Faculty of Computing and Informatics (FCI)

TDS3301 DATA MINING

Assignment Part 2

Prepared by:

Pay Eong Ian	1122703150
Tan Suk Mei	1132700522
Chew Yoke Teng	1132701833
Loh Yun Kuen	1131122813

PART 2: ASSOCIATION RULE MINING

1. Objectives

Association rule mining is mainly dedicated on finding the frequent co-occurring associations or correlations among a group of items from datasets. It is occasionally used for "**Market-Basket Analysis**". We perform this analysis using the **Apriori Algorithm** in R to identify the frequent patterns of the bakery itemsets. The goal of this technique is to discover the associations or correlations of the items that frequently occur together more than what we are expecting from a random sampling of all possibilities. In other words, it is typically used to find the trends or patterns from the data. Through the result of this technique, we can actually gain an insight of which items are often been acquired or purchased together. Hence, it can help the bakery chain to make some adjustments in order to raise the profit of sales.

2. Dataset Description

We have chosen the dataset in **full binary vector with 1000 receipts** in a bakery chain. This bakery chain has about 40 pastry items and 10 coffee drinks. The dataset contains 1000 rows representing the total transactions and 51 columns where 50 columns to represent 50 types of products and another column to show the number of receipts in the bakery chain.

Preprocessing and Decision to Ignore an Attribute

We have decided to ignore an attribute by removing the column which shows the number of receipts, because it is not useful in this case. Moreover, we have also converted the data into a set of transactions where each column is translated from numbers into item names.

Parameter Settings

Confidence value of 0.8 indicates when someone buys Product A, we can know that they are 80% likely to buy Product B. We have also set 0.002 as the **support** value**,** because it will show the transactions in the data involve a specific product purchases that happen at least twice. Whereas the **minlen** value, we have set it to be 2 in order to view the result with at least 2 products in the dataset.

3. Rule Mining Process

Parameter settings:

Confidence value: **0.8**

Support value: **0.002**

Minlen: **2**

Choice of algorithm: Apriori Algorithm with arules and arulesViz packages

Time required: The run time for experiment in Assignment-pt2.R

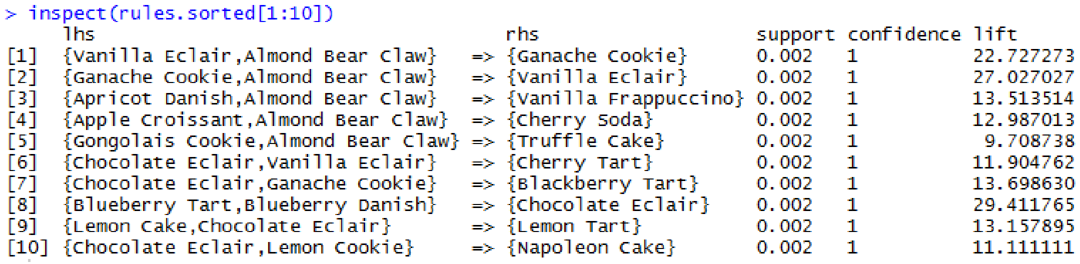
User run time: 6.487 seconds

System run time: 0.515 seconds

Time elapsed: 9.491 seconds

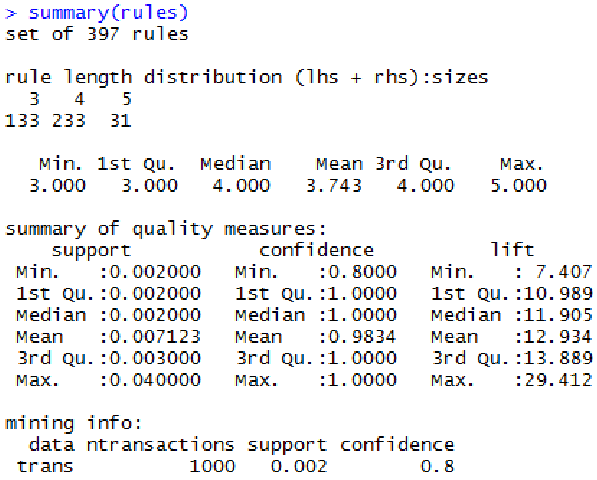
Refer to Assignment-pt2.R for all of the rule mining process.

4. Resulting Rules



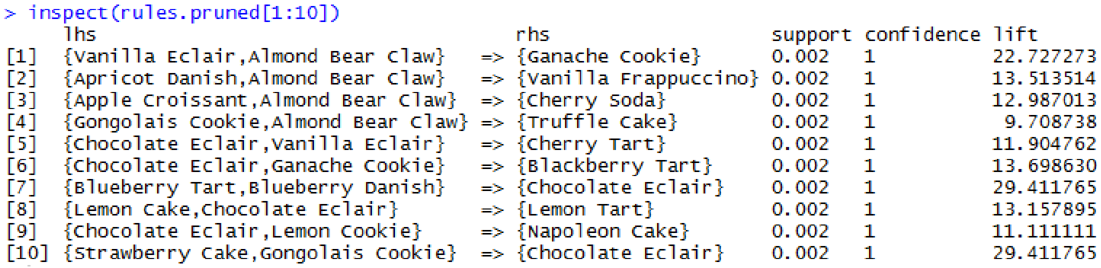
```
> inspect(rules.sorted[1:10])
     lhs                                 rhs                  support confidence lift
[1]  {Vanilla Eclair,Almond Bear Claw}   => {Ganache Cookie}      0.002   1          22.727273
[2]  {Ganache Cookie,Almond Bear Claw}   => {Vanilla Eclair}      0.002   1          27.027027
[3]  {Apricot Danish,Almond Bear Claw}   => {Vanilla Frappuccino} 0.002   1          13.513514
[4]  {Apple Croissant,Almond Bear Claw}  => {Cherry Soda}         0.002   1          12.987013
[5]  {Gongolais Cookie,Almond Bear Claw} => {Truffle Cake}        0.002   1           9.708738
[6]  {Chocolate Eclair,Vanilla Eclair}   => {Cherry Tart}         0.002   1          11.904762
[7]  {Chocolate Eclair,Ganache Cookie}   => {Blackberry Tart}     0.002   1          13.698630
[8]  {Blueberry Tart,Blueberry Danish}   => {Chocolate Eclair}    0.002   1          29.411765
[9]  {Lemon Cake,Chocolate Eclair}       => {Lemon Tart}          0.002   1          13.157895
[10] {Chocolate Eclair,Lemon Cookie}     => {Napoleon Cake}       0.002   1          11.111111
```

The first 10 rules out of the 397 set of rules generated by the Apriori algorithm is shown above. This is the result before pruning out the redundant set of rules. In order to get the accurate result, we will have to remove the redundant rules.



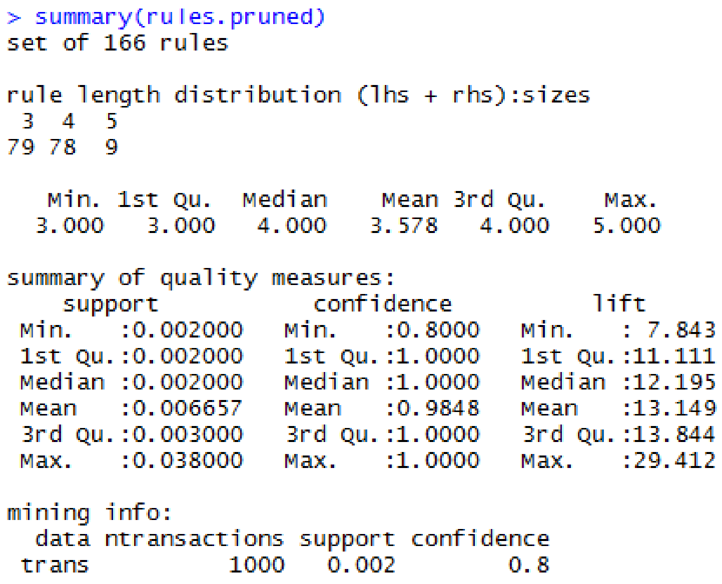
```
> summary(rules)
set of 397 rules

rule length distribution (lhs + rhs):sizes
  3   4   5 
133 233  31 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  3.000   3.000   4.000   3.743   4.000   5.000 

summary of quality measures:
    support           confidence          lift       
 Min.   :0.002000   Min.   :0.8000   Min.   : 7.407  
 1st Qu.:0.002000   1st Qu.:1.0000   1st Qu.:10.989  
 Median :0.002000   Median :1.0000   Median :11.905  
 Mean   :0.007123   Mean   :0.9834   Mean   :12.934  
 3rd Qu.:0.003000   3rd Qu.:1.0000   3rd Qu.:13.889  
 Max.   :0.040000   Max.   :1.0000   Max.   :29.412  

mining info:
  data ntransactions support confidence
 trans          1000   0.002        0.8
```

From the above result, we knew that there is a total of 397 set of rules can be found in this data set. There are 133 rules with 3 products in item, 233 rules with 4 products in item and 31 rules with 5 products in item. As there are some redundant rules in the dataset, 397 will not be the accurate number of rules. After removing the redundant set of rules, the new total number set of rules will be identified.



```
> inspect(rules.pruned[1:10])
     lhs                                    rhs                   support confidence lift
[1]  {Vanilla Eclair,Almond Bear Claw}   => {Ganache Cookie}      0.002   1          22.727273
[2]  {Apricot Danish,Almond Bear Claw}   => {Vanilla Frappuccino} 0.002   1          13.513514
[3]  {Apple Croissant,Almond Bear Claw}  => {Cherry Soda}         0.002   1          12.987013
[4]  {Gongolais Cookie,Almond Bear Claw} => {Truffle Cake}        0.002   1           9.708738
[5]  {Chocolate Eclair,Vanilla Eclair}   => {Cherry Tart}         0.002   1          11.904762
[6]  {Chocolate Eclair,Ganache Cookie}   => {Blackberry Tart}     0.002   1          13.698630
[7]  {Blueberry Tart,Blueberry Danish}   => {Chocolate Eclair}    0.002   1          29.411765
[8]  {Lemon Cake,Chocolate Eclair}       => {Lemon Tart}          0.002   1          13.157895
[9]  {Chocolate Eclair,Lemon Cookie}     => {Napoleon Cake}       0.002   1          11.111111
[10] {Strawberry Cake,Gongolais Cookie}  => {Chocolate Eclair}    0.002   1          29.411765
```

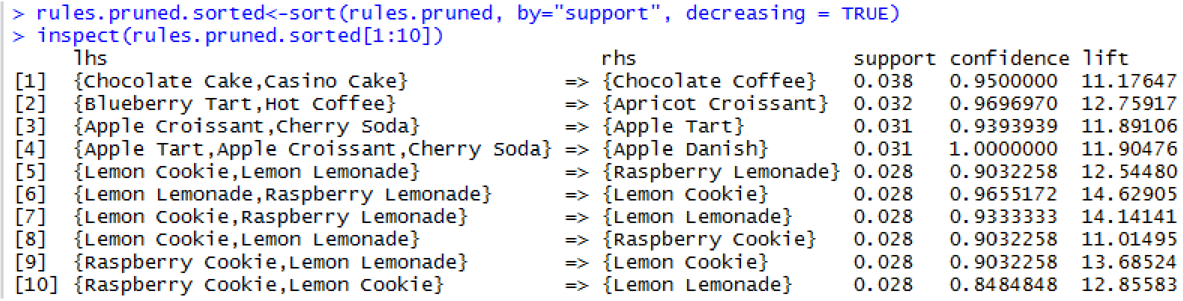
Above has shown the first 10 rules generated by Apriori algorithms after the pruning of data sets.



```
> summary(rules.pruned)
set of 166 rules

rule length distribution (lhs + rhs):sizes
 3  4  5 
79 78  9 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  3.000   3.000   4.000   3.578   4.000   5.000 

summary of quality measures:
    support           confidence          lift       
 Min.   :0.002000   Min.   :0.8000   Min.   : 7.843  
 1st Qu.:0.002000   1st Qu.:1.0000   1st Qu.:11.111  
 Median :0.002000   Median :1.0000   Median :12.195  
 Mean   :0.006657   Mean   :0.9848   Mean   :13.149  
 3rd Qu.:0.003000   3rd Qu.:1.0000   3rd Qu.:13.844  
 Max.   :0.038000   Max.   :1.0000   Max.   :29.412  

mining info:
  data ntransactions support confidence
 trans          1000   0.002        0.8
```

We have removed the redundant set of rules and as a result we got a total of 166 set of rules. In the new set of rules, 79 of them have 3 products while there are 78 set of rules with 4 products. Besides, there are only 9 set of rules with 5 products. From the result, we tried to sort the pruned rules with several parameters such as support, confidence and lift.

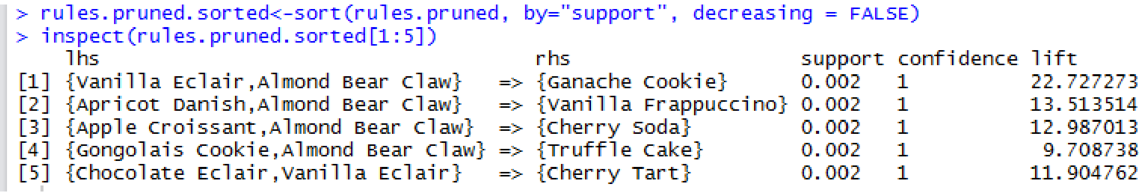
Sorted pruned rules with maximum support:



```
> rules.pruned.sorted<-sort(rules.pruned, by="support", decreasing = TRUE)
> inspect(rules.pruned.sorted[1:10])
     lhs                                      rhs                   support confidence lift    
[1]  {Chocolate Cake,Casino Cake}          => {Chocolate Coffee}    0.038   0.9500000  11.17647
[2]  {Blueberry Tart,Hot Coffee}           => {Apricot Croissant}   0.032   0.9696970  12.75917
[3]  {Apple Croissant,Cherry Soda}         => {Apple Tart}          0.031   0.9393939  11.89106
[4]  {Apple Tart,Apple Croissant,Cherry Soda} => {Apple Danish}     0.031   1.0000000  11.90476
[5]  {Lemon Cookie,Lemon Lemonade}         => {Raspberry Lemonade}  0.028   0.9032258  12.54480
[6]  {Lemon Lemonade,Raspberry Lemonade}   => {Lemon Cookie}        0.028   0.9655172  14.62905
[7]  {Lemon Cookie,Raspberry Lemonade}     => {Lemon Lemonade}      0.028   0.9333333  14.14141
[8]  {Lemon Cookie,Lemon Lemonade}         => {Raspberry Cookie}    0.028   0.9032258  11.01495
[9]  {Raspberry Cookie,Lemon Lemonade}     => {Lemon Cookie}        0.028   0.9032258  13.68524
[10] {Raspberry Cookie,Lemon Cookie}       => {Lemon Lemonade}      0.028   0.8484848  12.85583
```

By sorting the pruned rules with maximum support, we can get the frequent itemsets that appear the most in the datasets. From the result shown above, we can say that the frequent itemsets that appear the most in this dataset is Chocolate Cake, Casino Cake and Chocolate Coffee.

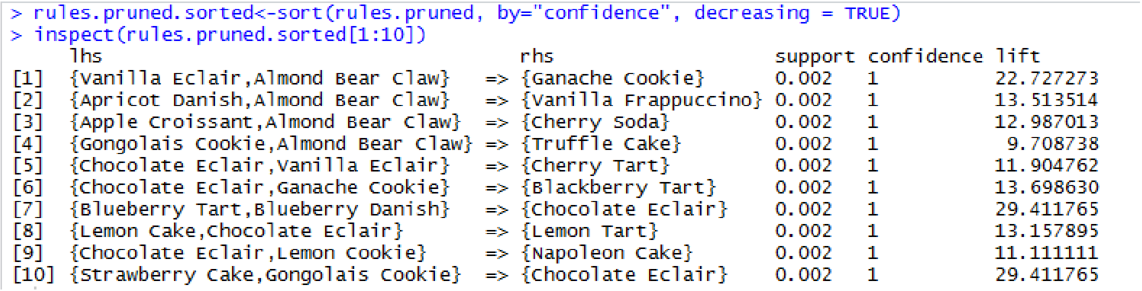
Sorted pruned rules with minimum support:



```
> rules.pruned.sorted<-sort(rules.pruned, by="support", decreasing = FALSE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                      rhs                   support confidence lift
[1] {Vanilla Eclair,Almond Bear Claw}     => {Ganache Cookie}      0.002   1          22.727273
[2] {Apricot Danish,Almond Bear Claw}     => {Vanilla Frappuccino} 0.002   1          13.513514
[3] {Apple Croissant,Almond Bear Claw}    => {Cherry Soda}         0.002   1          12.987013
[4] {Gongolais Cookie,Almond Bear Claw}   => {Truffle Cake}        0.002   1           9.708738
[5] {Chocolate Eclair,Vanilla Eclair}     => {Cherry Tart}         0.002   1          11.904762
```

By sorting the pruned rules with minimum support, we can know the itemsets that appear the less in the datasets. From the result, we understand that itemset with combinations of Ganache Cookie, Almond Bear Claw and Vanilla Eclair appear the less.

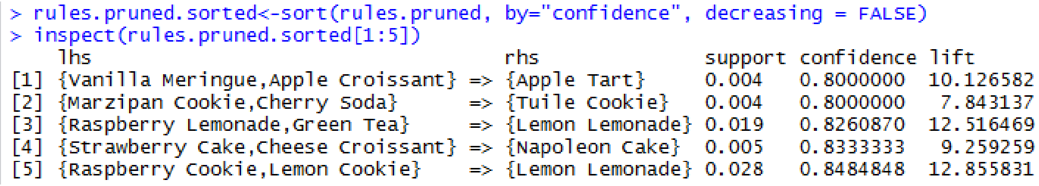
Sorted pruned rules with maximum confidence:



```
> rules.pruned.sorted<-sort(rules.pruned, by="confidence", decreasing = TRUE)
> inspect(rules.pruned.sorted[1:10])
     lhs                                      rhs                   support confidence lift
[1]  {Vanilla Eclair,Almond Bear Claw}     => {Ganache Cookie}      0.002   1          22.727273
[2]  {Apricot Danish,Almond Bear Claw}     => {Vanilla Frappuccino} 0.002   1          13.513514
[3]  {Apple Croissant,Almond Bear Claw}    => {Cherry Soda}         0.002   1          12.987013
[4]  {Gongolais Cookie,Almond Bear Claw}   => {Truffle Cake}        0.002   1           9.708738
[5]  {Chocolate Eclair,Vanilla Eclair}     => {Cherry Tart}         0.002   1          11.904762
[6]  {Chocolate Eclair,Ganache Cookie}     => {Blackberry Tart}     0.002   1          13.698630
[7]  {Blueberry Tart,Blueberry Danish}     => {Chocolate Eclair}    0.002   1          29.411765
[8]  {Lemon Cake,Chocolate Eclair}         => {Lemon Tart}          0.002   1          13.157895
[9]  {Chocolate Eclair,Lemon Cookie}       => {Napoleon Cake}       0.002   1          11.111111
[10] {Strawberry Cake,Gongolais Cookie}    => {Chocolate Eclair}    0.002   1          29.411765
```

By sorting the pruned rules with maximum confidence, it show us the itemsets that have higher probability to be purchased together. The above result shown that if customer purchase Vanilla Eclair and Almond Bear Claw, he or she will definitely purchase Ganache Cookie together as Ganache Cookie has 1 as confidence value.

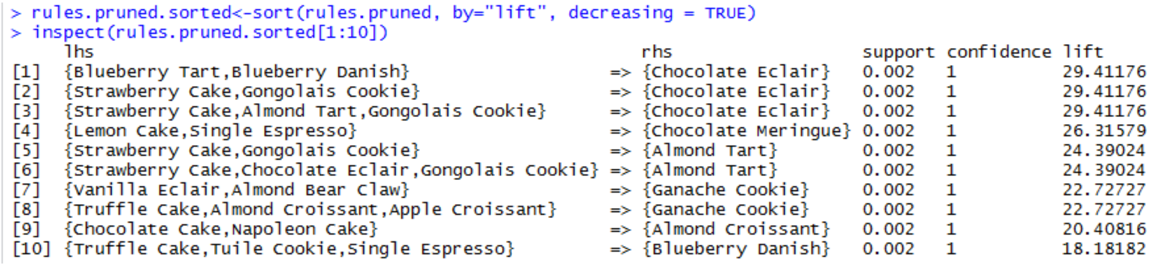
Sorted pruned rules with minimum confidence:



```
> rules.pruned.sorted<-sort(rules.pruned, by="confidence", decreasing = FALSE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                    rhs              support confidence lift
[1] {Vanilla Meringue,Apple Croissant} => {Apple Tart}      0.004   0.8000000  10.126582
[2] {Marzipan Cookie,Cherry Soda}      => {Tuile Cookie}    0.004   0.8000000   7.843137
[3] {Raspberry Lemonade,Green Tea}     => {Lemon Lemonade}  0.019   0.8260870  12.516469
[4] {Strawberry Cake,Cheese Croissant} => {Napoleon Cake}   0.005   0.8333333   9.259259
[5] {Raspberry Cookie,Lemon Cookie}    => {Lemon Lemonade}  0.028   0.8484848  12.855831
```

By sorting the pruned rules with minimum confidence, it will show us the itemsets that have lower probability to be purchased together. The 5 rules shown on above are the itemset that client should consider not to pack together as they carry lower confidence value.

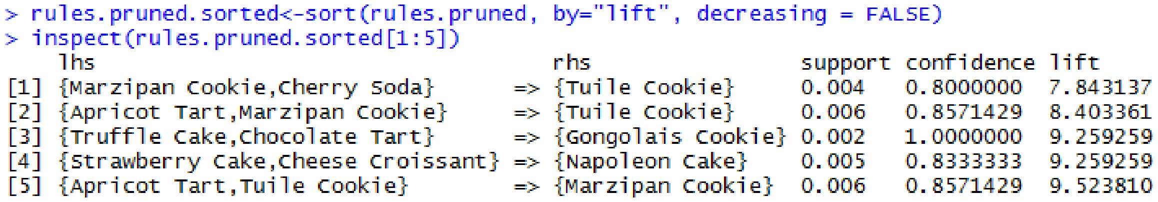
Sorted pruned rules with maximum lift:



```
> rules.pruned.sorted<-sort(rules.pruned, by="lift", decreasing = TRUE)
> inspect(rules.pruned.sorted[1:10])
     lhs                                                rhs                  support confidence lift
[1]  {Blueberry Tart,Blueberry Danish}               => {Chocolate Eclair}   0.002   1          29.41176
[2]  {Strawberry Cake,Gongolais Cookie}              => {Chocolate Eclair}   0.002   1          29.41176
[3]  {Strawberry Cake,Almond Tart,Gongolais Cookie}  => {Chocolate Eclair}   0.002   1          29.41176
[4]  {Lemon Cake,Single Espresso}                    => {Chocolate Meringue} 0.002   1          26.31579
[5]  {Strawberry Cake,Gongolais Cookie}              => {Almond Tart}        0.002   1          24.39024
[6]  {Strawberry Cake,Chocolate Eclair,Gongolais Cookie} => {Almond Tart}    0.002   1          24.39024
[7]  {Vanilla Eclair,Almond Bear Claw}               => {Ganache Cookie}     0.002   1          22.72727
[8]  {Truffle Cake,Almond Croissant,Apple Croissant} => {Ganache Cookie}     0.002   1          22.72727
[9]  {Chocolate Cake,Napoleon Cake}                  => {Almond Croissant}   0.002   1          20.40816
[10] {Truffle Cake,Tuile Cookie,Single Espresso}     => {Blueberry Danish}   0.002   1          18.18182
```

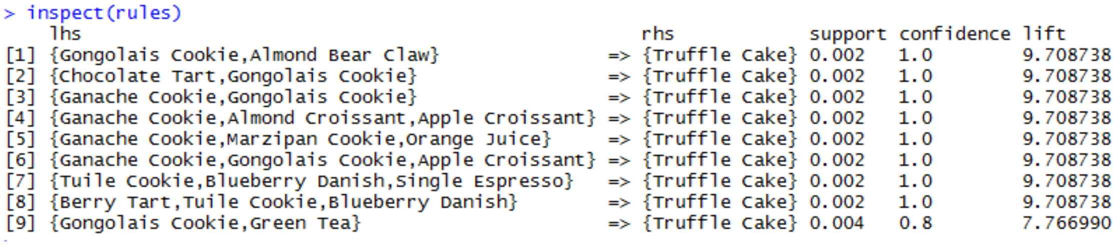
By sorting the pruned rules with maximum lift, we can know the hot selling itemsets in bakery compared with other itemsets. From the observation, we know that combination of products with Chocolate Eclair have the highest sales.

Sorted pruned rules with minimum lift:



```
> rules.pruned.sorted<-sort(rules.pruned, by="lift", decreasing = FALSE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                 rhs                support confidence lift
[1] {Marzipan Cookie,Cherry Soda}       => {Tuile Cookie}     0.004   0.8000000  7.843137
[2] {Apricot Tart,Marzipan Cookie}      => {Tuile Cookie}     0.006   0.8571429  8.403361
[3] {Truffle Cake,Chocolate Tart}       => {Gongolais Cookie} 0.002   1.0000000  9.259259
[4] {Strawberry Cake,Cheese Croissant}  => {Napoleon Cake}    0.005   0.8333333  9.259259
[5] {Apricot Tart,Tuile Cookie}         => {Marzipan Cookie}  0.006   0.8571429  9.523810
```

By sorting the pruned rules with minimum lift, we can know the itemsets that have less sales compared with other combination of itemsets. From the result shown in above, we can observed that the combination of products with Tuile Cookie have the lowest sales.



```
> inspect(rules)
    lhs                                             rhs            support confidence lift
[1] {Gongolais Cookie,Almond Bear Claw}          => {Truffle Cake} 0.002   1.0        9.708738
[2] {Chocolate Tart,Gongolais Cookie}            => {Truffle Cake} 0.002   1.0        9.708738
[3] {Ganache Cookie,Gongolais Cookie}            => {Truffle Cake} 0.002   1.0        9.708738
[4] {Ganache Cookie,Almond Croissant,Apple Croissant} => {Truffle Cake} 0.002   1.0   9.708738
[5] {Ganache Cookie,Marzipan Cookie,Orange Juice} => {Truffle Cake} 0.002   1.0       9.708738
[6] {Ganache Cookie,Gongolais Cookie,Apple Croissant} => {Truffle Cake} 0.002   1.0   9.708738
[7] {Tuile Cookie,Blueberry Danish,Single Espresso} => {Truffle Cake} 0.002   1.0     9.708738
[8] {Berry Tart,Tuile Cookie,Blueberry Danish}   => {Truffle Cake} 0.002   1.0        9.708738
[9] {Gongolais Cookie,Green Tea}                 => {Truffle Cake} 0.004   0.8        7.766990
```

The figure above shown the set of rules with Truffle Cake at right hand side. It helps client to identify which products would be purchased together with Truffle Cake.

Besides, we would like to show client the products that have lower sales in bakery and find a way to improve the product sales. The figures below are the graphs of Top 10 Highest Sales Products and Top 5 Lowest Sales Products in bakery.

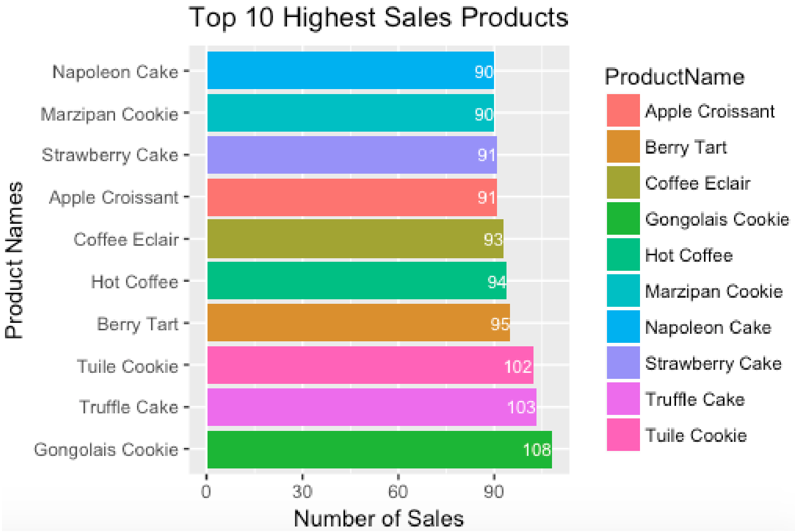


Figure 1 Bar chart of the top 10 highest sales in the bakery

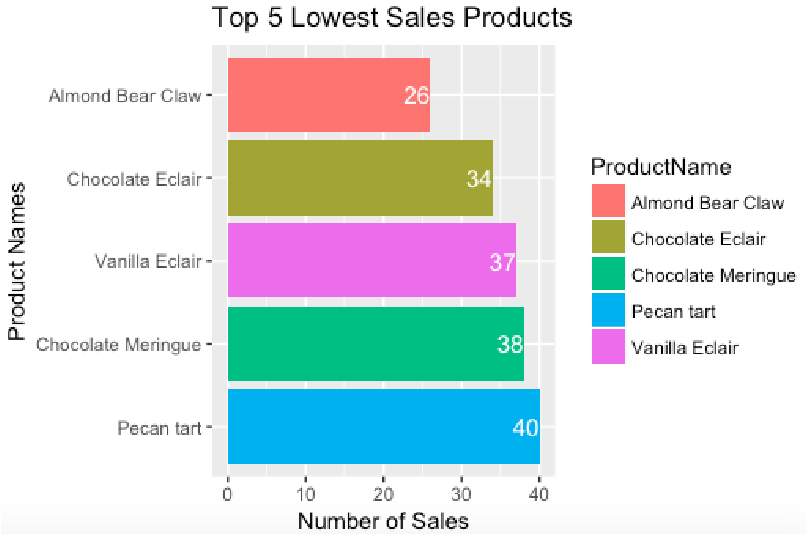


Figure 2 Bar chart of the top 5 lowest sales in the bakery

With these two bar charts, we can now relate the number of product sales with the rules that we have discovered earlier. Apart from that, these charts also give an insight on the product sales of the bakery shop to the client. Based on the graphs, adjustments with the low sales products can be done by clients as recommendations in Question 5.

5. Recommendations

- **Customized coupons**

  As the rules generated help to better understand the purchasing behaviour of consumer, client can offer coupons based on what the customer “may” buy instead of publishing the same coupons for all customers of a store. For instance, if the customer is expected to buy Vanilla Eclair, Almond Bear Claw and Ganache Cookie during every visit, then customized coupons such as RM5 off on 3 packs of Ganache Cookie can be offered to him, instead of issuing a coupon to buy Pecan Tart at a discounted price, which he may not purchase at all. If the customer is purchasing only on Saturdays every week, then he can be offered a coupon that expires on a Wednesday to increase the frequency of his visit to the store.

- **Bundle low sales products with high sales products for discount**

  It is no secret that consumers love sales, coupons, seasonal pricing among other promotion related markdowns. We suggest client to offer several products for sale as one combined product, set a new price for the bundle and give consumers discount value for the bundled purchase. According to the Top 10 Highest Sales Products graph, the highest sales product Gongolais Cookie can be sold together with one of the lowest sales product Vanilla Eclair with an offer price. This strategy should attract more consumers’ attention to low sales products, and get it out of the store quicker.

- **Place the low sales product front and center**

  The products in store shelves can be modified based on the rules discovered. Besides, the area just inside the bakery front door is an ideal spot to place the low sales product. For instance, client may choose to bring items with low sales including Almond Bear Claw and Chocolate Meringue in the shelves at the ideal spot. This gives the product a higher chance of being seen by consumers when they first walk into the bakery, which may also be converted to an unplanned purchase.

- **Provide product tester**

  It’s a proven fact that provide product tester to your customers can help to boost the sales. Based on the graph on Top 5 Lowest Sales Products, items with low sales such as Almond Bear Claw and Chocolate Meringue, we suggest client to provide product tester of these items for users to try out. Once customer tried the product, it will have increased trust and confidence in their purchasing decision. Consequently, it will also lead to a faster sale.

- **Product recommendation**

  According to the graphs generated, we know that some products do not guarantee profit. Thus, we recommend the client to produce more of the top selling products and less of the lower sales products. By producing suitable amount of products, it does not only help the client to increase profit but also minimize loss and reduce product waste at the same time.