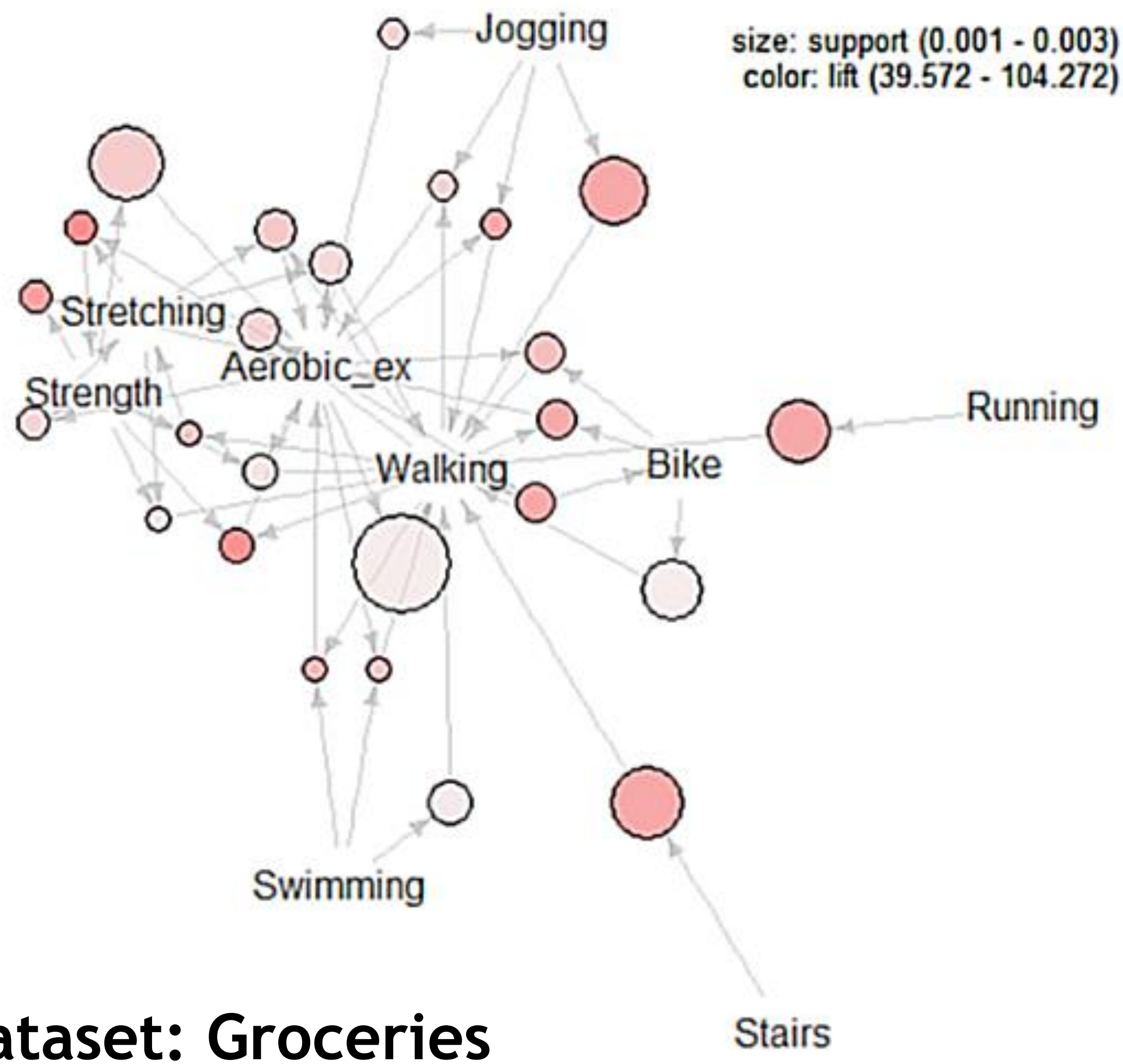
Association Rule Learning



1. Introduction
2. Transaction Dataset: Groceries
3. Tabular Dataset: Titanic

[Reference]
C. Lesmeister, Mastering Machine Learning with R, 2nd ed. (Packt Pub., Birmingham, 2017) Chap. 10.

1. Introduction

• Association rule learning known as **market basket analysis** or **association analysis** is useful for discovering desired information and knowledge hidden in large data sets.
• The uncovered relationships can be represented in the form of **association rules**.

• **Association rules:**
It is a model that identifies how the data items are associated with each other.

• **Structure of rule:**
If(condition) then (result)

If a customer purchases coke, then the customer also purchases orange juice.

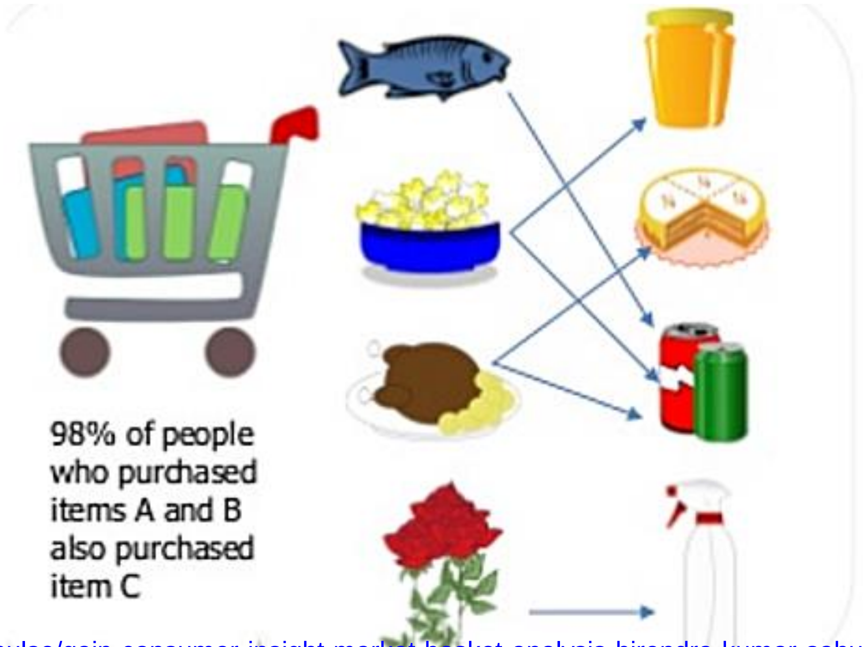


Figure: https://www.linkedin.com/pulse/gain-consumer-insight-market-basket-analysis-birendra-kumar-sahu

Applications

- **Product recommendation** – Amazon’s “customers who bought that, also bought this”
- **Library lending services** – Carrying out proposed borrow and recommended books to improve the efficiency of library management

- **Music recommendations** – music recommender system using association rule mining from music dataset

- **Medical diagnosis** – to find association rules indicating relationships between procedures performed on a patient and the reported diagnoses

- **Content optimization** – like in magazine websites or blogs

Some terminology

Itemset: a collection of one more items in the dataset

Support:
Proportion of the transaction in the data that contain an itemset of interest

Consider $A \Rightarrow B$

$$\text{Support} = \frac{\textit{Number of transactions with both A and B}}{\textit{Total number of transactions}} = P(A \cap B)$$

Confidence:

- Conditional probability that if a person purchases or does *x*, they will purchase or do *y*
- The act of doing *x* is referred to as the antecedent or Left-Hand-Side (LHS), and *y* is the consequence or Right-Hand Side (RHS)

$$\text{Confidence} = \frac{\textit{Number of transactions with both A and B}}{\textit{Total number of transactions with A}} = \frac{P(A \cap B)}{P(A)}$$

Some terminology

Lift: the ratio of the support of **A** occurring together with **B** divided by the probability that **A** and **B** occur if they are independent

$$\text{Expected Confidence} = \frac{Number\ of\ transactions\ with\ B}{Total\ number\ of\ transactions} = P(B)$$

$$\text{Lift} = \frac{Confidence}{Expected\ Confidence} = \frac{P(A \cap B)}{P(A)P(B)}$$

- How many more times A and B occur together than expected.
- Higher the lift, higher chance of A and B occurring together.

2. Transaction Dataset: Groceries

data(**Groceries**) {arules}

The Groceries data set contains 1 month of real-world point-of-sale transaction data from a typical local grocery outlet. The data set contains 9835 transactions and the items are aggregated to 169 categories.

[Data Source] M. Hahsler, K. Hornik, and T. Reutterer (2006) Implications of probabilistic data modeling for mining association rules.

```
> library(arules)
> data(Groceries)
> inspect(Groceries[1:4])
    items
[1] {citrus fruit,
     semi-finished bread,
     margarine,
     ready soups}
[2] {tropical fruit,
     yogurt,
     coffee}
[3] {whole milk}
[4] {pip fruit,
     yogurt,
     cream cheese ,
     meat spreads}
```

Unlike dataframe, *head(Groceries)* does not display the transaction items. We need to use *inspect()*. *inspect(head(Groceries,4)) should also work.*

http://www.salemmarafi.com/code/market-basket-analysis-with-r/

The eclat() takes in a transactions object and gives the most frequent items in the data based the support you provide to the supp arg

(1) Most Frequent Items

The eclat() gives the most frequent items in the data.

eclat(data, parameter, control) {arules}
Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal.

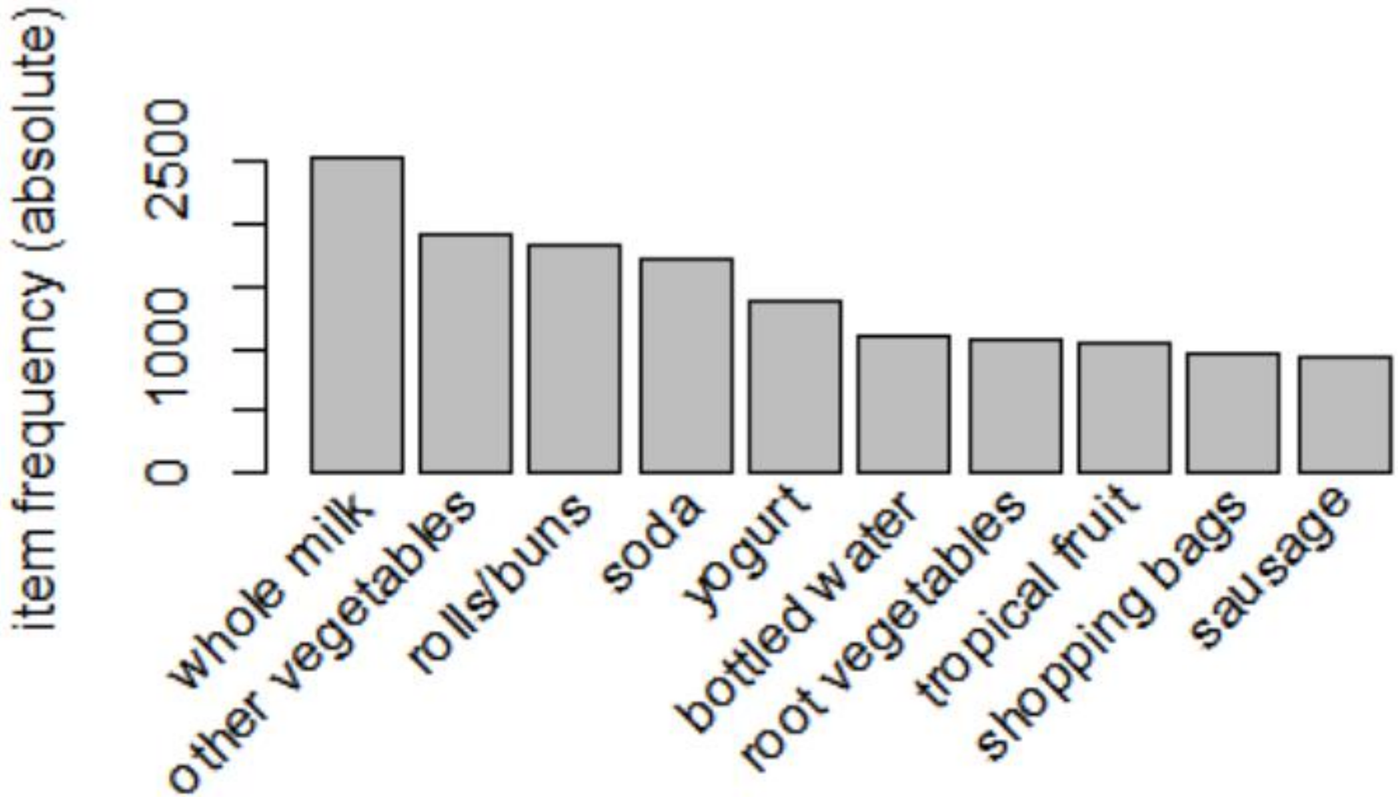
```
> fItem <- eclat (Groceries, parameter=list(supp=0.05,maxlen=15))
> sort_fItem <- sort(fItem, by='support')
> inspect(sort_fItem)
     items                        support
[1]  {whole milk}                 0.25551601
[2]  {other vegetables}           0.19349263
[3]  {rolls/buns}                 0.18393493
[4]  {soda}                       0.17437722
[5]  {yogurt}                     0.13950178
[6]  {bottled water}              0.11052364
[7]  {root vegetables}            0.10899847
[8]  {tropical fruit}             0.10493137
[9]  {shopping bags}              0.09852567
[10] {sausage}                    0.09395018
[11] {pastry}                     0.08896797
[12] {citrus fruit}               0.08276563
[13] {bottled beer}               0.08052872
[14] {newspapers}                 0.07981698
[15] {canned beer}                0.07768175
```

The *eclat()* reports the most frequent transaction items based the support defined. The *maxlen* defines the maximum number of items in each itemset of frequent items.

itemFrequencyPlot(x, type, topN) {arules}

Creates an item frequency bar plot for inspecting the item frequency distribution for objects based on itemMatrix.

```
#Fig 1
itemFrequencyPlot(Groceries, topN=10, type="absolute")
```

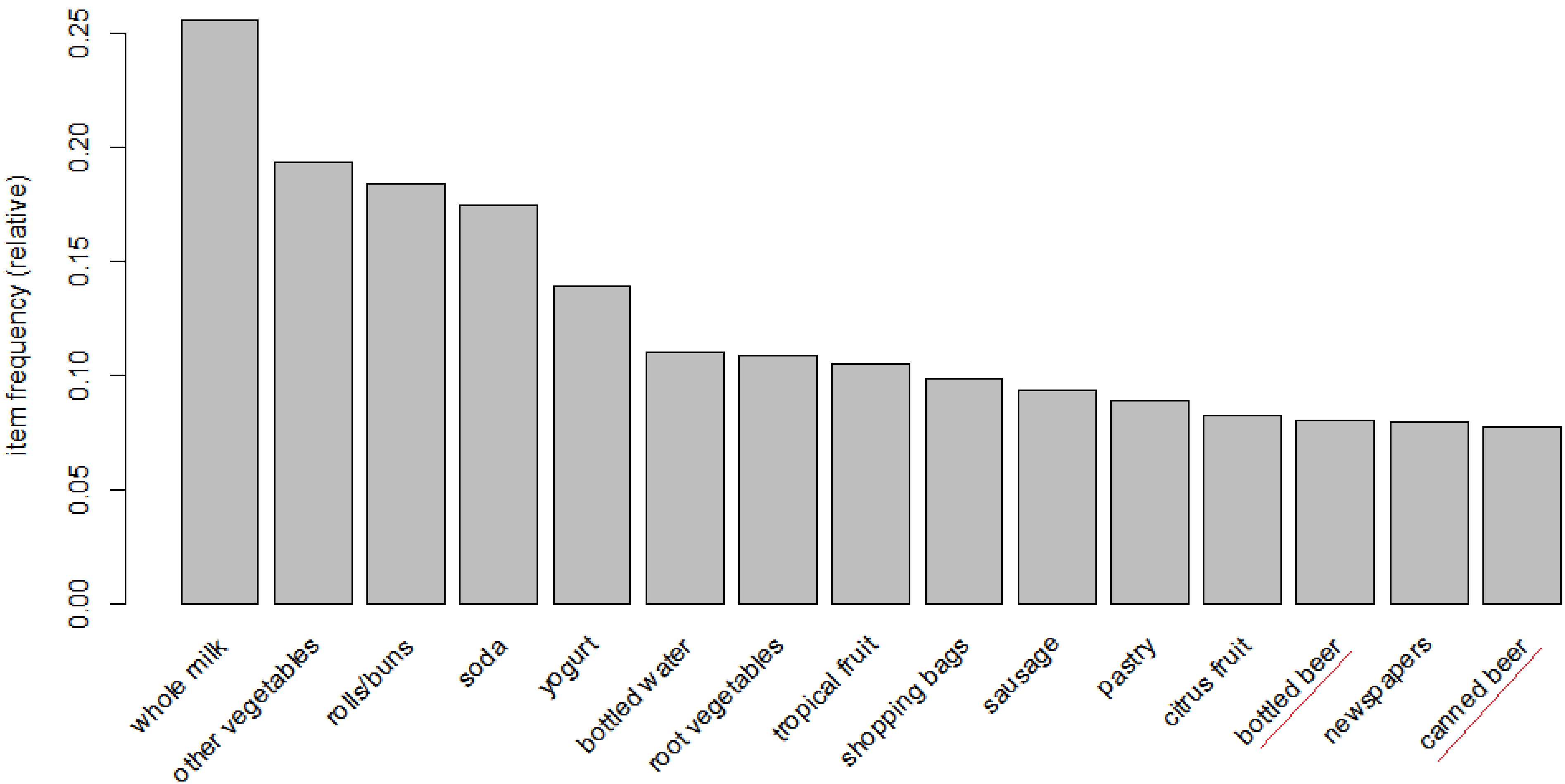


The top item purchased :

whole milk with ~ 2,500 of the 9,835 transactions in the basket.

(1) Most Frequent Items

```
#Fig 2
itemFrequencyPlot(Groceries, topN=15)
```



Here we see that beer shows up as the 13th (bottled beer) and 15th (canned beer).

Through the modeling process, we will use apriori algorithm.

[Reference] https://www.slideshare.net/INSOFE/apriori-algorithm-36054672

apriori(data, parameter = NULL, appearance = NULL, control) {arules}
Mine frequent itemsets, association rules or association hyperedges using the Apriori algorithm. The Apriori algorithm employs level-wise search for frequent itemsets.

```
> rules <- apriori(Groceries,parameter=list(supp=0.001,conf=0.9,maxlen=4))
> rules
set of 67 rules
> options(digits=4)
> lift_rules <- sort(rules, by='lift') #high-lift rules
> inspect(lift_rules[1:2])
    lhs                                       rhs            support  confidence lift   count
[1] {liquor,red/blush wine}                => {bottled beer} 0.001932 0.9048     11.235 19
[2] {root vegetables,butter,cream cheese } => {yogurt}       0.001017 0.9091      6.517 10
```

Relative strength of the rule

Confidence of 0.9048: This rule provides the best overall lift is the purchase of liquor and red/blush wine. **If someone buys liquor and red/blush wine, they are 90.48% likely to buy bottled beer too.**

Lift of 11.235: the items in *LHS (left-hand-side)* and *RHS (right-hand-side)* are 11.235 times more likely to be purchased together compared to the purchases when they are assumed to be "unrelated".

```
> rules <- apriori(Groceries,parameter=list(supp=0.001,conf=0.9,maxlen=4))
> rules
set of 67 rules
> options(digits=4)
> lift_rules <- sort(rules, by='lift') #high-lift rules
> inspect(lift_rules[1:2])
    lhs                                   rhs            support  confidence lift   count
[1] {liquor,red/blush wine}            => {bottled beer} 0.001932 0.9048     11.235 19
[2] {root vegetables,butter,cream cheese } => {yogurt}   0.001017 0.9091      6.517 10
```

Consider $A \Rightarrow B$

$$\text{Support} = \frac{Number\ of\ transactions\ with\ both\ A\ and\ B}{Total\ number\ of\ transactions} = P(A \cap B)$$

$$\text{Confidence} = \frac{Number\ of\ transactions\ with\ both\ A\ and\ B}{Total\ number\ of\ transactions\ with\ A} = \frac{P(A \cap B)}{P(A)}$$

$$\text{Expected Confidence} = \frac{Number\ of\ transactions\ with\ B}{Total\ number\ of\ transactions} = P(B)$$

$$\text{Lift} = \frac{Confidence}{Expected\ Confidence} = \frac{P(A \cap B)}{P(A)P(B)}$$

http://r-statistics.co/Association-Mining-With-R.html

First 5 rules by='confidence' in descending order:

```
> conf_rules <- sort(rules, by='confidence') #high-confidence rules.
> inspect(conf_rules[1:5])
    lhs                        rhs          support confidence  lift count
[1] {rice,
     sugar}                 => {whole milk} 0.001220          1 3.914    12
[2] {canned fish,
     hygiene articles}      => {whole milk} 0.001118          1 3.914    11
[3] {root vegetables,
     butter,
     rice}                  => {whole milk} 0.001017          1 3.914    10
[4] {root vegetables,
     whipped/sour cream,
     flour}                 => {whole milk} 0.001729          1 3.914    17
[5] {butter,
     soft cheese,
     domestic eggs}         => {whole milk} 0.001017          1 3.914    10
```

The rules with **confidence of 1** imply that whenever the LHS item was purchased, the RHS item was also purchased 100% of the time.

crossTable(x, measure, sort) {arules}

Cross-tabulate joint occurrences across pairs of items

```
> #Create a table with Groceries dataset
> dim(Groceries)
[1] 9835  169
> tab <- crossTable(Groceries)
> # Look at first 6 rows and columns
> tab[1:6,1:6]
                  frankfurter sausage liver loaf ham meat finished products
frankfurter               580      99          7  25   32                 3
sausage                    99     924         10  49   52                10
liver loaf                  7      10         50   3    0                 0
ham                        25      49          3 256    9                 2
meat                       32      52          0   9  254                 2
finished products           3      10          0   2    2                64
```

Shoppers only frankfurter : 580 times out of 9,835 transactions

Shoppers frankfurter and sausage : 99

```
> # Specify the rows and columns
> tab['bottled beer','bottled beer']
[1] 792
> tab['bottled beer','canned beer']
[1] 26
```

Transactions of bottled beer: 792

Joint occurrence between bottled beer and canned beer : 26

Transaction ratio: 'bottled beer' / 'red/blush wine'

```
> tab['bottled beer','red/blush wine']
[1] 48
> tab['red/blush wine','red/blush wine']
[1] 189
> 48 / 189  #0.2539683
[1] 0.2539683
```

When someone purchased red/blush wine, they also purchased bottled beer. It’s 25.4%.

Transaction ratio: 'bottled beer' / ‘white wine'

```
> tab['white wine','white wine']
[1] 187
> tab['bottled beer','white wine']
[1] 22
> 22 / 187  #0.1176471
[1] 0.1176471
```

When someone purchased white wine, a joint purchase of bottled beer only happened in 11.8% of the instances.

How to find rules related to given item/s?

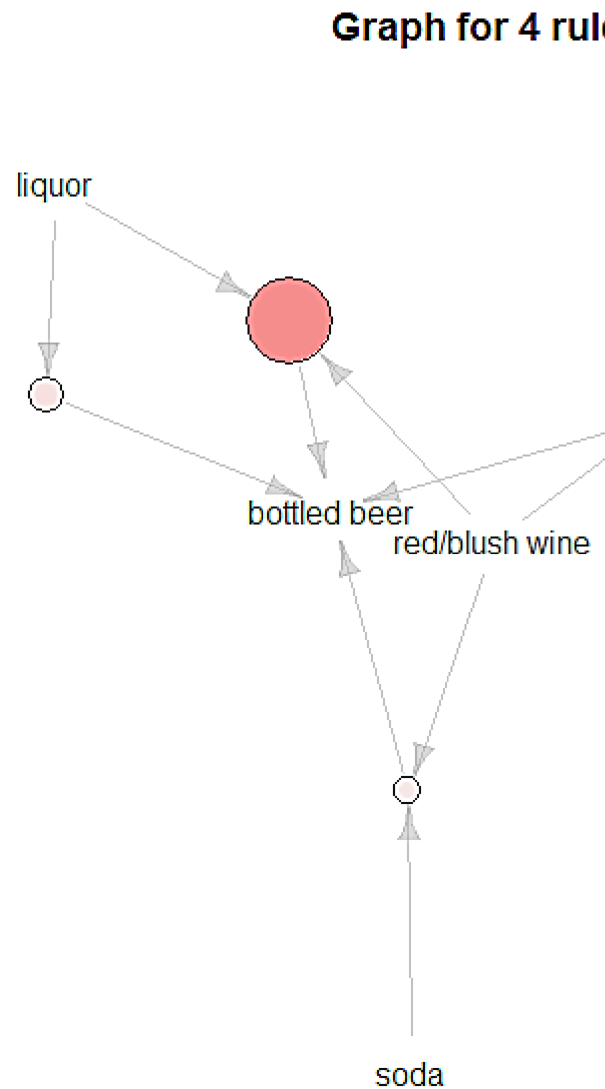
Let's find out what customers had purchased before buying 'bottled beer'. This will help you understand the patterns that led to the purchase of 'bottled beer'.

```
> # get rules that lead to buying 'bottled beer'
> rules <- apriori(Groceries, parameter=list(supp=0.0015,conf=0.3),
+                  appearance=list(default="lhs",rhs='bottled beer'))
> rules
set of 4 rules
> beer_rules <- sort(rules, by='lift')
> inspect(beer_rules)
    lhs                    rhs                support     confidence lift
[1] {liquor,
     red/blush wine}    => {bottled beer} 0.001931876  0.9047619 11.235269
[2] {liquor}            => {bottled beer} 0.004677173  0.4220183  5.240594
[3] {soda,
     red/blush wine}    => {bottled beer} 0.001626843  0.3555556  4.415264
[4] {other vegetables,
     red/blush wine}    => {bottled beer} 0.001525165  0.3061224  3.801407
```

There were only 4 association rules for RHS='bottled beer'.

(3) Visualizing Association Rules

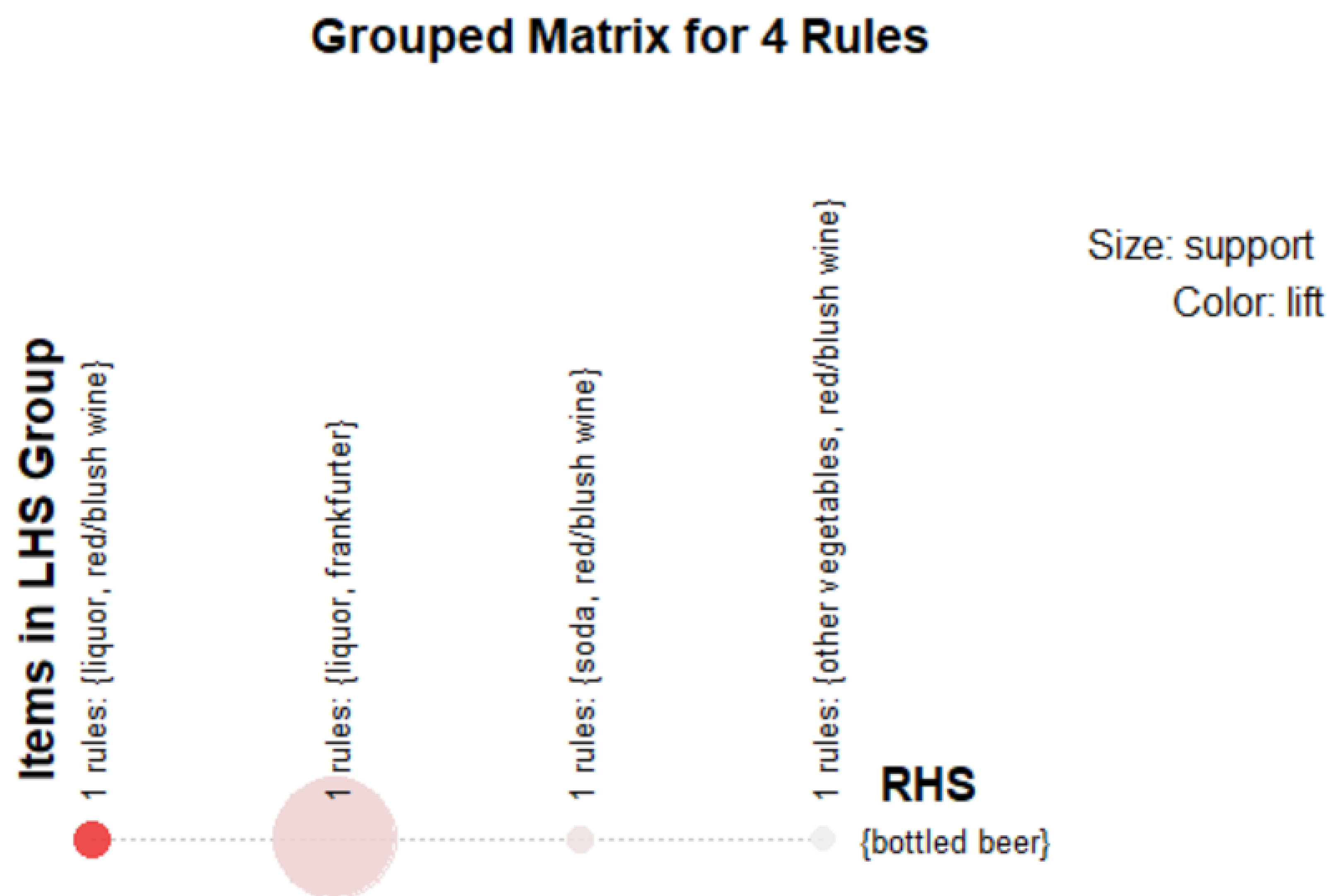
```
#plot 1
library(arules); library(arulesViz)
plot(rules, method="graph", measure='confidence', shading='lift',
     control=list(type="items"))
```




```
    lhs                    rhs             support     confidence lift
[1] {liquor,
     red/blush wine}    => {bottled beer} 0.001931876  0.9047619 11.235269
[2] {liquor}            => {bottled beer} 0.004677173  0.4220183  5.240594
[3] {soda,
     red/blush wine}    => {bottled beer} 0.001626843  0.3555556  4.415264
[4] {other vegetables,
     red/blush wine}    => {bottled beer} 0.001525165  0.3061224  3.801407
```

Liquor and red/blush wine indicate the best lift and the highest confidence level with both the size of the circle and its shading.

```
#plot 2
plot(rules, method="grouped", control=list(type="items"))
```



3. Tabular Dataset: Titanic

The Titanic dataset is a 4-dimensional table with summarized information on the fate of passengers on the Titanic according to social class, sex, age and survival.

```
> str(Titanic)
 table [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...
 - attr(*, "dimnames")=List of 4
  ..$ Class   : chr [1:4] "1st" "2nd" "3rd" "Crew"
  ..$ Sex     : chr [1:2] "Male" "Female"
  ..$ Age     : chr [1:2] "Child" "Adult"
  ..$ Survived: chr [1:2] "No" "Yes"
> class(Titanic)
[1] "table"
> dim(Titanic)
[1] 4 2 2 2
> head(as.data.frame(Titanic),10)
   Class    Sex   Age Survived Freq
1    1st   Male Child       No    0
2    2nd   Male Child       No    0
3    3rd   Male Child       No   35
4   Crew   Male Child       No    0
5    1st Female Child       No    0
6    2nd Female Child       No    0
7    3rd Female Child       No   17
8   Crew Female Child       No    0
9    1st   Male Adult       No  118
10   2nd   Male Adult       No  154
```

(1) Reconstructed titanic raw data

We must reconstruct the Titanic dataset as raw data to make it suitable for association rule mining. The reconstructed raw data can be downloaded at http://www.rdatamining.com/data/titanic.raw.rdata.

download.file(url, destfile, mode, ...)

This function can be used to download a file from the Internet.

```
> url <- "http://www.rdatamining.com/data/titanic.raw.rdata"
> download.file(url, destfile="titanic.raw.RData", mode="wb")
> load("titanic.raw.RData")
> head(titanic.raw,4)
  Class  Sex   Age Survived
1   3rd Male Child       No
2   3rd Male Child       No
3   3rd Male Child       No
4   3rd Male Child       No
> summary(titanic.raw)
  Class         Sex          Age        Survived
 1st :325   Female: 470   Adult:2092   No :1490
 2nd :285   Male  :1731   Child: 109   Yes: 711
 3rd :706
 Crew:885
```

(2) Association Rule Mining

We can set rhs=c("Survived=No", "Survived=Yes") in appearance to make sure that only "Survived=No" and "Survived=Yes" will appear in the rhs of rules.

```
library(arules)
titanic_rules <- apriori(titanic.raw,
  parameter=list(minlen=2,supp=0.005,conf=0.8),
  appearance=list(rhs=c("Survived=No","Survived=Yes"),
  default="lhs"), control=list(verbose=FALSE))
```

Setting rules with 5 decimal places

```
quality(titanic_rules) <- round(quality(titanic_rules), digits=5)
```

Sorting rules by lift

```
> titanic_rules.sort <- sort(titanic_rules, by="lift")
> inspect(titanic_rules.sort)
     lhs                                  rhs            support     confidence lift     count
[1]  {Class=2nd,Age=Child}             => {Survived=Yes} 0.010904134 1.0000000  3.095640  24
[2]  {Class=2nd,Sex=Female,Age=Child}  => {Survived=Yes} 0.005906406 1.0000000  3.095640  13
[3]  {Class=1st,Sex=Female}            => {Survived=Yes} 0.064061790 0.9724138  3.010243 141
[4]  {Class=1st,Sex=Female,Age=Adult}  => {Survived=Yes} 0.063607451 0.9722222  3.009650 140
[5]  {Class=2nd,Sex=Female}            => {Survived=Yes} 0.042253521 0.8773585  2.715986  93
[6]  {Class=Crew,Sex=Female}           => {Survived=Yes} 0.009086779 0.8695652  2.691861  20
[7]  {Class=Crew,Sex=Female,Age=Adult} => {Survived=Yes} 0.009086779 0.8695652  2.691861  20
[8]  {Class=2nd,Sex=Female,Age=Adult}  => {Survived=Yes} 0.036347115 0.8602151  2.662916  80
[9]  {Class=2nd,Sex=Male,Age=Adult}    => {Survived=No}  0.069968196 0.9166667  1.354083 154
[10] {Class=2nd,Sex=Male}              => {Survived=No}  0.069968196 0.8603352  1.270871 154
[11] {Class=3rd,Sex=Male,Age=Adult}    => {Survived=No}  0.175829169 0.8376623  1.237379 387
[12] {Class=3rd,Sex=Male}              => {Survived=No}  0.191731031 0.8274510  1.222295 422
```

(3) Interpreting Rules

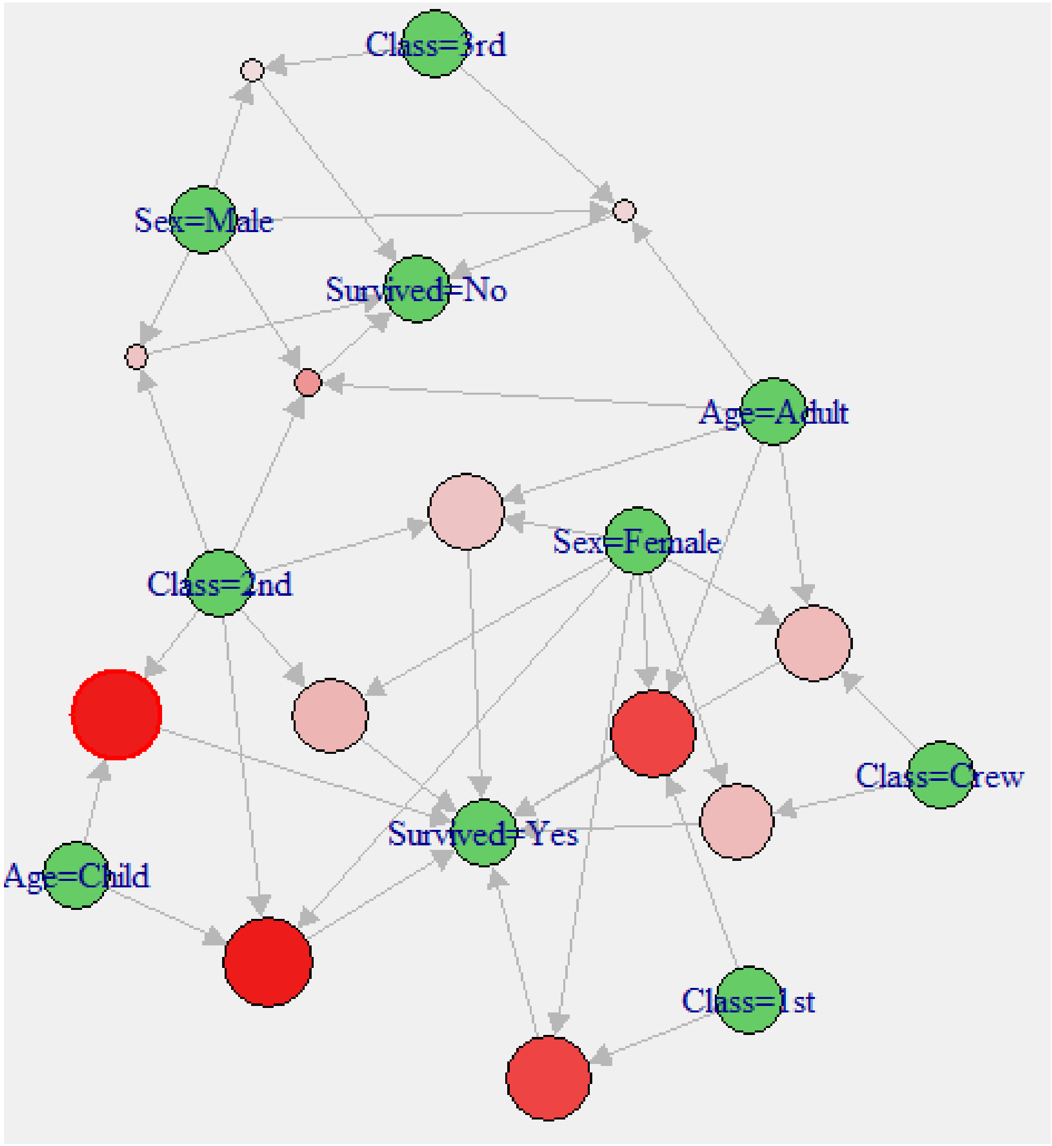
```
> #Removing Redundancy
> inspect(titanic_rules.sort[1:2])
    lhs                                  rhs            support confidence lift    count
[1] {Class=2nd,Age=Child}             => {Survived=Yes} 0.01090 1          3.09564 24
[2] {Class=2nd,Sex=Female,Age=Child} => {Survived=Yes} 0.00591 1          3.09564 13
```

[1] The rule states only that **all children** (24) **of class 2 survived**.

[2] The rule states only that **all female children** (13) **of class 2 survived**.

(4) Visualizing Association Rules


```
library(arulesViz)
plot(titanic_rules, method="graph", measure='lift', engine='interactive',
     shading='confidence')
```



size=lift
color=confidence

```
plot(titanic_rules, method='grouped',measure='lift', shading='confidence')
```