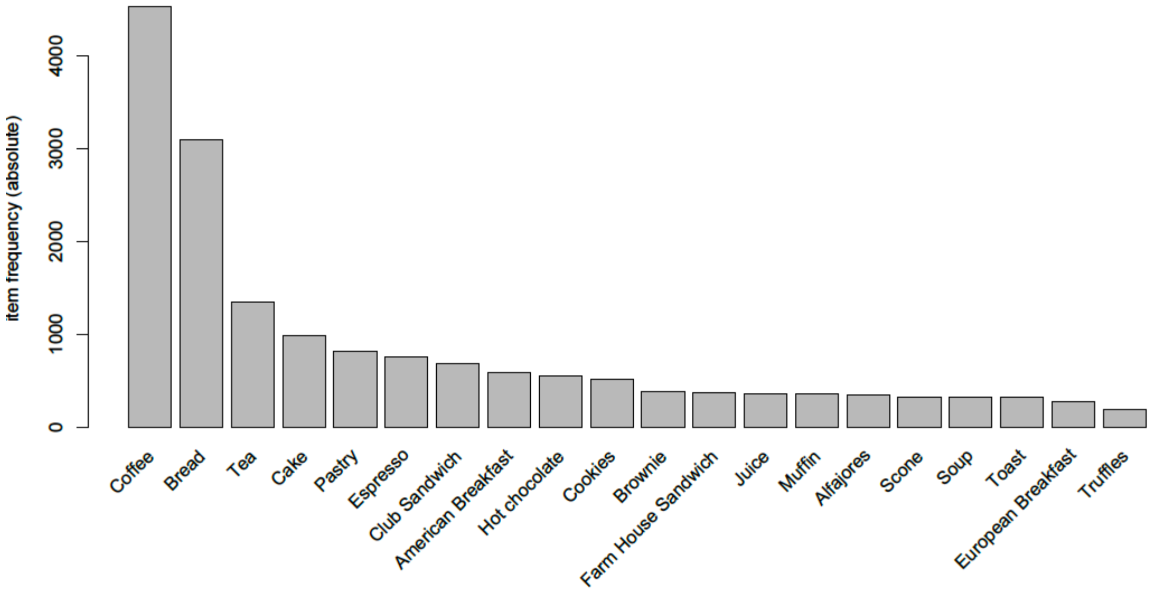
APPLIED EXERCISE 2 - MARKET BASKET ANALYSIS: CAFELEPETIT

EXECUTIVE SUMMARY

In this exercise, a Market Basket Analysis is conducted on a dataset (cafelepetit.csv), containing transaction information of items bought in a café in a busy city. Associative Rule Learning is used to find out rules about 'people who bought X also bought Y'. Here, Apriori model is employed to analyse the items ordered by customers of Cafelepetit. By tweaking the minimum support and confidence levels, successful insights can be gleaned from the buying patterns of the customer.

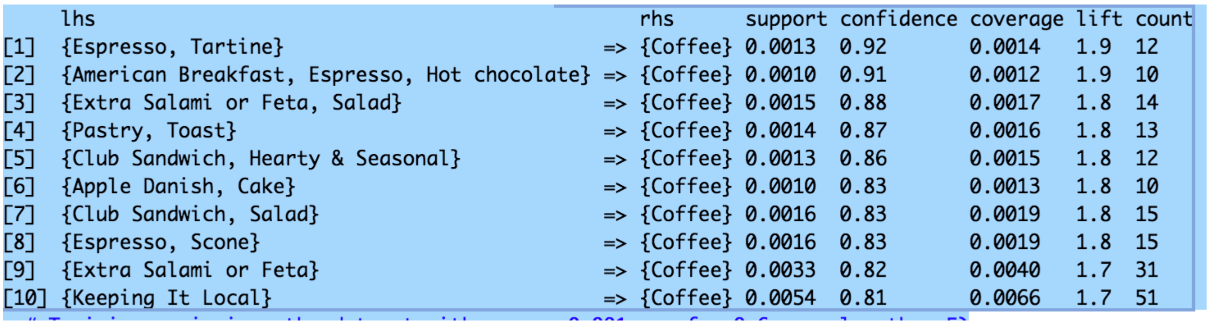
1.Examine the data and identify the variables that you should consider building a market basket analysis to understand how the patrons like to order at the cafe.

The csv file (cafelepetit.csv) contains 4 variables: "dateoftransaction", "timeofday", "orderno", and "item". Out of these, columns "orderno", and "item" are used to construct the sparse matrix. The matrix contains 9530 rows/transactions and 75 columns (unique items), with a minimum of 1 item and a maximum of 10 items. On average customers ordered **2.06** items per order. The following frequency plot gives us the top 20 most ordered items:



2) Choose an appropriate rule set and report your findings with a focus on rules that under a rule length of 5.

Iteration#1: # Training apriori on the dataset with **supp = 0.001, conf = 0.8**
Result:



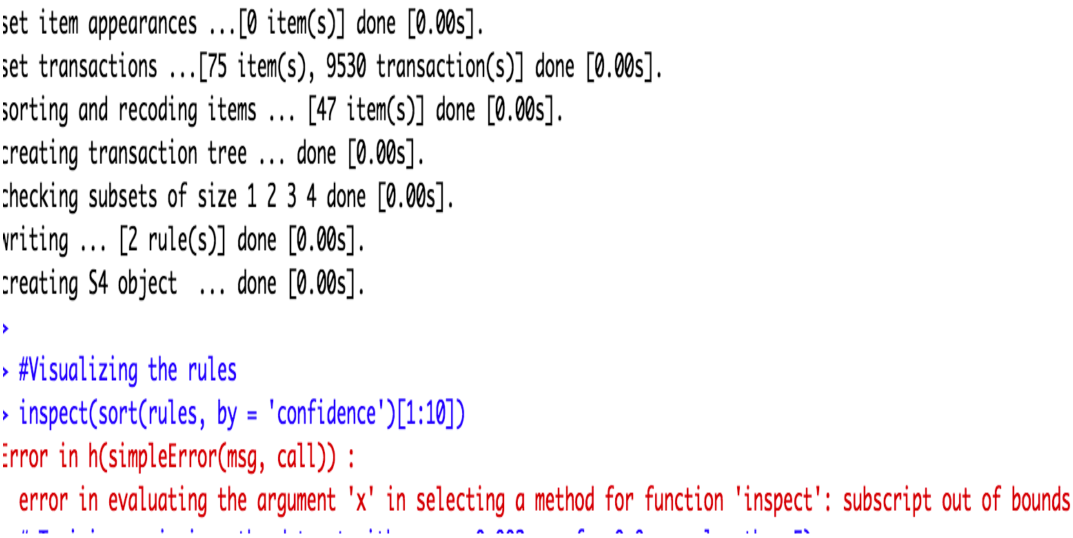
```
     lhs                                         rhs      support confidence coverage lift count
[1]  {Espresso, Tartine}                      => {Coffee} 0.0013  0.92       0.0014   1.9  12
[2]  {American Breakfast, Espresso, Hot chocolate} => {Coffee} 0.0010  0.91       0.0012   1.9  10
[3]  {Extra Salami or Feta, Salad}            => {Coffee} 0.0015  0.88       0.0017   1.8  14
[4]  {Pastry, Toast}                          => {Coffee} 0.0014  0.87       0.0016   1.8  13
[5]  {Club Sandwich, Hearty & Seasonal}       => {Coffee} 0.0013  0.86       0.0015   1.8  12
[6]  {Apple Danish, Cake}                     => {Coffee} 0.0010  0.83       0.0013   1.8  10
[7]  {Club Sandwich, Salad}                   => {Coffee} 0.0016  0.83       0.0019   1.8  15
[8]  {Espresso, Scone}                        => {Coffee} 0.0016  0.83       0.0019   1.8  15
[9]  {Extra Salami or Feta}                   => {Coffee} 0.0033  0.82       0.0040   1.7  31
[10] {Keeping It Local}                       => {Coffee} 0.0054  0.81       0.0066   1.7  51
```

With support = 0.001 & confidence = 0.8, yielded only 10 rules. And in all 10, Coffee featured in all the rhs. This is not surprising, given the fact that Coffee has a huge ‘support’ with almost 50% of the transactions contained ‘Coffee’. To get a meaningful set of rules, 3 things can be tried:

1. Increase support level
2. Decrease confidence level
3. Sort by ‘lift’ instead of ‘confidence’

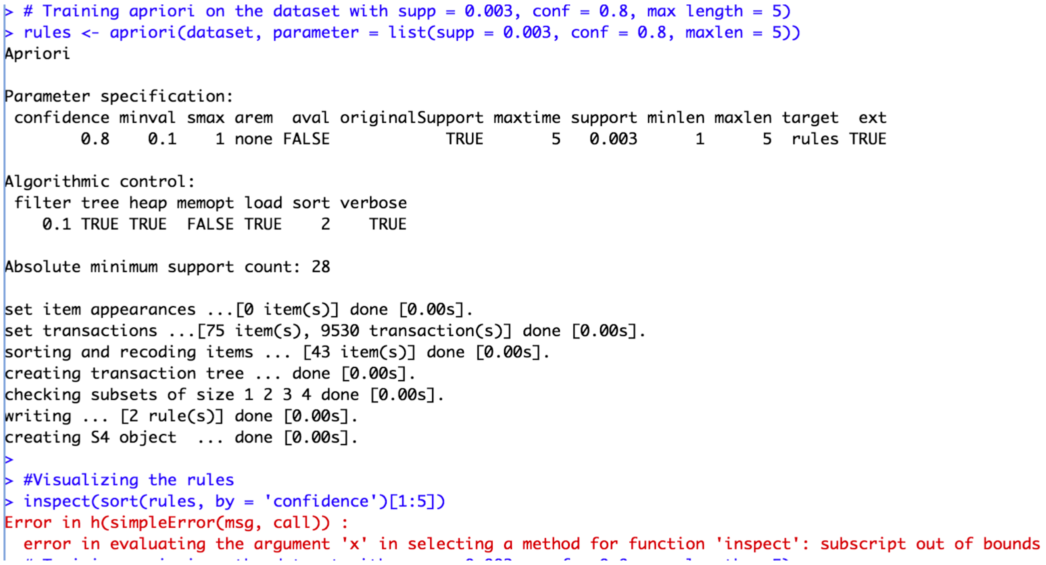
Increasing support level yielded hardly any results, as we can see from the following:

rules <- apriori(dataset, parameter = list(supp = **0.002, conf = 0.8**, maxlen = 5))



```
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[75 item(s), 9530 transaction(s)] done [0.00s].
sorting and recoding items ... [47 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [2 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>
> #Visualizing the rules
> inspect(sort(rules, by = 'confidence')[1:10])
Error in h(simpleError(msg, call)) :
  error in evaluating the argument 'x' in selecting a method for function 'inspect': subscript out of bounds
```

>rules <- apriori(dataset, parameter = list(supp = **0.003, conf = 0.8**)

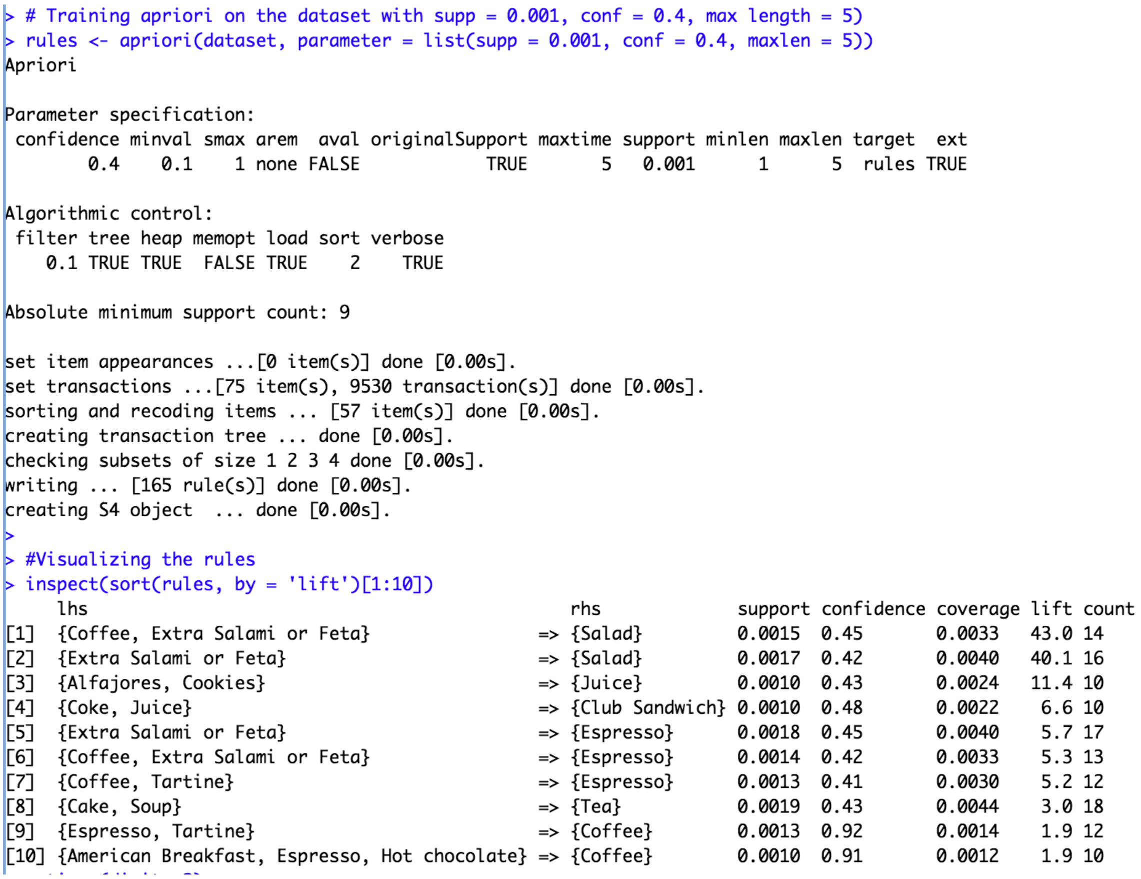


```
> # Training apriori on the dataset with supp = 0.003, conf = 0.8, max length = 5)
> rules <- apriori(dataset, parameter = list(supp = 0.003, conf = 0.8, maxlen = 5))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5   0.003      1      5  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 28

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[75 item(s), 9530 transaction(s)] done [0.00s].
sorting and recoding items ... [43 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [2 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>
> #Visualizing the rules
> inspect(sort(rules, by = 'confidence')[1:5])
Error in h(simpleError(msg, call)) :
  error in evaluating the argument 'x' in selecting a method for function 'inspect': subscript out of bounds
```

3) What limitations does your market basket analysis have?

One main limitation of this analysis is, since the rules are sorted by ‘confidence’ instead of ‘lift’, high support items like ‘Coffee’ tend to misrepresent the importance of association. This is because confidence measure only accounts for how popular Coffee is, but not, let us say ‘Salad’. If ‘Salad’ is also very popular in general, there will be a higher chance that a transaction containing Coffee will also contain bread, thus inflating the confidence measure. To account for the base popularity of both constituent items we therefore, need a third measure called **Lift.** Lift gives the correlation between A and B in the rule A=>B, how one item-set A affects the item-set B.

Therefore, to overcome the problem high-support item like Coffee appearing in all the rules, we need to
a) increase minimum support. But this option yielded errors, and therefore, no results as there were any combination of products that qualified the parameters of the rules.
b) decrease the minimum confidence limit
b) sort the rules by ‘lift’.
Here are the results if we adopt the above principles:



```
> # Training apriori on the dataset with supp = 0.001, conf = 0.4, max length = 5)
> rules <- apriori(dataset, parameter = list(supp = 0.001, conf = 0.4, maxlen = 5))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.4    0.1    1 none FALSE            TRUE       5   0.001      1      5  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 ...[75 item(s), 9530 transaction(s)] done [0.00s].
sorting and recoding items ... [57 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [165 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>
> #Visualizing the rules
> inspect(sort(rules, by = 'lift')[1:10])
     lhs                                       rhs             support confidence coverage lift count
[1]  {Coffee, Extra Salami or Feta}         => {Salad}         0.0015  0.45       0.0033   43.0 14
[2]  {Extra Salami or Feta}                 => {Salad}         0.0017  0.42       0.0040   40.1 16
[3]  {Alfajores, Cookies}                   => {Juice}         0.0010  0.43       0.0024   11.4 10
[4]  {Coke, Juice}                          => {Club Sandwich} 0.0010  0.48       0.0022    6.6 10
[5]  {Extra Salami or Feta}                 => {Espresso}      0.0018  0.45       0.0040    5.7 17
[6]  {Coffee, Extra Salami or Feta}         => {Espresso}      0.0014  0.42       0.0033    5.3 13
[7]  {Coffee, Tartine}                      => {Espresso}      0.0013  0.41       0.0030    5.2 12
[8]  {Cake, Soup}                           => {Tea}           0.0019  0.43       0.0044    3.0 18
[9]  {Espresso, Tartine}                    => {Coffee}        0.0013  0.92       0.0014    1.9 12
[10] {American Breakfast, Espresso, Hot chocolate} => {Coffee}        0.0010  0.91       0.0012    1.9 10
```

Insights

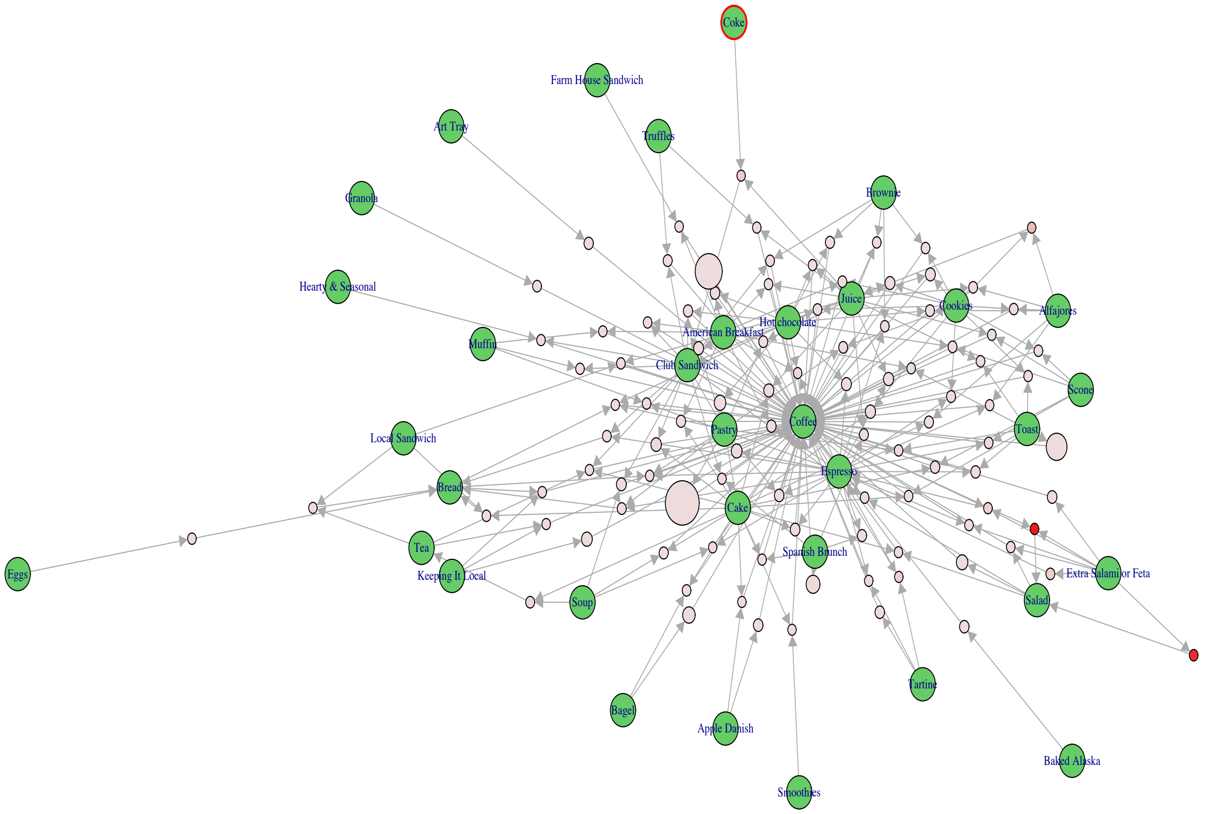
- We can form meaningful insights from the above stats, for example, **rule#4** states 48% of the time a Coke & Juice is bought, a Club Sandwich is bought as well. Is there an opportunity to convert a customer buying just a sandwich to go for a 'Meal Deal' kind of option, where there is a deal if the customer buys a sandwich+drink+snack, where the sum of the products is less than the price of buying separately? Would it push the person to go for the extra snack or a drink, who otherwise would have bought just a sandwich?

- Similarly, since 42 to 45% of the time Extra Salami or Feta is bought along with a Salad **(rule#1&2)**, would introducing a new Salad with Salami or Feta be more popular? This shows, a little tweaking with support, confidence and lift, it is possible to come up with associations that can benefit the customer by delivering right mix, and greater revenue for the business by making use of these patterns in associations.

- **Rule# 6,7&10** suggest that when more than 1 hot drink is purchased is indicative of a group purchase rather than an individual customer. Here's an opportunity, to plug a group deal, of 'Buy 2 Drinks and Get a Snack Free', kind of promotions.

- **Rule#7** - 41% of the cases a light snack like Tartine is accompanied with a hot drink like Coffee. Is there an opportunity to include more lighter snacks like a Tartine, or a Croque Monsieur, something less filling than a full ClubSandwich, in Cafelepetit's product offering?

Finally, an interactive graph can be useful in understanding the connections between the items in a basket:


Coke
Farm House Sandwich
Art Tray
Truffles
Granola
Brownie
Hearty & Seasonal
Juice
Cookies
Alfajores
Hot chocolate
American Breakfast
Muffin
Club Sandwich
Scone
Coffee
Pastry
Toast
Local Sandwich
Espresso
Bread
Cake
Spanish Brunch
Tea
Keeping It Local
Extra Salami or Feta
Eggs
Soup
Salad
Tartine
Bagel
Apple Danish
Baked Alaska
Smoothies