Individual Assignment 4: Association Rules by Shimony Agrawal

Download the necessary packages for Market Basket Analysis and Association Rules.

```
library(DBI)
```

```
## Warning: package 'DBI' was built under R version 4.0.2
```

```
library(odbc)
```

```
## Warning: package 'odbc' was built under R version 4.0.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyverse
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
## Warning: package 'dplyr' was built under R version 4.0.2
```

```
## Warning: package 'forcats' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------------------------------------ tidyverse_conflicts
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
library(ggplot2)
library(arules)
```

```
## Warning: package 'arules' was built under R version 4.0.2
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
```

```
## Warning: package 'arulesViz' was built under R version 4.0.2
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

Part 1: Load the data set.

```
data(Groceries)
str(Groceries)
```

```
## Formal class 'transactions' [package "arules"] with 3 slots
##   ..@ data       :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
##   .. .. ..@ i       : int [1:43367] 13 60 69 78 14 29 98 24 15 29 ...
##   .. .. ..@ p       : int [1:9836] 0 4 7 8 12 16 21 22 27 28 ...
##   .. .. ..@ Dim     : int [1:2] 169 9835
##   .. .. ..@ Dimnames:List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : NULL
```

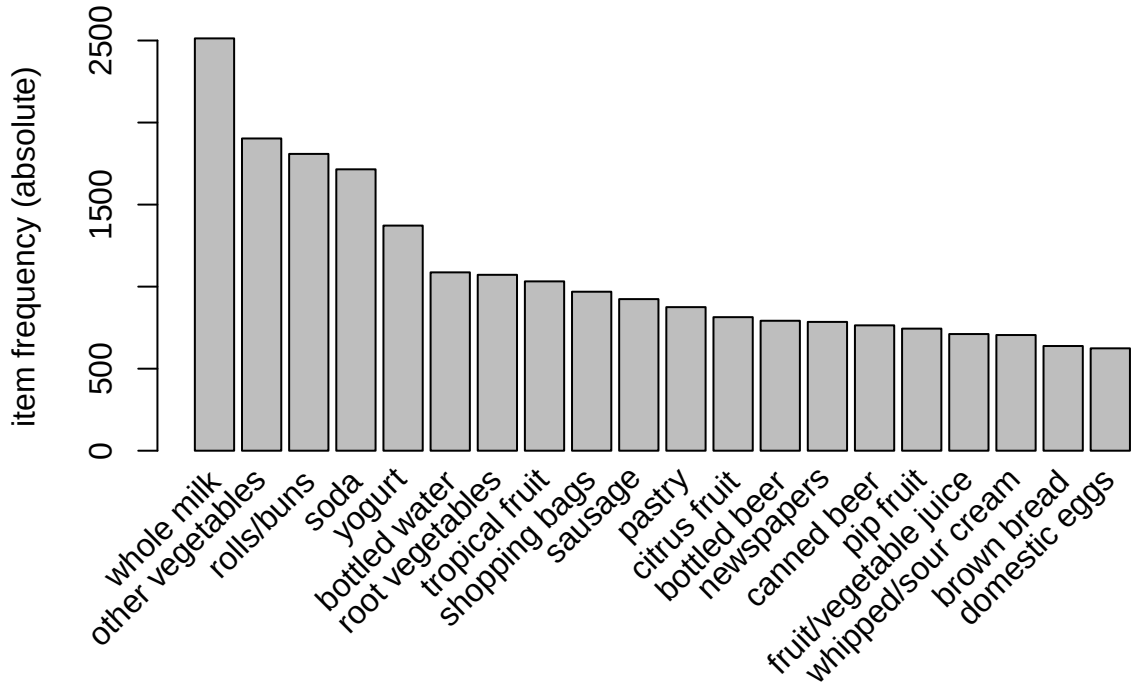
```
##   .. .. ..@ factors : list()
##   ..@ itemInfo   :'data.frame':  169 obs. of  3 variables:
##   .. ..$ labels: chr [1:169] "frankfurter" "sausage" "liver loaf" "ham" ...
##   .. ..$ level2: Factor w/ 55 levels "baby food","bags",..: 44 44 44 44 44 44 44 42 42 41 ...
##   .. ..$ level1: Factor w/ 10 levels "canned food",..: 6 6 6 6 6 6 6 6 6 6 ...
##   ..@ itemsetInfo:'data.frame':  0 obs. of  0 variables
```

```
dim(Groceries)
```

```
## [1] 9835  169
```

Part 2: Data Visualisation - Generate an item frequency barplot for the Top 20 grocery item. - Generate an item frequency barplot for the grocery items with support rate greater than 5%. - Generate an item frequency barplot for the grocery items with support rate greater than 3%.

```
itemFrequencyPlot(Groceries,topN=20,type="absolute")
```



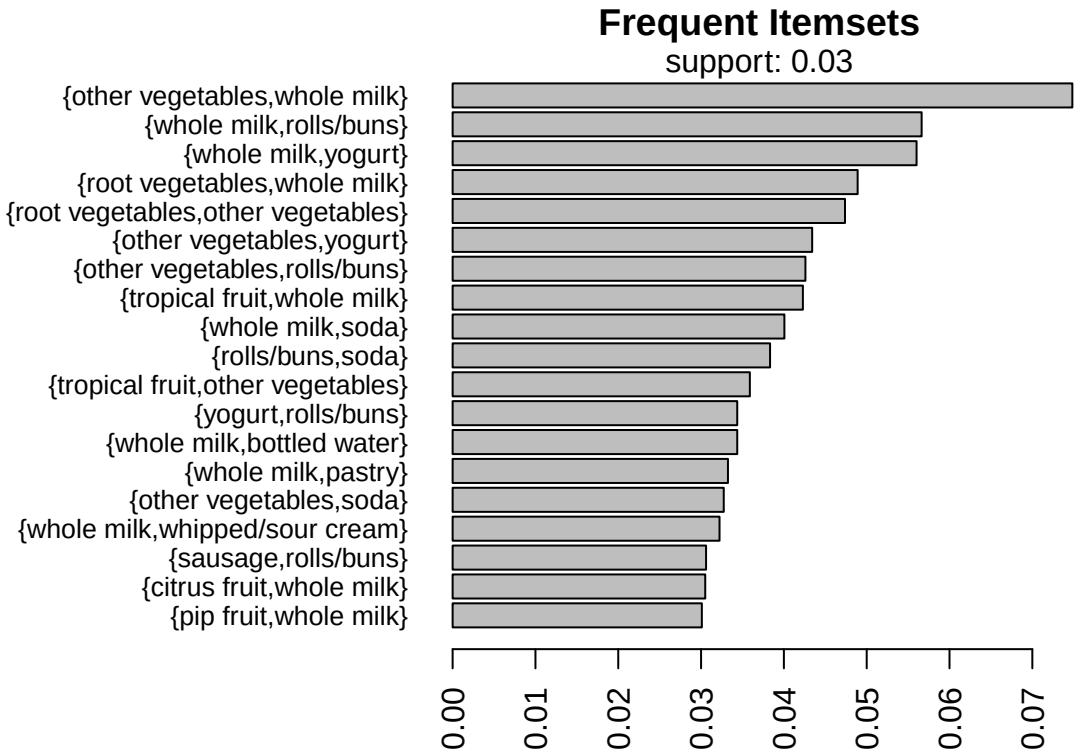

```
support = 0.05
itemsets <- apriori(Groceries,
                    parameter = list(target = "frequent itemsets", supp = support, conf = 0.60, minlen =
                    control = list (verbose = FALSE))

par(mar = c(5,14,2,2)+.1)
order_sets <- DATAFRAME(sort(itemsets, by = "support", decreasing = F))
```

```r
barplot(order_sets$support, names.arg = order_sets$items,
        horiz = T, las = 2, cex.names = .8, main = "Frequent Itemsets")
mtext(paste("support:", support), padj = .8)
```


```r
support = 0.03
itemsets <- apriori(Groceries,
                    parameter = list(target = "frequent itemsets", supp = support, conf = 0.60, minlen =
                    control = list (verbose = FALSE))

par(mar = c(5,14,2,2)+.1)
order_sets <- DATAFRAME(sort(itemsets, by = "support", decreasing = F))
barplot(order_sets$support, names.arg = order_sets$items,
        horiz = T, las = 2, cex.names = .8, main = "Frequent Itemsets")
mtext(paste("support:", support), padj = .8)
```



Part 2: Use the apriori algorithm to identify the top 20 rules.

```
# Get the rules
rules <- apriori(Groceries, 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 TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.01s].
## writing ... [410 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# Show the top 20  rules
arules::inspect(rules[1:20])
```

```
##      lhs                            rhs                    support confidence    coverage      lift cou
## [1]  {liquor,
##       red/blush wine}         => {bottled beer}     0.001931876  0.9047619 0.002135231 11.235269    1
## [2]  {curd,
##       cereals}                => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [3]  {yogurt,
##       cereals}                => {whole milk}       0.001728521  0.8095238 0.002135231  3.168192    1
## [4]  {butter,
##       jam}                    => {whole milk}       0.001016777  0.8333333 0.001220132  3.261374    1
## [5]  {soups,
##       bottled beer}           => {whole milk}       0.001118454  0.9166667 0.001220132  3.587512    1
## [6]  {napkins,
##       house keeping products} => {whole milk}       0.001321810  0.8125000 0.001626843  3.179840    1
## [7]  {whipped/sour cream,
##       house keeping products} => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599    1
## [8]  {pastry,
##       sweet spreads}          => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [9]  {turkey,
##       curd}                   => {other vegetables} 0.001220132  0.8000000 0.001525165  4.134524    1
## [10] {rice,
##       sugar}                  => {whole milk}       0.001220132  1.0000000 0.001220132  3.913649    1
## [11] {butter,
##       rice}                   => {whole milk}       0.001525165  0.8333333 0.001830198  3.261374    1
## [12] {domestic eggs,
##       rice}                   => {whole milk}       0.001118454  0.8461538 0.001321810  3.311549    1
## [13] {rice,
##       bottled water}          => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599    1
## [14] {yogurt,
##       rice}                   => {other vegetables} 0.001931876  0.8260870 0.002338587  4.269346    1
## [15] {oil,
##       mustard}                => {whole milk}       0.001220132  0.8571429 0.001423488  3.354556    1
## [16] {canned fish,
##       hygiene articles}       => {whole milk}       0.001118454  1.0000000 0.001118454  3.913649    1
## [17] {herbs,
##       fruit/vegetable juice}  => {other vegetables} 0.001220132  0.8000000 0.001525165  4.134524    1
## [18] {herbs,
##       shopping bags}          => {other vegetables} 0.001931876  0.8260870 0.002338587  4.269346    1
## [19] {tropical fruit,
##       herbs}                  => {whole milk}       0.002338587  0.8214286 0.002846975  3.214783    1
## [20] {herbs,
##       rolls/buns}             => {whole milk}       0.002440264  0.8000000 0.003050330  3.130919    1
```

```r
summary(rules)
```

```
## set of 410 rules
##
## rule length distribution (lhs + rhs):sizes
##   3   4   5   6
##  29 229 140  12
##
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   4.000   4.000   4.329   5.000   6.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001017   Min.   :0.8000   Min.   :0.001017   Min.   : 3.131
##  1st Qu.:0.001017   1st Qu.:0.8333   1st Qu.:0.001220   1st Qu.: 3.312
##  Median :0.001220   Median :0.8462   Median :0.001322   Median : 3.588
##  Mean   :0.001247   Mean   :0.8663   Mean   :0.001449   Mean   : 3.951
##  3rd Qu.:0.001322   3rd Qu.:0.9091   3rd Qu.:0.001627   3rd Qu.: 4.341
##  Max.   :0.003152   Max.   :1.0000   Max.   :0.003559   Max.   :11.235
##      count
##  Min.   :10.00
##  1st Qu.:10.00
##  Median :12.00
##  Mean   :12.27
##  3rd Qu.:13.00
##  Max.   :31.00
##
## mining info:
##       data ntransactions support confidence
##  Groceries          9835   0.001        0.8
```

Part 4: Sort out the rules by confidence.

```
rules<-sort(rules, by="confidence", decreasing=TRUE)
rules <- apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8, maxlen=3))
```

```
## 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
##       3  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3

## Warning in apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8, :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!

##  done [0.00s].
## writing ... [29 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```r
summary(rules)
```

```
## set of 29 rules
##
## rule length distribution (lhs + rhs):sizes
##  3
## 29
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       3       3       3       3       3       3
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001017   Min.   :0.8000   Min.   :0.001118   Min.   : 3.131
##  1st Qu.:0.001118   1st Qu.:0.8125   1st Qu.:0.001220   1st Qu.: 3.261
##  Median :0.001220   Median :0.8462   Median :0.001525   Median : 3.613
##  Mean   :0.001473   Mean   :0.8613   Mean   :0.001732   Mean   : 4.000
##  3rd Qu.:0.001729   3rd Qu.:0.9091   3rd Qu.:0.002135   3rd Qu.: 4.199
##  Max.   :0.002542   Max.   :1.0000   Max.   :0.003152   Max.   :11.235
##      count
##  Min.   :10.00
##  1st Qu.:11.00
##  Median :12.00
##  Mean   :14.48
##  3rd Qu.:17.00
##  Max.   :25.00
##
## mining info:
##       data ntransactions support confidence
##  Groceries          9835   0.001        0.8
```

```r
arules::inspect(rules[1:20])
```

```
##      lhs                          rhs                    support confidence    coverage      lift cou
## [1]  {liquor,
##       red/blush wine}         => {bottled beer}     0.001931876  0.9047619 0.002135231 11.235269    1
## [2]  {curd,
##       cereals}                => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [3]  {yogurt,
##       cereals}                => {whole milk}       0.001728521  0.8095238 0.002135231  3.168192    1
## [4]  {butter,
##       jam}                    => {whole milk}       0.001016777  0.8333333 0.001220132  3.261374    1
## [5]  {soups,
##       bottled beer}           => {whole milk}       0.001118454  0.9166667 0.001220132  3.587512    1
## [6]  {napkins,
##       house keeping products} => {whole milk}       0.001321810  0.8125000 0.001626843  3.179840    1
## [7]  {whipped/sour cream,
##       house keeping products} => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599    1
## [8]  {pastry,
##       sweet spreads}          => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [9]  {turkey,
##       curd}                   => {other vegetables} 0.001220132  0.8000000 0.001525165  4.134524    1
```

```
## [10] {rice,
##       sugar}                 => {whole milk}       0.001220132  1.0000000 0.001220132  3.913649
## [11] {butter,
##       rice}                  => {whole milk}       0.001525165  0.8333333 0.001830198  3.261374
## [12] {domestic eggs,
##       rice}                  => {whole milk}       0.001118454  0.8461538 0.001321810  3.311549
## [13] {rice,
##       bottled water}         => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599
## [14] {yogurt,
##       rice}                  => {other vegetables} 0.001931876  0.8260870 0.002338587  4.269346
## [15] {oil,
##       mustard}               => {whole milk}       0.001220132  0.8571429 0.001423488  3.354556
## [16] {canned fish,
##       hygiene articles}      => {whole milk}       0.001118454  1.0000000 0.001118454  3.913649
## [17] {herbs,
##       fruit/vegetable juice} => {other vegetables} 0.001220132  0.8000000 0.001525165  4.134524
## [18] {herbs,
##       shopping bags}         => {other vegetables} 0.001931876  0.8260870 0.002338587  4.269346
## [19] {tropical fruit,
##       herbs}                 => {whole milk}       0.002338587  0.8214286 0.002846975  3.214783
## [20] {herbs,
##       rolls/buns}            => {whole milk}       0.002440264  0.8000000 0.003050330  3.130919
```

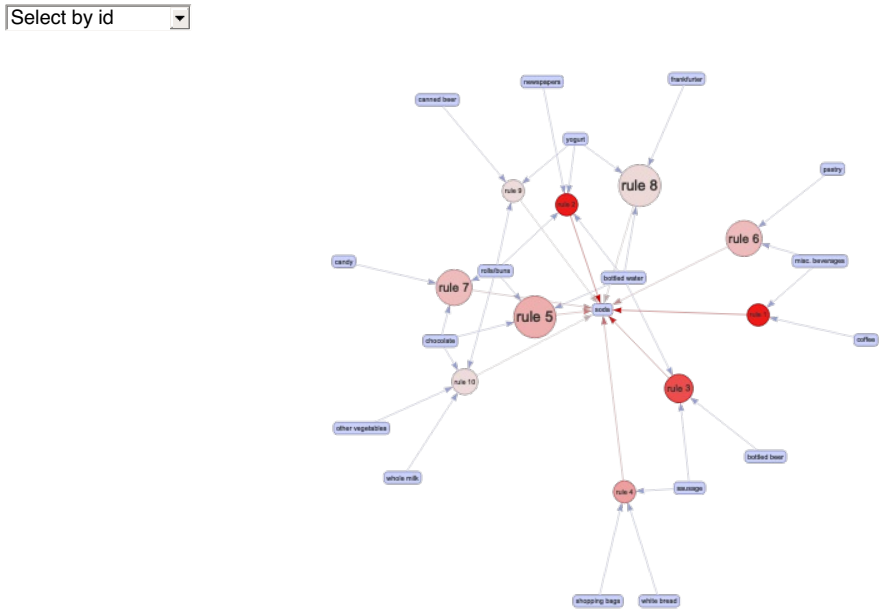
Part 5: Targeting items from the Top 20 items based on frequency.

```
# RHS rules
rules<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.08),
               appearance = list(default="lhs",rhs="soda"),
               control = list(verbose=F))
rules<-sort(rules, decreasing=TRUE,by="confidence")
arules::inspect(rules[1:10])
```

```
##      lhs                    rhs        support confidence    coverage     lift count
## [1]  {coffee,
##       misc. beverages}   => {soda} 0.001016777  0.7692308 0.001321810 4.411303    10
## [2]  {yogurt,
##       rolls/buns,
##       bottled water,
##       newspapers}        => {soda} 0.001016777  0.7692308 0.001321810 4.411303    10
## [3]  {sausage,
##       bottled water,
##       bottled beer}      => {soda} 0.001118454  0.7333333 0.001525165 4.205442    11
## [4]  {sausage,
##       white bread,
##       shopping bags}     => {soda} 0.001016777  0.6666667 0.001525165 3.823129    10
## [5]  {rolls/buns,
##       bottled water,
##       chocolate}         => {soda} 0.001321810  0.6500000 0.002033554 3.727551    13
## [6]  {pastry,
##       misc. beverages}   => {soda} 0.001220132  0.6315789 0.001931876 3.621912    12
## [7]  {rolls/buns,
##       chocolate,
##       candy}             => {soda} 0.001220132  0.6315789 0.001931876 3.621912    12
```

```
## [8]  {frankfurter,
##       yogurt,
##       bottled water}    => {soda} 0.001321810  0.5909091 0.002236909 3.388683    13
## [9]  {yogurt,
##       rolls/buns,
##       canned beer}      => {soda} 0.001016777  0.5882353 0.001728521 3.373349    10
## [10] {other vegetables,
##       whole milk,
##       rolls/buns,
##       chocolate}        => {soda} 0.001016777  0.5882353 0.001728521 3.373349    10
```

```
toprules_soda <- (rules[1:10])
plot(toprules_soda, method = 'graph', engine = 'htmlwidget')
```

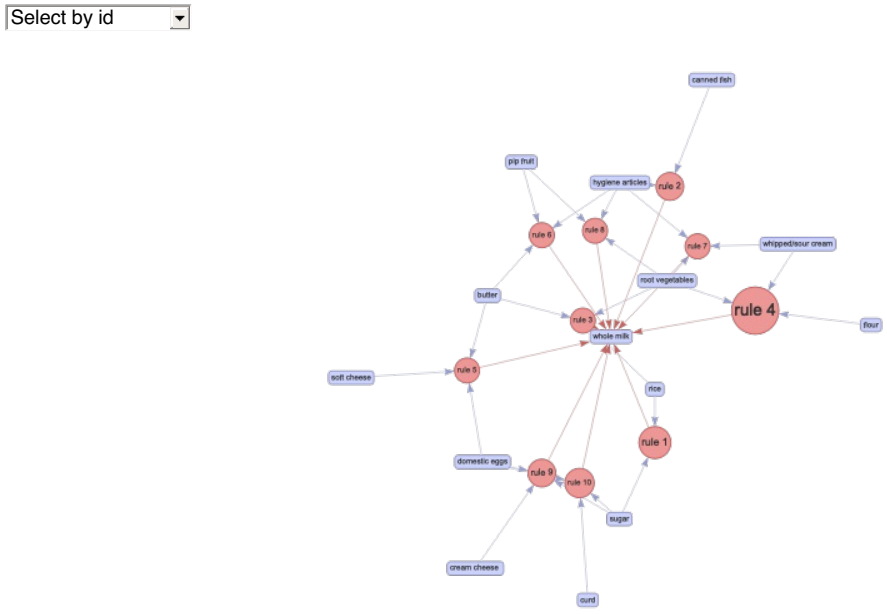



```
rules<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.08),
               appearance = list(default="lhs",rhs="whole milk"),
               control = list(verbose=F))
rules<-sort(rules, decreasing=TRUE,by="confidence")
arules::inspect(rules[1:10])
```

```
##      lhs                     rhs              support confidence    coverage     lift count
## [1]  {rice,
##       sugar}              => {whole milk} 0.001220132          1 0.001220132 3.913649    12
## [2]  {canned fish,
##       hygiene articles}   => {whole milk} 0.001118454          1 0.001118454 3.913649    11
## [3]  {root vegetables,
##       butter,
##       rice}               => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [4]  {root vegetables,
```

```
##       whipped/sour cream,
##       flour}              => {whole milk} 0.001728521          1 0.001728521 3.913649    17
## [5]  {butter,
##       soft cheese,
##       domestic eggs}      => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [6]  {pip fruit,
##       butter,
##       hygiene articles}   => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [7]  {root vegetables,
##       whipped/sour cream,
##       hygiene articles}   => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [8]  {pip fruit,
##       root vegetables,
##       hygiene articles}   => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [9]  {cream cheese ,
##       domestic eggs,
##       sugar}              => {whole milk} 0.001118454          1 0.001118454 3.913649    11
## [10] {curd,
##       domestic eggs,
##       sugar}              => {whole milk} 0.001016777          1 0.001016777 3.913649    10
```

```
toprules_milk <- (rules[1:10])
plot(toprules_milk, method = 'graph', engine = 'htmlwidget')
```



For the rhs {Soda}: It can be seen that when rolls/buns, bottled water and chocolate are purchased there is a 65% chance of soda also being bought with a support of 0.13% indicating the highest frequency in the rules. Also, with the purchase of coffee and misc. beverages there is a 77% chance that soda will also be purchased.

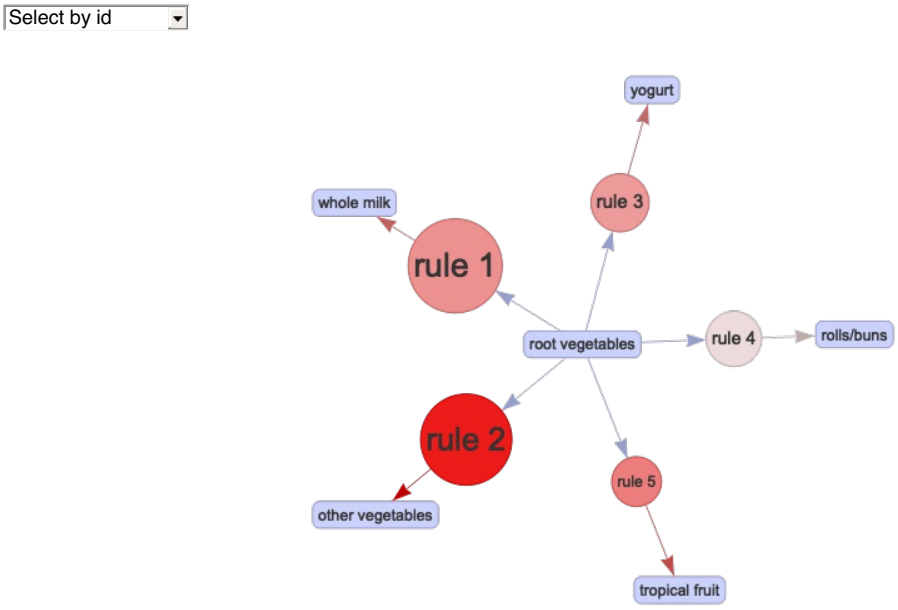
For the rhs {Whole Milk}: Interestingly, for all the items bought on lhs, there is a 100% chance that the item on rhs will be purchased. For instnace, whenever root vegetables, whipped/sour cream and flour are purhcased there is a 100% likelihood that whole will also be bought.

All the rules have a lift greater than 1, showing a positive correlation between the products in the itemset, thereby indicating that the two products are more likely to be bought together. Those rules that have the higher confidence, support and lift are the strongest.

```
# LHS Rules
rules<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.15,minlen=2),
               appearance = list(default="rhs",lhs="root vegetables"),
               control = list(verbose=F))
rules<-sort(rules, decreasing=TRUE,by="confidence")
arules::inspect(rules[1:5])
```

```
##     lhs                   rhs                support    confidence coverage
## [1] {root vegetables} => {whole milk}       0.04890696 0.4486940  0.1089985
## [2] {root vegetables} => {other vegetables} 0.04738180 0.4347015  0.1089985
## [3] {root vegetables} => {yogurt}           0.02582613 0.2369403  0.1089985
## [4] {root vegetables} => {rolls/buns}       0.02430097 0.2229478  0.1089985
## [5] {root vegetables} => {tropical fruit}   0.02104728 0.1930970  0.1089985
##     lift     count
## [1] 1.756031 481
## [2] 2.246605 466
## [3] 1.698475 254
## [4] 1.212101 239
## [5] 1.840222 207
```

```
toprules_vegetables <- (rules[1:5])
plot(toprules_vegetables, method = 'graph', engine = 'htmlwidget')
```

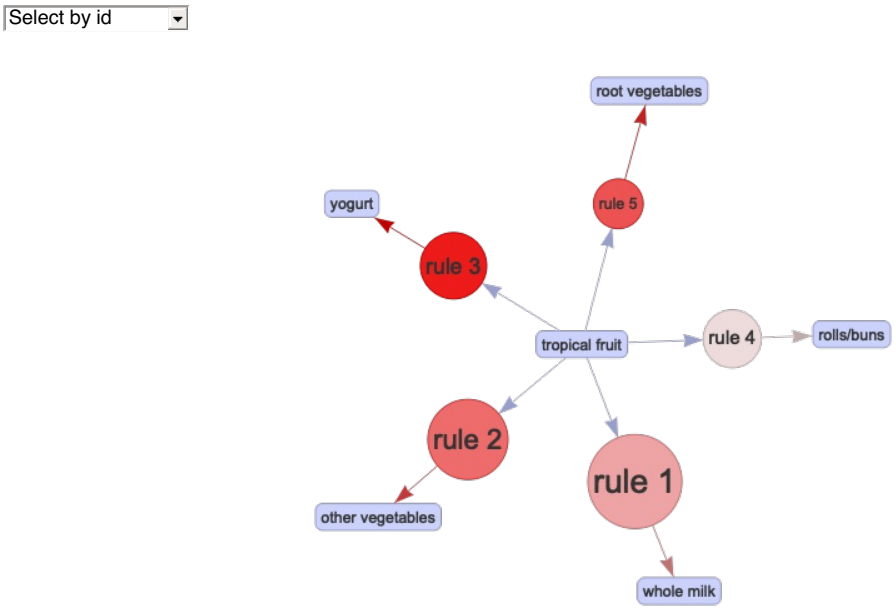


```
rules<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.15,minlen=2),
               appearance = list(default="rhs",lhs="tropical fruit"),
               control = list(verbose=F))

rules<-sort(rules, decreasing=TRUE,by="confidence")
arules::inspect(rules[1:5])
```

```
##     lhs                 rhs                support    confidence coverage
## [1] {tropical fruit} => {whole milk}       0.04229792 0.4031008  0.1049314
## [2] {tropical fruit} => {other vegetables} 0.03589222 0.3420543  0.1049314
## [3] {tropical fruit} => {yogurt}           0.02928317 0.2790698  0.1049314
## [4] {tropical fruit} => {rolls/buns}       0.02460600 0.2344961  0.1049314
## [5] {tropical fruit} => {root vegetables}  0.02104728 0.2005814  0.1049314
##     lift     count
## [1] 1.577595 416
## [2] 1.767790 353
## [3] 2.000475 288
## [4] 1.274886 242
## [5] 1.840222 207
```

```
toprules_fruit <- (rules[1:5])
plot(toprules_fruit, method = 'graph', engine = 'htmlwidget')
```

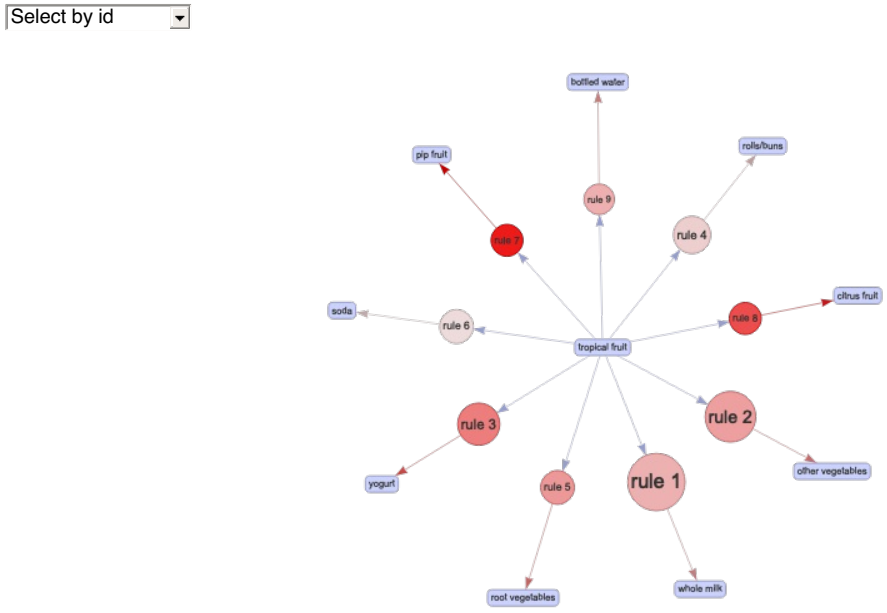


For lhr {root vegetables}: The highest association for root vegetables is whole milk with a 45% chance of whole milk being purchased with root vegetables. Moreover, this association has been purchased 481 times by consumers.

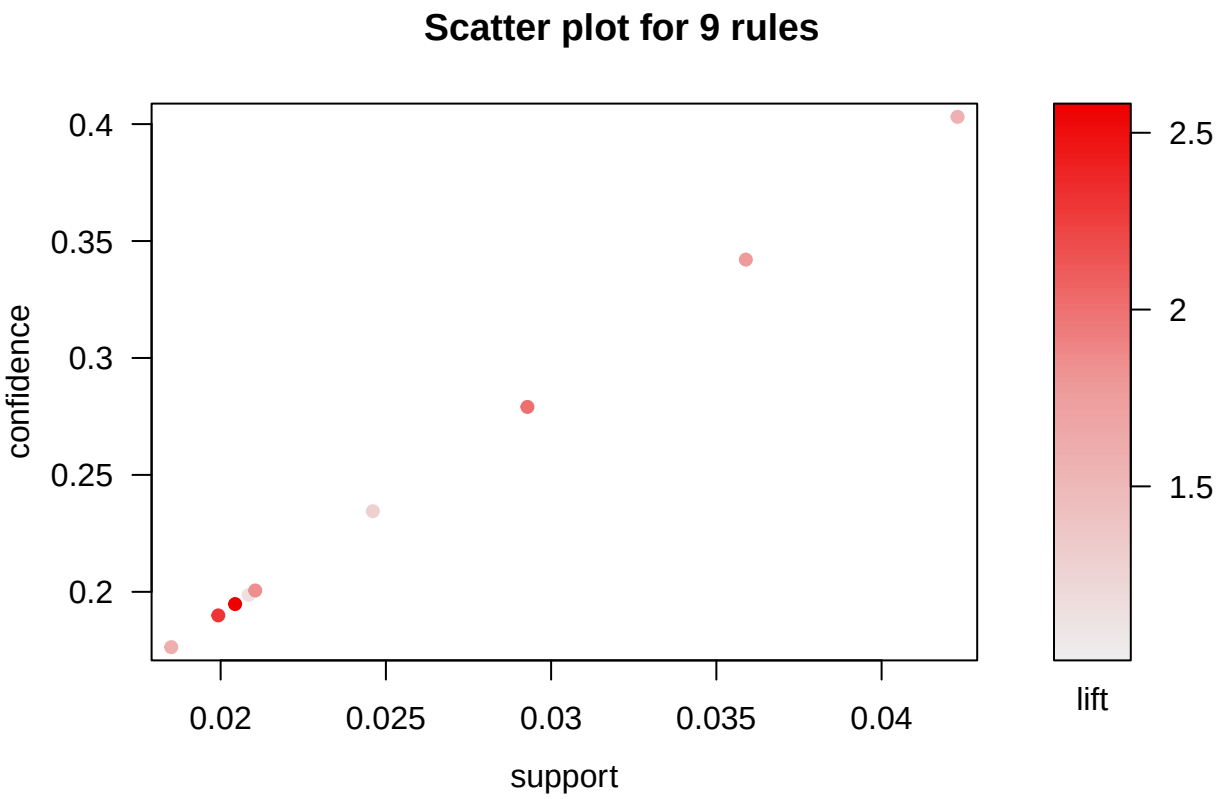
For rhs {tropical fruit}: Here too, whole milk has the strongest association with a 40% likelihood of tropical fruit and whole being purchased together.

All the rules have a lift greater than 1, showing a positive correlation between the products in the itemset, thereby indicating that the two products are more likely to be bought together. Those rules that have the higher confidence, support and lift are the strongest.

```
plot(rules, method = 'graph', engine = 'htmlwidget')
```




```
plot(rules, jitter = 0)
```


Scatter plot for 9 rules
confidence
0.4
0.35
0.3
0.25
0.2
0.02
0.025
0.03
0.035
0.04
support
2.5
2
1.5
lift