7_Association_Analysis

Park Ju ho

2022 6 9

Titanic 데이터를 이용한 연관 분석

Freq 값은 해당 행과 일치하는 데이터의 빈도수 이를 loop를 활용하여 데이터를 해제시켜주고 모두 문자열로 변경시켜 줌

```
titanic.df = as.data.frame(Titanic)
head(titanic.df)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```
summary(titanic.df)
```

```
##   Class       Sex         Age     Survived      Freq
##  1st :8   Male  :16   Child:16   No :16   Min.   :  0.00
##  2nd :8   Female:16   Adult:16   Yes:16   1st Qu.:  0.75
##  3rd :8                                  Median : 13.50
##  Crew:8                                  Mean   : 68.78
##                                          3rd Qu.: 77.00
##                                          Max.   :670.00
```

```
titanic <- NULL
for(i in 1:4) { titanic <- cbind(titanic,
                                 rep(as.character(titanic.df[,i]), titanic.df$Freq)) }
titanic <- as.data.frame(titanic)
names(titanic) <- names(titanic.df)[1:4]
head(titanic)
```

```
##   Class  Sex   Age Survived
## 1   3rd Male Child       No
## 2   3rd Male Child       No
## 3   3rd Male Child       No
## 4   3rd Male Child       No
## 5   3rd Male Child       No
## 6   3rd Male Child       No
```

```
library(arules)

rules.all = apriori(titanic)
```

```
## 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: 220
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[10 item(s), 2201 transaction(s)] done [0.00s].
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [27 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
options(digits=3)
inspect(rules.all)
```

```
##      lhs                                rhs           support confidence
## [1]  {}                              => {Age=Adult}   0.950   0.950
## [2]  {Class=2nd}                     => {Age=Adult}   0.119   0.916
## [3]  {Class=1st}                     => {Age=Adult}   0.145   0.982
## [4]  {Sex=Female}                    => {Age=Adult}   0.193   0.904
## [5]  {Class=3rd}                     => {Age=Adult}   0.285   0.888
## [6]  {Survived=Yes}                  => {Age=Adult}   0.297   0.920
## [7]  {Class=Crew}                    => {Sex=Male}    0.392   0.974
## [8]  {Class=Crew}                    => {Age=Adult}   0.402   1.000
## [9]  {Survived=No}                   => {Sex=Male}    0.620   0.915
## [10] {Survived=No}                   => {Age=Adult}   0.653   0.965
## [11] {Sex=Male}                      => {Age=Adult}   0.757   0.963
## [12] {Sex=Female, Survived=Yes}      => {Age=Adult}   0.144   0.919
## [13] {Class=3rd, Sex=Male}           => {Survived=No} 0.192   0.827
## [14] {Class=3rd, Survived=No}        => {Age=Adult}   0.216   0.902
## [15] {Class=3rd, Sex=Male}           => {Age=Adult}   0.210   0.906
## [16] {Sex=Male, Survived=Yes}        => {Age=Adult}   0.154   0.921
## [17] {Class=Crew, Survived=No}       => {Sex=Male}    0.304   0.996
## [18] {Class=Crew, Survived=No}       => {Age=Adult}   0.306   1.000
## [19] {Class=Crew, Sex=Male}          => {Age=Adult}   0.392   1.000
## [20] {Class=Crew, Age=Adult}         => {Sex=Male}    0.392   0.974
## [21] {Sex=Male, Survived=No}         => {Age=Adult}   0.604   0.974
## [22] {Age=Adult, Survived=No}        => {Sex=Male}    0.604   0.924
```

```
## [23] {Class=3rd, Sex=Male, Survived=No}   => {Age=Adult}   0.176   0.917
## [24] {Class=3rd, Age=Adult, Survived=No}  => {Sex=Male}    0.176   0.813
## [25] {Class=3rd, Sex=Male, Age=Adult}     => {Survived=No} 0.176   0.838
## [26] {Class=Crew, Sex=Male, Survived=No}  => {Age=Adult}   0.304   1.000
## [27] {Class=Crew, Age=Adult, Survived=No} => {Sex=Male}    0.304   0.996
##      coverage lift  count
## [1]  1.000    1.000 2092
## [2]  0.129    0.964  261
## [3]  0.148    1.033  319
## [4]  0.214    0.951  425
## [5]  0.321    0.934  627
## [6]  0.323    0.968  654
## [7]  0.402    1.238  862
## [8]  0.402    1.052  885
## [9]  0.677    1.164 1364
## [10] 0.677    1.015 1438
## [11] 0.786    1.013 1667
## [12] 0.156    0.966  316
## [13] 0.232    1.222  422
## [14] 0.240    0.948  476
## [15] 0.232    0.953  462
## [16] 0.167    0.969  338
## [17] 0.306    1.266  670
## [18] 0.306    1.052  673
## [19] 0.392    1.052  862
## [20] 0.402    1.238  862
## [21] 0.620    1.025 1329
## [22] 0.653    1.175 1329
## [23] 0.192    0.965  387
## [24] 0.216    1.034  387
## [25] 0.210    1.237  387
## [26] 0.304    1.052  670
## [27] 0.306    1.266  670
```

support = 지지도, confidence = 신뢰도, lift = 향상도 lhs = X, rhs = Y라고 생각하면 됨 1을 해석해보면 X가 공집합이기에 지지도,신뢰도 = 전체에서 어른의 비율, 그렇기에 향상도도 1이 나올 수 밖에 없음 이 때 지지도,신뢰도,향상도 중 어떤 것을 기준으로 삼을지는 분석가의 판단에 맡김

위의 데이터에서 우리는 결국 Y가 생존 여부가 되는 것에만 관심이 있기에 이를 추출해서 분석

minlen = 최소 부분 집합의 크기 supp = 최소지지도 설정 conf = 최소 신뢰도 설정

```
rules <- apriori(titanic, control = list(verbose=F),
                 parameter = list(minlen=2, supp=0.005, conf=0.8),#minlen = lhs
                 appearance = list(rhs=c("Survived=No", "Survived=Yes"),#     Y
                                   default="lhs"))

#   (lift)
rules.sorted <- sort(rules, by="lift")
#
inspect(rules.sorted)
```

```
##      lhs                                    rhs            support confidence
## [1]  {Class=2nd, Age=Child}              => {Survived=Yes} 0.01090 1.000
## [2]  {Class=2nd, Sex=Female, Age=Child}  => {Survived=Yes} 0.00591 1.000
## [3]  {Class=1st, Sex=Female}             => {Survived=Yes} 0.06406 0.972
## [4]  {Class=1st, Sex=Female, Age=Adult}  => {Survived=Yes} 0.06361 0.972
## [5]  {Class=2nd, Sex=Female}             => {Survived=Yes} 0.04225 0.877
## [6]  {Class=Crew, Sex=Female}            => {Survived=Yes} 0.00909 0.870
## [7]  {Class=Crew, Sex=Female, Age=Adult} => {Survived=Yes} 0.00909 0.870
## [8]  {Class=2nd, Sex=Female, Age=Adult}  => {Survived=Yes} 0.03635 0.860
## [9]  {Class=2nd, Sex=Male, Age=Adult}    => {Survived=No}  0.06997 0.917
## [10] {Class=2nd, Sex=Male}               => {Survived=No}  0.06997 0.860
## [11] {Class=3rd, Sex=Male, Age=Adult}    => {Survived=No}  0.17583 0.838
## [12] {Class=3rd, Sex=Male}               => {Survived=No}  0.19173 0.827
##      coverage lift count
## [1]  0.01090  3.10  24
## [2]  0.00591  3.10  13
## [3]  0.06588  3.01 141
## [4]  0.06542  3.01 140
## [5]  0.04816  2.72  93
## [6]  0.01045  2.69  20
## [7]  0.01045  2.69  20
## [8]  0.04225  2.66  80
## [9]  0.07633  1.35 154
## [10] 0.08133  1.27 154
## [11] 0.20990  1.24 387
## [12] 0.23171  1.22 422
```

생존자 중에서 3등급 객실이 없는 것을 보니 3등급 객실 사람들은 대부분 사망했다는 것을 확인 할 수 있다

중복제거

is.subset을 통하여 부분 함수인지 테스트 이 때 정렬이 되어 있어야 더 큰 측도의 조건이 살아남음 => 조건이 다른 조건의 부분 집합인데 측도가 더 작다면 그것을 없애는 방식

```
subset.matrix <- is.subset(rules.sorted, rules.sorted)#
subset.matrix[lower.tri(subset.matrix, diag=T)] <- F#           F
redundant <- colSums(subset.matrix, na.rm = T) >= 1#
which(redundant)#
```

```
##  {Class=2nd,Sex=Female,Age=Child,Survived=Yes}
##                                             2
##  {Class=1st,Sex=Female,Age=Adult,Survived=Yes}
##                                             4
## {Class=Crew,Sex=Female,Age=Adult,Survived=Yes}
##                                             7
##  {Class=2nd,Sex=Female,Age=Adult,Survived=Yes}
##                                             8
```

```
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)
```

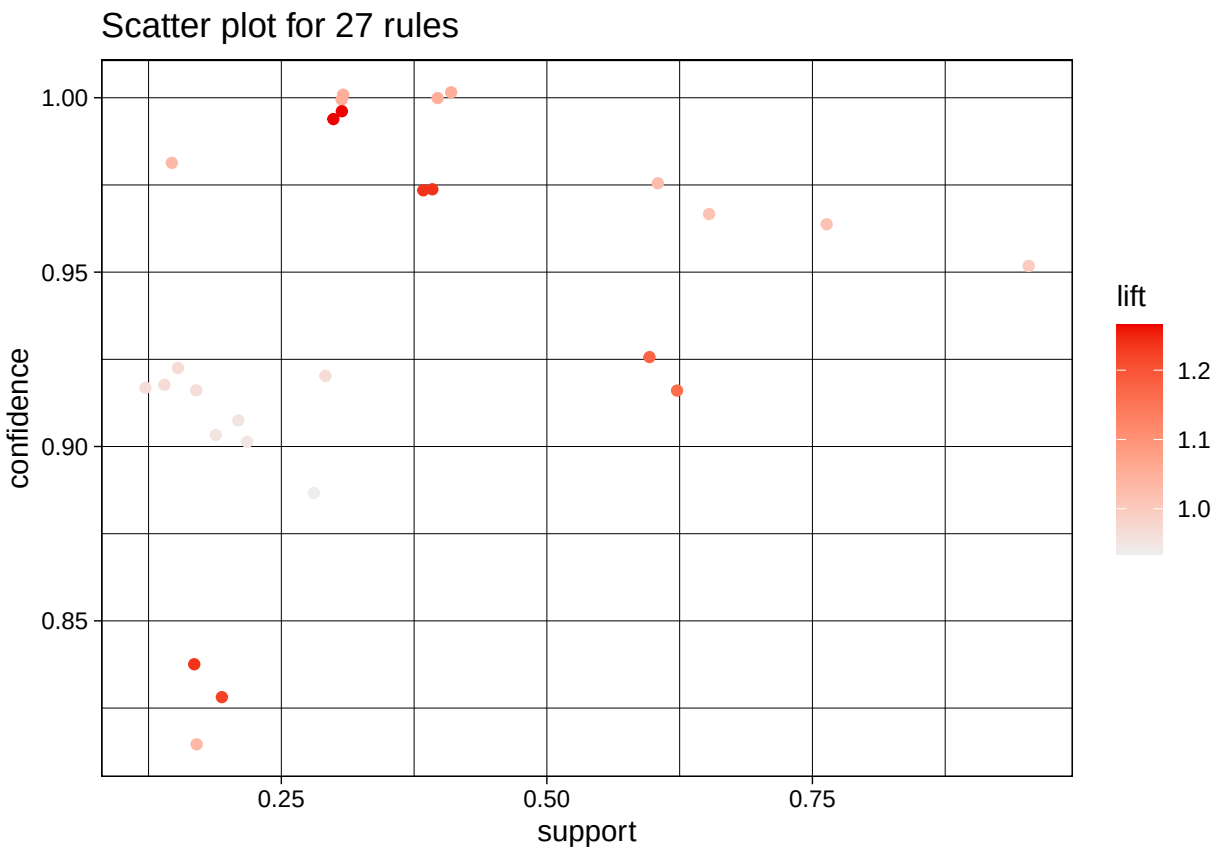
```
##     lhs                                rhs            support confidence
## [1] {Class=2nd, Age=Child}          => {Survived=Yes} 0.01090 1.000
## [2] {Class=1st, Sex=Female}         => {Survived=Yes} 0.06406 0.972
## [3] {Class=2nd, Sex=Female}         => {Survived=Yes} 0.04225 0.877
## [4] {Class=Crew, Sex=Female}        => {Survived=Yes} 0.00909 0.870
## [5] {Class=2nd, Sex=Male, Age=Adult} => {Survived=No}  0.06997 0.917
## [6] {Class=2nd, Sex=Male}           => {Survived=No}  0.06997 0.860
## [7] {Class=3rd, Sex=Male, Age=Adult} => {Survived=No}  0.17583 0.838
## [8] {Class=3rd, Sex=Male}           => {Survived=No}  0.19173 0.827
##     coverage lift count
## [1] 0.0109   3.10  24
## [2] 0.0659   3.01 141
## [3] 0.0482   2.72  93
## [4] 0.0104   2.69  20
## [5] 0.0763   1.35 154
## [6] 0.0813   1.27 154
## [7] 0.2099   1.24 387
## [8] 0.2317   1.22 422
```

연관 규칙의 시각화

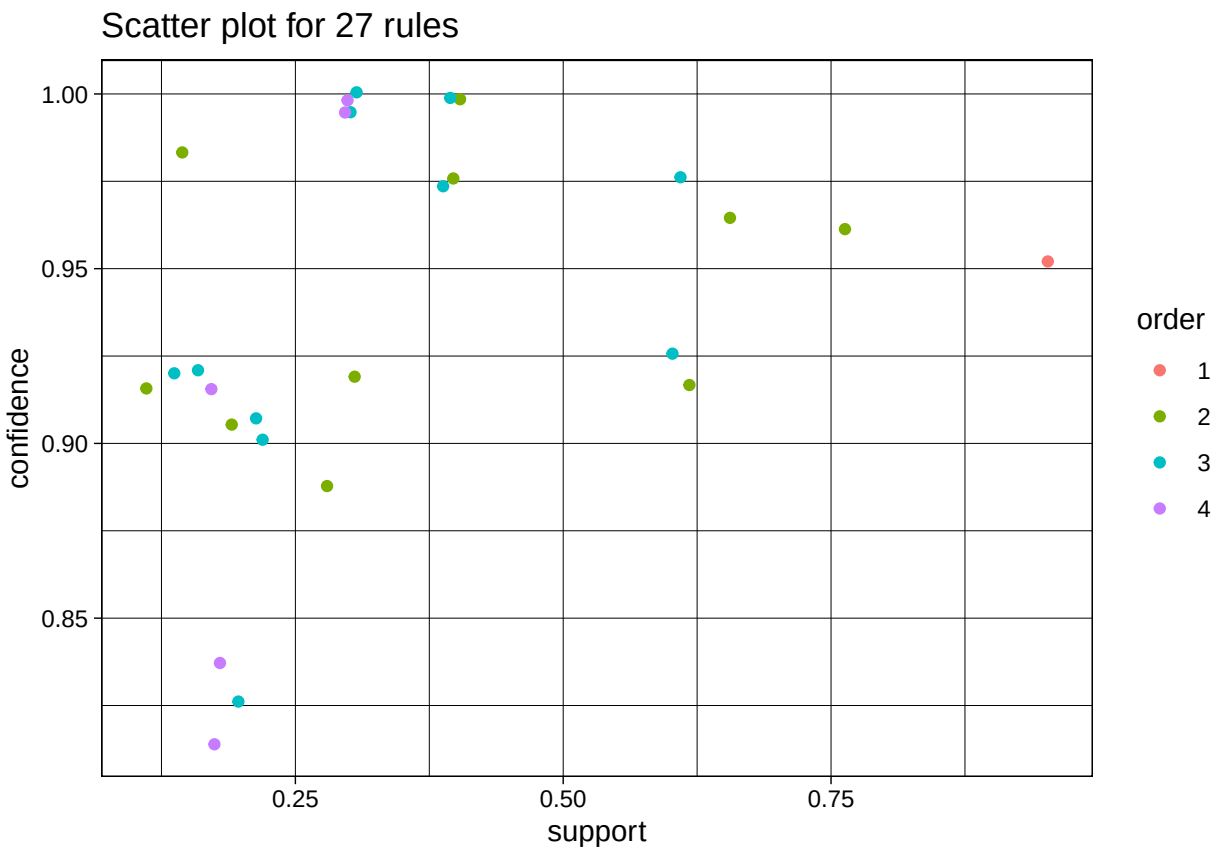
1

3가지 측도를 모두 표현하기는 힘들기에 디폴트인 지지도와 신뢰도만 표현

```
library(arulesViz)
plot(rules.all) #     : measure=c("support", "confidence"), shading="lift"
```

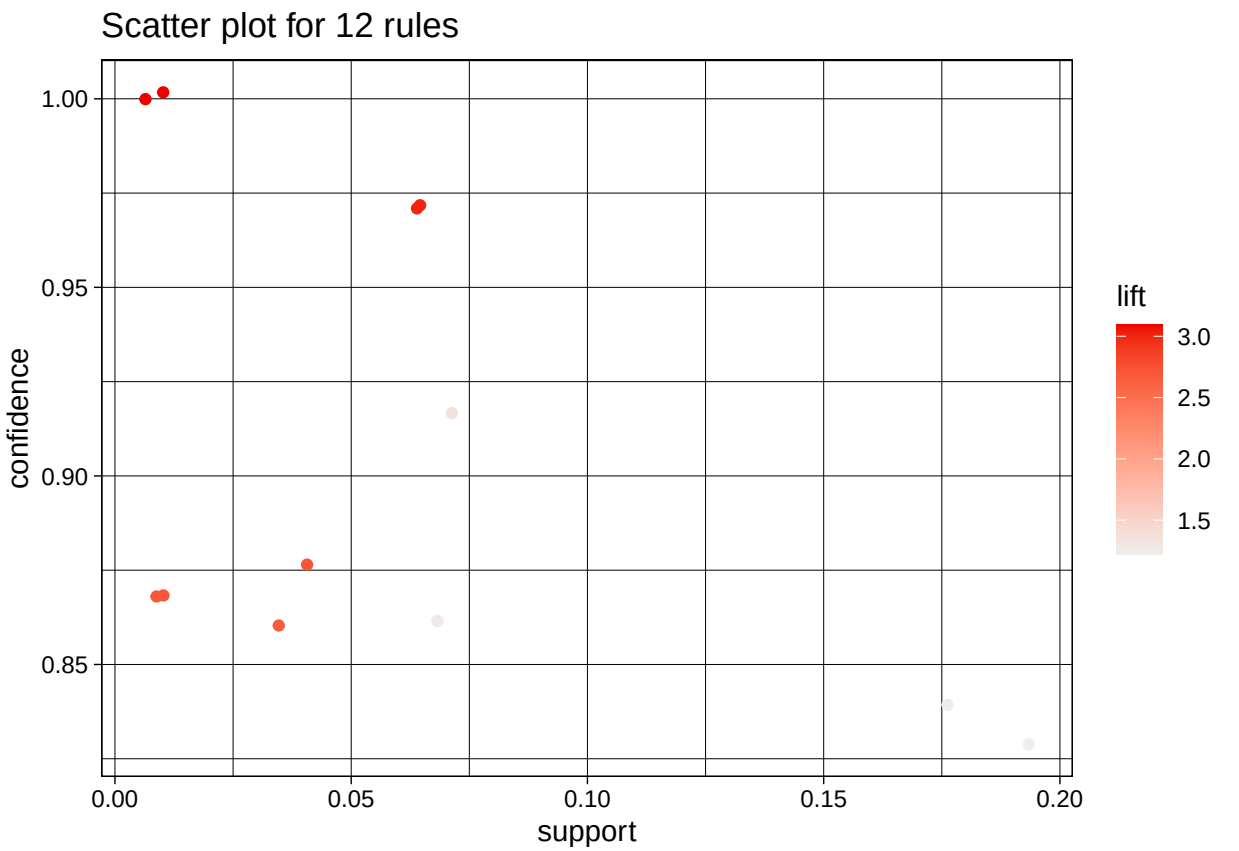



```
plot(rules.all, shading="order") #
```

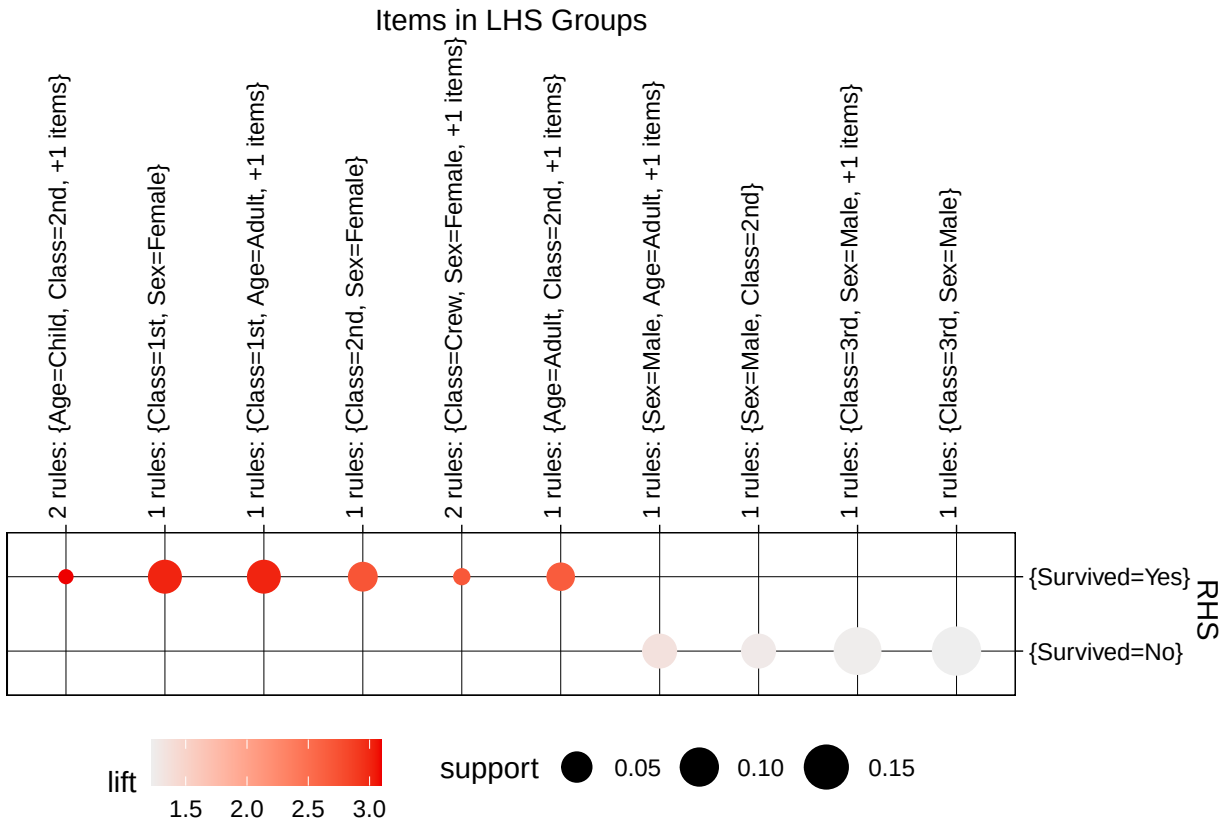


2

```
plot(rules.sorted) # 12
```



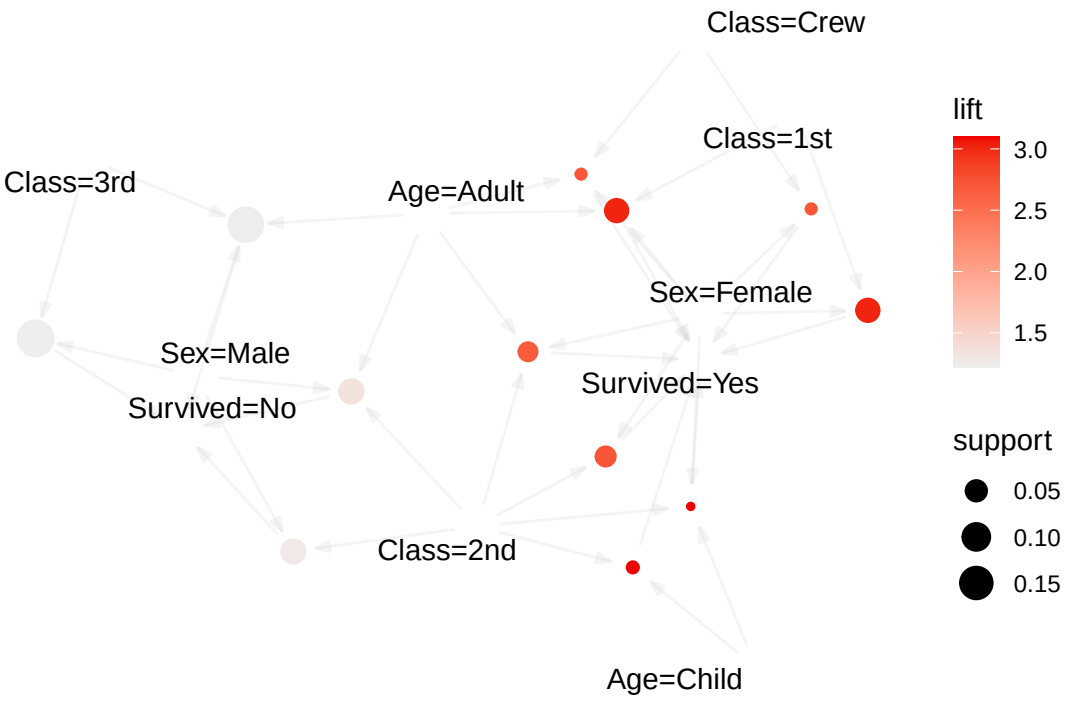

```
plot(rules.sorted, method="grouped")
```



의미 파악 자체는 이 그래프가 조금 더 쉬움

3

```
plot(rules.sorted, method="graph")
```

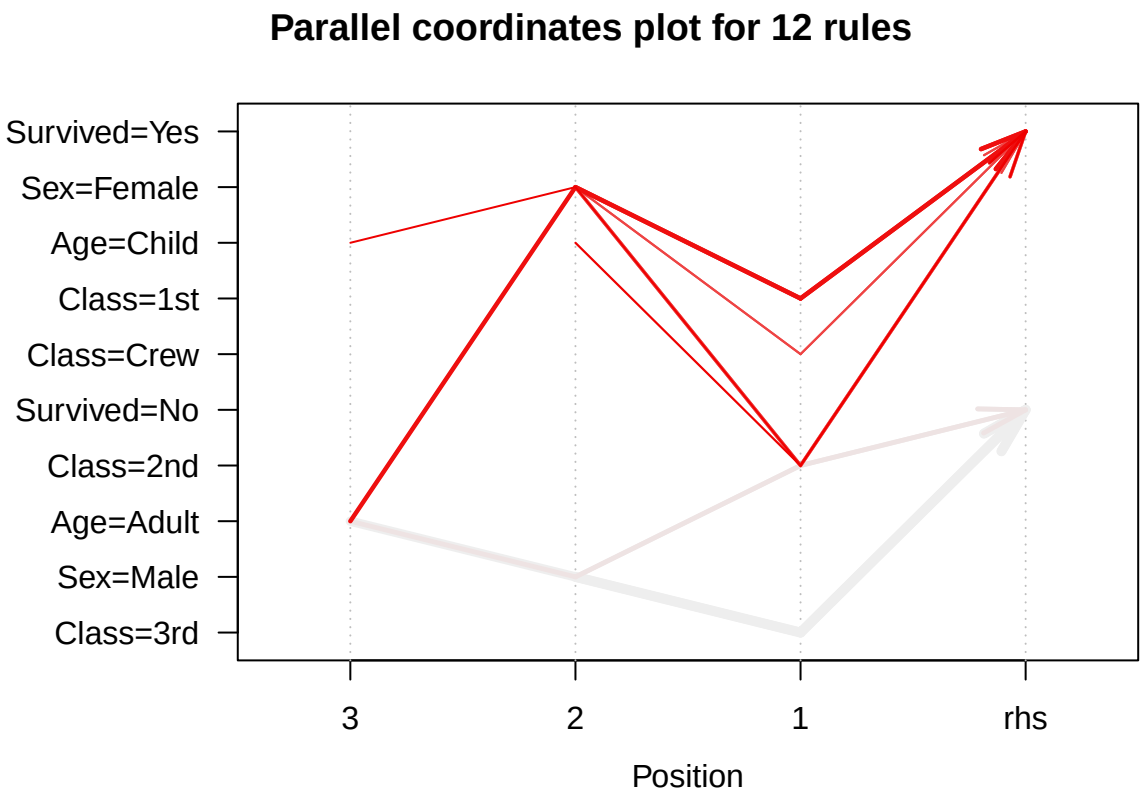


상당히 해석이 난해한 것을 확인 할 수 있음

4

평행좌표그림으로 x축은 조건을 거쳐오는 횟수임

```
plot(rules.sorted, method="paracoord", control=list(reorder=TRUE))
```



5

대화식 그림, 선택된 규칙을 조사하거나, 줌인, 필터링 등을 할 수 있음(코드 에러 뜸….)

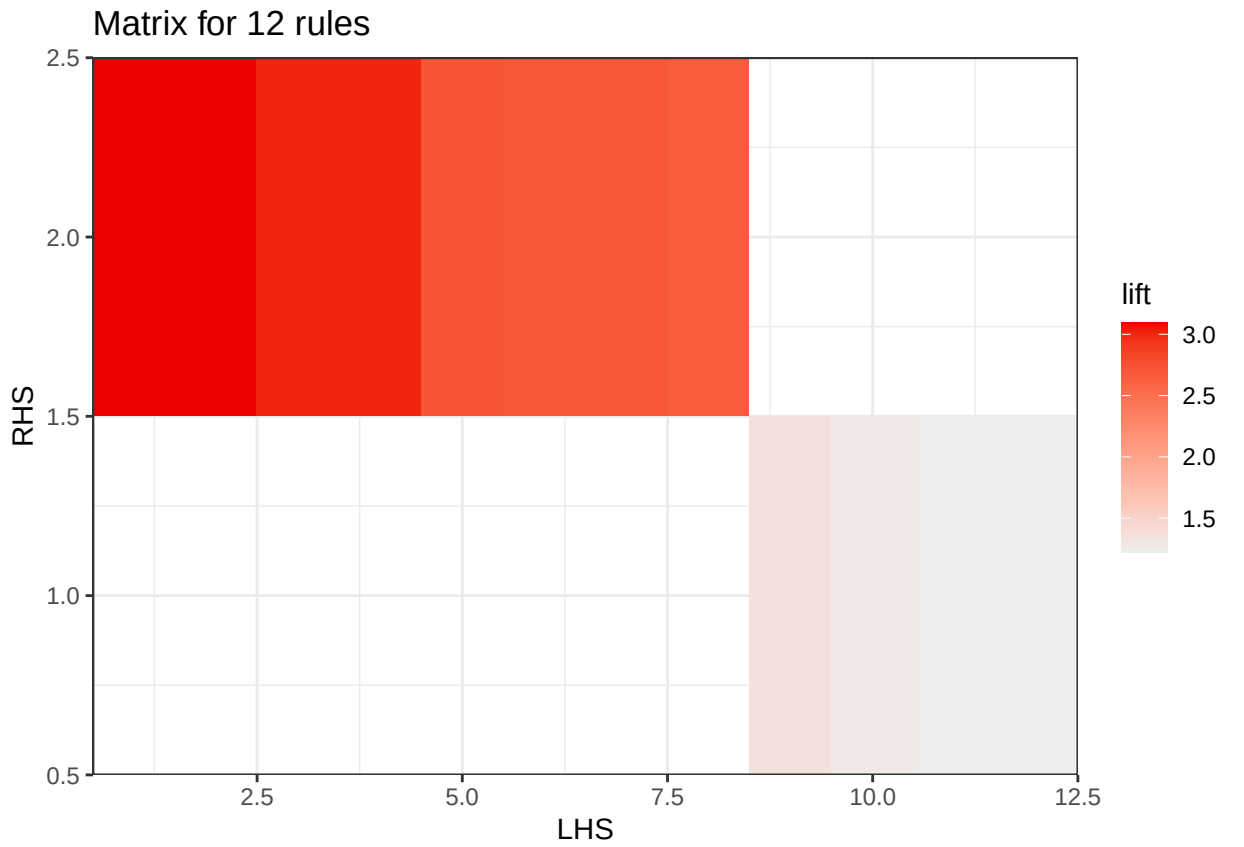
```
#plot(rules.sorted, measure=c("support", "lift"), shading="confidence", interactive=TRUE)
```

6

행렬-기반 시각화

```
plot(rules.sorted, method="matrix", measure="lift")
```

```
## Itemsets in Antecedent (LHS)
##  [1] "{Class=2nd,Age=Child}"                "{Class=2nd,Sex=Female,Age=Child}"
##  [3] "{Class=1st,Sex=Female}"               "{Class=1st,Sex=Female,Age=Adult}"
##  [5] "{Class=2nd,Sex=Female}"               "{Class=Crew,Sex=Female}"
##  [7] "{Class=Crew,Sex=Female,Age=Adult}" "{Class=2nd,Sex=Female,Age=Adult}"
##  [9] "{Class=2nd,Sex=Male,Age=Adult}"       "{Class=2nd,Sex=Male}"
## [11] "{Class=3rd,Sex=Male,Age=Adult}"       "{Class=3rd,Sex=Male}"
## Itemsets in Consequent (RHS)
## [1] "{Survived=No}"  "{Survived=Yes}"
```

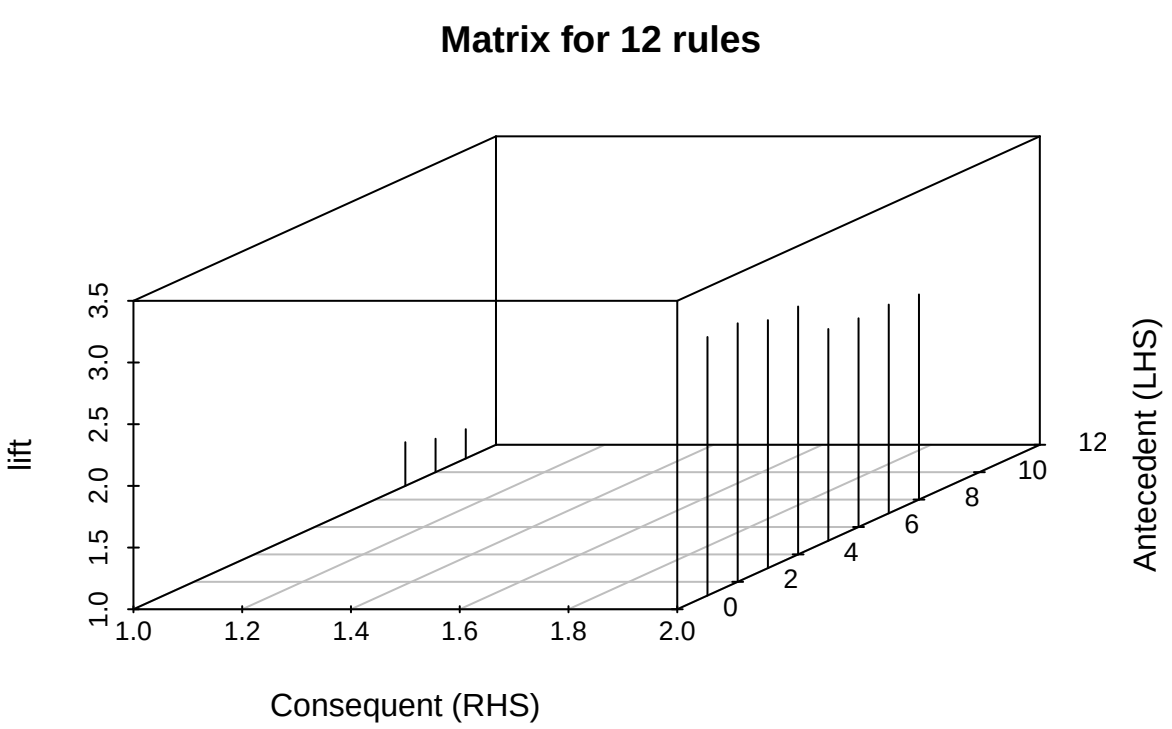


7

3D

```
plot(rules.sorted, method="matrix3D", measure="lift")
```

```
## Itemsets in Antecedent (LHS)
##  [1] "{Class=2nd,Age=Child}"              "{Class=2nd,Sex=Female,Age=Child}"
##  [3] "{Class=1st,Sex=Female}"             "{Class=1st,Sex=Female,Age=Adult}"
##  [5] "{Class=2nd,Sex=Female}"             "{Class=Crew,Sex=Female}"
##  [7] "{Class=Crew,Sex=Female,Age=Adult}" "{Class=2nd,Sex=Female,Age=Adult}"
##  [9] "{Class=2nd,Sex=Male,Age=Adult}"     "{Class=2nd,Sex=Male}"
## [11] "{Class=3rd,Sex=Male,Age=Adult}"     "{Class=3rd,Sex=Male}"
## Itemsets in Consequent (RHS)
## [1] "{Survived=No}"  "{Survived=Yes}"
```


Matrix for 12 rules
lift
1.0
1.5
2.0
2.5
3.0
3.5
1.0
1.2
1.4
1.6
1.8
2.0
Consequent (RHS)
0
2
4
6
8
10
12
Antecedent (LHS)