1. Objectives: What is the domain and what are the potential benefits to be derived from association rule mining. This is high level - not find patterns, but what would improve because of the use of the patterns.

Association rule mining is a procedure of finding frequent patterns, associations, or correlations from data sets. It is frequently used in transactions data to discover interesting relations between items data and allows us to identify a set of rules that can be understood as “if this, then that”. As such, we will be able to gain insight on what items are always purchased together. Using the information obtained from the patterns, many improvements can be made such as related product recommendation, better shelf arrangement which can allow the store to gain more profits by increasing the sales.

2. Data set description: What is in the data, and what preprocessing was done to make it amenable for association rule mining. Where choices were made (e.g., parameter settings for discretization, or decisions to ignore an attribute), describe your reasoning behind the choices.

The dataset we choose is a **full binary vector representation** dataset which is showing 5000 transactions of 50 foods include 40 pastry items and 10 coffee drinks in a bakery chain. There is one “Receipt No” column representing receipt number and 50 columns representing 50 foods in bakery. Each receipt has its own receipt number followed by 0’s and 1’s indicating if that particular food item was purchased in the transaction, whereby 0 stands for no while 1 stands for yes.

<u>Preprocessing</u>: Firstly, we removed the first column which is ‘Receipt No.’ column because it is not important. The dataset was also converted from dataframe into matrix to be able to read in transaction format. As the items are still represented by numbers which is rather confusing, we converted it into item names. Our dataset are represented in the form below, each row represent a transaction and the contents are the items purchased in that particular transaction.

```
    items
[1] {Strawberry Cake,Truffle Cake,Chocolate Eclair,Almond Tart}
[2] {Lemon Cake,Apple Tart,Lemon Tart,Chocolate Coffee,Vanilla Frappuccino}
[3] {Blueberry Tart,Apricot Croissant}
[4] {Cherry Tart,Lemon Cookie,Apricot Danish}
[5] {Apple Tart,Gongolais Cookie,Marzipan Cookie,Almond Croissant}
```

Decision to ignore an attribute: Other than the removed ‘Receipt No.’ column, we also decided to ignore ‘Quantity’ of purchased, and ‘Price’ of item because they are not important in applying association rules mining.

3. Rule mining process: Parameter settings, choice of algorithm, and the time required.

<u>Parameter setting</u>:
We set **support** to be 0.001, **confidence** to be 0.8 and **minlen** to be 2. This will give the result of 174 rules, which we think that this set of rules is appropriate to observe how is the

market going. Our team have tried several different values of support, confidence and minlen to observe the number of rules provided.

Support: When we set support to 0.002 with the same confidence and minlen value, the set of rules decreases from 174 to 85, that's mean that we get lesser rules to observe how food sell in bakery. However, when support value is 0.001, it results in 127 rules which we think that this number is sufficient for us to conduct our investigation in food sales in the bakery.
Confidence: We set confidence to be 0.08 is because we find the itemsets which carry 80% probability when customer buy a set of food, he or she will also going to buy the other considered food.

Minlen: We set it to 2, it's because we want to view itemsets with minimum 2 items.

Choice of algorithm：
The rules were created using the *apriori* function on the dataset. Package arules is used to apply association rules while package arulesViz is used for additional features such as graphing and plotting the rules.

Time required：
Run time to finish all code for the experiment in Assignment Part 2.R： within 1 minute

4. Resulting rules: Summary (number of rules, general description), and a selection of those you would show to a client.

```
> inspect(rules[1:10])
     lhs                                    rhs                support confidence lift
[1]  {Blackberry Tart,Single Espresso}   => {Coffee Eclair}    0.0286  0.911       8.22
[2]  {Coffee Eclair,Single Espresso}     => {Blackberry Tart}  0.0286  0.966      12.71
[3]  {Coffee Eclair,Blackberry Tart}     => {Single Espresso}  0.0286  0.803      12.28
[4]  {Apple Tart,Cherry Soda}            => {AppleCroissant}   0.0230  0.906      12.20
[5]  {AppleCroissant,Cherry Soda}        => {Apple Tart}       0.0230  0.913      12.43
[6]  {Apple Tart,Cherry Soda}            => {Apple Danish}     0.0228  0.898      11.48
[7]  {Apple Danish,Cherry Soda}          => {Apple Tart}       0.0228  0.912      12.43
[8]  {AppleCroissant,Cherry Soda}        => {Apple Danish}     0.0230  0.913      11.67
[9]  {Apple Danish,Cherry Soda}          => {AppleCroissant}   0.0230  0.920      12.40
[10] {Walnut Cookie,Vanilla Frappuccino} => {Chocolate Tart}   0.0266  0.893      11.71
```

These are the first 10 rules that generated by the apriori algorithm from the 174 set of rules. Before pruning out those redundant and infrequent rules, we cannot really get an accurate rules, so we decided to clean that out.

```
> summary(rules)
set of 174 rules

rule length distribution (lhs + rhs):sizes
  3   4   5
 53 104  17

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   3.000   4.000   3.793   4.000   5.000

summary of quality measures:
    support           confidence          lift
 Min.   :0.00100   Min.   :0.8030   Min.   : 7.521
 1st Qu.:0.00120   1st Qu.:0.8750   1st Qu.:10.253
 Median :0.00140   Median :0.9331   Median :12.165
 Mean   :0.01316   Mean   :0.9319   Mean   :11.859
 3rd Qu.:0.02300   3rd Qu.:1.0000   3rd Qu.:13.478
 Max.   :0.04080   Max.   :1.0000   Max.   :15.625

mining info:
  data ntransactions support confidence
 trans          5000   0.001        0.8
```

This is the summary result after we set the parameters that we mention is **Question.3**, we get 174 set of rules, from the summary result show up we know that there is 53 rules generated with 3 items in the set, 104 rules with 4 items in the set, and 17 rules with 5 items in the set. So after we clear out the infrequent and redundant rules, we get another amount of the rules.

After pruning:

```
> inspect(rules.pruned[1:10])
     lhs                                    rhs                    support confidence lift
[1]  {Blackberry Tart,Single Espresso}  => {Coffee Eclair}        0.0286  0.9108280   8.22047
[2]  {Coffee Eclair,Blackberry Tart}    => {Single Espresso}      0.0286  0.8033708  12.28396
[3]  {Apple Tart,Cherry Soda}           => {Apple Danish}         0.0228  0.8976378  11.47874
[4]  {Apple Danish,Cherry Soda}         => {Apple Tart}           0.0228  0.9120000  12.42507
[5]  {AppleCroissant,Cherry Soda}       => {Apple Danish}         0.0230  0.9126984  11.67134
[6]  {Apple Danish,Cherry Soda}         => {AppleCroissant}       0.0230  0.9200000  12.39892
[7]  {Chocolate Tart,Walnut Cookie}     => {Vanilla Frappuccino} 0.0266  0.9300699  12.67125
[8]  {Lemon Lemonade,Green Tea}         => {Raspberry Cookie}     0.0212  0.9217391  14.40217
[9]  {Raspberry Cookie,Green Tea}       => {Lemon Lemonade}       0.0212  0.9137931  14.10175
[10] {Lemon Lemonade,Green Tea}         => {Lemon Cookie}         0.0214  0.9304348  14.49275
```

These are the 10 rules that generated after we did the pruning.

```
> summary(rules.pruned)
set of 60 rules

rule length distribution (lhs + rhs):sizes
 3  4
37 23

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    3.00    3.00    3.38    4.00    4.00

summary of quality measures:
    support          confidence         lift
 Min.   :0.0010   Min.   :0.803   Min.   : 7.52
 1st Qu.:0.0212   1st Qu.:0.901   1st Qu.:11.38
 Median :0.0228   Median :0.922   Median :12.95
 Mean   :0.0215   Mean   :0.923   Mean   :12.55
 3rd Qu.:0.0271   3rd Qu.:0.950   3rd Qu.:14.16
 Max.   :0.0408   Max.   :1.000   Max.   :15.62

mining info:
  data ntransactions support confidence
 trans          5000   0.001        0.8
```

From the new result of the summary of the pruned rules, we get 60 set of rules. Which we can see that there is 37 rules with 3 items in the set, and 23 rules with 4 items in the set.

Then, we tried to sort the pruned rules by several parameters.

The pruned rules sorted by the maximum lift:
Sorting by maximum lift is to show which itemsets are getting more popular than others. It's because the higher lift value, the better sales of the itemsets. Therefore, we can observe that good selling set of foods in bakery.

```
> rules.pruned.sorted<-sort(rules.pruned, by="lift", decreasing=TRUE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                                 rhs               support confidence lift
[1] {Lemon Lemonade,Raspberry Lemonade,Green Tea}       => {Raspberry Cookie} 0.0212  1.0000000  15.62500
[2] {Lemon Lemonade,Raspberry Lemonade,Green Tea}       => {Lemon Cookie}     0.0212  1.0000000  15.57632
[3] {Raspberry Cookie,Lemon Lemonade,Raspberry Lemonade} => {Lemon Cookie}    0.0262  1.0000000  15.57632
[4] {Lemon Cookie,Lemon Lemonade,Raspberry Lemonade}    => {Raspberry Cookie} 0.0262  0.9924242  15.50663
[5] {Raspberry Cookie,Raspberry Lemonade,Green Tea}     => {Lemon Lemonade}   0.0212  1.0000000  15.43210
```

The pruned rules sorted by the minimum lift:
Sorting by minimum lift is to show which itemsets are getting less popular than others. In this case, we can observe that the combination of food with Coffee Eclair is getting less sales than other combinations.

```
> rules.pruned.sorted<-sort(rules.pruned, by="lift", decreasing=FALSE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                                  rhs             support confidence lift
[1] {Blackberry Tart,Raspberry Lemonade,Single Espresso} => {Coffee Eclair} 0.0010  0.8333333  7.521059
[2] {Napoleon Cake,Blackberry Tart,Single Espresso}      => {Coffee Eclair} 0.0010  0.8333333  7.521059
[3] {Blackberry Tart,Single Espresso}                    => {Coffee Eclair} 0.0286  0.9108280  8.220470
[4] {Apple Pie,Hot Coffee}                               => {Coffee Eclair} 0.0308  0.9166667  8.273165
[5] {Apple Pie,Almond Twist}                             => {Coffee Eclair} 0.0382  0.9695431  8.750389
```

The pruned rules sorted by the maximum support:
Sorting by maximum support is to show the itemsets that more frequent appear that in this dataset than others. From the screenshot below, we understand that itemset that has highest occurrence is Opera Cake, Cherry Tart and Apricot Danish.

```
> rules.pruned.sorted<-sort(rules.pruned, by="support", decreasing=TRUE)
> inspect(rules.pruned.sorted[1:5])
    lhs                               rhs              support confidence lift
[1] {Opera Cake,Cherry Tart}       => {Apricot Danish} 0.0408  0.9357798  10.490805
[2] {Apple Pie,Almond Twist}       => {Coffee Eclair}  0.0382  0.9695431   8.750389
[3] {Coffee Eclair,Apple Pie}      => {Almond Twist}   0.0382  0.9408867  11.558805
[4] {Coffee Eclair,Almond Twist}   => {Apple Pie}      0.0382  0.9271845  11.856579
[5] {Apricot Croissant,Hot Coffee} => {Blueberry Tart} 0.0328  0.9425287  11.062544
```

The pruned rules sorted by the minimum support:
Sorting by minimum support is to show the itemsets that less frequent appear that in this dataset than others. Therefore, from the result below, we found the combinations which carry lesser sales in bakery. That's meaning that we should put more attention to those itemsets, discuss some alternative to improve the sales.

```
> rules.pruned.sorted<-sort(rules.pruned, by="support", decreasing=FALSE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                                  rhs               support confidence lift
[1] {Blackberry Tart,Raspberry Lemonade,Single Espresso} => {Coffee Eclair}   0.001   0.8333333   7.521059
[2] {Coffee Eclair,Raspberry Lemonade,Single Espresso}   => {Blackberry Tart} 0.001   1.0000000  13.157895
[3] {Coffee Eclair,Blackberry Tart,Raspberry Lemonade}   => {Single Espresso} 0.001   0.8333333  12.742100
[4] {Napoleon Cake,Blackberry Tart,Single Espresso}      => {Coffee Eclair}   0.001   0.8333333   7.521059
[5] {Coffee Eclair,Napoleon Cake,Single Espresso}        => {Blackberry Tart} 0.001   1.0000000  13.157895
```

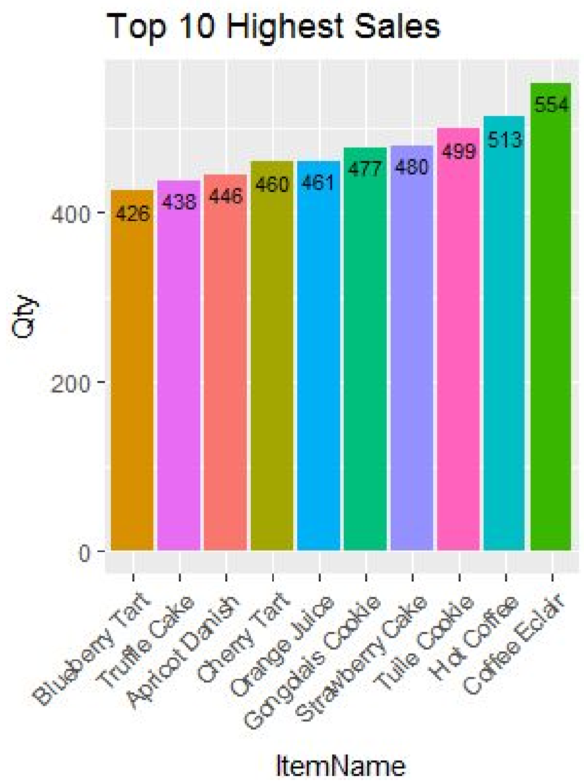
The pruned rules sorted by the maximum confidence:
Sorting by maximum confidence is to show those itemsets that carry higher probability to be purchased together. From one of the itemsets in this result, we can know that if customer buy Coffee Eclair, Raspberry Lemonade and Single Espresso, then he or she will definitely buy Blackberry Tart because it carry 100% confidence.

```
> rules.pruned.sorted<-sort(rules.pruned, by="confidence", decreasing=TRUE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                                rhs                  support confidence lift
[1] {Coffee Eclair,Raspberry Lemonade,Single Espresso} => {Blackberry Tart}    0.0010  1          13.157895
[2] {Coffee Eclair,Napoleon Cake,Single Espresso}      => {Blackberry Tart}    0.0010  1          13.157895
[3] {Blackberry Tart,Almond Twist,Single Espresso}     => {Coffee Eclair}      0.0012  1           9.025271
[4] {Apple Tart,Apple Danish,Cherry Soda}              => {AppleCroissant}     0.0228  1          13.477089
[5] {Raspberry Cookie,Lemon Lemonade,Green Tea}        => {Raspberry Lemonade} 0.0212  1          14.749263
```

The pruned rules sorted by the minimum confidence:
Sorting by maximum confidence is to show the itemsets that carry lower probability to be purchased together. The 5 rules stated below are the rules that the client should consider not to group the items together when packaging the items.

```
> rules.pruned.sorted<-sort(rules.pruned, by="confidence", decreasing=FALSE)
> inspect(rules.pruned.sorted[1:5])
    lhs                                                    rhs               support confidence lift
[1] {Raspberry Cookie,Lemon Cookie,Lemon Lemonade}         => {Green Tea}       0.0212  0.8030303  12.952102
[2] {Lemon Cookie,Lemon Lemonade,Raspberry Lemonade}       => {Green Tea}       0.0212  0.8030303  12.952102
[3] {Coffee Eclair,Blackberry Tart}                        => {Single Espresso} 0.0286  0.8033708  12.283957
[4] {Raspberry Cookie,Lemon Lemonade,Raspberry Lemonade}   => {Green Tea}       0.0212  0.8091603  13.050973
[5] {Blackberry Tart,Raspberry Lemonade,Single Espresso}   => {Coffee Eclair}   0.0010  0.8333333   7.521059
```

What we want to show to client is the itemsets that involves lower sales food in bakery, to find out a way of how to improve the food selling rate. Therefore, we found out the top ten highest sales and top 10 lowest sales foods in bakery.



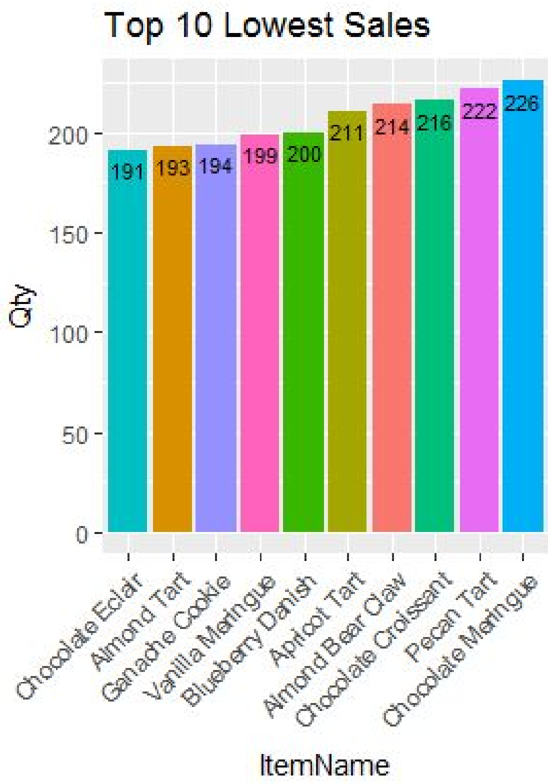


Figure 1: A bar chart that display the top 10 lowest sales in the bakery
Figure 2: A bar chart that display the top 10 highest sales in the bakery

Figure 1 shows the top 10 highest sales item, while Figure 2 shows the top 10 lowest sales based on the dataset. By using these two charts, we can now relate the number of sales with the rules we have discovered and shown at above. The charts also give an insight about the condition of item sales of the bakery shop to the client. With this, he can do some adjustment with the low sales items as recommended in Question 5.

The screenshot below is showing the set of rules with the lowest sales item, Chocolate Eclair at right hand side, to see which items would be purchased together with Chocolate Eclair. In this case, rule no.20 is considered whereby a customer who purchased Tuile Cookies(high sales item) would likely to purchase Chocolate Eclair as well.

```
> inspect(rules.pruned.sorted)
     lhs                        rhs                 support confidence lift
[1]  {Vanilla Eclair}        => {Chocolate Eclair} 0.0032  0.06956522 1.821079
[2]  {Vanilla Meringue}      => {Chocolate Eclair} 0.0024  0.06030151 1.578573
[3]  {Chocolate Croissant}   => {Chocolate Eclair} 0.0024  0.05555556 1.454334
[4]  {Chocolate Meringue}    => {Chocolate Eclair} 0.0024  0.05309735 1.389983
[5]  {Berry Tart}            => {Chocolate Eclair} 0.0040  0.05115090 1.339029
[6]  {AppleCroissant}        => {Chocolate Eclair} 0.0036  0.04851752 1.270092
[7]  {Chocolate Tart}        => {Chocolate Eclair} 0.0036  0.04724409 1.236756
[8]  {Chocolate Coffee}      => {Chocolate Eclair} 0.0038  0.04702970 1.231144
[9]  {Almond Bear Claw}      => {Chocolate Eclair} 0.0020  0.04672897 1.223272
[10] {Marzipan Cookie}       => {Chocolate Eclair} 0.0038  0.04545455 1.189910
[11] {Walnut Cookie}         => {Chocolate Eclair} 0.0032  0.04532578 1.186539
[12] {Lemon Cake}            => {Chocolate Eclair} 0.0038  0.04470588 1.170311
[13] {Napoleon Cake}         => {Chocolate Eclair} 0.0036  0.04400978 1.152088
[14] {Almond Croissant}      => {Chocolate Eclair} 0.0020  0.04385965 1.148158
[15] {Casino Cake}           => {Chocolate Eclair} 0.0032  0.04289544 1.122917
[16] {Strawberry Cake}       => {Chocolate Eclair} 0.0040  0.04166667 1.090750
[17] {Truffle Cake}          => {Chocolate Eclair} 0.0036  0.04109589 1.075809
[18] {Apple Danish}          => {Chocolate Eclair} 0.0032  0.04092072 1.071223
[19] {Apple Tart}            => {Chocolate Eclair} 0.0030  0.04087193 1.069946
[20] {Tuile Cookie}          => {Chocolate Eclair} 0.0040  0.04008016 1.049219
[21] {Cherry Tart}           => {Chocolate Eclair} 0.0036  0.03913043 1.024357
[22] {Orange Juice}          => {Chocolate Eclair} 0.0036  0.03904555 1.022135
[23] {Cherry Soda}           => {Chocolate Eclair} 0.0026  0.03880597 1.015863
[24] {}                      => {Chocolate Eclair} 0.0382  0.03820000 1.000000
```

The screenshot below is showing the set of rules with the lowest sales item, Almond Tart at right hand side, to see which items would be purchased together with Almond Tart. According to the screenshot below, rule no 7 and rule no. 10 are considered.

```
> inspect(rules)
     lhs                                  rhs            support confidence lift
[1]  {Tuile Cookie,Chocolate Coffee}   => {Almond Tart} 0.0012  0.2609     6.758
[2]  {Strawberry Cake,Almond Bear Claw} => {Almond Tart} 0.0010  0.2273     5.888
[3]  {Tuile Cookie,Almond Bear Claw}   => {Almond Tart} 0.0010  0.2273     5.888
[4]  {Almond Bear Claw}                => {Almond Tart} 0.0036  0.0841     2.179
[5]  {Pecan Tart}                      => {Almond Tart} 0.0026  0.0586     1.517
[6]  {Apple Danish}                    => {Almond Tart} 0.0044  0.0563     1.458
[7]  {Tuile Cookie}                    => {Almond Tart} 0.0056  0.0561     1.454
[8]  {Ganache Cookie}                  => {Almond Tart} 0.0020  0.0515     1.335
[9]  {Vanilla Meringue}                => {Almond Tart} 0.0020  0.0503     1.302
[10] {Orange Juice}                    => {Almond Tart} 0.0046  0.0499     1.293
[11] {Chocolate Meringue}              => {Almond Tart} 0.0022  0.0487     1.261
[12] {Almond Croissant}                => {Almond Tart} 0.0022  0.0482     1.250
[13] {Vanilla Eclair}                  => {Almond Tart} 0.0022  0.0478     1.239
[14] {Lemon Cake}                      => {Almond Tart} 0.0038  0.0447     1.158
[15] {Blackberry Tart}                 => {Almond Tart} 0.0032  0.0421     1.091
[16] {Lemon Cookie}                    => {Almond Tart} 0.0026  0.0405     1.049
[17] {Lemon Lemonade}                  => {Almond Tart} 0.0026  0.0401     1.039
[18] {Blueberry Danish}                => {Almond Tart} 0.0016  0.0400     1.036
[19] {Bottled Water}                   => {Almond Tart} 0.0030  0.0389     1.007
[20] {}                                => {Almond Tart} 0.0386  0.0386     1.000
```

5. Recommendations: What should the client do because of the rules discovered?

- Reallocate the low sales items at the counter
    - In order to increase client's bakery sales, we can recommend the client to reallocate the lower sales items or food near or at the counter to attract customers' attention to the selected items. For example, by referring to Figure

2, Item No.6 (Chocolate Eclair) has the lowest sales among the 50 foods, so we can encourage them to put it near to the counter, and staff can suggest customer to purchase Chocolate Eclair when customer approaches to counter.

- Below are the low sales items that can recommend to customer when they are making payment at the counter:
    - Chocolate Eclair
    - Almond Tart
    - Ganache Cookie
    - Vanilla Meringue
    - Blueberry Danish
    - Apricot Tart
    - Almond Bear Claw
    - Chocolate Croissant
    - Pecan Tart
    - Chocolate Meringue

- Reallocate the low sales items around high sales item
    - Reallocating related low sales items with high sales item can also attract customers' attention to those low sales items, increasing the chance of the items to be purchased. For example, Item No.10 (one of lowest sales food) and Item No.28 (one of highest sales food), which are Almond Tart and Tuile Cookie, is one of the itemsets in association rules mining, we can suggest client to reallocate Almond Tart around Tuile Cookie, so that there are chances for the customers to buy Almond Tart along when they buy Tuile Cookie.
    - The items that can be relocate: **(format: low sales item -> around good sales item)**
        - Chocolate Eclair -> Blueberry Tart
        - Almond Tart -> Tuile Cookie
        - Ganache Cookie -> Truffle Cake
        - Vanilla Meringue -> Apricot Danish
        - Blueberry Danish -> Cherry Tart
- Held promotion for low sales items
    - The sales can be increased by offering discounts and promotions on low sales items. For instance, promoting Chocolate Eclair to customers by setting a discount price for it. We can also suggest client to set different low sales items as promotional items for different day. For example, customers can get discount on Chocolate Eclair on Mondays, Almond Tart on Tuesday, and so on. This allows low sales items to be introduced to customer in a more efficient way.
- Bundle low sales items with high sales item
    - Introducing the low sales items in a bundle of high sales items while offering discount on whole bundle is also a good way to increase the sales. As bundle is going to sell because by purchasing the bundle, the low sales items can be sold together. For example, one of the highest sales item no. 45, Hot Coffee can be sold together with one if the lowest sales item no.26, Vanilla Meringue

with an offer price. By doing so, Vanilla Meringue could get a chance to gain popularity.

- For bundle, we recommend client to set up a tea time package whereby the package items included blackberry tart, raspberry lemonade, single espresso and coffee eclair. From the rules we can see that although these are the low sales item, but these item are always come together, so we recommend client to set up this package to push the sales of other related item to this package.