Superstor_Market_Basket_Analysis

JinpingBai

10/11/2020

Market Basket Analysis and Apriori

We will use the Market Basket Analysis and The Apriori Alforithm to see what kind of products that cutomers usually purchase together.

```
## [1] 51290    24
```

We will use the Association Rule Mining method to find frequent patterns in the transaction of 2011 to 2015. By knowing what items that customers frequently buy together, it will generate a set of rules. Store owner will use those rules for many marketing strategies:

- Change the store layout according to trends
- Customer behavior analysis
- Catalogue design
- Cross marketing on online stores
- Customed emails with add-on sales
- Consumer items-buys trending

First, let check how many unique Order.ID in the whole dataset. Order.ID was assigned to each transaction.

```
## [1] 25035
```

There are tatal 25035 unique Order.ID. That means there might be more than 1 times of product in each transaction. We will find out the patten what kind of products that cutomers usually purchase together.So that the store can rearrange the outlays of products, either put them together to maxmize the sales of relevant products or put them far apart so that customer have chance to see other products.

Then, we will check how many knids of products in total has been sold in the 4 years range.

```
## [1] 3788
```

There are 3788 kinds of products names. Remove "," in the column of products description for creating a transaction data.

```
## [1] "Tenex Lockers, Blue"
## [2] "Acme Trimmer, High Speed"
## [3] "Tenex Box, Single Width"
## [4] "Enermax Note Cards, Premium"
## [5] "Eldon Light Bulb, Duo Pack"
## [6] "Eaton Computer Printout Paper, 8.5 x 11"
```

Apply Market Basket Analysis Method

We use the Apriori “transaction” function to create a sparse matrix representing the all transactions and producst name has been sold. Other attrubute will not appear to the sparse matrix.

```
## Warning in asMethod(object): removing duplicated items in transactions
```

Let’s have a general look and the transactions information.

```
## transactions as itemMatrix in sparse format with
##  25035 rows (elements/itemsets/transactions) and
##  3788 columns (items) and a density of 0.000540405
##
## most frequent items:
##                        Staples      Cardinal Index Tab  Clear
##                            222                             92
##  Eldon File Cart  Single Width Rogers File Cart  Single Width
##                             90                             84
##         Ibico Index Tab  Clear                        (Other)
##                             83                          50677
##
## element (itemset/transaction) length distribution:
## sizes
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 12264  6218  3214  1627   806   443   236    97    64    39    15     6     5
##    14
##     1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.047   3.000  14.000
##
## includes extended item information - examples:
##                                                  labels
## 1 "While you Were Out" Message Book  One Form per Page
## 2             #10 Gummed Flap White Envelopes  100/Box
## 3                        #10 Self-Seal White Envelopes
##
## includes extended transaction information - examples:
##   transactionID
## 1  AE-2011-9160
## 2  AE-2013-1130
## 3  AE-2013-1530
```

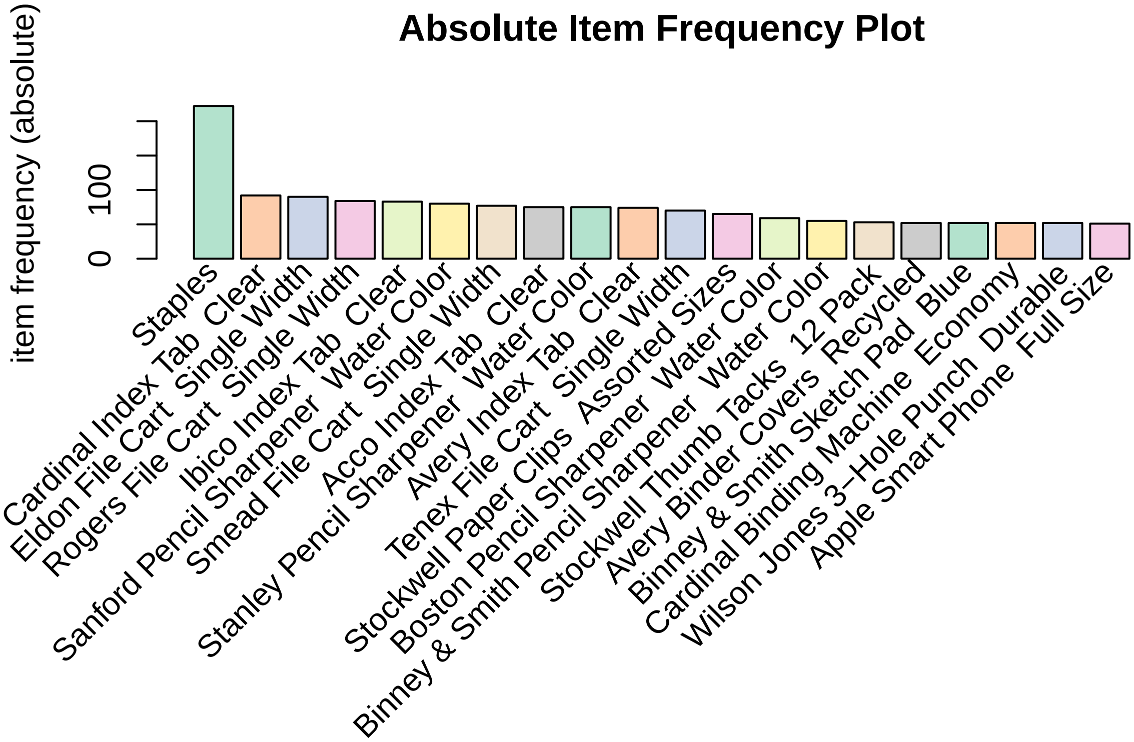
There are 25035 transactions (row) and 3788 items (columns). The rows of transactions reflect the total number of Order.ID; the columns of items reflect the total Products.Name in the raw dataset.

Density is 0.000540405. Density tells us the percentage of non-zero cells in a sparse matrix. We can calculate how many items were purchased by using the density.

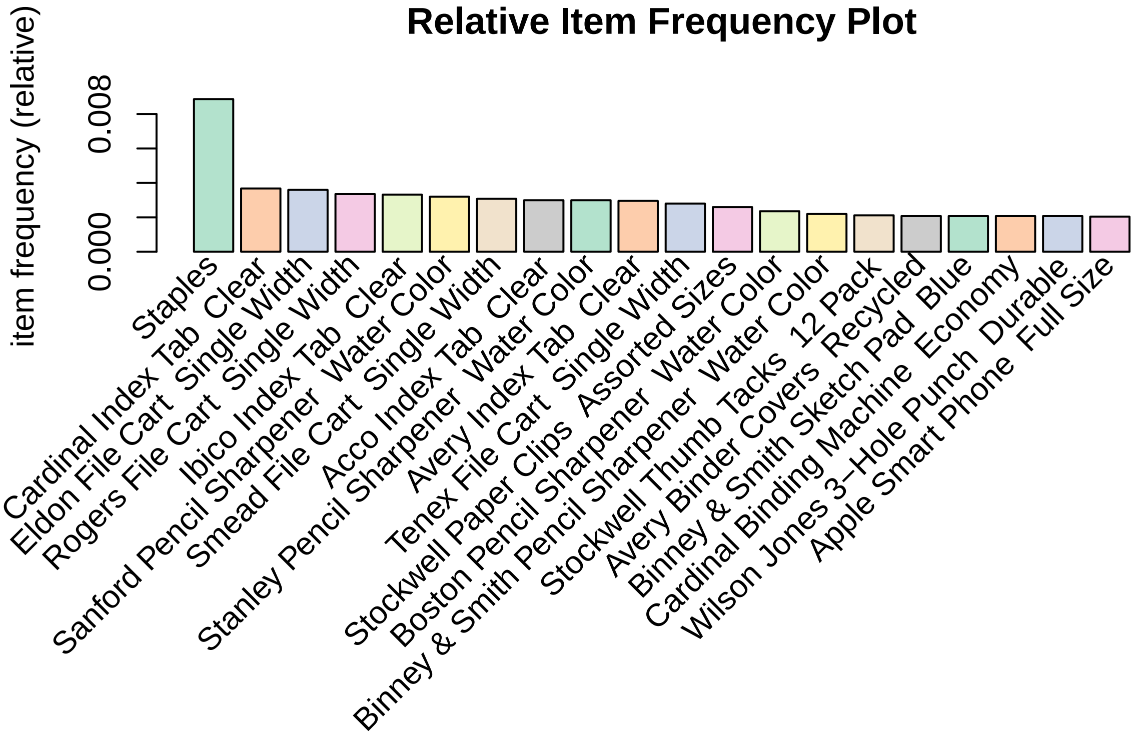
```
## [1] 51248
```

Visualization of the MBA.

Using“itemFrwquencyPlot” to visualize the distribution of frequency of items that purchase. It can be evaluate base on absolute numbers of relative propotion.



Absolute will plot numeric frequencies of each item independently.



Relative plot will show how many times these items have appeared as compard to others.

Generating Rules

Using Apriori algorithm to generate association rules.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   7e-05      1
##  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: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3788 item(s), 25035 transaction(s)] done [0.02s].
## sorting and recoding items ... [3690 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.08s].
## writing ... [1625 rule(s)] done [0.03s].
## creating S4 object  ... done [0.01s].
```

There are total 1625 rules generated. Let's select the top 10 rules to have a look.

```
##      lhs                                            rhs
## [1]  {Ativa V4110MDD Micro-Cut Shredder}         => {Staples}                                   7.988
## [2]  {Bevis Conference Table  Fully Assembled,
##       Hon Conference Table  with Bottom Storage} => {Sharp Personal Copier  Laser}              7.988
## [3]  {Hon Conference Table  with Bottom Storage,
##       Sharp Personal Copier  Laser}              => {Bevis Conference Table  Fully Assembled}   7.988
## [4]  {Bevis Conference Table  Fully Assembled,
##       Sharp Personal Copier  Laser}              => {Hon Conference Table  with Bottom Storage} 7.988
## [5]  {Bevis Conference Table  Fully Assembled,
##       Hon Conference Table  with Bottom Storage} => {Jiffy Mailers  Set of 50}                  7.988
## [6]  {Hon Conference Table  with Bottom Storage,
##       Jiffy Mailers  Set of 50}                  => {Bevis Conference Table  Fully Assembled}   7.988
## [7]  {Bevis Conference Table  Fully Assembled,
##       Jiffy Mailers  Set of 50}                  => {Hon Conference Table  with Bottom Storage} 7.988
## [8]  {Bevis Conference Table  Fully Assembled,
##       Hon Conference Table  with Bottom Storage} => {Motorola Audio Dock  with Caller ID}       7.988
## [9]  {Hon Conference Table  with Bottom Storage,
##       Motorola Audio Dock  with Caller ID}       => {Bevis Conference Table  Fully Assembled}   7.988
## [10] {Bevis Conference Table  Fully Assembled,
##       Motorola Audio Dock  with Caller ID}       => {Hon Conference Table  with Bottom Storage} 7.988
```

Explaine the rules: For example the top one rule explains that 79.88% transaction show "Ativa V4110MDD Micro-Cut Shredder" is bought with purchase of "Staples"; 100% of customers who purchase "Ativa V4110MDD Micro-cut Shredder" also bought "Staples".

Removing redundant rules

```
## [1] 1622
```

There are 1622 are subset rules. There are only 3 main rules.

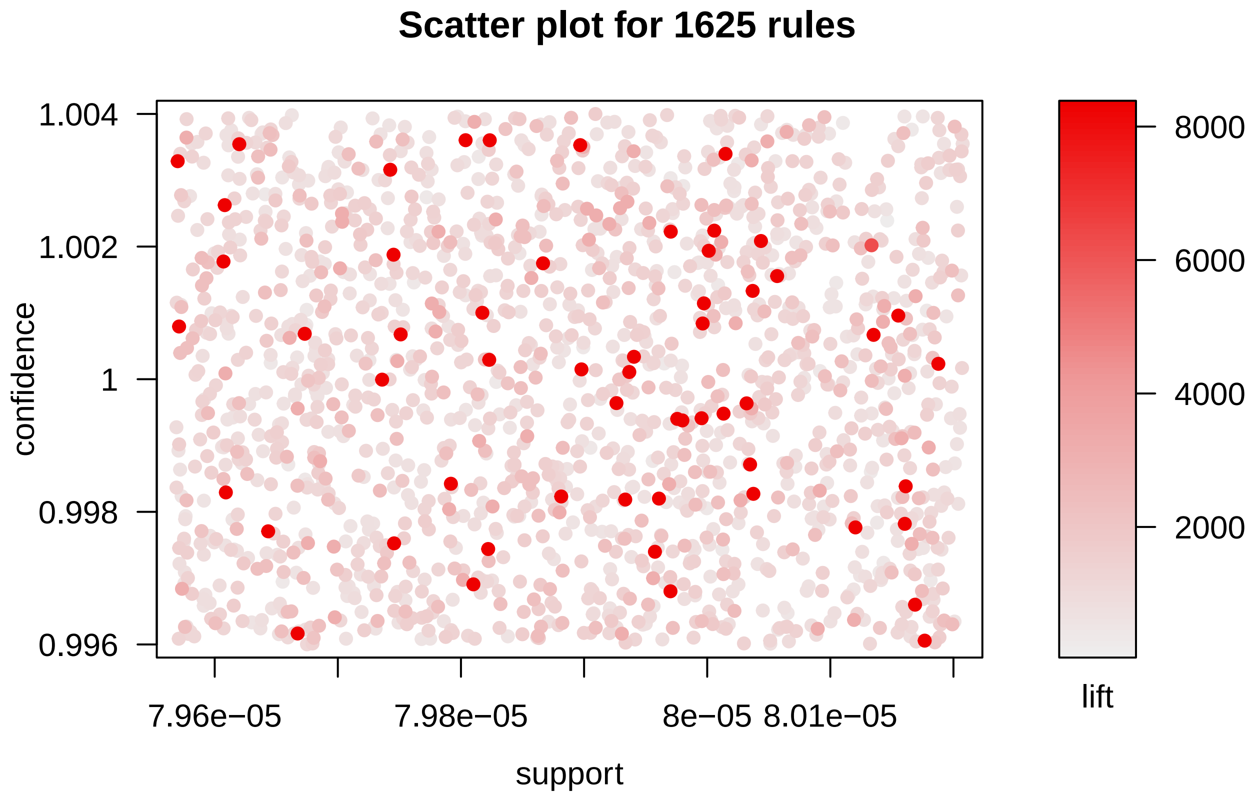
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   5e-05      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[3788 item(s), 25035 transaction(s)] done [0.02s].
## sorting and recoding items ... [3690 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.08s].
## writing ... [1 rule(s)] done [0.04s].
## creating S4 object  ... done [0.01s].
```

Check rules by giveing a item, for example “staples.rules”

```
##     lhs                                     rhs       support      confidence
## [1] {Ativa V4110MDD Micro-Cut Shredder} => {Staples} 7.988816e-05 1
##     coverage     lift     count
## [1] 7.988816e-05 112.7703 2
```

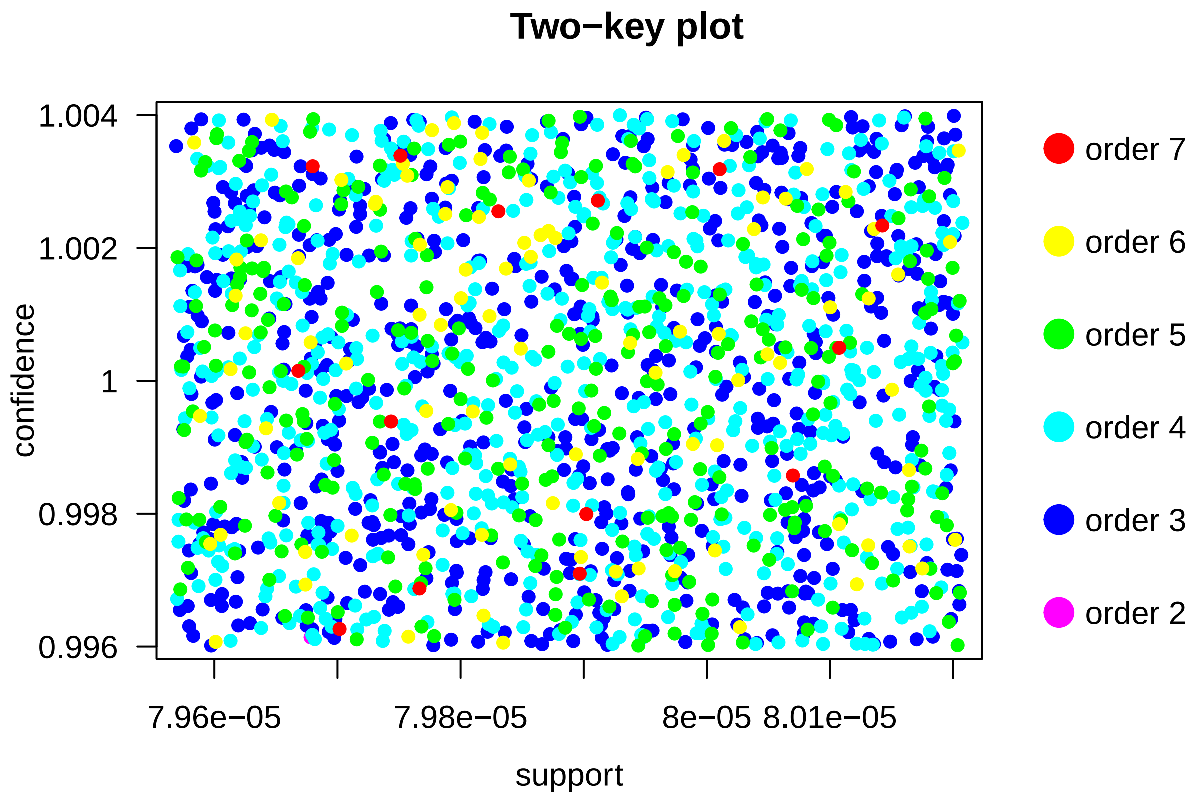
Visualization of Association Rules

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



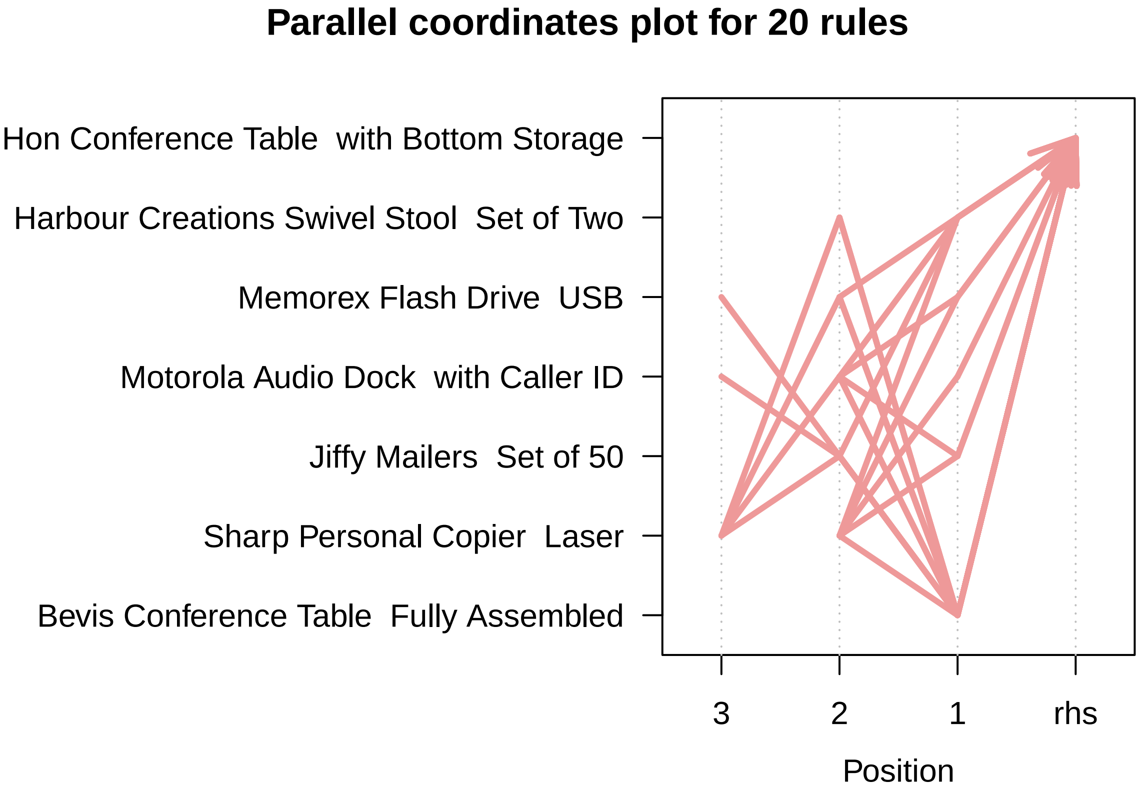
The above plot shows that rules with higher lift have a little less support. But in general it was evenly spred.

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Above two-key plot shows support and confidence respectively. The order shows how many items in the relevant rule.

Use Parallel Coordinates Plot to chech individual rule representation. The plot will explaine which products along with with items cause what kind of sales.



Above plot shows that If customer bought “Memorex Flash Drive USB” and “Motorola Audio Dock with Caller ID”, the customer likely to buy“Jiffy Mailers Set of 50”.