Analysing Market Basket

Association Rule

Dr Akhter Raza

Association rule

An unsupervised learning model

Fast Algorithms for Mining Association

By

(R. Agrawal & R. Srikant) 1994.

Association rule

An unsupervised learning model

90% of transactions that purchase bread and butter also purchase milk

Antecedent: bread and butter
Consequent: milk
Confidence factor: 90%

Association rule

- Find all the rules that have “bread” as consequent.
- Find all rules that have “Diet Coke” in the antecedent.
- Find all rules that have “sausage” in the antecedent and “mustard” in the consequent.
- Find all the rules relating items located on shelves A and B in the store.
- Find the “best” (most *confident*) *k* rules that have “margarine” in the consequent.

Association rule help us

- placing things near to things that are associated to each other on the shelves in a physical store
- it's also used for music recommendation systems
- on websites where to place advertisements where to place articles and content in general

Grocery Data

Market Basket analysis in a grocery store so this data set is a quite popular among data scientists it's used to illustrate these techniques and other techniques as well machine learning techniques

Grocery Data

there is no target feature to learn the model

for example if a customer buy butter and jam then there is a high probability of purchasing bread

Grocery Data

This type of data is unstructured because each row represent a transaction but there will not be same number of columns in each transacation

may be some customer bought only 3 items but some took as many as they want so columns are not fixed for each row

Grocery Data – a look

```
citrus fruit,semi-finished bread,margarine,ready soups
tropical fruit,yogurt,coffee
whole milk
pip fruit,yogurt,cream cheese,meat spreads
other vegetables,whole milk,condensed milk,long life bakery product
whole milk,butter,yogurt,rice,abrasive cleaner
rolls/buns
other vegetables,UHT-milk,rolls/buns,bottled beer,liquor (appetizer)
potted plants
whole milk,cereals
tropical fruit,other vegetables,white bread,bottled water,chocolate
citrus fruit,tropical fruit,whole milk,butter,curd,yogurt,flour,bottled water,dishes
beef
```

- Each individual purchase is in a row
- Items are separated by commas
- Data doesn’t organized in tabular form
- 5 col in row 1, 3 in row 2, and 1 col in row3

Grocery Data – a look

- If we want to keep this data in table then we need a grid having 169 columns and 9835 rows
- In 169 columns most of the columns remains blanks and waste memory
- A sample table is represented in the following slide

Grocery Data – a look

Tr. #	milk	butter	drink	rice	…	…	Item 169
1	1	0	0	1			0
2	0	1	1	0			0
3	1	1	1	1			1
…							
…							
9835	1	0	0	1			0

- 1 indicates the person bought that item
- 0 indicates the person didn't buy it
- Most cells contains 0 and waste memory

Sparse Matrix

- It saves the memory to keep such data that has most of the cells 0's
- Sparse doesn't keep 0's if that item not purchased but it remain blank
- It saves memory
- It gives us solution to make structure of unstructured data
- read.csv cant be used b/c it mixes the data

Sparse Matrix

Two methods of representation of a sparse matrix

- Triplet representation
- Linked list representation

Triplet representation

we consider only non-zero values along with their row and column index values.

0^{th} row stores total rows, total columns and total non-zero values in matrix

Triplet representation

For example, consider a matrix of size 5 X 6 containing 6 number of non-zero values. This matrix can be represented as shown in the image on next slide...

Triplet representation

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	2	2
3	5	5
4	2	2

Linked List representation

Tr#	Start
1	1
2	5
3	
4	
5	
9835	

rowId	item	Next
1	Citrus fruit	2
2	Semi finished bread	3
3	Margrine	4
4	Ready soup	0
5	Tropical fruit	6
6	Yogurt	7
7	coffee	0
…		
…		
…		

Grocery Data

Sparse matrix is used to handle the data
arules package is needed for analysis

```
install.packages("arules")
require("arules")
```

Grocery Data

Groceries data set is the part of arules package

```
data("Groceries")
Groc  <-  Groceries
```

if you have your own data set then read it using following command

```
Groc  <-  read.transactions("groceries.csv",
Sep = ",")
```

Grocery Data

Groc

```
> Groc
transactions in sparse format with
 9835 transactions (rows) and
 169 items (columns)
>
```

head(Groc)

```
> head(Groc)
transactions in sparse format with
 6 transactions (rows) and
 169 items (columns)
>
```

Grocery Data

summary(Groc)

```
> summary(Groc)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns             soda
            2513             1903             1809             1715

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14
2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77
  26   27   28   29   32
   1    1    1    3    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   4.409   6.000  32.000

includes extended item information - examples:
       labels  level2           level1
1 frankfurter sausage meat and sausage
2     sausage sausage meat and sausage
3  liver loaf sausage meat and sausage
> |
```

First three rows in groceries

inspect(Groc[1:3])

```
> inspect(Groc[1:3])
    items
[1] {citrus fruit,semi-finished bread,margarine,ready soups}
[2] {tropical fruit,yogurt,coffee}
[3] {whole milk}
```

 # association rule is called support how frequently an item occur in our data

itemFrequency(Groc[,1])

```
> itemFrequency(Groc[,1])
frankfurter
 0.05897306
>
```

Finding occurrences of First item

itemFrequency(Groc[,1])

```
> itemFrequency(Groc[,1])
frankfurter
 0.05897306
>
```

Total item are 9835 in this data set then

Frankfurter occurs 0.0589*9835 = 580 times

The item Frankfurter occurs 580 times in our data

Probabilities and Frequencies

First four items are

```
> itemFrequency(Groc[,1:4])
frankfurter     sausage liver loaf         ham
0.058973055 0.093950178 0.005083884 0.026029487
```

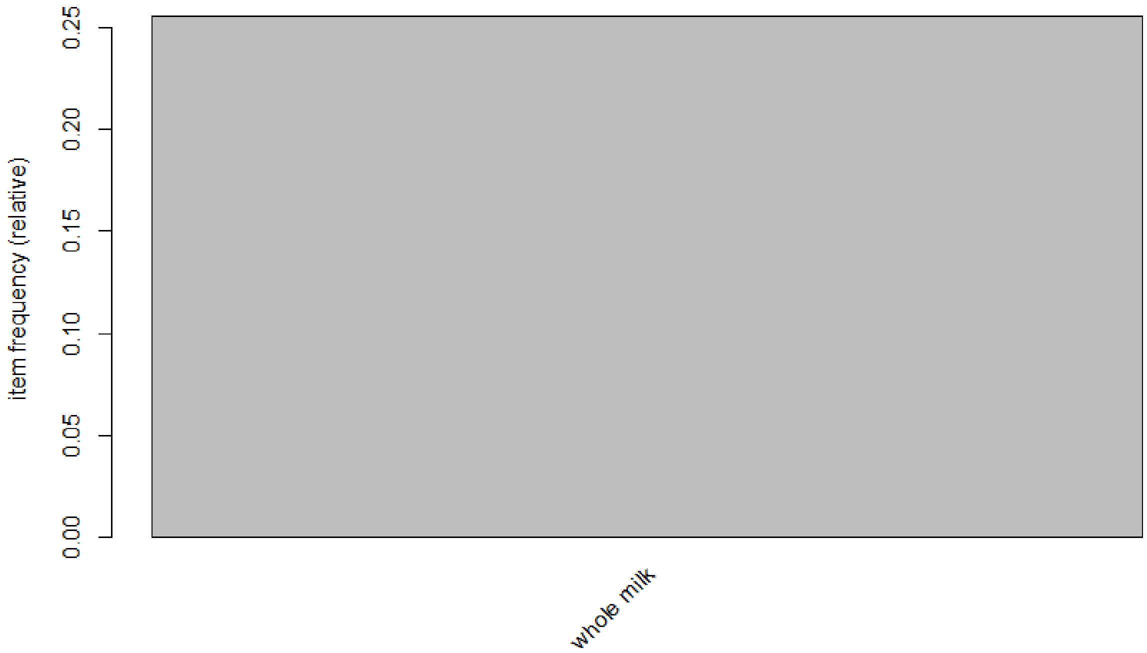
Frequencies of first four items

```
> f <- itemFrequency(Groc[,1:4])
> f*9835
frankfurter     sausage liver loaf         ham
        580         924         50         256
```

Item Frequencies

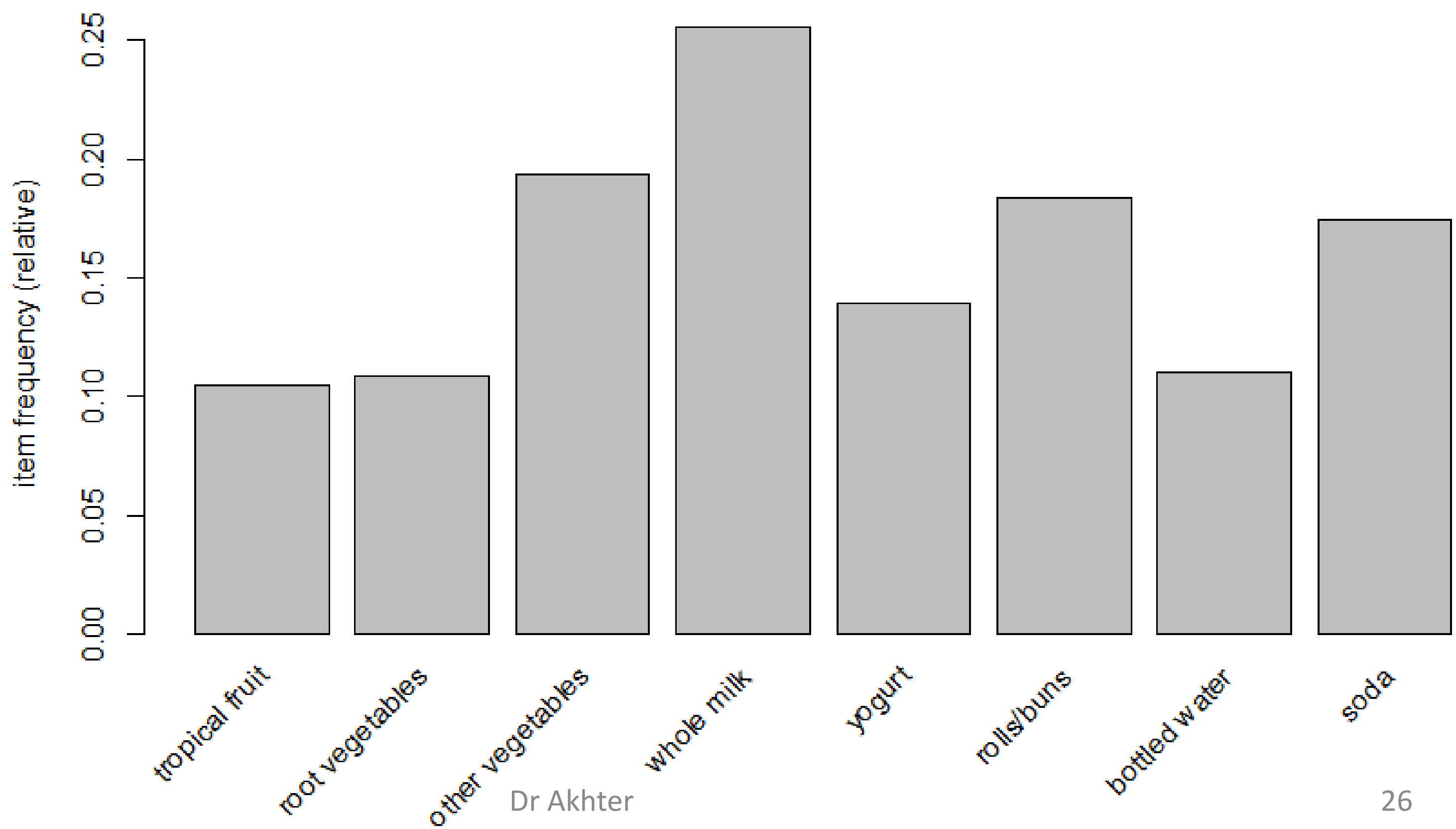
- We want to plot the frequencies

  itemFrequencyPlot(Groc, support = .20)



Item Frequencies

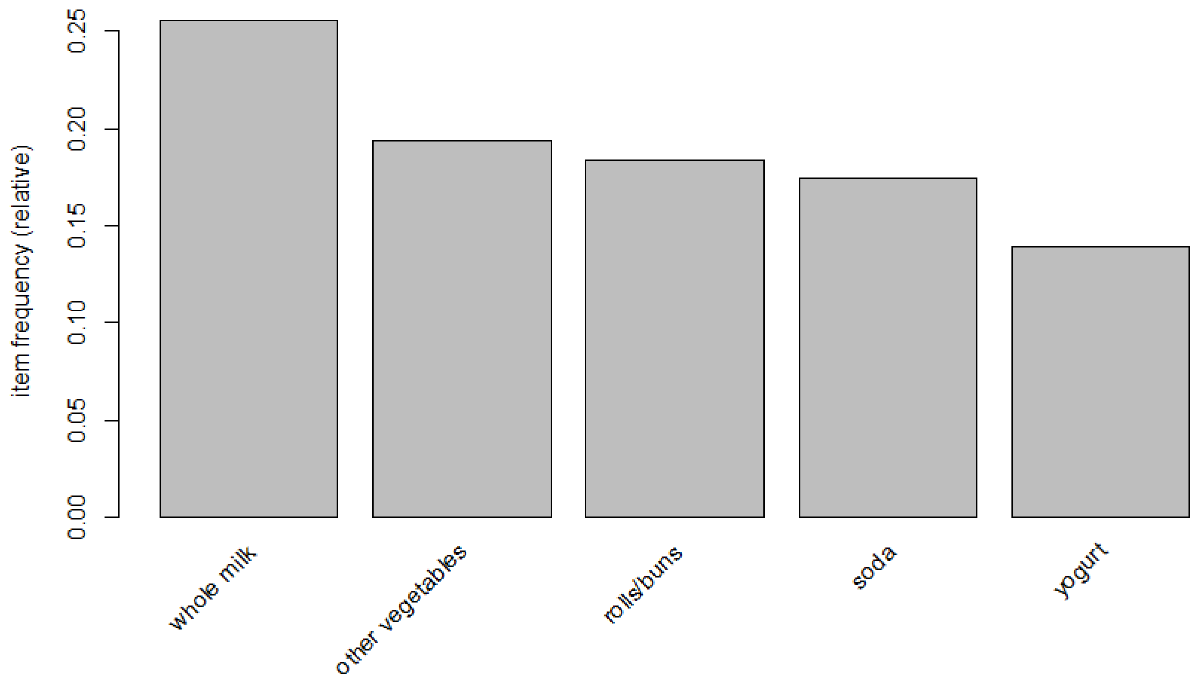
itemFrequencyPlot(Groc, support = .10)



Five Items having highest Support

Draw item frequency for top 5 items

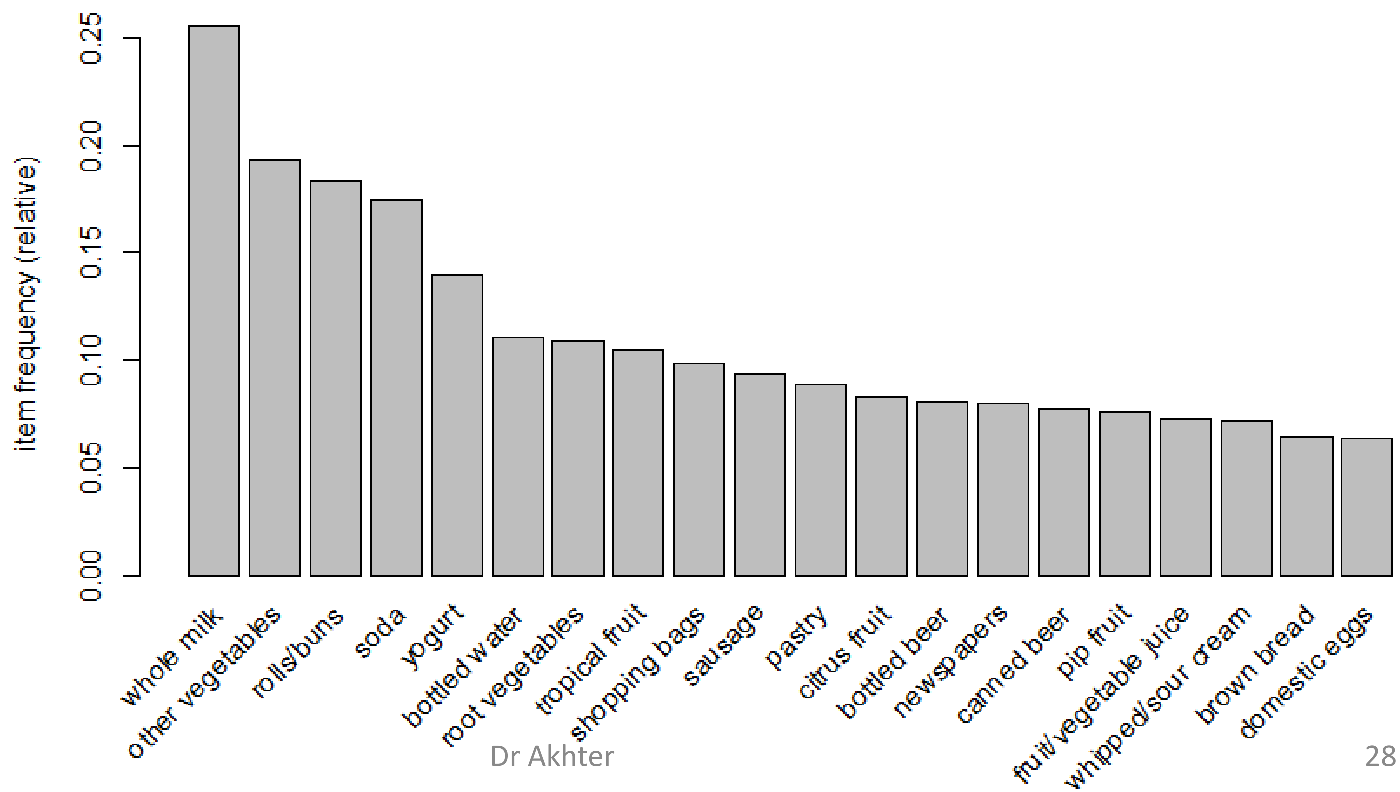
itemFrequencyPlot(Groc, topN = 5)



Twenty Items having highest Support

Top 20 items in the market basket

itemFrequencyPlot(Groc, topN = 20)



Support of an Item

an item that has high support shows up frequently in the data so whole-milk was the one with the highest support it showed up in the most transactions

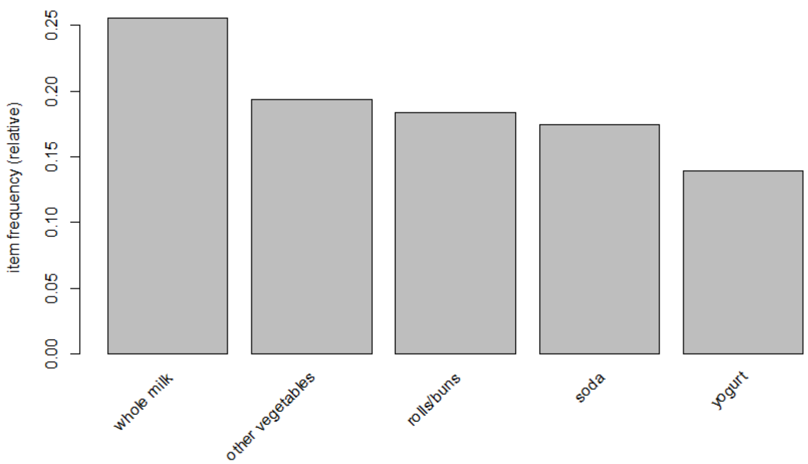
Confidence of an item

- Confidence is a measure of the proportion of transactions where the presence of an item or a set of items results in the presence of another set of items

- basically it's like conditional probability so if I buy item A and item B how likely is it that I'll also buy item C

- so the confidence of A, B implying C is how likely is it that given I bought A and B that I'll also buy C

Support of an Item



Support of whole milk is 0.25 and other vegetables is about 0.18 and so on. The highest support means the item having high sales in market basket. This item shows up in most transactions

Support of an Item

$$Support\ of\ (C|AB) = \frac{Support\ of\ (ABC)}{Support\ of\ (AB)}$$

It is a conditional probability of C given AB means Probability of item C given that the customer bought item A and item B.

Apriori algorithm

```
# apriori algorithm
# apriori(data_set)
# if we leave other parameters the alogirthm use default values min support=0.1 and min confidence=0.8

m1 <- apriori(Groc, parameter = list(support=0.007, confidence = 0.25, minlen=2))

summary(m1)
```

Item distribution and its summary

```
rule length distribution (lhs + rhs):sizes
  2   3   4
137 214  12

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.656   3.000   4.000

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.007016   Min.   :0.2500   Min.   :0.9932   Min.   : 69.0
 1st Qu.:0.008134   1st Qu.:0.2962   1st Qu.:1.6060   1st Qu.: 80.0
 Median :0.009659   Median :0.3551   Median :1.9086   Median : 95.0
 Mean   :0.012945   Mean   :0.3743   Mean   :2.0072   Mean   :127.3
 3rd Qu.:0.013777   3rd Qu.:0.4420   3rd Qu.:2.3289   3rd Qu.:135.5
 Max.   :0.074835   Max.   :0.6389   Max.   :3.9565   Max.   :736.0

mining info:
 data ntransactions support confidence
 Groc          9835   0.007       0.25
>
```

Apriori model Summary

```
> summary(m1)
set of 363 rules

rule length distribution (lhs + rhs):sizes
  2   3   4
137 214  12
```

- Total number of rules 363
- 137 rules bases on two items means if you buy item A you also buy item B
- 214 rules bases on 3 items means if you buy item A and B you also buy item C
- 12 rules bases on 4 items means if you buy item A, B and C you also buy item D

Apriori model Summary

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.656   3.000   4.000

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.007016   Min.   :0.2500   Min.   :0.9932   Min.   : 69.0
 1st Qu.:0.008134   1st Qu.:0.2962   1st Qu.:1.6060   1st Qu.: 80.0
 Median :0.009659   Median :0.3551   Median :1.9086   Median : 95.0
 Mean   :0.012945   Mean   :0.3743   Mean   :2.0072   Mean   :127.3
 3rd Qu.:0.013777   3rd Qu.:0.4420   3rd Qu.:2.3289   3rd Qu.:135.5
 Max.   :0.074835   Max.   :0.6389   Max.   :3.9565   Max.   :736.0
```

- Total number of rules 363

Apriori model Summary

```
> inspect(m1)
     lhs                        rhs                  support     confidence lift      count
[1]  {herbs}                 => {root vegetables}    0.007015760 0.4312500  3.9564774  69
[2]  {herbs}                 => {other vegetables}   0.007727504 0.4750000  2.4548739  76
[3]  {herbs}                 => {whole milk}         0.007727504 0.4750000  1.8589833  76
[4]  {processed cheese}      => {whole milk}         0.007015760 0.4233129  1.6566981  69
[5]  {semi-finished bread}   => {whole milk}         0.007117438 0.4022989  1.5744565  70
[6]  {detergent}             => {whole milk}         0.008947636 0.4656085  1.8222281  88
[7]  {pickled vegetables}    => {whole milk}         0.007117438 0.3977273  1.5565650  70
[8]  {baking powder}         => {other vegetables}   0.007320793 0.4137931  2.1385471  72
[9]  {baking powder}         => {whole milk}         0.009252669 0.5229885  2.0467935  91
[10] {flour}                 => {whole milk}         0.008439248 0.4853801  1.8996074  83
```

- Out of all rules first 10 displayed here

Apriori Summary 2 rules

```
> # to show firtst two rules use inspect command
> inspect(m1[1:2])
    lhs        rhs                 support     confidence lift     count
[1] {herbs} => {root vegetables}  0.007015760 0.43125    3.956477 69
[2] {herbs} => {other vegetables} 0.007727504 0.47500    2.454874 76
```

- Customers who buy herbs is likely to buy root vegetables having support of 0.007 and confidence of 43% and lift of 3.96
- Higher the lift mean higher chances of purchasing root vegetables with herbs
- Total such rules are 69 means out of total customers 69 chooses root vegs with herbs

Apriori inspect

You can sort the output of inspect by any of four features either support, confidence, lift or count

```
inspect(sort(m1, by="lift")[1:4])

inspect(sort(m1, by="support")[1:4])

inspect(sort(m1, by="confidence")[1:4])

inspect(sort(m1, by="count")[1:4])
```

Output sort by lift

```
> inspect(sort(m1, by="lift")[1:4])
    lhs                           rhs                    support     confidence lift     count
[1] {herbs}                    => {root vegetables}      0.007015760 0.4312500  3.956477 69
[2] {berries}                  => {whipped/sour cream}   0.009049314 0.2721713  3.796886 89
[3] {tropical fruit,other      => {root vegetables}      0.007015760 0.4107143  3.768074 69
     vegetables,whole milk}

[4] {beef,other vegetables}    => {root vegetables}      0.007930859 0.4020619  3.688692 78
```

Customers who bought barries will likely to buy whipped and or sour cream with a lift of 3.8

Whipped and cream shows up 3.8 times more in general transactions

These rules help in grocery setting in shelves