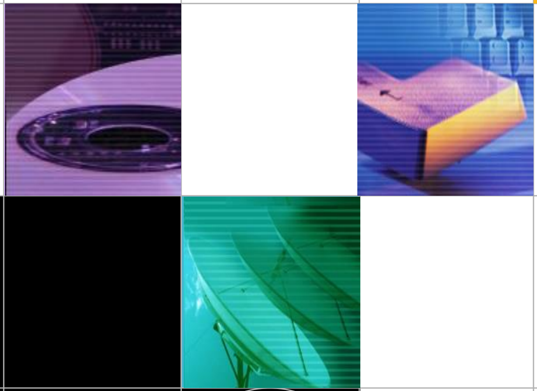
Machine learning



Chapter 9

Association Rule

Sejong Oh

Bio Information Technology Lab.

Contents

- Summary
- Apriori algorithm
- Exercise

summary

- Example of association rule
  - Item set

```
{peanut butter, jelly, candy, bread}
```

  - Association rule

```
{peanut butter, jelly} ⇒ {bread}
```

    - 땅콩버터와 젤리를 구매하면 빵도 함께 구매한다

  - 이러한 규칙을 찾아내는 것이 목표
  - Unsupervised learning

summary

- Application of association rule
  - 암 데이터 분석에서 단백질 서열과 자주 발견되는 흥미로운 DNA 패턴 찾기
  - 구매패턴, 사기 신용카드나 보험과 복합해 발생하는 의료청구 발견
  - 휴대폰 서비스를 정지하거나 케이블 TV 패키지를 업그레이드 하려는 행위의 사전 조합 식별하기

summary

- Problem in finding association rule
  - There exist too many candidate item sets !
  - If you sell 10 items, you should test 2^{10} item sets.

`{peanut butter, jelly} ⇒ {bread}`

이런 규칙을 찾으려면 다음과 같은 후보 item set 을 식별해야 한다

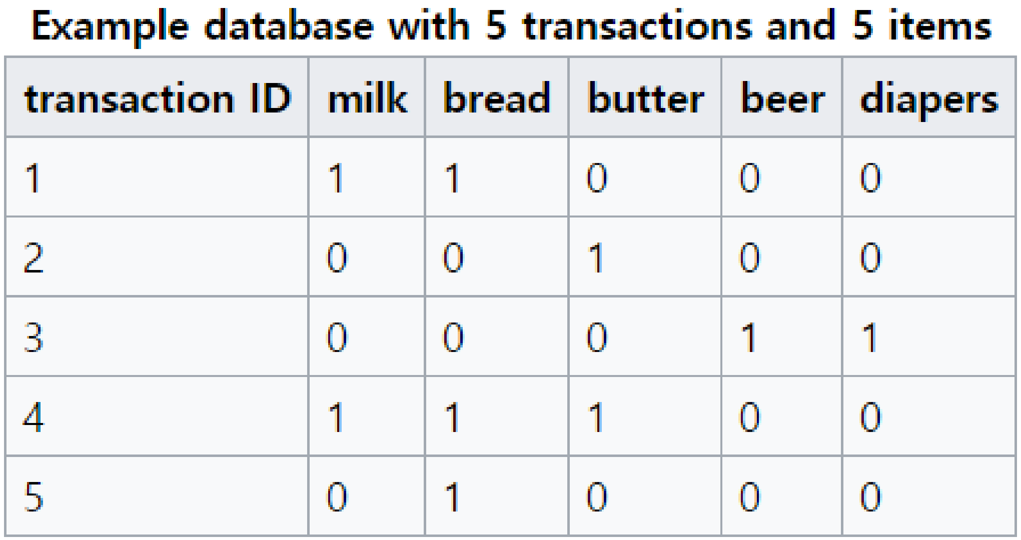
`{peanut butter, jelly, bread}`

그런데 이런 후보 item set 의 수는 너무 많다.

- association rule 의 문제는 테스트해보아야 하는 후보 item set 의 수를 줄이는 것

Apriori algorithm

- Apriori uses a breadth-first search strategy to count the support of item sets and uses a candidate generation function which exploits the downward closure property of support.



Example database with 5 transactions and 5 items

transaction ID	milk	bread	butter	beer	diapers
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

(https://en.wikipedia.org/wiki/Association_rule_learning)

Apriori algorithm

- Let
  - ***X*** be an item set,
  - ***X*** $\Rightarrow$ ***Y*** an association rule and
  - ***T*** a set of transactions of a given database.

- Support (지지도)

$$\mathrm{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|} = \frac{\text{freq. of } \boldsymbol{X}}{\#\text{ of } \boldsymbol{T}}$$

ex) ***X*** = {milk, bread}

$$\mathrm{supp}(X) = \frac{2}{5} = 0.4$$

Apriori algorithm

- Confidence (신뢰도)

$$\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X)\,.$$

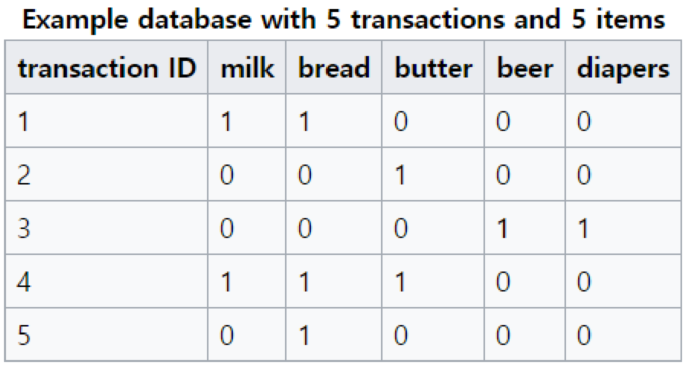
ex) $\mathrm{conf}(\{\mathrm{milk}\} \Rightarrow \{\mathrm{bread}\}) = \dfrac{2}{2} = 1$

$\mathrm{conf}(\{\mathrm{bread}\} \Rightarrow \{\mathrm{milk}\}) = \dfrac{2}{3} = 0.66$

Support : 검토해야할 item set 의 수를 줄이는데 사용
Confidence : association 의 정도를 파악하는데 사용

Apriori algorithm

- Step 1. Select candidate item sets
  - Remove item set if its **supp** is small



Example database with 5 transactions and 5 items

transaction ID	milk	bread	butter	beer	diapers
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

Round 1

supp({milk}) = 2/5 = 0.4

supp({bread}) = 3/5 = 0.6

supp({butter}) = 2/5 = 0.4

supp({beer}) = 1/5 = 0.2 remove

supp({diapers}) = 1/5 = 0.2 remove

Round 2

supp({milk, bread}) = 2/5 = 0.4

supp({milk, butter}) = 1/5 = 0.2 remove

supp({bread,butter}) = 1/5 = 0.2 remove

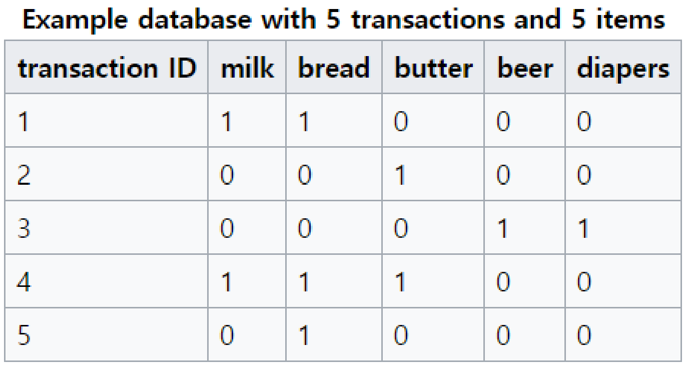
stop

there is no {milk, bread} will be tested in the next step

Apriori algorithm

- Step 2. Test candidate association
  - Remove association rule if its **conf** is small

$$\text{conf}(\{\text{milk}\} \Rightarrow \{\text{bread}\}) = 2/2 = 1$$
$$\text{conf}(\{\text{bread}\} \Rightarrow \{\text{milk}\}) = 2/3 = 0.6$$



Example database with 5 transactions and 5 items

transaction ID	milk	bread	butter	beer	diapers
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

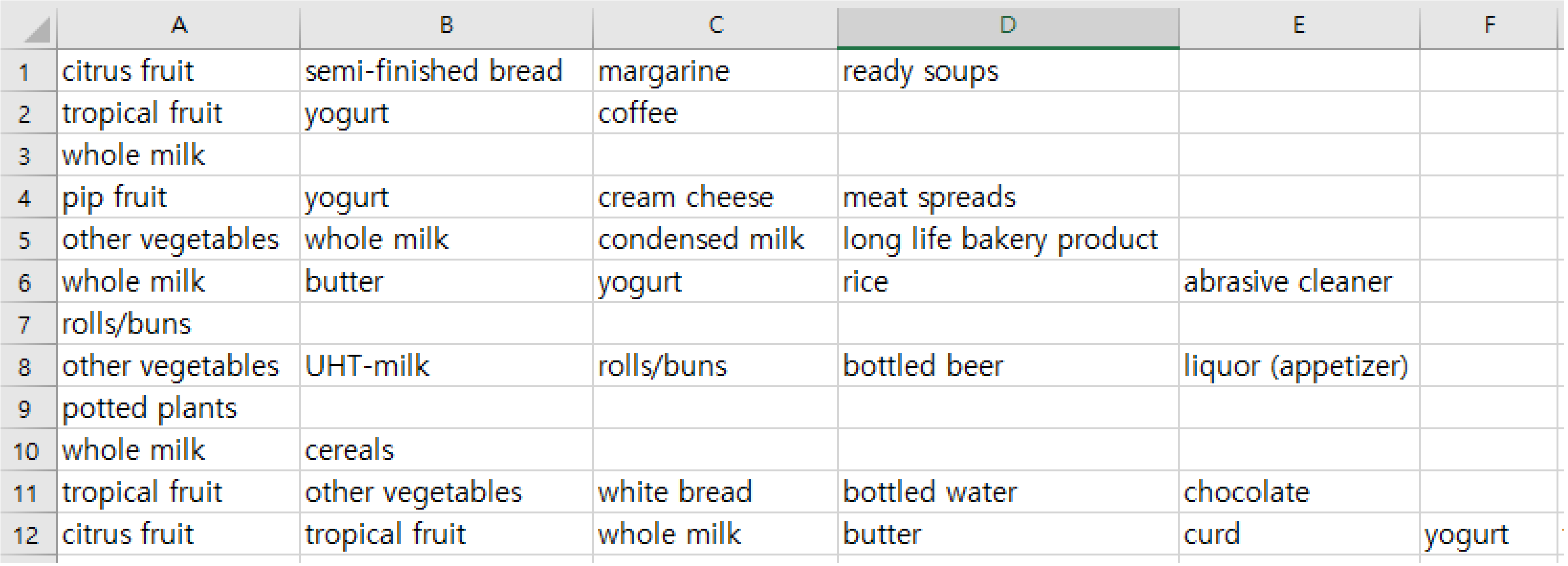
We obtain two rules:

$$\{\text{milk}\} \Rightarrow \{\text{bread}\}$$
$$\{\text{bread}\} \Rightarrow \{\text{milk}\}$$

[Exercise]

- 1단계: dataset 준비

groceries.csv

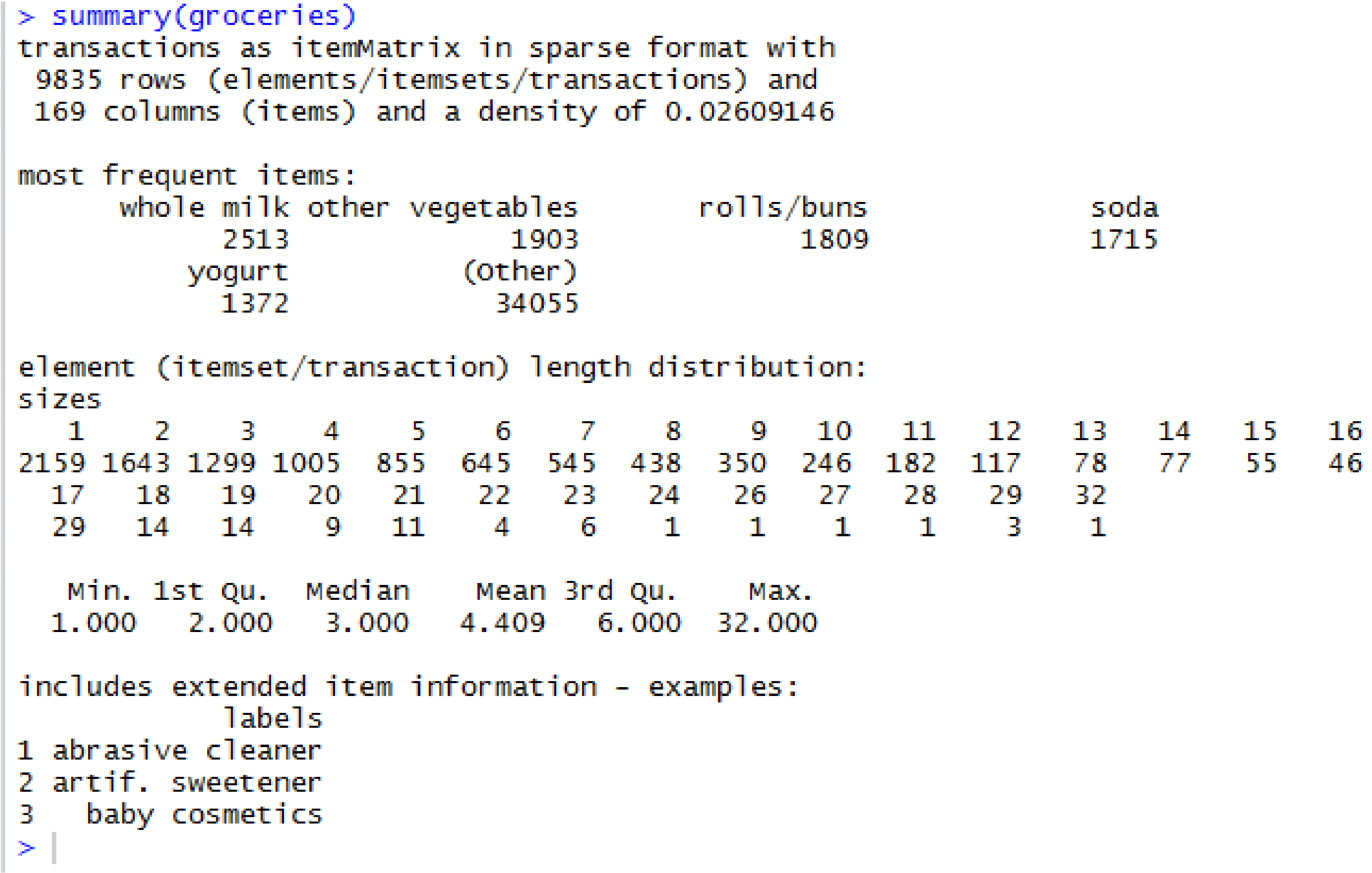


	A	B	C	D	E	F
1	citrus fruit	semi-finished bread	margarine	ready soups		
2	tropical fruit	yogurt	coffee			
3	whole milk					
4	pip fruit	yogurt	cream cheese	meat spreads		
5	other vegetables	whole milk	condensed milk	long life bakery product		
6	whole milk	butter	yogurt	rice	abrasive cleaner	
7	rolls/buns					
8	other vegetables	UHT-milk	rolls/buns	bottled beer	liquor (appetizer)	
9	potted plants					
10	whole milk	cereals				
11	tropical fruit	other vegetables	white bread	bottled water	chocolate	
12	citrus fruit	tropical fruit	whole milk	butter	curd	yogurt

[Exercise]

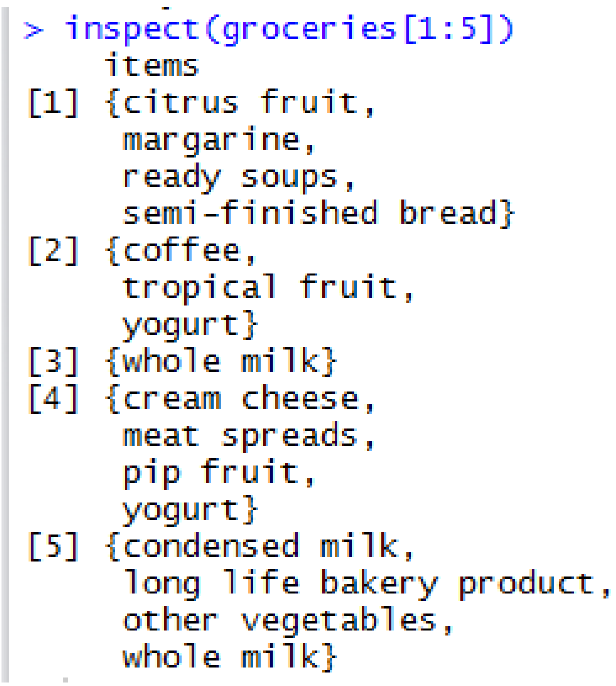
```
## 2단계: 데이터 준비와 살펴보기 ----
# 식료품 데이터를 희소 매트릭스로 로드
library(arules)
setwd("C:/chapter 8")
groceries <- read.transactions("groceries.csv", sep = ",")
summary(groceries)
# 처음 5개 거래 확인
inspect(groceries[1:5])
# 3개 물품 빈도 확인
itemFrequency(groceries[, 1:3])
```

[Exercise]

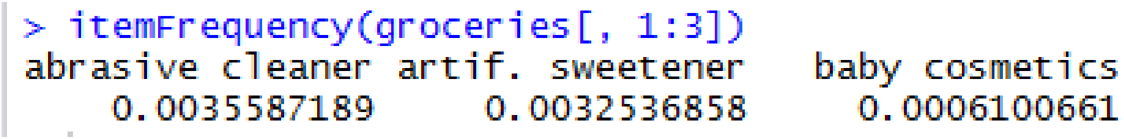


```
> summary(groceries)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns             soda
            2513             1903             1809             1715
          yogurt          (Other)
            1372            34055

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46
  17   18   19   20   21   22   23   24   26   27   28   29   32
  29   14   14    9   11    4    6    1    1    1    1    3    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   4.409   6.000  32.000

includes extended item information - examples:
            labels
1 abrasive cleaner
2 artif. sweetener
3   baby cosmetics
> 
```

[Exercise]



```
> inspect(groceries[1:5])
    items
[1] {citrus fruit,
     margarine,
     ready soups,
     semi-finished bread}
[2] {coffee,
     tropical fruit,
     yogurt}
[3] {whole milk}
[4] {cream cheese,
     meat spreads,
     pip fruit,
     yogurt}
[5] {condensed milk,
     long life bakery product,
     other vegetables,
     whole milk}
```

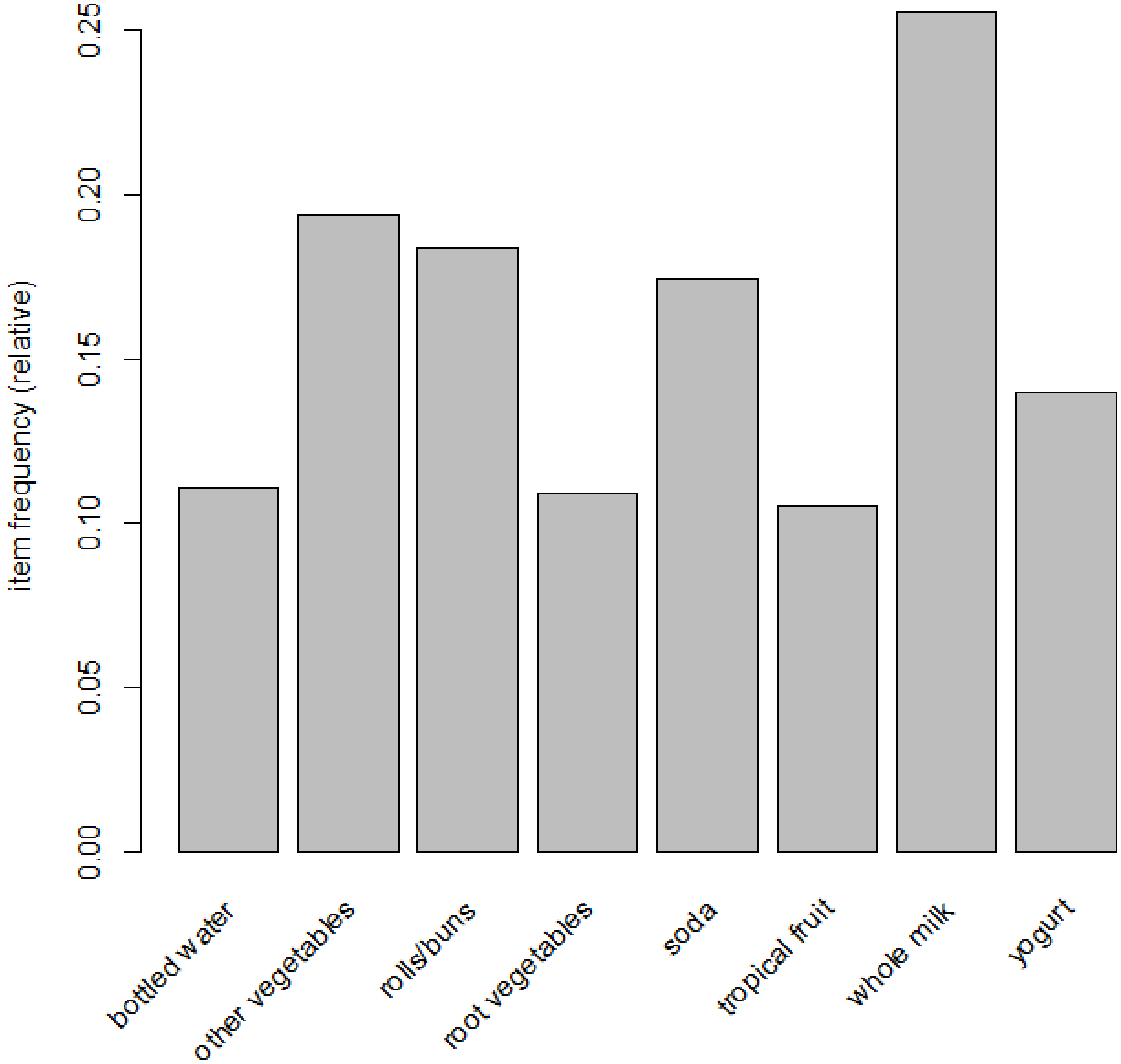


```
> itemFrequency(groceries[, 1:3])
abrasive cleaner artif. sweetener   baby cosmetics
    0.0035587189     0.0032536858     0.0006100661
```

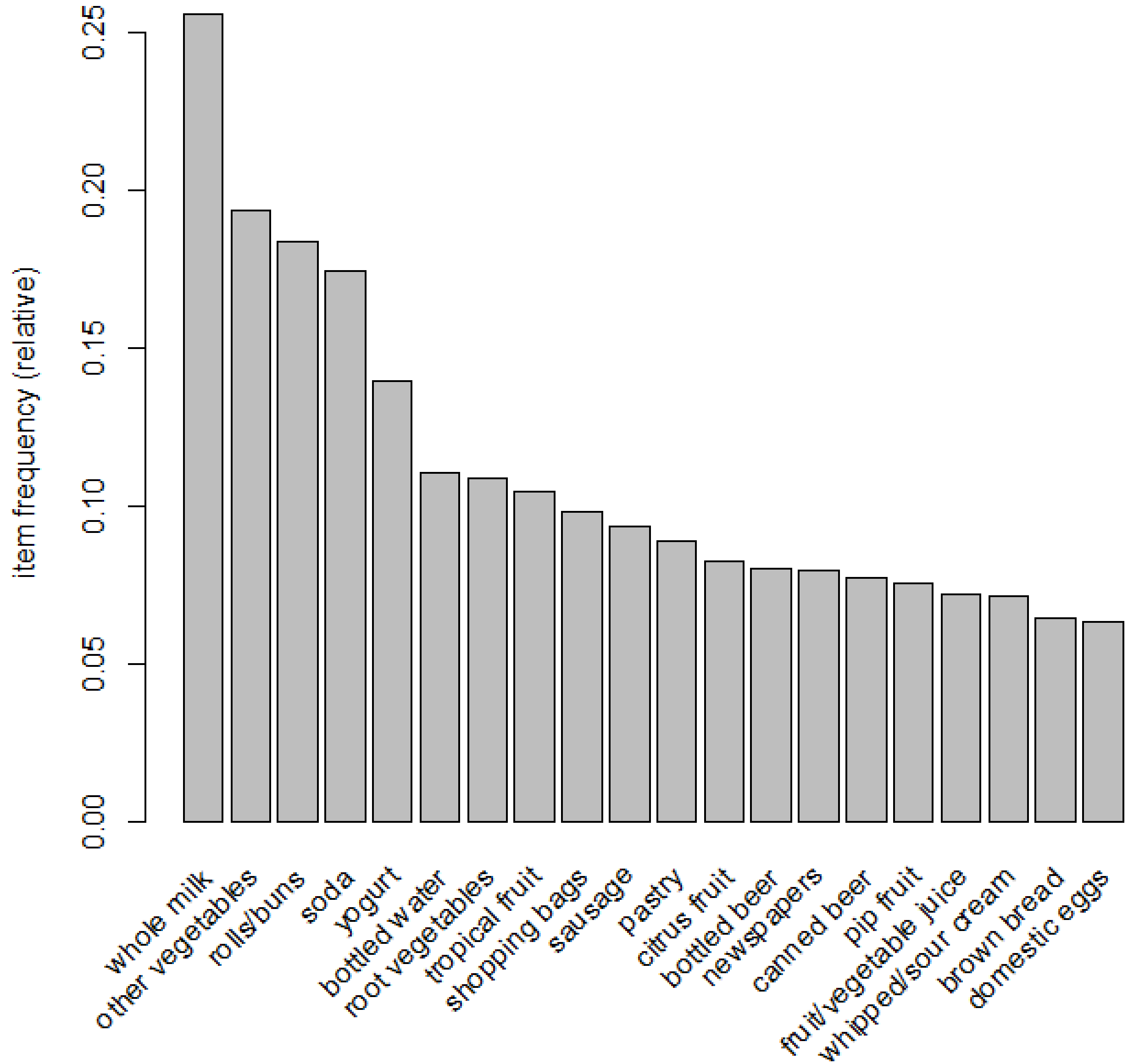
[Exercise]

```
# 식료품의 빈도 시각화
itemFrequencyPlot(groceries, support = 0.1)
itemFrequencyPlot(groceries, topN = 20)

# 처음 5개 거래에 대한 희소 매트릭스 시각화
image(groceries[1:5])

# 100개 식료품의 무작위 샘플 시각화
image(sample(groceries, 100))
```

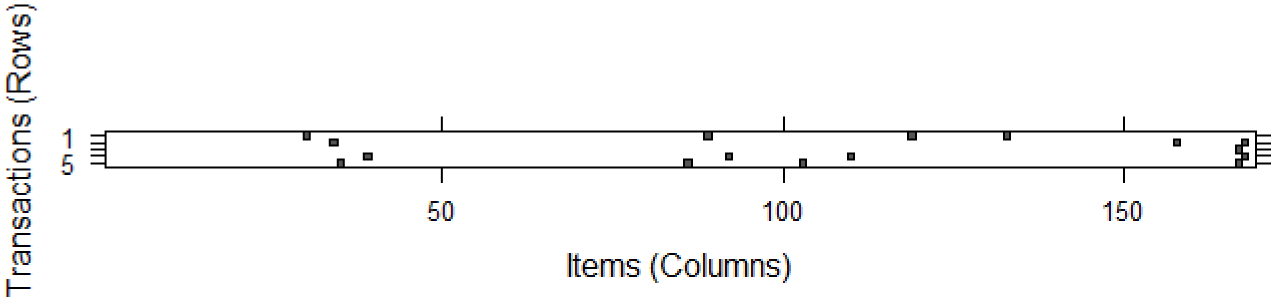
```
> itemFrequencyPlot(groceries, support = 0.1)
```



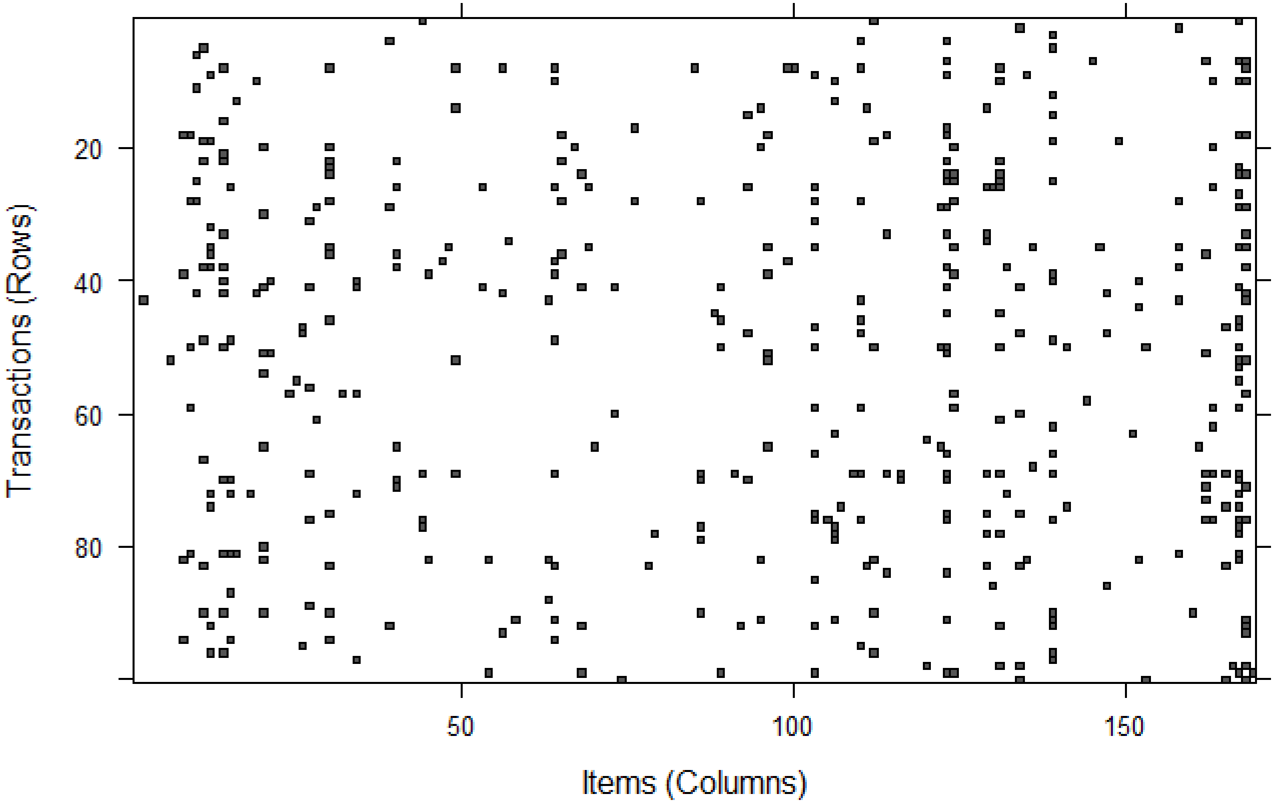
```
> itemFrequencyPlot(groceries, topN = 20)
```



```
> # 처음 5개 거래에 대한 희소 매트릭스 시각화
> image(groceries[1:5])
```

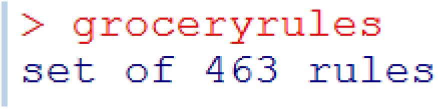


```
> # 100개 식료품의 무작위 샘플 시각화
> image(sample(groceries, 100))
```



[Exercise]

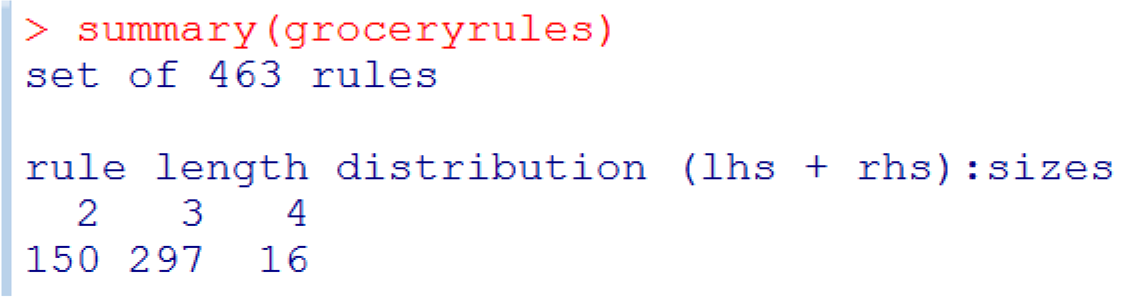
```
## 3단계 : 데이터에 대한 모델 훈련 ----
# 기본 설정
apriori(groceries)
# 규칙을 좀 더 학습히기 위해 지지도(support)와 신뢰도
# (confidence) 설정 변경
groceryrules <- apriori(groceries, parameter =
        list(support = 0.006, confidence = 0.25,
        minlen = 2))    # itemset 의 최소 item 수
groceryrules
```



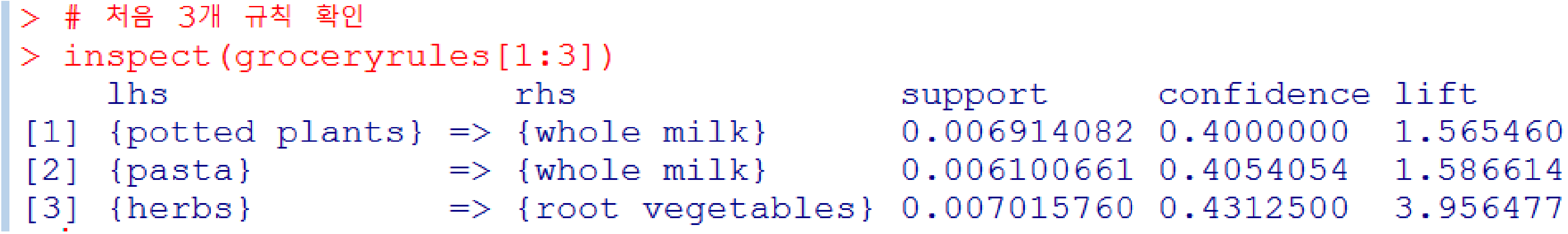
```
> groceryrules
set of 463 rules
```

[Exercise]

```
## 4단계 : 모델 성능 평가 ----
# 식료품 연관 규칙의 요약
summary(groceryrules)

# 처음 3개 규칙 확인
inspect(groceryrules[1:3])
```

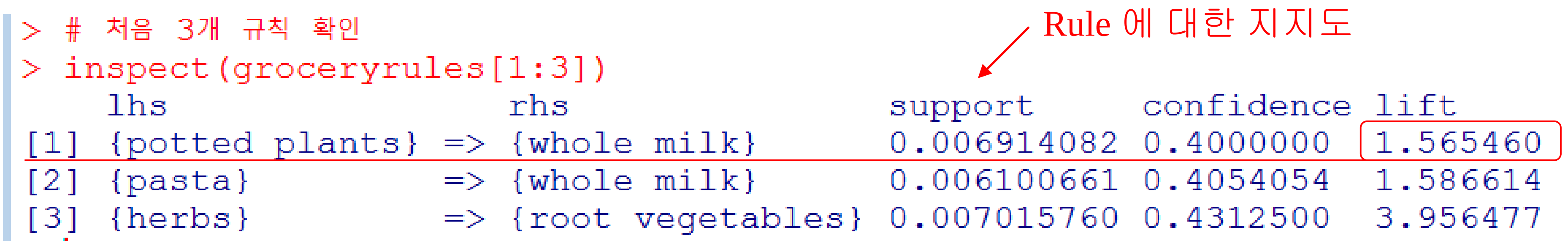


```
> summary(groceryrules)
set of 463 rules

rule length distribution (lhs + rhs):sizes
  2   3   4
150 297  16
```



```
> # 처음 3개 규칙 확인
> inspect(groceryrules[1:3])
    lhs                  rhs                support     confidence lift
[1] {potted plants} => {whole milk}       0.006914082 0.4000000  1.565460
[2] {pasta}         => {whole milk}       0.006100661 0.4054054  1.586614
[3] {herbs}         => {root vegetables} 0.007015760 0.4312500  3.956477
```

lift



```
> # 처음 3개 규칙 확인
> inspect(groceryrules[1:3])
    lhs                rhs                  support     confidence lift
[1] {potted plants} => {whole milk}       0.006914082 0.4000000  1.565460
[2] {pasta}         => {whole milk}       0.006100661 0.4054054  1.586614
[3] {herbs}         => {root vegetables}  0.007015760 0.4312500  3.956477
```

Rule 에 대한 지지도

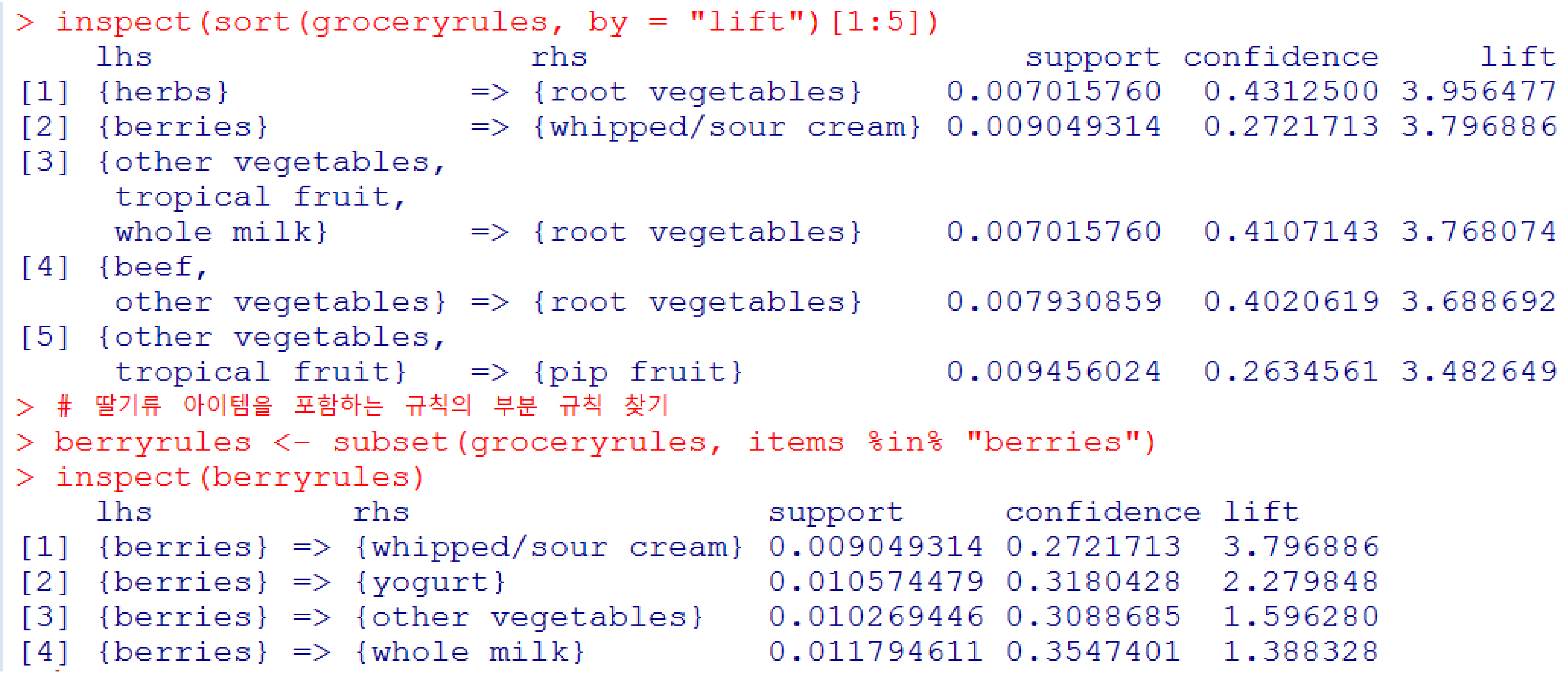
- whole milk 를 구매한 평균 소비자와 비교해 볼 때 potted plants 를 함께 구매한 소비자가 얼마나 더 있는지의 비율

- lift(X→Y) = confidence(X→Y)/support(Y)
  (X가 주어졌을 때 Y의 구매 확률) / (X가 주어지지 않았을 때의 Y의 구매 확률)
- lift값이 1 보다 크면 우연히 X,Y를 함께 구매했을 확률보다 크다는 의미
- lift 값이 크면 클수록 X,Y 의 구매 연관성이 높다

[Exercise]

```
## 5단계 : 모델 성능 향상 ----
# lift로 규칙 정렬
inspect(sort(groceryrules, by = "lift")[1:5])

# 딸기류 아이템을 포함하는 규칙의 부분 규칙 찾기
berryrules <- subset(groceryrules, items %in% "berries")
inspect(berryrules)
```

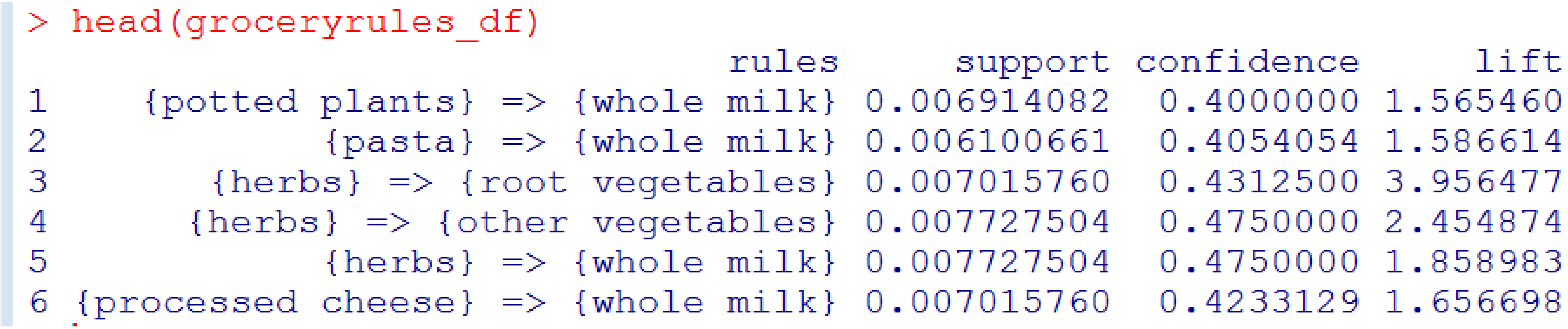
[Exercise]



```
> inspect(sort(groceryrules, by = "lift")[1:5])
    lhs                     rhs                      support confidence     lift
[1] {herbs}              => {root vegetables}    0.007015760  0.4312500 3.956477
[2] {berries}            => {whipped/sour cream} 0.009049314  0.2721713 3.796886
[3] {other vegetables,
     tropical fruit,
     whole milk}         => {root vegetables}    0.007015760  0.4107143 3.768074
[4] {beef,
     other vegetables}   => {root vegetables}    0.007930859  0.4020619 3.688692
[5] {other vegetables,
     tropical fruit}     => {pip fruit}          0.009456024  0.2634561 3.482649
> # 딸기류 아이템을 포함하는 규칙의 부분 규칙 찾기
> berryrules <- subset(groceryrules, items %in% "berries")
> inspect(berryrules)
    lhs          rhs                  support     confidence lift
[1] {berries} => {whipped/sour cream} 0.009049314 0.2721713  3.796886
[2] {berries} => {yogurt}             0.010574479 0.3180428  2.279848
[3] {berries} => {other vegetables}   0.010269446 0.3088685  1.596280
[4] {berries} => {whole milk}         0.011794611 0.3547401  1.388328
```

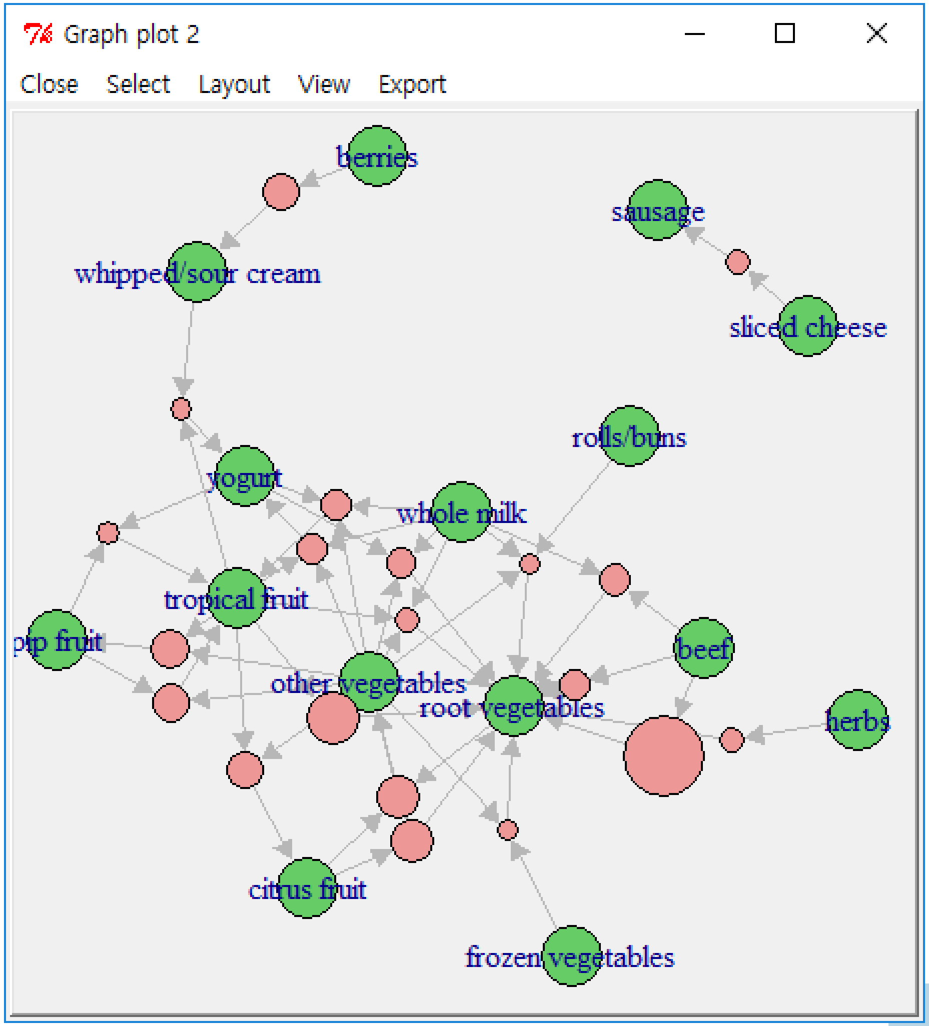
[Exercise]

```
# csv 파일에 규칙 쓰기
write(groceryrules, file = "groceryrules.csv",
      sep = ",", quote = TRUE, row.names = FALSE)

# 규칙들을 데이터 프레임으로 변환
groceryrules_df <- as(groceryrules, "data.frame")
str(groceryrules_df)
head(groceryrules_df)
```



```
> head(groceryrules_df)
                                 rules     support confidence     lift
1     {potted plants} => {whole milk} 0.006914082  0.4000000 1.565460
2             {pasta} => {whole milk} 0.006100661  0.4054054 1.586614
3        {herbs} => {root vegetables} 0.007015760  0.4312500 3.956477
4       {herbs} => {other vegetables} 0.007727504  0.4750000 2.454874
5             {herbs} => {whole milk} 0.007727504  0.4750000 1.858983
6 {processed cheese} => {whole milk} 0.007015760  0.4233129 1.656698
```

```
# extra visualization
library(arulesViz)
plot(sort(groceryrules, by = "lift")[1:20],
     method="graph", interactive=TRUE, shading=NA)
```



[과제]

- Download another basket dataset form web
- Find best 20 association rules according to “lift” value and visualize them using **`arulesViz`** package.
- Investigate parameters of plot() in **`arulesViz`**, and draw extra plots for best 20 association rules.

추천 article

- 고객의 장바구니에는 맥주와 기저귀가 함께 있다! - 데이터 마이닝과 장바구니 분석
  - 출처: http://blog.lgcns.com/511