Market Basket Analysis

2020-03-10

Introduction

Market Basket Analysis identifies the strength of association between products purchased together and patterns of two or more things taking place together.

For example, if Bread is purchased then Butter is likely to be purchased. or if Bread is purchase then Butter and Milk are likely to be purchased. These associated purchases are useful in cross selling strategies.

We will use the Apriori algorithm for this analysis. Apriori is used for frequent item set mining and association rule learning. It identifies frequent individual items in the dataset and extends them to larger and larger item sets as long as those item sets appear sufficiently often in the dataset.

Data:

We will use Restaurant Orders data for this analysis. This data has online orders for Indian Cuisine. It would be interesting to know the combination of items that people order together.

First, load dependent libraries

```r
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(arules)) install.packages("arules", repos = "http://cran.us.r-project.org")
if(!require(arulesViz)) install.packages("arulesViz", repos = "http://cran.us.r-project.org")
```

Load data from CSV file

```r
Orders <-
  read_csv("https://raw.githubusercontent.com/madankundapur/DataAnalytics/master/Data/RestaurantOrders.csv")



# remove spaces from variable names
names(Orders)<-str_replace_all(names(Orders), c(" " = ""))

# remove rows with NA
Orders <- Orders[complete.cases(Orders), ]

# change product name type to factor
Orders$ItemName <- as.factor(Orders$ItemName)

# change order date type to date
Orders$Date <- as.Date(Orders$OrderDate, "%m/%d/%Y")
```

```r
str(Orders)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    74818 obs. of  7 variables:
##  $ OrderNumber  : num  16118 16118 16118 16118 16118 ...
##  $ OrderDate    : chr  "03/08/2019 20:25" "03/08/2019 20:25" "03/08/2019 20:25" "03/08/2019 20:25"
...
##  $ ItemName     : Factor w/ 248 levels "Aloo Chaat","Aloo Gobi",..: 189 91 82 164 174 145 188 164
225 246 ...
##  $ Quantity     : num  2 1 1 1 1 1 1 1 1 1 ...
##  $ ProductPrice : num  0.8 12.95 2.95 3.95 8.95 ...
##  $ Totalproducts: num  6 6 6 6 6 6 7 7 7 7 ...
##  $ Date         : Date, format: "2019-03-08" "2019-03-08" ...
```

The dataset has 74818 observations and 7 variables

Analysis

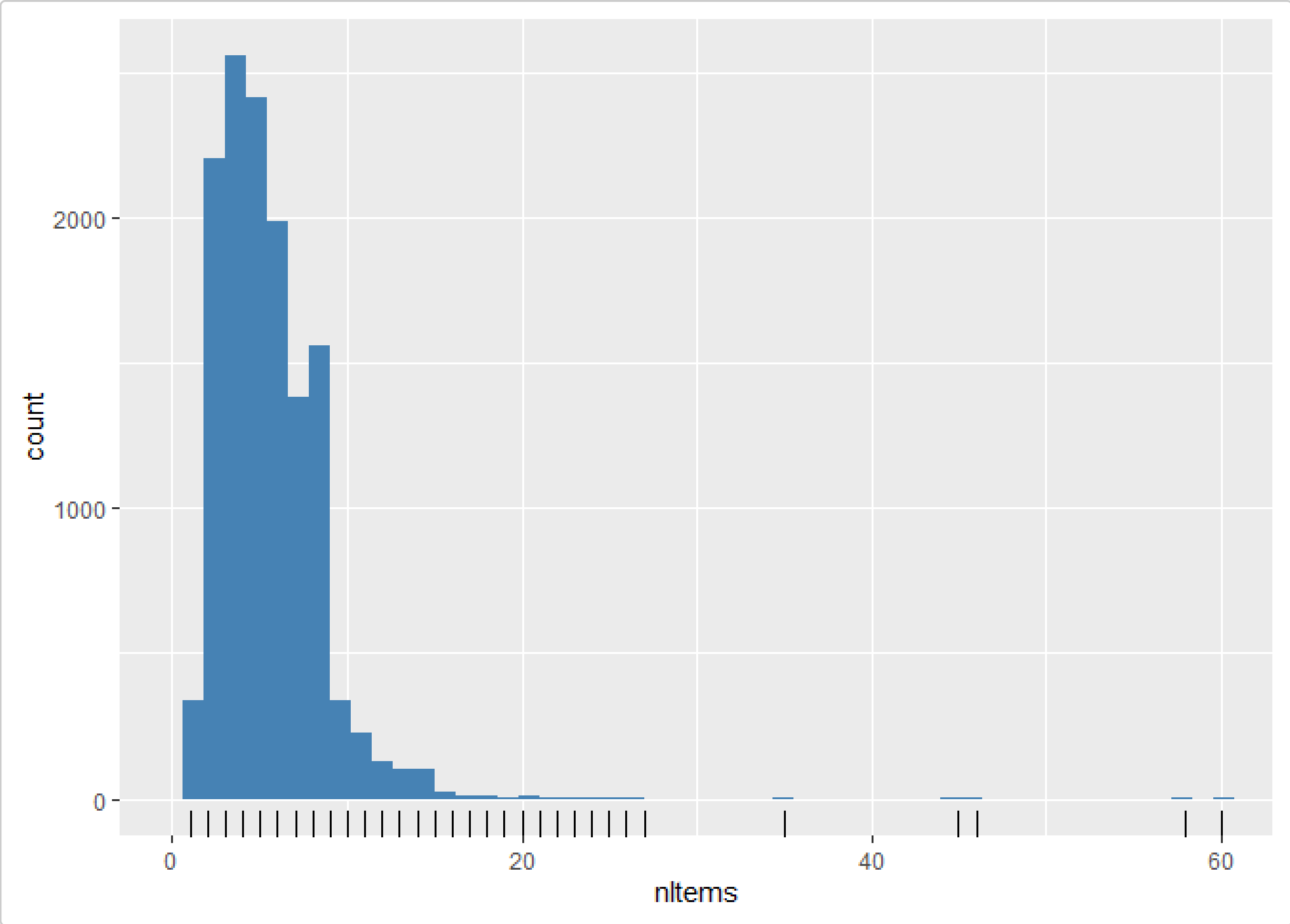
Number of Items in each order

```
ItemsByOrder <- Orders %>%
  group_by(OrderNumber) %>%
  summarize(nItems = n())

knitr::kable(summary(ItemsByOrder))
```

OrderNumber	nItems
Min. : 630	Min. : 1.000
1st Qu.: 5674	1st Qu.: 4.000
Median : 9231	Median : 5.000
Mean : 9173	Mean : 5.585
3rd Qu.:12685	3rd Qu.: 7.000
Max. :16118	Max. :60.000

```
ItemsByOrder %>%
  ggplot(aes(x=nItems))+
  geom_histogram(fill="steelblue", bins = 50) +
  geom_rug()+
  coord_cartesian(xlim=c(0,60))
```



Customers mostly order5 to 6 items

Ten best selling items

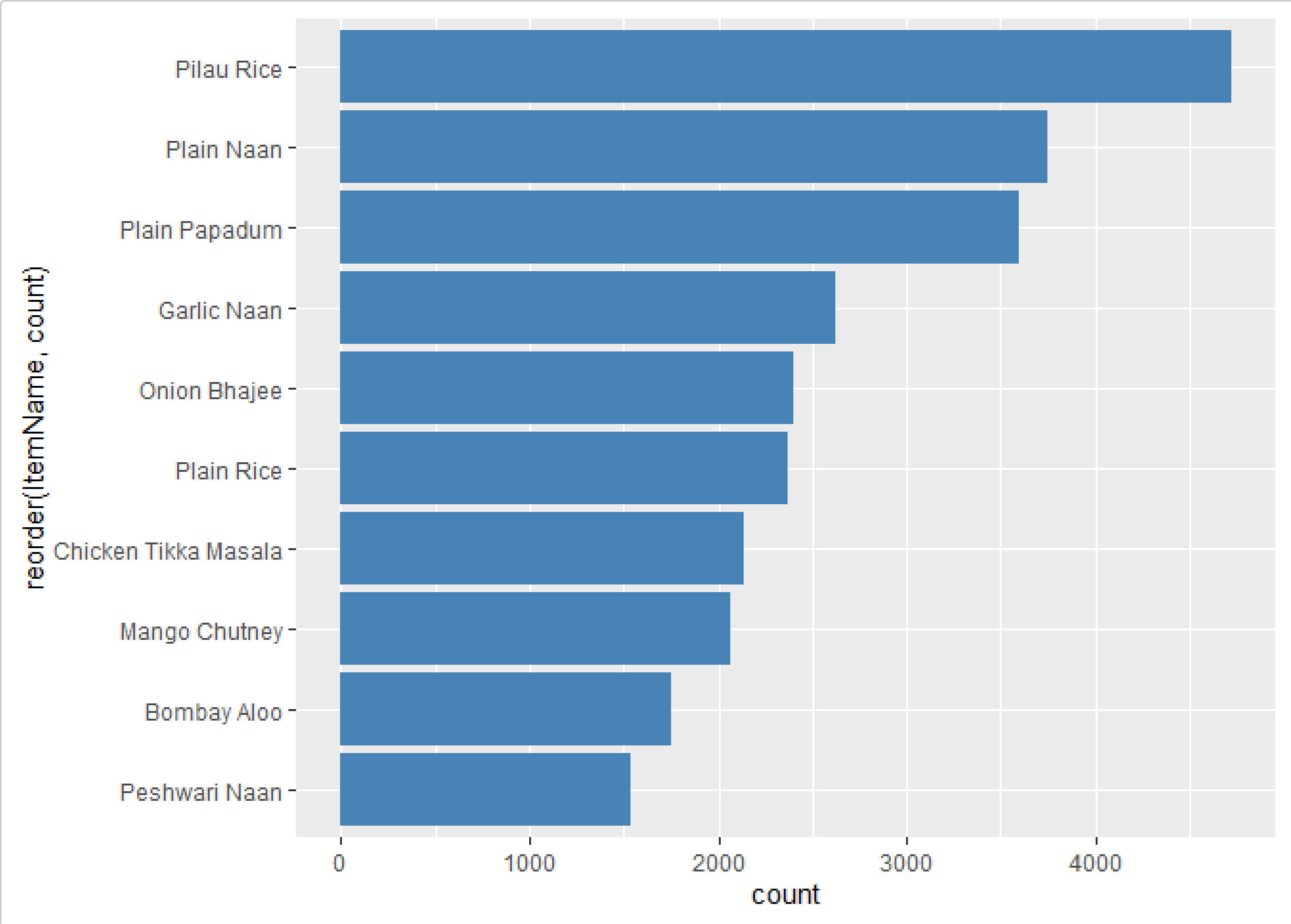
```
TopTen <- Orders %>%
  group_by(ItemName) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

TopTen <- head(TopTen, n=10)

knitr::kable(TopTen)
```

ItemName	count
Pilau Rice	4721
Plain Naan	3753
Plain Papadum	3598
Garlic Naan	2628
Onion Bhajee	2402
Plain Rice	2369
Chicken Tikka Masala	2133
Mango Chutney	2070
Bombay Aloo	1752
Peshwari Naan	1535

```
TopTen %>%
  ggplot(aes(x=reorder(ItemName,count), y=count))+
  geom_bar(stat="identity",fill="steelblue")+
  coord_flip()
```



Itemset Summary

Transform data from the data frame format into transactions such that we have all the items bought together in one row using ddply()

```
library(plyr)
ItemList <- ddply(Orders,c("OrderNumber"),
                  function(df)paste(df$ItemName,
                                    collapse = "|"))
```

Remove Order Number variable since we need only Items data

```
ItemList$OrderNumber <- NULL
colnames(ItemList) <- c("items")
```

Persist the data in a csv file for further use.

```
write.csv(ItemList,"MarketBasket.csv", quote = FALSE, row.names = TRUE)
```

We now have the dataset that shows the matrix of items bought together.

Inspect how many transactions we have and what they are.

```
write.csv(ItemList,"MarketBasket.csv", quote = FALSE, row.names = TRUE)
```

```r
Trn <- read.transactions('MarketBasket.csv', format = 'basket', sep='|')

summary(Trn)
```

```
## transactions as itemMatrix in sparse format with
##  13398 rows (elements/itemsets/transactions) and
##  13646 columns (items) and a density of 0.0004084798
##
## most frequent items:
##           Pilau Rice        Plain Papadum         Onion Bhajee
##                 3751                 2264                 2231
## Chicken Tikka Masala           Plain Rice              (Other)
##                 2111                 1874                62451
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##  338  605 1611 2559 2422 1985 1377  934  618  331  227  128  101   63   39
##   16   17   18   19   20   21   22   24   25   26   27   35   45   46   58
##   19   10    8    4    3    5    2    1    1    1    1    1    1    1    1
##   60
##    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   5.000   5.574   7.000  60.000
##
## includes extended item information - examples:
##           labels
## 1          ,items
## 2  1,Onion Bhaji
## 3 10,Onion Bhaji
```

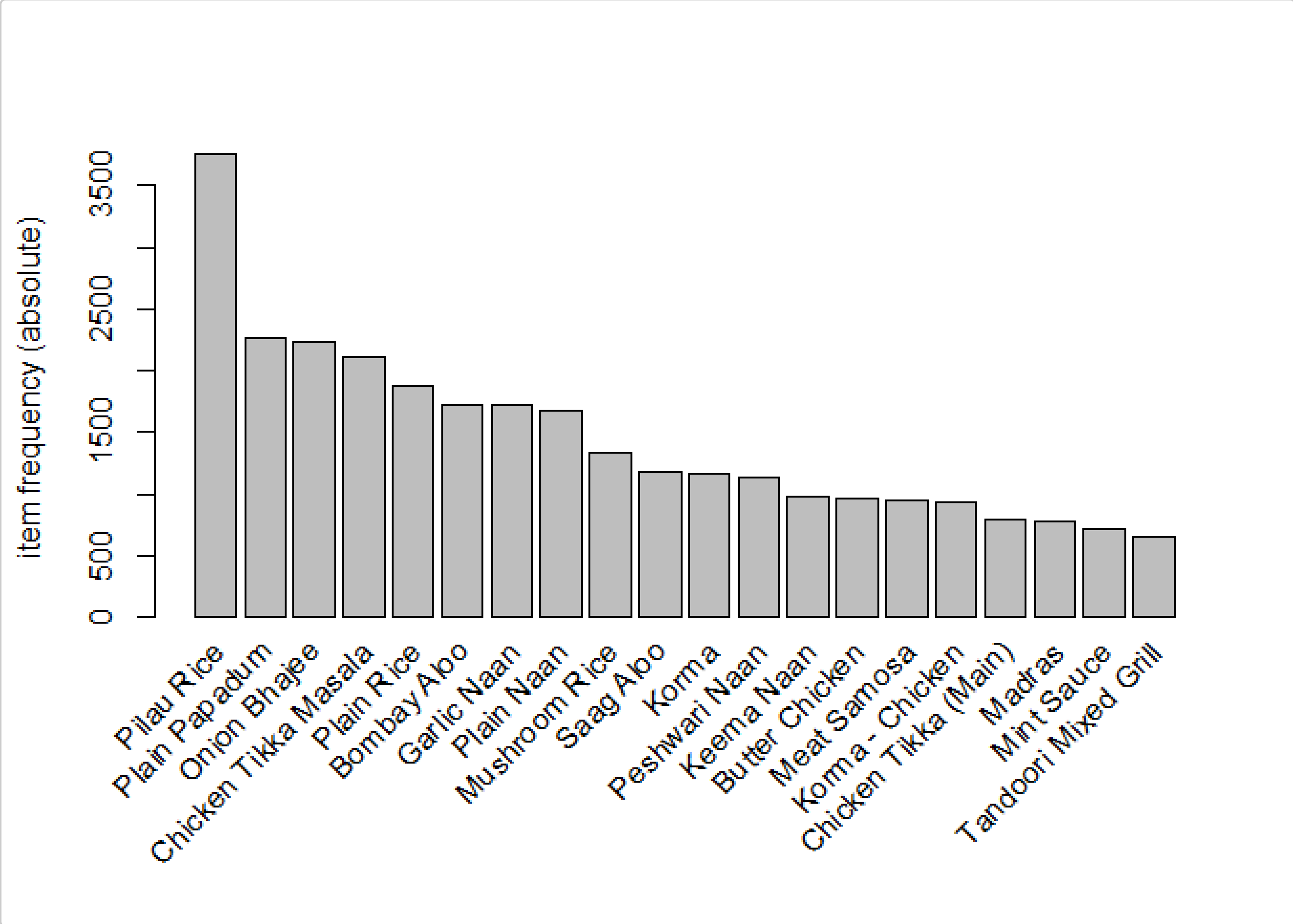
We have 13398 transactions and 13646 items

Summary gives some very useful information:

- Density: The percentage of non-empty cells in the sparse matrix. i.e. the total number of items that were purchased, divided by the total number of possible items in the matrix.
- Most frequent items: Pilau rice was the most frequently purchased item
- Sizes: Most customers buy about 5 items. 2559 transactions for 4 items, 2422 transactions for 5 items

Item frequency plot:

```r
itemFrequencyPlot(Trn, topN=20, type='absolute')
```



Applying Apriori

- Let us use the Apriori algorithm in arules library to mine frequent itemsets and association rules. The algorithm employs level-wise search for frequent itemsets.
- Pass supp=0.001 and conf=0.8 to return all the rules have a support of at least 0.1% and confidence of at least 80%.
- Sort the rules by decreasing confidence.
- The summary of the rules:

```
Rules <- apriori(Trn, parameter = list(supp=0.001, conf=0.8))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 13
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[13646 item(s), 13398 transaction(s)] done [0.21s].
## sorting and recoding items ... [221 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.03s].
```

```
## writing ... [7657 rule(s)] done [0.01s].
## creating S4 object  ... done [0.02s].
```

```
Rules <- sort(Rules, by='confidence', decreasing = TRUE)
summary(Rules)
```

```
## set of 7657 rules
##
## rule length distribution (lhs + rhs):sizes
##    2    3    4    5    6    7    8    9   10
##    3  307 1303 2167 2045 1237  479  106   10
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   6.000   5.588   6.000  10.000
##
## summary of quality measures:
##     support           confidence          lift               count
##  Min.   :0.001045   Min.   :0.8000   Min.   :  2.857   Min.   : 14.00
##  1st Qu.:0.001194   1st Qu.:0.9000   1st Qu.:  5.918   1st Qu.: 16.00
##  Median :0.001269   Median :0.9643   Median :  7.794   Median : 17.00
##  Mean   :0.001432   Mean   :0.9465   Mean   : 14.272   Mean   : 19.19
##  3rd Qu.:0.001418   3rd Qu.:1.0000   3rd Qu.: 19.038   3rd Qu.: 19.00
##  Max.   :0.046201   Max.   :1.0000   Max.   :200.369   Max.   :619.00
##
## mining info:
##  data ntransactions support confidence
##   Trn         13398   0.001        0.8
```

Summary of rules gives some very useful information:

- Total number of rules are 7657
- Most rules are 6 items long
- Summary of quality measures:

  Support: This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears

  Confidence: This says how likely item Y is purchased when item X is purchased

  Lift: This says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is.
- Data mining information

Conclusion

Let us now inspect Top 10 rules

```
inspect(Rules[1:10])
```

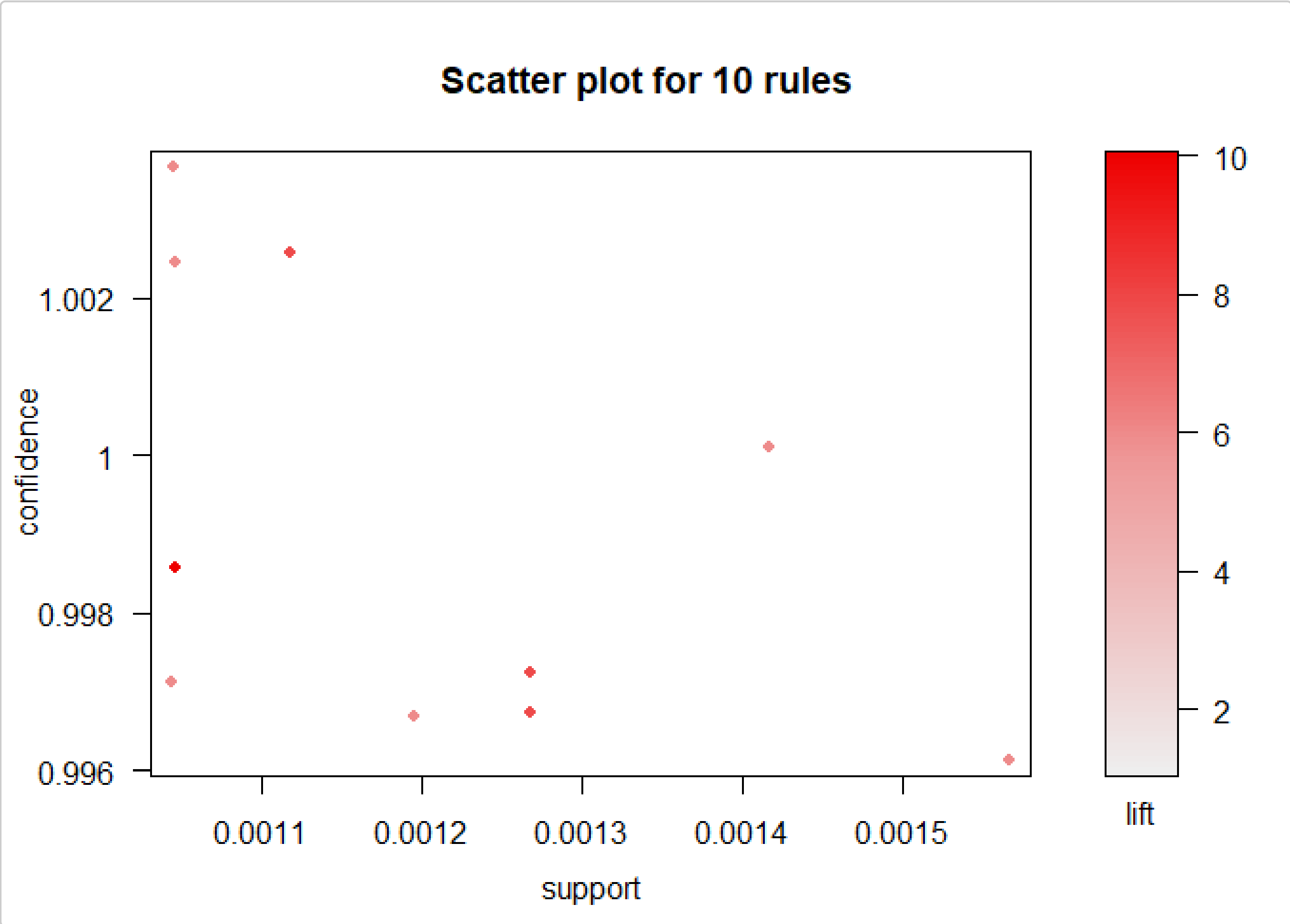
```
##      lhs                                rhs            support
## [1]  {Curry - Prawn,Onion Chutney}    => {Garlic Naan}   0.001119570
## [2]  {Lamb Haryali,Lamb Shashlick}    => {Bombay Aloo}   0.001268846
## [3]  {Lamb Haryali,Lamb Shashlick}    => {Garlic Naan}   0.001268846
## [4]  {Brinjal Bhajee,Red Sauce}       => {Plain Papadum} 0.001418122
## [5]  {Brinjal Bhajee,Onion Chutney}   => {Plain Papadum} 0.001194208
## [6]  {Brinjal Bhajee,Mint Sauce}      => {Plain Papadum} 0.001567398
## [7]  {Mint Sauce,Pathia}              => {Plain Papadum} 0.001044932
## [8]  {Dupiaza,Onion Chutney}          => {Plain Papadum} 0.001044932
## [9]  {Mint Sauce,Paneer Tikka Masala} => {Plain Papadum} 0.001044932
```

```
## [10] {Saag Bhajee,Saag Paneer}        => {Mushroom Rice} 0.001044932
##      confidence lift      count
## [1]  1           7.798603 15
## [2]  1           7.794066 17
## [3]  1           7.798603 17
## [4]  1           5.917845 19
## [5]  1           5.917845 16
## [6]  1           5.917845 21
## [7]  1           5.917845 14
## [8]  1           5.917845 14
## [9]  1           5.917845 14
## [10] 1          10.013453 14
```

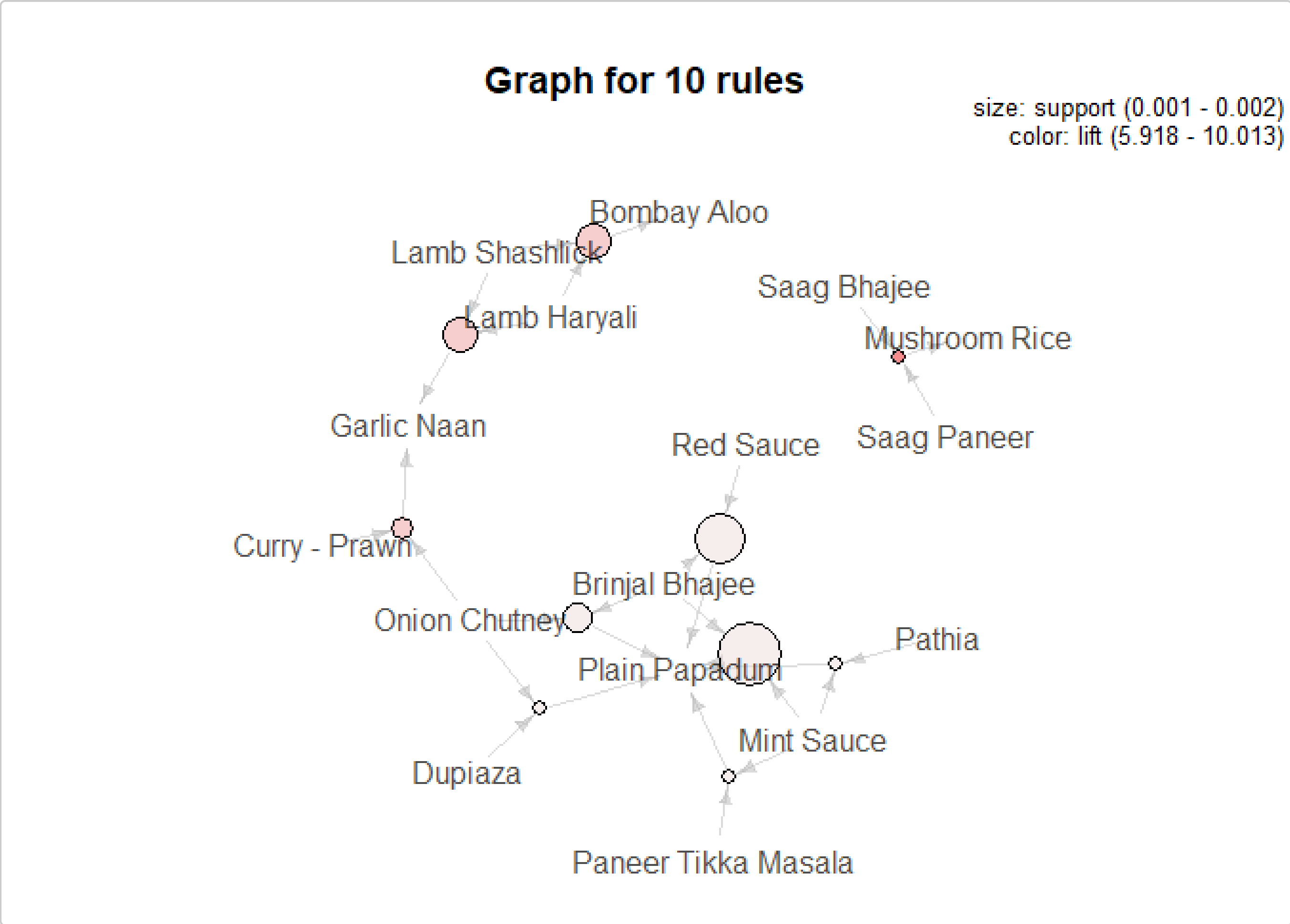
- Top 10 rules shows items on left hand side and the associated items on right hand side with support, confidence and lift

Let us plot the top 10 rules.

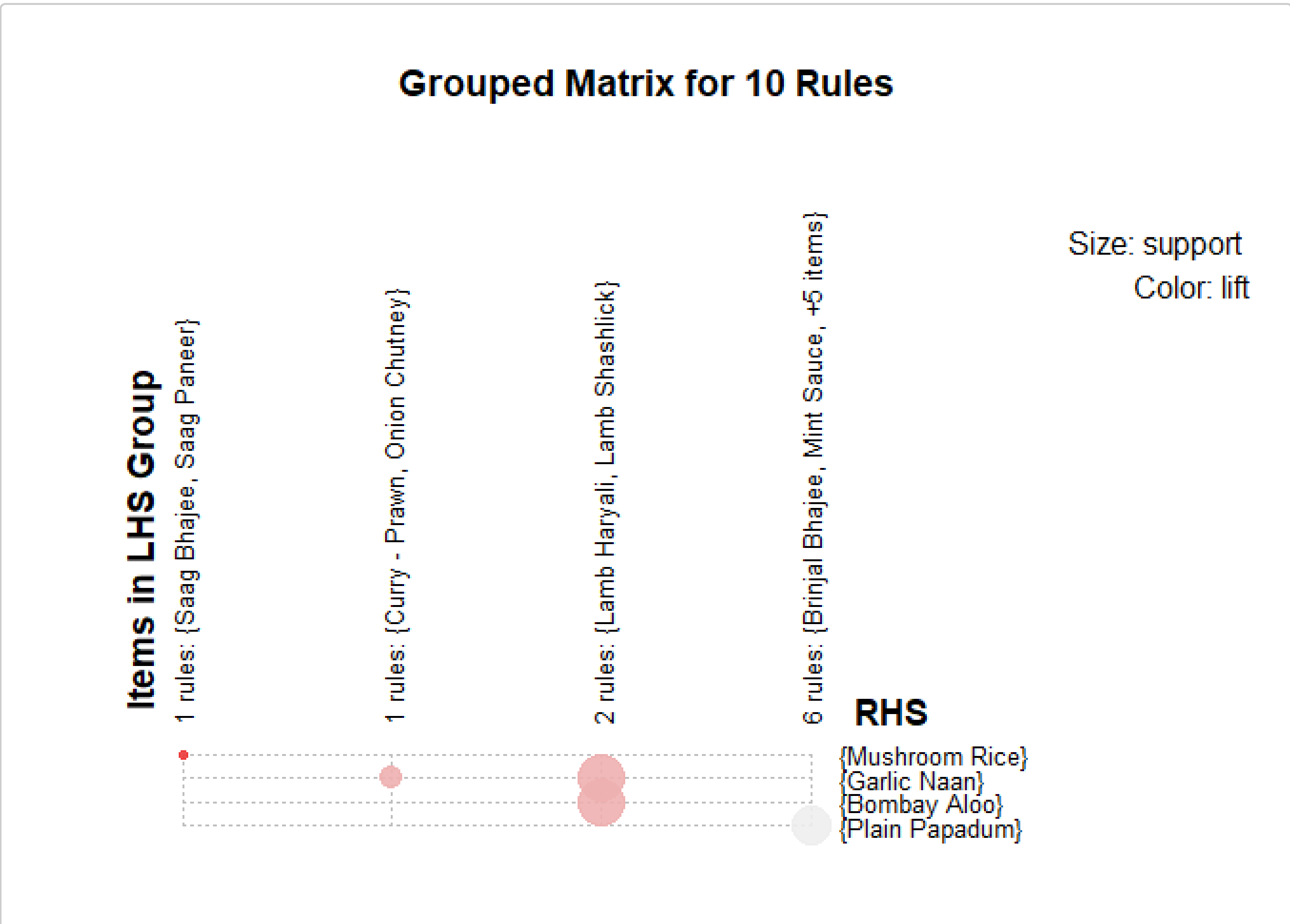
```
TopRules <- Rules[1:10]
plot(TopRules)
```




```
plot(TopRules, method="graph")
```




```
plot(TopRules, method = "grouped")
```



If you like Indian cuisine, you may not find the plots surprising. I truly found the associations interesting.

Market Basket Analysis need not be limited to shopping carts and supermarket shoppers. It can be used to analyze credit card purchases of customers. In Healthcare, it can be used for symptom analysis with which a profile of illness can be better identified.

R-Markdown