Association rule mining

###Question Revisit the notes on association rule mining and the R example on music playlists: playlists.R and playlists.csv. Then use the data on grocery purchases in groceries.txt and find some interesting association rules for these shopping baskets. The data file is a list of shopping baskets: one person's basket for each row, with multiple items per row separated by commas. Pick your own thresholds for lift and confidence; just be clear what these thresholds are and say why you picked them. Do your discovered item sets make sense? Present your discoveries in an interesting and visually appealing way.

Notes:

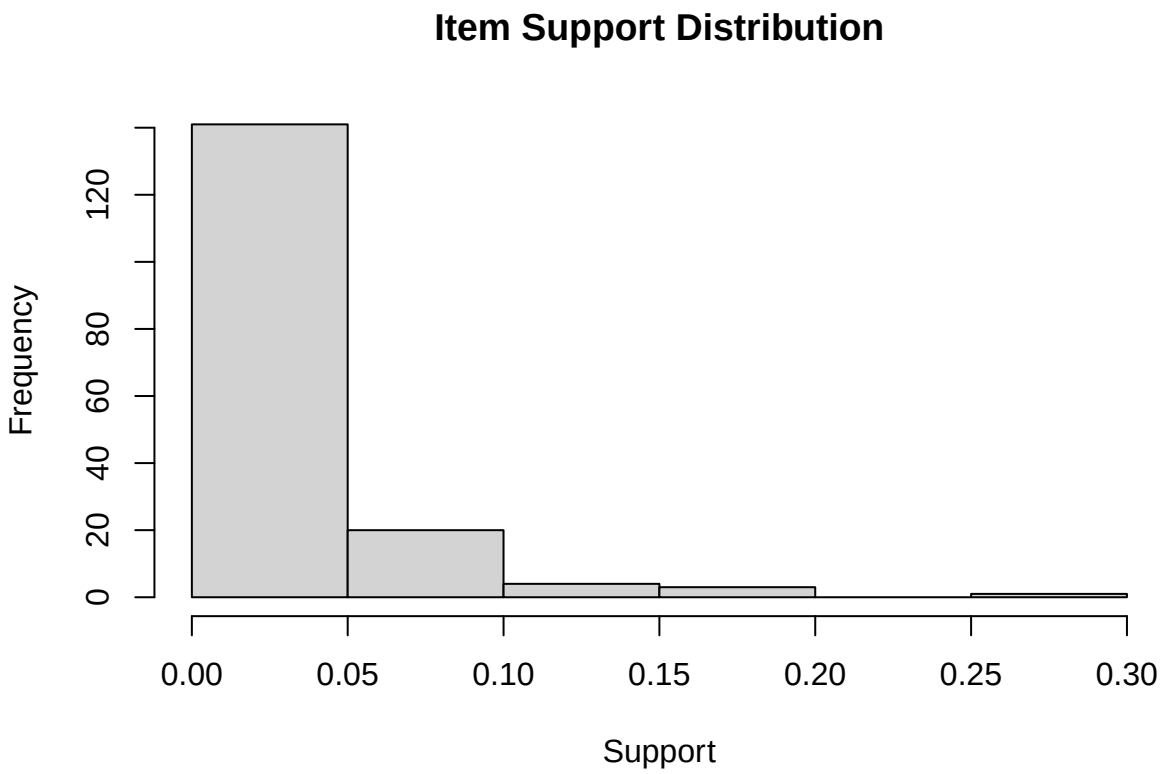
This is an exercise in visual and numerical story-telling. Do be clear in your description of what you've done, but keep the focus on the data, the figures, and the insights your analysis has drawn from the data, rather than technical details. The data file is a list of baskets: one row per basket, with multiple items per row separated by commas. You'll have to cobble together your own code for processing this into the format expected by the "arules" package. This is not intrinsically all that hard, but it is the kind of data-wrangling wrinkle you'll encounter frequently on real problems, where your software package expects data in one format and the data comes in a different format. Figuring out how to bridge that gap is part of the assignment, and so we won't be giving tips on this front.

###Answer First we read the text file given and convert it into "transaction" class from

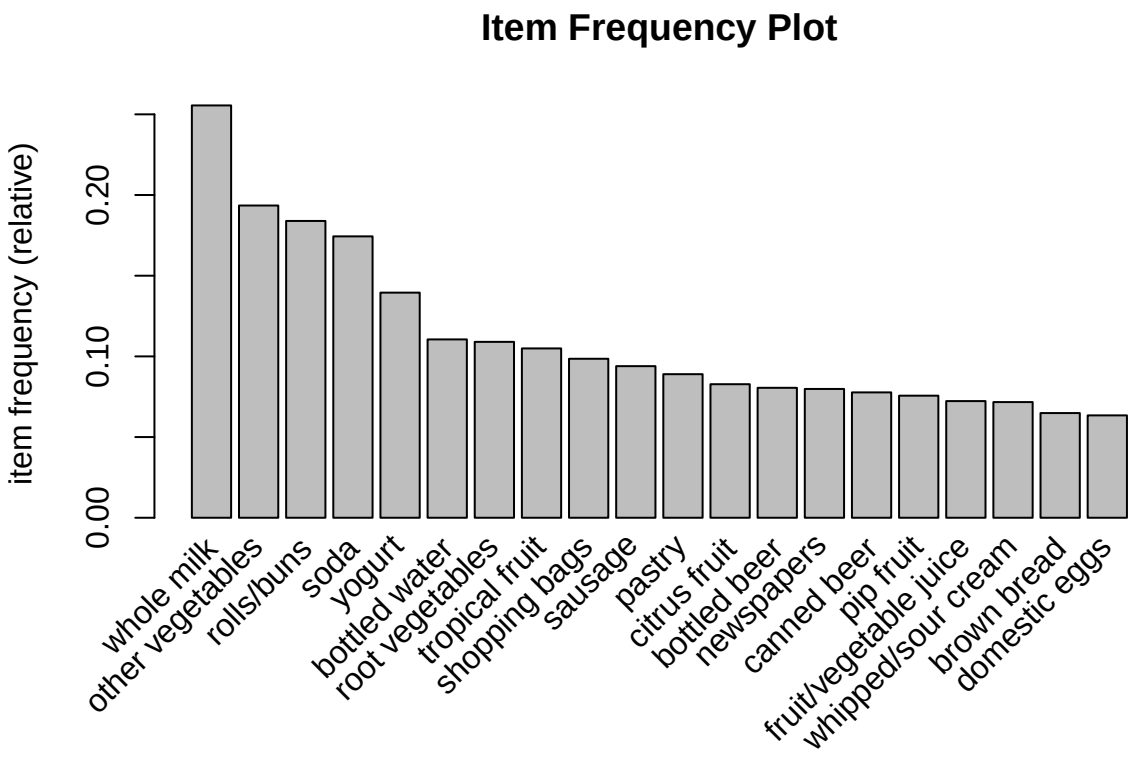
```
transactions <- readLines("groceries.txt")
transactions <- strsplit(transactions, split = ",")
transactions <- as(transactions, "transactions")
```

Now we'll plot Item Support Distribution Histogram to identify freuquency of Single Item Support and Item Frequency Plot to determine most common items.

```
item_support <- itemFrequency(transactions)
hist(item_support, main = "Item Support Distribution", xlab = "Support")
```




```
itemFrequencyPlot(transactions, topN = 20, type = "relative", main = "Item Frequency Plot")
```



As we can see, Whole Milk is the most common item bought in more than 25% of the transactions.

Let's take the support as 0.025 which is near the mean support. Now we'll look for association rules by setting initial parameters as 2 for min-len and 0.3 for confidence.

```
rules <- apriori(transactions, parameter = list(supp = 0.025, conf = 0.3, target = "rules", minlen = 2))
```

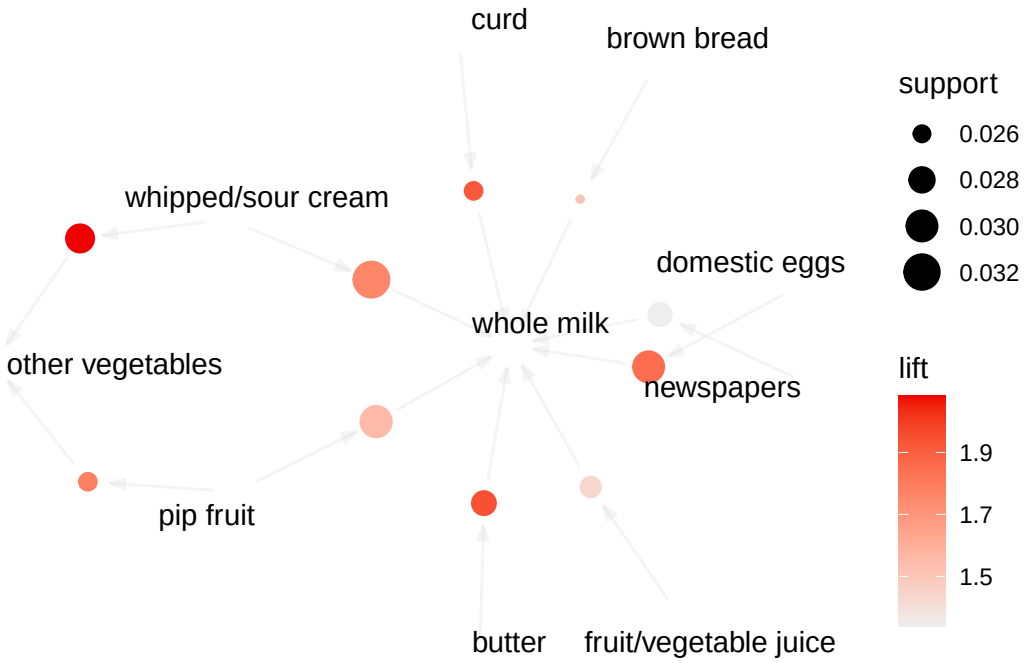
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.3    0.1    1 none FALSE            TRUE       5   0.025      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 245
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [54 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
```

```
## writing ... [24 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
plot(rules[1:10], method = "graph", control = list(type = "items"))
```

```
## Warning: Unknown control parameters: type
```

```
## Available control parameters (with default values):
## layout    =  stress
## circular  =  FALSE
## ggraphdots    =  NULL
## edges     =  <environment>
## nodes     =  <environment>
## nodetext  =  <environment>
## colors    =  c("#EE0000FF", "#EEEEEEFF")
## engine    =  ggplot2
## max   =  100
## verbose   =  FALSE
```



Now let's try to reduce the Support to 1st Qt that is 0.013 and increasing minlen to 3.

```
rules <- apriori(transactions, parameter = list(supp = 0.013, conf = 0.4, target = "rules", minlen = 3))
```

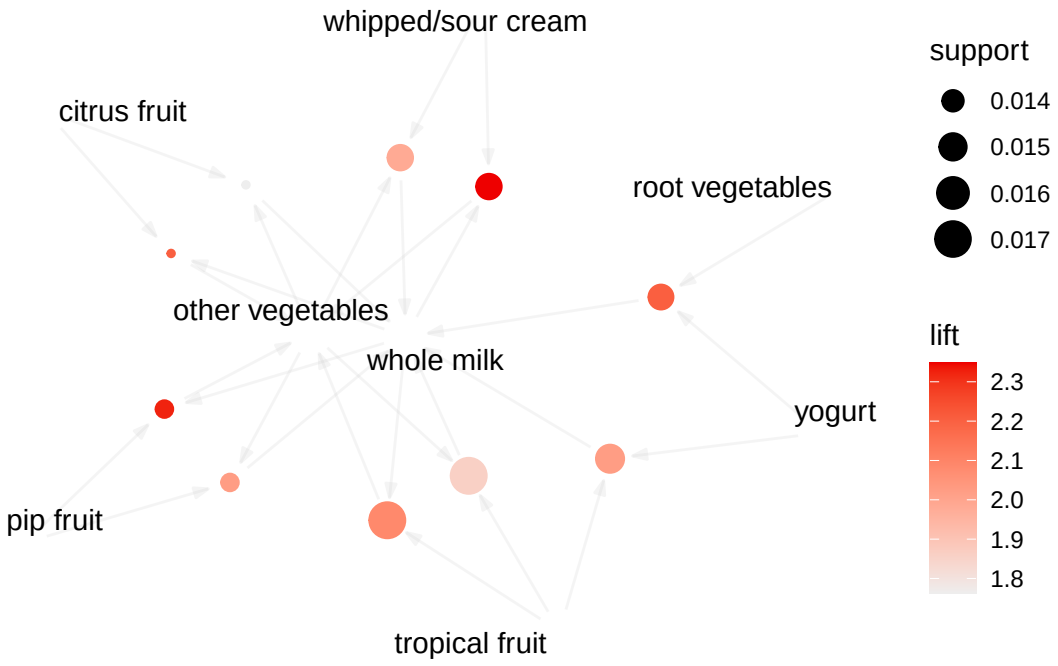
```
## Apriori
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5   0.013      3
##  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: 127
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [76 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [16 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
plot(rules[1:10], method = "graph", control = list(type = "items"))
```

```
## Warning: Unknown control parameters: type

## Available control parameters (with default values):
## layout    =  stress
## circular  =  FALSE
## ggraphdots    =  NULL
## edges     =  <environment>
## nodes     =  <environment>
## nodetext  =  <environment>
## colors    =  c("#EE0000FF", "#EEEEEEFF")
## engine    =  ggplot2
## max   =  100
## verbose   =  FALSE
```



Finally, let's try with a lower Support of 0.01 and min-len 2 with confidence 0.5.

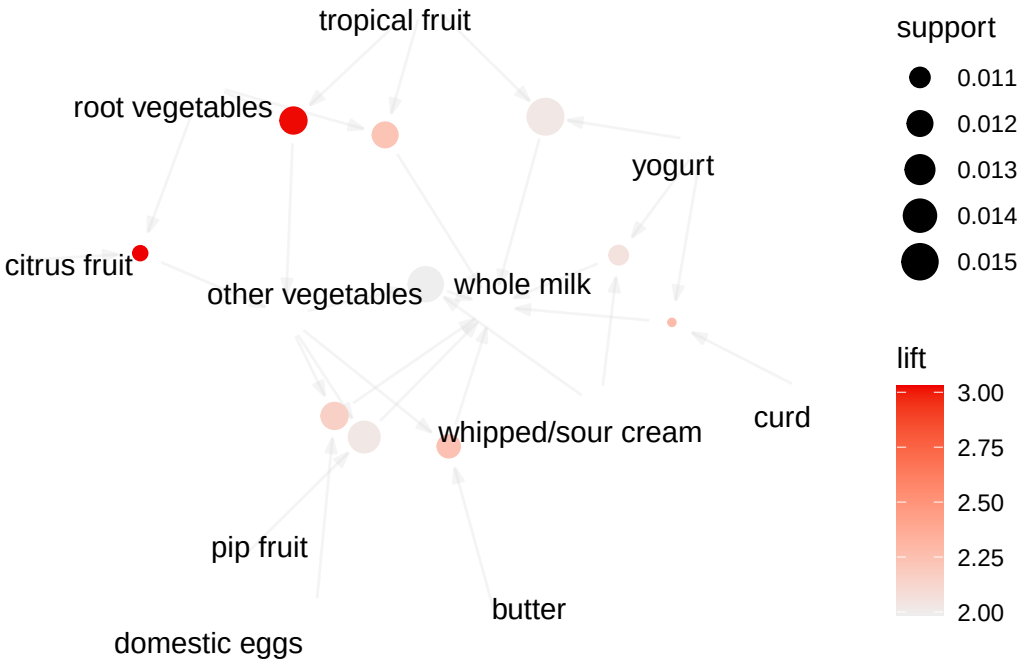
```
rules <- apriori(transactions, parameter = list(supp = 0.01, conf = 0.5, target = "rules", minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      2
##  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: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [88 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
plot(rules[1:10], method = "graph", control = list(type = "items"))
```

```
## Warning: Unknown control parameters: type
```

```
## Available control parameters (with default values):
## layout    =  stress
## circular  =  FALSE
## ggraphdots    =  NULL
## edges     =  <environment>
## nodes     =  <environment>
## nodetext  =  <environment>
## colors    =  c("#EE0000FF", "#EEEEEEFF")
## engine    =  ggplot2
## max   =  100
## verbose   =  FALSE
```



We can observe a few things from this:

1. Whenever pip fruit is bought, whole milk is likely bought too.

2. If people purchase root vegetables or citrus fruits, they are likely to purchase other vegetables too.

3. Whenever people purchase tropical fruits, they likely purchased other vegetables. Support 0.01 and confidence 0.5 is chosen as the top 10 items provide a good lift between 2 to 3.