제 13회 KRUG Meetup

2014/01/24

association rule using arules

R PACKAGES DEPENDENCE

유 충 현

AGENDA

- association rules
- arules packages & arulesViz package
- packages dependence
- packages dependence analysis with arules

association rules 개요

- association rule (연관 규칙)
  - 구매한 아이템들 간의 유용한 연관 패턴을 찾아내는 마이닝 방법
- 활용 예
  - 장바구니 분석 : “기저귀를 구매하는 남성이 맥주를 함께 구매한다.”
    - buys(x, “기저귀”) → buys(y, “맥주”)

association rules 데이터 구조

Transaction ID	Items
001	**러닝 머신, 운동화, 훌라후프, 트레이닝 복**
002	**러닝 머신, 운동화, 트레이닝 복**
003	**운동화, 트레이닝 복, 훌라후프**
004	**러닝 머신, 운동화, 훌라후프**
005	**트레이닝 복**
006	**러닝 머신, 운동화, 줄넘기**

Transaction : 고객의 제품 구매 단위로 Item들의 목록으로 구성됨 (예: 구매한 장바구니)

Items : 구매한 상품

association rules measure

Measure	의 미
Support **(지지도)**	$Support(A \Rightarrow B) = Pr(A \cap B)$
	전체 거래 중 ***A***와 ***B***를 함께 구매한 거래의 비율
Confidence **(신뢰도)**	$Confidence(A \Rightarrow B) = Pr(A \cap B)/Pr(A)$
	항목 ***A*** 거래 중 항목 ***B***가 포함된 거래의 비율
Lift **(향상도)**	$Lift(A \Rightarrow B) = Pr(A \cap B)/(Pr(A) \cdot Pr(B))$
	항목 ***A*** 거래 중 항목 ***B***가 포함된 거래와 ***B***를 구매한 거래와의 비율

association rules measure

▪ Support (지지도)

$$Supp(A \Rightarrow B) = \Pr(A \cap B) = \frac{n(A \cap B)}{n(Total)}$$

* 전체 거래 중 A와 B를 함께 구매한 거래의 비율

Transaction	Items
001	러닝 머신, 운동화, 훌라후프, 트레이닝복
002	러닝 머신, 운동화, 트레이닝복
003	운동화, 트레이닝 복, 훌라후프
004	러닝 머신, 운동화, 훌라후프
005	트레이닝 복
006	러닝 머신, 운동화, 줄넘기

Support(러닝 머신, 운동화) =

러닝머신과 운동화를 함께 구매한 Transaction 수/
전체 Transaction 수 = 4/6 = 0.67

association rules measure

▪ Confidence (신뢰도)

$$Conf\left(A \Rightarrow B\right) = \Pr(B \mid A) = \frac{\Pr(A \cap B)}{\Pr(A)} = \frac{n(A \cap B)}{n(A)}$$

* 항목 A 거래 중 항목 B가 포함된 거래의 비율 ➔ A를 구매하고 B를 구매할 확률

Transaction	Items
001	러닝 머신, 운동화, 훌라후프, 트레이닝복
002	러닝 머신, 운동화, 트레이닝복
003	운동화, 트레이닝 복, 훌라후프
004	러닝 머신, 운동화, 훌라후프
005	트레이닝 복
006	러닝 머신, 운동화, 줄넘기

Confidence(러닝 머신, 운동화) =

러닝 머신과 운동화를 함께 구매한 Transaction 수/
러닝 머신 구매 Transaction 수 = 4/4 = 1

association rules measure

- **Lift (향상도)**

$$Lift(A \Rightarrow B) = \frac{\Pr(B \mid A)}{\Pr(B)} = \frac{\Pr(A \cap B)}{\Pr(A) \cdot \Pr(B)} = \frac{n(Total) \cdot n(A \cap B)}{n(A) \cdot n(B)}$$

* 항목 A 거래 중 항목 B가 포함된 거래와 B를 구매한 거래와의 비율

Transaction	Items
001	러닝 머신, 운동화, 훌라후프, 트레이닝복
002	러닝 머신, 운동화, 트레이닝복
003	운동화, 트레이닝 복, 훌라후프
004	러닝 머신, 운동화, 훌라후프
005	트레이닝 복
006	러닝 머신, 운동화, 줄넘기

Lift(러닝 머신, 운동화) =

러닝 머신 거래 항목 중 운동화 포함 비율 / 운동화 구매 비율 = 1/(5/6) = 1.25

association rules measure

▪ Lift 해석

$$Lift(A \Rightarrow B) = \frac{\Pr(B \mid A)}{\Pr(B)} = \frac{\Pr(A \cap B)}{\Pr(A) \cdot \Pr(B)} = corr_{A,B}$$

Lift	의미	해석	예제
1	***A, B*가 서로 독립적인 관계**	***A*와 *B*는 상관관계가 없음**	**과자와 후추**
Lift>1	***A, B*가 서로 양의 상관관계**	***A*의 발생은 *B*의 발생에 긍정적으로 상관**	**빵과 버터**
Lift<1	***A, B*가 서로 음의 상관관계**	***A*의 발생은 *B*의 발생에 부정적으로 상관**	**지사제와 변비약**

예제)

	Game	Not Game	Sum(Row)
Video	*4,000*	*3,500*	*7,500*
Not Video	*2,000*	*500*	*2,500*
Sum(Col)	*6,000*	*4,000*	*10,000*

Supp(Game→Video)=4000/10000=0.4

Conf(Game→Video)=4000/6000=0.67

Lift(Game→Video)=0.67/0.75=0.89

association rules measure

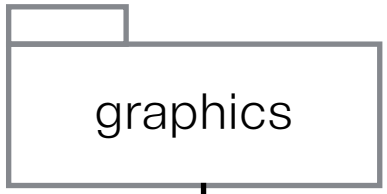
▪ 연관규칙 분석 순서

최소지지도와 최소 신뢰도를 이용하여 발견된 연관 규칙을 항목들 간의 상관 관계(향상도)를 고려하여 최종적으로 선택함

arules/arulesViz

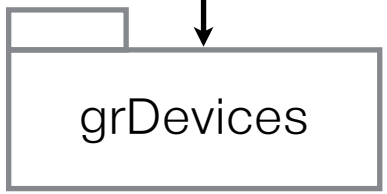
- arules package
  - 트랜잭션 데이터와 패턴을 나타내는 조작 및 분석을위한 인프라를 제공
- arulesViz package
  - association rules과 itemsets에 대한 다양한 시각화 기술 제공
  - 규칙 탐사를 위한 몇 개의 interactive visualizations
  - extends package arules

packages dependence



R functions for base graphics

graphics package는 grDevices package에 종속되어 있음



The R Graphics Devices and Support for Colours and Fonts

packages dependence

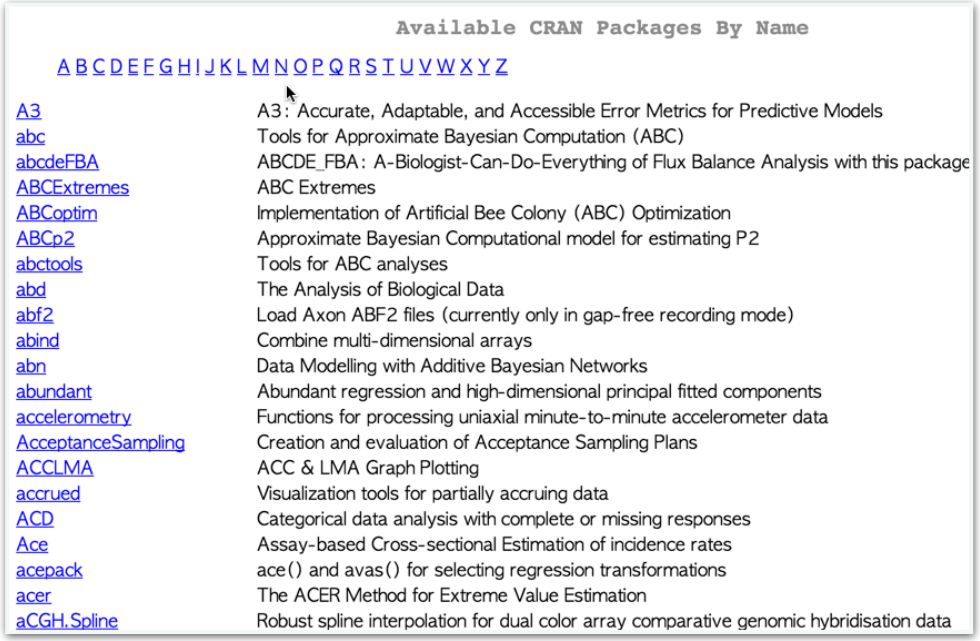
- Depends
  - package를 로드하기 전에 자동으로 종속 패키지를 로드함
- Imports
  - NAMESPACE file에 기술해서 패키지 build.
  - ::, ::: 연산자를 이용하여 종속 패키지의 함수를 호출함
- LinkingTo
  - C++ 등의 라이브러리에 종속되어, 종속 패키지의 헤더파일(*.h)을 참조
- Suggests
  - 종속 패키지의 examples/tests/vignettes를 사용함

packages dependence analysis with arules

- Raw Data
  - CRAN home pages
- Data Import
  - RCurl package : CRAN home page에서 해당 html 입수
- Data Manipulation
  - XML package : html tag에서 분석에 필요한 데이터 추출
- Data Analysis
  - arules package : transaction object로 데이터 변환, apriori 분석 수행
  - arulesViz packages : visualization
- Etc : parallel package를 이용해서 병렬로 데이터 수집

Analytics - Raw Data

- http://cran.nexr.com/web/packages/available_packages_by_name.html
- CRAN 패키지 목록

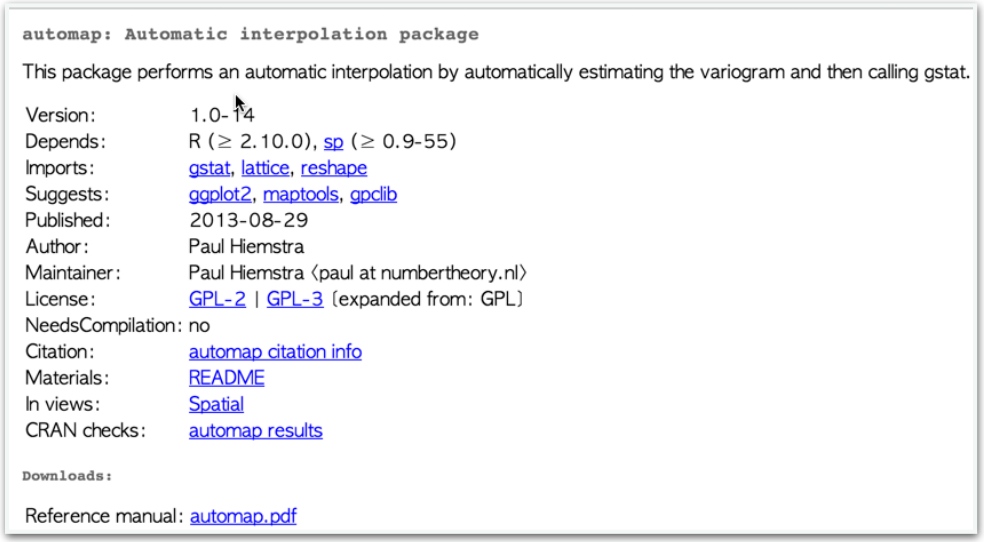


Available CRAN Packages By Name

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A3	A3: Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
abc	Tools for Approximate Bayesian Computation (ABC)
abcdeFBA	ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
ABCExtremes	ABC Extremes
ABCoptim	Implementation of Artificial Bee Colony (ABC) Optimization
ABCp2	Approximate Bayesian Computational model for estimating P2
abctools	Tools for ABC analyses
abd	The Analysis of Biological Data
abf2	Load Axon ABF2 files (currently only in gap-free recording mode)
abind	Combine multi-dimensional arrays
abn	Data Modelling with Additive Bayesian Networks
abundant	Abundant regression and high-dimensional principal fitted components
accelerometry	Functions for processing uniaxial minute-to-minute accelerometer data
AcceptanceSampling	Creation and evaluation of Acceptance Sampling Plans
ACCLMA	ACC & LMA Graph Plotting
accrued	Visualization tools for partially accruing data
ACD	Categorical data analysis with complete or missing responses
Ace	Assay-based Cross-sectional Estimation of incidence rates
acepack	ace() and avas() for selecting regression transformations
acer	The ACER Method for Extreme Value Estimation
aCGH.Spline	Robust spline interpolation for dual color array comparative genomic hybridisation data

- http://cran.nexr.com/web/packages/*/index.html
- 개별 패키지 정보



automap: Automatic interpolation package

This package performs an automatic interpolation by automatically estimating the variogram and then calling gstat.

Version:	1.0-14
Depends:	R (≥ 2.10.0), sp (≥ 0.9-55)
Imports:	gstat, lattice, reshape
Suggests:	ggplot2, maptools, gpclib
Published:	2013-08-29
Author:	Paul Hiemstra
Maintainer:	Paul Hiemstra ⟨paul at numbertheory.nl⟩
License:	GPL-2 | GPL-3 [expanded from: GPL]
NeedsCompilation:	no
Citation:	automap citation info
Materials:	README
In views:	Spatial
CRAN checks:	automap results

Downloads:

Reference manual: automap.pdf

Analytics - Import Datas

- CRAN 패키지 목록 가져 오기
  - RCurl - 데이터 긁어 오기
  - XML - 패키지 목록 추출하기

```
#------------------------------------------------------------------------------
# 2. CRAN에서 패키지 정보 가져 오기
#------------------------------------------------------------------------------
url <- "http://cran.nexr.com/web/packages/available_packages_by_name.html"

#------------------------------------------------------------------------------
# HTML 페이지 불러오기
#------------------------------------------------------------------------------
html.content <- getURL(url)

#------------------------------------------------------------------------------
# table 태그 추출하기
#------------------------------------------------------------------------------
tables <- getNodeSet(htmlParse(html.content),"//table") [[1]]

#------------------------------------------------------------------------------
# table 태그에서 TR 태그를 데이터 프레임 레코드로 추출 후 첫 변수만 가져오기
# package의 이름을 가져 오는 로직
#------------------------------------------------------------------------------
xt <- readHTMLTable(tables, stringsAsFactors = FALSE)[, 1]
head(xt)

#------------------------------------------------------------------------------
# NA 필터링
# NA는 패키지명의 알파벳을 나타내는 첫 줄에서 발생
#------------------------------------------------------------------------------
xt <- xt[!is.na(xt)]
```

- 개별 패키지의 종속 정보 가져 오기
  - RCurl - 데이터 긁어 오기
  - XML - 패키지 종속 정보 추출하기

```
getImports <- function(pkg, idx="Imports:") {
  library(RCurl)
  library(XML)

  url <- sprintf("http://cran.nexr.com/web/packages/%s/index.html", pkg)

  html.content <- getURL(url)
  tables <- getNodeSet(htmlParse(html.content),"//table") [[1]]
  xt <- readHTMLTable(tables, stringsAsFactors = FALSE)
  xt[xt$V1 %in% idx, "V2"]
}
```

- 병렬 처리로 정보 가져 오기
  - parallel - 병렬 처리

```
#------------------------------------------------------------
# parallel 패키지 함수에서 사용할 core의 개수 지정
#------------------------------------------------------------
cl <- makeCluster(getOption("cl.cores", 2))

#------------------------------------------------------------
# package 이름으로 Imports 패키지의 이름을 가져오기
#------------------------------------------------------------
system.time(
imports <- parSapply(cl, xt, getImports))
```

Analytics - Data Preparation

- 필요 없는 문자열 제거 및 문자열 분할

```
#---------------------------------------------------------------------------
# NA 제거 및 벡터로 생성하기
#---------------------------------------------------------------------------
imports <- do.call("rbind", imports)

#---------------------------------------------------------------------------
# 패키지 이름 전처리 (버전 정보 제거)
#---------------------------------------------------------------------------
# MASS, R.methodsS3 (≥ 1.5.2), plyr(≥1.8), R.utils (≥\n1.27.1), multcomp(≥0.991-7),
# sandwich(≥2.2-6), tm(≥0.5-8.5), lpSolveAPI(≥5.5.0.14), openNLPdata(≥1.5.3-1),
# lme4(≥0.999999-2), network(≥1.4-1-1), TSdbi (≥ 2013.9-1)

delete.str <- "("
delete.str <- paste(delete.str, "\n", sep = "")
delete.str <- paste(delete.str, "[[:blank:]]*", sep = "|")
delete.str <- paste(delete.str, "\\([[:print:]]{1,2}[0-9]{1,6}[\\.-][0-9]{1,6}([\\.-][0-9]{1,3})*([\\.-][0-9]{1,3})*\\)", sep = "|")
delete.str <- paste(delete.str, ")", sep="")

imports <- gsub(delete.str, "", imports)

#---------------------------------------------------------------------------
# 컴마를 기준으로 패키지의 이름을 분함
#---------------------------------------------------------------------------
trans <- strsplit(imports, ",")
```

- transaction 데이터 생성 : arules package

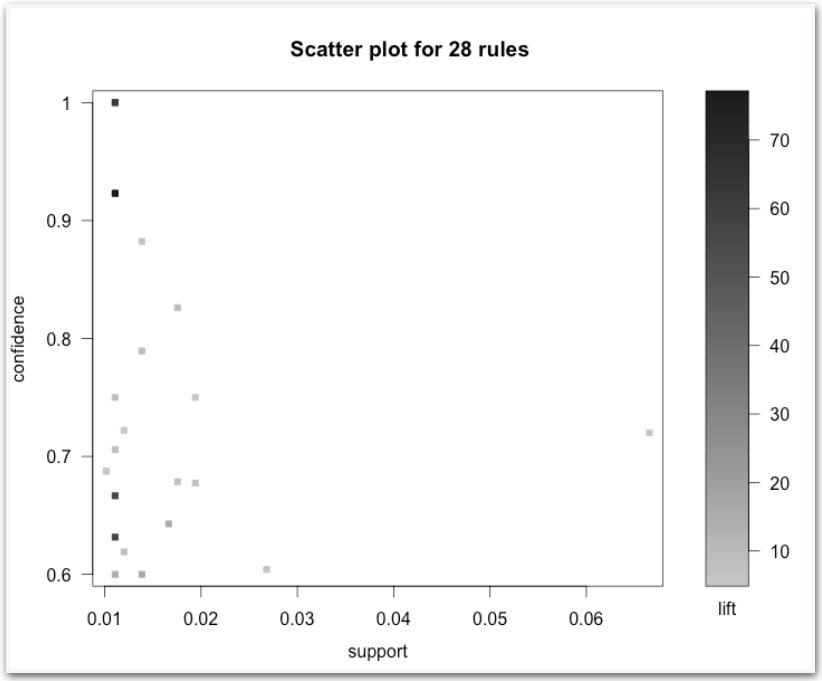
```
#---------------------------------------------------------------------------
# Transaction 객체로 변환
# trans : 대상 패키지를 제외한 Imports 정보를 이용한 transaction
# trans2 : 대상 패키지와 Imports 정보를 이용한 transactio
#---------------------------------------------------------------------------
names(trans) <- row.names(imports)
trans <- as(trans, "transactions")

summary(trans)
image(trans)

imports2 <- paste(row.names(imports), imports, sep=",")
trans2 <- strsplit(imports2, ",")

names(trans2) <- row.names(imports2)
trans2 <- as(trans2, "transactions")

summary(trans2)
image(trans2)
```

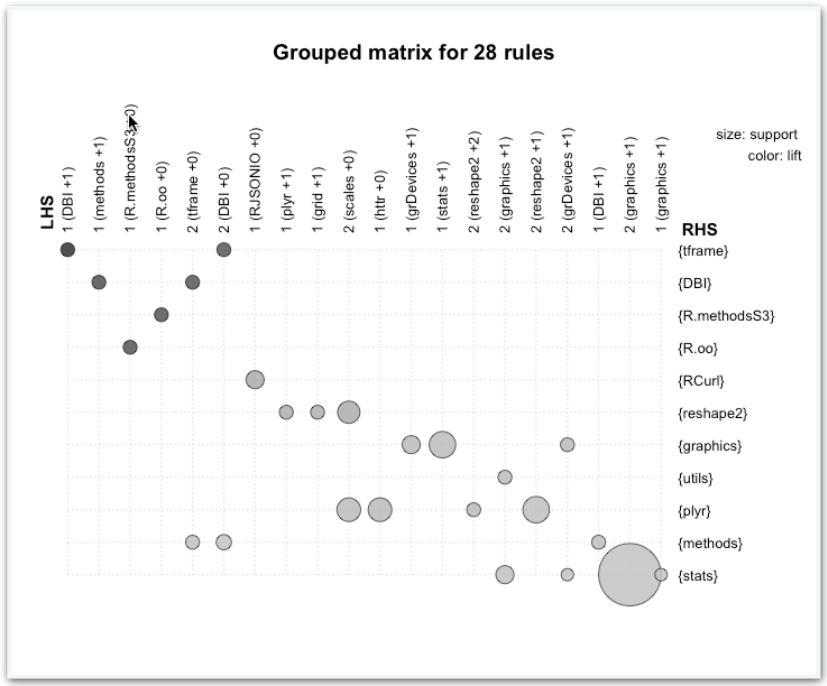
Analytics - association rule analysis using arules

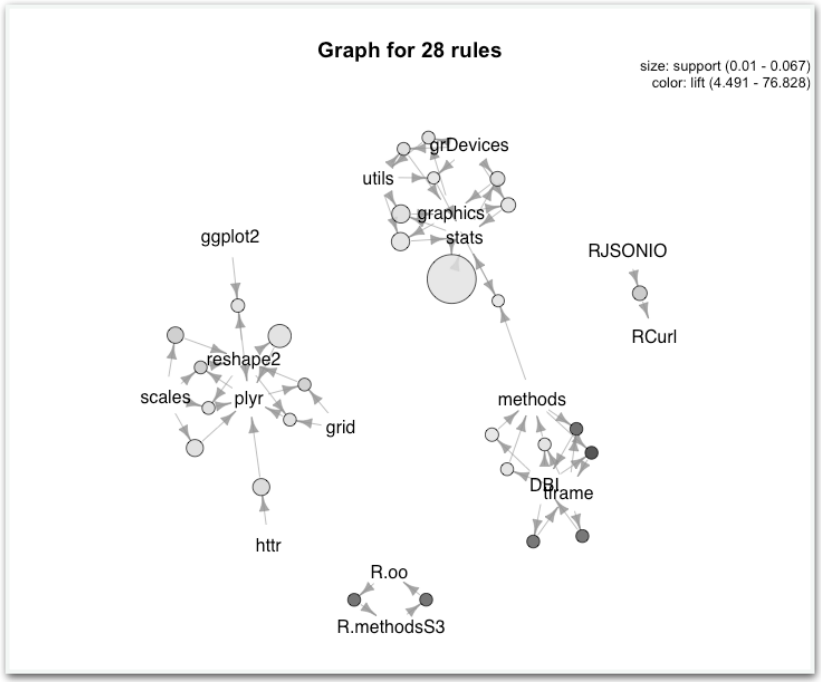
```
> #---------------------------------------------------------------------
> # rule을 생성함
> # support = 0.01, Confidence = 0.6
> #---------------------------------------------------------------------
> rules <- apriori(trans, parameter=list(supp=0.01, conf=0.6, target="rules"))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.6    0.1    1 none FALSE            TRUE    0.01      1     10  rules FALSE

algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[681 item(s), 1082 transaction(s)] done [0.00s].
sorting and recoding items ... [71 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [28 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> summary(rules)
set of 28 rules

rule length distribution (lhs + rhs):sizes
 2  3
12 16

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.571   3.000   3.000

summary of quality measures:
    support         confidence          lift
 Min.   :0.01017   Min.   :0.6000   Min.   : 4.491
 1st Qu.:0.01109   1st Qu.:0.6607   1st Qu.: 6.165
 Median :0.01109   Median :0.6967   Median : 7.686
 Mean   :0.01518   Mean   :0.7491   Mean   :19.216
 3rd Qu.:0.01456   3rd Qu.:0.8402   3rd Qu.:14.827
 Max.   :0.06654   Max.   :1.0000   Max.   :76.828

mining info:
  data ntransactions support confidence
 trans          1082    0.01        0.6
```

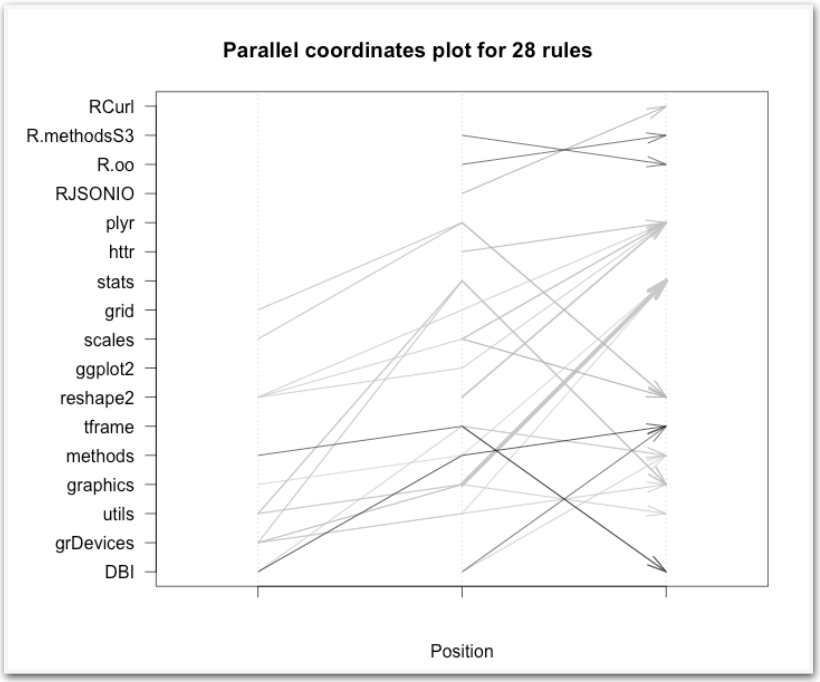
```
> inspect(rules)
   lhs              rhs              support confidence      lift
1  {tframe}      => {DBI}         0.01109057  0.9230769 55.487179
2  {DBI}         => {tframe}      0.01109057  0.6666667 55.487179
3  {tframe}      => {methods}     0.01109057  0.9230769  5.740053
4  {R.oo}        => {R.methodsS3} 0.01109057  1.0000000 56.947368
5  {R.methodsS3} => {R.oo}        0.01109057  0.6315789 56.947368
6  {DBI}         => {methods}     0.01201479  0.7222222  4.491060
7  {RJSONIO}     => {RCurl}       0.01386322  0.6000000 15.834146
8  {httr}        => {plyr}        0.01756007  0.8260870  9.028546
9  {scales}      => {reshape2}    0.01663586  0.6428571 14.491071
10 {scales}      => {plyr}        0.01756007  0.6785714  7.416306
11 {reshape2}    => {plyr}        0.02680222  0.6041667  6.603114
12 {graphics}    => {stats}       0.06654344  0.7200000  4.899623
13 {DBI,
    tframe}      => {methods}     0.01109057  1.0000000  6.218391
14 {methods,
    tframe}      => {DBI}         0.01109057  1.0000000 60.111111
15 {DBI,
    methods}     => {tframe}      0.01109057  0.9230769 76.828402
16 {graphics,
    grDevices}   => {utils}       0.01109057  0.7058824  7.955882
17 {grDevices,
    utils}       => {graphics}    0.01109057  0.7500000  8.115000
18 {graphics,
    grDevices}   => {stats}       0.01386322  0.8823529  6.004440
19 {grDevices,
    stats}       => {graphics}    0.01386322  0.7894737  8.542105
20 {grDevices,
    utils}       => {stats}       0.01016636  0.6875000  4.678459
21 {reshape2,
    scales}      => {plyr}        0.01109057  0.6666667  7.286195
22 {plyr,
    scales}      => {reshape2}    0.01109057  0.6315789 14.236842
23 {ggplot2,
    reshape2}    => {plyr}        0.01201479  0.6190476  6.765753
24 {grid,
    reshape2}    => {plyr}        0.01109057  0.6666667  7.286195
25 {grid,
    plyr}        => {reshape2}    0.01109057  0.6000000 13.525000
26 {graphics,
    utils}       => {stats}       0.01940850  0.7500000  5.103774
27 {stats,
    utils}       => {graphics}    0.01940850  0.6774194  7.329677
28 {graphics,
    methods}     => {stats}       0.01016636  0.6875000  4.678459
>
```

Analytics - visualization using arules









감사합니다