Association Rules. The most commonly bought products.

Introduction

By using association rules method, I would like to demonstrate following project; Here, you can see what are the most common products people buy.

Processing the Data:

```
install.packages("arules")
install.packages("arulesViz")
install.packages("arulesCBA")
```

```
library(ggplot2)
library(readr)
library(arules)
library(arulesViz)
```

Dataset

```
setwd("D:\\R and R Studio\\Association Rules")
```

```
getwd()
```

```
data<-read.csv("dataset.csv", header=TRUE, sep=";")
```

```
names(data)
```

```
[1]
"pork..sandwich.bags..lunch.meat..all..purpose..flour..soda..butter..vegetables..beef..aluminum.foil..all..purpose..dinner.rolls..shampoo..all..purpose"
```

```
head(data)
```

```
pork..sandwich.bags..lunch.meat..all..purpose..flour..soda..butter..vegetables..beef..aluminum.foil..all..purpose..dinner.rolls..shampoo..all..purpose
1  shampoo, hand soap, waffles, vegetables, cheeses, mixes, milk, sandwich bags, laundry detergent, dishwashing liquid/detergent, waffles, individual meals, hand soap, vegetables
2                  pork, soap, ice cream, toilet paper, dinner rolls, hand soap, spaghetti sauce, milk, ketchup, sandwich loaves, poultry, toilet paper, ice cream, ketchup
3                                                                                  juice, lunch meat, soda, toilet paper, all- purpose, ,,,,,,,
4                        pasta, tortillas, mixes, hand soap, toilet paper, vegetables, vegetables, paper towels, vegetables, flour, vegetables, pork, poultry, eggs
5    toilet paper, eggs, toilet paper, vegetables, bagels, dishwashing liquid/detergent, cereals, paper towels, laundry detergent, butter, cereals, bagels, paper towels, shampoo
6            paper towels, tortillas, vegetables, milk, ice cream, juice, dishwashing liquid/detergent, soap, sandwich bags, pasta, ketchup, all- purpose, yogurt, mixes
```

```
system("ls ../input")
```

```
[1] 127
```

al<- read.csv('dataset.csv', header=F)

str(al)

```
> al<- read.csv('dataset.csv', header=F)
> str(al)
'data.frame':   1499 obs. of  14 variables:
 $ V1 : chr  " pork" " shampoo" " pork" " juice" ...
 $ V2 : chr  " sandwich bags" " hand soap" " soap" " lunch meat" ...
 $ V3 : chr  " lunch meat" " waffles" " ice cream" " soda" ...
 $ V4 : chr  " all- purpose" " vegetables" " toilet paper" " toilet paper" ...
 $ V5 : chr  " flour" " cheeses" " dinner rolls" " all- purpose" ...
 $ V6 : chr  " soda" " mixes" " hand soap" " " ...
 $ V7 : chr  " butter" " milk" " spaghetti sauce" "" ...
 $ V8 : chr  " vegetables" " sandwich bags" " milk" "" ...
 $ V9 : chr  " beef" " laundry detergent" " ketchup" "" ...
 $ V10: chr  " aluminum foil" " dishwashing liquid/detergent" " sandwich loaves" "" ...
 $ V11: chr  " all- purpose" " waffles" " poultry" "" ...
 $ V12: chr  " dinner rolls" " individual meals" " toilet paper" "" ...
 $ V13: chr  " shampoo" " hand soap" " ice cream" "" ...
 $ V14: chr  " all- purpose" " vegetables" " ketchup" "" ...
```

al1<-al[,1:6]

alnan<-as(al1, "transactions")

Here, I set supp=0.001 and conf=0.001, minimum support should be 0.001 and minimum confidence should be 0.001.

rule<-apriori(alnan, parameter = list(minlen=2, maxlen=4, supp=0.001, conf=0.001))

```
> rules<-apriori(alnan, parameter = list(minlen=2, maxlen=4, supp=0.001, conf=0.001))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
      0.001    0.1    1 none FALSE            TRUE       5   0.001      2      4  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 ...[231 item(s), 1499 transaction(s)] done [0.00s].
sorting and recoding items ... [231 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [13189 rule(s)] done [0.01s].
creating S4 object  ... done [0.00s].
Warning message:
In apriori(alnan, parameter = list(minlen = 2, maxlen = 4, supp = 0.001,  :
  Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!
```

inspect(alnan[1:5])

```
  items                transactionID
[1] {V1= pork,
     V2= sandwich bags,
     V3= lunch meat,
     V4= all- purpose,
     V5= flour,
     V6= soda}                    1
[2] {V1= shampoo,
     V2= hand soap,
     V3= waffles,
     V4= vegetables,
```

```
    V5= cheeses,
    V6= mixes}                  2
[3] {V1= pork,
    V2= soap,
    V3= ice cream,
    V4= toilet paper,
    V5= dinner rolls,
    V6= hand soap}              3
[4] {V1= juice,
    V2= lunch meat,
    V3= soda,
    V4= toilet paper,
    V5= all- purpose,
    V6= }                       4
[5] {V1= pasta,
    V2= tortillas,
    V3= mixes,
    V4= hand soap,
    V5= toilet paper,
    V6= vegetables}             5
```

inspect(yontem[1:5])

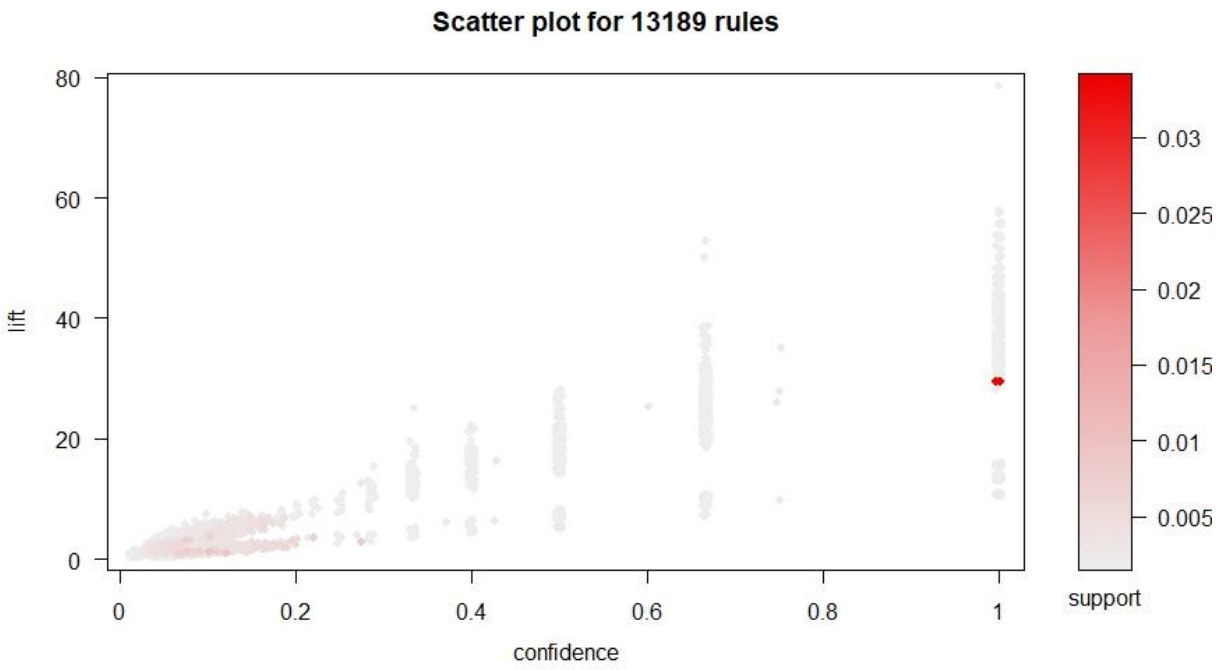
```
    lhs             rhs          support     confidence coverage   lift     count
[1] {V5= pork}    => {V2= ketchup} 0.001334223 0.10526316 0.01267512 6.068826 2
[2] {V2= ketchup} => {V5= pork}    0.001334223 0.07692308 0.01734490 6.068826 2
[3] {V5= pork}    => {V1= cheeses} 0.002001334 0.15789474 0.01267512 7.396382 3
[4] {V1= cheeses} => {V5= pork}    0.002001334 0.09375000 0.02134757 7.396382 3
[5] {V5= pork}    => {V6= waffles} 0.001334223 0.10526316 0.01267512 4.781499 2
```

rule<-sort(yontem, by="support", decreasing = T)

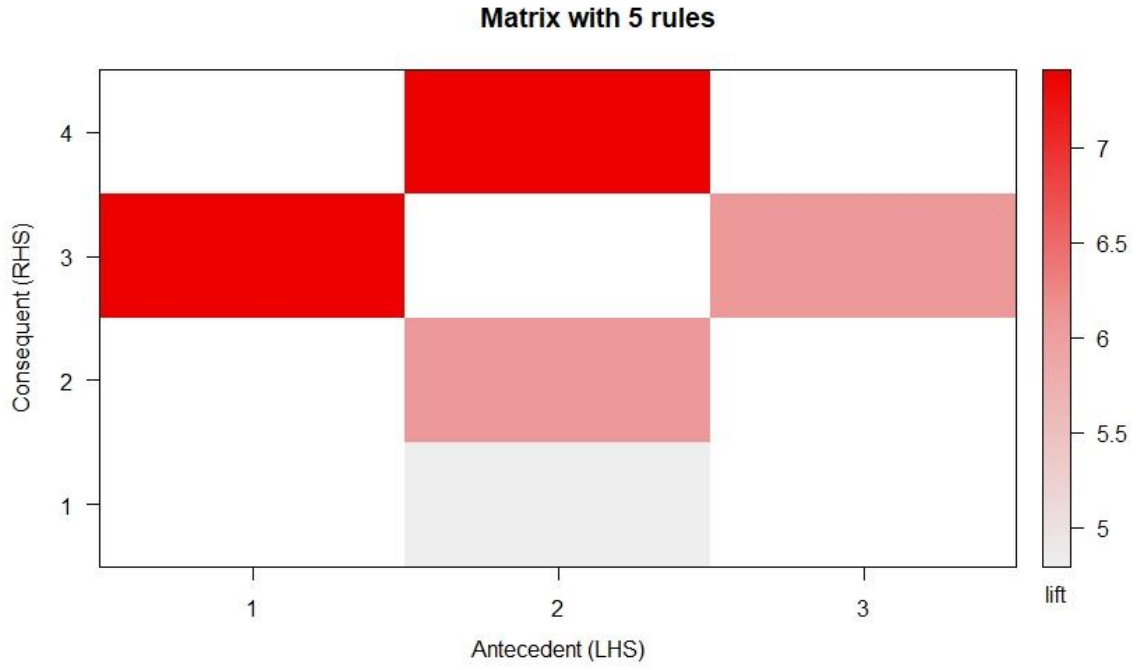
inspect(rule[1:5])

```
    lhs                rhs                support     confidence coverage   lift      count
[1]  {V5= }          => {V6=}           0.034022682 1.00000000 0.03402268 29.392157 51
[2]  {V6=}           => {V5= }          0.034022682 1.00000000 0.03402268 29.392157 51
[3]  {V5= vegetables} => {V4= vegetables} 0.009339560 0.12389381 0.07538359  1.326549 14
[4]  {V4= vegetables} => {V5= vegetables} 0.009339560 0.10000000 0.09339560  1.326549 14
[5]  {V3= poultry}    => {V4= vegetables} 0.007338225 0.27500000 0.02668446  2.944464 11
[6]  {V4= vegetables} => {V3= poultry}    0.007338225 0.07857143 0.09339560  2.944464 11
[7]  {V6= vegetables} => {V4= vegetables} 0.007338225 0.11340206 0.06470981  1.214212 11
[8]  {V4= vegetables} => {V6= vegetables} 0.007338225 0.07857143 0.09339560  1.214212 11
[9]  {V4= waffles}    => {V6= vegetables} 0.006671114 0.22222222 0.03002001  3.434135 10
[10] {V6= vegetables} => {V4= waffles}    0.006671114 0.10309278 0.06470981  3.434135 10
```

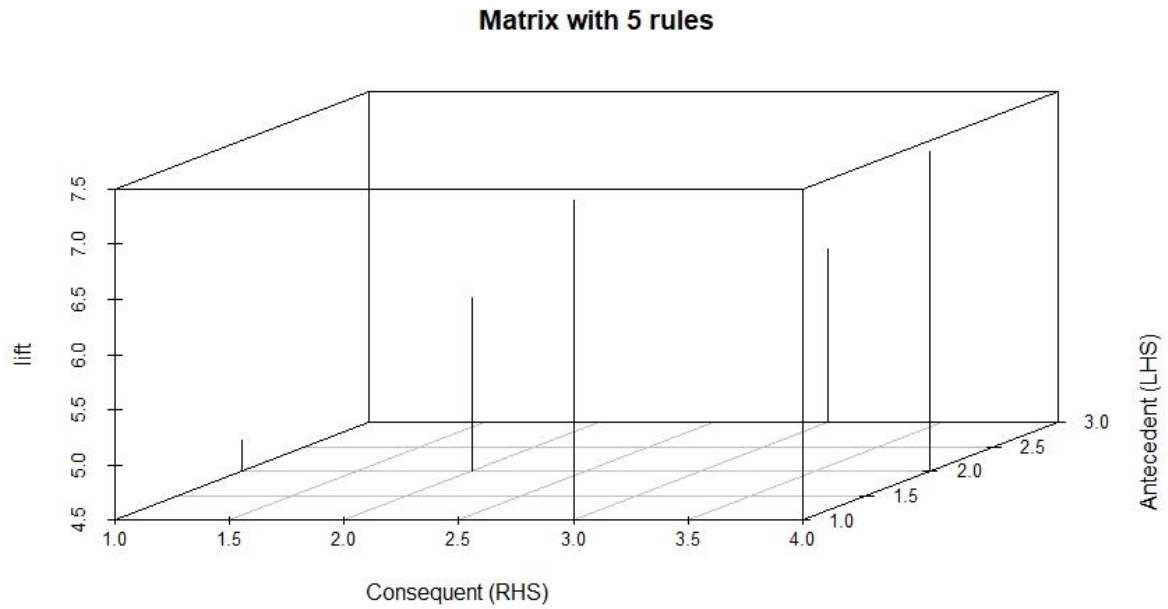
plot(yontem, measure=c("confidence", "lift"), shading="support")



plot(yontem[1:5], method="matrix", measure="lift")

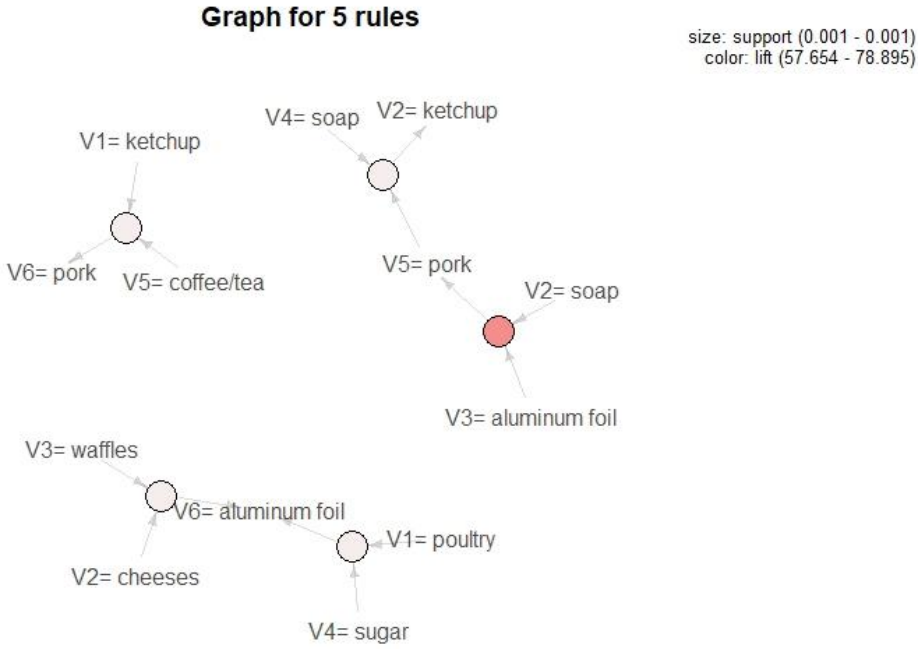


plot(yontem[1:5], method="matrix3D", measure="lift")

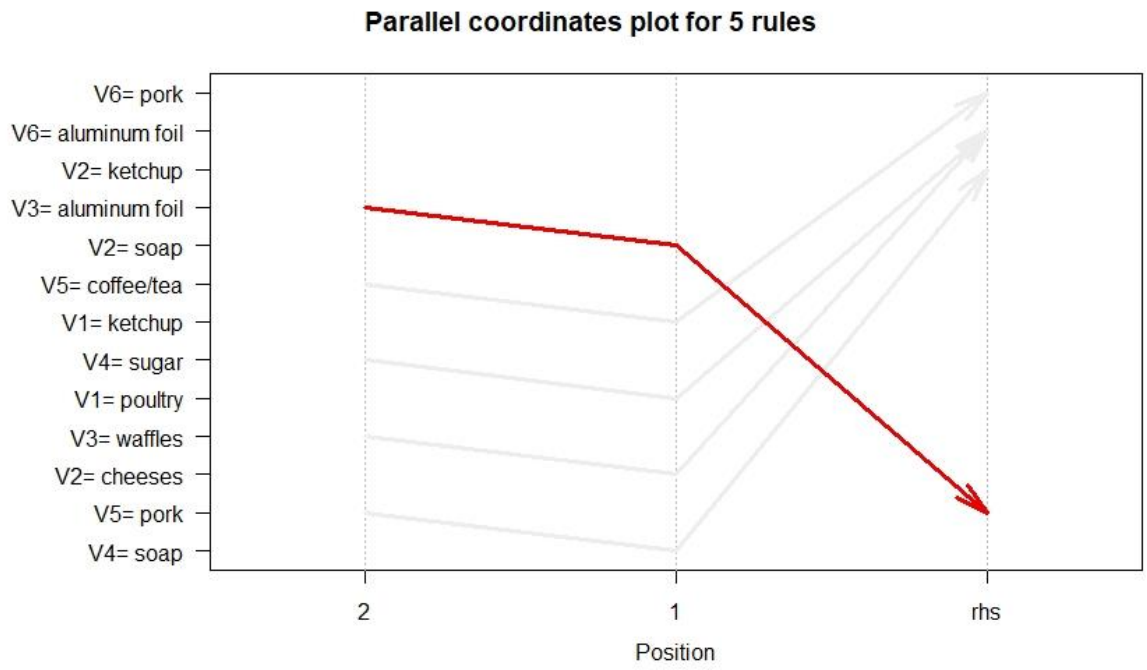


satylan <- head(sort(yontem, by="lift"), 5)

plot(satylan, method="graph")



plot(satylan, method="paracoord")



Conclusions

In the result you can see (last graph) that list of the products. İt is very useful method to analysis such dataset.

Reference

"What people purchase | Kaggle " <-databases if in following link.