# 연관성 분석

# 연관 검색어는 뭘까

# Top5 검색어 찾기

```
# 생성된 리스트 보기 - 28개의 고객 정보를 리스트화 시킨 결과
str(searchL)
# 리스트 안의 모든 항목을 풀어 하나의 벡터로 변환
searchVec <- unlist(search)
# 생성된 벡터 보기(217개 항목의 벡터 1개로 변환)
str(searchVec)
# 검색어별 빈도수 확인을 위해 테이블로 변환
searchT <- table(searchVec)
# 생성된 테이블 객체 확인(각 검색어별 빈도수가 계산됨)
searchT
# 빈도수 기준으로 정렬(decreasing=T: 내림차순 정렬. 오름차순은 F)
searchT <- sort(searchT, decreasing=T)
# 상위 5개 검색어 확인(정렬된 항목 중 앞 5개 항목 출력)
searchT[1:5]
```

# Sapply 확인

```
#데이터 비교
"유럽" %in% "동유럽"
"유럽" %in% "유럽“
#문자열 벡터 생성 후 비교
Search.data=c( " 동유럽 " ,  ' 유럽 ' )
 " 유럽 "  %in% search.data
#sapply를 이용한 비교
sapply(search.data, function(x){"유럽" %in% x})
```

# 유럽을 검색한 사람들

```
# 리스트 중 유럽을 조회한 사람들의 검색어가 있는 벡터들의 위치를 논리벡터로 산출
# sapply: 리스트 각 항목에 함수 적용한 결과를 벡터로 반환
# "유럽" %in% x: 벡터 x 요소 중 "유럽"이 있으면 TRUE, 없으면 FALSE
searchEuropeIdx <- sapply(searchL, function(x){"유럽" %in% x})
# 산출된 논리벡터 확인("유럽"이라는 단어를 포함한 벡터의 위치는 TRUE)
searchEuropeIdx
# 유럽을 조회한 사람들의 검색어만 따로 분리
searchEuropeL <- searchL[searchEuropeIdx]
# 분리한 검색어 내역 확인(유럽을 조회한 11명의 검색어 모음)
str(searchEuropeL)
```

# Quiz - 유럽을 검색한 사람들의 Top5

```
> searchEuropeT[1:5]

    유럽   지중해 배낭여행       날씨 이탈리아
      11        6        5        4        4
```

# Quiz - 유럽을 검색한 사람들의 Top5

```
> searchEuropeT[1:5]

    유럽   지중해 배낭여행       날씨 이탈리아
      11        6        5        4        4
```

```
# 테이블 변환
searchEuropeT <- table(unlist(searchEuropeL))
# 빈도수 기준으로 정렬
searchEuropeT <- sort(searchEuropeT, decreasing=T)
# 상위 5개 검색어 확인
searchEuropeT[1:5]
```

# 연관성?

```
> searchEuropeT[1:5]

    유럽   지중해 배낭여행     날씨 이탈리아
      11       6        5        4        4
```

같이 많이 검색됐다는 것만으로 연관성이 있을까?

연관성을 수치화 할 수 있을까?

여러 단어에 대한 연관성을 구할 수 있을까?

# 연관성 분석

# 이해하기

❖ **연관성 분석이란**

- 대량의 데이터에 숨겨진 항목 간의 연관 규칙을 찾아내는 기법
- 장바구니 분석이라고도 함

❖ **활용예**

- 카트에 담긴 물건을 보고 사람의 취향 분석
- “A물건을 사멸 “B물건을 살 확률이 높다”?
- A물건과 B물건을 같이 두어 구매 유도
- A물건과 B물건의 거리를 멀리두어 이동하는 동안 다른 물건을 구매하도록 유도

# 연관성 관련 지표

- 연관성 분석을 통해 도출된 연관성 규칙을 다음과 같이 표현
  - {조건}=>{결과}
- 예
  - 와인과 소금을 사면 치즈를 산다
  - {와인, 소금}=>{치즈}
- 규칙성 평가 지표
  - 지지도(support)
  - 신뢰도(confidence)
  - 향상도(lift)

# 구매 물품 및 연관성 규칙 지정

| 거래ID | 구매물품 |
|---|---|
| 1 | 삼겹살, 생수, 소주, 과자 |
| 2 | 삼겹살, 생수, 소주, 사과 |
| 3 | 장어, 생수, 소주, 양파 |
| 4 | 땅콩, 생수, 맥주, 오이 |
| 5 | 땅콩, 생수, 맥주, 감 |

| 연관성 규칙 |
|---|
| {삼겹살, 생수}=>사과 |
| {생수}=>{사과} |
| {삼겹살}=>{생수} |
| {땅콩, 생수}=> {맥주} |
| {땅콩}=>{맥주} |

# 지지도(support)

- ❖ 전체 거래 중 연관성 규칙을 구성하는 항목들이 포함된 거래 비율
- ❖ 지지도 = (조건과 결과 항목을 포함하는 거래수) / (전체 거래수)

| 거래ID | 구매물품 |
|---|---|
| 1 | 삼겹살, 생수, 소주, 과자 |
| 2 | 삼겹살, 생수, 소주, 사과 |
| 3 | 장어, 생수, 소주, 양파 |
| 4 | 땅콩, 생수, 맥주, 오이 |
| 5 | 땅콩, 생수, 맥주, 감 |

| 연관성 규칙 | 지지도 |
|---|---|
| {삼겹살, 생수}=>{사과} | 1/5(20%) |
| {생수}=>{사과} | 1/5(20%) |
| {삼겹살}=>{생수} | 2/5(40%) |
| {땅콩, 생수}=> {맥주} | 2/5(40%) |
| {땅콩}=>{맥주} | 2/5(40%) |

# 신뢰도(confidence)

- ❖ 조건이 발생했을 때 결과가 동시에 일어날 확률
- ❖ 신뢰도가 1에 가까울 수록 의미 있는 연관성을 가짐
- ❖ 신뢰도 = (조건과 결과 항목을 포함하는 거래수) / (조건 항목을 포함한 거래수)

| 거래ID | 구매물품 |
|---|---|
| 1 | 삼겹살, 생수, 소주, 과자 |
| 2 | 삼겹살, 생수, 소주, 사과 |
| 3 | 장어, 생수, 소주, 양파 |
| 4 | 땅콩, 생수, 맥주, 오이 |
| 5 | 땅콩, 생수, 맥주, 감 |

| 연관성 규칙 | 신뢰도 |
|---|---|
| {삼겹살, 생수}=>{사과} | 1/2(50%) |
| {생수}=>{사과} | 1/5(20%) |
| {삼겹살}=>{생수} | 2/2(100%) |
| {땅콩, 생수}=> {맥주} | 2/2(100%) |
| {땅콩}=>{맥주} | 2/2(100%) |

- ❖ 위의 내용으로 삼겹살을 사면 무조건 생수를 사며
- ❖ 땅콩을 사면 무조건 맥주를 사는 것을 알 수 있다.

# 향상도(lift)

- ❖ 우연적인 관계까지도 감안해 산출하는 지표
- ❖ 향상도가 1인 경우 우연에 의한 관계이며 클 수록 연관성을 가짐
- ❖ 향상도 = 연관성 규칙의 지지도 / (조건의 지지도 * 결과의 지지도)

| 거래ID | 구매물품 |
|---|---|
| 1 | 삼겹살, 생수, 소주, 과자 |
| 2 | 삼겹살, 생수, 소주, 사과 |
| 3 | 장어, 생수, 소주, 양파 |
| 4 | 땅콩, 생수, 맥주, 오이 |
| 5 | 땅콩, 생수, 맥주, 감 |

| 연관성 규칙 | 향상도 |
|---|---|
| {삼겹살, 생수}=>{사과} | 0.2/(0.4*0.2) = 2.5 |
| {생수}=>{사과} | 0.2/(1*0.2)=1 |
| {삼겹살}=>{생수} | 0.4/(0.4*1)=1 |
| {땅콩, 생수}=> {맥주} | 0.4/(0.4*0.4)=2.5 |
| {땅콩}=>{맥주} | 0.4/(0.4*0.4)=2.5 |

- ❖ 위의 결과 {생수}=>{사과}, {삼겹살}=>{생수}는 결과가 1임으로 우연
- ❖ 2.5인 부분이 의미가 있는 부분임

# Apriori 알고리즘

# 이해하기

❖ Apriori 알고리즘
- “한 항목이 자주 발생하지 않으면 이 항목을 포함하는 집합들도 자주 발행하지 않는다“
- 발생 빈도를 기준으로 최소 지지도를 충족하지 못하는 항목을 제거함으로써 연관성 분석을 더 효율적으로 할 수 있음.

❖ 함수
- apriori(data, parameter)
- data : 트랜잭션 객체
- suport : 최소 지지도
- minlen : 연관성 규칙의 최소 항목 수
- maxlen : 연관성 규칙의 최대 항목 수
- confidence : 최소 신뢰도

❖ 트랜잭션 객체
- 1과 0으로 이루어져 있는 데이터에서 0이 훨씬 많을 때(spase format – 희소 형태의 데이터). 즉, 의미 없는 정보가 많고 크기가 커서 데이터를 처리하기 힘들 때 transactions class로 처리

# 설치 및 데이터 생성

```
# 최초 수행 시 패키지 설치
install.packages("arules")
# arules 패키지 로드
library(arules)
# 고객별 구매 품목을 리스트로 생성
buyItems <- list(
  c("삼겹살", "생수", "소주", "과자")
  ,c("삼겹살", "생수", "소주", "사과")
  ,c("장어", "생수", "소주", "양파")
  ,c("땅콩", "생수", "맥주", "오이")
  ,c("땅콩", "생수", "맥주", "감")
)
# 트랜잭션 데이터로 형변환
buyItemStr <- as(buyItems, "transactions")
# 변환된 트랜잭션 확인(11개 항목에 대해 5개 거래 존재)
buyItemStr
```

# 81개의 규칙 생성

```
# 트랜잭션 데이터는 inspect 함수를 통해 내용을 확인
inspect(buyItemStr)
# apriori 함수 수행(지지도 0.1, 신뢰도 0.8 이상인 연관성 규칙 구하기)
buyItemResult <- apriori(buyItemStr, parameter=list(
                          support=0.1, confidence=0.8
                         ))
```

```
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 5 transaction(s)] done [0.00s].
sorting and recoding items ... [11 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [81 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

# 5개의 데이터 분석하기

```
# 도출된 연관성 규칙 5개만 확인
buyItemResult[1:5]
# 연관성 규칙 상세 보기
inspect(buyItemResult[1:5])
```

```
    lhs        rhs        support confidence coverage lift     count
[1] {}      => {생수}     1.0     1          1.0      1.000000 5
[2] {과자}  => {삼겹살}   0.2     1          0.2      2.500000 1
[3] {과자}  => {소주}     0.2     1          0.2      1.666667 1
[4] {과자}  => {생수}     0.2     1          0.2      1.000000 1
[5] {사과}  => {삼겹살}   0.2     1          0.2      2.500000 1
```

lhs(left hand side) : 원인
rhs(right hand side) : 결과
suport : 지지도
confidence : 신뢰도
coverage : 범위(원인이 나타날 확률)
lift : 향상도
count : 개수

# subset 조건 처리

```
# 향상도가 1 초과인 연관성 규칙만 선택
subBuyResult <- subset(buyItemResult, subset=lift > 1 )
# subset 결과
subBuyResult
# 연관성 규칙 5개만 확인
inspect(subBuyResult[1:5])
```

# 검색 키워드

| 연산자 | 사용 예 | 의미 |
|---|---|---|
| %in% | lhs %in% c("과자", "삽겹살") | 일부가 포함되어 있는지 확인 |
| %ain% | lhs %ain% c("과자", "삽겹살") | 모두가 포함되어 있는지 확인 |
| %oin% | lhs %oin% c("과자", "삽겹살") | 모든 경우의 수 확인 |
| %pin% | lhs %pin% "과" | 철자가 포함된 모든 경우 확인 |

# 결과 확인

```
# lhs에 삼겹살이 포함된 연관성 규칙
inspect(subset(buyItemResult, subset=lhs %in% c("삼겹살")))
# lhs에 삼겹살과 과자가 포함된 연관성 규칙
inspect(subset(buyItemResult, subset=lhs %ain% c("삼겹살", "과자")))
# lhs가 삼겹살 or 과자 or 삼겹살과 과자인 연관성 규칙
inspect(subset(buyItemResult, subset=lhs %oin% c("삼겹살", "과자")))
# lhs 항목 중 "겹"이라는 글자를 포함하는 연관성 규칙
inspect(subset(buyItemResult, subset=lhs %pin% "겹"))
```

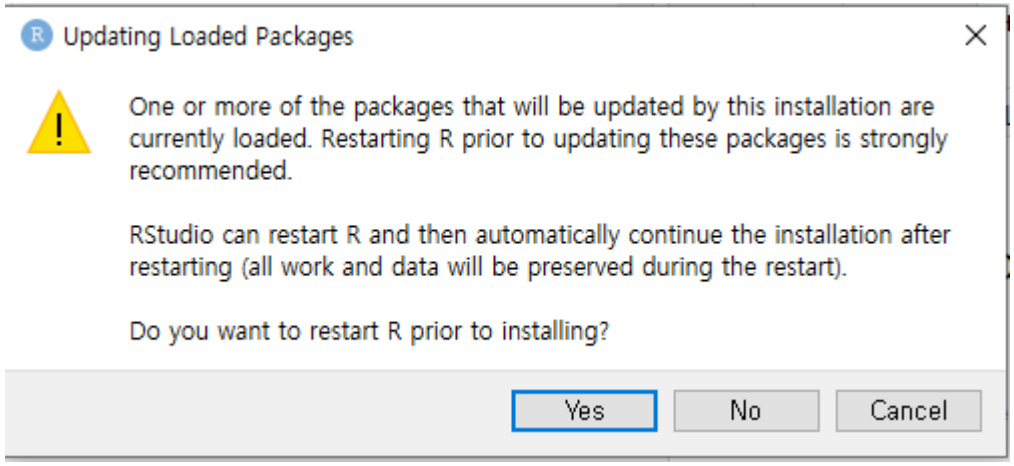
# 상위 10 추출

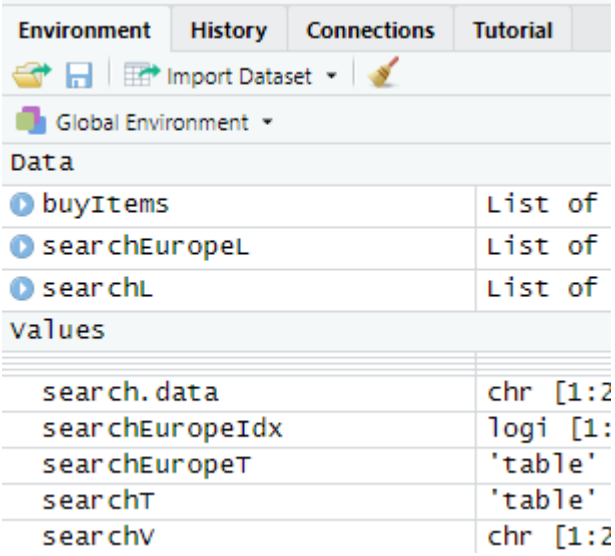
```
# 지지도, 신뢰도, 향상도 기준으로 정렬
subBuyResult_order <- sort(subBuyResult, by=c("support", "lift",
"confidence"))
# 상위 10개만 확인
inspect(subBuyResult_order[1:10])
```

# 시각화

# 에러

❖ install.packages("arules")



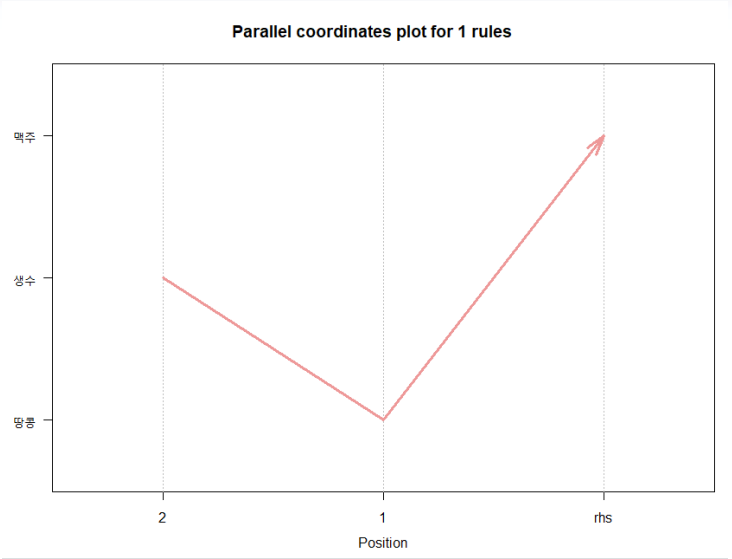

설치 시 위와 같이 창이 뜨고 설치가 안되면
변수 전체를 지우고 다시 설치

# 막대 그래프 그리기

```
# 처음 실행 시 패키지를 설치
install.packages("arules")
# arules 패키지 로드
library(arules)
# 항목별 빈도수 시각화(최소 지지도 0.2 이상인 항목에 대해서만 빈도수 확인)
# itemFrequencyPlot 함수는 "트랜잭션 데이터"를 입력 항목으로 받습니다.
itemFrequencyPlot(buyItemStr, support=0.2)
```

# 평행좌표 그래프

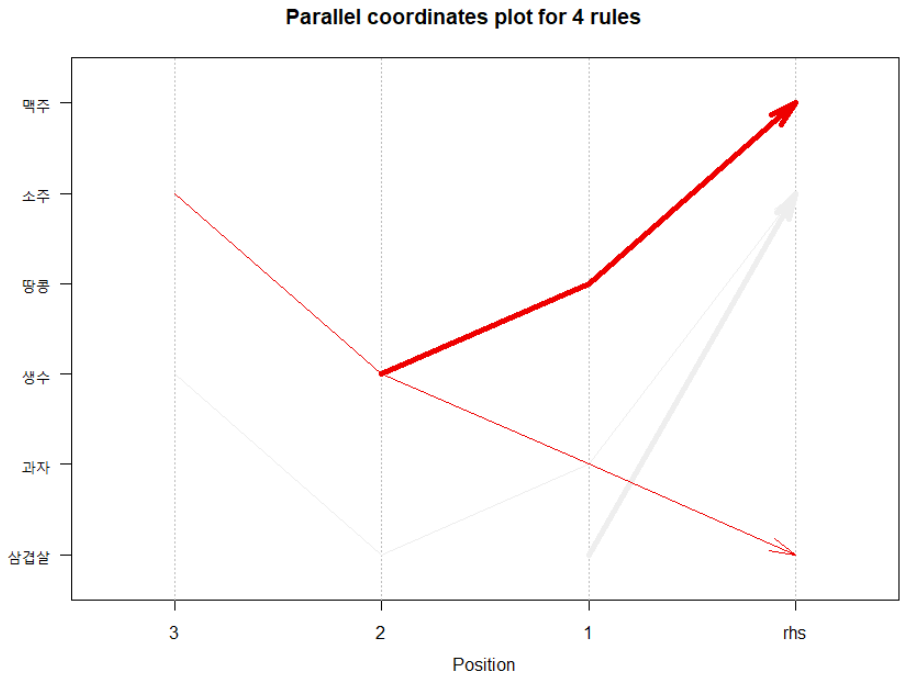
```
# 최초 실행 시 패키지 설치 필요
install.packages("arulesViz")
# 패키지 로드
library(arulesViz)
# 연관성 분석 결과 객체
subBuyResult_order
# 3번째 연관성 규칙 확인
inspect(subBuyResult_order[3])
# 3번째 연관성 규칙을 평행좌표 그래프(paracoord)로 표현
plot(subBuyResult_order[3], method="paracoord")
```




생수와 땅콩을 사면 맥주를 산다

- # 3,5,33,50번째 연관성 규칙 확인
- inspect(subBuyResult_order[c(3, 5, 33, 50)])
- # 3,5,33,50번째 연관성 규칙을 하나의 평행좌표 그래프에 표현
- plot(subBuyResult_order[c(3, 5, 33, 50)], method="paracoord")

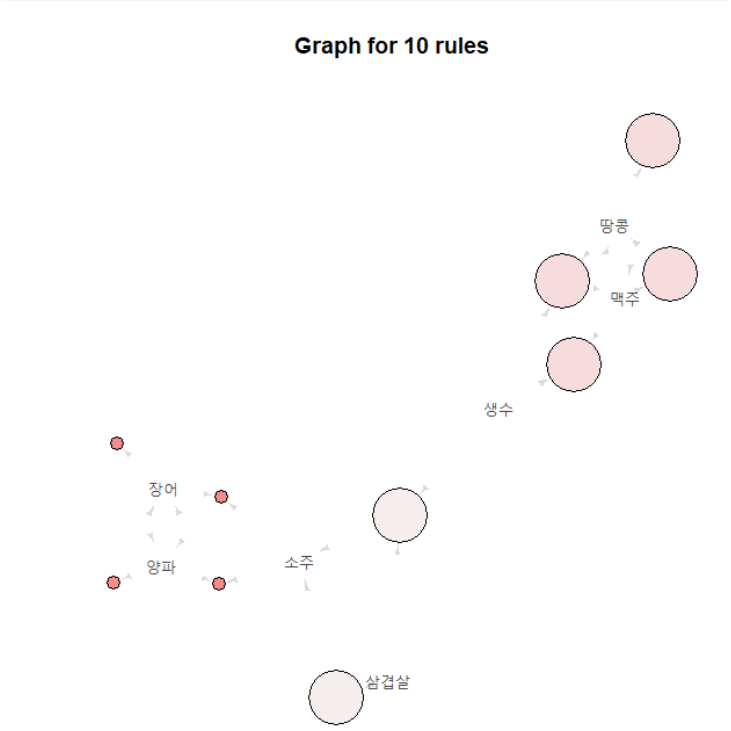
굵을수록 지지도가 높음
생수와 땅콩을 사는 사람은 맥주를 사며 삼겹살을 사는 사람은 소주를 살 확률이 높은 것을 알 수 있다.



# 네트워크 그래프

```
# 처음 10개의 연관성 분석 확인
inspect(subBuyResult_order[1:10])
# 10개의 연관성 규칙에 대한 네트워크 그래프 그리기
plot(subBuyResult_order[1:10], method="graph")
```

원의 크기가 클수록
지지도가 높음
땅콩과 맥주
소주와 삼겹살은 연관관계가
높으며 생수는 모두 산다



# 여행사 분석

# 트랜젝션 데이터 변환

```
# 고객 검색 내역 확인
str(searchL)
# 트랜잭션 데이터 변환
searchT <- as(searchL, "transactions")
# 생성된 트랜잭션 확인
searchT
# 트랜잭션 내용 확인
inspect(searchT)
```

# 연관성 분석

```
# 처음 실행 시 설치
install.packages("arules")
# arules 패키지 로드
library(arules)
# 연관성 분석(지지도 0.1 이상, 신뢰도 0.8 이상 연관성 규칙 도출)
aResult <- apriori(searchT, parameter=list(support=0.1, confidence=0.8))
# 도출된 연관성 규칙 지지도, 향상도, 신뢰도 기준으로 정렬
aResult <- sort(aResult, by=c("support", "lift", "confidence"))
# 연관성 규칙 확인
inspect(aResult)
```

# 배낭여행이 들어간 모든 경우 추출

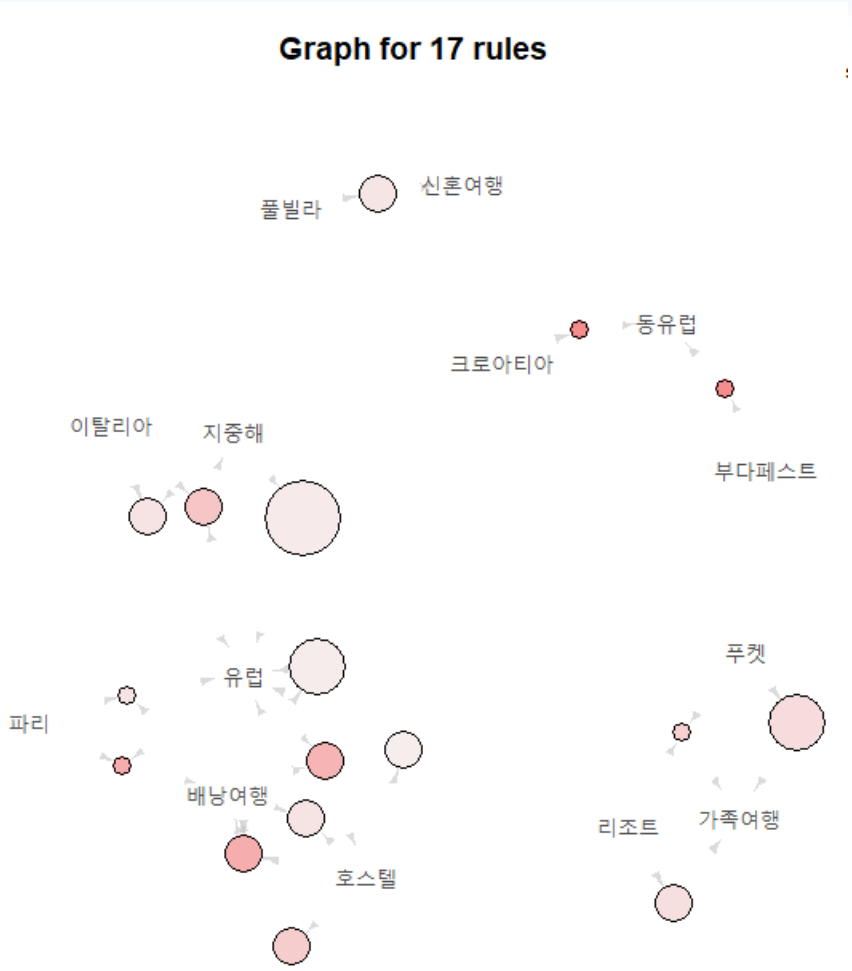
```
# 배낭여행을 포함하는 연관성 규칙 추출
packResult <- subset(aResult, subset=lhs %in% c("배낭여행") | rhs %in% c("배낭여행"))
# 연관성 규칙 확인
inspect(packResult)
```

# 배낭여행 연관 키워드 추출

```
# 조건 항목만 별도 추출해 리스트로 변환
packLhs <- as(lhs(packResult), "list")
# 조건 항목 확인
str(packLhs)
# 결과 항목만 별도 추출해 리스트로 변환
packRhs <- as(rhs(packResult), "list")
# 결과 항목 확인
str(packRhs)
# 조건과 결과 항목을 벡터로 변환
vPackWord <- unlist(packLhs, packRhs)
# 배낭여행과 연관된 검색어 확인
vPackWord
# 중복 항목 제거
unique(vPackWord)
```

# 네트워크 그래프 분석

```
# 최초 실행 시 패키지 설치 필요
install.packages("arulesViz")
# 패키지 로드
library(arulesViz)
# 검색어 네트워크 그래프
plot(aResult, method="graph")
```




신혼여행 – 풀빌라 소개
배낭여행 – 호스텔 소개
가족여행 – 리조트 소개