Sane's STATS Academy of Statistics

Market Basket Analysis

Association Rules Mining: Apriori Algorithm

June 9, 2018

A Customer’s Basket

- If a customer buys bananas and she buys apple then she buys a fruit beverage also.
- If its a late noon time and a customer buys coconut biscuits then he also buys chips.

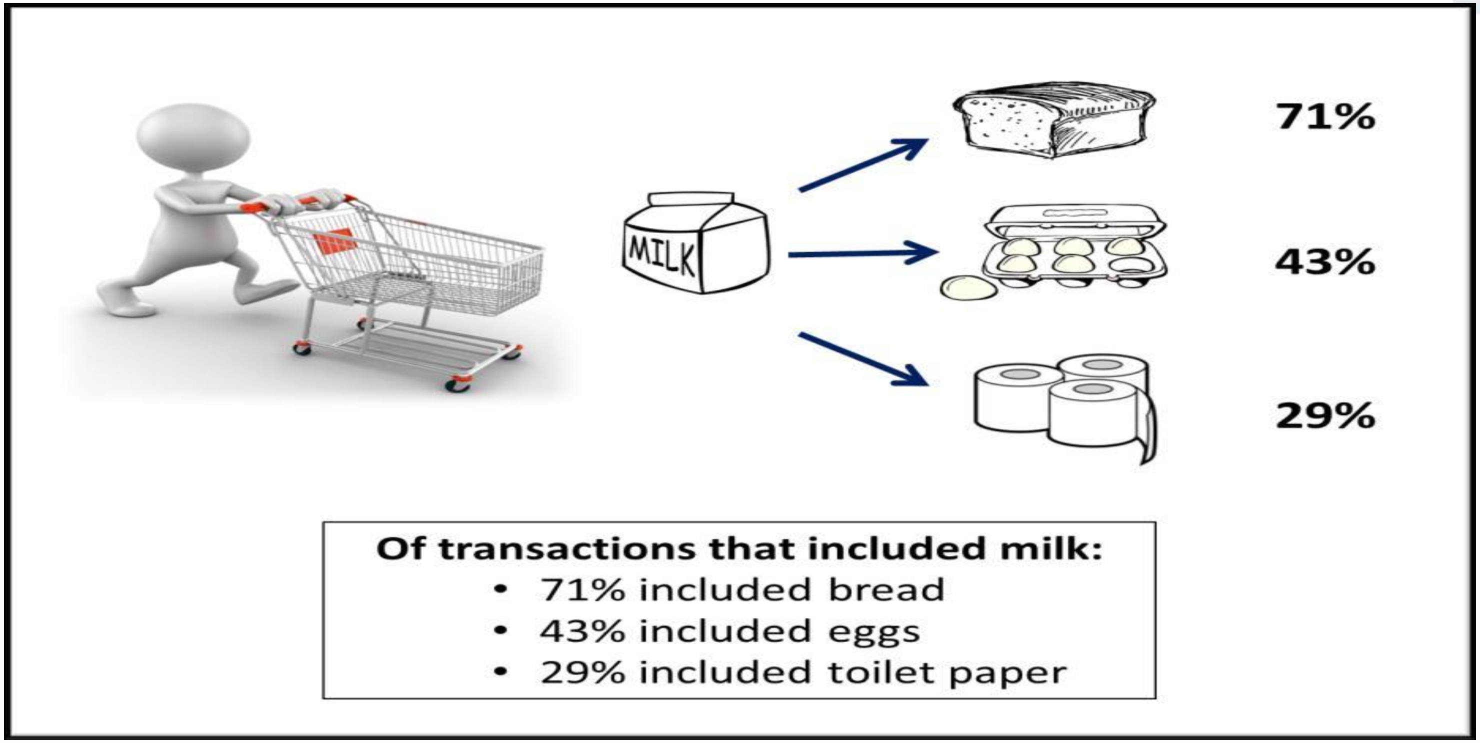
Association Rules

- Association rules provide information of this type in the form of “if–then” statements.
- These rules are computed from the data.

CUSTOMERS WHO BOUGHT THIS ITEM:
ALSO BOUGHT:
piccolo

Generating Rules

- Examine all possible rules between items in an if–then format, and select only those that are most likely to be indicators of true dependence.



Of transactions that included milk:
- 71% included bread
- 43% included eggs
- 29% included toilet paper

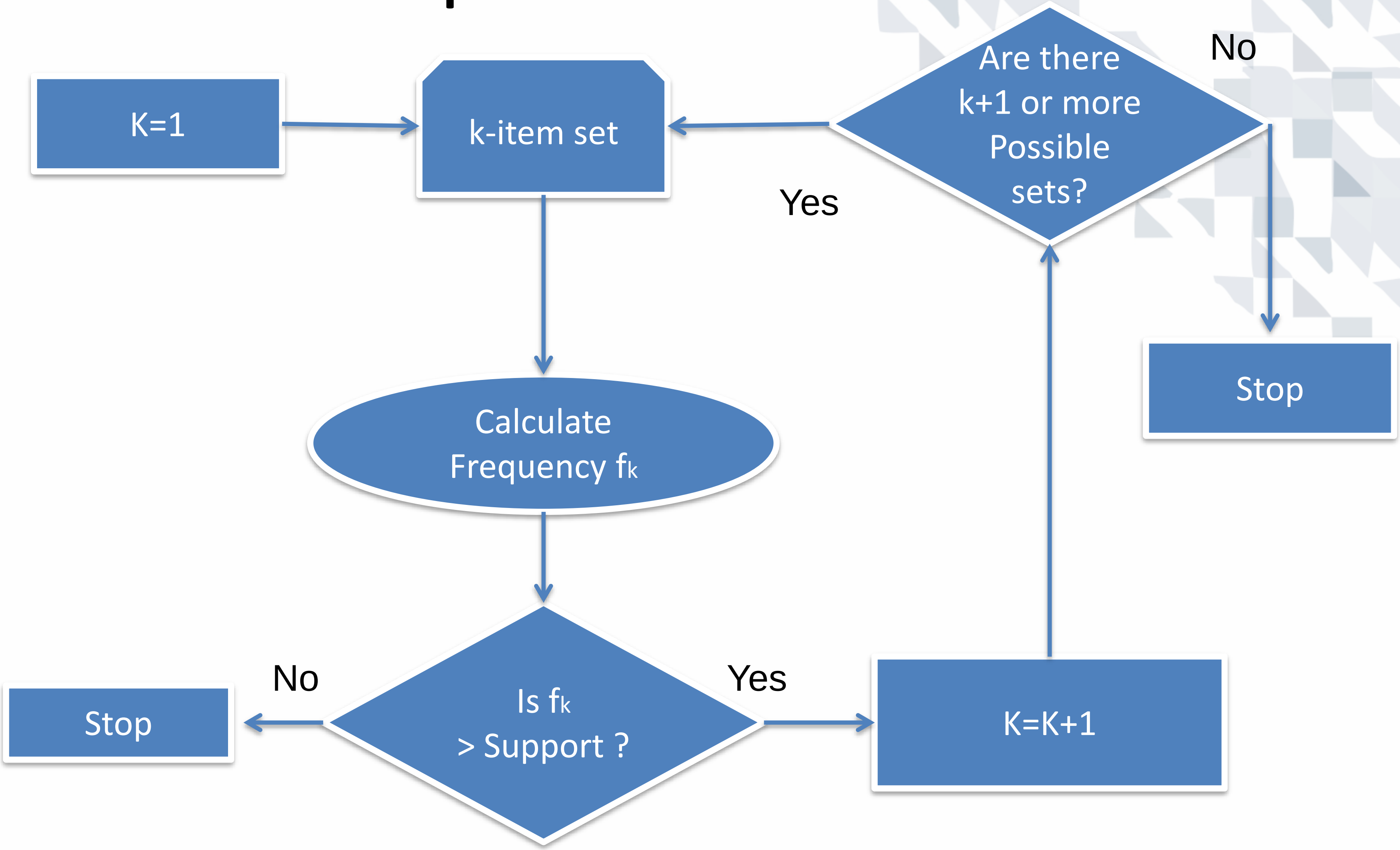
Apriori Algorithm

- Generate frequent item sets with just one item (one-item sets)
- Recursively generate frequent item sets with two items, then with three items, and so on, until we have generated frequent item sets of all sizes.
- Count, for each item, how many transactions in the database include the item.

Apriori Algorithm

- These transaction counts are the supports for the one-item sets.
- We drop one-item sets that have support below the desired minimum support to create a list of the frequent one-item sets.
- To generate frequent two-item sets, we use the frequent one-item sets.

Apriori Flow Chart



Apriori in R

- R function apriori() of package *arules* uses apriori algorithm

Syntax : apriori(data, parameter, ...)

Where

data : object of class *transactions*

parameter : object of class *APparameter*

Strength of Association

- Support
- Confidence
- Lift Ratio

Support

- The support of an item set is the number of transactions that include that item set.
- The support of a rule is the number of transactions that include both the antecedent (if-part) and consequent (then-part) item sets.

Confidence

- Confidence is defined as the measure of trustworthiness associated with each discovered rule

$$Confidence = \frac{Support(if - part\ and\ then - part)}{Support(if - part)}$$

Example

- Consider a database of shopping mall of about 10,00,000 transactions. Out of these transactions, there are 40,000 transactions with purchase of soft toys and hand towel (purchased together) and 24,000 of these transactions include the room freshener purchases.
  - n{Soft Toy, Hand Towel} = 40,000 ,
  - n{Soft Toy, Hand Towel, Room Freshener} = 24,000
- Rule : “If anyone purchases soft toys and hand towel then he/she also purchases room freshener in the same trip” has Support of 24,000 (2.4%) transactions and a confidence of 24,000/40,000 %= 60%.
  - Conf ({Soft Toy, Hand Towel} → {Room Freshener}) = 0.6

Support as Probability

- Support is the (estimated) probability that a transaction selected randomly from the database will contain all items in the if-part and the then-part:
  - P(if-part AND then-part)

Confidence as Probability

- Confidence is the (estimated) conditional probability that a transaction selected randomly will include all the items in the consequent given that the transaction includes all the items in the antecedent.

$$Confidence = \text{P(then−part|if−part)} = \frac{P(if - part\ AND\ then - part)}{P(if - part)}$$

Possible Independence

- If if-part and then-part are independent then the support would be

$$P(if - part\ AND\ then - part) = P(if - part) * P(then - part)$$

- Based on this, the benchmark confidence is defined as

$$P(then - part|if - part) = \frac{P(if - part\ AND\ then - part)}{P(if - part)}$$

$$= \frac{\mathrm{P(if{-}part)} * P(then{-}part)}{P(if{-}part)}$$

$$= P(then - part)$$

Benchmark Confidence

- Benchmark Confidence can be estimated from the data as,

$$Benchmark\ Confidence = \frac{No.of\ transactions\ with\ then - part}{No.of\ transactions\ in\ the\ database}$$

- In shopping mall example if 300,000 transactions are of then-part (room freshener purchases) then benchmark confidence can be calculated as

$$Benchmark\ Confidence = \frac{300000}{1000000} = 0.3$$

Lift Ratio

- The lift ratio is the confidence of the rule divided by the confidence, assuming independence of consequent from antecedent.

$$Lift\ Ratio = \frac{Confidence}{Benchmark\ Confidence}$$

- A lift ratio greater than 1.0 suggests that there is some usefulness to the rule.

- In shopping mall example if 300,000 transactions are of then-part (room freshener purchases) then lift ratio of the said transaction can be calculated as

$$Lift\ Ratio = \frac{Confidence}{Benchmark\ Confidence} = \frac{0.6}{0.3} = 2$$

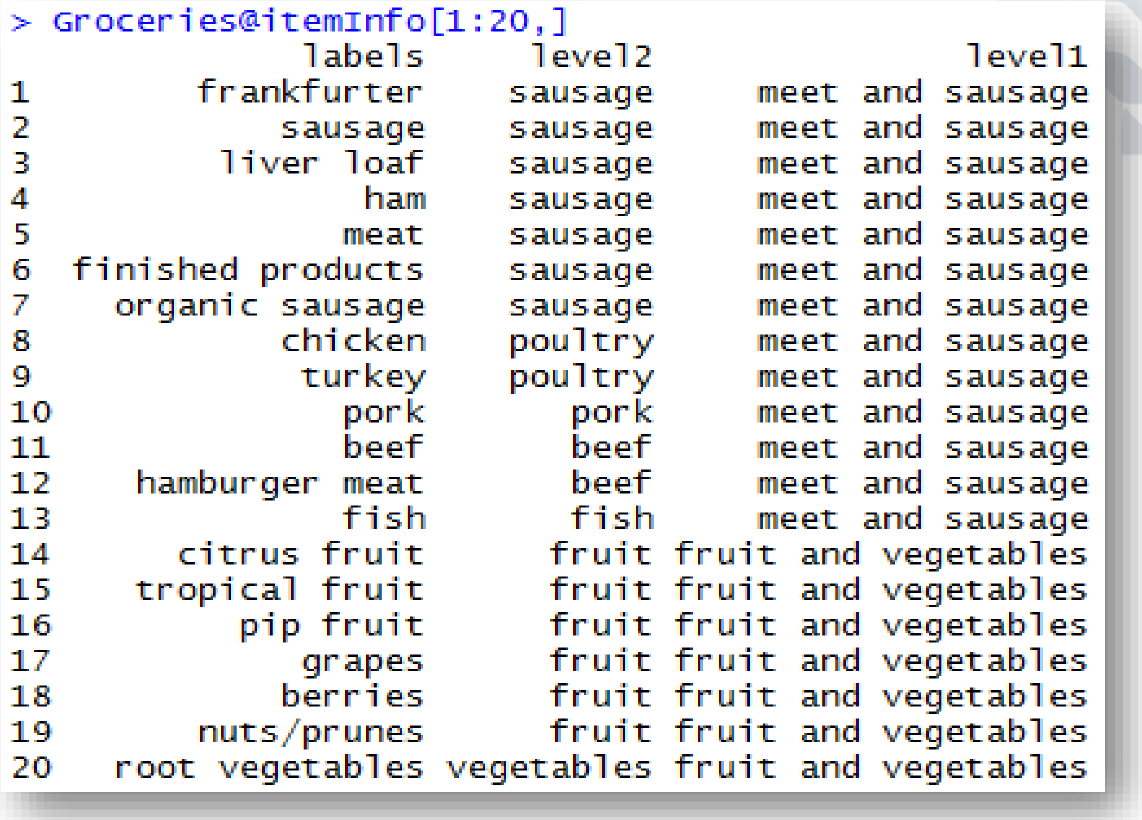
Interpreting the Results

- The support for the rule indicates its impact in terms of overall size as proportion of transactions getting affected.
- If only a small number of transactions are affected, the rule may be of little use.
- The lift ratio indicates how efficient the rule is in finding consequents, compared to random selection.

Example: Transactions in Groceries Store

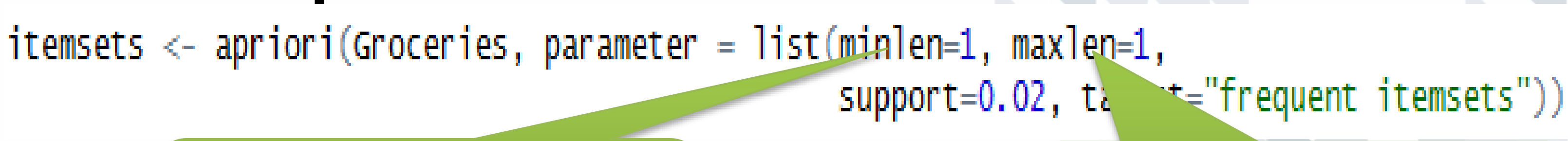
- Groceries dataset is collected from 30 days of point of sale transactions of a grocery store. The dataset can be obtained from package *arules*.
- The class of dataset is transactions, as defined in *arules* package.
- The transactions class contains following components:
  - *itemInfo* : A data frame to store item labels
  - *data* : A binary matrix that indicates which item labels appear in every transaction

itemInfo



```
> Groceries@itemInfo[1:20,]
              labels     level2               level1
1        frankfurter    sausage     meet and sausage
2            sausage    sausage     meet and sausage
3         liver loaf    sausage     meet and sausage
4                ham    sausage     meet and sausage
5               meat    sausage     meet and sausage
6  finished products    sausage     meet and sausage
7    organic sausage    sausage     meet and sausage
8            chicken    poultry     meet and sausage
9             turkey    poultry     meet and sausage
10              pork       pork     meet and sausage
11              beef       beef     meet and sausage
12    hamburger meat       beef     meet and sausage
13              fish       fish     meet and sausage
14      citrus fruit      fruit fruit and vegetables
15    tropical fruit      fruit fruit and vegetables
16         pip fruit      fruit fruit and vegetables
17            grapes      fruit fruit and vegetables
18           berries      fruit fruit and vegetables
19       nuts/prunes      fruit fruit and vegetables
20   root vegetables vegetables fruit and vegetables
```

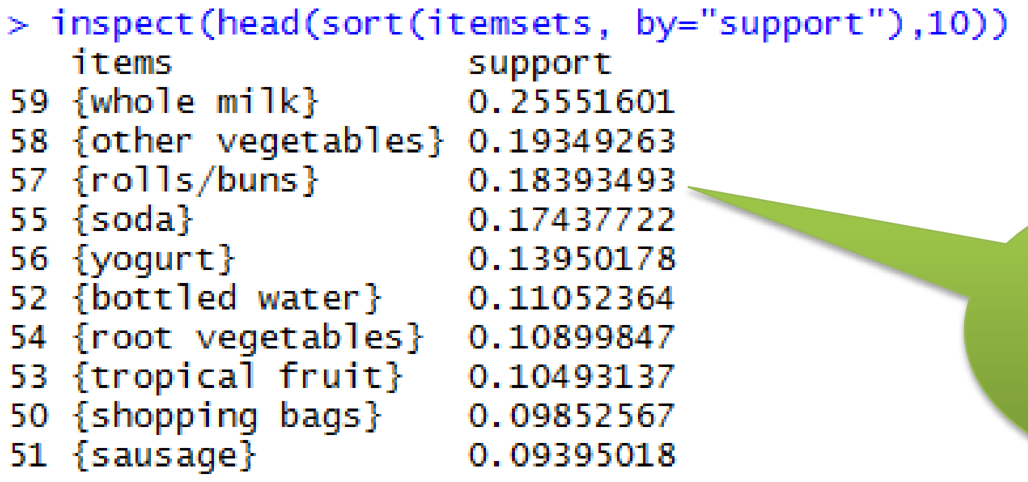
Frequent Itemset Generation



```
itemsets <- apriori(Groceries, parameter = list(minlen=1, maxlen=1,
                                        support=0.02, target="frequent itemsets"))
```

an integer value for the minimal number of items per item set (default: 1)

an integer value for the maximal number of items per item set (default: 10)

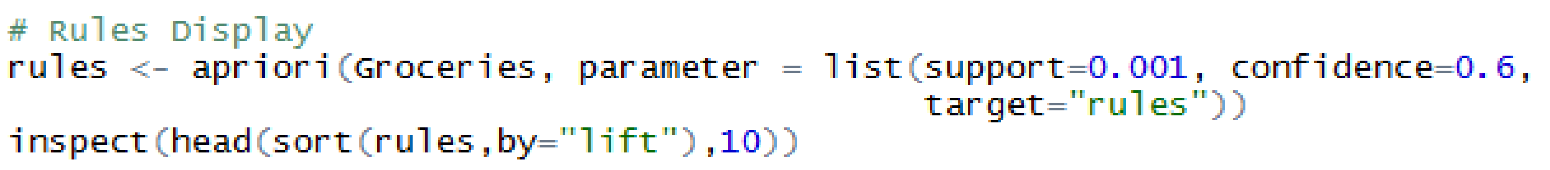


```
> inspect(head(sort(itemsets, by="support"),10))
   items                support
59 {whole milk}         0.25551601
58 {other vegetables}   0.19349263
57 {rolls/buns}         0.18393493
55 {soda}               0.17437722
56 {yogurt}             0.13950178
52 {bottled water}      0.11052364
54 {root vegetables}    0.10899847
53 {tropical fruit}     0.10493137
50 {shopping bags}      0.09852567
51 {sausage}            0.09395018
```

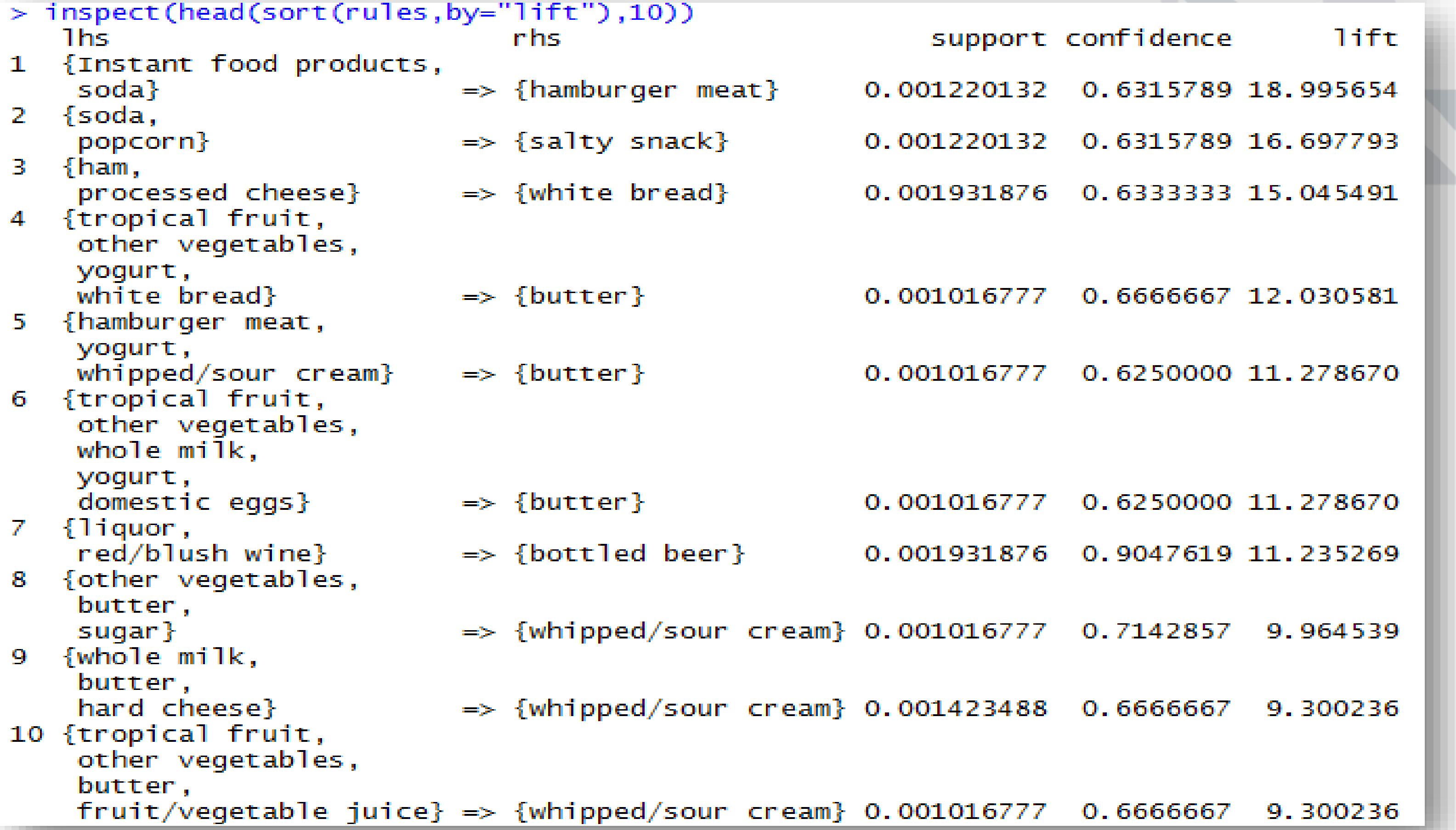
Itemset sorted by support

Sane's STATS Academy of Statistics

Displaying Rules



```
# Rules Display
rules <- apriori(Groceries, parameter = list(support=0.001, confidence=0.6,
                                              target="rules"))
inspect(head(sort(rules,by="lift"),10))
```



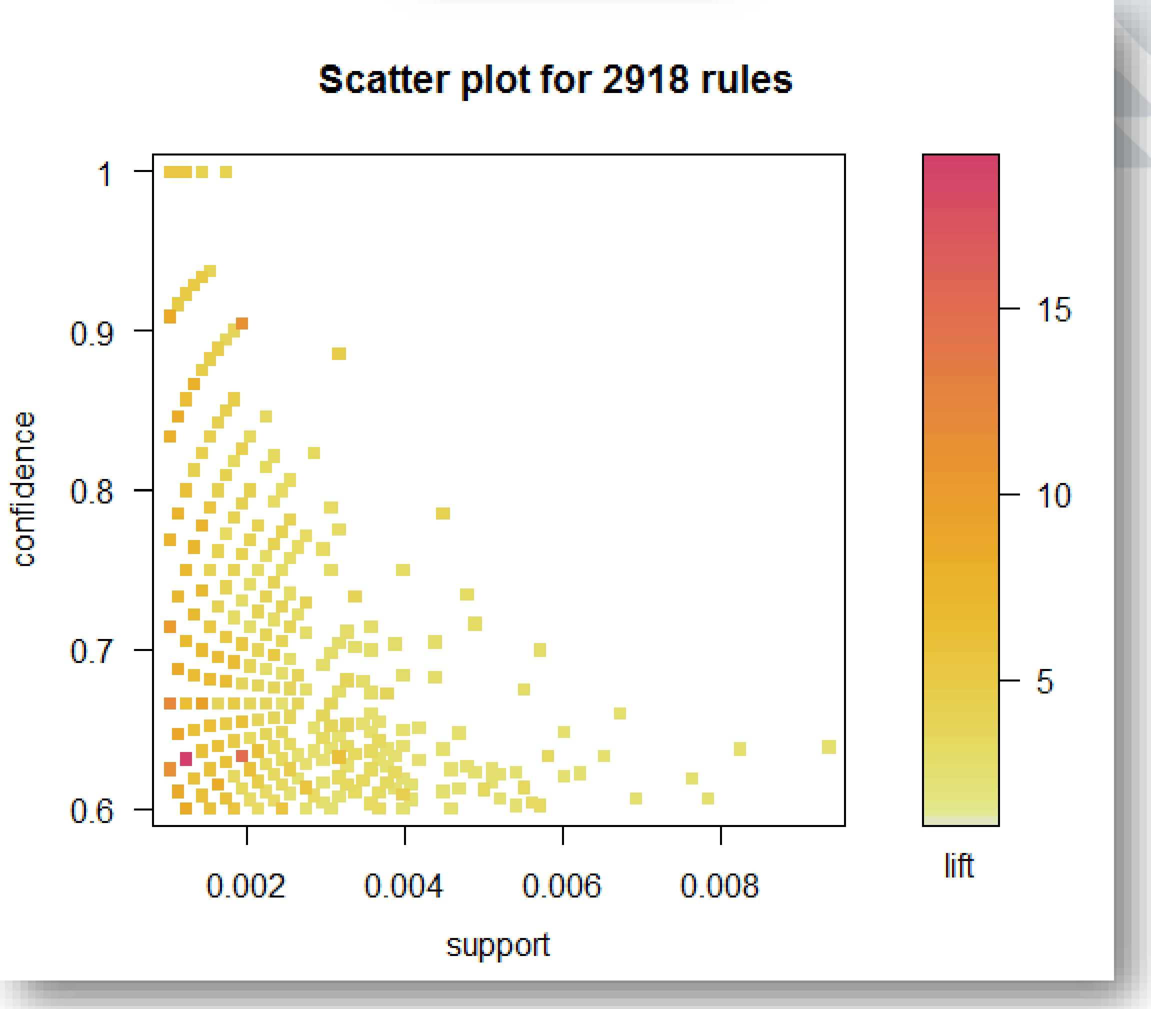
```
> inspect(head(sort(rules,by="lift"),10))
   lhs                          rhs                    support confidence      lift
1  {Instant food products,
    soda}                    => {hamburger meat}     0.001220132  0.6315789 18.995654
2  {soda,
    popcorn}                 => {salty snack}        0.001220132  0.6315789 16.697793
3  {ham,
    processed cheese}        => {white bread}        0.001931876  0.6333333 15.045491
4  {tropical fruit,
    other vegetables,
    yogurt,
    white bread}             => {butter}             0.001016777  0.6666667 12.030581
5  {hamburger meat,
    yogurt,
    whipped/sour cream}      => {butter}             0.001016777  0.6250000 11.278670
6  {tropical fruit,
    other vegetables,
    whole milk,
    yogurt,
    domestic eggs}           => {butter}             0.001016777  0.6250000 11.278670
7  {liquor,
    red/blush wine}          => {bottled beer}       0.001931876  0.9047619 11.235269
8  {other vegetables,
    butter,
    sugar}                   => {whipped/sour cream} 0.001016777  0.7142857  9.964539
9  {whole milk,
    butter,
    hard cheese}             => {whipped/sour cream} 0.001423488  0.6666667  9.300236
10 {tropical fruit,
    other vegetables,
    butter,
    fruit/vegetable juice} => {whipped/sour cream} 0.001016777  0.6666667  9.300236
```

Visualizing Association Rules

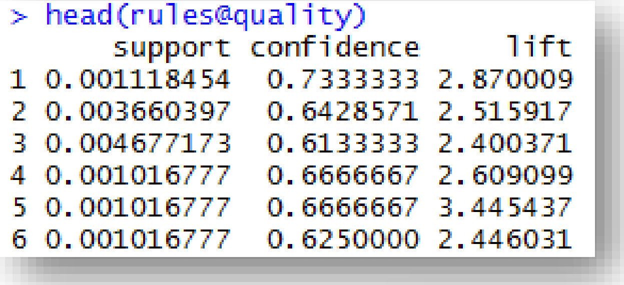
- Package *arulesViz* extends package *arules* with various visualization techniques for association rules and itemsets.
- This package also includes several interactive visualizations for rule exploration.

Visualizing Rules

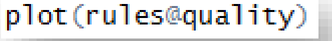
```
plot(rules)
```



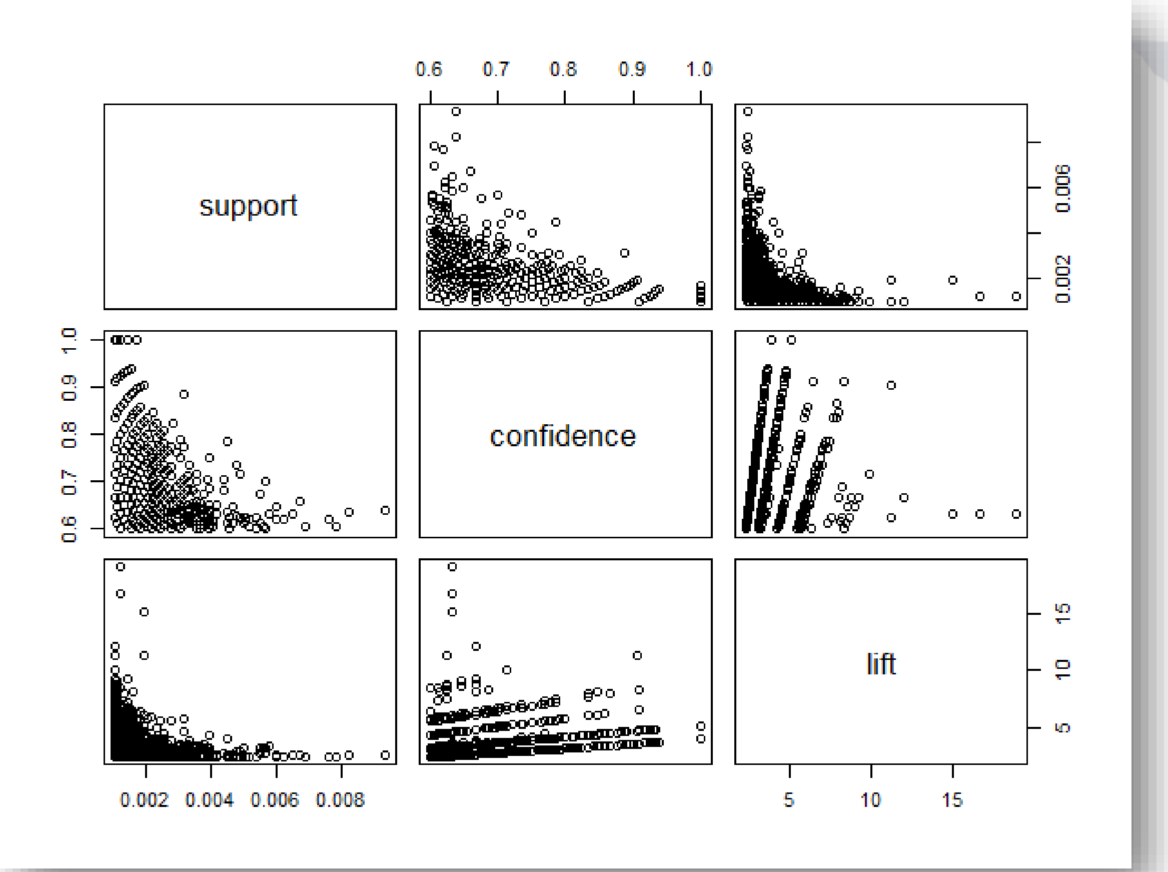
Visualizing Rules



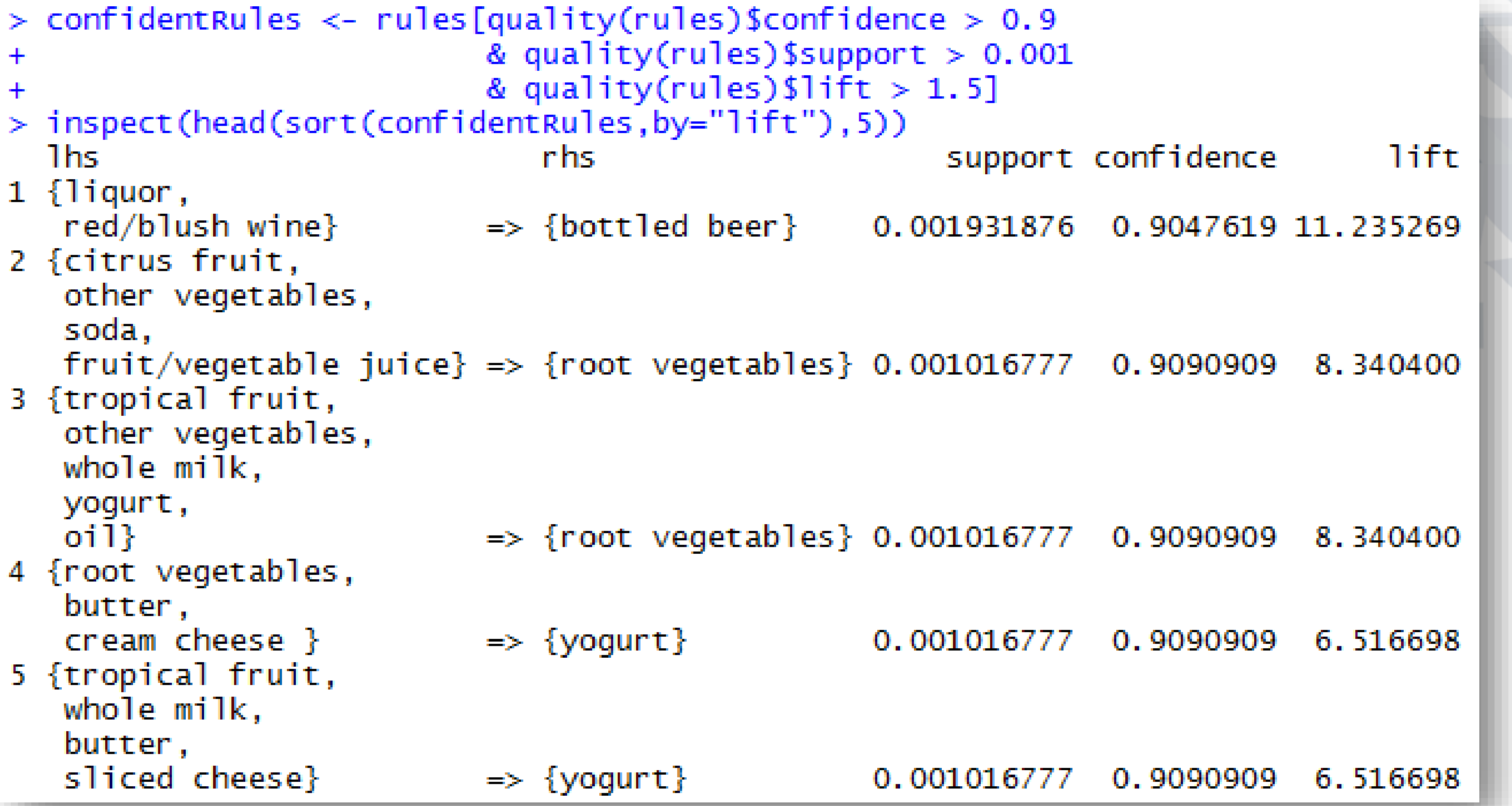
```
> head(rules@quality)
      support confidence     lift
1 0.001118454  0.7333333 2.870009
2 0.003660397  0.6428571 2.515917
3 0.004677173  0.6133333 2.400371
4 0.001016777  0.6666667 2.609099
5 0.001016777  0.6666667 3.445437
6 0.001016777  0.6250000 2.446031
```



```
plot(rules@quality)
```

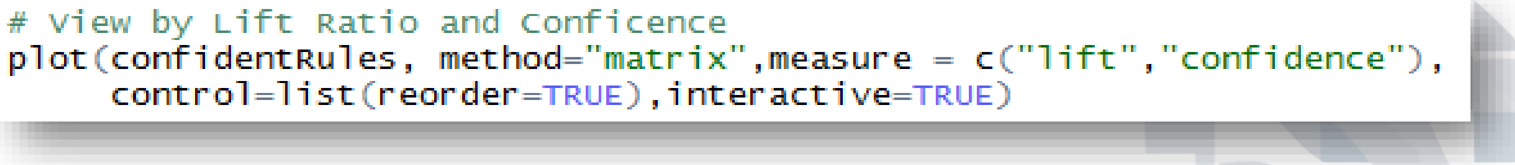


Sub-setting Rules

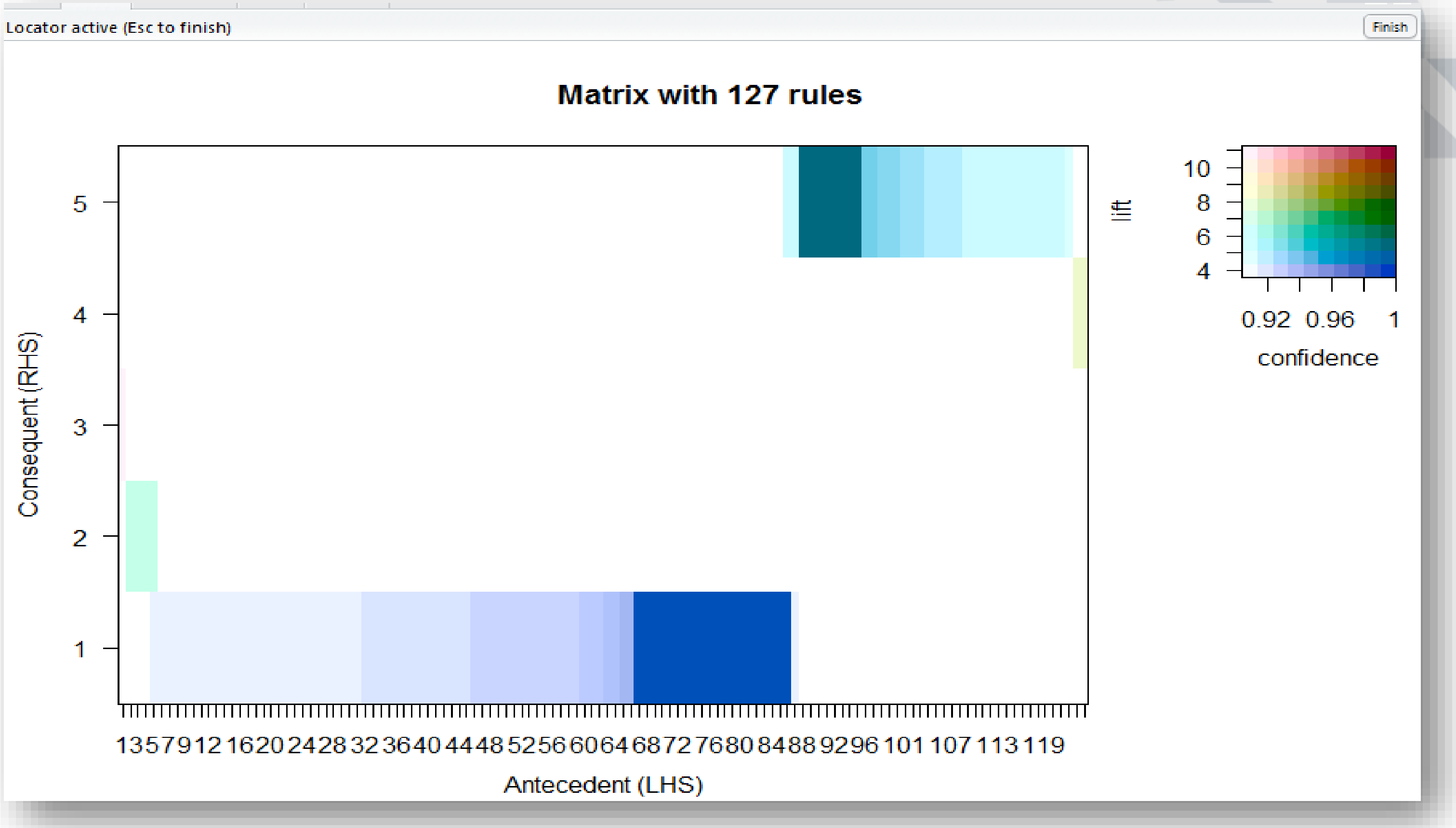


```
> confidentRules <- rules[quality(rules)$confidence > 0.9
+                         & quality(rules)$support > 0.001
+                         & quality(rules)$lift > 1.5]
> inspect(head(sort(confidentRules,by="lift"),5))
  lhs                        rhs                    support confidence      lift
1 {liquor,
   red/blush wine}        => {bottled beer}     0.001931876  0.9047619 11.235269
2 {citrus fruit,
   other vegetables,
   soda,
   fruit/vegetable juice} => {root vegetables} 0.001016777  0.9090909  8.340400
3 {tropical fruit,
   other vegetables,
   whole milk,
   yogurt,
   oil}                   => {root vegetables} 0.001016777  0.9090909  8.340400
4 {root vegetables,
   butter,
   cream cheese }         => {yogurt}          0.001016777  0.9090909  6.516698
5 {tropical fruit,
   whole milk,
   butter,
   sliced cheese}         => {yogurt}          0.001016777  0.9090909  6.516698
```

Interactive Graph

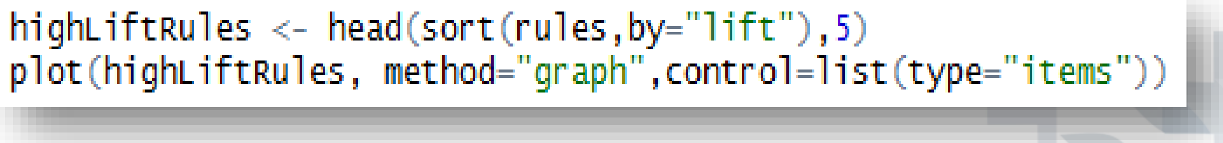


```
# View by Lift Ratio and Confidence
plot(confidentRules, method="matrix",measure = c("lift","confidence"),
     control=list(reorder=TRUE),interactive=TRUE)
```

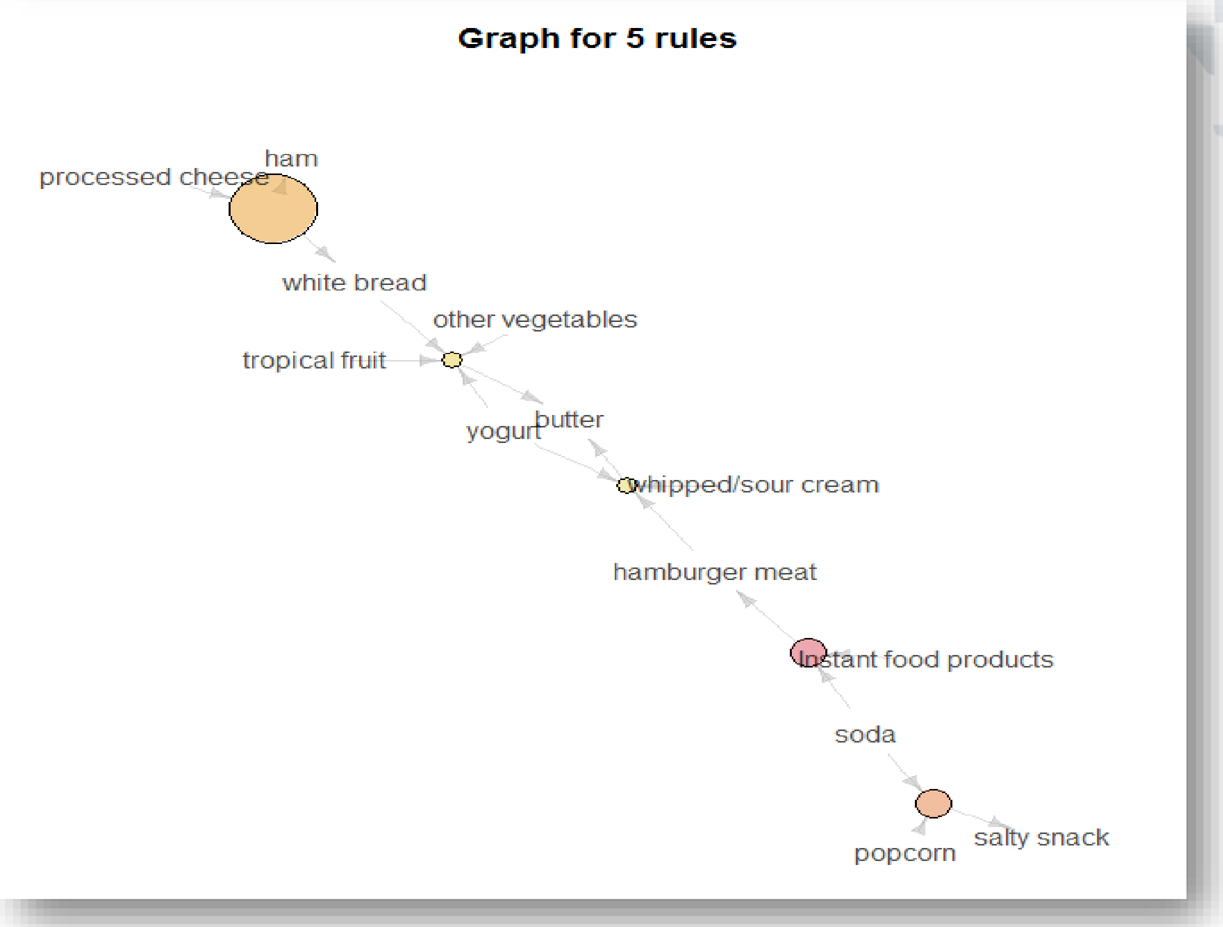


Locator active (Esc to finish)

Visualizing Top Rules



```
highLiftRules <- head(sort(rules,by="lift"),5)
plot(highLiftRules, method="graph",control=list(type="items"))
```



Graph for 5 rules
ham
processed cheese
white bread
other vegetables
tropical fruit
yogurt
butter
whipped/sour cream
hamburger meat
Instant food products
soda
salty snack
popcorn