BlackFriday Analysis

Nisha Selvarajan

11/1/2020

BlackFriday - Customer Purchasing Behavior

Market Basket Analysis /APRIORI - Black Friday Examined

With the holiday season fast approaching, I found it intriguing to examine a dataset revolving around a hypothetical store and data of its shoppers.Ability to recognize and track patterns in data help businesses shift through the layers of seemingly unrelated data for meaningful relationships. Through this analysis it becomes easy for the online retailers to determine the dimensions that influence the uptake of online shopping and plan effective marketing strategies. This project builds a roadmap for analyzing consumer's online buying behavior with the help of Apriori algorithm.

Your client gives you data for all transactions that consists of items bought in the store by several customers over a period of time and asks you to use that data to help boost their business. Your client will use your findings to not only change/update/add items in inventory but also use them to change the layout of the physical store or rather an online store. To find results that will help your client, you will use Market Basket Analysis (MBA) which uses Association Rule Mining on the given transaction data.

Association Rule Mining

- Association Rule Mining is used when you want to find an association between different objects in a set, find frequent patterns in a transaction database, relational databases or any other information repository. The applications of Association Rule Mining are found in Marketing, Basket Data Analysis (or Market Basket Analysis) in retailing, clustering and classification. It can tell you what items do customers frequently buy together by generating a set of rules called Association Rules. In simple words, it gives you output as rules in form if this then that. Clients can use those rules for numerous marketing strategies:
  - Changing the store layout according to trends
  - Customer behavior analysis -Catalogue design -Cross marketing on online stores -What are the trending items customers buy -Customized emails with add-on sales
- Association Rule Mining is viewed as a two-step approach:

  -Frequent Itemset Generation: Find all frequent item-sets with support >= pre-determined min_support count. Frequent Itemset Generation is the most computationally expensive step because it requires a full database scan.

  -Rule Generation: List all Association Rules from frequent item-sets. Calculate Support and Confidence for all rules. Prune rules that fail min_support and min_confidence thresholds.

Challenge

- Find hidden relationships between the products ,and to analyze purchase behaviors using APRIORI.
- Look for combinations of items that occur together frequently in transactions, providing information to understand the purchase behavior. The outcome of this type of technique is, in simple terms, a set of rules that can be understood as "if this, then that"

Data Description

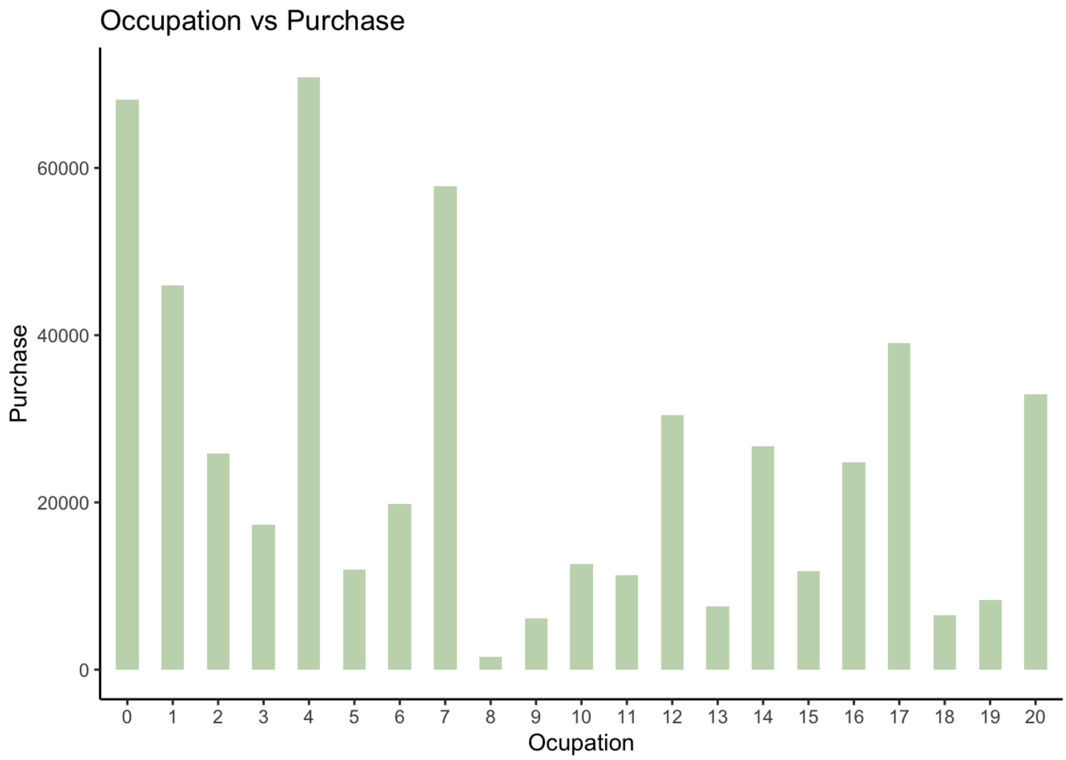
- The data used for this particular project is "Black Friday Sales Analysis"(https://www.kaggle.com/mehdidag/black-friday). Detailed description of the variables:

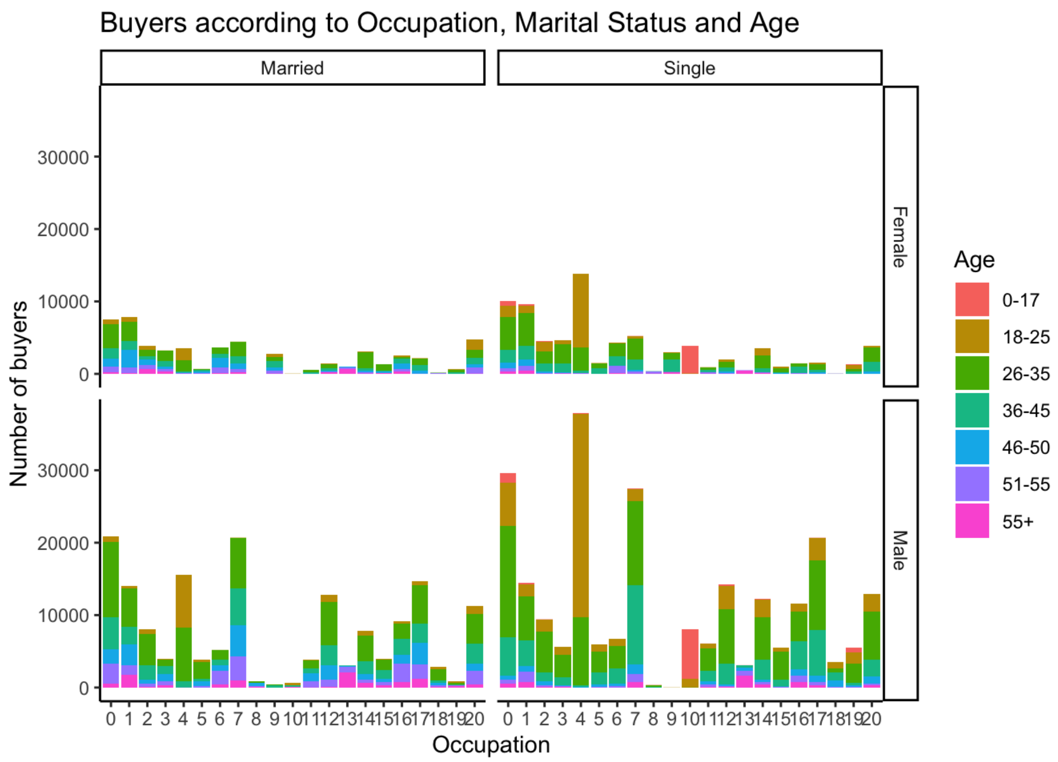
Names	Description
User_ID	Categorical - User ID
Product_ID	Categorical - Product ID
Gender	Categorical - Sex of User
Age	Categorical - Age in bins
Occupation	Categorical - Occupation (Masked)
City_Category	Categorical - Category of the City (A,B,C)
Stay_In_Current_City_Years	Numerical - Number of years stay in current city
Marital_Status	Categorical - Marital Status
Product_Category_1	Categorical - Product Category (Masked)
Product_Category_2	Categorical - Product may belongs to other category also (Masked)
Product_Category_3	Categorical - Product may belong to other category also (Masked)
Purchase	Numerical - Purchase Amount (Target Variable)

Data Analysis & Clean up

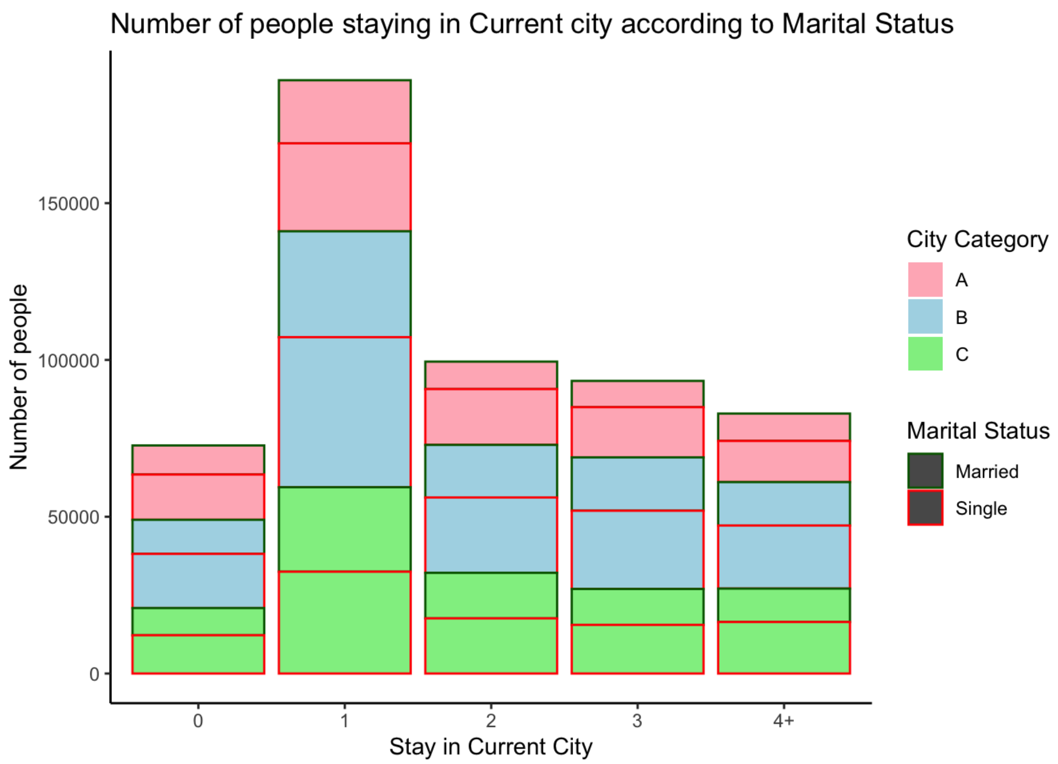
- Black Friday data set is further cleaned by changing the format of each variable. This included changing Product_ID, Gender, Age, City_Category, Marital_Status and Product_Category from character variables to factors.
- Product Category 2 & Product Category 3 has many missing values. Input 0 for Product Category 2/ Product Category 3.

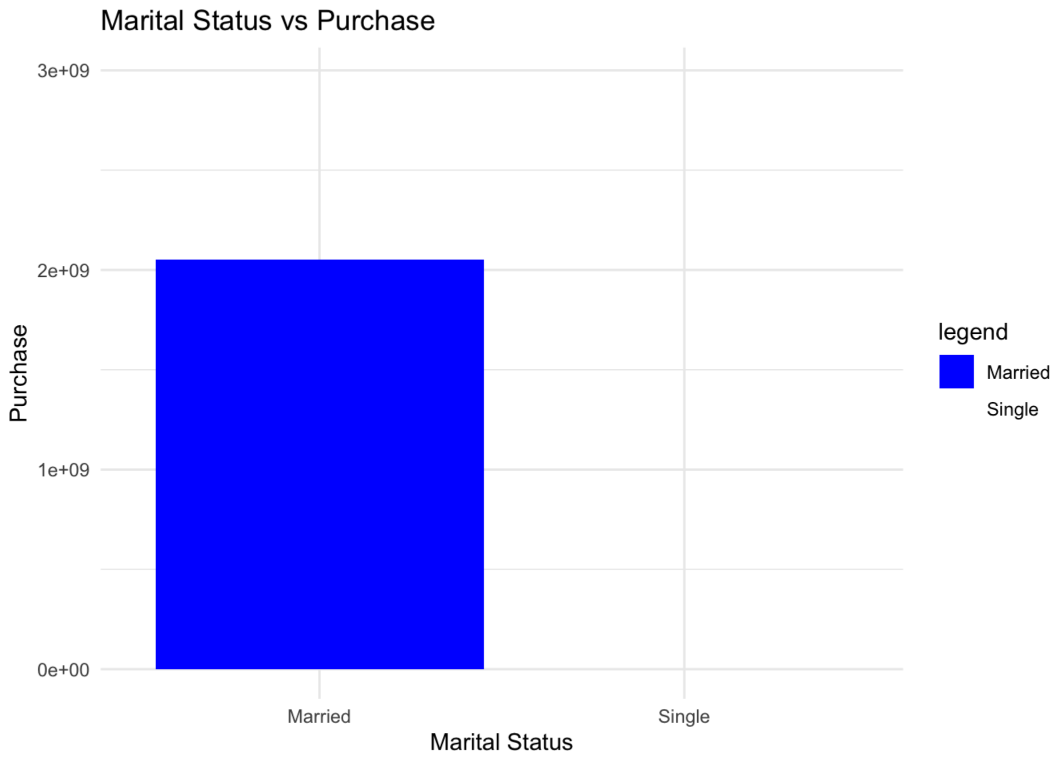
Exploratory Data Analysis


Occupation vs Purchase
Purchase
60000
40000
20000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Ocupation

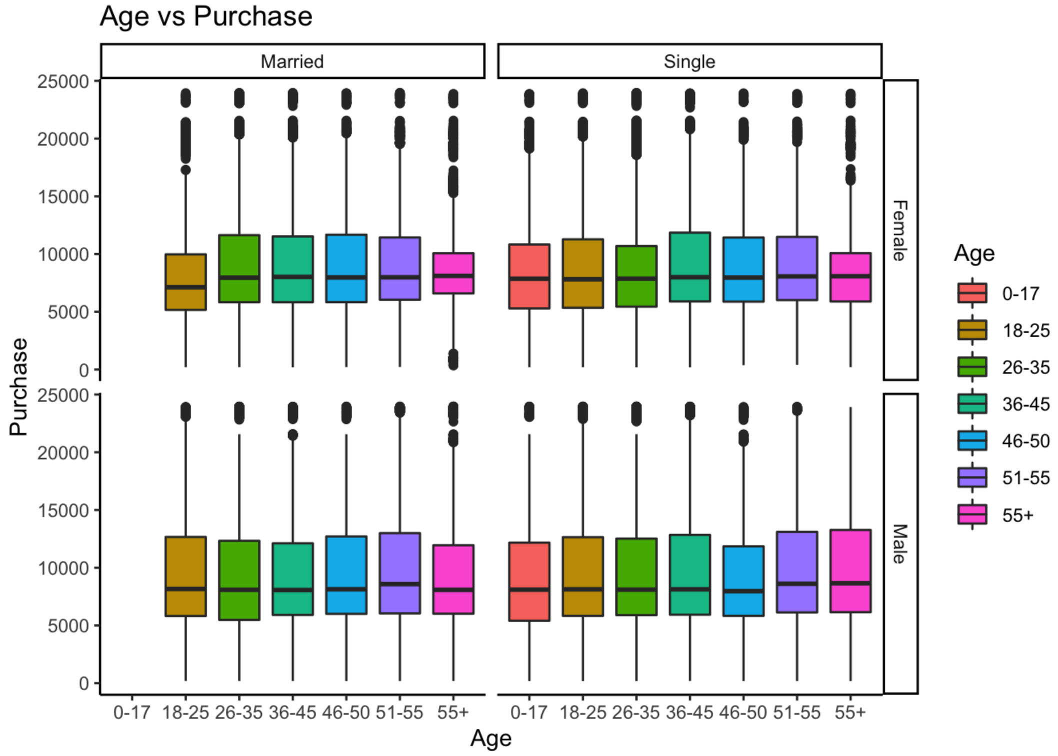

Buyers according to Occupation, Marital Status and Age
Married
Single
Female
Male
Number of buyers
30000
20000
10000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Occupation
Age
0-17
18-25
26-35
36-45
46-50
51-55
55+

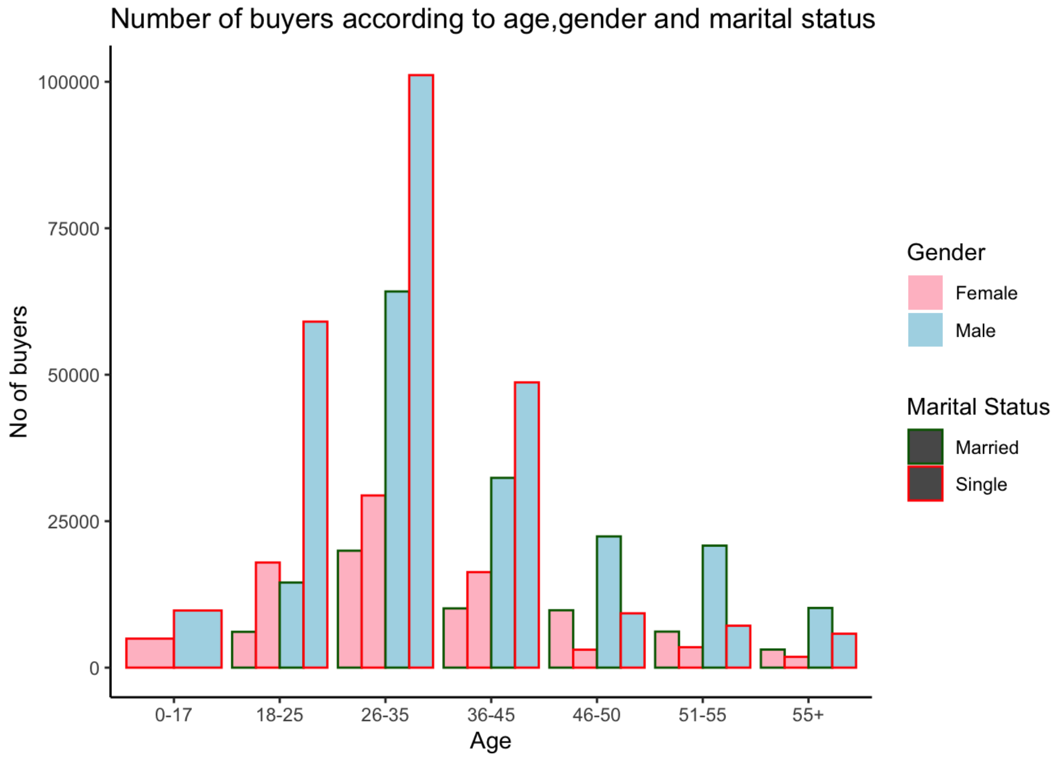

City Category vs Purchase
Purchase
2.0e+09
1.5e+09
1.0e+09
5.0e+08
0.0e+00
A
B
C
City Category
legend
A
B
C

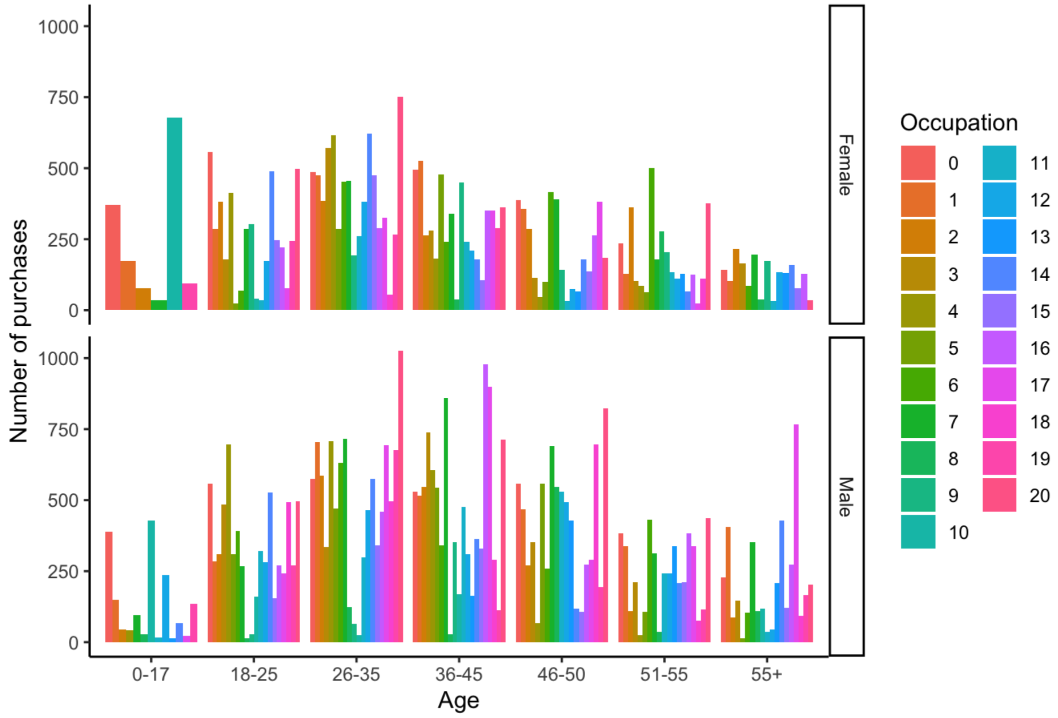

Number of people staying in Current city according to Marital Status
Number of people
150000
100000
50000
0
0
1
2
3
4+
Stay in Current City
City Category
A
B
C
Marital Status
Married
Single

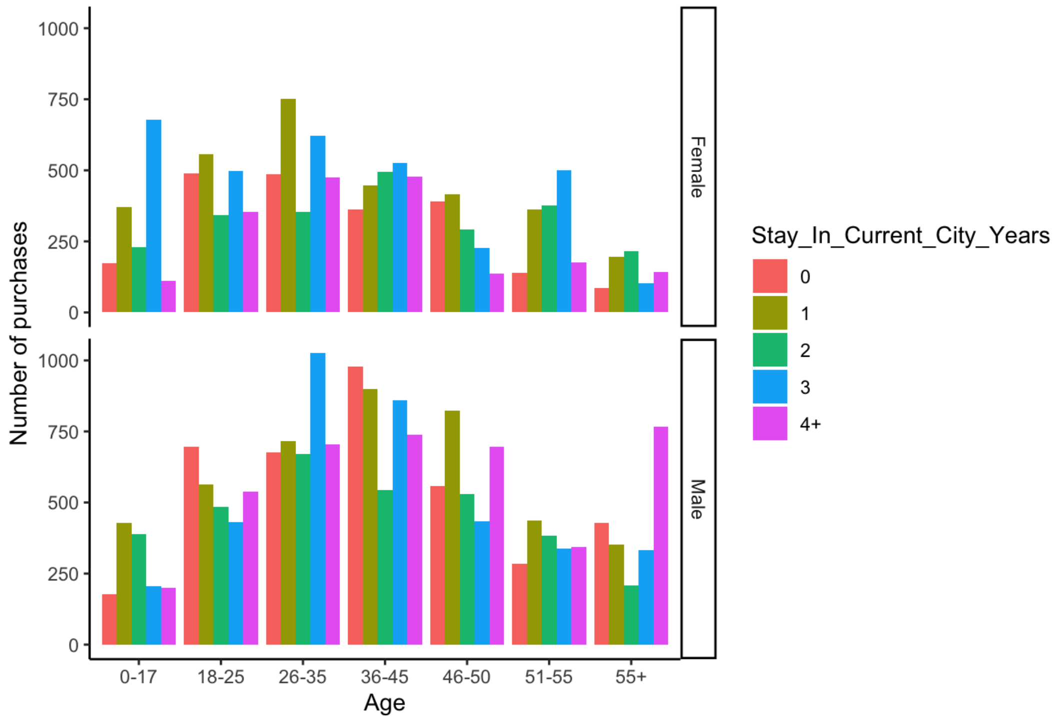

Marital Status vs Purchase
3e+09
2e+09
1e+09
0e+00
Purchase
legend
Married
Single
Married
Single
Marital Status


Stay in current city vs Purchase
150000
100000
50000
0
Purchase
0
1
2
3
4+
Stay in city

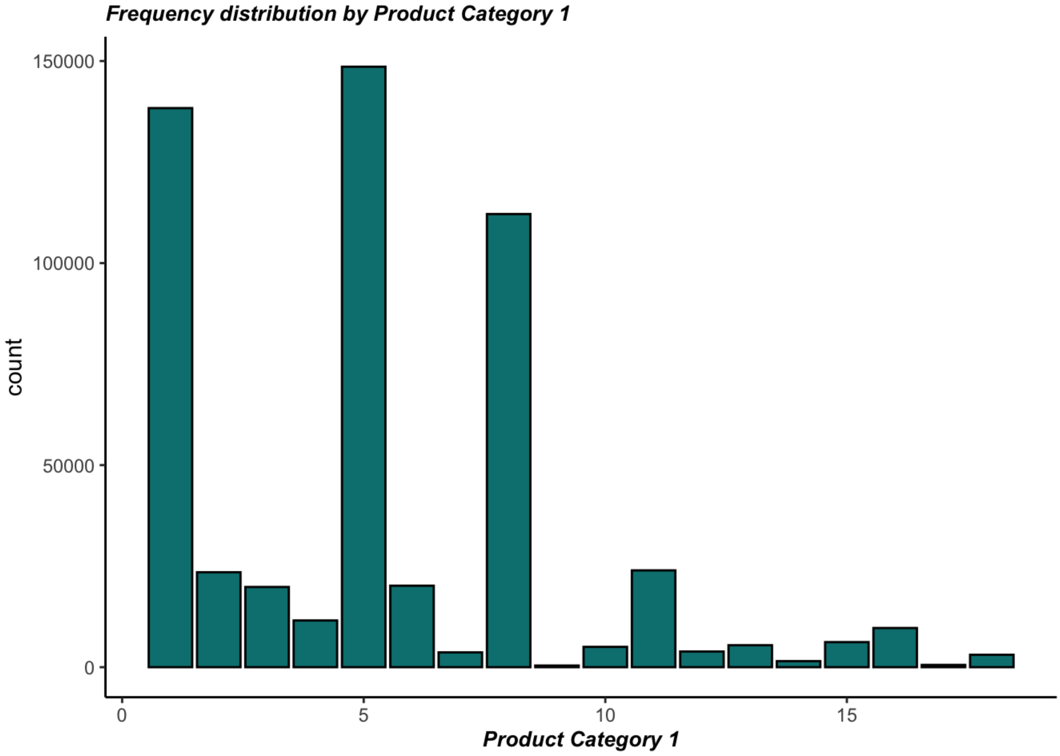

Age vs Purchase
Married
Single
Female
Male
Purchase
25000
20000
15000
10000
5000
0
0-17 18-25 26-35 36-45 46-50 51-55 55+
Age
Age
0-17
18-25
26-35
36-45
46-50
51-55
55+

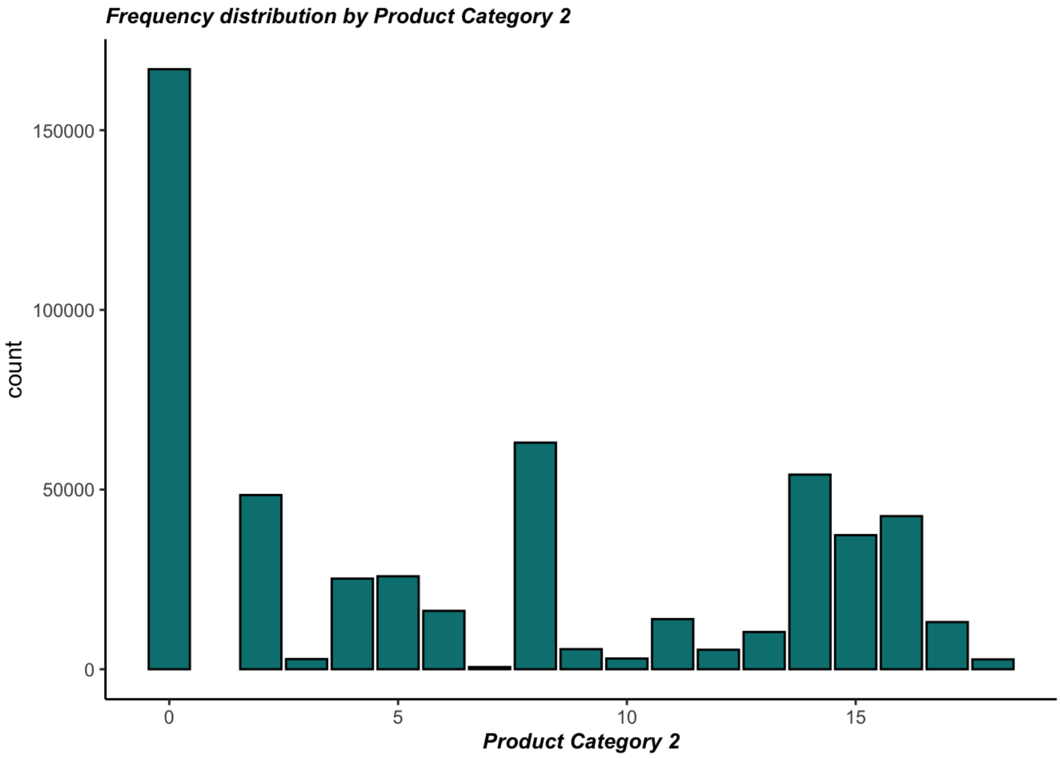

Number of buyers according to age,gender and marital status
No of buyers
100000
75000
50000
25000
0
0-17
18-25
26-35
36-45
46-50
51-55
55+
Age
Gender
Female
Male
Marital Status
Married
Single

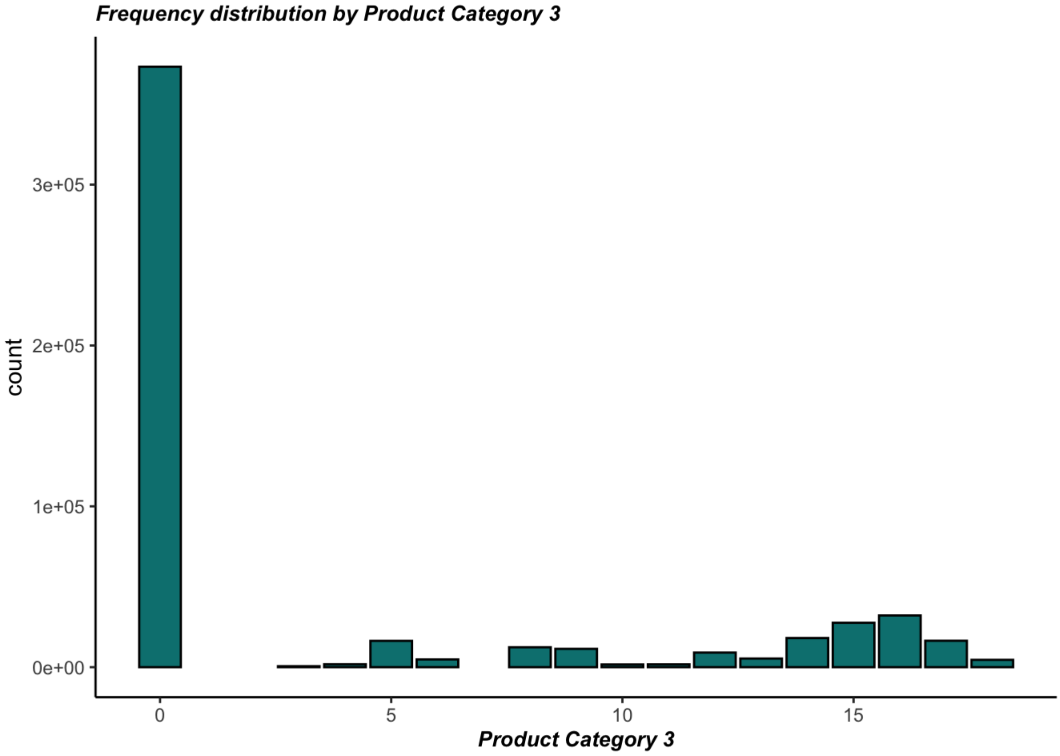




Frequency Distribution By Product Category


Frequency distribution by Product Category 1
count
150000
100000
50000
0
0
5
10
15
Product Category 1


Frequency distribution by Product Category 2
count
150000
100000
50000
0
0
5
10
15
Product Category 2



Impleentation of APRIORI

- Step 1: Load the dataset & Clean the dataset.

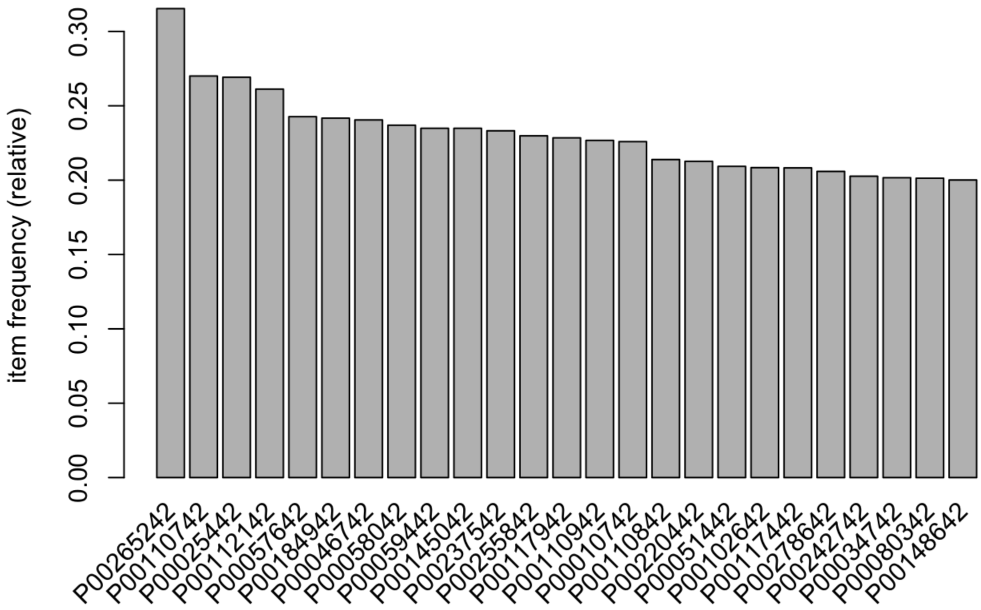
```
transactions as itemMatrix in sparse format with
 537578 rows (elements/itemsets/transactions) and
 537578 columns (items) and a density of 1.860195e-06

most frequent items:
 1000001,P00000142,F,0-17,10,A,2,0,3,4,5,13650
                                             1
1000001,P00004842,F,0-17,10,A,2,0,3,4,12,13645
                                             1
 1000001,P00025442,F,0-17,10,A,2,0,1,2,9,15416
                                             1
  1000001,P00051442,F,0-17,10,A,2,0,8,17,,9938
                                             1
   1000001,P00051842,F,0-17,10,A,2,0,4,8,,2849
                                             1
                                       (Other)
                                        537573

element (itemset/transaction) length distribution:
sizes
     1
537578

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       1       1       1       1       1

includes extended item information - examples:
                                          labels
1   1000001,P00000142,F,0-17,10,A,2,0,3,4,5,13650
2  1000001,P00004842,F,0-17,10,A,2,0,3,4,12,13645
3   1000001,P00025442,F,0-17,10,A,2,0,1,2,9,15416
```

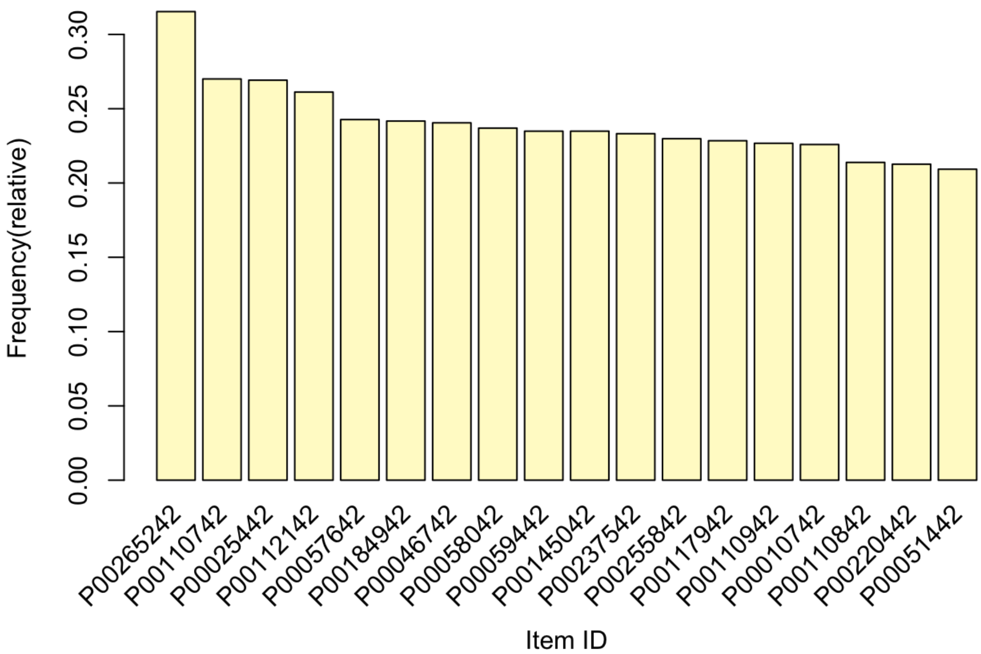
- Step 2: Data cleaning and manipulations using R.
  - Group the transactions by USER ID. The data required for Apriori must be in the basket format.The basket format must have first column as a unique identifier of each transaction, something like a unique product Id. The second

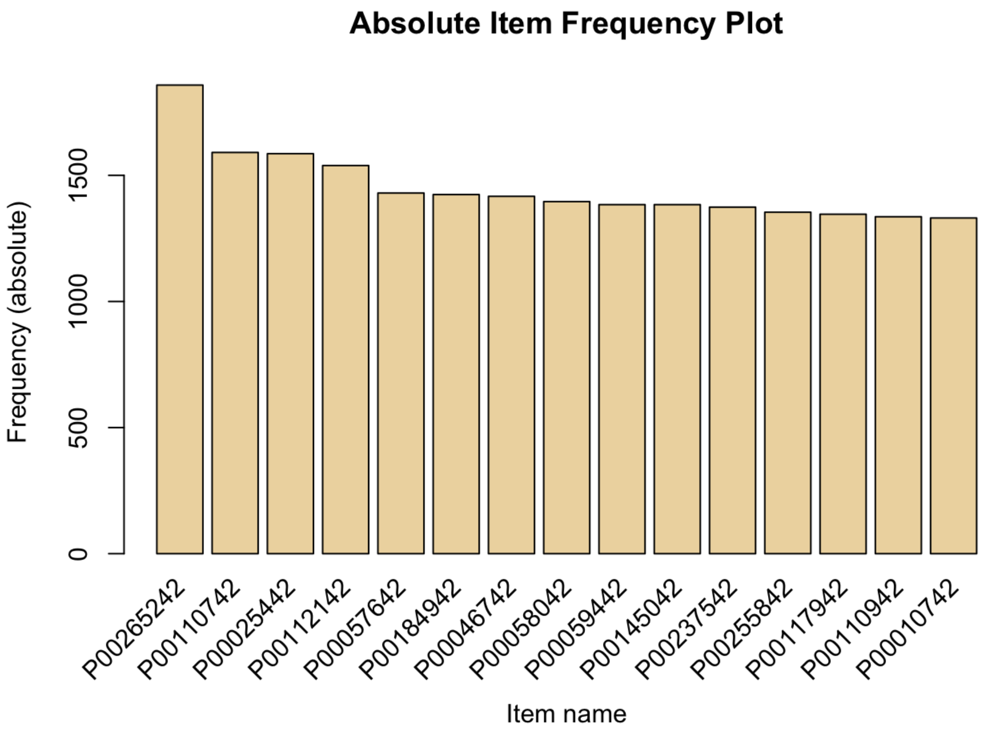
columns consists of the items bought in that transaction, separated by spaces or commas or some other separator.

- APRIORI needs the data in transaction format. Convert grouped customer Id data frame to transaction.
- read.transactions in R reads a transaction data file from disk and creates a transactions object.

Item Frequency Plot







- Step 3: Find the association rules.
  - Next step is to mine the rules using the APRIORI algorithm. The function apriori() is from package arules.
  - Association rules analysis is a technique to uncover how items are associated to each other. There are three common ways to measure association.
  - Measure 1: Support. This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears.
  - Measure 2: Confidence. This says how likely item Y is purchased when item X is purchased, expressed as {X -> Y}. This is measured by the proportion of transactions with item X, in which item Y also appears.
  - Measure 3: Lift. This says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is.
  - Measure 4:minlen is the minimum number of items required in the rule.
  - Measure 5:maxlen is the maximum number of items that can be present in the rule.

```
summary(itemFrequency(customers_products))
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
0.0001697 0.0001697 0.0001697 0.0087686 0.0037339 0.3153428 
```

```
rules = apriori(data = customers_products, parameter = list(support = 
                                                  0.01, confidence = 0.74, minlen = 4))
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
       0.74    0.1    1 none FALSE            TRUE       5    0.01      4
 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: 58

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10539 item(s), 5892 transaction(s)] done [0.17s].
sorting and recoding items ... [1958 item(s)] done [0.02s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [8.04s].
writing ... [10 rule(s)] done [0.07s].
creating S4 object  ... done [0.17s].
```

- Step 5: Print the association rules. To print the association rules, we use a function called inspect().

```
inspect(rules[1:10])
```

```
     lhs                                  rhs         support    confidence
[1]  {P00057642,P00105142,P00127342} => {P00025442} 0.01154107 0.7640449
[2]  {P00025442,P00034042,P00112442} => {P00110742} 0.01001358 0.7468354
[3]  {P00034042,P00057942,P00112542} => {P00110742} 0.01052274 0.7469880
[4]  {P00034042,P00111142,P00112542} => {P00110742} 0.01052274 0.7469880
[5]  {P00003242,P00111142,P00127842} => {P00145042} 0.01120163 0.7857143
[6]  {P00057942,P00105142,P00182242} => {P00110742} 0.01086219 0.7529412
[7]  {P00111742,P00295942,P00323942} => {P00052842} 0.01052274 0.7469880
[8]  {P00128942,P00144642,P00329542} => {P00057642} 0.01001358 0.7763158
[9]  {P00070042,P00117942,P00277642} => {P00145042} 0.01137135 0.7613636
[10] {P00000142,P00086442,P00140742} => {P00145042} 0.01018330 0.7407407
     coverage   lift     count
[1]  0.01510523 2.838432 68
[2]  0.01340801 2.765779 59
[3]  0.01408690 2.766344 62
[4]  0.01408690 2.766344 62
[5]  0.01425662 3.344963 66
[6]  0.01442634 2.788391 64
[7]  0.01408690 4.546749 62
[8]  0.01289885 3.198638 59
[9]  0.01493551 3.241297 67
[10] 0.01374745 3.153500 60
```

- Sort by Confidence

```
inspect(sort(rules, by = 'confidence'))
```

```
     lhs                                 rhs         support    confidence
[1]  {P00003242,P00111142,P00127842} => {P00145042} 0.01120163 0.7857143
[2]  {P00128942,P00144642,P00329542} => {P00057642} 0.01001358 0.7763158
[3]  {P00057642,P00105142,P00127342} => {P00025442} 0.01154107 0.7640449
[4]  {P00070042,P00117942,P00277642} => {P00145042} 0.01137135 0.7613636
[5]  {P00057942,P00105142,P00182242} => {P00110742} 0.01086219 0.7529412
[6]  {P00034042,P00057942,P00112542} => {P00110742} 0.01052274 0.7469880
[7]  {P00034042,P00111142,P00112542} => {P00110742} 0.01052274 0.7469880
[8]  {P00111742,P00295942,P00323942} => {P00052842} 0.01052274 0.7469880
[9]  {P00025442,P00034042,P00112442} => {P00110742} 0.01001358 0.7468354
[10] {P00000142,P00086442,P00140742} => {P00145042} 0.01018330 0.7407407
     coverage   lift     count
[1]  0.01425662 3.344963 66
[2]  0.01289885 3.198638 59
[3]  0.01510523 2.838432 68
[4]  0.01493551 3.241297 67
[5]  0.01442634 2.788391 64
[6]  0.01408690 2.766344 62
[7]  0.01408690 2.766344 62
[8]  0.01408690 4.546749 62
[9]  0.01340801 2.765779 59
[10] 0.01374745 3.153500 60
```

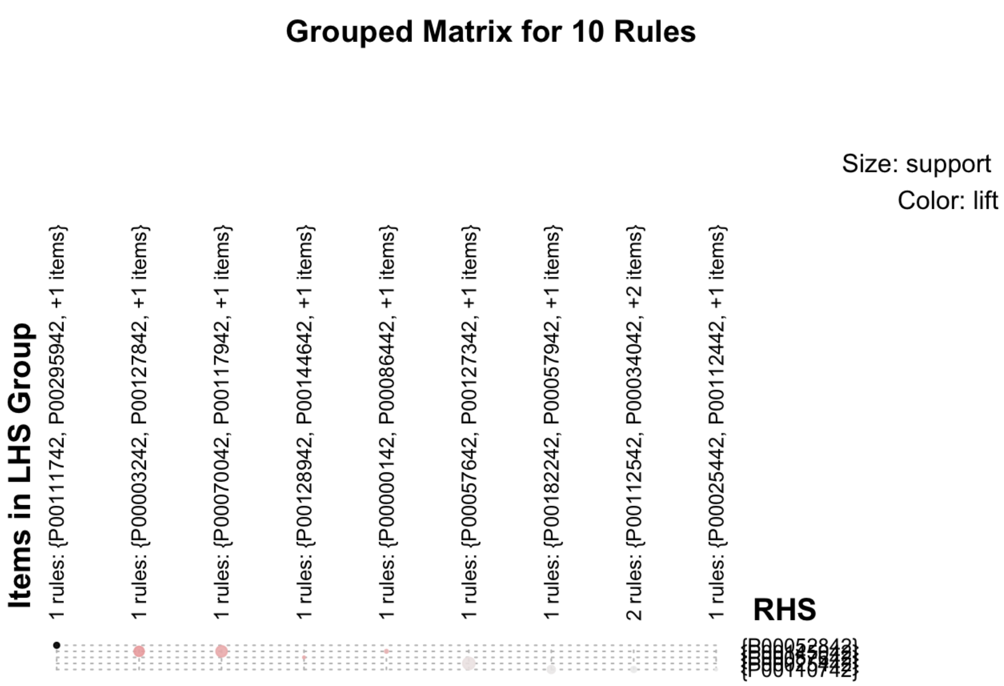
- Sort by Lift

```
inspect(sort(rules, by = 'lift'))
```

```
     lhs                                 rhs         support    confidence
[1]  {P00111742,P00295942,P00323942} => {P00052842} 0.01052274 0.7469880
[2]  {P00003242,P00111142,P00127842} => {P00145042} 0.01120163 0.7857143
[3]  {P00070042,P00117942,P00277642} => {P00145042} 0.01137135 0.7613636
[4]  {P00128942,P00144642,P00329542} => {P00057642} 0.01001358 0.7763158
[5]  {P00000142,P00086442,P00140742} => {P00145042} 0.01018330 0.7407407
[6]  {P00057642,P00105142,P00127342} => {P00025442} 0.01154107 0.7640449
[7]  {P00057942,P00105142,P00182242} => {P00110742} 0.01086219 0.7529412
[8]  {P00034042,P00057942,P00112542} => {P00110742} 0.01052274 0.7469880
[9]  {P00034042,P00111142,P00112542} => {P00110742} 0.01052274 0.7469880
[10] {P00025442,P00034042,P00112442} => {P00110742} 0.01001358 0.7468354
     coverage   lift     count
[1]  0.01408690 4.546749 62
[2]  0.01425662 3.344963 66
[3]  0.01493551 3.241297 67
[4]  0.01289885 3.198638 59
[5]  0.01374745 3.153500 60
[6]  0.01510523 2.838432 68
[7]  0.01442634 2.788391 64
[8]  0.01408690 2.766344 62
[9]  0.01408690 2.766344 62
[10] 0.01340801 2.765779 59
```

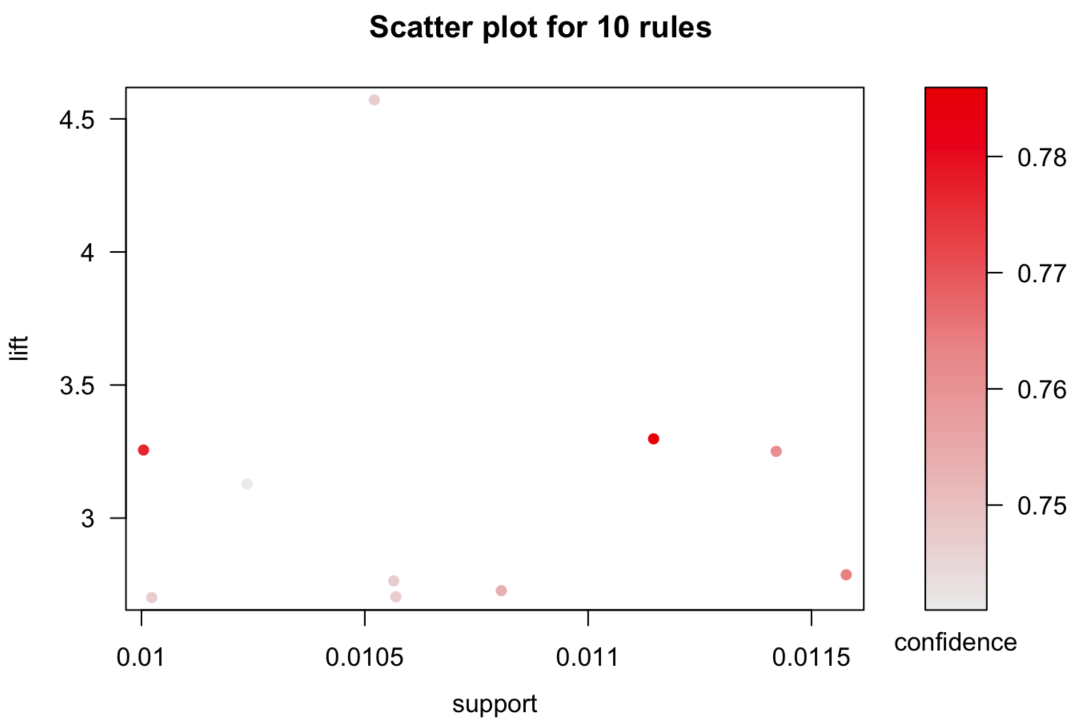
- Step 6: Plot a few graphs that can help you visualize the rules

```
library(arulesViz)
library(arules)
plot(rules, method = 'grouped', max = 4)
```

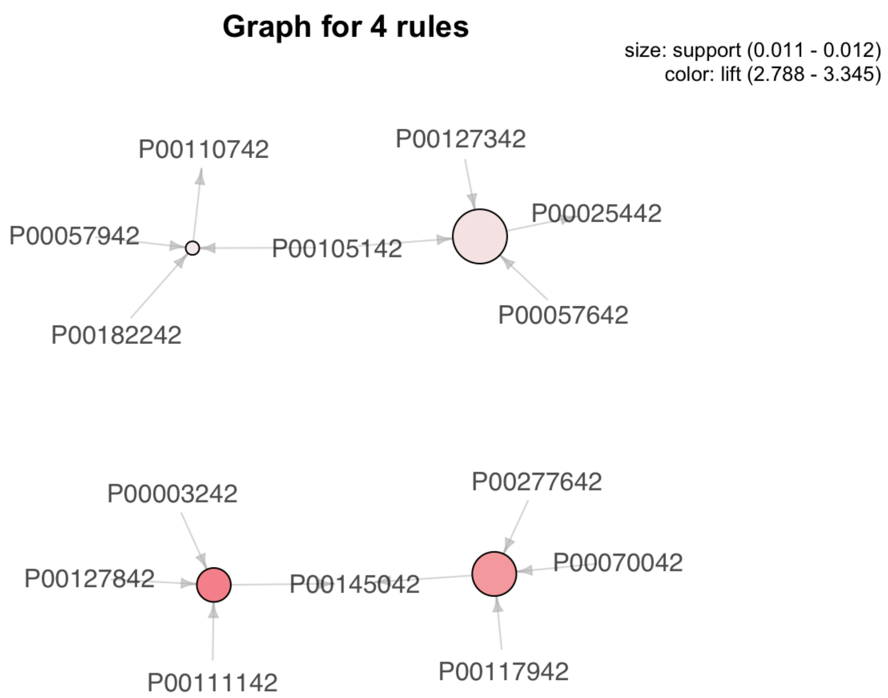


- Scatter Plot for the rules.

```
plot(rules,measure = c("support","lift"),shading = "confidence",jitter = 2)
```




```
plot(rules, method="graph",max = 4)
```



What rules lead to consequent?

- This can be done by filtering the rules to see what leads to a particular product

```
filter = 'P00110742'
rules_filtered <- subset(rules, subset = rhs %in% filter)

inspect(rules_filtered)
```

```
    lhs                                   rhs          support    confidence
[1] {P00025442,P00034042,P00112442} => {P00110742} 0.01001358 0.7468354
[2] {P00034042,P00057942,P00112542} => {P00110742} 0.01052274 0.7469880
[3] {P00034042,P00111142,P00112542} => {P00110742} 0.01052274 0.7469880
[4] {P00057942,P00105142,P00182242} => {P00110742} 0.01086219 0.7529412
    coverage   lift     count
[1] 0.01340801 2.765779 59
[2] 0.01408690 2.766344 62
[3] 0.01408690 2.766344 62
[4] 0.01442634 2.788391 64
```

CONCLUSION

In conclusion, the market basket analysis is studied in this analysis and it is one of the most popular association rules approach. In this study, “market basket optimization” dataset is analyzed, and results were obtained.. “arules” and “arulesViz” packages are mainly used in the analysis. Then, set of transactions are determined and rules for these transactions are analyzed. Moreover, support, confidence, lift and set of rules are found. After this step, all outputs were sorted for each method. The results are plotted and then the analysis is tested.