연관 규칙을 이용한

장바구니 분석

4 & 5조

김서희 김장미 우현우

목차

연관 규칙 & 알고리즘

1. 연관 규칙이란?

- **장바구니 분석(Market Basket Analysis)**으로 널리 알려져 있는 방법론
- 어떤 두 아이템 집합이 빈번히 발생하는가를 알려주는 일련의 규칙들을 생성하는 알고리즘
- 구매 이력 데이터를 토대로 "**X 아이템을 구매하는 고객은 Y 아이템 역시 구매할 가능성이 높다**"의 결론을 도출함

* 아이템 (item) : 기본 단위

* 아이템 집합 (item set) : 아이템의 그룹, 대괄호로 표현함

item item item

v.s.

item set

어떠한 거래에도 나타날 수 있는 장바구니 분석의 기본 단위

아이템 집합은 규칙성을 갖고 데이터에 나타남

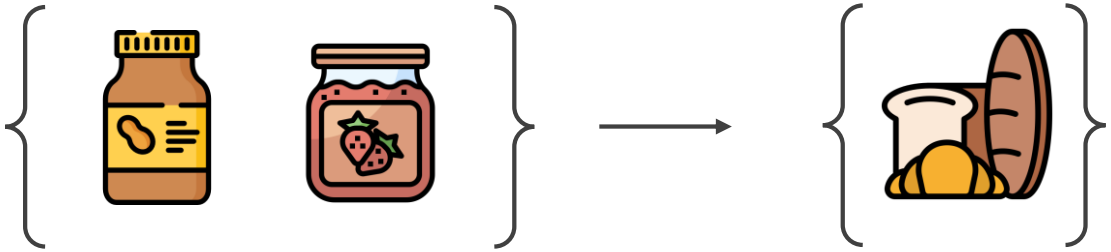
1. 연관 규칙이란?

- 연관 규칙은 **아이템의 부분집합**으로 구성되며, 아이템 집합 사이의 관계에 존재하는 패턴을 명시함

* 연관 규칙 생성 전 : {땅콩 버터, 잼, 빵}

item set

* 연관 규칙 생성 후 : {땅콩 버터, 잼} -> {빵}



규칙의 좌측 = LHS
(Left-hand side)

규칙의 실행을 위해
만족돼야 하는 조건절

규칙의 우측 = RHS
(Right-hand side)

조건을 만족했을 때
기대되는 결과절

조건절의 아이템과 결과절의 아이템은 상호배반 !

- ➢ 땅콩 버터와 잼을 구매하는 고객은 '빵'을 구매할 가능성이 높다
- ➢ 땅콩 버터와 잼은 '빵'을 암시한다

1. 연관 규칙이란?

- 연관 규칙의 장단점

장점	단점
1. 조건 반응으로 표현되는 연관성 분석의 결과를 쉽게 이해 2. 비목적성 분석 기법이기 때문에 목적 변수가 없고 데이터의 형태 변환도 필요 없음	1. 품목수가 증가하면 계산이 기하급수적으로 늘어남 2. 너무 세분화한 품목을 가지고 연관 규칙을 찾으면 의미 없는 분석이 될 수 있다 *--> 따라서, 유사한 품목을 한 범주로 일반화하여, 큰 범주를 전체 분석에 포함시킨 후 결과중에서 세부적인 연관 규칙을 찾음*

1. 연관 규칙이란?

- 좋은 규칙은 어떻게 구할까? 지지도, 신뢰도, 향상도
- 조건절을 A, 결과절을 B 라고 할 때,

** 떠올려봅시다!*
연관 규칙 {땅콩 버터, 잼} -> {빵}이 있다고 하면,
왼쪽 규칙을 **조건절**, 오른쪽 규칙을 **결과절**이라고 해요 ☺

지지도 Support

$$\frac{A\text{와 }B\text{를 동시에 포함하는 거래수}}{\text{전체 거래수}} = P(A \cap B)$$

- **빈발 아이템 집합**을 판별
- 아이템 하나에 대한 지지도는 등장 확률과 같음

신뢰도 Confidence

$$\frac{A\text{와 }B\text{를 동시에 포함하는 거래수}}{A\text{를 포함하는 거래수}} = \frac{P(A \cap B)}{P(A)} = P(B|A)$$

- **아이템 집합 간의 연관성 강도** 측정
- 조건절이 주어졌을 때, 결과절이 일어날 조건부 확률

향상도 Lift

$$\frac{A\text{와 }B\text{를 동시에 포함하는 거래수} \times \text{전체거래수}}{A\text{ 포함 거래수} \times B\text{포함 거래수}} = \frac{P(A \cap B)}{P(A) \times P(B)} = \frac{P(B|A)}{P(B)}$$

- **생성된 규칙이 실제 효용가치가 있는지** 판별
- 두 사건이 동시에 얼마나 발생하는지의 비율

1. 연관 규칙이란?

- 지지도, 신뢰도, 향상도의 특징

지지도 Support

☺ 가장 단순하면서, 가장 기본적인 기준
☹ 표본수가 적은 경우 통계적 유의성을 증명하기 어려움
☹ {A}->{B}와 {B}->{A}의 차이를 알 수 없으므로 다른 평가지표가 필요함

신뢰도 Confidence

☺ 지지도가 기준 사건이 전체 집합인데 비하여, 신뢰도는 기준 사건을 특정 품목 집합에 한정
☹ 하지만 {빵}->{우유}의 신뢰도가 50%일 때, 빵이 없는 장바구니 400개 중
우유가 있는 경우가 200개라면 이 경우에도 신뢰도가 50%라, 또 다른 평가지표가 필요함

향상도 Lift

☺ 신뢰도를 item Y의 발생확률로 나눈 것으로, Y의 확률대비 X가 발생할 때 Y가 발생할 확률
☺ X-Y와 X-Z의 신뢰도가 같을 때, Y의 발생확률이 Z보다 낮다면 X-Y의 향상도가 높음

* 향상도가 1이라면 조건절과 결과절은 서로 독립 (=규칙 사이에 유의미한 연관성 없음)
* 향상도가 2라면 두 사건이 독립이라는 걸 가정했을 때 대비 2배로 긍정적인 연관성

1. 연관 규칙이란?

- 예시를 통해 직접 구해보자!

  예시 : 연관 규칙 {빵} -> {우유}

품목	거래건수
빵	100
우유	100
맥주	100
빵, 우유, 맥주	50
우유, 맥주	200
빵, 우유	250
빵, 맥주	200

* 전체 거래 건수 = 1000

지지도 Support

$$\frac{\text{빵과 우유를 동시에 포함하는 거래수}}{\text{전체 거래수}} = \frac{300}{1000} = 30\%$$

신뢰도 Confidence

$$\frac{\text{빵과 우유를 동시에 포함하는 거래수}}{\text{빵을 포함하는 거래수}} = \frac{\frac{300}{1000}}{\frac{600}{1000}} = 20\%$$

향상도 Lift

$$\frac{\text{빵과 우유를 동시에 포함하는 거래수} \times \text{전체거래수}}{\text{빵 포함 거래수} \times \text{ 우유 포함 거래수}}$$

$$= \frac{\frac{300}{1000}}{\frac{600}{1000} \times \frac{600}{1000}} = 83\%$$

세 가지 규칙 효용 지표를 배웠다! 그럼 이제...

모든 규칙의 경우의 수를 탐색하여, 지지도, 신뢰도, 향상도가 높은 규칙을 찾아내면 될까?

NO!

모든 아이템의 잠재적인 규칙을 하나씩 확인하는 것은 너무 많은 시간이 소요됨

따라서 아프리오리 알고리즘을 사용하여 흔치 않고 덜 중요한 조합은 무시하고,

빈발 집합만을 고려하여 검색할 아이템 집합의 개수를 줄인다

2. 알고리즘 설명

- A priori = 사전의, 선험적인
- **빈번한 아이템 집합**의 속성에 대해 단순한 사전의(선험적인) 믿음을 이용하여, 연관 규칙의 검색 공간을 축소함

- 아프리오리 속성

  빈번한 아이템 집합의 모든 부분집합도 빈번해야 한다는 것

  즉, {땅콩 버터, 잼} 집합은 {땅콩 버터}, {잼}이 모두 빈번하게 발생해야만 빈번함

  {땅콩 버터}와 {잼}이 빈번하지 않다면, 이 아이템을 포함하는 어떤 집합이든 검색에서 제외될 수 있음

- 장단점

장점	단점
1. 대규모 거래 데이터에 대해 작업 가능	1. 작은 데이터셋에 유용하지 않음
2. 이해하기 쉬운 규칙을 생성	2. 통찰과 상식을 분리하기 어려움
3. DB에서 예상치 못한 지식을 발굴 가능	3. 랜덤 패턴에서 비논리적 결론 도출 가능성

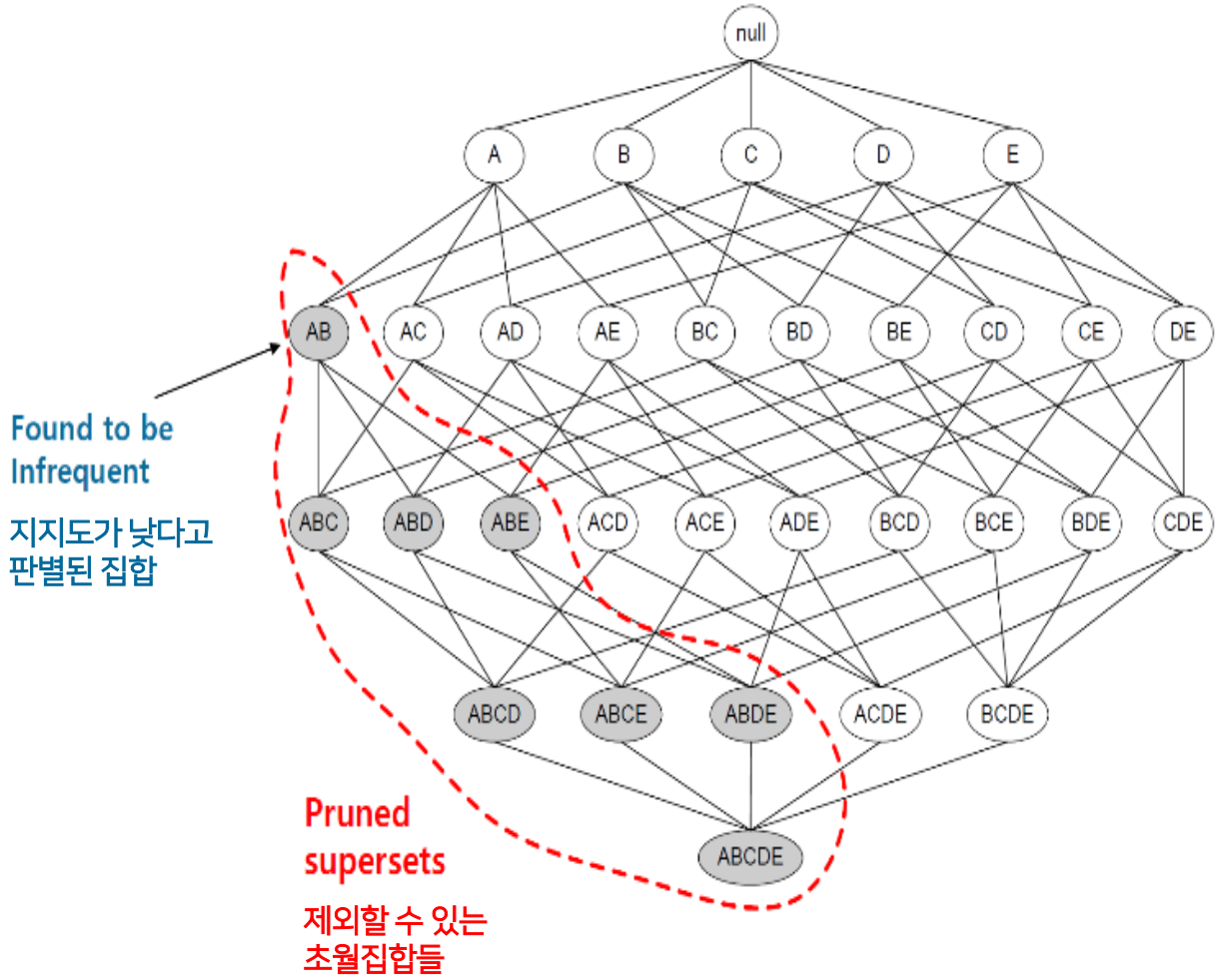
2. 알고리즘 설명

예를 들어 아이템 집합 {A}의 지지도,
즉 P(A)가 0.1이라고 가정하자

A가 단독으로 등장할 확률인 P(A)는,
A와 B가 동시에 나타날 확률인 P(A,B) 보다는 크거나 같으므로,
아이템 집합 {A,B}의 지지도는 아무리 높아도 0.1을 넘지 못함.
* 이 때 {A,B}는 {A}와 {B}의 초월집합이라고 부름

따라서 **{A,B}의 지지도가 낮다**고 판단이 들면,

{A,B}는 물론
{A,B}의 초월집합들까지 검색에서 제외할 수 있음



연관 규칙 선정 예시

3. 연관 규칙 선정 예시

- 간단한 소규모 데이터셋을 활용하여, 어떻게 연관 규칙이 선정되는지 알아보자 :D

* 어느 병원 선물가게 DB

거래 번호	거래 내용 (아이템)
1	꽃, 카드, 소다
2	꽃, 곰인형, 풍선, 캔디바
3	꽃, 카드, 캔디바
4	소다, 곰인형, 풍선
5	꽃, 카드, 소다

첫째, DB를 희소 행렬로 바꾼다

거래 번호	꽃	카드	소다	곰인형	풍선	캔디바
1	1	1	1	0	0	0
2	1	0	0	1	1	1
3	1	1	0	0	0	1
4	0	0	1	1	1	0
5	1	1	1	0	0	0

- ➢ 희소 행렬은 0이 아닌 값이 아주 적다는 뜻, 대부분 0으로 변환
- ➢ 존재하면 1, 존재하지 않으면 0

3. 연관 규칙 선정 예시

둘째, 희소 행렬의 지지도를 구한다

꽃	카드	소다	곰인형	풍선	캔디바
1	1	1	0	0	0
1	0	0	1	1	1
1	1	0	0	0	1
0	0	1	1	1	0
1	1	1	0	0	0

0.8 0.6 0.6 0.4 0.4 0.4

- 지지도는 아이템 등장 횟수 / 총 거래 횟수와 같음

  * 꽃의 지지도 = 4번 등장 / 총 거래 횟수 5번

  = 4/5 = 0.8

  * 카드의 지지도 = 3번 등장 / 총 거래 횟수 5번

  = 3/5 = 0.6

- 최소 지지도 요건을 0.4로 설정하여, 이 숫자를 넘는 아이템만 걸러 낸다 --> 꽃, 카드, 소다

3. 연관 규칙 선정 예시

셋째, 대칭 행렬의 지지도를 구한다

	꽃	카드	소다
꽃			
카드			
소다			

↓

	꽃	카드	소다
꽃	X	0.6	0.4
카드	X	X	0.4
소다	X	X	X

- 최소 지지도를 넘는 요소 - 꽃, 카드, 소다 - 로
  다음과 같이 대칭 행렬을 생성하고 채운다
- 2가지 아이템 집합의 지지도는
  아이템 집합의 (동시) 등장 횟수 / 총 거래 횟수와 같음
  * 꽃과 카드의 지지도 = 3번 / 5번 = 0.6
  * 꽃과 소다의 지지도 = 2번 / 5번 = 0.4
  * 카드와 소다의 지지도 = 2번 / 5번 = 0.4

꽃	카드	소다
1	1	1
1	0	0
1	1	0
0	0	1
1	1	1

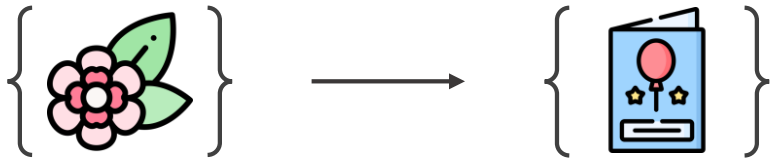
- 대칭 행렬 결과, 최소 지지도 요건 0.4가 넘는
  아이템 집합은 {꽃, 카드} 집합임

3. 연관 규칙 선정 예시

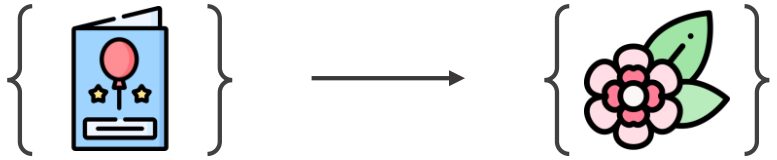
넷째, 규칙 효용성 지표로 평가한다

- 카드를 구매하는 사람은 꽃을 100% 구매한다
- 가장 높은 연관 규칙은 {카드} -> {꽃}

조건절 → 결과절



지지도	신뢰도	향상도
0.6	0.75	1.25
* 지지도는 방금 전에 대칭행렬에서 구한 것	* 0.6/0.8=0.75	* 0.6/0.8*0.6=1.25

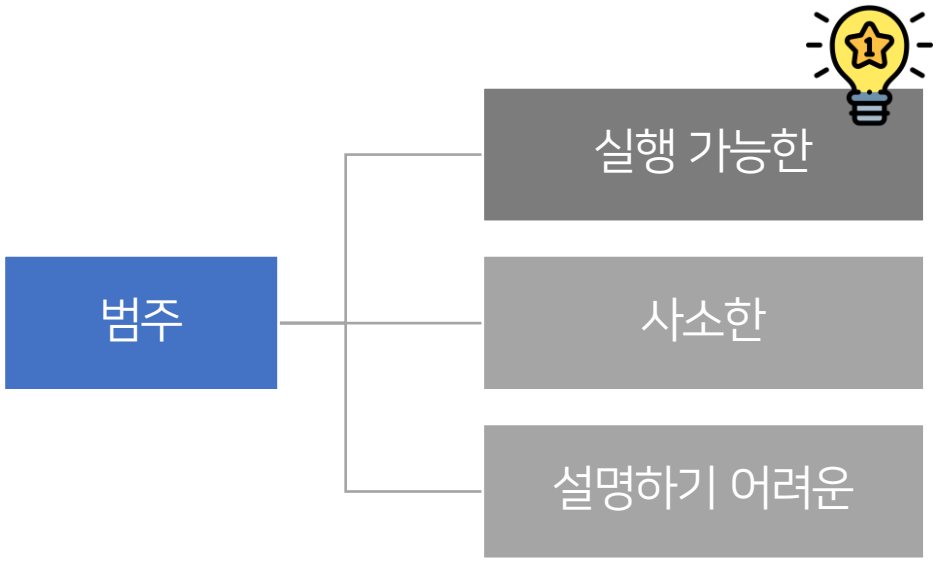


지지도	신뢰도	향상도
0.6	1	1.25
* 지지도는 방금 전에 대칭행렬에서 구한 것	* 0.6/0.6=1 * 높은 신뢰도를 보임	* 0.6/0.8*0.6=1.25 * 1보다 크면 우연이 아님

BUT, 신뢰도와 향상도가 높다고 전부 신뢰할 수 있는 규칙은 아니다 !

3. 연관 규칙 선정 예시

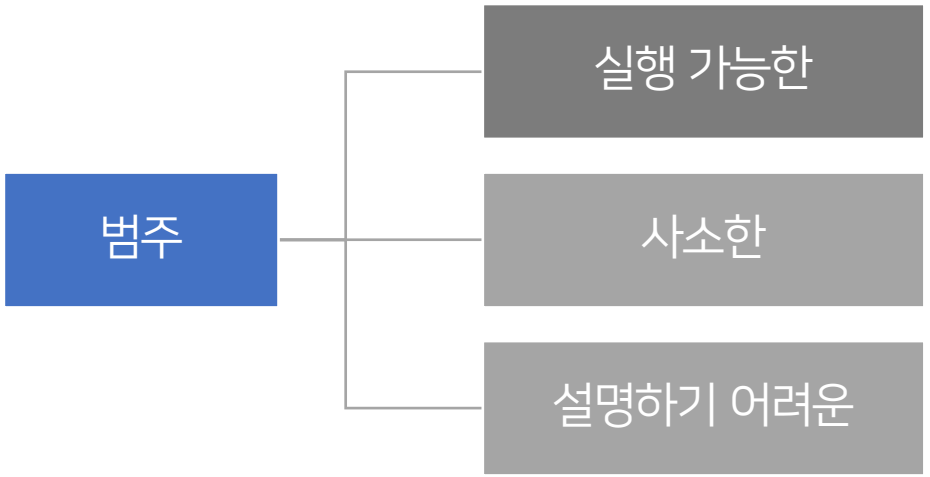
다섯째, 연관 규칙이 어떤 범주에 들어가는지 평가한다



- 명확하고 유용한 통찰을 제공하는 규칙
- 명확하지만 유용하지 않은 규칙 (= 너무나 당연한 것)
  ex. {페인트통} -> {페인트붓}
- 연관성이 불명확해서 설명하기 어려운 규칙
  ex. {피클} -> {아이스크림}

{카드} -> {꽃} 규칙은 실행 가능한 규칙 범주에 들어간다, 따라서 수용 가능 !

3. 연관 규칙 선정 예시



?

- 어떠한 규칙이 실행 가능한 규칙인지,
  설명하기 어려운 규칙인지 판단하는 것이 중요
- 설명하기 어려운 규칙을 실행 가능한 규칙이라고 판단하거나,
  실행 가능한 규칙을 설명하기 어려운 규칙이라고 판단하는 오류를 줄이자!

장바구니 분석 실습

4. 장바구니 분석 실습

- 식료품 매장 거래 데이터 'groceries.csv'를 통해, **어떤 종류의 아이템이 함께 구매되는지** 알아보자
- 'groceries.csv'는 거래 번호 당 거래된 아이템 목록들을 담고 있다

	A	B	C	D	E	F	G	H	I
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	pot 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	flour	bottled wa	dishes
13	beef								
14	frankfurter	rolls/buns	soda						
15	chicken	tropical fruit							
16	butter	sugar	fruit/vegetable jui	newspapers					

* 파일을 엑셀로 열어본 모습
* read.csv()로 읽어면 이대로 읽힘 ☹

--> 희소 행렬로 변환해야 함

4. 장바구니 분석 실습

- 거래 데이터를 read.transactions() 함수를 사용하여 희소 행렬로 변환하여 불러오자 (*arules 패키지)

```
library(arules)
groceries <- read.transactions("groceries.csv", sep=',')    (* 쉼표로 구분된 파일이라서 sep=',')
summary(groceries)
```

arules 패키지 소개

- Adult, Groceries 데이터 셋 제공
- As(), labels(), crossTable() 등
  연관 분석에 필요한 여러 가지 함수를 제공
- 아프리오리 알고리즘을 구현할 수 있다

다른 함수 소개

- 희소 행렬 관련
  - read.transactions() : 희소 행렬 생성
  - inspect() : 희소 행렬의 내용
  - image() : 희소 행렬 시각화
- 아이템 거래 관련
  - itemFrequency() : 아이템 거래 빈도
  - itemFrequencyPlot() : 아이템 거래 시각화

4. 장바구니 분석 실습

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

- sparse format = 희소 행렬 형식이다
- 9835 행 = 거래 건수
- 169 열 = 169개의 아이템 수
- density of 0.026 = 행렬에서 0 이 아닌 셀 비율

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

- 가장 흔히 발견되는 아이템들
- 전유 > 다른 채소류 > 롤/번 > 소다 > 요거트

```
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
```

- sizes = 아이템 수
- 1개의 아이템을 포함하는 거래 2159번
- 2개의 아이템을 포함하는 거래 1643번

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

- 아이템 수의 사분위수, 평균 4.4개의 아이템을 거래함

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

➢ 불러온 데이터 설명

4. 장바구니 분석 실습

- 희소 행렬의 내용을 보려면 inspect(), 아이템 거래 빈도을 보려면 itemFrequency() 함수를 사용한다

```
inspect(groceries[1:3])
itemFrequency(groceries[,1:3])
```

```
> inspect(groceries[1:3])
    items
[1] {citrus fruit,
     margarine,
     ready soups,
     semi-finished bread}
[2] {coffee,
     tropical fruit,
     yogurt}
[3] {whole milk}
```

- 첫 번째 거래 정보는 {감귤류 과일, 마가린, 즉석 수프, 반제품 빵}
- 두 번째 거래 정보는 {커피, 열대 과일, 요거트}
- 세 번째 거래 정보는 {전유}

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

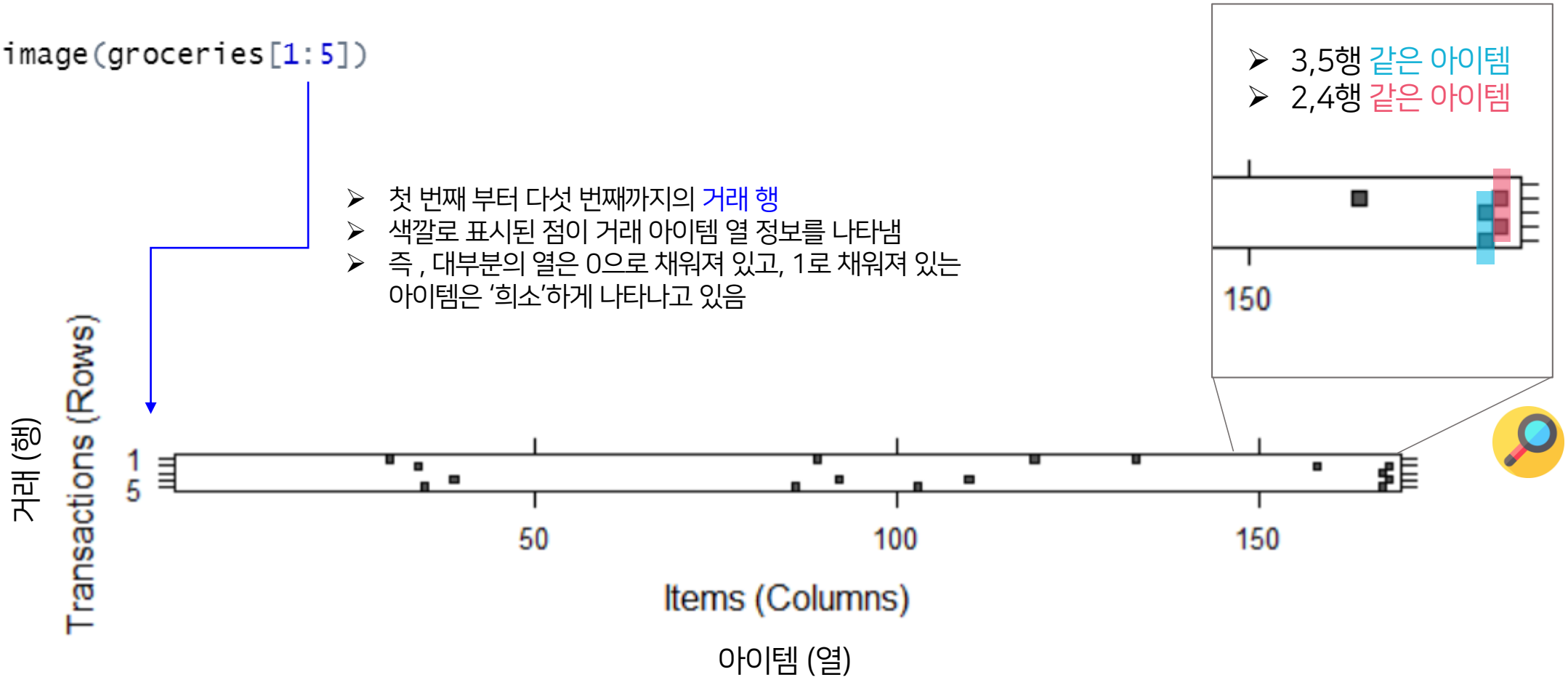
- 데이터의 열은 특정 아이템을 나타냄
- 첫 번째 열인 연마용 청소기는 0.0035% 거래됨
- 두 번째 열인 인공 감미료는 0.0032% 거래됨
- 세 번째 열인 유아용 화장품은 0.0006% 거래됨

4. 장바구니 분석 실습

- 희소 행렬을 시각화 하려면, image() 함수를 사용한다

```
image(groceries[1:5])
```

- 첫 번째 부터 다섯 번째까지의 거래 행
- 색깔로 표시된 점이 거래 아이템 열 정보를 나타냄
- 즉 , 대부분의 열은 0으로 채워져 있고, 1로 채워져 있는 아이템은 '희소'하게 나타나고 있음



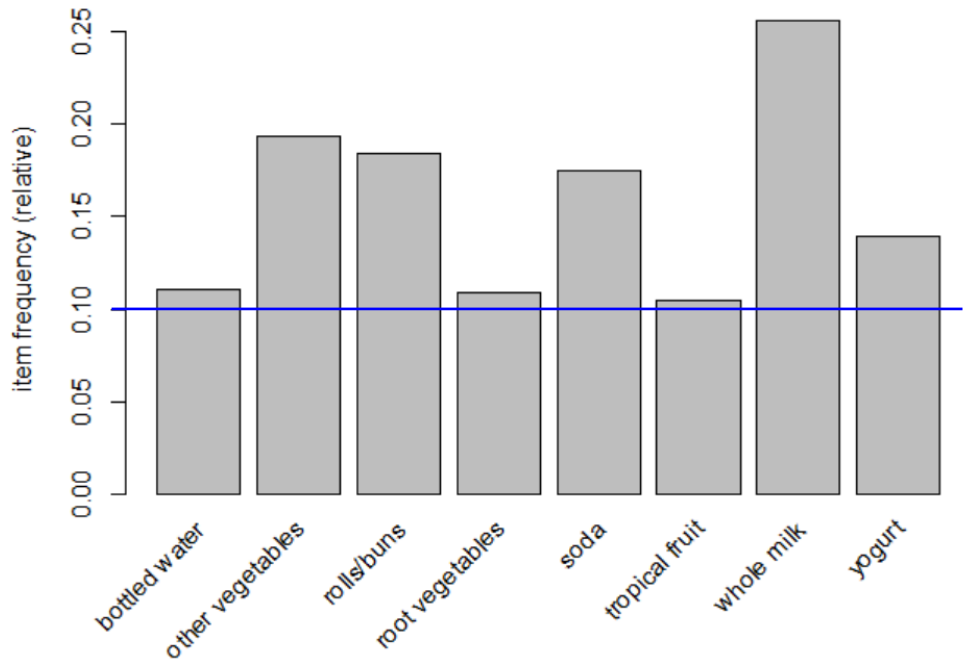
4. 장바구니 분석 실습

- 거래 빈도 바 플랏을 그리려면, itemFrequencyPlot() 함수를 사용한다

  support 옵션으로 최소 지지도를 설정하여 아이템 수를 제한하거나, topN 옵션으로 상위 아이템 개수를 설정할 수 있음

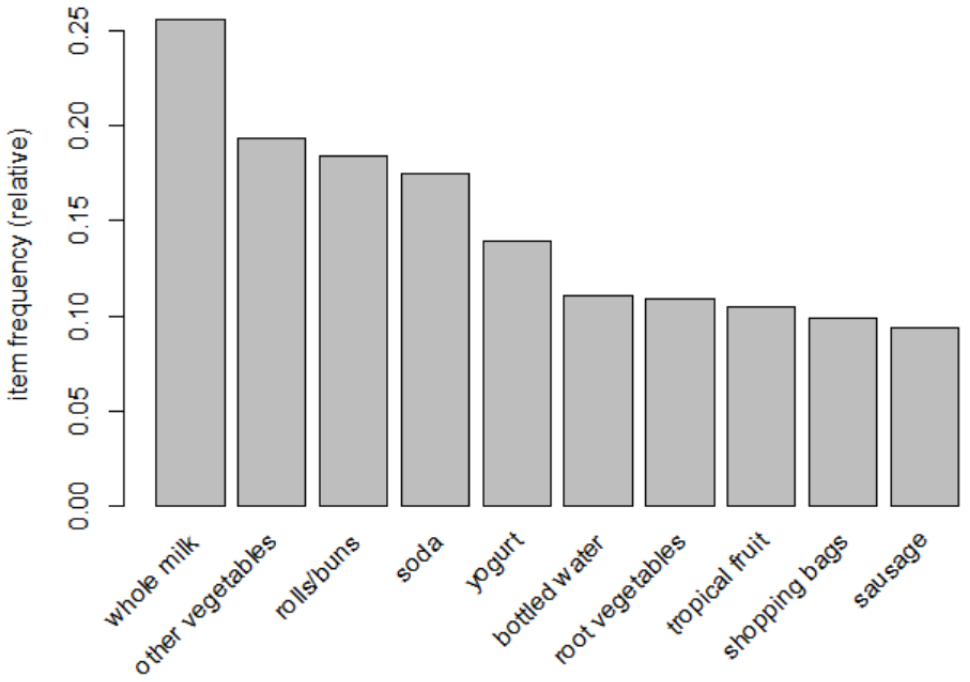
```
itemFrequencyPlot(groceries, support=0.1)
```

* 지지도가 0.1이 넘는 아이템들은 8개




```
itemFrequencyPlot(groceries, topN=10)
```

* 많이 거래되는 아이템 상위 10개 출력



4. 장바구니 분석 실습

- 이제 데이터를 살펴봤으니, 연관 규칙을 알아보기 위해 apriori() 함수를 사용하자!
- apriori() 함수는 최소 기준을 만족하는 모든 규칙을 저장하고 있는 규칙 객체를 반환한다.

```
rule <- apriori(data, parameter = list(support = 0.1, confidence = 0.8, minlen = ···), ··· )
```

- data : 희소 아이템 행렬
- *parameter = list()* 형태로 파라미터 줄 수 있음
  - support : 최소 지지도 요건 (디폴트 0.1)
  - confidence : 최소 신뢰도 요건 (디폴트 0.8)
  - minlen : 요구되는 최소 규칙 아이템

파라미터 설정이 중요함! (후에 조정 방법 설명)

* 너무 높게 잡으면 규칙을 찾지 못하거나 포괄적이어서 유용하지 않음
* 너무 낮게 잡으면 규칙이 너무 많거나 메모리를 많이 잡아먹음

- inspect() 함수와 sort() 함수를 함께 사용하여, 연관 규칙 중 어떤 것이 의미있는지 확인할 수 있다.

```
inspect(sort(rule, by = ··· )
```

* by 옵션에 규칙 효용성 지표인 'confidence', 'lift' 등을 적는다

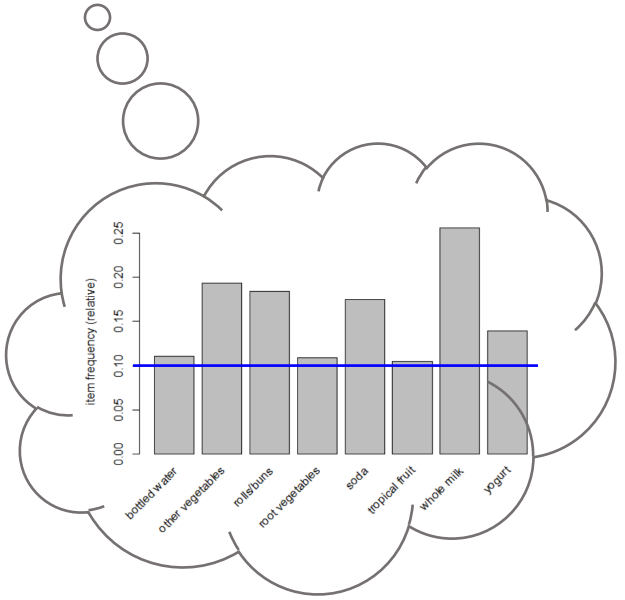
4. 장바구니 분석 실습

- 지지도와 신뢰도 파라미터를 디폴트 값으로 두고 연관 규칙을 생성 (rule)

```
> rule <- apriori(groceries, parameter=list(support=0.1, confidence=0.8))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1
 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: 983

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

- **0개**의 연관 규칙이 만들어졌다...!
- Why? 0.1로 설정한 지지도 파라미터가 너무 높았음

* itemFrequencyPlot에서 지지도 0.1일 때 아이템이 8개 밖에 없었던 것을 떠올려보자



파라미터 조정이 필요해 !

- support (최소 지지도 요건) 파라미터 조정
  - 아이템이 하루에 2번 구매되는 패턴을 눈여겨볼 만한 패턴이라고 보고, 최소 지지도 요건으로 설정하자
  - 하루에 2번 거래는 한 달에 60번 거래를 의미하므로, 아이템 거래 건수 (60) / 전체 거래 건수 (9835) = 0.006

- confidence (최소 신뢰도 요건) 파라미터 조정
  - 디폴트 0.8이 너무 높다고 판단이 들면, 0.25를 임계치로 놓고 서서히 올려가자
  - 0.25는 규칙이 결과에 포함되려면 최소 25% 정확해야 한다는 뜻

- minlen (요구되는 최소 규칙 아이템 수) 파라미터 조정
  - 2로 설정하면 1로 설정되었을 때보다 불필요한 규칙이 생성되는 것을 방지할 수 있음

```
rule2 <- apriori(groceries,
                 parameter=list(support=0.006, confidence=0.25, minlen=2))
```

4. 장바구니 분석 실습

- 지지도와 신뢰도 파라미터를 조정한 값을 넣은 식으로 연관 규칙을 다시 생성 (rule2)

```
> rule2 <- apriori(groceries,
+                  parameter=list(support=0.006, confidence=0.25, minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
       0.25    0.1    1 none FALSE            TRUE       5   0.006      2
 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: 59

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [109 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [463 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

- 463개의 연관 규칙이 만들어진 것을 확인

4. 장바구니 분석 실습

- 연관 규칙을 시각화 해보자! (*arulesViz 패키지)
- plot() 함수에 여러가지 옵션을 주어, 연관 규칙과 아이템 집합을 시각화 할 수 있음

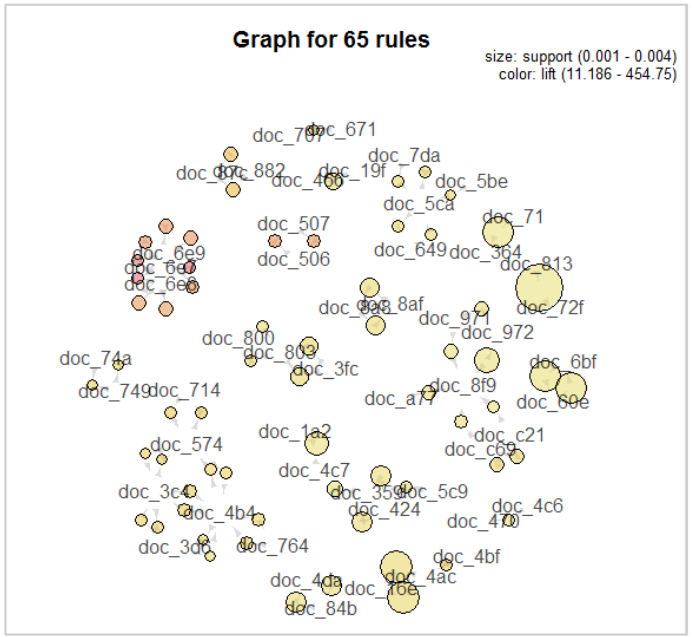
plot(rule, method = 'graph', control = list(type = 'items'))

- rule = 연관 규칙 객체
- method 옵션
  - 'graph' : 큰 원과 작은 원으로 그림 그려줌. 화살표의 두께는 지지도를 색상의 진하기는 향상도를 나타냄
  - 'grouped' : 좌측을 연관 규칙의 조건, 우측을 결과로 함. 원의 크기는 지지도를 색상의 진하기는 향상도를 의미
  - 'scatterplot' : 지지도와 신뢰도를 산점도로 보여준다. X축은 지지도, y축은 신뢰도 색상은 향상도
- control 옵션
  - list(type = 'items') : 아이템들의 연관 규칙 관계를 시각화
  - list(type = 'itemsets') : 아이템 집합들의 연관 규칙 관계를 시각화

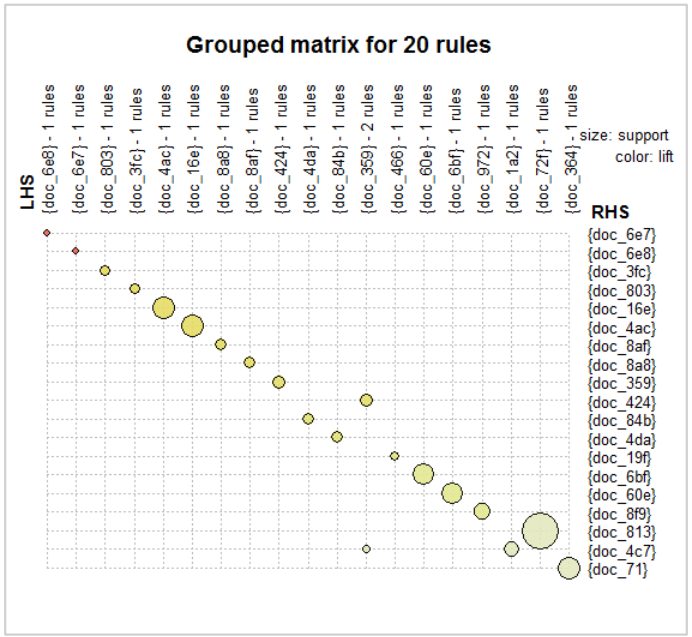
4. 장바구니 분석 실습

- plot(rule, method = 'graph', control = list(type = 'items'))
  - ➢ 'graph' : 큰 원과 작은 원으로 그림 그려줌. 화살표의 두께는 지지도를 색상의 진하기는 향상도를 나타냄
  - ➢ 'grouped' : 좌측을 연관 규칙의 조건, 우측을 결과로 함. 원의 크기는 지지도를 색상의 진하기는 향상도를 의미
  - ➢ 'scatterplot' : 지지도와 신뢰도를 산점도로 보여준다. X축은 지지도, y축은 신뢰도 색상은 향상도

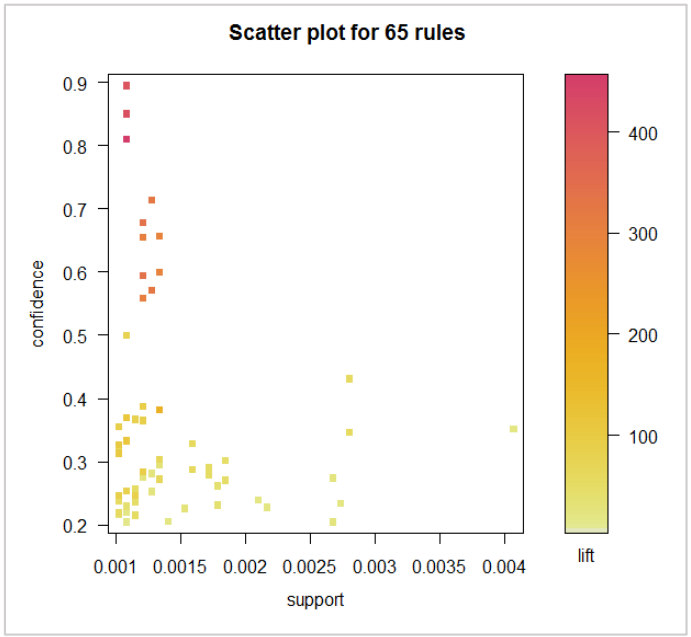
method = 'graph' (예시 데이터)



method = 'grouped' (예시 데이터)



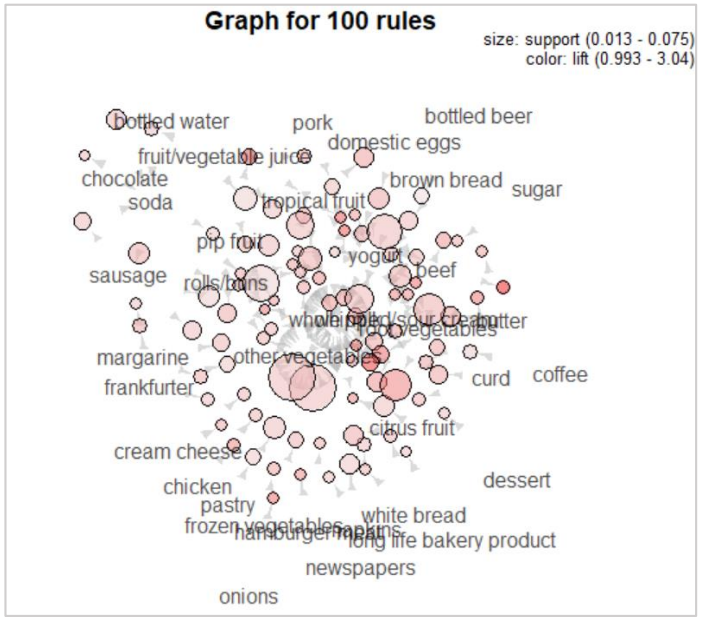
method = 'scatterplot' (예시 데이터)



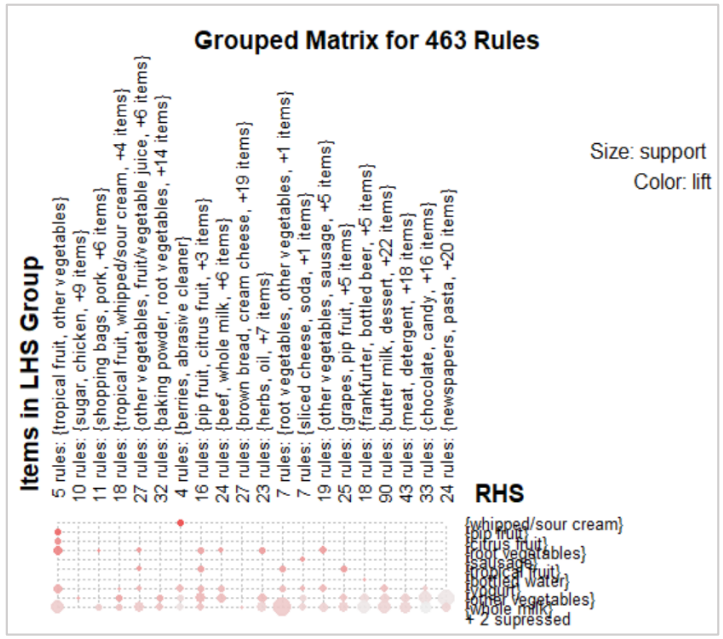
4. 장바구니 분석 실습

- plot(rule, method = 'graph', control = list(type = 'items'))
  - ➢ 'graph' : 큰 원과 작은 원으로 그림 그려줌. 화살표의 두께는 지지도를 색상의 진하기는 향상도를 나타냄
  - ➢ 'grouped' : 좌측을 연관 규칙의 조건, 우측을 결과로 함. 원의 크기는 지지도를 색상의 진하기는 향상도를 의미
  - ➢ 'scatterplot' : 지지도와 신뢰도를 산점도로 보여준다. X축은 지지도, y축은 신뢰도 색상은 향상도

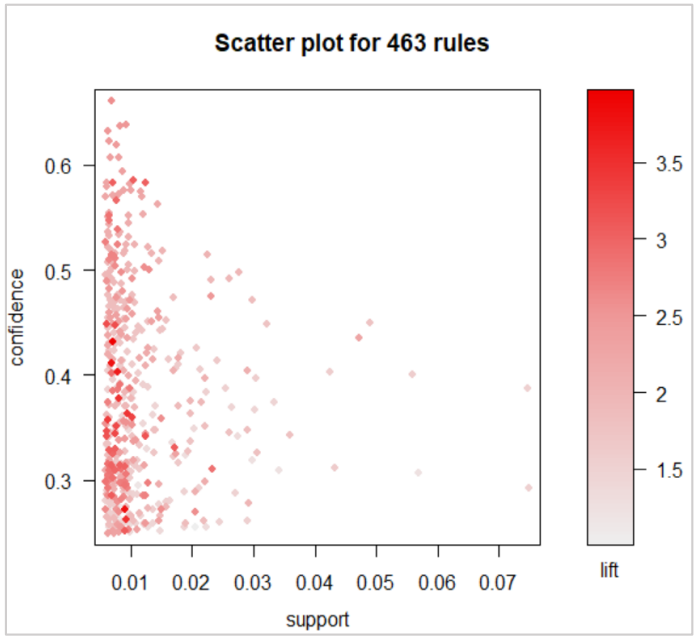
method = 'graph'



method = 'grouped'



method = 'scatterplot'

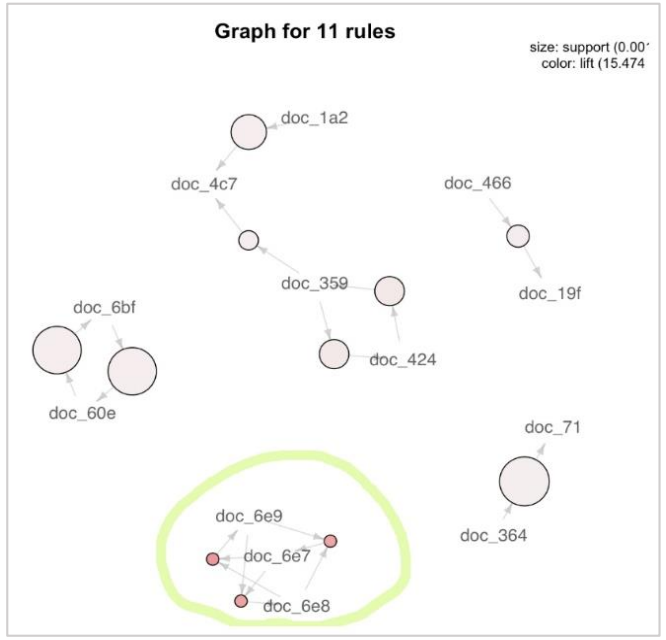


4. 장바구니 분석 실습

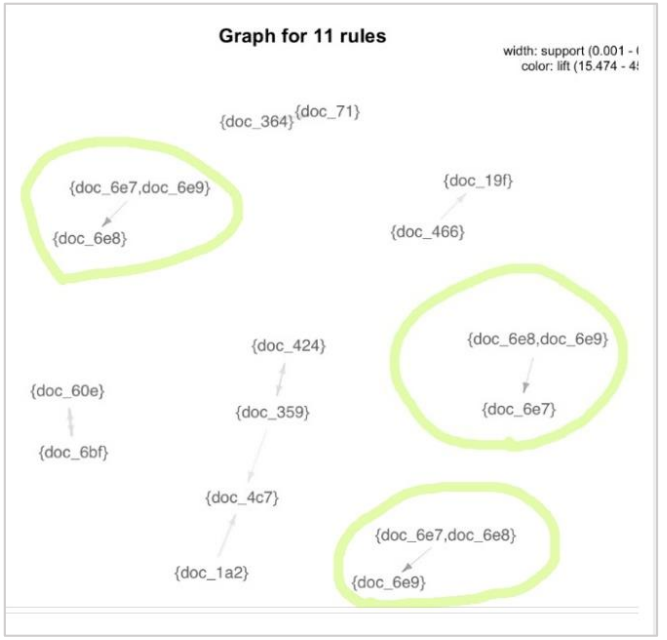
- plot(rule, method = 'graph', control = list(type = 'items'))

  - list(type = 'items') : 아이템들의 연관 규칙 관계를 시각화
  - list(type = 'itemsets') : 아이템 집합들의 연관 규칙 관계를 시각화

control = list(type = 'items') (예시 데이터)



control = list(type = 'itemsets') (예시 데이터)



4. 장바구니 분석 실습

```
> summary(rule2)
set of 463 rules

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

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.711   3.000   4.000
```

```
summary of quality measures:
    support         confidence        coverage         lift
 Min.   :0.006101   Min.   :0.2500   Min.   :0.009964   Min.   :0.9932
 1st Qu.:0.007117   1st Qu.:0.2971   1st Qu.:0.018709   1st Qu.:1.6229
 Median :0.008744   Median :0.3554   Median :0.024809   Median :1.9332
 Mean   :0.011539   Mean   :0.3786   Mean   :0.032608   Mean   :2.0351
 3rd Qu.:0.012303   3rd Qu.:0.4495   3rd Qu.:0.035892   3rd Qu.:2.3565
 Max.   :0.074835   Max.   :0.6600   Max.   :0.255516   Max.   :3.9565
     count
 Min.   : 60.0
 1st Qu.: 70.0
 Median : 86.0
 Mean   :113.5
 3rd Qu.:121.0
 Max.   :736.0
```

```
mining info:
      data ntransactions support confidence
 groceries          9835   0.006       0.25
```

➢ summary 설명

- 463개의 연관 규칙 생성
- 규칙이 포함하는 아이템 수와 규칙의 개수
  * LHS (조건절), RHS(결과절)의 아이템은 상호배반 관계이기 때문에 겹치는 아이템이 없음
- 아이템 2개를 포함하는 규칙은 150개
- 아이템 3개를 포함하는 규칙은 297개

- quality measures = 규칙 효용성 지표
- 지지도, 신뢰도, 향상도

- mining info = 규칙을 어떻게 발굴했는가?
- ntransactions : 전체 거래수

4. 장바구니 분석 실습

- inspect()와 sort() 함수를 함께 사용하여, 연관 규칙 중 어떤 것이 의미있는지 확인할 수 있다.

```
inspect(sort(rule2, by = ··· ))
```

* by 옵션에 규칙 효용성 지표인 'confidence', 'lift' 등을 적는다

- 향상도 순으로 연관 규칙을 sort하고, 그 희소 행렬의 내용을 inspect 함수로 살펴봄
- 향상도가 높은 규칙 첫 번째부터 다섯 번째까지만 나열

```
inspect(sort(rule2, by='lift')[1:5])
inspect(sort(rule2, by='confidence')[1:5])
```

- 신뢰도 순으로 연관 규칙을 sort하고, 그 희소 행렬의 내용을 inspect 함수로 살펴봄
- 신뢰도가 높은 규칙 첫 번째부터 다섯 번째까지만 나열

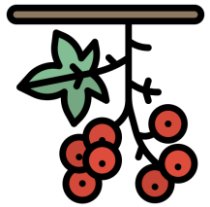
4. 장바구니 분석 실습

```
> inspect(sort(rule2, by='lift')[1:5])
    lhs                                                rhs                    support     confidence coverage   lift     count
[1] {herbs}                                         => {root vegetables}      0.007015760 0.4312500  0.01626843 3.956477 69
[2] {berries}                                       => {whipped/sour cream}   0.009049314 0.2721713  0.03324860 3.796886 89
[3] {other vegetables,tropical fruit,whole milk} => {root vegetables}      0.007015760 0.4107143  0.01708185 3.768074 69
[4] {beef,other vegetables}                         => {root vegetables}      0.007930859 0.4020619  0.01972547 3.688692 78
[5] {other vegetables,tropical fruit}               => {pip fruit}            0.009456024 0.2634561  0.03589222 3.482649 93
> inspect(sort(rule2, by='confidence')[1:5])
    lhs                             rhs           support     confidence coverage    lift     count
[1] {butter,whipped/sour cream} => {whole milk} 0.006710727 0.6600000  0.010167768 2.583008 66
[2] {butter,yogurt}             => {whole milk} 0.009354347 0.6388889  0.014641586 2.500387 92
[3] {butter,root vegetables}    => {whole milk} 0.008235892 0.6377953  0.012913066 2.496107 81
[4] {curd,tropical fruit}       => {whole milk} 0.006507372 0.6336634  0.010269446 2.479936 64
[5] {butter,tropical fruit}     => {whole milk} 0.006202339 0.6224490  0.009964413 2.436047 61
```

- 향상도 순 : [1] 허브를 사는 사람은 뿌리 식물을 함께 구매함, [2] 베리류를 사는 사람은 크림을 함께 구매함
- 어떤 범주? 실행 가능한 범주 --> 수용 가능!

- 신뢰도 순 : [1] 버터와 크림을 사는 사람은 전유를 함께 구매함, [2] 버터와 요거트를 사는 사람이 전유를 함께 구매함
- 어떤 범주 ? 실행 가능한 범주 --> 수용 가능!

4. 장바구니 분석 실습



만약, 베리가 어떤 아이템과 같이 자주 구매되는지 알고 싶다면?

연관 규칙에서 베리의 부분 집합을 구한다

```
berry_rule <- subset(rule2, items %in% "berries")
```

- 연산자 %in%는 아이템 중 최소 하나가 정의한 목록에서 발견돼야만 함
  두 가지 아이템을 적고 싶다면 c('berries', 'yogurt')와 같이 적을 수 있음
- 부분 매칭 **%pin%** 또는 완전 매칭 **%ain%**을 사용할 수 있다
- 매칭 조건은 and, or, not과 같은 논리 연산자와 같이 결합하여 사용할 수 있다

4. 장바구니 분석 실습

➢ 규칙 효용 평가

```
> inspect(sort(berry_rule, by='lift'))
    lhs           rhs                    support     confidence coverage  lift     count
[1] {berries} => {whipped/sour cream} 0.009049314 0.2721713  0.0332486 3.796886  89
[2] {berries} => {yogurt}             0.010574479 0.3180428  0.0332486 2.279848 104
[3] {berries} => {other vegetables}   0.010269446 0.3088685  0.0332486 1.596280 101
[4] {berries} => {whole milk}         0.011794611 0.3547401  0.0332486 1.388328 116
> inspect(sort(berry_rule, by='confidence'))
    lhs           rhs                    support     confidence coverage  lift     count
[1] {berries} => {whole milk}         0.011794611 0.3547401  0.0332486 1.388328 116
[2] {berries} => {yogurt}             0.010574479 0.3180428  0.0332486 2.279848 104
[3] {berries} => {other vegetables}   0.010269446 0.3088685  0.0332486 1.596280 101
[4] {berries} => {whipped/sour cream} 0.009049314 0.2721713  0.0332486 3.796886  89
```

➢ 향상도 순 : [1] 베리를 사는 사람은 크림을 함께 구매함, [2] 베리를 사는 사람은 요거트를 함께 구매함

➢ 어떤 범주? 실행 가능한 범주 --> 수용 가능!

➢ 신뢰도 순 : [1] 베리를 사는 사람은 전유를 함께 구매함, [2] 베리를 사는 사람이 요거트를 함께 구매함

➢ 어떤 범주 ? 실행 가능한 범주 --> 수용 가능!

감사합니다 :D