R Code Assoication Analysis

Amin Fesharaki

11/29/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
```

```
data(Zoo, package="mlbench")
head(Zoo)
```

```
##           hair feathers  eggs  milk airborne aquatic predator toothed backbone
## aardvark  TRUE    FALSE FALSE  TRUE    FALSE   FALSE     TRUE    TRUE     TRUE
## antelope  TRUE    FALSE FALSE  TRUE    FALSE   FALSE    FALSE    TRUE     TRUE
## bass     FALSE    FALSE  TRUE FALSE    FALSE    TRUE     TRUE    TRUE     TRUE
## bear      TRUE    FALSE FALSE  TRUE    FALSE   FALSE     TRUE    TRUE     TRUE
## boar      TRUE    FALSE FALSE  TRUE    FALSE   FALSE     TRUE    TRUE     TRUE
## buffalo   TRUE    FALSE FALSE  TRUE    FALSE   FALSE    FALSE    TRUE     TRUE
##          breathes venomous  fins legs  tail domestic catsize   type
## aardvark     TRUE    FALSE FALSE    4 FALSE    FALSE    TRUE mammal
## antelope     TRUE    FALSE FALSE    4  TRUE    FALSE    TRUE mammal
## bass        FALSE    FALSE  TRUE    0  TRUE    FALSE   FALSE   fish
## bear         TRUE    FALSE FALSE    4 FALSE    FALSE    TRUE mammal
## boar         TRUE    FALSE FALSE    4  TRUE    FALSE    TRUE mammal
## buffalo      TRUE    FALSE FALSE    4  TRUE    FALSE    TRUE mammal
```

```
try(trans <- as(Zoo, "transactions"))
```

```
## Warning: Column(s) 13 not logical or factor. Applying default discretization
## (see '? discretizeDF').
```

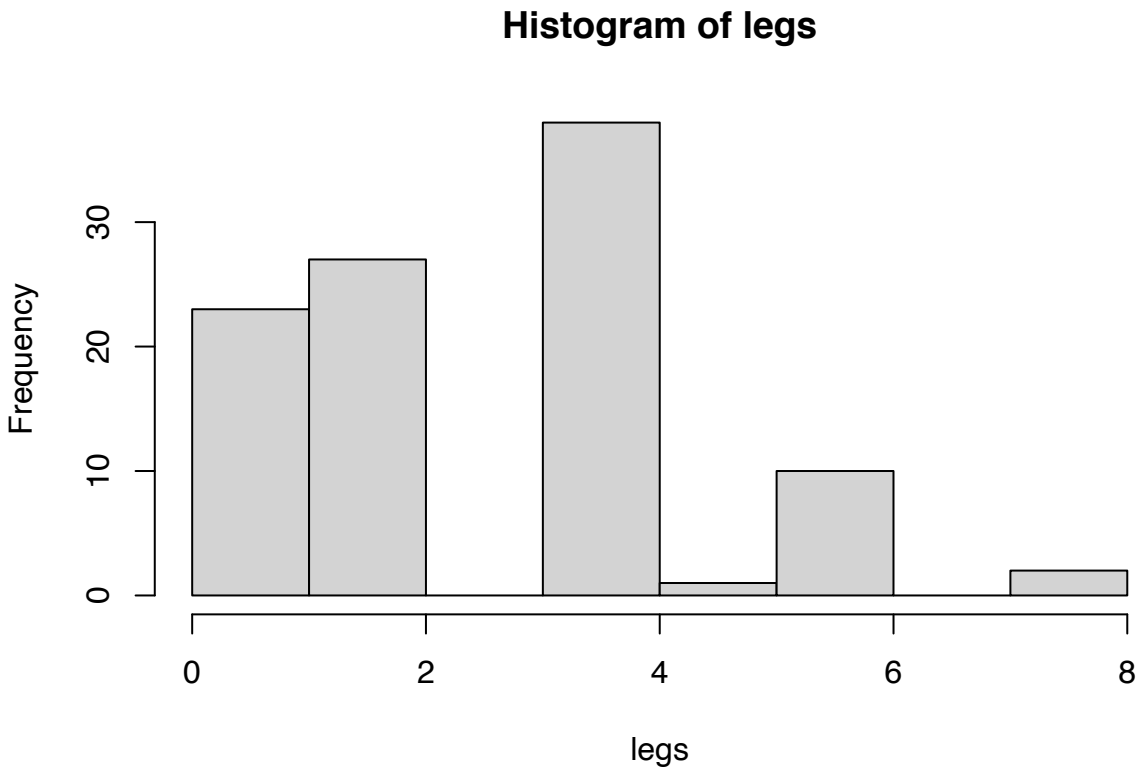
```
#' What is column 13?
colnames(Zoo)[13]
```

```
## [1] "legs"
```

```
legs <- Zoo[["legs"]]
summary(legs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   4.000   2.842   4.000   8.000
```

```
hist(legs)
```




```
table(legs)
```

```
## legs
##  0  2  4  5  6  8
## 23 27 38  1 10  2
```

```
#' Possible solution: Make legs into has/does not have legs
has_legs <- legs>0
has_legs
```

```
##   [1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
##  [13] FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
##  [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
##  [37]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [49]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [61] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [73]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [85]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
##  [97]  TRUE  TRUE  TRUE FALSE  TRUE
```

```
table(has_legs)
```

```
## has_legs
```

```
## FALSE  TRUE
##    23    78
```

```
Zoo[["legs"]] <- has_legs
```

```
#'  Convert data into a set of transactions
trans <- as(Zoo, "transactions")
trans
```

```
## transactions in sparse format with
##  101 transactions (rows) and
##  23 items (columns)
```

```
#' ## Inspect Transactions
summary(trans)
```

```
## transactions as itemMatrix in sparse format with
##  101 rows (elements/itemsets/transactions) and
##  23 columns (items) and a density of 0.3611709
##
## most frequent items:
## backbone breathes     legs     tail  toothed  (Other)
##       83       80       78       75       61      462
##
## element (itemset/transaction) length distribution:
## sizes
##  3  4  5  6  7  8  9 10 11 12
##  3  2  6  5  8 21 27 25  3  1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   8.000   9.000   8.307  10.000  12.000
##
## includes extended item information - examples:
##     labels variables levels
## 1     hair      hair   TRUE
## 2 feathers  feathers   TRUE
## 3     eggs      eggs   TRUE
##
## includes extended transaction information - examples:
##   transactionID
## 1      aardvark
## 2      antelope
## 3          bass
```

```
#' Look at created items. They are still called column names since the transactions are actually stored
colnames(trans)
```

```
##  [1] "hair"               "feathers"           "eggs"
##  [4] "milk"               "airborne"           "aquatic"
##  [7] "predator"           "toothed"            "backbone"
## [10] "breathes"           "venomous"           "fins"
## [13] "legs"               "tail"               "domestic"
## [16] "catsize"            "type=mammal"        "type=bird"
## [19] "type=reptile"       "type=fish"          "type=amphibian"
## [22] "type=insect"        "type=mollusc.et.al"
```

```
#' Compare with the original features (column names) from Zoo
colnames(Zoo)
```

```
##  [1] "hair"     "feathers" "eggs"     "milk"     "airborne" "aquatic"
##  [7] "predator" "toothed"  "backbone" "breathes" "venomous" "fins"
## [13] "legs"     "tail"     "domestic" "catsize"  "type"
```

```
#' Look at a (first) few transactions as a matrix. 1 indicates the presence of an item.
as(trans, "matrix")[1:3,]
```

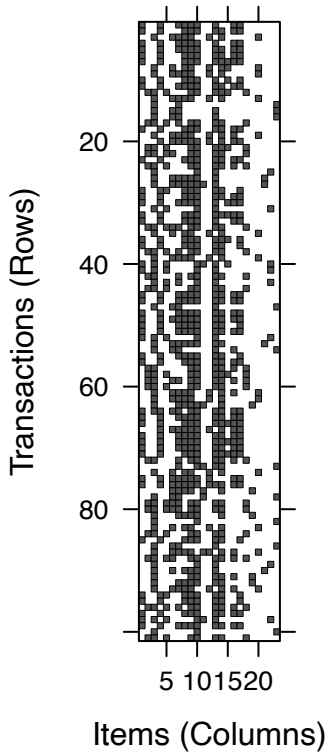
```
##           hair feathers  eggs  milk airborne aquatic predator toothed backbone
## aardvark  TRUE    FALSE FALSE  TRUE    FALSE   FALSE     TRUE    TRUE     TRUE
## antelope  TRUE    FALSE FALSE  TRUE    FALSE   FALSE    FALSE    TRUE     TRUE
## bass     FALSE    FALSE  TRUE FALSE    FALSE    TRUE     TRUE    TRUE     TRUE
##          breathes venomous  fins  legs  tail domestic catsize type=mammal
## aardvark     TRUE    FALSE FALSE  TRUE FALSE    FALSE    TRUE         TRUE
## antelope     TRUE    FALSE FALSE  TRUE  TRUE    FALSE    TRUE         TRUE
## bass        FALSE    FALSE  TRUE FALSE  TRUE    FALSE   FALSE        FALSE
##          type=bird type=reptile type=fish type=amphibian type=insect
## aardvark     FALSE        FALSE     FALSE          FALSE       FALSE
## antelope     FALSE        FALSE     FALSE          FALSE       FALSE
## bass         FALSE        FALSE      TRUE          FALSE       FALSE
##          type=mollusc.et.al
## aardvark              FALSE
## antelope              FALSE
## bass                  FALSE
```

```
#' Look at the transactions as sets of items
inspect(trans[1:3])
```

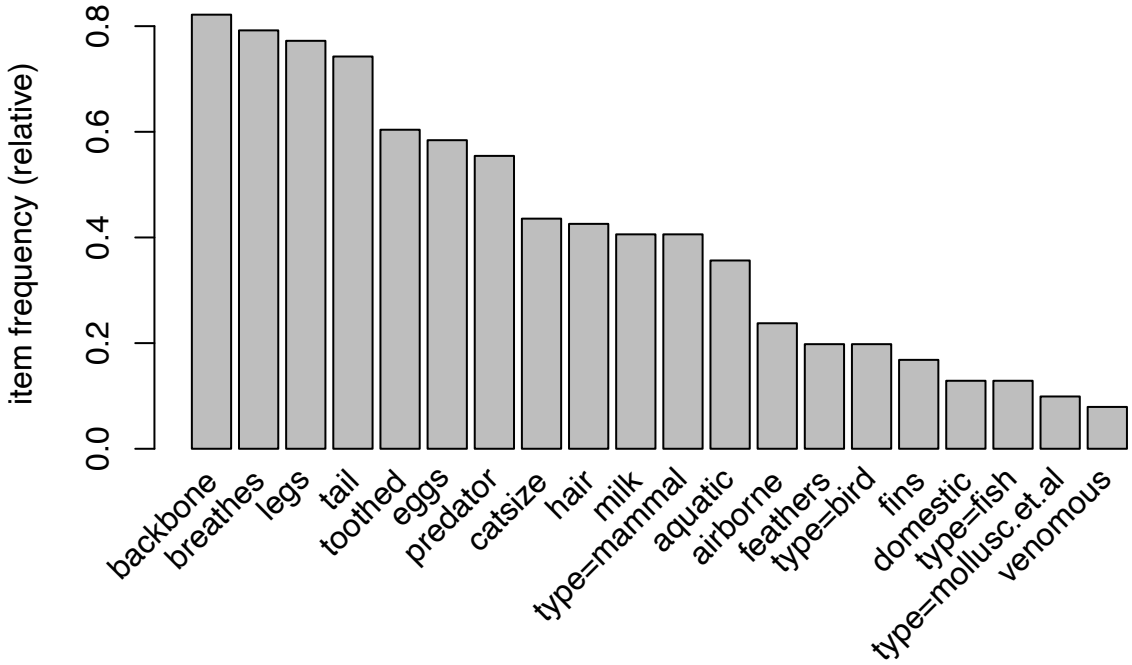
```
##     items         transactionID
## [1] {hair,
##      milk,
##      predator,
##      toothed,
##      backbone,
##      breathes,
##      legs,
##      catsize,
##      type=mammal}      aardvark
## [2] {hair,
##      milk,
##      toothed,
##      backbone,
##      breathes,
##      legs,
##      tail,
##      catsize,
##      type=mammal}      antelope
## [3] {eggs,
##      aquatic,
##      predator,
##      toothed,
##      backbone,
##      fins,
```

```
##      tail,
##      type=fish}        bass
```

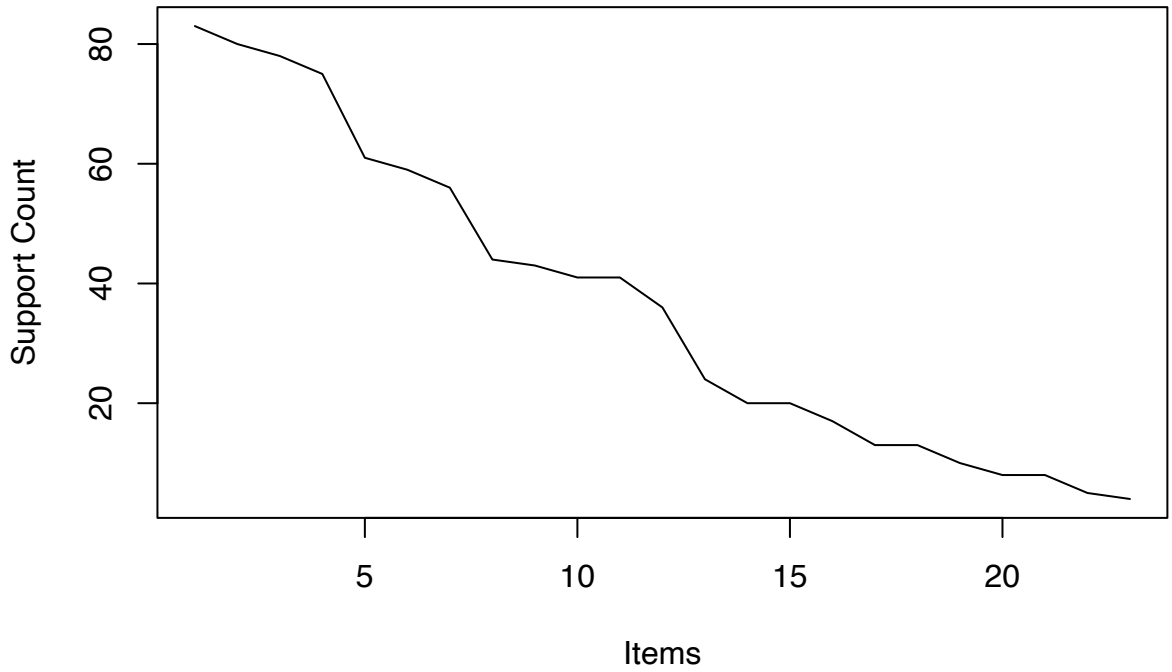
```
#' Plot the binary matrix. Dark dots represent 1s.
image(trans)
```




```
#' Look at the relative frequency (=support) of items in the data set. Here we look at the 10 most freq
itemFrequencyPlot(trans,topN=20)
```



```
plot(sort(itemFrequency(trans, type="absolute"), decreasing=TRUE),
  xlab = "Items", ylab="Support Count", type="l")
```




```
#' __Alternative encoding:__ Also create items for FALSE (use factor)
sapply(Zoo, class)
```

```
##      hair  feathers      eggs      milk  airborne   aquatic  predator   toothed
## "logical" "logical" "logical" "logical" "logical" "logical" "logical" "logical"
##  backbone  breathes  venomous      fins      legs      tail  domestic   catsize
## "logical" "logical" "logical" "logical" "logical" "logical" "logical" "logical"
##      type
##  "factor"
```

```
Zoo2 <- Zoo
for(i in 1:ncol(Zoo2)) Zoo2[[i]] <- as.factor(Zoo2[[i]])
sapply(Zoo2, class)
```

```
##     hair feathers     eggs     milk airborne  aquatic predator  toothed
## "factor" "factor" "factor" "factor" "factor" "factor" "factor" "factor"
## backbone breathes venomous     fins     legs     tail domestic  catsize
## "factor" "factor" "factor" "factor" "factor" "factor" "factor" "factor"
##     type
## "factor"
```

```
summary(Zoo2)
```

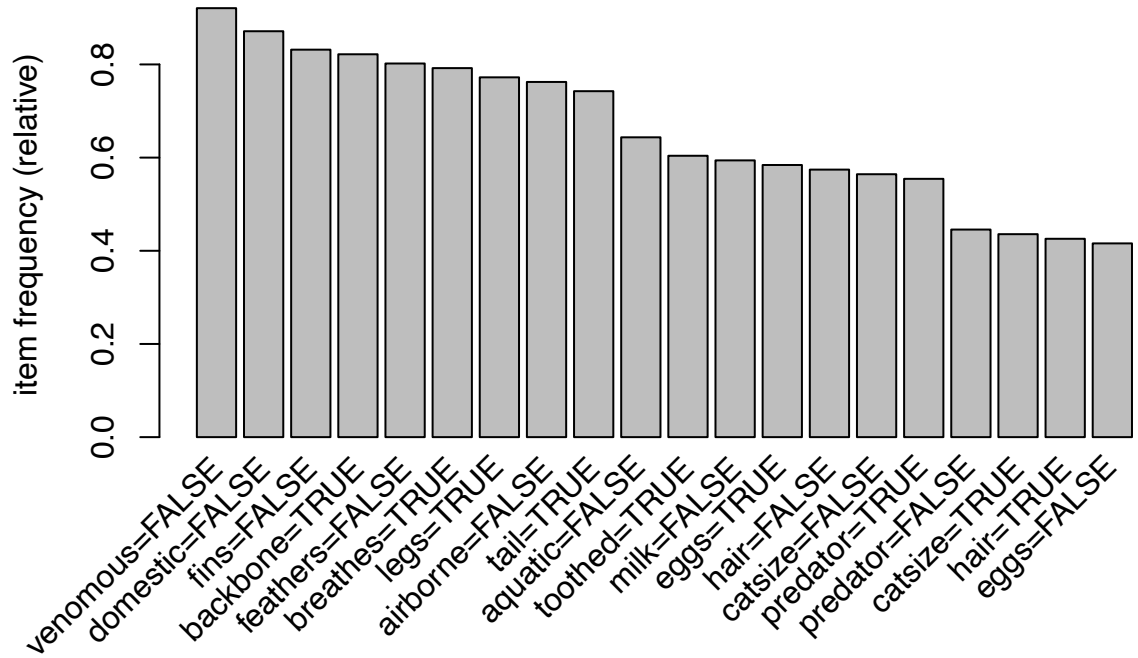
```
##     hair     feathers     eggs       milk     airborne   aquatic    predator
##  FALSE:58   FALSE:81   FALSE:42   FALSE:60   FALSE:77   FALSE:65   FALSE:45
##  TRUE :43   TRUE :20   TRUE :59   TRUE :41   TRUE :24   TRUE :36   TRUE :56
##
##
##
##
##
```

```
##   toothed    backbone   breathes   venomous      fins        legs        tail
##  FALSE:40   FALSE:18   FALSE:21   FALSE:93   FALSE:84   FALSE:23   FALSE:26
##  TRUE :61   TRUE :83   TRUE :80   TRUE : 8   TRUE :17   TRUE :78   TRUE :75
##
##
##
##
##
##   domestic   catsize             type
##  FALSE:88   FALSE:57   mammal       :41
##  TRUE :13   TRUE :44   bird         :20
##                        reptile      : 5
##                        fish         :13
##                        amphibian    : 4
##                        insect       : 8
##                        mollusc.et.al:10
```

```
trans2 <- as(Zoo2, "transactions")
trans2
```

```
## transactions in sparse format with
##  101 transactions (rows) and
##  39 items (columns)
```

```
itemFrequencyPlot(trans2, topN=20)
```




```
# Select transactions that contain a certain item
trans_insects <- trans2[trans %in% "type=insect"]
trans_insects
```

```
## transactions in sparse format with
##  8 transactions (rows) and
##  39 items (columns)
```

```
inspect(trans_insects)
```

```
##     items             transactionID
## [1] {hair=FALSE,
##      feathers=FALSE,
##      eggs=TRUE,
##      milk=FALSE,
##      airborne=FALSE,
##      aquatic=FALSE,
##      predator=FALSE,
##      toothed=FALSE,
##      backbone=FALSE,
##      breathes=TRUE,
##      venomous=FALSE,
##      fins=FALSE,
##      legs=TRUE,
##      tail=FALSE,
##      domestic=FALSE,
##      catsize=FALSE,
##      type=insect}          flea
## [2] {hair=FALSE,
##      feathers=FALSE,
##      eggs=TRUE,
##      milk=FALSE,
##      airborne=TRUE,
##      aquatic=FALSE,
##      predator=FALSE,
##      toothed=FALSE,
##      backbone=FALSE,
##      breathes=TRUE,
##      venomous=FALSE,
##      fins=FALSE,
##      legs=TRUE,
##      tail=FALSE,
##      domestic=FALSE,
##      catsize=FALSE,
##      type=insect}          gnat
## [3] {hair=TRUE,
##      feathers=FALSE,
##      eggs=TRUE,
##      milk=FALSE,
##      airborne=TRUE,
##      aquatic=FALSE,
##      predator=FALSE,
##      toothed=FALSE,
##      backbone=FALSE,
##      breathes=TRUE,
##      venomous=TRUE,
##      fins=FALSE,
##      legs=TRUE,
##      tail=FALSE,
##      domestic=TRUE,
##      catsize=FALSE,
##      type=insect}          honeybee
```

```
## [4] {hair=TRUE,
##      feathers=FALSE,
##      eggs=TRUE,
##      milk=FALSE,
##      airborne=TRUE,
##      aquatic=FALSE,
##      predator=FALSE,
##      toothed=FALSE,
##      backbone=FALSE,
##      breathes=TRUE,
##      venomous=FALSE,
##      fins=FALSE,
##      legs=TRUE,
##      tail=FALSE,
##      domestic=FALSE,
##      catsize=FALSE,
##      type=insect}          housefly
## [5] {hair=FALSE,
##      feathers=FALSE,
##      eggs=TRUE,
##      milk=FALSE,
##      airborne=TRUE,
##      aquatic=FALSE,
##      predator=TRUE,
##      toothed=FALSE,
##      backbone=FALSE,
##      breathes=TRUE,
##      venomous=FALSE,
##      fins=FALSE,
##      legs=TRUE,
##      tail=FALSE,
##      domestic=FALSE,
##      catsize=FALSE,
##      type=insect}          ladybird
## [6] {hair=TRUE,
##      feathers=FALSE,
##      eggs=TRUE,
##      milk=FALSE,
##      airborne=TRUE,
##      aquatic=FALSE,
##      predator=FALSE,
##      toothed=FALSE,
##      backbone=FALSE,
##      breathes=TRUE,
##      venomous=FALSE,
##      fins=FALSE,
##      legs=TRUE,
##      tail=FALSE,
##      domestic=FALSE,
##      catsize=FALSE,
##      type=insect}          moth
## [7] {hair=FALSE,
##      feathers=FALSE,
##      eggs=TRUE,
```

```
##       milk=FALSE,
##       airborne=FALSE,
##       aquatic=FALSE,
##       predator=FALSE,
##       toothed=FALSE,
##       backbone=FALSE,
##       breathes=TRUE,
##       venomous=FALSE,
##       fins=FALSE,
##       legs=TRUE,
##       tail=FALSE,
##       domestic=FALSE,
##       catsize=FALSE,
##       type=insect}          termite
## [8] {hair=TRUE,
##       feathers=FALSE,
##       eggs=TRUE,
##       milk=FALSE,
##       airborne=TRUE,
##       aquatic=FALSE,
##       predator=FALSE,
##       toothed=FALSE,
##       backbone=FALSE,
##       breathes=TRUE,
##       venomous=TRUE,
##       fins=FALSE,
##       legs=TRUE,
##       tail=FALSE,
##       domestic=FALSE,
##       catsize=FALSE,
##       type=insect}          wasp
```

```
#' ## Vertical layout (Transaction ID Lists)
#'
#' The default layout for transactions is horizontal layout (i.e. each transaction is a row).
#' The vertical layout represents transaction data as a list of transaction IDs for each item (= transa
vertical <- as(trans, "tidLists")
as(vertical, "matrix")[1:10,1:5]
```

```
##          aardvark antelope  bass  bear  boar
## hair         TRUE     TRUE FALSE  TRUE  TRUE
## feathers    FALSE    FALSE FALSE FALSE FALSE
## eggs        FALSE    FALSE  TRUE FALSE FALSE
## milk         TRUE     TRUE FALSE  TRUE  TRUE
## airborne    FALSE    FALSE FALSE FALSE FALSE
## aquatic     FALSE    FALSE  TRUE FALSE FALSE
## predator     TRUE    FALSE  TRUE  TRUE  TRUE
## toothed      TRUE     TRUE  TRUE  TRUE  TRUE
## backbone     TRUE     TRUE  TRUE  TRUE  TRUE
## breathes     TRUE     TRUE FALSE  TRUE  TRUE
```

```
#' # Frequent Itemsets
#' ## Mine Frequent Itemsets
#'
#' For this dataset we have already a huge number of possible itemsets
```

```
2^ncol(trans)
```

```
## [1] 8388608
```

```
#' Find frequent itemsets (target="frequent") with the default settings.
is <- apriori(trans, parameter=list(target="frequent"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen            target  ext
##      10 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 10
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[23 item(s), 101 transaction(s)] done [0.00s].
## sorting and recoding items ... [18 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
## Warning in apriori(trans, parameter = list(target = "frequent")): Mining stopped
## (maxlen reached). Only patterns up to a length of 10 returned!
```

```
##  done [0.00s].
## sorting transactions ... done [0.00s].
## writing ... [1465 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
is
```

```
## set of 1465 itemsets
```

```
#' Default minimum support is .1 (10\%).
#' __Note:__ We use here a very small data set. For larger datasets
#' the default minimum support might be to low and you may run out of memory. You probably want to star
#' .5 (50\%) and then work your way down.

5/nrow(trans)
```

```
## [1] 0.04950495
```

```
#' In order to find itemsets that effect 5 animals I need to go down to a
#' support of about 5\%.
is <- apriori(trans, parameter=list(target="frequent", support=0.05))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen            target  ext
```

```
##      10 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 5
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[23 item(s), 101 transaction(s)] done [0.00s].
## sorting and recoding items ... [21 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
## Warning in apriori(trans, parameter = list(target = "frequent", support =
## 0.05)): Mining stopped (maxlen reached). Only patterns up to a length of 10
## returned!
```

```
##  done [0.00s].
## sorting transactions ... done [0.00s].
## writing ... [2537 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

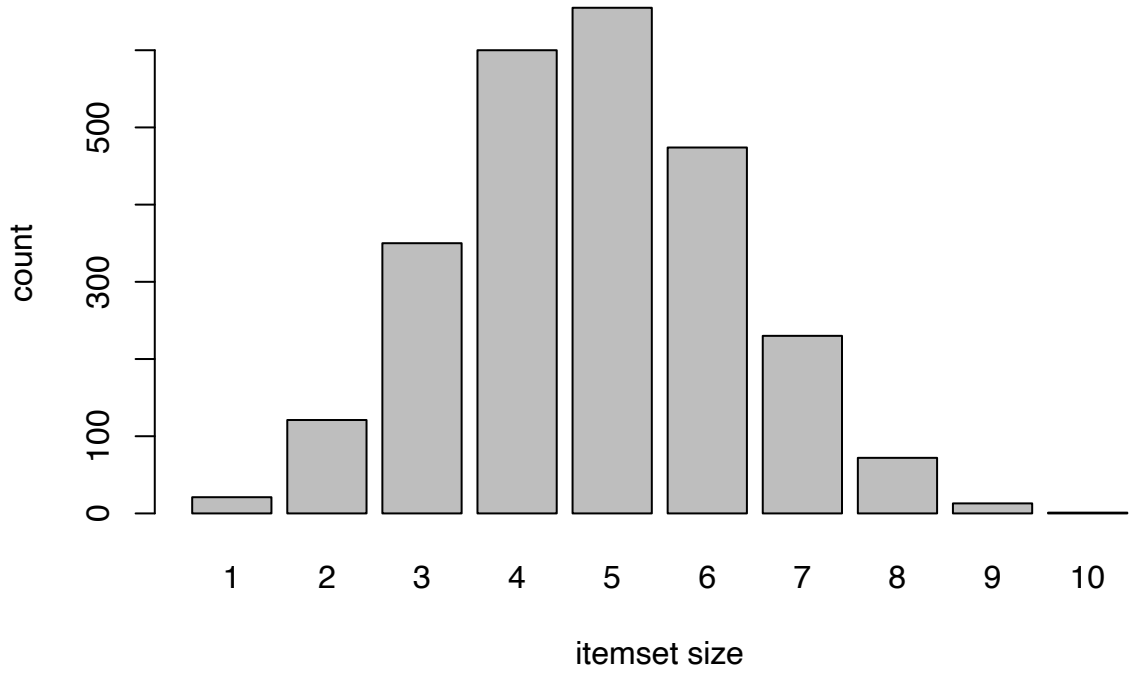
```
is
```

```
## set of 2537 itemsets
```

```
#' Sort by support
is <- sort(is, by="support")
inspect(head(is, n=10))
```

```
##      items                      support   count
## [1]  {backbone}                 0.8217822 83
## [2]  {breathes}                 0.7920792 80
## [3]  {legs}                     0.7722772 78
## [4]  {tail}                     0.7425743 75
## [5]  {backbone, tail}           0.7326733 74
## [6]  {breathes, legs}           0.7227723 73
## [7]  {backbone, breathes}       0.6831683 69
## [8]  {backbone, legs}           0.6336634 64
## [9]  {backbone, breathes, legs} 0.6336634 64
## [10] {toothed}                  0.6039604 61
```

```
#' Look at frequent itemsets with many items (set breaks manually since
#' Automatically chosen breaks look bad)
barplot(table(size(is)), xlab="itemset size", ylab="count")
```




```
inspect(is[size(is)>8])
```

```
##      items            support count
## [1]  {hair,
##       milk,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.23762376    24
## [2]  {hair,
##       milk,
##       predator,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       catsize,
##       type=mammal} 0.15841584    16
## [3]  {hair,
##       milk,
##       predator,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       type=mammal} 0.14851485    15
## [4]  {hair,
##       milk,
```

```
##       predator,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.13861386    14
## [5]  {hair,
##       milk,
##       predator,
##       toothed,
##       breathes,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.12871287    13
## [6]  {hair,
##       milk,
##       predator,
##       toothed,
##       backbone,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.12871287    13
## [7]  {hair,
##       milk,
##       predator,
##       toothed,
##       backbone,
##       breathes,
##       tail,
##       catsize,
##       type=mammal} 0.12871287    13
## [8]  {milk,
##       predator,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.12871287    13
## [9]  {hair,
##       milk,
##       predator,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       catsize}     0.12871287    13
## [10] {hair,
##       predator,
```

```
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.12871287    13
## [11] {hair,
##       milk,
##       predator,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       catsize,
##       type=mammal} 0.12871287    13
## [12] {hair,
##       milk,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       domestic,
##       catsize,
##       type=mammal} 0.05940594     6
## [13] {hair,
##       milk,
##       toothed,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       domestic,
##       type=mammal} 0.05940594     6
## [14] {feathers,
##       eggs,
##       airborne,
##       predator,
##       backbone,
##       breathes,
##       legs,
##       tail,
##       type=bird}   0.05940594     6
```

```
#' ## Concise Representation of Itemsets
#'
#' Find maximal frequent itemsets (no superset if frequent)
is_max <- is[is.maximal(is)]
is_max
```

```
## set of 22 itemsets
```

```
inspect(head(sort(is_max, by="support")))
```

```
##     items             support count
## [1] {hair,
##      milk,
##      predator,
##      toothed,
##      backbone,
##      breathes,
##      legs,
##      tail,
##      catsize,
##      type=mammal} 0.12871287    13
## [2] {eggs,
##      aquatic,
##      predator,
##      toothed,
##      backbone,
##      fins,
##      tail,
##      type=fish}   0.08910891     9
## [3] {aquatic,
##      predator,
##      toothed,
##      backbone,
##      breathes}    0.07920792     8
## [4] {aquatic,
##      predator,
##      toothed,
##      backbone,
##      fins,
##      tail,
##      catsize}     0.06930693     7
## [5] {eggs,
##      venomous}    0.05940594     6
## [6] {predator,
##      venomous}    0.05940594     6
```

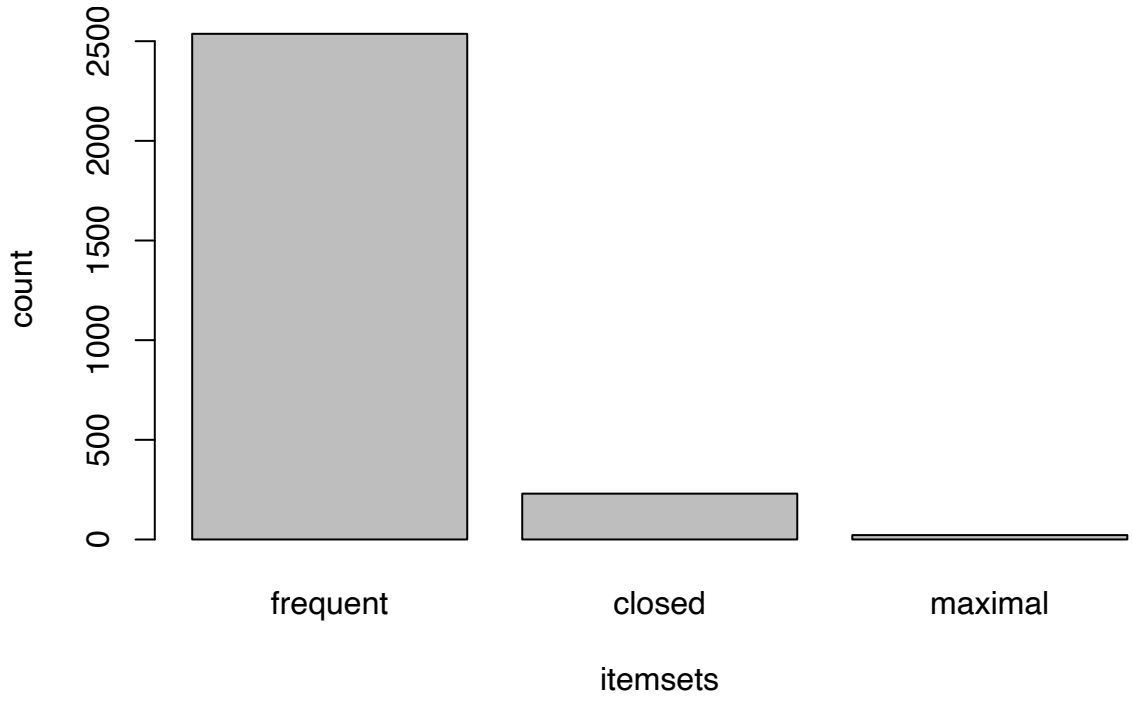
```
#' Find closed frequent itemsets (no superset if frequent)
is_closed <- is[is.closed(is)]
is_closed
```

```
## set of 230 itemsets
```

```
inspect(head(sort(is_closed, by="support")))
```

```
##     items             support   count
## [1] {backbone}        0.8217822 83
## [2] {breathes}        0.7920792 80
## [3] {legs}            0.7722772 78
## [4] {tail}            0.7425743 75
## [5] {backbone, tail}  0.7326733 74
## [6] {breathes, legs}  0.7227723 73
```

```
barplot(c(
  frequent=length(is),
  closed=length(is_closed),
  maximal=length(is_max)
  ), ylab="count", xlab="itemsets")
```




```
#' We use the APRIORI algorithm (see [`? apriori`](https://www.rdocumentation.org/packages/arules/topic

rules <- apriori(trans, parameter=list(support=0.05, confidence=.9))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5    0.05      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: 5
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[23 item(s), 101 transaction(s)] done [0.00s].
## sorting and recoding items ... [21 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
## Warning in apriori(trans, parameter = list(support = 0.05, confidence = 0.9)):
## Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!
```

```
##  done [0.00s].
## writing ... [7174 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
length(rules)
```

```
## [1] 7174
```

```
inspect(head(rules))
```

```
##     lhs                     rhs        support    confidence coverage
## [1] {type=insect}        => {eggs}     0.07920792 1.0        0.07920792
## [2] {type=insect}        => {legs}     0.07920792 1.0        0.07920792
## [3] {type=insect}        => {breathes} 0.07920792 1.0        0.07920792
## [4] {type=mollusc.et.al} => {eggs}     0.08910891 0.9        0.09900990
## [5] {type=fish}          => {fins}     0.12871287 1.0        0.12871287
## [6] {type=fish}          => {aquatic}  0.12871287 1.0        0.12871287
##     lift     count
## [1] 1.711864  8
## [2] 1.294872  8
## [3] 1.262500  8
## [4] 1.540678  9
## [5] 5.941176 13
## [6] 2.805556 13
```

```
quality(head(rules))
```

```
##      support confidence   coverage     lift count
## 1 0.07920792        1.0 0.07920792 1.711864     8
## 2 0.07920792        1.0 0.07920792 1.294872     8
## 3 0.07920792        1.0 0.07920792 1.262500     8
## 4 0.08910891        0.9 0.09900990 1.540678     9
## 5 0.12871287        1.0 0.12871287 5.941176    13
## 6 0.12871287        1.0 0.12871287 2.805556    13
```

```
#' Look at rules with highest lift
rules <- sort(rules, by="lift")
inspect(head(rules, n=10))
```

```
##      lhs                               rhs         support    confidence
## [1]  {eggs, fins}                   => {type=fish} 0.12871287 1
## [2]  {eggs, aquatic, fins}          => {type=fish} 0.12871287 1
## [3]  {eggs, predator, fins}         => {type=fish} 0.08910891 1
## [4]  {eggs, toothed, fins}          => {type=fish} 0.12871287 1
## [5]  {eggs, fins, tail}             => {type=fish} 0.12871287 1
## [6]  {eggs, backbone, fins}         => {type=fish} 0.12871287 1
## [7]  {eggs, aquatic, predator, fins} => {type=fish} 0.08910891 1
## [8]  {eggs, aquatic, toothed, fins}  => {type=fish} 0.12871287 1
## [9]  {eggs, aquatic, fins, tail}    => {type=fish} 0.12871287 1
## [10] {eggs, aquatic, backbone, fins} => {type=fish} 0.12871287 1
##      coverage   lift     count
## [1]  0.12871287 7.769231 13
## [2]  0.12871287 7.769231 13
## [3]  0.08910891 7.769231  9
## [4]  0.12871287 7.769231 13
## [5]  0.12871287 7.769231 13
## [6]  0.12871287 7.769231 13
```

```
## [7]  0.08910891 7.769231  9
## [8]  0.12871287 7.769231 13
## [9]  0.12871287 7.769231 13
## [10] 0.12871287 7.769231 13
```

```
#' Create rules using the alternative encoding (with "FALSE" item)
r <- apriori(trans2)
```

```
## 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: 10
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[39 item(s), 101 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
## Warning in apriori(trans2): Mining stopped (maxlen reached). Only patterns up to
## a length of 10 returned!
```

```
##  done [0.06s].
## writing ... [1517191 rule(s)] done [0.13s].
## creating S4 object  ... done [0.50s].
```

```
r
```

```
## set of 1517191 rules
```

```
print(object.size(r), unit="Mb")
```

```
## 110.2 Mb
```

```
inspect(r[1:10])
```

```
##      lhs                 rhs                 support   confidence coverage
## [1]  {}                => {feathers=FALSE} 0.8019802 0.8019802  1.0000000
## [2]  {}                => {backbone=TRUE}  0.8217822 0.8217822  1.0000000
## [3]  {}                => {fins=FALSE}     0.8316832 0.8316832  1.0000000
## [4]  {}                => {domestic=FALSE} 0.8712871 0.8712871  1.0000000
## [5]  {}                => {venomous=FALSE} 0.9207921 0.9207921  1.0000000
## [6]  {domestic=TRUE} => {predator=FALSE} 0.1089109 0.8461538  0.1287129
## [7]  {domestic=TRUE} => {aquatic=FALSE}  0.1188119 0.9230769  0.1287129
## [8]  {domestic=TRUE} => {legs=TRUE}      0.1188119 0.9230769  0.1287129
## [9]  {domestic=TRUE} => {breathes=TRUE}  0.1188119 0.9230769  0.1287129
## [10] {domestic=TRUE} => {backbone=TRUE}  0.1188119 0.9230769  0.1287129
##      lift     count
## [1]  1.000000 81
```

```
## [2]  1.000000 83
## [3]  1.000000 84
## [4]  1.000000 88
## [5]  1.000000 93
## [6]  1.899145 11
## [7]  1.434320 12
## [8]  1.195266 12
## [9]  1.165385 12
## [10] 1.123262 12
```

```
inspect(head(r, by="lift", n = 10))
```

```
##      lhs                  rhs           support confidence  coverage     lift count
## [1]  {breathes=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [2]  {eggs=TRUE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [3]  {milk=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [4]  {breathes=FALSE,
##       fins=TRUE,
##       legs=FALSE}      => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [5]  {aquatic=TRUE,
##       breathes=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [6]  {hair=FALSE,
##       breathes=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [7]  {eggs=TRUE,
##       breathes=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [8]  {milk=FALSE,
##       breathes=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [9]  {toothed=TRUE,
##       breathes=FALSE,
##       fins=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
## [10] {breathes=FALSE,
##       fins=TRUE,
##       tail=TRUE}       => {type=fish} 0.1287129          1 0.1287129 7.769231    13
```

```
#' ## Additional Interest Measures
interestMeasure(rules[1:10], measure=c("phi", "gini"),
  trans=trans)
```

```
##          phi      gini
## 1  1.0000000 0.2242917
## 2  1.0000000 0.2242917
## 3  0.8137612 0.1485276
## 4  1.0000000 0.2242917
## 5  1.0000000 0.2242917
## 6  1.0000000 0.2242917
## 7  0.8137612 0.1485276
## 8  1.0000000 0.2242917
## 9  1.0000000 0.2242917
## 10 1.0000000 0.2242917
```

```
#' Add measures to the rules
quality(rules) <- cbind(quality(rules),
  interestMeasure(rules, measure=c("phi", "gini"),
    trans=trans))

#' Find rules which score high for Phi correlation
inspect(head(rules, by="phi"))
```

```
##     lhs                              rhs         support   confidence
## [1] {eggs, fins}                  => {type=fish} 0.1287129 1
## [2] {eggs, aquatic, fins}         => {type=fish} 0.1287129 1
## [3] {eggs, toothed, fins}         => {type=fish} 0.1287129 1
## [4] {eggs, fins, tail}            => {type=fish} 0.1287129 1
## [5] {eggs, backbone, fins}        => {type=fish} 0.1287129 1
## [6] {eggs, aquatic, toothed, fins} => {type=fish} 0.1287129 1
##     coverage  lift     count phi gini
## [1] 0.1287129 7.769231 13    1   0.2242917
## [2] 0.1287129 7.769231 13    1   0.2242917
## [3] 0.1287129 7.769231 13    1   0.2242917
## [4] 0.1287129 7.769231 13    1   0.2242917
## [5] 0.1287129 7.769231 13    1   0.2242917
## [6] 0.1287129 7.769231 13    1   0.2242917
```

```
#' ## Mine using Templates
#'
#' Sometimes it is beneficial to specify what items should be where in the rule. For apriori we can use
#' the following we restrict rules to an animal `type` in the RHS and any item in
#' the LHS.
type <- grep("type=", itemLabels(trans), value = TRUE)
type
```

```
## [1] "type=mammal"        "type=bird"          "type=reptile"
## [4] "type=fish"          "type=amphibian"     "type=insect"
## [7] "type=mollusc.et.al"
```

```
rules_type <- apriori(trans,
  appearance= list(rhs=type, default="lhs"))
```

```
## 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: 10
##
## set item appearances ...[7 item(s)] done [0.00s].
## set transactions ...[23 item(s), 101 transaction(s)] done [0.00s].
## sorting and recoding items ... [18 item(s)] done [0.00s].
```

```
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
## Warning in apriori(trans, appearance = list(rhs = type, default = "lhs")):
## Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!
```

```
##  done [0.00s].
## writing ... [571 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

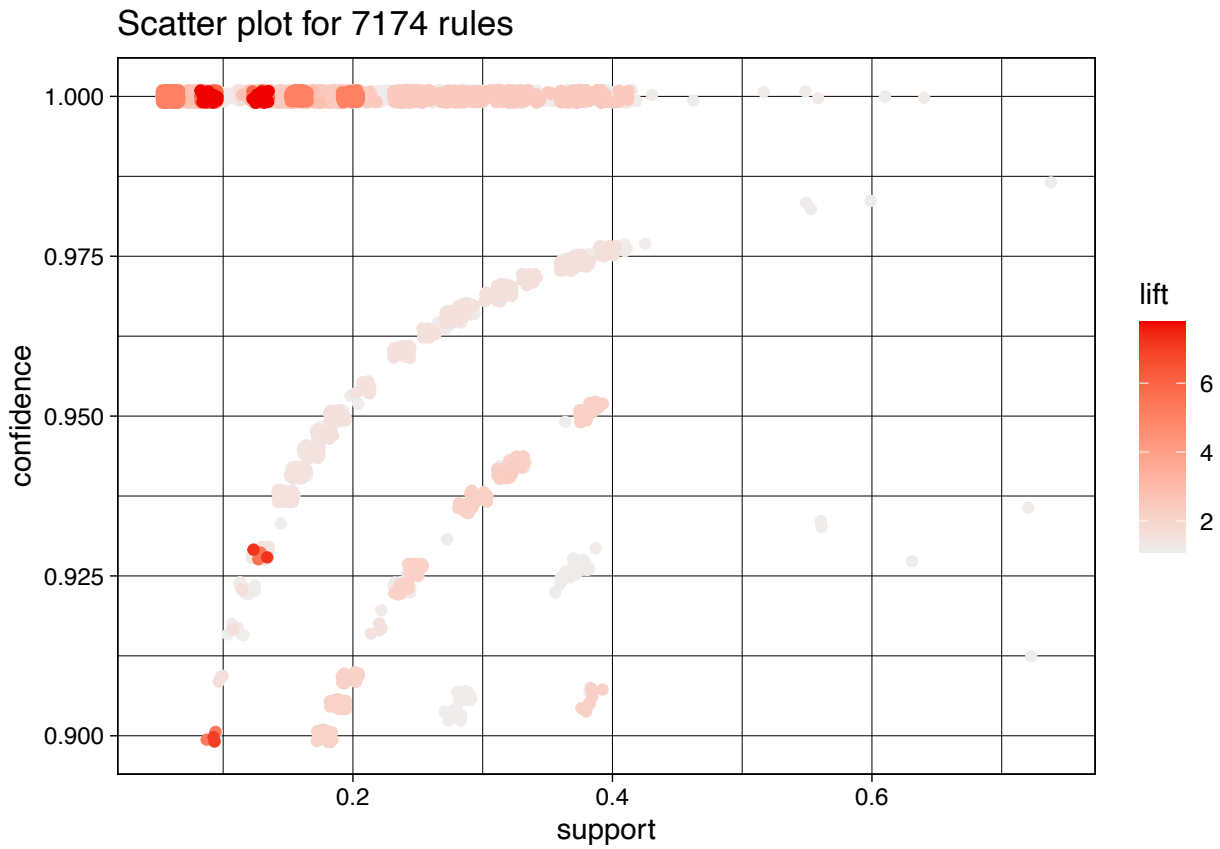
```
inspect(head(sort(rules_type, by="lift")))
```

```
##     lhs                               rhs         support   confidence
## [1] {eggs, fins}                   => {type=fish} 0.1287129 1
## [2] {eggs, aquatic, fins}          => {type=fish} 0.1287129 1
## [3] {eggs, toothed, fins}          => {type=fish} 0.1287129 1
## [4] {eggs, fins, tail}             => {type=fish} 0.1287129 1
## [5] {eggs, backbone, fins}         => {type=fish} 0.1287129 1
## [6] {eggs, aquatic, toothed, fins} => {type=fish} 0.1287129 1
##     coverage  lift     count
## [1] 0.1287129 7.769231 13
## [2] 0.1287129 7.769231 13
## [3] 0.1287129 7.769231 13
## [4] 0.1287129 7.769231 13
## [5] 0.1287129 7.769231 13
## [6] 0.1287129 7.769231 13
```

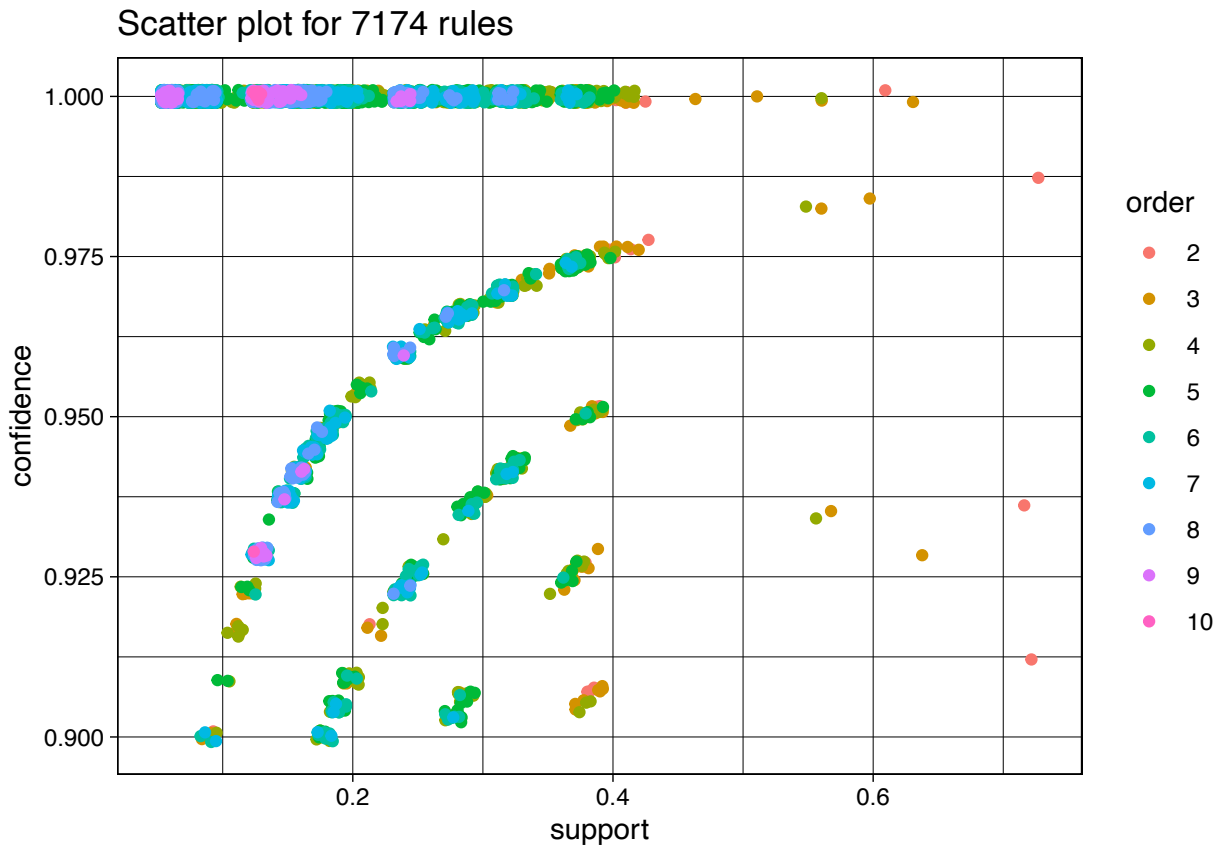
```
library(arulesViz)

#' Default scatterplot
plot(rules)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```

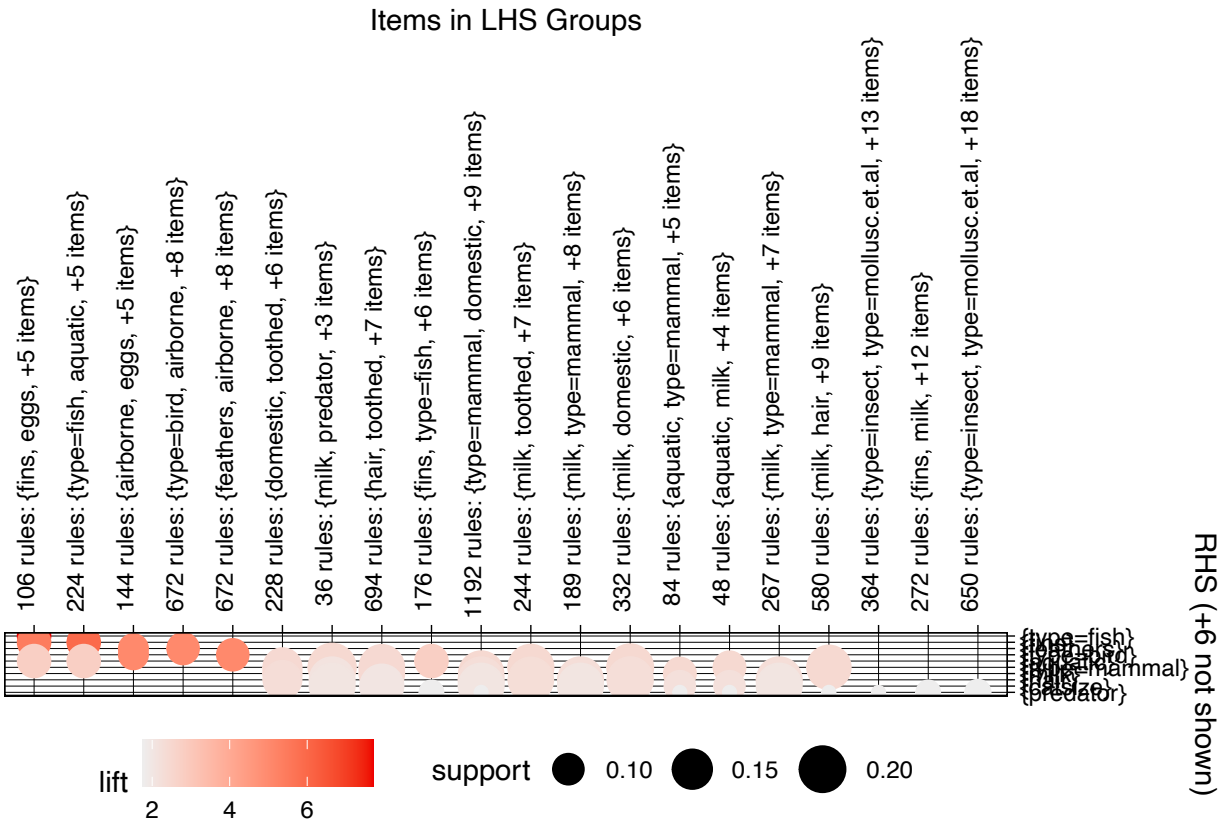

```
#' Add some jitter (randomly move points) to show how many rules have the
#' same confidence and support value.
plot(rules, control=list(jitter=.5))
```




```
plot(rules, shading="order", control=list(jitter=.5))
```

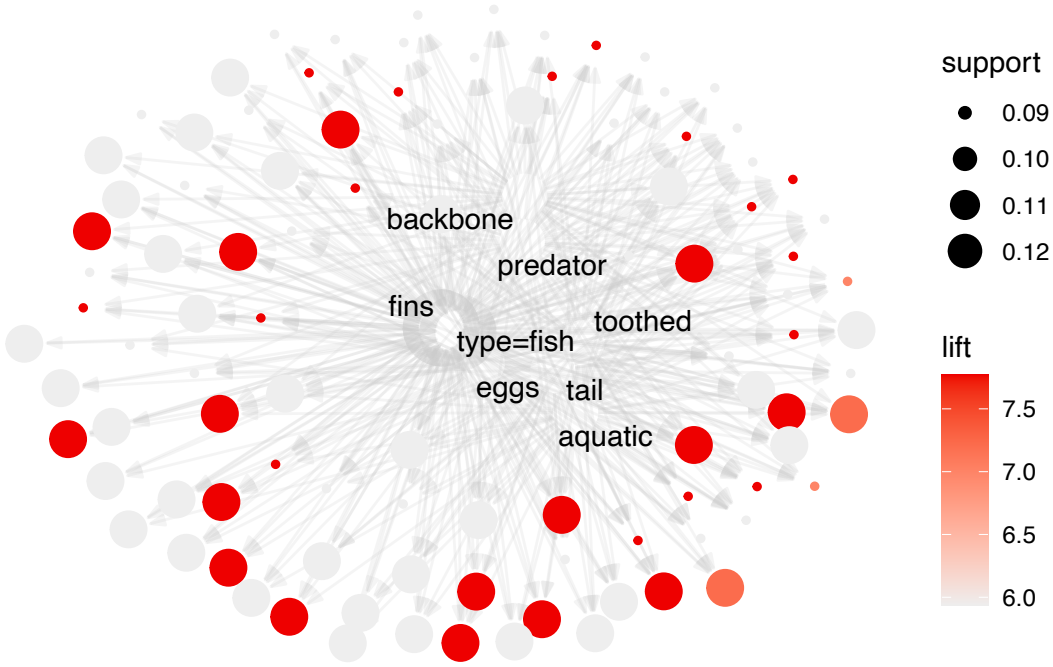



```
#plot(rules, interactive=TRUE)

#' Grouped plot
plot(rules, method="grouped")
```




```
#plot(rules, method="grouped", engine = "interactive")

#' As a graph
plot(rules, method="graph")
```

```
## Warning: Too many rules supplied. Only plotting the best 100 using
## 'lift' (change control parameter max if needed).
```




```
plot(head(rules, by="phi", n = 100), method="graph")
```