BlackFriday Analysis

Nisha Selvarajan

11/1/2020

Contents

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

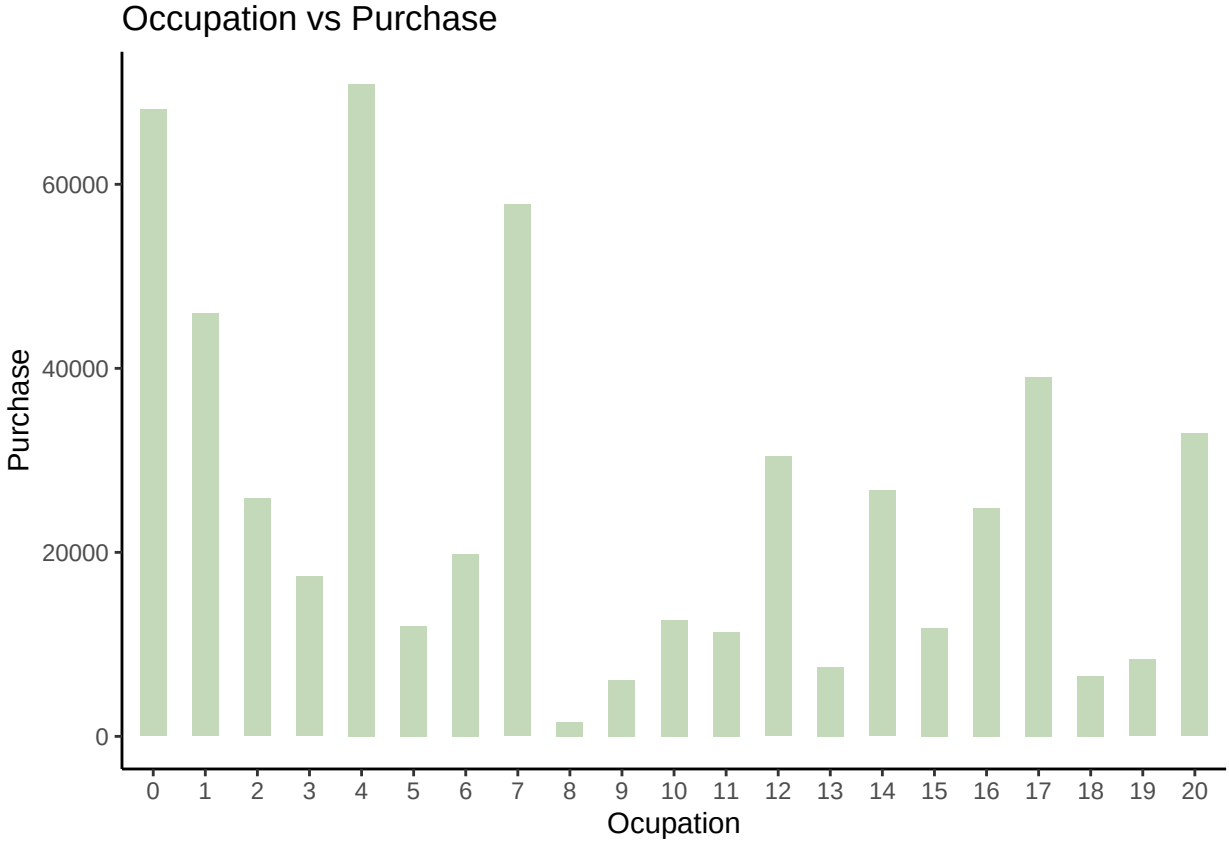
- The data used for this particular project is "Black Friday Sales Analysis"(https://www.kaggle.com/mehdidag/black-friday). Detailed description of the variables:

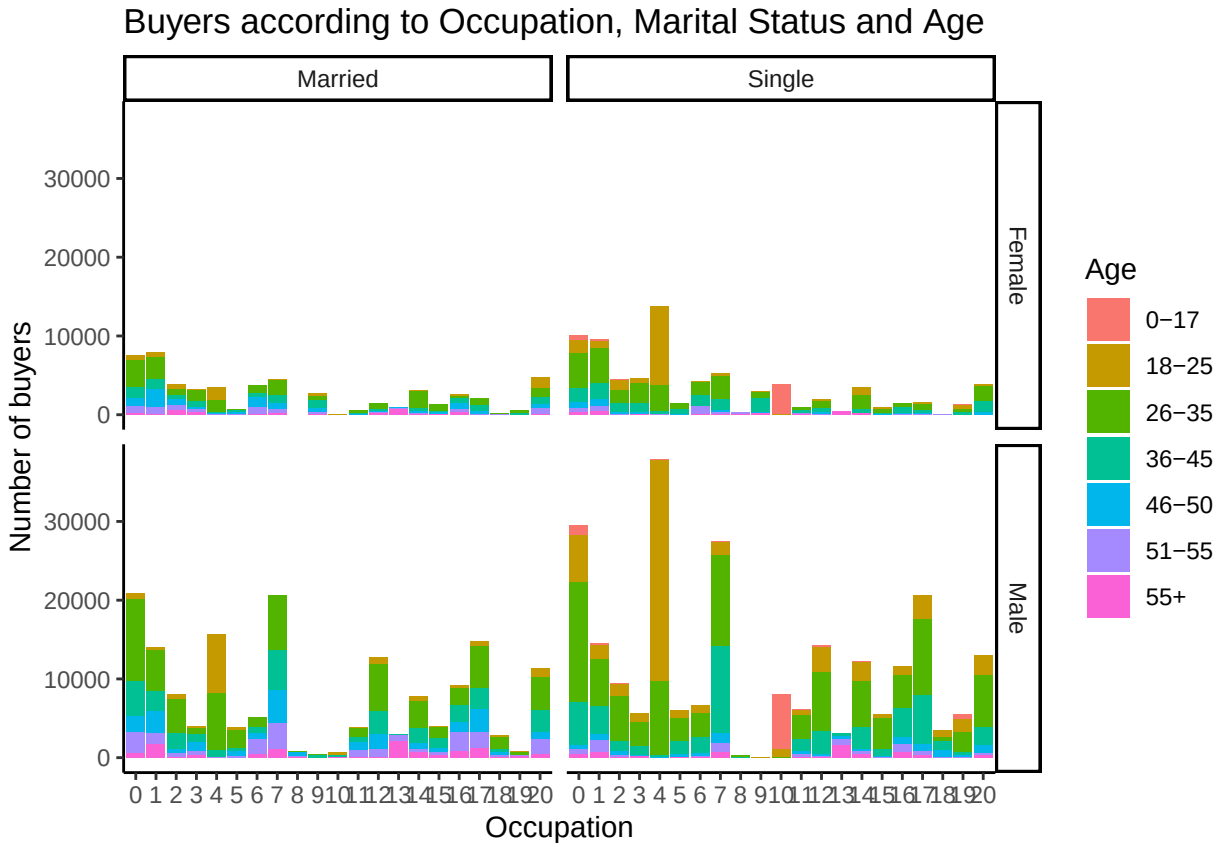
Names	Description
User_ID	Categorical - User ID
Product_ID	Categorical - Product ID
Gender	Categorical - Sex of User
Age	Categorical - Age in bins
Occupation	Categorical - Occupation (Masked)
City_Category	Categorical - Category of the City (A,B,C)
Stay_In_Current_City_Years	Numerical - Number of years 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

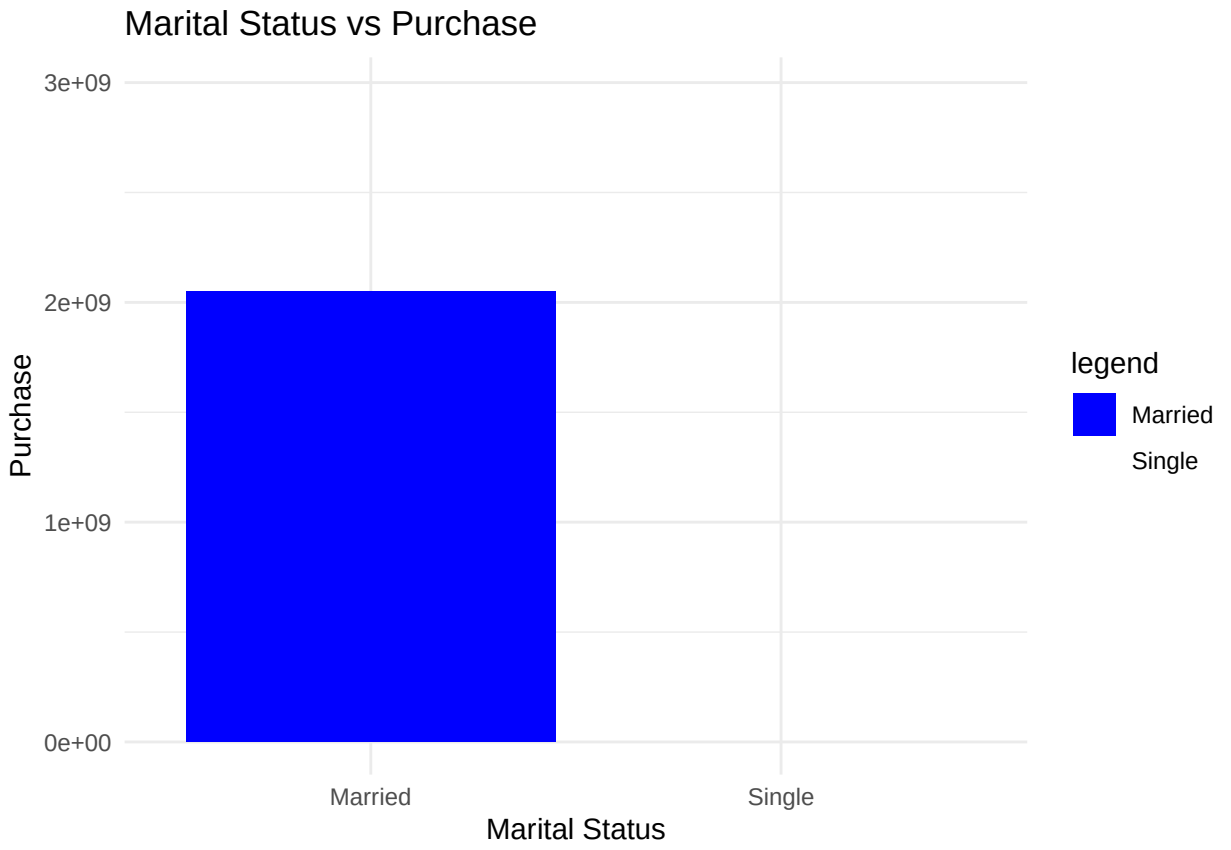





```
##
##          0     1     2     3    4+
##   A 23700 48160 26548 24389 21841
##   B 28143 81622 40800 41964 33964
##   C 20882 59410 32111 26959 27084
```

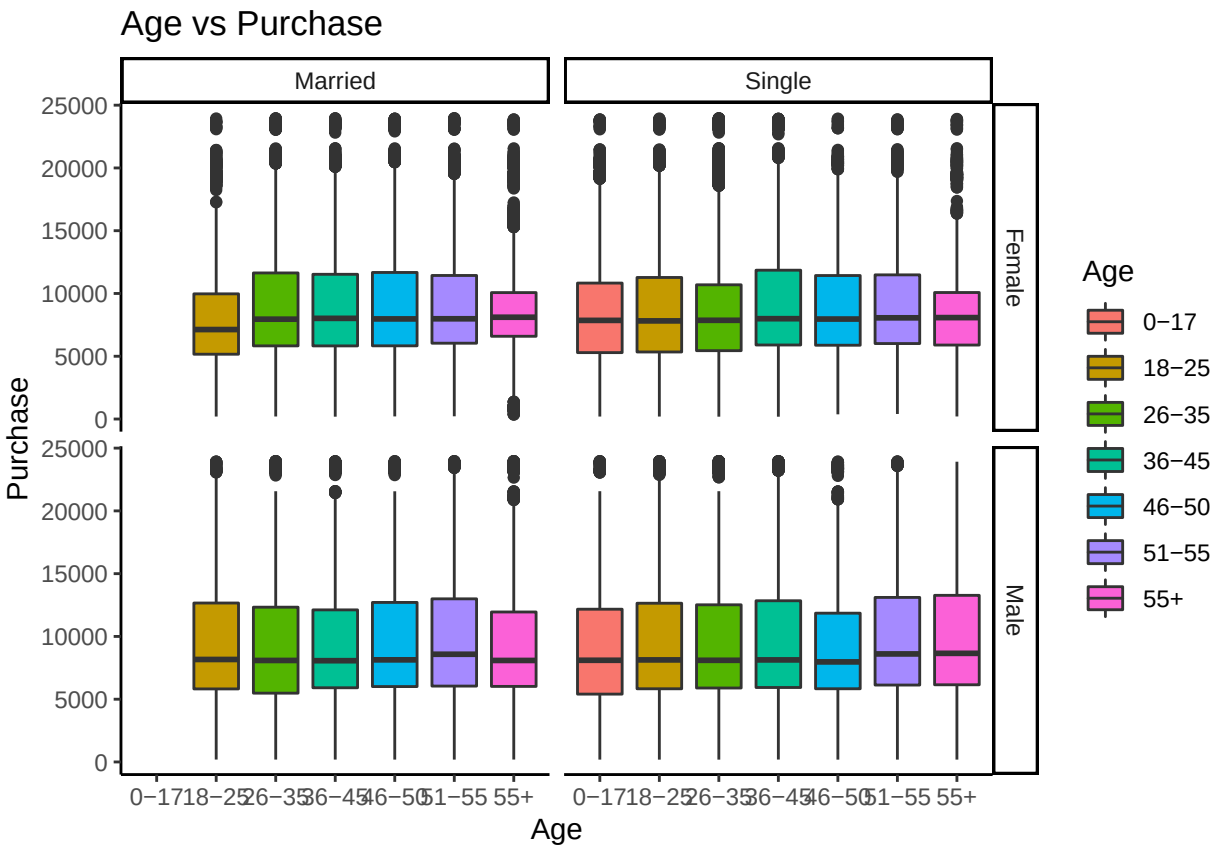

City Category vs Purchase
2.0e+09
1.5e+09
1.0e+09
5.0e+08
0.0e+00
Purchase
A
B
C
City Category
legend
A
B
C

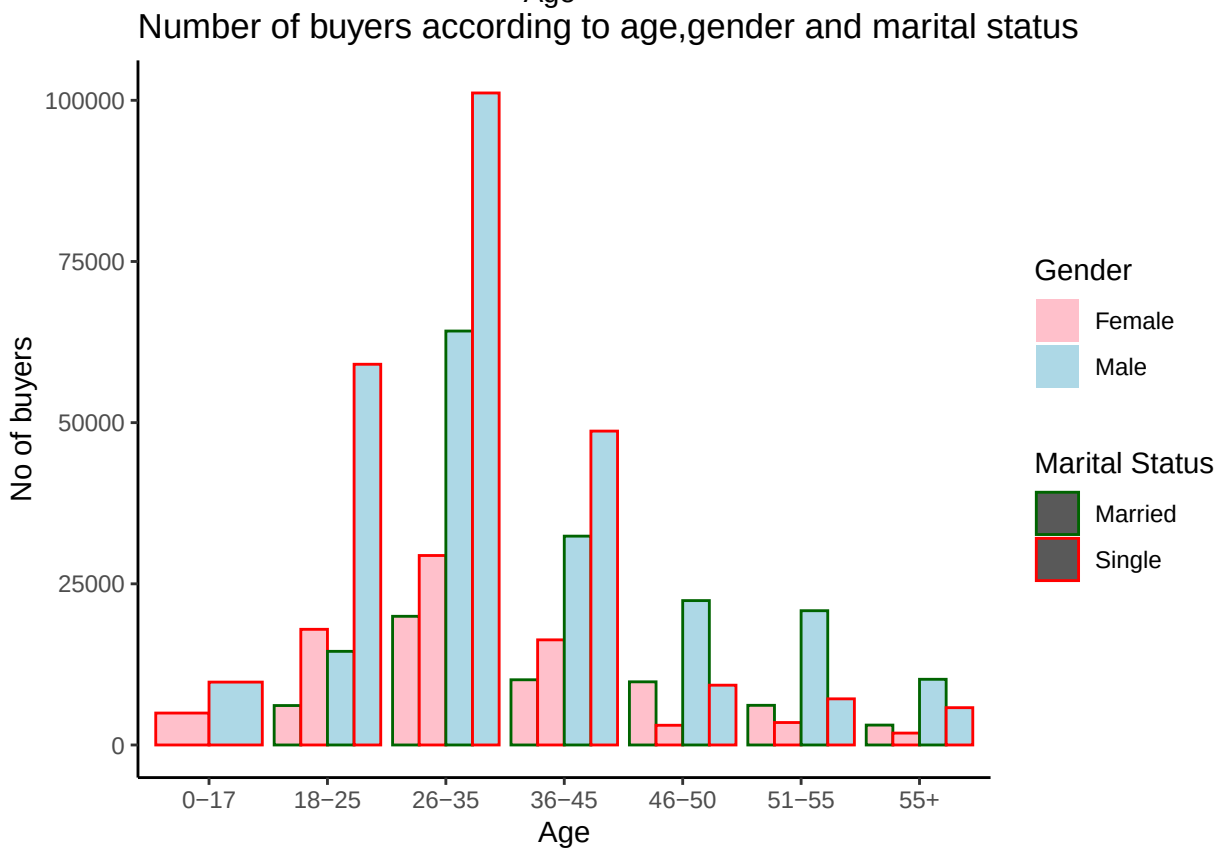

Number of people staying in Current city according to Marital Status
150000
100000
50000
0
Number of people
0
1
2
3
4+
Stay in Current City
City Category
A
B
C
Marital Status
Married
Single

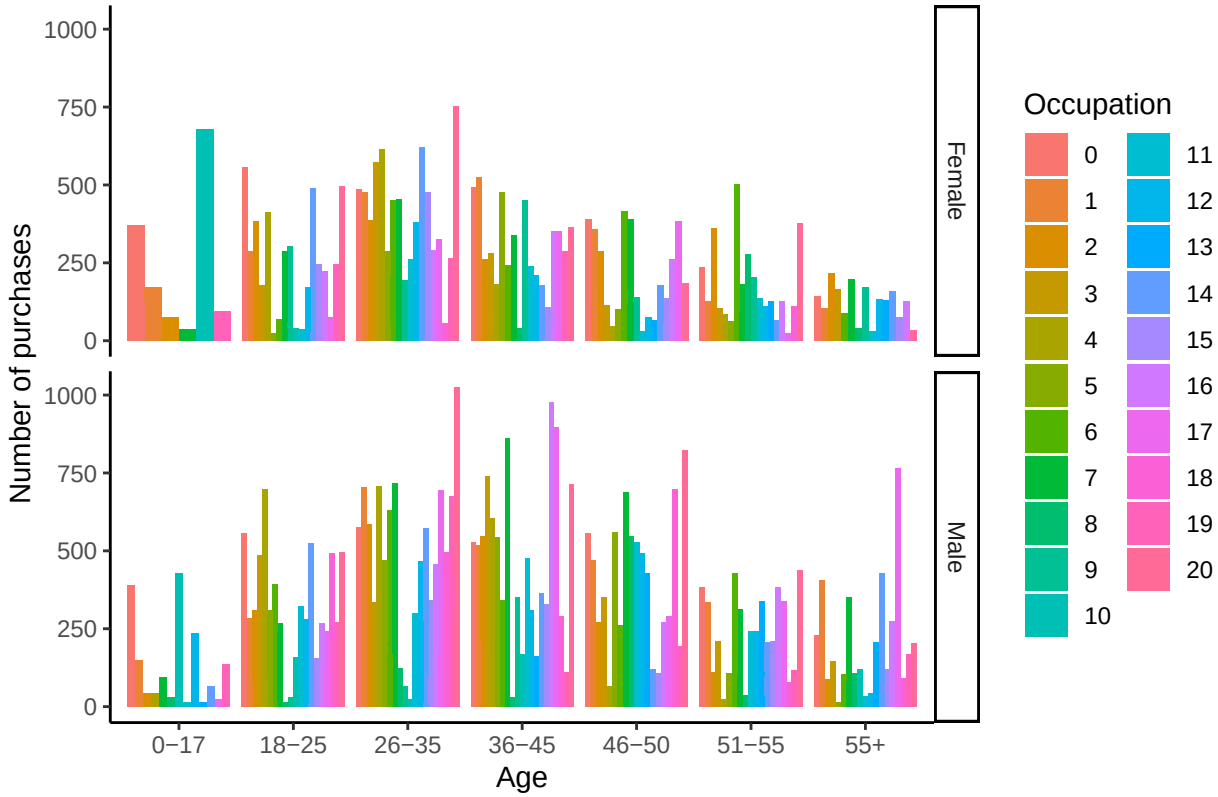

Marital Status vs Purchase
3e+09
2e+09
1e+09
0e+00
Purchase
Married
Single
Marital Status
legend
Married
Single

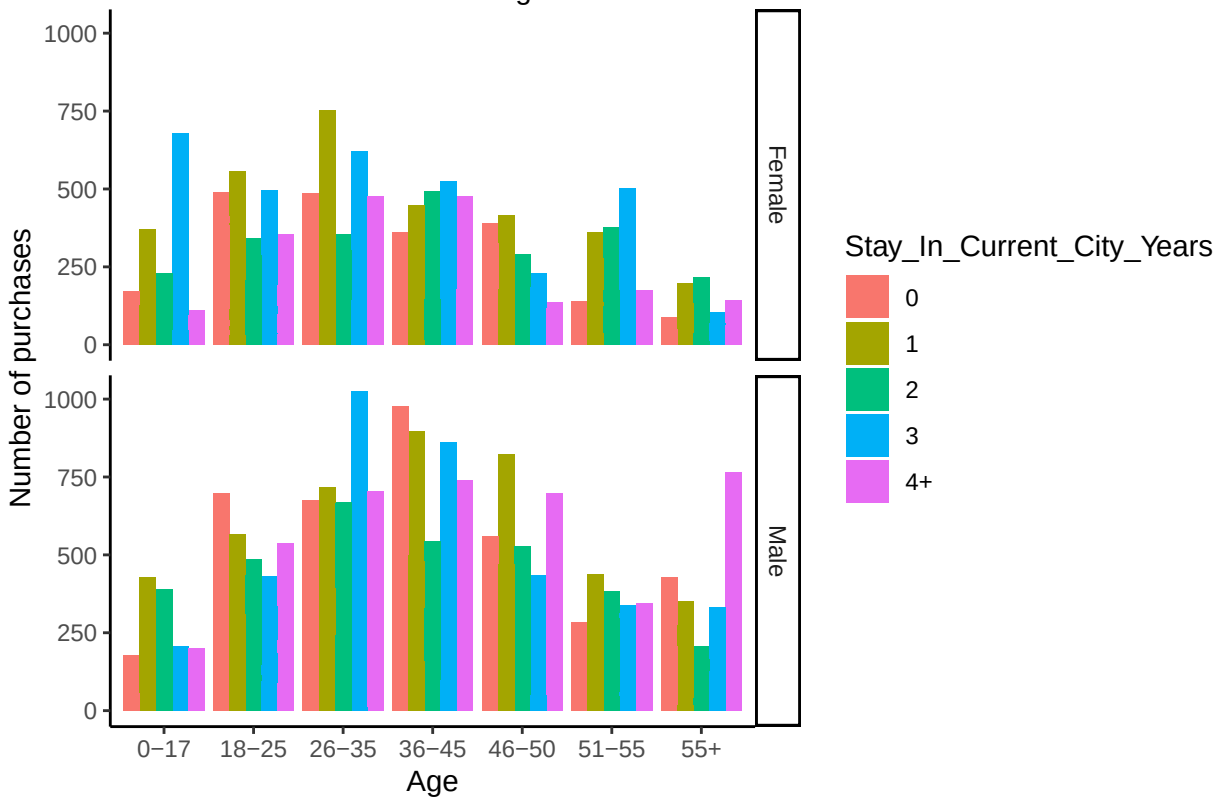

Stay in current city vs Purchase
150000
100000
50000
0
Purchase
0
1
2
3
4+
Stay in city

```
## , ,  = 0-17
##
##
##           Female   Male
##   Married      0      0
##   Single    4953   9754
##
## , ,  = 18-25
##
##
##           Female   Male
##   Married   6117  14524
##   Single   17940  59053
##
## , ,  = 26-35
##
##
##           Female   Male
##   Married  19959  64207
##   Single   29389 101135
##
## , ,  = 36-45
##
##
##           Female   Male
##   Married  10115  32392
##   Single   16305  48687
##
## , ,  = 46-50
##
##
##           Female   Male
##   Married   9797  22397
##   Single    3059   9273
##
## , ,  = 51-55
##
##
##           Female   Male
##   Married   6150  20829
##   Single    3484   7155
##
## , ,  = 55+
##
##
##           Female   Male
##   Married   3085  10188
##   Single    1844   5786
```

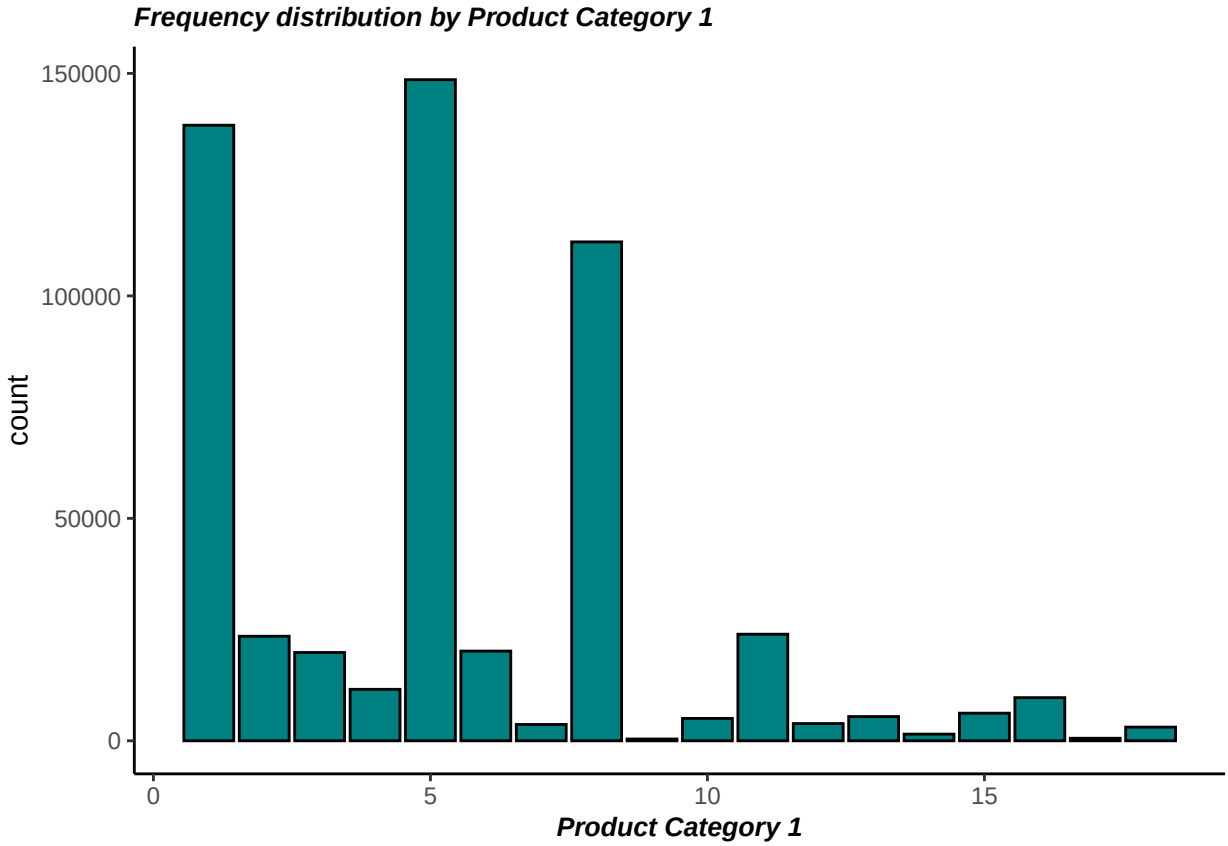

Age vs Purchase
Married
Single
Female
Male
Purchase
25000
20000
15000
10000
5000
0
0−17
18−25
26−35
36−45
46−50
51−55
55+
Age
Age
0−17
18−25
26−35
36−45
46−50
51−55
55+

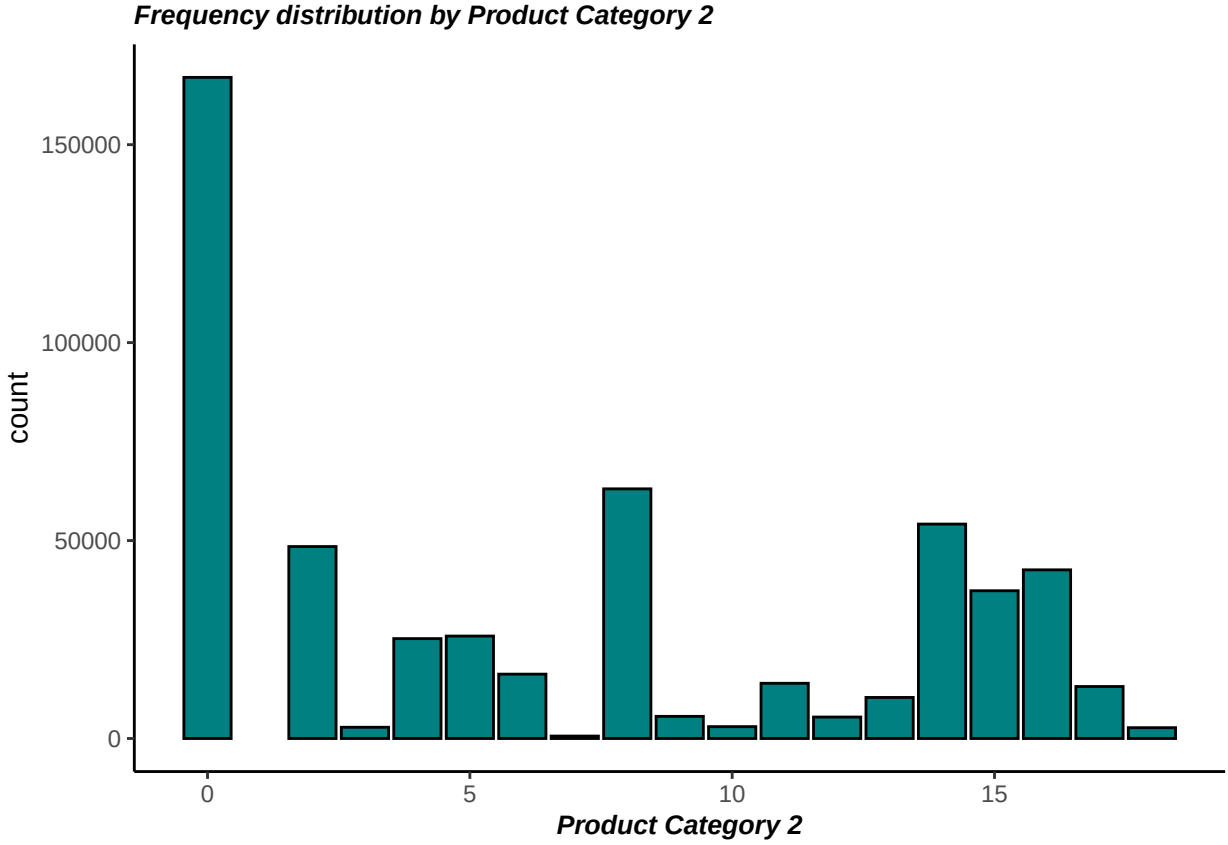

Number of buyers according to age,gender and marital status
No of buyers
100000
75000
50000
25000
0
0−17
18−25
26−35
36−45
46−50
51−55
55+
Age
Gender
Female
Male
Marital Status
Married
Single

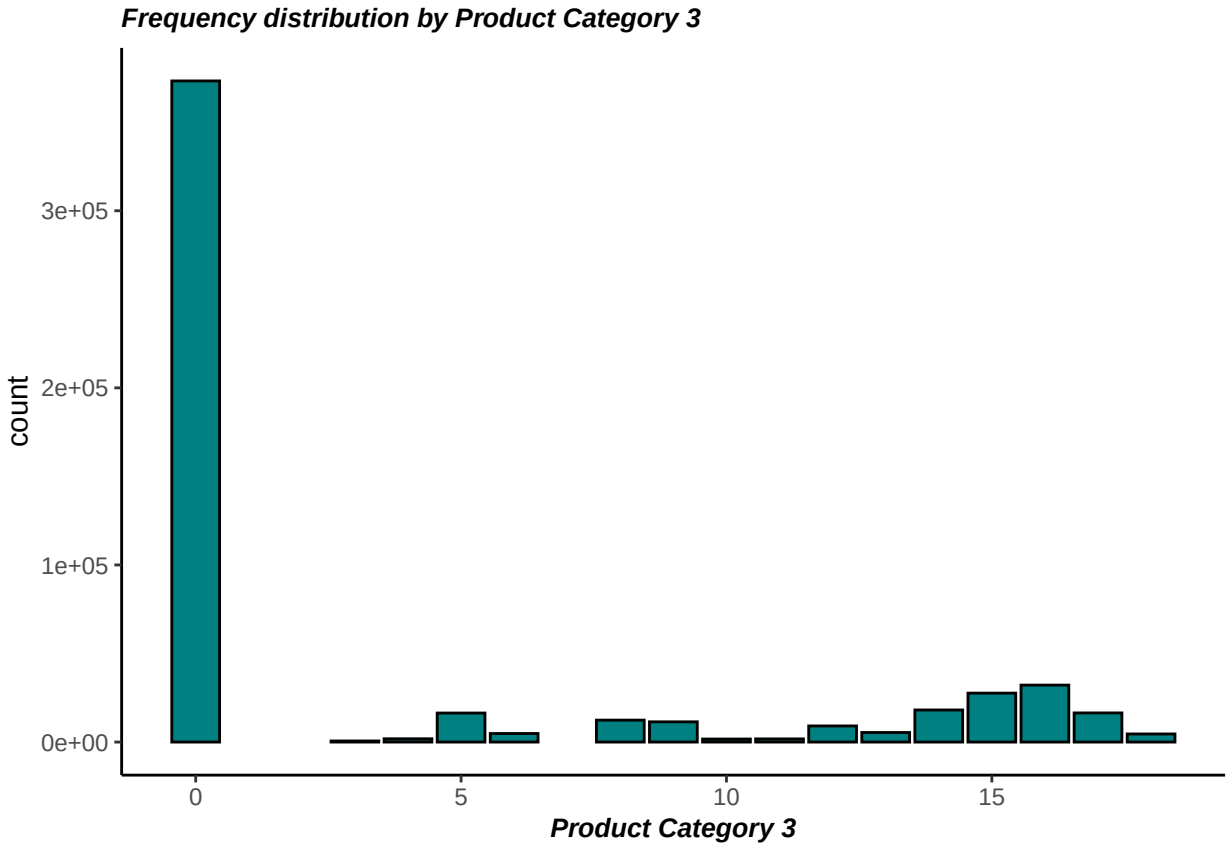

Number of purchases
1000
750
500
250
0
Female
Male
0–17
18–25
26–35
36–45
46–50
51–55
55+
Age
Occupation
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20


Number of purchases
1000
750
500
250
0
Female
Male
0–17
18–25
26–35
36–45
46–50
51–55
55+
Age
Stay_In_Current_City_Years
0
1
2
3
4+

Frequency Distribution By Product Category







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
##
```

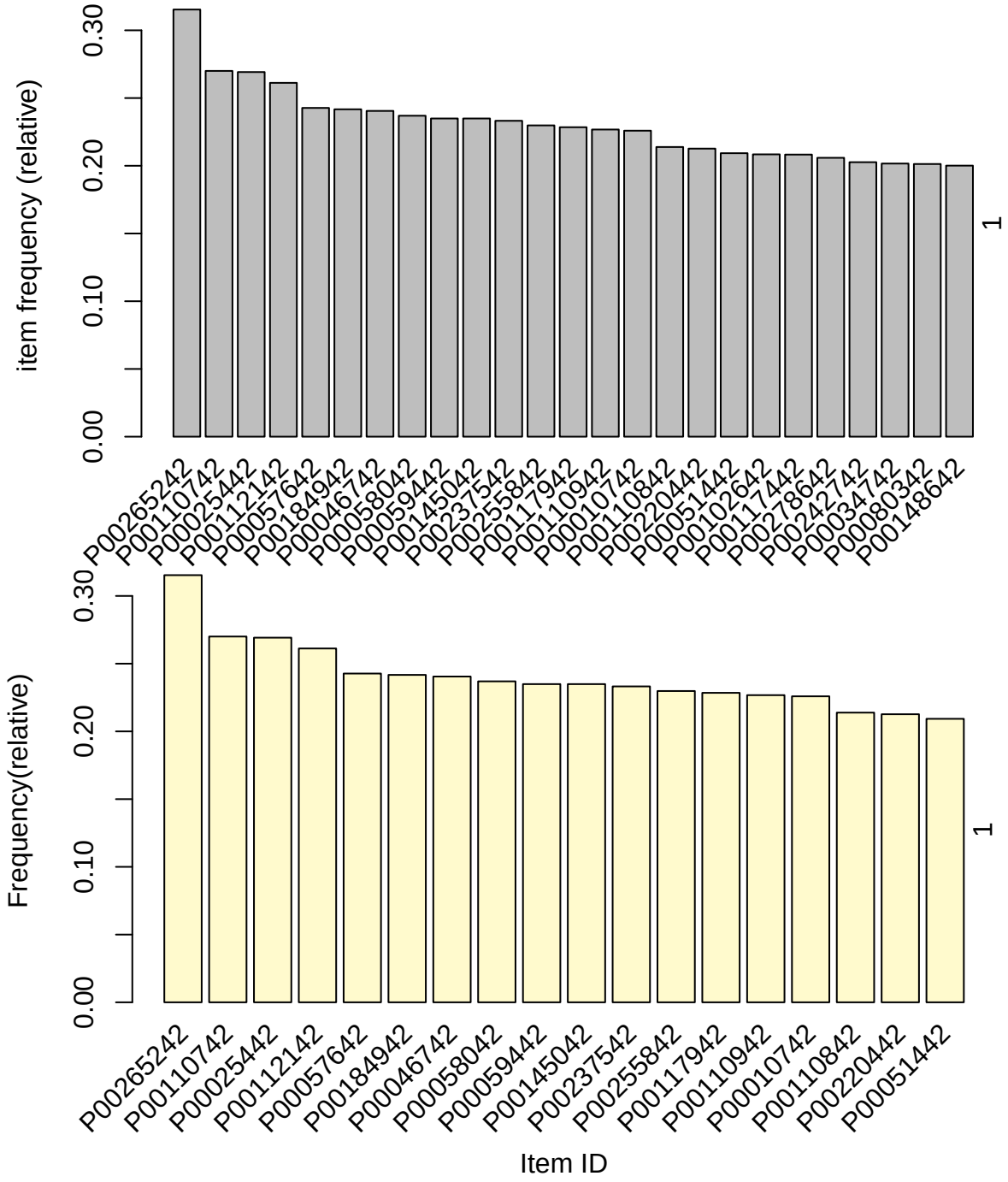
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
##
## includes extended item information - examples:
##                                             labels
## 1  1000001,P00000142,F,0-17,10,A,2,0,3,4,5,13650
## 2 1000001,P00004842,F,0-17,10,A,2,0,3,4,12,13645
## 3  1000001,P00025442,F,0-17,10,A,2,0,1,2,9,15416
```

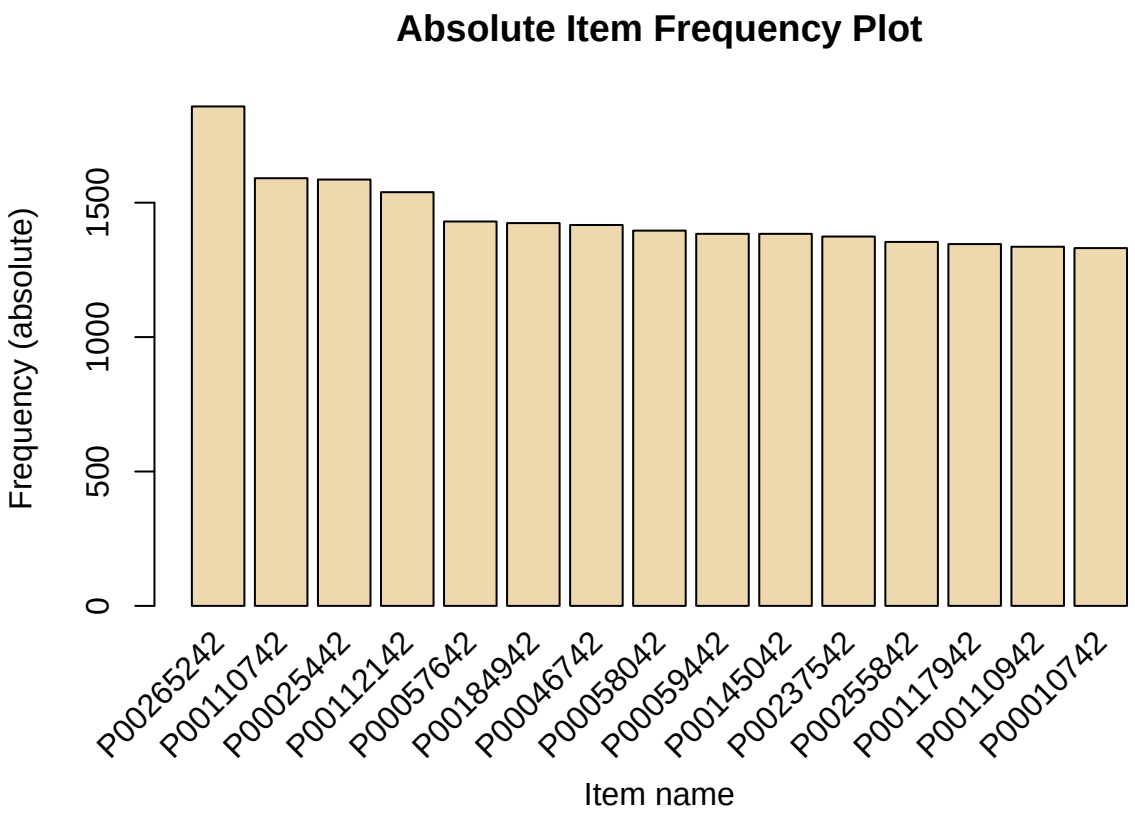
- Step 2: Data cleaning and manipulations using R.
  - Group the transactions by USER ID. The data required for Apriori must be in the basket format.The basket format must have first column as a unique identifier of each transaction, something like a unique product Id. The second columns consists of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

```
## distribution of transactions with duplicates:
## items
##   46  126  163  202  258  272  285  307  310  316  319  327  330  334  340  344
##    1    1    1    1    1    1    1    1    1    1    1    2    1    1    1    1
##  345  348  354  357  373  393  402  408  419  437  441  449  450  452  454  456
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  459  465  466  467  475  476  477  481  487  491  495  498  507  523  524  526
##    1    1    2    2    1    1    1    1    2    2    3    2    1    1    2    1
##  527  528  530  531  532  533  535  537  538  539  540  545  546  548  549  553
##    2    1    1    2    1    1    1    1    2    3    1    1    1    3    1    1
##  554  555  556  558  563  566  567  570  572  574  575  577  578  580  583  584
##    2    1    1    2    1    3    1    3    2    4    1    2    3    2    1    1
##  586  588  589  590  591  592  593  594  595  597  598  601  602  604  607  608
##    2    3    5    1    1    1    1    1    3    2    1    1    1    1    2    1
##  610  612  613  614  615  616  617  618  619  620  623  625  632  633  634  635
##    1    2    1    2    1    1    2    1    3    2    1    1    6    1    5    2
##  638  640  641  642  643  644  645  646  647  648  653  654  657  658  659  661
##    2    2    1    2    1    2    3    1    4    1    1    3    2    1    1    2
##  662  663  664  665  666  667  668  669  670  671  672  674  676  677  678  679
##    1    3    1    1    2    1    2    4    2    1    3    1    1    2    3    3
##  681  682  683  685  686  687  688  689  690  691  692  694  695  697  698  699
##    2    2    4    4    5    2    2    1    1    1    1    2    1    1    1    1
##  700  702  703  704  705  706  707  708  709  710  712  713  714  715  716  717
##    2    1    3    1    1    2    2    3    2    2    4    5    2    2    1    1
##  718  719  720  721  722  723  724  725  726  727  728  729  730  732  733  734
##    2    3    3    5    2    1    1    2    5    2    1    3    3    2    1    5
##  735  736  737  738  739  740  741  742  743  744  745  747  748  749  750  751
##    6    2    4    6    3    4    1    6    7    4    6    1    1    5    5    3
##  752  753  754  755  756  757  758  759  760  761  763  764  765  766  767  768
##    2    3    4    6    6    2    6    2    1    5    3    4    2    3    2    5
##  769  770  771  772  773  774  775  776  777  778  779  780  781  782  783  784
##    2    2    3    1    3    4    2    2    3    3    2    4    7    3    4    5
##  785  786  787  788  789  790  791  792  793  794  795  796  797  798  799  800
##    5    3    3    6    5    5    2    3    5    3    8    5    5    9    3    4
##  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816
##    4    7    4    3    4    5    7    5    4    5    3    2    6    6    3   11
```

```
##  817  818  819  820  821  822  823  824  825  826  827  828  829  830  831  832
##    5   10    6    6    4    7    7    2    5    4    7    5    5    4    5    4
##  833  834  835  836  837  838  839  840  841  842  843  844  845  846  847  848
##    3    5    4   11    5    5    4    9    7    6    4    7    9   11    4    6
##  849  850  851  852  853  854  855  856  857  858  859  860  861  862  863  864
##   10    6   10    7   12   16   11    8    7    4   12    9   11   11    9    6
##  865  866  867  868  869  870  871  872  873  874  875  876  877  878  879  880
##   11   10    7    6    5   12    6    7    8   11    9    9    8    7    5    4
##  881  882  883  884  885  886  887  888  889  890  891  892  893  894  895  896
##   15   13   12    8    4    6   12   15   13   10   11   13    6   21    7   14
##  897  898  899  900  901  902  903  904  905  906  907  908  909  910  911  912
##    9    7   11   18    5   14   10    9   19   15   10   17   18   23    8   19
##  913  914  915  916  917  918  919  920  921  922  923  924  925  926  927  928
##   15   12   18   21   17   12   11   13   13   12   20   20   16   13   15   17
##  929  930  931  932  933  934  935  936  937  938  939  940  941  942  943  944
##   27   22   20   28   18   14   20   20   20   14   22   30   23   23   21   20
##  945  946  947  948  949  950  951  952  953  954  955  956  957  958  959  960
##   25   19   30   31   30   24   27   25   40   30   31   16   29   30   32   48
##  961  962  963  964  965  966  967  968  969  970  971  972  973  974  975  976
##   27   27   24   30   26   35   43   30   51   49   40   41   36   32   36   38
##  977  978  979  980  981  982  983  984  985  986  987  988  989  990  991  992
##   43   41   42   37   49   44   51   57   55   40   53   56   63   39   58   50
##  993  994  995  996  997  998  999 1000 1001 1002 1003 1004 1005 1006 1007 1008
##   58   77   74   72   72   84   74   66   77   85   93   79   94  118  122  104
## 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019
##  121  113  120   78   77   55   37   20    7    5    1
```

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 [0.01s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [7.77s].
## writing ... [10 rule(s)] done [0.07s].
## creating S4 object  ... done [0.15s].
```

- 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'))
```

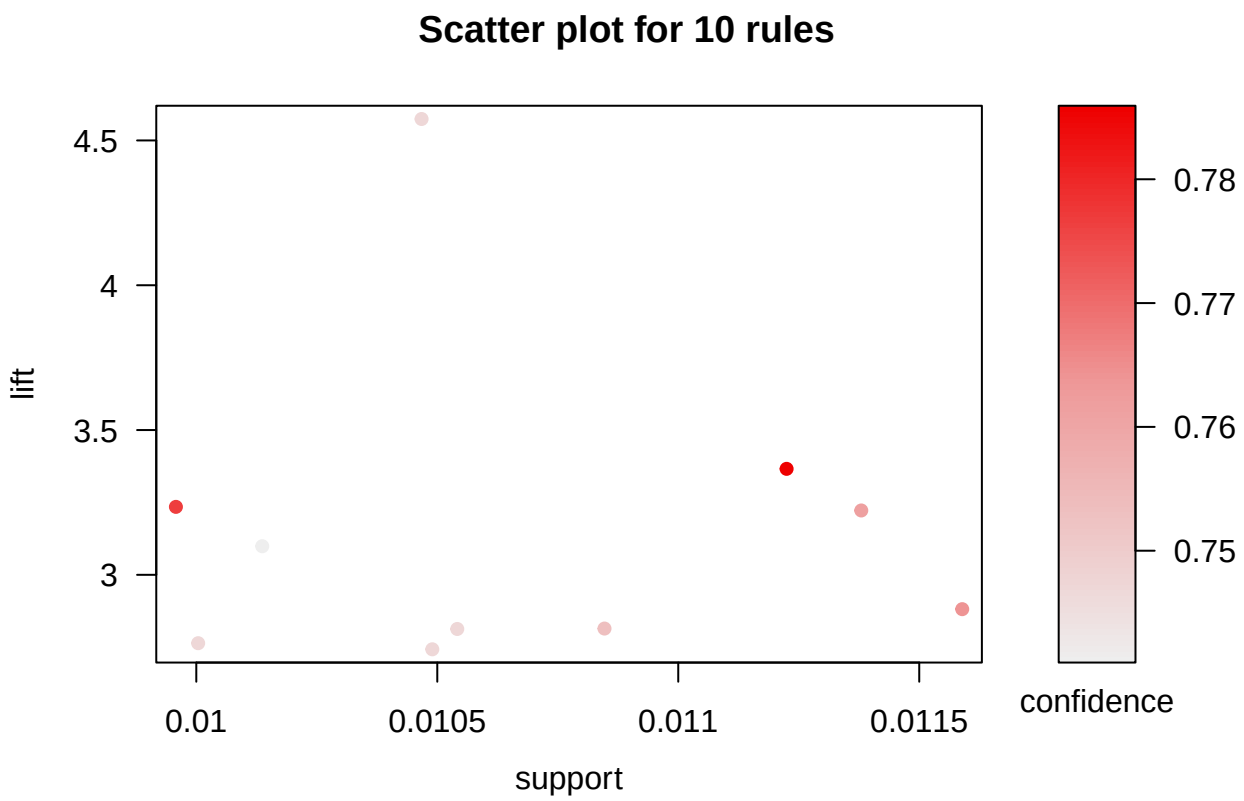
```
##      lhs                                rhs         support    confidence
## [1]  {P00111742,P00295942,P00323942} => {P00052842} 0.01052274 0.7469880
## [2]  {P00003242,P00111142,P00127842} => {P00145042} 0.01120163 0.7857143
## [3]  {P00070042,P00117942,P00277642} => {P00145042} 0.01137135 0.7613636
## [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

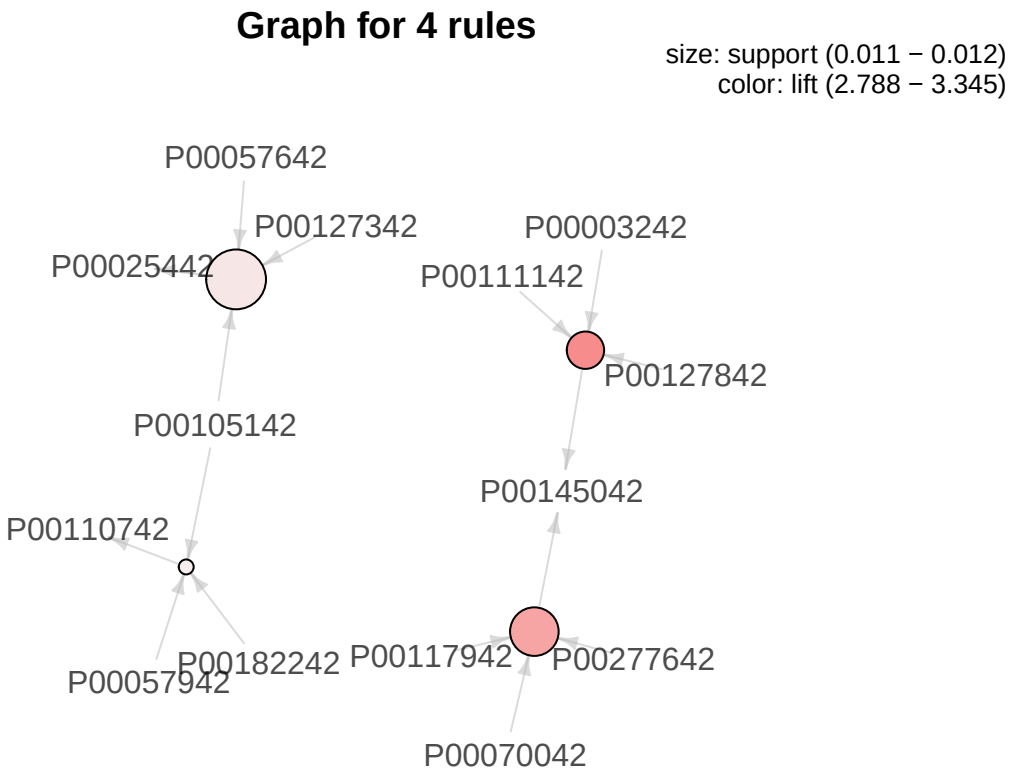
```
library(arulesViz)
library(arules)
plot(rules, method = 'grouped', max = 4)
```

- 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.