Assignment -1

Name :Arpitha Somayaji
ID:1001398103

Perform a market-basket analysis on the marketbasketNew.csv file that is under “datasets” on Blackboard. You must first convert the file to the right format before you run your analysis.

Python Code : Convert the file to right format.

```
import pandas as pd
import csv

#Read CSV file from location to marketBasket_data
marketBasket_data = pd.read_csv("C:/Users/Arpitha Somayaji/Desktop/Fall
2017/DATA SCIENCE/Homework/MarketBasket/marketbasketData.csv")
#Store the product names in Columnnames
columnnames = list(marketBasket_data)

#iterate through each row in marketBasket_data . Append it to csv file
if  row[column] equals to true
for index, row in marketBasket_data.iterrows():
    column = []
    for names in columnnames:
        if row[names].strip() == 'true':
            column.append(names)

    with open('C:/Users/Arpitha Somayaji/Desktop/Fall 2017/DATA
SCIENCE/Homework/MarketBasket/Modified_marketbasketData.csv','a') as
fp:
        wr = csv.writer(fp, dialect='excel')
        wr.writerow(column)
```

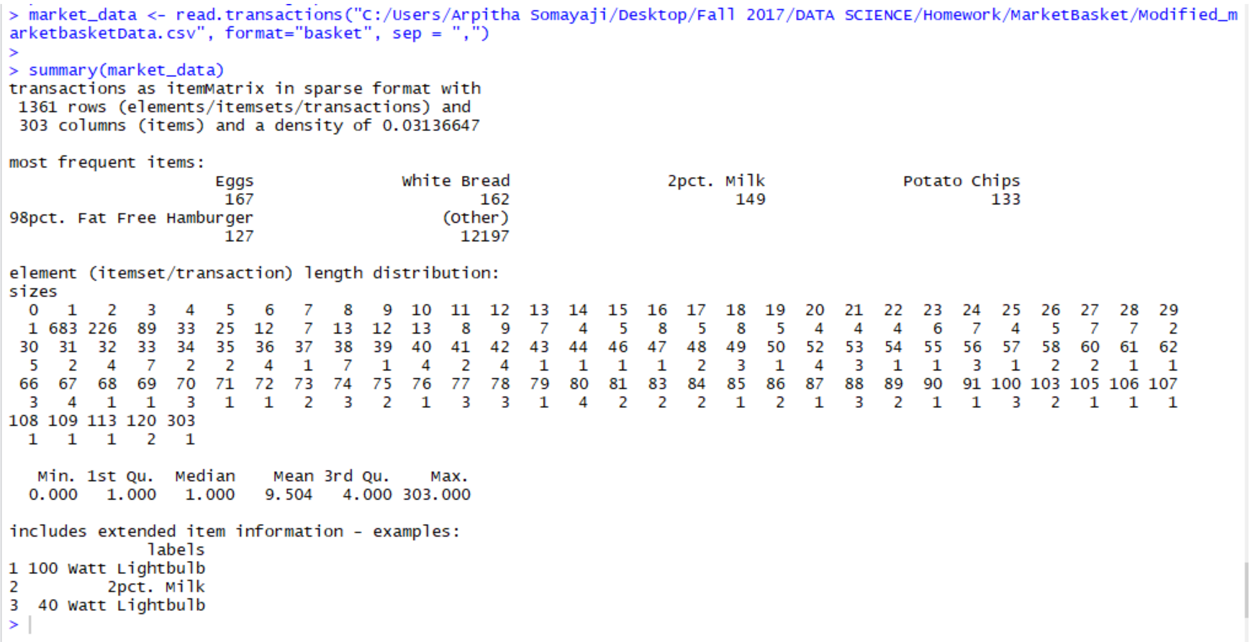
R Program to perform Market Basket Analysis

```
install.packages("arules")
install.packages("arulesViz")
library(arules)
library(arulesViz)
market_data <- read.transactions("C:/Users/Arpitha
Somayaji/Desktop/Fall 2017/DATA
SCIENCE/Homework/MarketBasket/Modified_marketbasketData.csv",
format="basket", sep = ",")
```

Assignment -1

Name :Arpitha Somayaji

ID:1001398103



```
> market_data <- read.transactions("C:/Users/Arpitha Somayaji/Desktop/Fall 2017/DATA SCIENCE/Homework/MarketBasket/Modified_m
arketbasketData.csv", format="basket", sep = ",")
> 
> summary(market_data)
transactions as itemMatrix in sparse format with
 1361 rows (elements/itemsets/transactions) and
 303 columns (items) and a density of 0.03136647 

most frequent items:
                     Eggs                White Bread                  2pct. Milk               Potato Chips 
                      167                        162                         149                        133 
98pct. Fat Free Hamburger                    (Other) 
                      127                      12197 

element (itemset/transaction) length distribution:
sizes
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 
  1 683 226  89  33  25  12   7  13  12  13   8   9   7   4   5   8   5   8   5   4   4   4   6   7   4   5   7   7   2 
 30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  46  47  48  49  50  52  53  54  55  56  57  58  60  61  62 
  5   2   4   7   2   2   4   1   7   1   4   2   4   1   1   1   1   2   3   1   4   3   1   1   3   1   2   2   1   1 
 66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  83  84  85  86  87  88  89  90  91 100 103 105 106 107 
  3   4   1   1   3   1   1   2   3   2   1   3   3   1   4   2   2   2   1   2   1   3   2   1   1   3   2   1   1   1 
108 109 113 120 303 
  1   1   1   2   1 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   1.000   1.000   9.504   4.000 303.000 

includes extended item information - examples:
              labels
1 100 Watt Lightbulb
2         2pct. Milk
3  40 Watt Lightbulb
> |
```

A. Display an item frequency plot. To display a reasonable number of items, use the support parameter or the topN parameter. An example of how to use the item frequency plot command is given below:

itemFrequencyPlot(t, support = 0.06, horiz = TRUE, type = “absolute”)

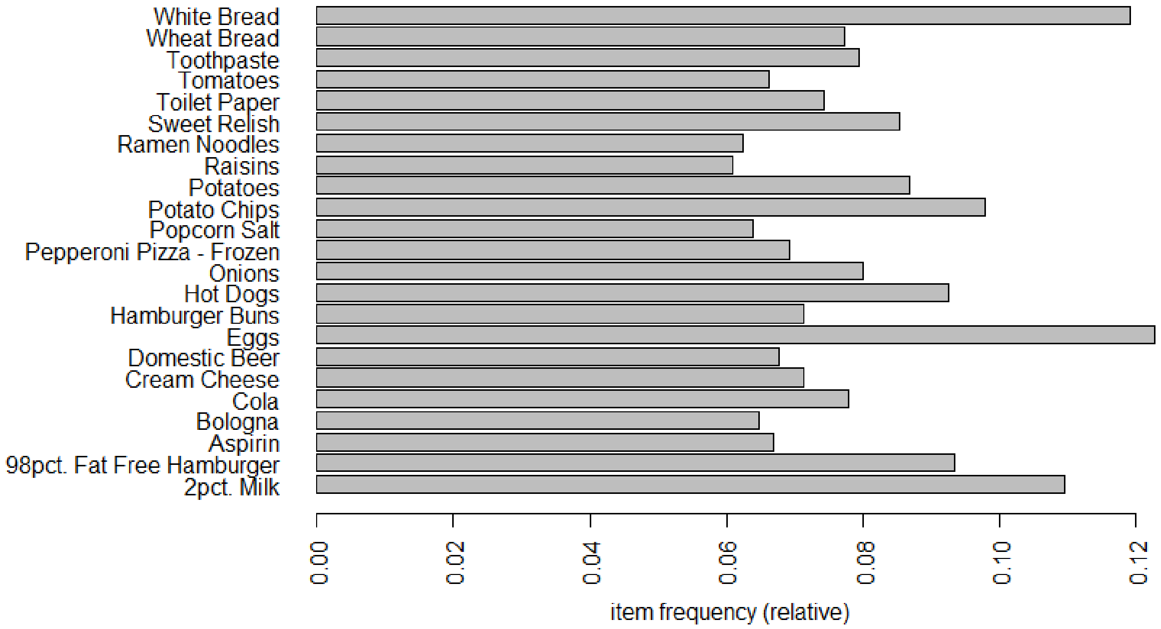
support = 0.06 implies that only items that occur in at least 6% of the transactions will be displayed. For this dataset, 5% to 8% should be reasonable. Type can be changed to “relative” if you desire relative frequencies. You may use either “absolute” or “relative”.

Assignment -1

Name :Arpitha Somayaji

ID:1001398103

```
itemFrequencyPlot(market_data , support = 0.06, horiz = TRUE, type="relative"
)
```



B. Display the top 5 rules sorted in descending order by “Lift”.

```
rules <- apriori(market_data , parameter = list(supp = 0.006, conf = 0.8));
```

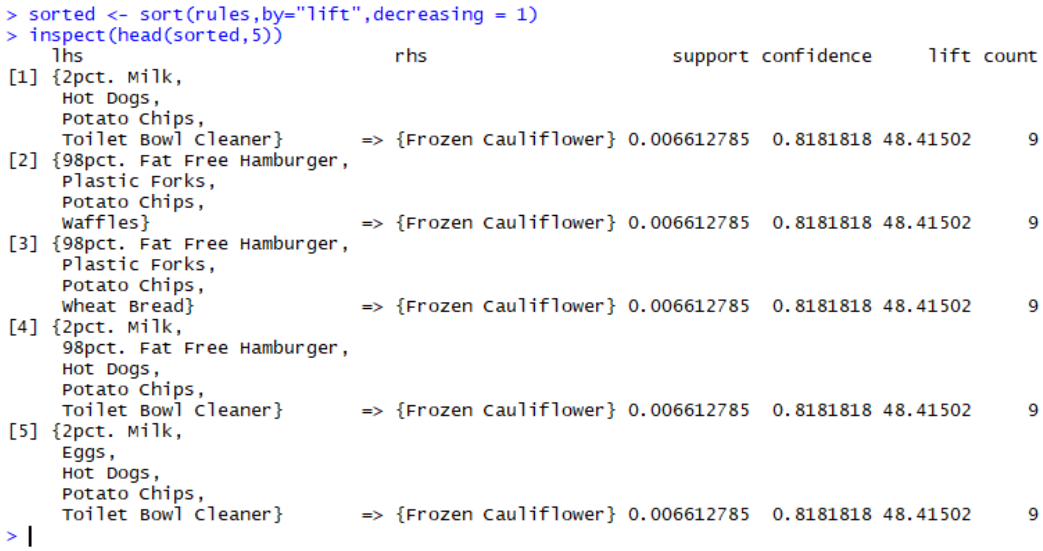
```
> rules <- apriori(market_data , parameter = list(supp = 0.006, conf = 0.8));
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE       5   0.006      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 8

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[303 item(s), 1361 transaction(s)] done [0.00s].
sorting and recoding items ... [295 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.60s].
writing ... [1229533 rule(s)] done [0.40s].
creating S4 object  ... done [4.46s].
```

Assignment -1

Name :Arpitha Somayaji

ID:1001398103

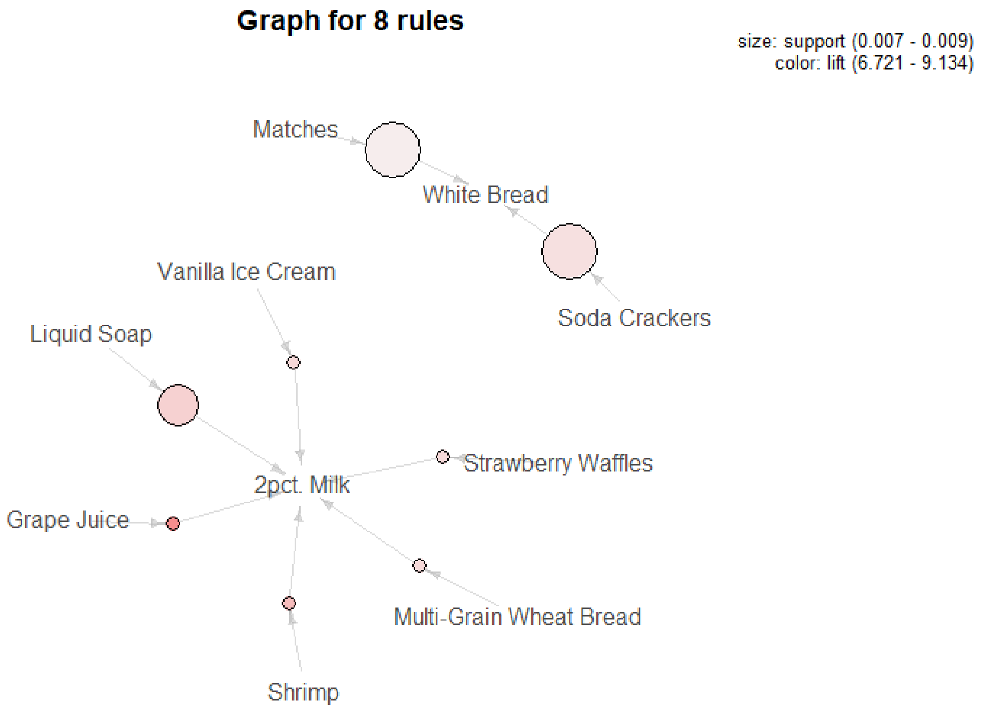
```
sorted <- sort(rules,by="lift",decreasing = 1)
inspect(head(sorted,5))
```



```
> sorted <- sort(rules,by="lift",decreasing = 1)
> inspect(head(sorted,5))
    lhs                            rhs                      support confidence     lift count
[1] {2pct. Milk,
     Hot Dogs,
     Potato Chips,
     Toilet Bowl Cleaner}       => {Frozen Cauliflower} 0.006612785  0.8181818 48.41502     9
[2] {98pct. Fat Free Hamburger,
     Plastic Forks,
     Potato Chips,
     Waffles}                   => {Frozen Cauliflower} 0.006612785  0.8181818 48.41502     9
[3] {98pct. Fat Free Hamburger,
     Plastic Forks,
     Potato Chips,
     Wheat Bread}               => {Frozen Cauliflower} 0.006612785  0.8181818 48.41502     9
[4] {2pct. Milk,
     98pct. Fat Free Hamburger,
     Hot Dogs,
     Potato Chips,
     Toilet Bowl Cleaner}       => {Frozen Cauliflower} 0.006612785  0.8181818 48.41502     9
[5] {2pct. Milk,
     Eggs,
     Hot Dogs,
     Potato Chips,
     Toilet Bowl Cleaner}       => {Frozen Cauliflower} 0.006612785  0.8181818 48.41502     9
> |
```

C. Display a “graph” of the top 8 rules.

```
> plot(rules[1:8],method="graph")
```



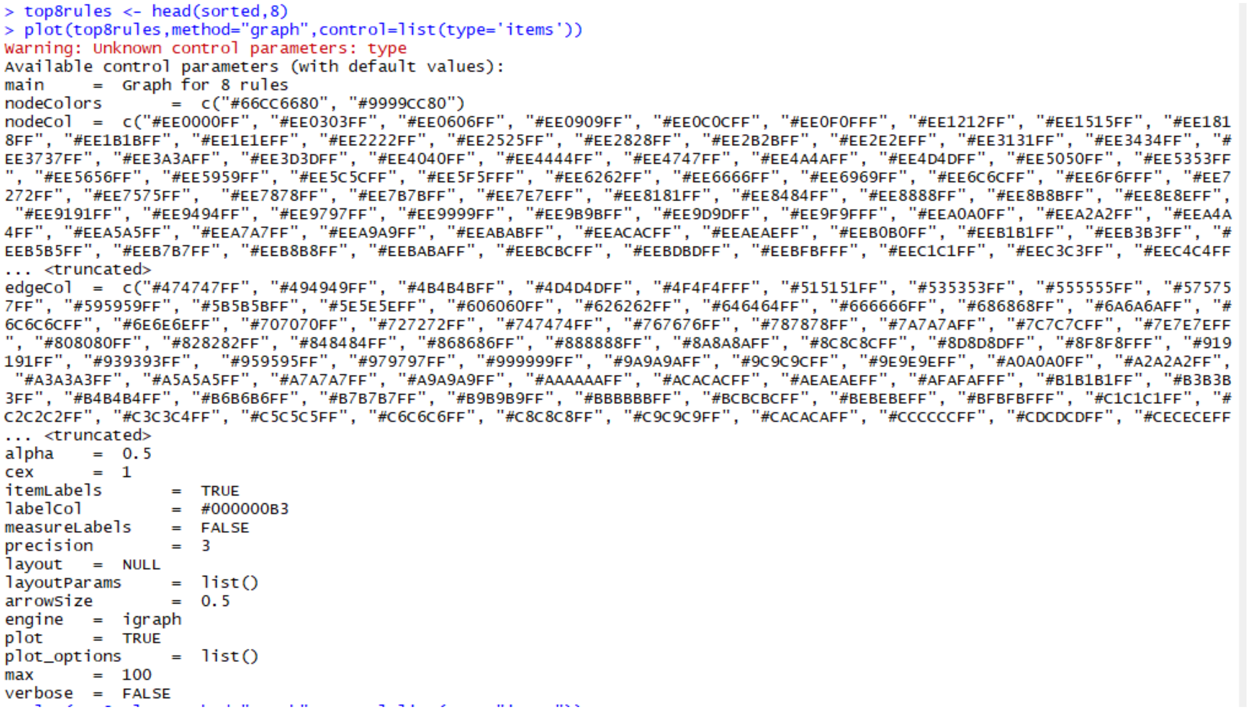
Assignment -1

Name :Arpitha Somayaji

ID:1001398103

Display the top 8 rules sorted in descending order by “Lift”.

```
> top8rules <- head(sorted,8)
> plot(top8rules,method="graph",control=list(type='items'))
```

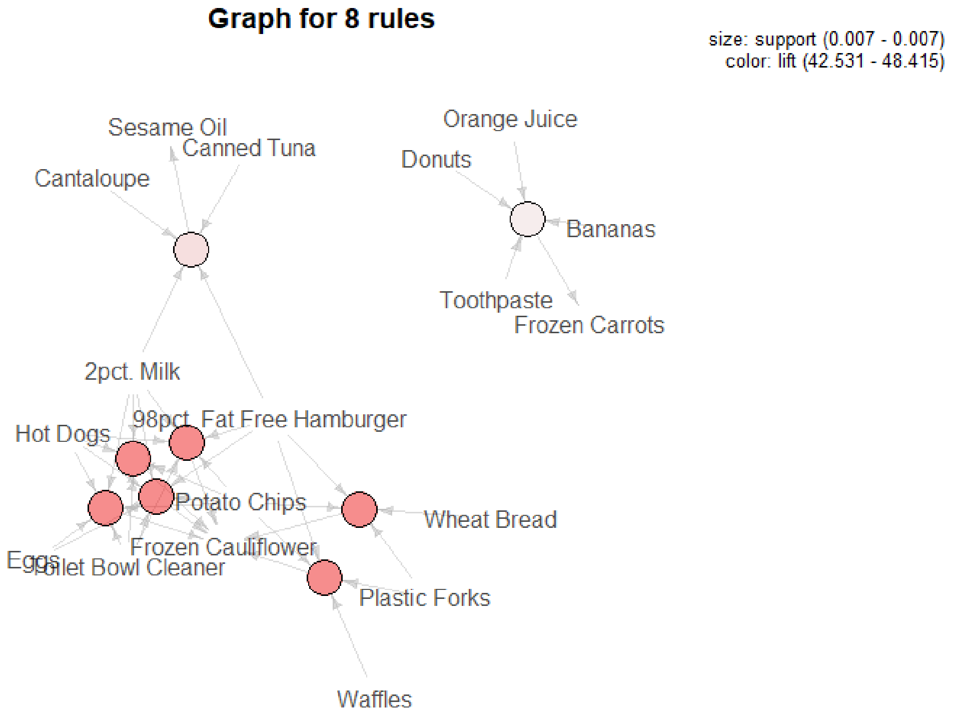


```
> top8rules <- head(sorted,8)
> plot(top8rules,method="graph",control=list(type='items'))
Warning: Unknown control parameters: type
Available control parameters (with default values):
main     =  Graph for 8 rules
nodeColors       =  c("#66CC6680", "#9999CC80")
nodeCol  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#EE1515FF", "#EE181
8FF", "#EE1B1BFF", "#EE1E1EFF", "#EE2222FF", "#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF", "#EE3131FF", "#EE3434FF", "#
EE3737FF", "#EE3A3AFF", "#EE3D3DFF", "#EE4040FF", "#EE4444FF", "#EE4747FF", "#EE4A4AFF", "#EE4D4DFF", "#EE5050FF", "#EE5353FF
", "#EE5656FF", "#EE5959FF", "#EE5C5CFF", "#EE5F5FFF", "#EE6262FF", "#EE6666FF", "#EE6969FF", "#EE6C6CFF", "#EE6F6FFF", "#EE7
272FF", "#EE7575FF",  "#EE7878FF", "#EE7B7BFF", "#EE7E7EFF", "#EE8181FF", "#EE8484FF", "#EE8888FF", "#EE8B8BFF", "#EE8E8EFF",
 "#EE9191FF", "#EE9494FF", "#EE9797FF", "#EE9999FF", "#EE9B9BFF", "#EE9D9DFF", "#EE9F9FFF", "#EEA0A0FF", "#EEA2A2FF", "#EEA4A
4FF", "#EEA5A5FF", "#EEA7A7FF", "#EEA9A9FF", "#EEABABFF", "#EEACACFF", "#EEAEAEFF", "#EEB0B0FF", "#EEB1B1FF", "#EEB3B3FF", "#
EEB5B5FF", "#EEB7B7FF", "#EEB8B8FF", "#EEBABAFF", "#EEBCBCFF", "#EEBDBDFF", "#EEBFBFFF", "#EEC1C1FF", "#EEC3C3FF", "#EEC4C4FF
... <truncated>
edgeCol  =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353FF", "#555555FF", "#57575
7FF", "#595959FF", "#5B5B5BFF", "#5E5E5EFF", "#606060FF", "#626262FF", "#646464FF", "#666666FF", "#686868FF", "#6A6A6AFF", "#
6C6C6CFF", "#6E6E6EFF", "#707070FF", "#727272FF", "#747474FF", "#767676FF", "#787878FF", "#7A7A7AFF", "#7C7C7CFF", "#7E7E7EFF
", "#808080FF", "#828282FF", "#848484FF", "#868686FF", "#888888FF", "#8A8A8AFF", "#8C8C8CFF", "#8D8D8DFF", "#8F8F8FFF", "#919
191FF", "#939393FF",  "#959595FF", "#979797FF", "#999999FF", "#9A9A9AFF", "#9C9C9CFF", "#9E9E9EFF", "#A0A0A0FF", "#A2A2A2FF",
 "#A3A3A3FF", "#A5A5A5FF", "#A7A7A7FF", "#A9A9A9FF", "#AAAAAAFF", "#ACACACFF", "#AEAEAEFF", "#AFAFAFFF", "#B1B1B1FF", "#B3B3B
3FF", "#B4B4B4FF", "#B6B6B6FF", "#B7B7B7FF", "#B9B9B9FF", "#BBBBBBFF", "#BCBCBCFF", "#BEBEBEFF", "#BFBFBFFF", "#C1C1C1FF", "#
C2C2C2FF", "#C3C3C4FF", "#C5C5C5FF", "#C6C6C6FF", "#C8C8C8FF", "#C9C9C9FF", "#CACACAFF", "#CCCCCCFF", "#CDCDCDFF", "#CECECEFF
... <truncated>
alpha    =  0.5
cex      =  1
itemLabels       =  TRUE
labelCol         =  #000000B3
measureLabels    =  FALSE
precision        =  3
layout   =  NULL
layoutParams     =  list()
arrowSize        =  0.5
engine   =  igraph
plot     =  TRUE
plot_options     =  list()
max      =  100
verbose  =  FALSE
```

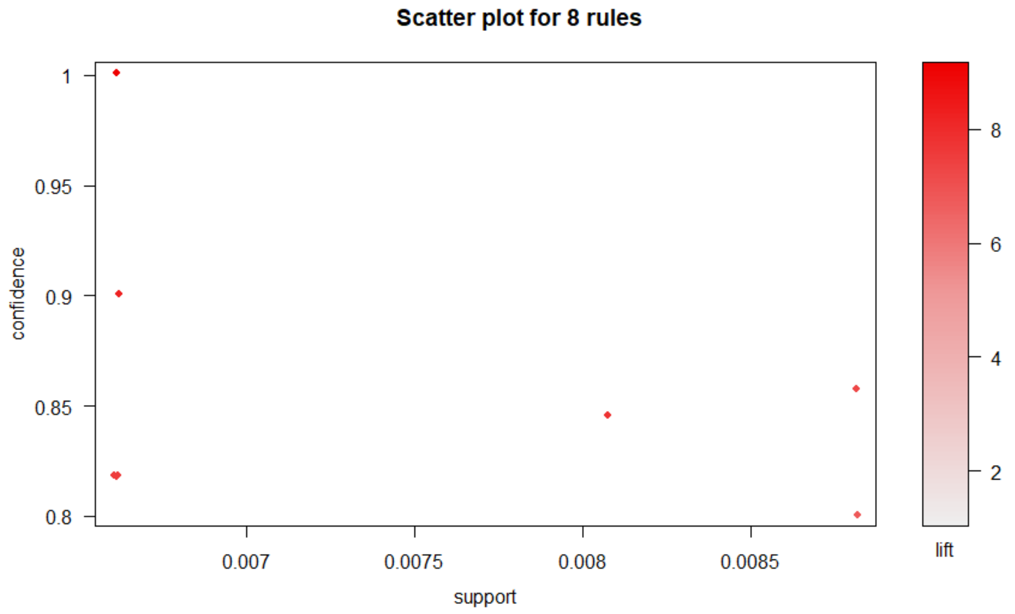
Assignment -1

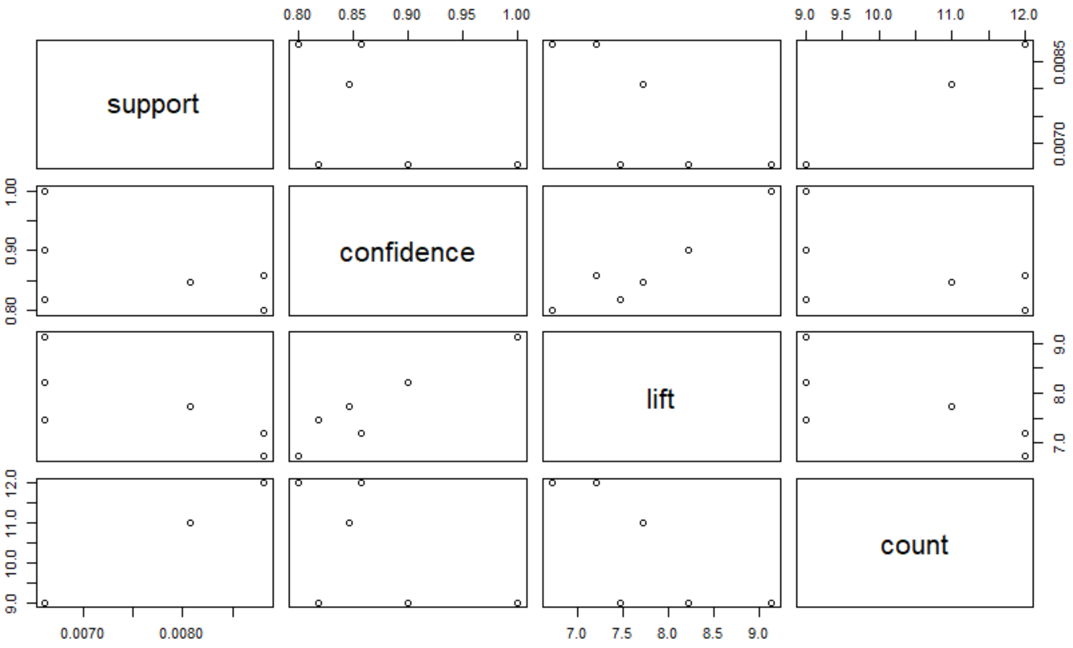
Name :Arpitha Somayaji

ID:1001398103



```
> plot(rules[1:8])
> plot(rules[1:8]@quality)
```


Scatter plot for 8 rules
confidence
support
lift


support
confidence
lift
count