics on Figure 8.

Discretization method is equal frequency

Number of value ranges (levels)

Variable to discretize

bank$age<-discretize(bank$age, "frequency", categories=6)



98 Data rows have an age value in 60-67 range

Figure 8: Age Variable Statistics after Discretization

Run the following command to discretize an age variable. Then run the summary command to display the variable statistics on Figure 9.

bank$income<-discretize(bank$income, "frequency", categories=6)

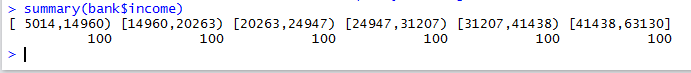


Figure 9: Income Variable Statistics after Discretization

## Factor Function

Figure 10 shows the descriptive statistics for children variable before running the factor function. The minimum number of children is 0, and the maximum number of children is three. The variable has 4 distinct values.

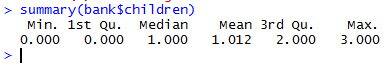


Figure 10: Children Variable Statistics before Running Factor Function

Run the following factor command on children attribute then run the summary command to view the variable statistics on Figure 11.

bank$children<-factor(bank$children)



Figure 11: Children Variable Statistics after Running Factor Function

# Running Association Rules Method

Apriori algorithm looks for patterns in the dataset and selects the rules that apply to the most instances. The left hand side of the rule is called an **antecedent**, and the right hand side of the rule is called a **consequent**.

For example, the following rule suggests that a 60-67 year old person has a savings account. An itemset on each side contains one item.

Antecedent (left hand side)

Consequent (right hand side)

{age=[60,67]} => {save\_act=YES}

An **item** is a combination of an attribute name and an attribute value. Save\_act=YES is an example of an item.

**Rule length**=left had side size + right had side size. Left hand side size is the number of items in the left hand side itemset. Right hand side size is the number of items in the right hand side itemset.

The rule above has one item on the left hand side and one item on the right hand side. Hence, the rule length is 2.

## Apriori Rules Metrics

An **itemset support** is the proportion of data rows that contain an itemset. For example, the dataset contains 98 data rows with an age between 60 and 67. Hence, the itemset {age=[60,67]} support is 98/600=0.163 where 600 is the number of data rows in the dataset.

The **rule support** is the proportion of data rows that meet the condition on both sides of the rule. For the rule above, the support is the proportion of data rows with an age between 60 and 67 and save\_act=YES.

To find out how many data rows meet the left hand side and the right hand side condition, we can run a summary command on a subset of rows with save\_act=”YES”. The statistics for a save\_act variable on Figure 12 show that all 414 rows in a subset have. The statistics for an age variable show that 83 data rows in a subset have an age in 60-67 range.

Hence, left and right hand side of a rule are true for 83 out of 600 rows. The itemset support is 83/600=0.138

summary(subset.data.frame(bank, bank$save\_act=="YES"))

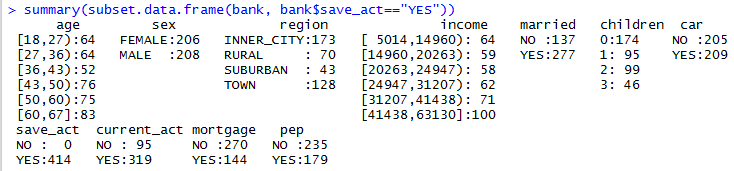


Figure 12: Summary of the Subset with save\_act=YES

**Rule confidence** is the rule accuracy - a proportion of data rows meeting the condition on the left hand side of the rule that also meets the condition on the right hand side of the rule.

For example, 98 data rows have an age value in 60-67 range. Out of 98 rows, 83 rows also have save\_act=YES. Hence, the confidence of a rule above is 83/98=0.847

**Rule lift** – The probability that antecedent and consequent occur together divided by a product of the individual probability.

In another words, lift =the proportion of the data rows that meet the condition on the left and right hand side of the rule divided by the product of proportion of the rules that meet the left hand side condition and the proportion of rules that meet the right hand side condition.

83 out of 600 data rows have an age in 60-67 range and save\_act=YES (left and right hand side conditions are met)

98 out of 600 data rows have an age in 60-67 range (left hand side condition is met)

414 out of 600 data rows have save\_act=YES (right hand side condition is met)

Rule lift ==1.2274

The higher the lift value, the stronger is the relationship between antecedent and consequent. Lift =1 indicates the antecedent and consequent are independent.

## Method Arguments

**Data** - Data frame name – is required to specify

**Parameters list** – Is a multidimensional parameter

* Support – the minimum support constraint. The rules with support below constraints will be omitted. Default is 0.1
* Confidence –the minimum confidence constraint. Rules with confidence below constraint will be omitted. Default is 0.8
* Maxlen – the maximum number of items per itemset. The default is 10
* Minlen– the minimum number of items per itemset. Default is 1
* Target – What associations are mined – rules is the default.

**Appearance** – allows eliminating the rules that do not contain the specified itemset(s). All items appear by default.

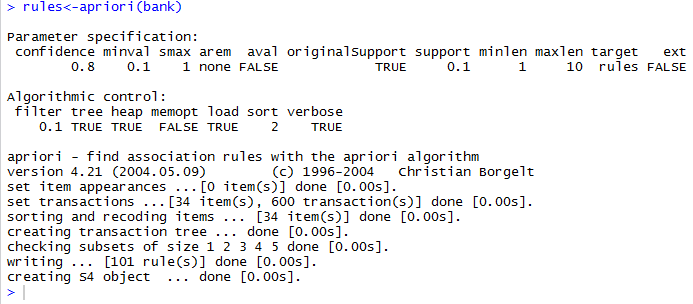
**Control** – control the sorting

## Run the Method with Default Argument Values

Enter the following command to run the method with default argument values. Rules is a variable that will store the generated rules.

rules<-apriori(bank)

An algorithm output on Figure 13 shows the algorithm input parameters, number of items, number of data rows, and the number of generated rules. When parameters are not specified, an output shows the default values used by the method.



Number of data rows in the dataset used as a method input

What associations are mined

Each rule needs to have at least on item

The rules with support below 0.1 are omitted

Each rule cannot have more than 10 items

Omit the rules with confidence below 0.8

The method returned 101 rules

Figure 13: Apriori Output

To display the number of rules, enter rules at the command prompt and hit enter.

rules

Figure 14 shows that the method generated 101 rules.



Figure 14: Number of Rules

To display the generate rules, run the inspect command.

inspect(rules)

Figure 15 Shows the first 10 out of 101 rules. The first column is a rule number in the output. The second column is the left hand side of the rule. The third column is the right hand side of the rule. The remaining columns are the support, confidence, and lift metrics for ranking rules.

Left Hand side - Antecedent

Metrics

Right hand side - consequent

Rule Number

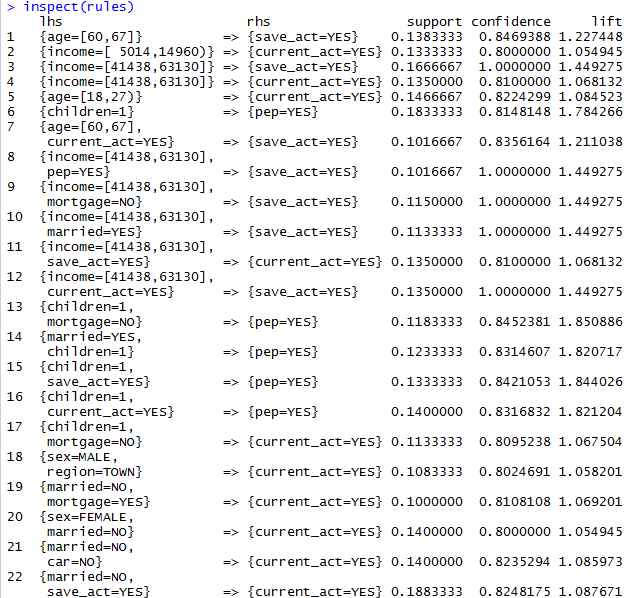


Figure 15: The first 10 Rules

You may specify the number of rules to display in the square brackets. For instance, to display only the first 10 rules, specify 1:10 in the brackets.

inspect(rules[1:10])

To display in the rules 10-20, specify 10:20 in the brackets.

inspect(rules[10:20])

Figure 16 shows the rules 10-20 and the support, confidence, and lift metrics for each rule.

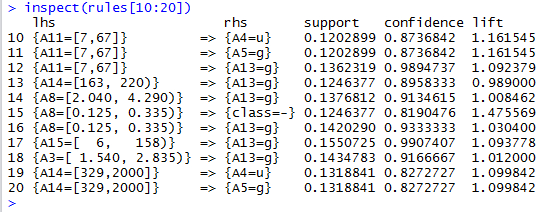


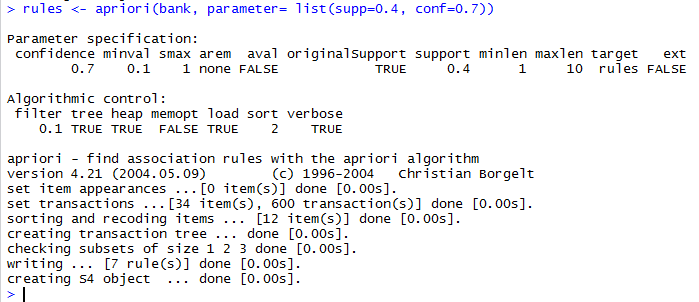
Figure 16: Rules 10-20

## Run the Method with Specified Minimum Support and Minimum Confidence

To generate the rules with support 0.4 or above and with confidence 0.7 and above, enter the following command. The first parameter is the data frame name. The second parameter is the list of settings that control which rules are eliminated and which rules are generated.

rules <- apriori(bank, parameter= list(supp=0.4, conf=0.7))

The algorithm output on Figure 17 shows that the method generated 7 rules. The minimum confidence parameter is 0.4, as specified. The minimum support parameter is 0.7 as specified.



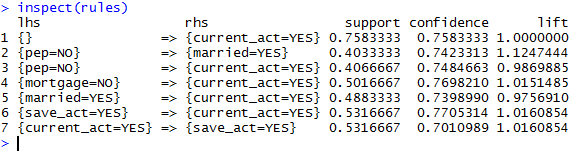
The method generated 7 rules

Omit the rules with confidence below 0.7

The rules with support below 0.4 are omitted

Figure 17: Apriori Output for the Specified Support and Confidence

Run inspect command to view the rules. The first rule on Figure 18 has a blank item set on the left hand side. The lift metric for that rule is 1, which means that the left hand side and right hand side of the rule are independent.



Blank itemset

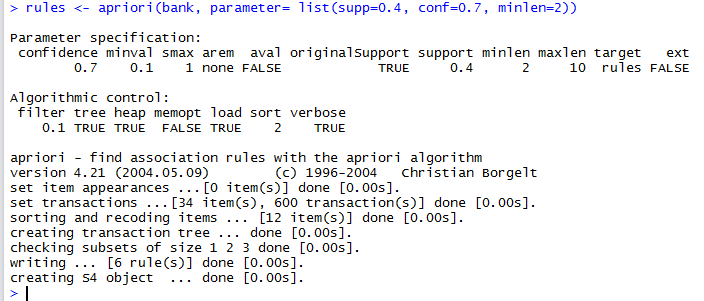
Figure 18: Rules with support>=0.4 and confidence >=0.7

## Eliminating Blank Itemsets

To eliminate the rules with the blank itemset, we need to set the minlen parameter =2. Run the following command to generate the rules with support equal to or above 0.4, confidence equals to or greater than 0.7, and with the sum of itemset size on the left and right hand side of the rule greater or equal to 2.

rules <- apriori(bank, parameter= list(supp=0.4, conf=0.7, minlen=2))

An output on Figure 19 shows that the method generated 6 rules.



The function generated 6 rules

Omit the rules with confidence below 0.7

The rules with support below 0.4 are omitted

Figure 19: Apriori Output for the Specified Support, Confidence, and minlen=2

An output from inspect command on Figure 20 no longer has a rule with a blank itemset.

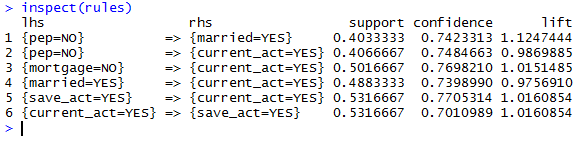


Figure 20: Rules with support>=0.4, confidence >=0.7, and minlen=2

## Showing Additional Metrics

By default, the method output shows only support, confidence, and lift measures. To view the additional measures, we run insertMeasure command. The command takes the variable that holds the rules, the list of measures, and the data frame as an input.

interestMeasure(rules, c("support", "chiSquare", "confidence", "conviction", "cosine", "coverage", "leverage", "lift", "oddsRatio"), bank)

Figure 21 shows additional metrics for each rule. The first column is the rule number, and the remaining columns are the rules metrics.

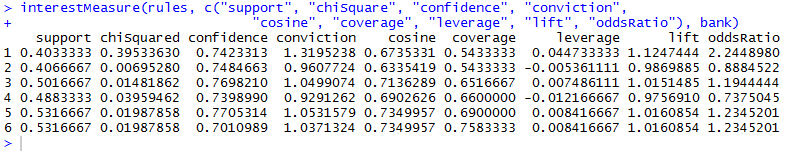


Figure 21: Additional Measures

## Generating Rules for Specified Itemsets

You may use the appearance filter to generate the rules only with the specified itemsets on the right hand side. For example, enter the following command to generate the rules with only pep=NO or pep=YES on the right hand side.

rules<-apriori(bank, parameter= list(supp=0.1, conf=0.8, minlen=2), appearance=list(rhs=c("pep=NO", "pep=YES"), default="lhs"))

An output on Figure 22 shows that the method generated 17 rules. The minimum allowed rule length is 2, and the maximum allowed rule length is 10.

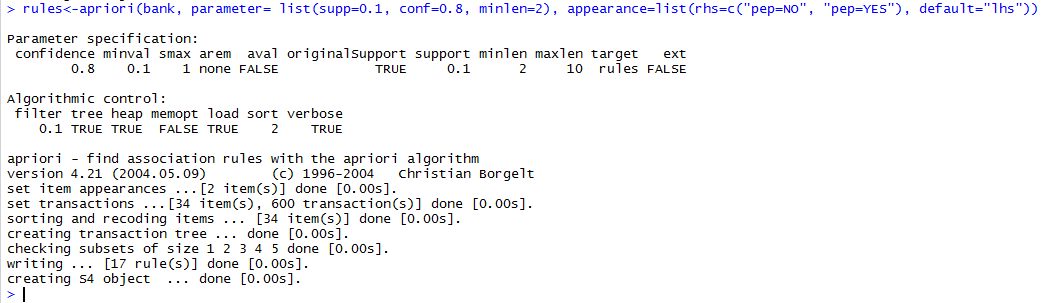


Figure 22: Apriori output for pep=NO or pep=YES on the right hand side.

You may also sort the rules by the metric value. Enter the following command to sort the rules by lift metric. Instead of overwriting the rules variable, we create another variable called rules.sorted.

rules.sorted <- sort(rules, by="lift")

Enter inspect command to preview the rules.

inspect(rules.sorted)

Figure 23shows 17 rules with the right hand side containing only pep=YES or pep=NO itemsets. The rules aresorted by lift descending.

An itemset on the left hand contains 1 item.

An itemset on the left hand side contains 3 items

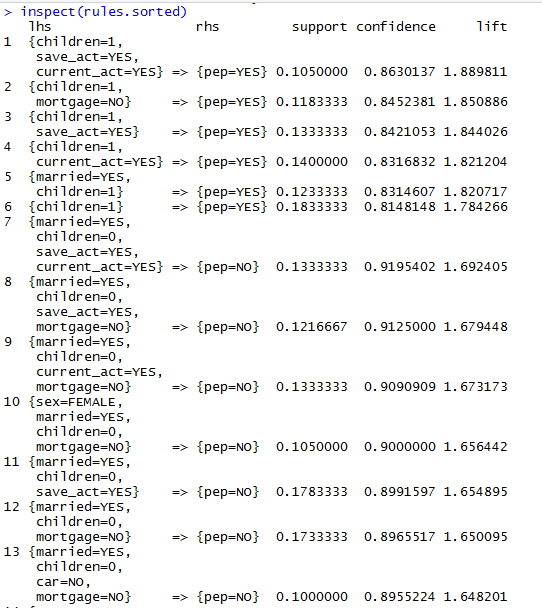
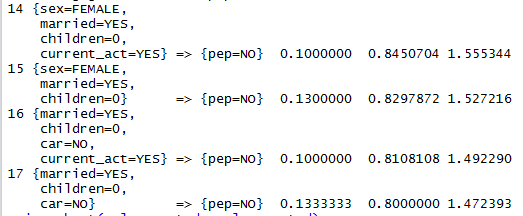


Figure 23: Sorted rules for pep=NO or pep=YES on the right hand side

## Pruning the Redundant Rules

Run the following commands to find redundant rules

subset.matrix <- is.subset(rules.sorted, rules.sorted, sparse=FALSE)

subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA

redundant <- colSums(subset.matrix, na.rm=T) >= 1

**Step 1** – We build a matrix where the first column is the itemsets from the left and and right hand side of the rule. The headings of the remaining columns are the rule numbers. For each column and row intersection we enter

**TRUE** if the corresponding itemsets in the first column are contained in the rule corresponding to the rule number in the column header.

**FALSE** if the corresponding itemsets in the first column are not contained in the rule corresponding to the rule number in the column header.

For example, the sixth rule on Figure 24 contains the itemsets {children=1,pep=YES}

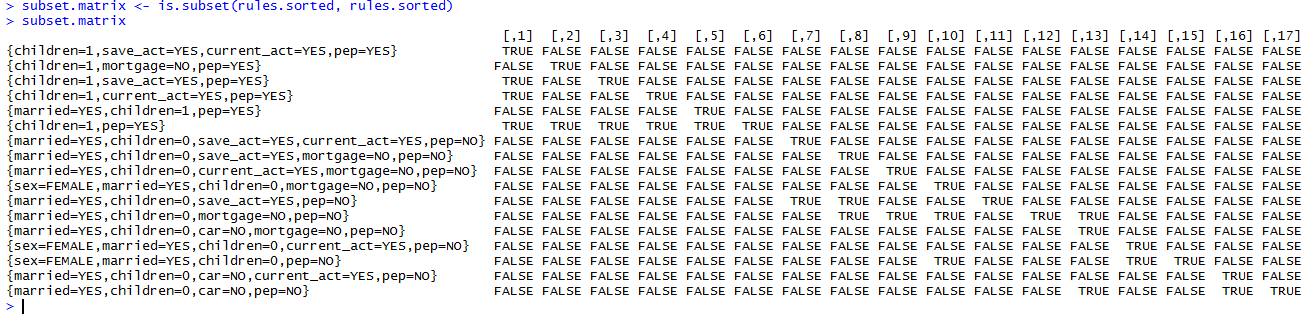
The first rule contains the itemsets {children=1,save\_act=YES,current\_act=YES,pep=YES}

Since the itemsets in the first rule contain children=1 and pep=yes, the sixth rule is the subset of the first rule. We enter TRUE in the intersection of the row for 6th rule and the column for the first rule.

The second rule with the temsets {children=1,mortgage=NO,pep=YES} is not a subset of the first rule. Hence, we enter FALSE at the intersection of the second rule row and the first rule column.

Second rule is not a subset of the first rule

Sixth rule is a subset of the first rule



Rule

Rule number

Figure 24: Rules Subset Matrix

Step 2 – Change all entries on and bellow the matrix diagonal from top left to bottom right corner to NA.

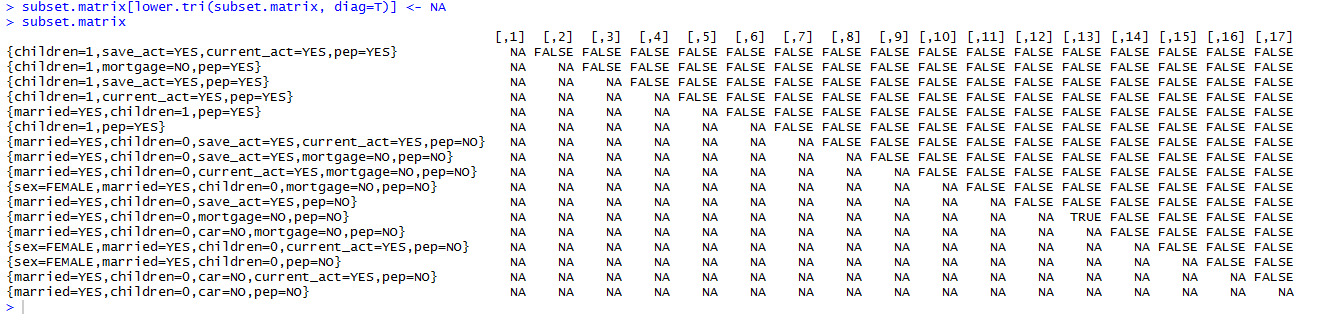


Figure 25: Set the Lower Triangle Entries to NA

Step 3 - Create an array on Figure 26 with an entry for each column starting from the column for the first rule. If the column contains one or more entry=TRUE, then set the corresponding array entry to TRUE. Otherwise, set the corresponding array entry to FALSE. Note: T is the same as TRUE

Rule 13

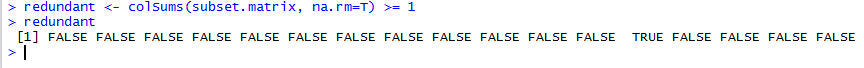


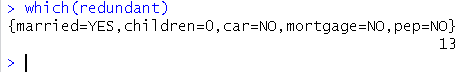
Figure 26: Redundant Rules Array

**Step 4 -** An array entry equals TRUE indicates that the corresponding rule is redundant. In this case, it’s the 13th entry which corresponds to rule number 13.

To display the rule numbers for redundant rule, run the which command. The command returns the indexes of array elements with the value=TRUE

which(redundant)

Figure 27 shows the redundant rule 13.



Rule

Rule Number

Figure 27: Redundant Rule

Run the following command to remove the redundant rule and store the remaining rules in rules.pruned variable.

rules.pruned <- rules.sorted[!redundant]

Run the inspect command to preview the remaining rules.

inspect(rules.pruned)

Figure 28 shows the remaining 16 rules after the redundant rule has been removed.

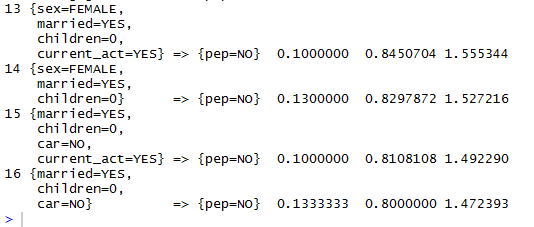
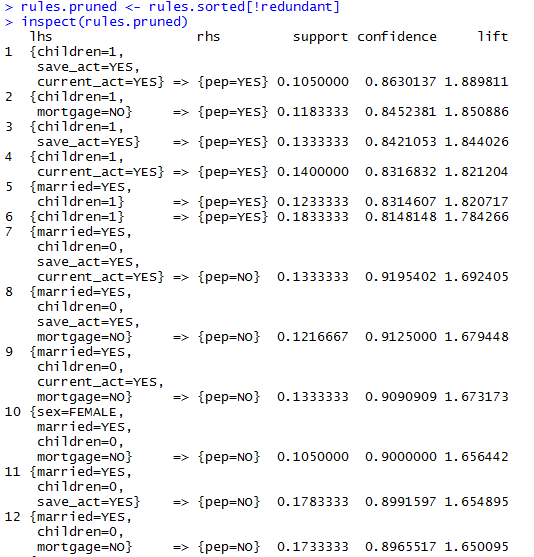
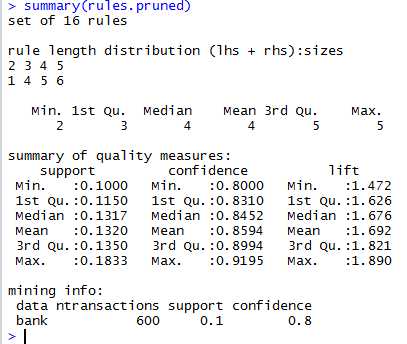


Figure 28: Rules Remaining After the Redundant Rule has been Removed

Run the summary command on Figure 29 to display the statistics for the pruned rules metrics and the input parameters used to generate the rules.



600 instances in the dataset

Minimum support parameter

Minimum confidence parameter

The method generated 16 rules total.

Rule length distribution shows the number of rules with each length. For example, one rule has length 1, 4 rules have length 3, etc. The sum of counts for each length=number of generated rules

Metrics summary measures include the minimum, 1st Quartile, mean, median, 3rd quartile, and maximum metric value. The summarized measures are support, confidence, and lift.

Mining info section shows the dataset names, data rows count, and minimum support and minimum confidence Apriori method parameters.

Dataset name

Rule length

Rule count

Figure 29: Pruned Rules Properties

# Apriori Rules Visualization

Scatter Plot is the default plot when the method parameter is unspecified.

Each data point represents a single rule. An x coordinate is the rule support, and a y coordinate is the rule confidence The point color is based on lift value. The dark red color means the higher lift value. Light yellow color means low lift value

You may use export menu options to save the plot or to copy and paste the plot into a Word document.

Run the plot command to build a scatterplot on Figure 30.

plot(rules.pruned)

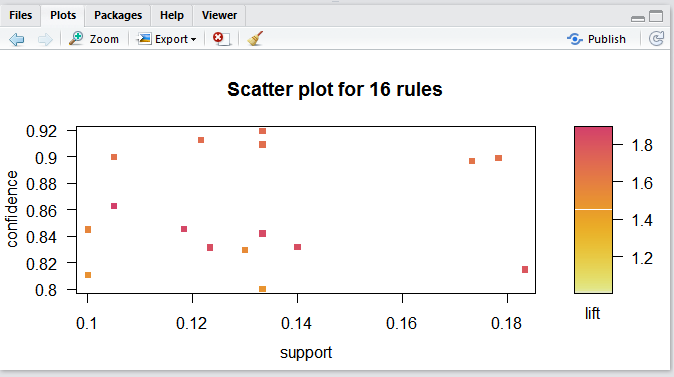
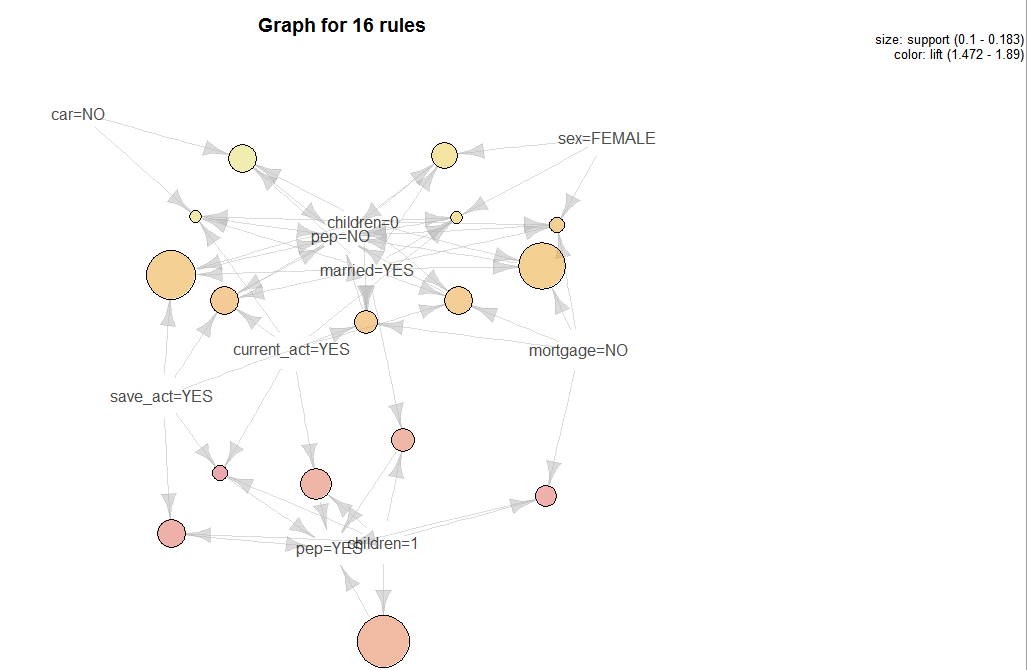


Figure 30: Pruned Rules Scatterplot

Run the following command to visualize the rules as a graph on Figure 31. The method=”graph” means that the rules are visualized as a graph, where each rule is represented by a circle.

plot(rules.pruned, method="graph", control=list(type="items"))

The circle size is proportional to the rule support. The darker the circle color, the higher is the lift value. The top right corner shows the confidence value ranges for the plotted rules.



Small light/yellow circle= Lower support value, lower lift value

Big dark/red circle= Higher support value, higher lift value

Large light/yellow circle= Higher support value, lower lift value

Small dark/red circle= Lower support value, higher lift value

Figure 31: Pruned Rules graph

Enter the following command to build the parallel coordinates plot on Figure 32

plot(rules.pruned, method="paracoord", control=list(reorder=TRUE))

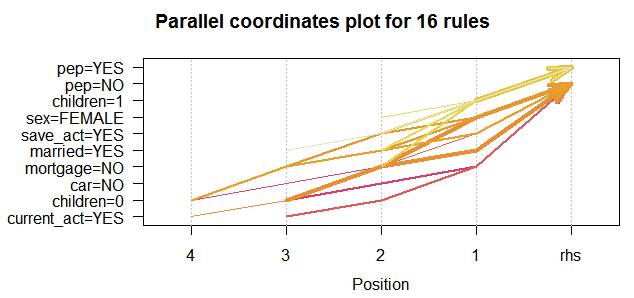


Figure 32: Parallel Coordinates plot

Run the following command to build the grouped plot on Figure 33. The left hand side itemsets are represented as columns, and the right hand side itemsets are represented as rows. For each rule, the intersection of right and left hand side itemsets is marked with a circle.

The circle size represents to the rule support. The larger the circle, the higher the support is. The circle background color represents the rule lift. Darker color means higher lift value.

plot(rules.pruned, method = "grouped")

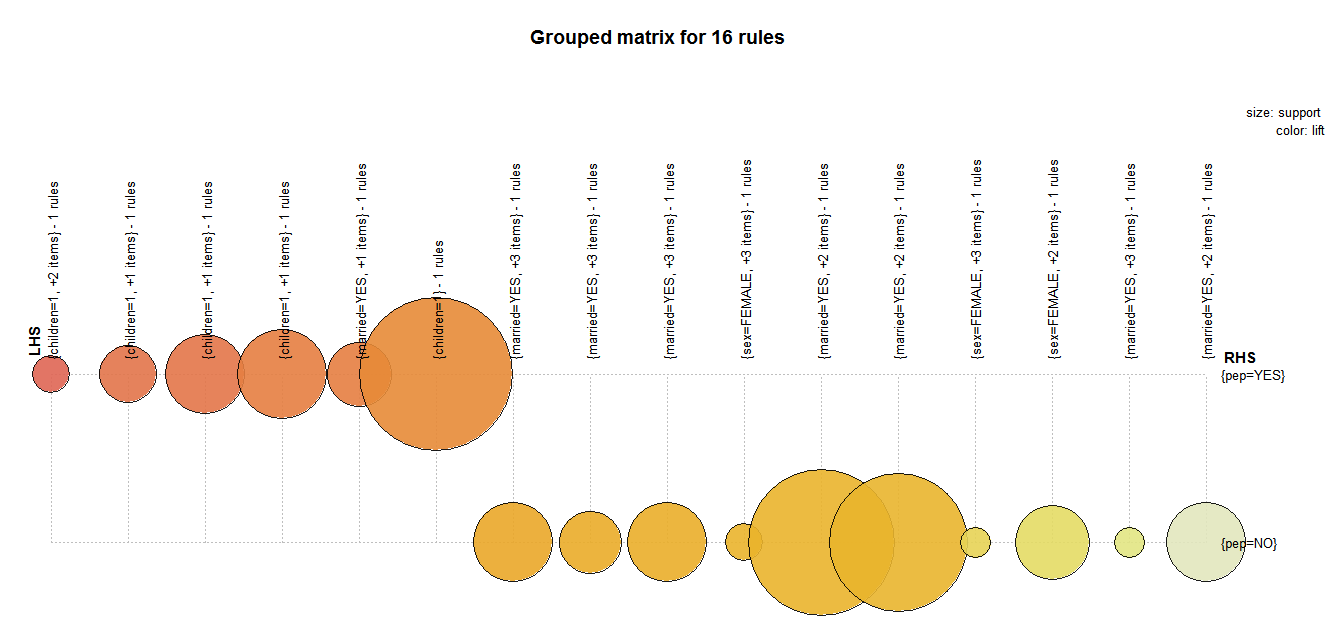


Figure 33: Grouped Plot

plot(rules.pruned, method="matrix", measure=c("lift", "confidence"))

The top section of an output on **Figure 34** is a list of the itemsets on the left hand side of the rules. The bottom section is the list of itemsets on the right hand side.

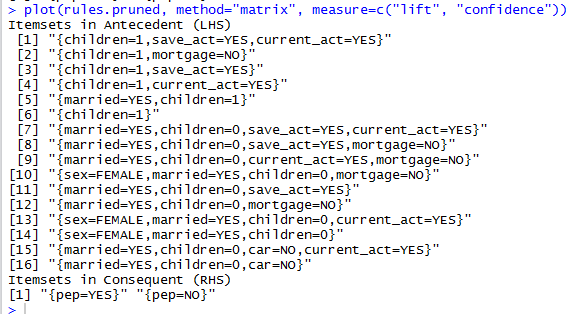


Figure 34: Matrix Plot Output

Figure 35 is a matrix plot displayed in the plots panel.

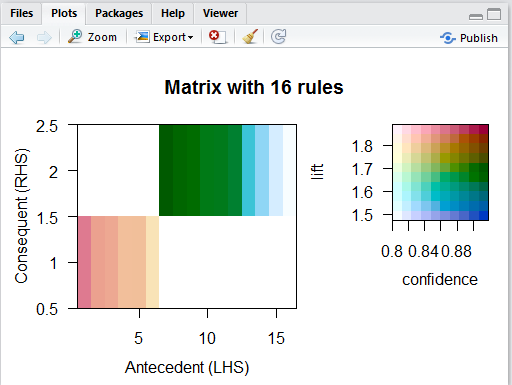


Figure 35: Matrix Plot