Pontificia Universidad Católica Madre y Maestra

Campus de Santiago

Subject:

Business Intelligence

Teacher:

Lisibonny Beato

Title:

Final Project about data mining

Name/ID:

Gabriel Cepeda Garcia

ID: 1014-1803

Santiago, 08 de Diciembre de 2022

The cafeteria is an establishment offering a wide range of food items, from mints and chewing gum to elaborately prepared meals like sandwiches and hot dogs. One of the most common issues in cafés is the lack of variety in customer purchases; typically, customers tend to opt for specific products when making a purchase. Consequently, other products are underutilized, leading to waste and business losses. By leveraging high-demand products to create combos with those that have lower demand, it's possible to boost sales of the less popular items.

To achieve this objective, I focused on using an association rules algorithm, which identifies relationships within a set of transactions—items that tend to occur together. The algorithm of choice was the Apriori algorithm, which involves identifying all items that occur with a frequency above a certain threshold and then converting these into association rules.

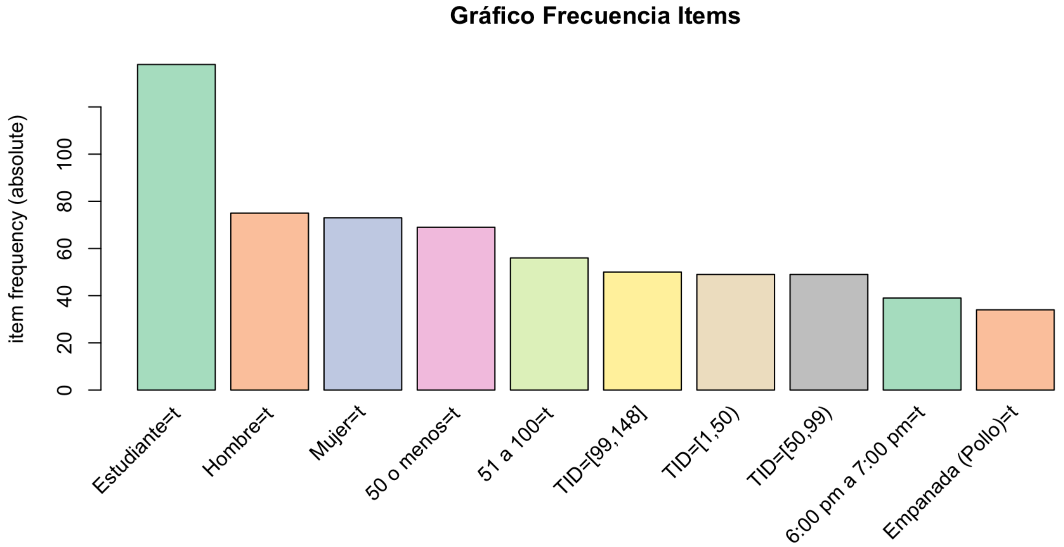
The dataset containing transactional purchases made at the kiosk is in arff format, ASCII text files describing a list of instances with common attributes. This dataset comprises 148 rows representing transaction quantities and 99 columns indicating various variables. The 'summary' function provides an overview of the dataset, including frequently recurring attributes in transactions, such as 'student,' 'male,' 'female,' '50 or less,' and '51 to 100.' The last two refer to the amount in Dominican pesos spent on these transactions.

```
> summary(kiosco2)
transactions as itemMatrix in sparse format with
 148 rows (elements/itemsets/transactions) and
 99 columns (items) and a density of 0.07869233

most frequent items:
Estudiante=t      Hombre=t       Mujer=t 50 o menos=t    51 a 100=t       (Other)
         138            75            73            69            56           742

element (itemset/transaction) length distribution:
sizes
 6  7  8  9 10
 3 58 56 29  2

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.000   7.000   8.000   7.791   8.000  10.000

includes extended item information - examples:
        labels variables    levels
1    TID=[1,50)       TID    [1,50)
2   TID=[50,99)       TID   [50,99)
3 TID=[99,148]       TID [99,148]
```

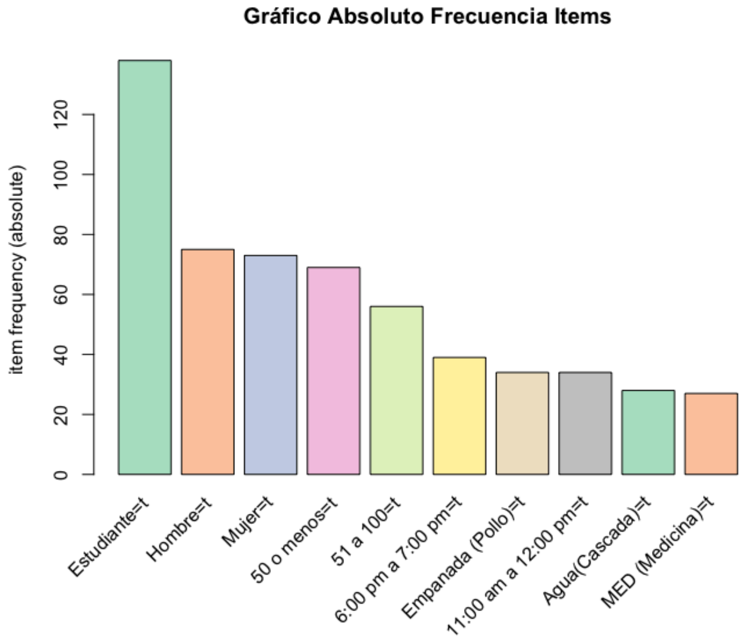
When inspecting individual transactions using the 'inspect' function, I noticed that the Tid is counted as an attribute. Consequently, this function allows me to identify values that may affect the quality of my model.

```
[145] {TID=[99,148],
       Agua(Cascada)=t,
       Hombre=t,
       Estudiante=t,
       ADM (Adm. Empresa)=t,
       3:00 pm a 4:00 pm=t,
       50 o menos=t}                                    145
[146] {TID=[99,148],
       Country Club Rojo=t,
       Mujer=t,
       Estudiante=t,
       ADM (Adm. Empresa)=t,
       3:00 pm a 4:00 pm=t,
       50 o menos=t}                                    146
[147] {TID=[99,148],
       Coca Cola=t,
       Hombre=t,
       Estudiante=t,
       ISC (Ingenier�_a Sistema)=t,
       7:00 pm a 8:00 pm=t,
       50 o menos=t}                                    147
[148] {TID=[99,148],
       Coca Cola=t,
       Hombre=t,
       Estudiante=t,
       ISC (Ingenier�_a Sistema)=t,
       7:00 pm a 8:00 pm=t,
       50 o menos=t}                                    148
```

Plotting the data using the 'ItemFrecuencyPlot' function visually confirms theoretical expectations. Transaction IDs' attributes are displayed as frequent items, which do not assist in achieving my objective.



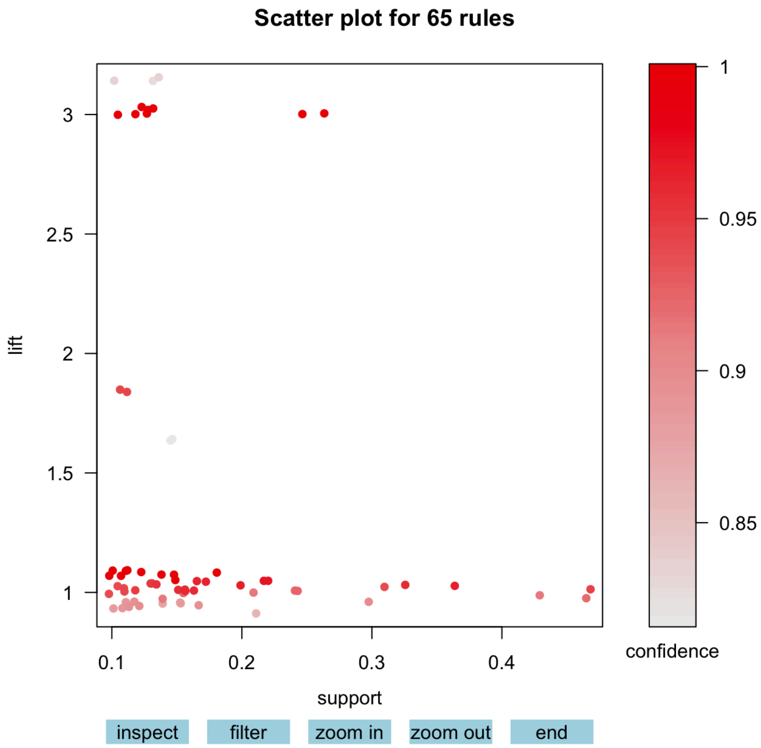
To improve the model, I had to transform the data and eliminate attributes that hindered the construction of a good model. After modifying the dataset, we obtained an interesting dataset suitable for our purpose.



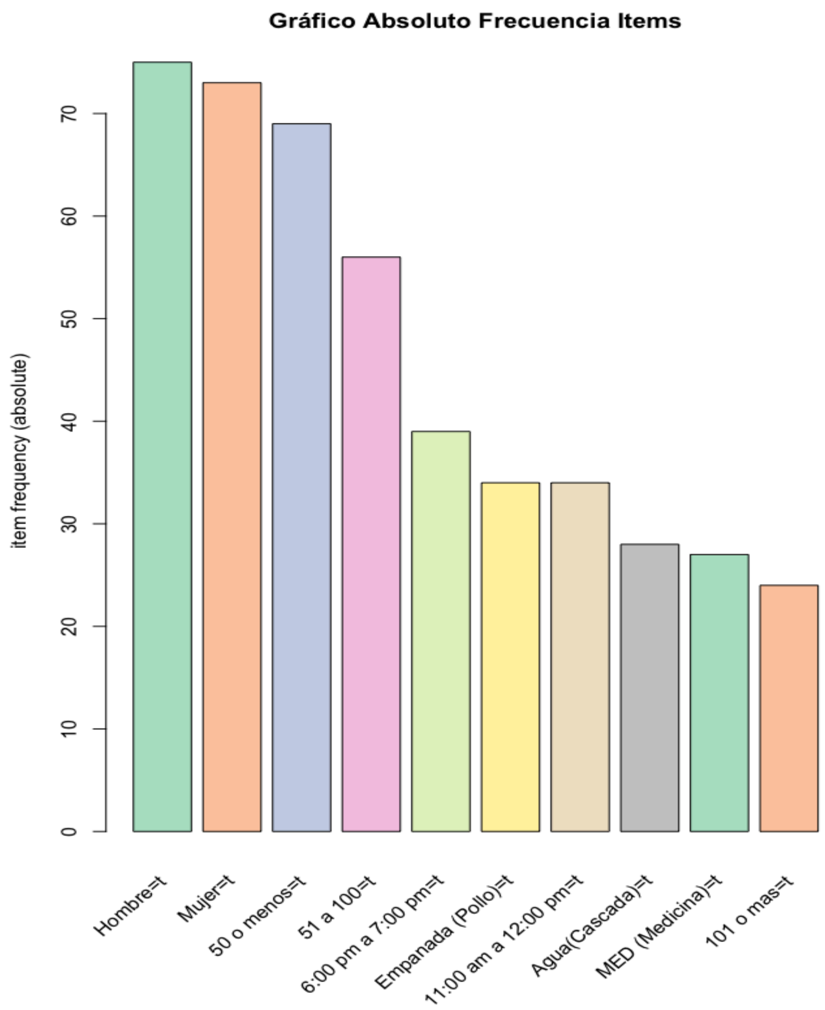
To develop the model, I applied the Apriori algorithm, specifying certain parameters as references for rule selection. These parameters include support, confidence, minLen, and maxLen. The support of item X is the number of transactions containing X divided by the total number of transactions. For our case, we are interested in items with a minimum sale of at least 4. Therefore, we applied the formula support = 15/148 = 0.1. Confidence is interpreted as the probability of a transaction containing items X also containing items Y. We are interested in a minimum confidence of 0.8 or 80%. Finally, minLen and maxLen indicate the minimum and maximum number of items contained in an association rule. For our case, the minimum is 2 items and the maximum is 4.

After executing the Apriori function, we obtained a total of 65 association rules. To view the graph and rules interactively, I used the 'plot' function with the 'interactive' parameter.

Clicking twice in an area and selecting options from below displays the rules included in that area. This shaded area represents the selected rules. This graph allows observation of each rule's behavior, as well as the relationship between confidence, lift, and support. An interesting observation is the presence of rules with a very high lift, indicating that although the rule is a pattern in transactions, the items have low support. Therefore, items did not see many purchases relative to the total transactions.



To further refine the model, it would be advisable to exclude the variable that repeats in all transactions, 'student.' After transformation, the new dataset without the 'student' variable was obtained.



Applying the Apriori algorithm again with the same parameters as before yielded only 2 rules, as opposed to the 65 rules previously generated.

```
> modelo <- apriori(kiosco, parameter = list(supp=0.1, conf=0.8,minlen=2,maxlen=4))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      2      4  rules TRUE

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

Absolute minimum support count: 14

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[95 item(s), 148 transaction(s)] done [0.00s].
sorting and recoding items ... [16 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [2 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

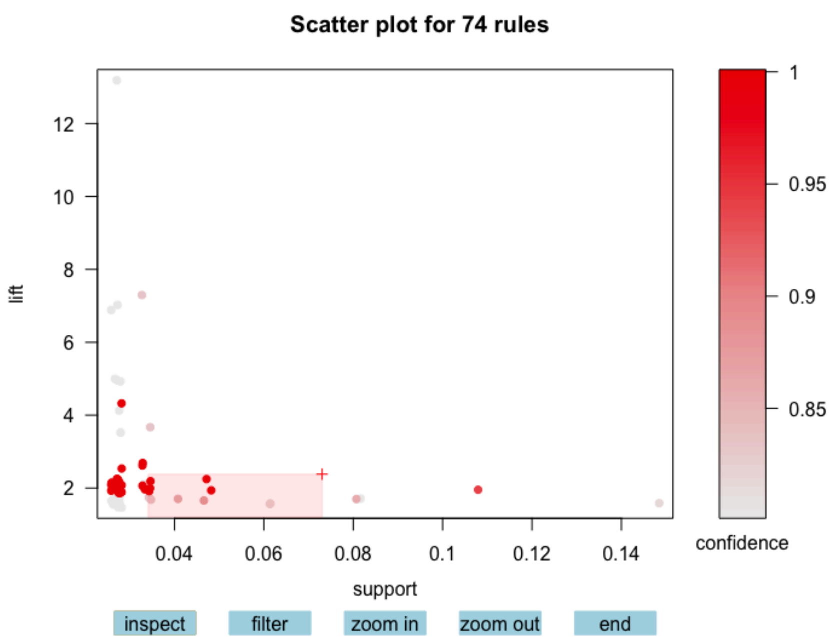
Upon inspection, I realized that the two rules generated by the algorithm were not relevant to our objective, as they did not involve the products. Therefore, to address this, I lowered the support parameter of the algorithm.

```
> inspect(modelo)
    lhs                                  rhs          support   confidence coverage  lift     count
[1] {ISC (Ingenier�_a Sistema)=t} => {Hombre=t} 0.1081081 0.9411765  0.1148649 1.857255 16
[2] {MED (Medicina)=t}            => {Mujer=t}  0.1486486 0.8148148  0.1824324 1.651953 22
>
```

With the modified support parameter to 0.025, we obtained 74 rules. Graphing the model interactively, we can observe the different selected rules.

```
> modelo <- apriori(kiosco, parameter = list(supp=0.025, conf=0.8,minlen=2,maxlen=4))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5   0.025      2      4  rules TRUE

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

Absolute minimum support count: 3

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[95 item(s), 148 transaction(s)] done [0.00s].
sorting and recoding items ... [48 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [74 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Graficando el modelo de manera interactiva podemos observar las diferentes reglas seleccionadas de manera interactiva:



```
Interactive mode.
Select a region with two clicks!

Number of rules selected: 6
    lhs                                         rhs             support    confidence coverage   lift     count order id
[1] {IIS (Ingenier�_a Industrial)=t}         => {50 o menos=t} 0.04729730 1.0000000  0.04729730 2.144928 7     2      9
[2] {Agua(Cascada)=t, MED (Medicina)=t}      => {Mujer=t}      0.04729730 1.0000000  0.04729730 2.027397 7     3     55
[3] {ISC (Ingenier�_a Sistema)=t, 51 a 100=t} => {Hombre=t}     0.04729730 0.8750000  0.05405405 1.726667 7     3     45
[4] {Empanada (Pizza)=t, 6:00 pm a 7:00 pm=t} => {Mujer=t}      0.04054054 0.8571429  0.04729730 1.737769 6     3     49
[5] {Jugo de Carton Rica peq=t}              => {Mujer=t}      0.06081081 0.8181818  0.07432432 1.658780 9     2     12
[6] {6:00 pm a 7:00 pm=t, 101 o mas=t}       => {Mujer=t}      0.06081081 0.8181818  0.07432432 1.658780 9     3     58
```

Here we can observe a better balance among the items involved, as there are fewer rules with a lift value of 1, indicating randomness, and more rules with a lift greater than 1, suggesting a departure from randomness.

The lift value serves as evidence that the rule represents a real pattern rather than a random artifact. Sometimes, we may want to examine rules involving a specific item to identify which products prompt the purchase of another product X. In the Apriori function, there's a parameter called "appearance" that allows us to specify which items we want to appear on the left-hand side (LHS) or right-hand side (RHS) of the rule.

For our case, it would be interesting to observe which products are most frequently purchased when water is bought. Therefore, we execute the function with the appropriate parameters to achieve this.

```r
modelo_find <- apriori(kiosco, parameter = list(supp=0.025, conf=0.6,minlen=2),
                       appearance = list(default="lhs",rhs="Agua(Cascada)=t"))
```

And thus, we can observe 4 rules that are fulfilled with a confidence level of over 65%.

```
> inspect(modelo_find)
    lhs                                          rhs               support    confidence coverage   lift     count
[1] {Hombre=t, 3:00 pm a 4:00 pm=t}           => {Agua(Cascada)=t} 0.02702703 0.6666667  0.04054054 3.523810 4
[2] {12:00 pm a 1:00 pm=t, 50 o menos=t}      => {Agua(Cascada)=t} 0.03378378 0.6250000  0.05405405 3.303571 5
[3] {Hombre=t, 3:00 pm a 4:00 pm=t, 50 o menos=t} => {Agua(Cascada)=t} 0.02702703 0.8000000  0.03378378 4.228571 4
[4] {Mujer=t, 12:00 pm a 1:00 pm=t, 50 o menos=t} => {Agua(Cascada)=t} 0.02702703 0.6666667  0.04054054 3.523810 4
```

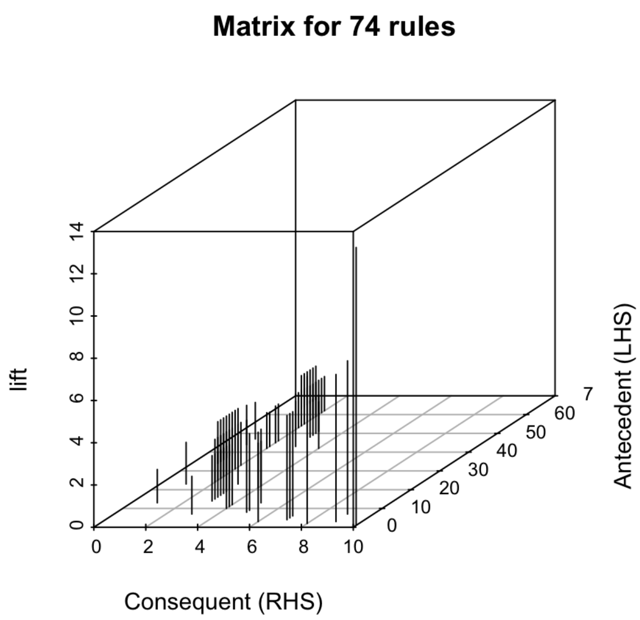
We can continue examining other rules, such as those items purchased when the total expenditure is between 51 and 100 pesos.

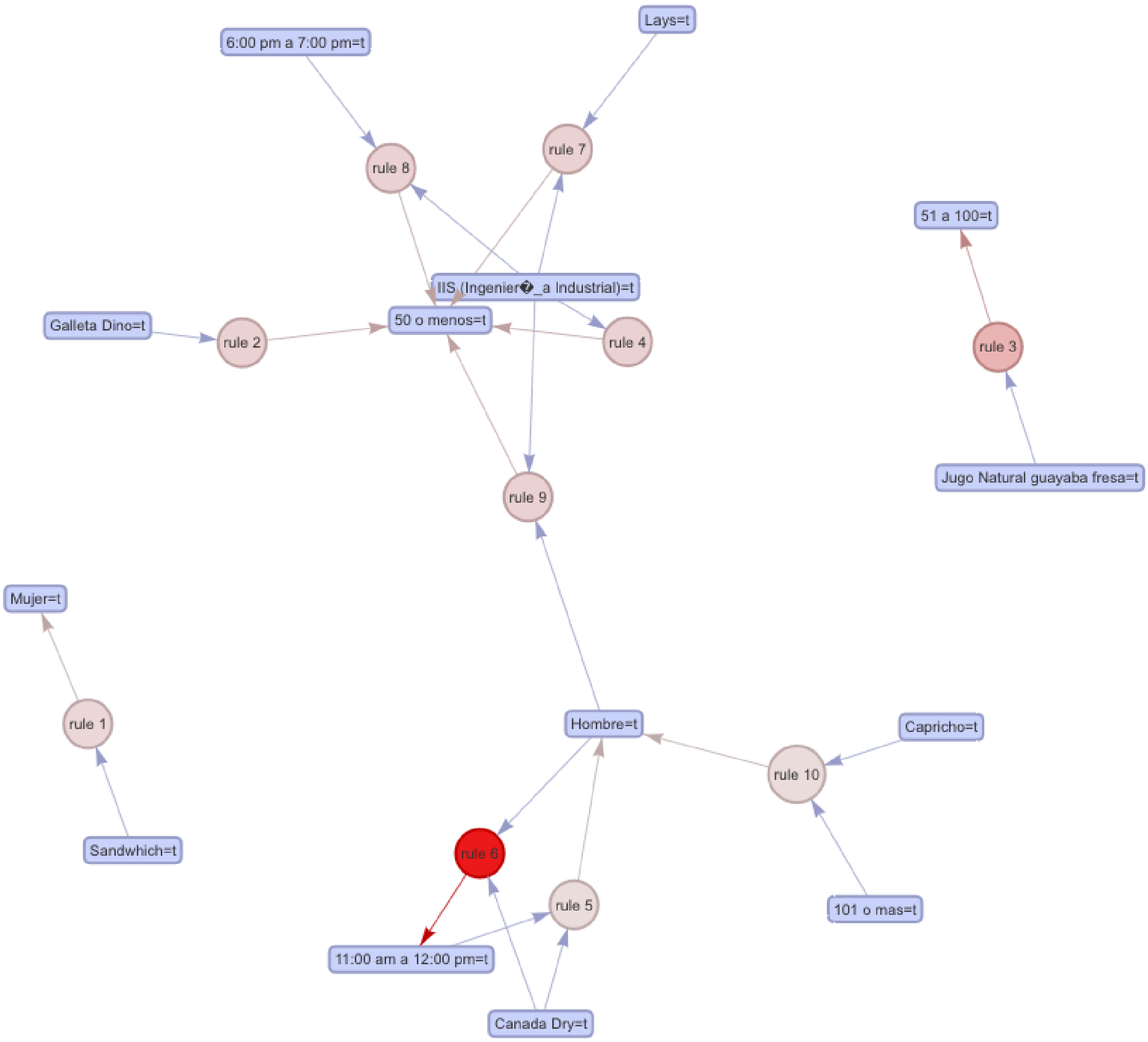
Although the support is low, the confidence is high, and the lift is greater than 1, indicating strong rules.

```
modelo_find <- apriori(kiosco, parameter = list(supp=0.025, conf=0.8,minlen=2),
                       appearance = list(default="lhs",rhs="51 a 100=t"))
```

```
> inspect(modelo_find)
    lhs                                         rhs          support    confidence coverage   lift     count
[1] {Jugo Natural guayaba fresa=t}           => {51 a 100=t} 0.03378378 1.0000000  0.03378378 2.642857 5
[2] {Jugo Natural chinola=t}                 => {51 a 100=t} 0.03378378 0.8333333  0.04054054 2.202381 5
[3] {Jugo Natural chinola=t, Mujer=t}        => {51 a 100=t} 0.02702703 0.8000000  0.03378378 2.114286 4
[4] {Empanada (Pizza)=t, Iced Tea de limon=t} => {51 a 100=t} 0.02702703 1.0000000  0.02702703 2.642857 4
[5] {Hombre=t, MCT (Mercadotecnia)=t}        => {51 a 100=t} 0.02702703 0.8000000  0.03378378 2.114286 4
[6] {Empanada (Pollo)=t, DER (Derecho)=t}    => {51 a 100=t} 0.03378378 1.0000000  0.03378378 2.642857 5
```

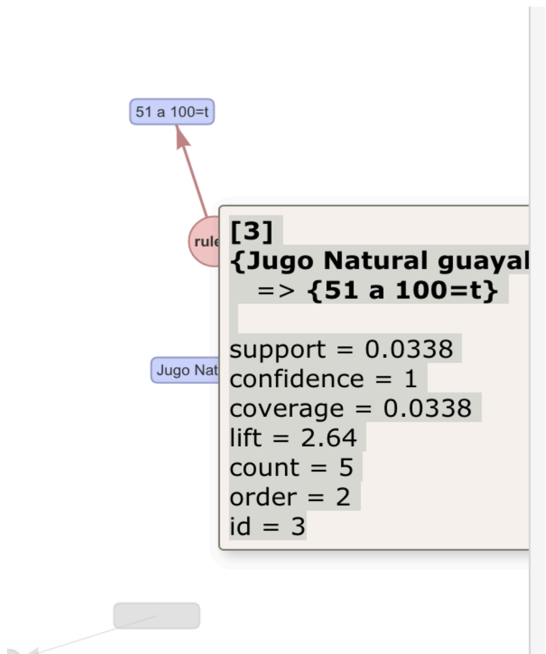
Additionally, we can visualize a 3D graph of the rules, where the antecedent and consequent have a lift greater than 1, indicating patterns.



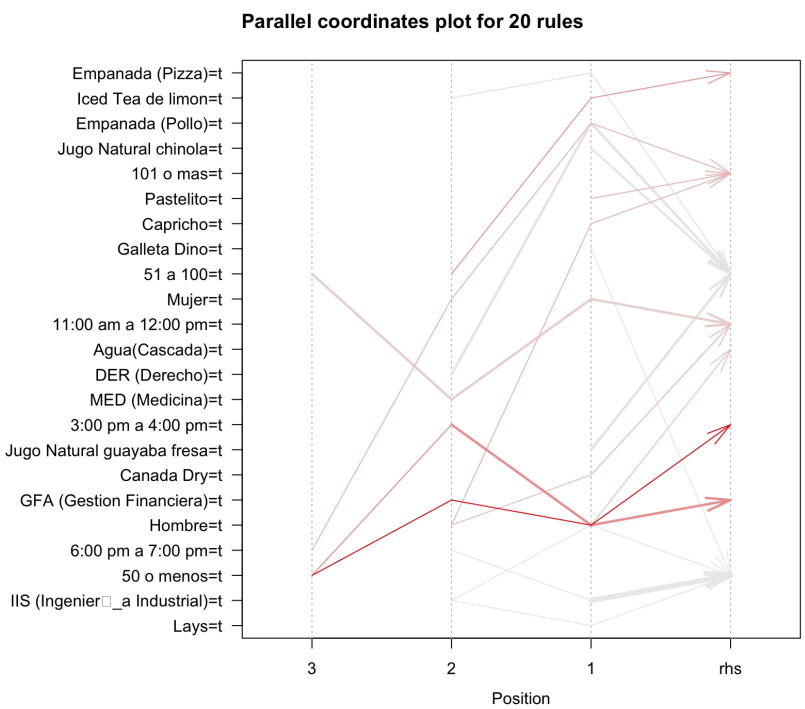


Another interesting graphic we can create is a graph using vertices and edges to represent the rules. Vertices are labeled with element names, and rules are represented as a second set of vertices. Elements are connected to rule sets via directed arrows. However, when there are many rules, this graph can become cluttered, so we'll obtain the top 10 rules with the highest confidence and plot this result as a graphic using the "plot" function, changing the parameter "method" to graph.

For instance, Rule 3 indicates that if a natural guava strawberry juice is purchased, the expenditure is between 51 and 100 pesos. To further explore this rule, we can use an interactive graph and click on the rule for more details. This rule has a confidence of 100%, indicating that it follows a pattern 100% of the time.



Another type of graph that provides more information about individual items is the parallel coordinates graph, which displays antecedents and consequents. For example, the arrow indicates that if an ice tea and an empanada are purchased, the expenditure is between 51 and 100 pesos. All of this information is presented graphically, making it easy to understand.



In conclusion, with this model, business owners can evaluate which products or variables can be combined into combos to increase the sales of those products with less significant sales, leveraging those products that do have significant sales when purchased together. Additionally, this approach addresses one of the most common issues in café-type businesses, where there is a wide variety of offerings.

Bibliografía

Coder, R. (2021, 18 noviembre). Plot in R. R CODER. *https://r-coder.com/plot-r/*

Kumar, K. (s.f.). *Visualize Market Basket analysis in R DataScience*. https://datascienceplus.com/visualize-market-basket-analysis-in-r/

Market Basket Analysis using R. (2018, agosto). DataCamp. https://www.datacamp.com/tutorial/market-basket-analysis-r