实验四：关联规则及其应用

```
#install and load package arules
# install.packages("arules")
library(arules)
```

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

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

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

```
#install and load arulesViz
# install.packages("arulesViz")
library(arulesViz)
#install and load tidyverse
# install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------
- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ---------------------------------------- tidy
verse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x dplyr::recode() masks arules::recode()
## x tidyr::unpack() masks Matrix::unpack()
```

```
#install and load readxml
# install.packages("readxml")
library(readxl)
#install and load knitr
# install.packages("knitr")
library(knitr)
#load ggplot2 as it comes in tidyverse
library(ggplot2)
#install and load lubridate
# install.packages("lubridate")
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:arules':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#install and load plyr
# install.packages("plyr")
library(plyr)
```

```
## ------------------------------------------------------------
-----------------
```

```
## You have loaded plyr after dplyr - this is likely to cause p
roblems.
## If you need functions from both plyr and dplyr, please load
plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------
-----------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summ
arise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(dplyr)

#install.packages("MASS", repos = "https://mirrors.ustc.edu.cn/
CRAN/")
#install.packages("reshape2", repos = "https://mirrors.ustc.ed
u.cn/CRAN/")
#install.packages("reshape", repos = "https://mirrors.ustc.edu.
cn/CRAN/")

library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

```
## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast
```

```
## The following objects are masked from 'package:plyr':
##
##     rename, round_any
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
## The following object is masked from 'package:Matrix':
##
##     expand
```

```
#install.packages('e1071')
#install.packages('purrr')
#install.packages('dplyr')
#library(e1071)
#library(purrr)         # Functional programming
#ibrary(dplyr)         # Functional programming
"done"
```

```
## [1] "done"
```

1、购物篮分析的数据表示。购物篮的数据表示方法有两种：使用事物数据格式或者表数据格式。其中，采用表数据格式时，每条记录表示

不同的事务，每个项采用0/1 标志表示。请将表1表示成表数据格式表2。

表2：路边蔬菜摊的表数据格式 事务 芦笋 豆类 花椰菜 玉米 青辣椒 南瓜 西红柿 1
2
3
4
5
6
7
8
9
10
11
12
13
14

```
df <- read.csv(file = '../data/exp4.csv')
df = df %>% mutate(Business = as.factor(Business))
df
```

```
##    Business Shopping.Basket             X           X.1
X.2      X.3
## 1         1        broccoli green peppers          corn
## 2         2       asparagus        squash          corn
## 3         3            corn      tomatoes         beans   sq
uash
## 4         4   green peppers          corn      tomatoes    b
eans
## 5         5           beans     asparagus      broccoli
## 6         6          squash     asparagus         beans toma
toes
## 7         7        tomatoes          corn
## 8         8        broccoli      tomatoes green peppers
## 9         9          squash     asparagus         beans
## 10       10           beans          corn
## 11       11   green peppers      broccoli         beans   sq
uash
## 12       12       asparagus         beans        squash
## 13       13          squash          corn     asparagus    b
eans
## 14       14            corn green peppers      tomatoes    b
eans broccoli
```

```
df2 = cast( melt(df, id=c("Business")), Business~value)[-1:-2]
```

```
## Aggregation requires fun.aggregate: length used as default
```

```
df2
```

```
##    asparagus beans broccoli corn green peppers squash tomato
es
## 1          0     0        1    1             1      0
0
## 2          1     0        0    1             0      1
0
## 3          0     1        0    1             0      1
1
## 4          0     1        0    1             1      0
1
## 5          1     1        1    0             0      0
0
## 6          1     1        0    0             0      1
1
## 7          0     0        0    1             0      0
1
## 8          0     0        1    0             1      0
1
## 9          1     1        0    0             0      1
0
## 10         0     1        0    1             0      0
0
## 11         0     1        1    0             1      1
0
## 12         1     1        0    0             0      1
0
## 13         1     1        0    1             0      1
0
## 14         0     1        1    1             1      0
1
```

2、建立频繁项集。考虑表2的事务集合D. 这里设在D中出现次数超过4次的项是频繁项集，请给出频繁1-项集，频繁2-项集，频繁3-项集（要求列表表示所有频率计算过程和结果）.

```
write.csv(df[2:5], '../data/exp4-tr.csv', quote = FALSE, row.names = FALSE, col.names = FALSE, sep=",")
```

```
## Warning in write.csv(df[2:5], "../data/exp4-tr.csv", quote = FALSE, row.names =
## FALSE, : attempt to set 'col.names' ignored
```

```
## Warning in write.csv(df[2:5], "../data/exp4-tr.csv", quote = FALSE, row.names =
## FALSE, : attempt to set 'sep' ignored
```

```
tr <- read.transactions('../data/exp4-tr.csv', format = 'basket', sep=',', header = TRUE)
summary(tr)
```

```
## transactions as itemMatrix in sparse format with
##  14 rows (elements/itemsets/transactions) and
##  7 columns (items) and a density of 0.4693878
##
## most frequent items:
##     beans      corn    squash asparagus  tomatoes   (Other)
##        10         8         7         6         6         9
##
## element (itemset/transaction) length distribution:
## sizes
## 2 3 4
## 2 6 6
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   3.000   3.286   4.000   4.000
##
## includes extended item information - examples:
##      labels
## 1 asparagus
## 2     beans
## 3  broccoli
```

```
minsupport = 4
items = names(df2)

c1 =data.frame(combn(items, 1))
c2 =data.frame(combn(items, 2))
c3 =data.frame(combn(items, 3))

# c3 = combn(items, 3)
dfx1 = sapply(c1, function(x) pmap(df2[x],function(...)sum(...
)==1))
dfx2 = sapply(c2, function(x) pmap(df2[x],function(...)sum(...
)==2))
dfx3 = sapply(c2, function(x) pmap(df2[x],function(...)sum(...
)==3))

items = unique(c(df2))
items
```

```
## [[1]]
##  [1] 0 1 0 0 1 1 0 0 1 0 0 1 1 0
##
## [[2]]
##  [1] 0 0 1 1 1 1 0 0 1 1 1 1 1 1
##
## [[3]]
##  [1] 1 0 0 0 1 0 0 1 0 0 1 0 0 1
##
## [[4]]
##  [1] 1 1 1 1 0 0 1 0 0 1 0 0 1 1
##
## [[5]]
##  [1] 1 0 0 1 0 0 0 1 0 0 1 0 0 1
##
## [[6]]
##  [1] 0 1 1 0 0 1 0 0 1 0 1 1 1 0
##
## [[7]]
##  [1] 0 0 1 1 0 1 1 1 0 0 0 0 0 1
```

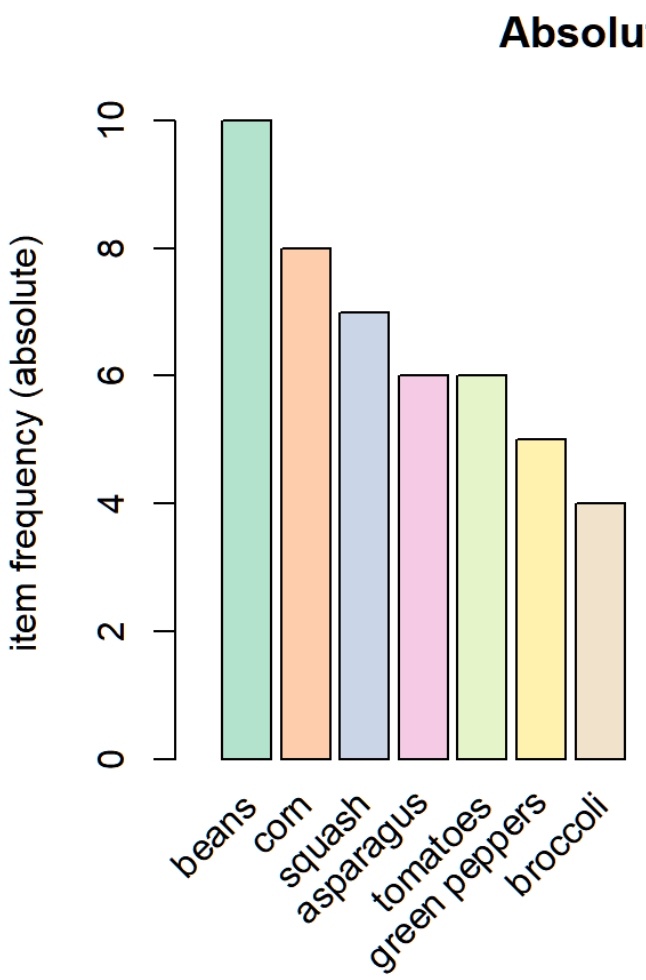
```
inspect(tr)
```

```
##      items
## [1]  {broccoli,corn,green peppers}
## [2]  {asparagus,corn,squash}
## [3]  {beans,corn,squash,tomatoes}
## [4]  {beans,corn,green peppers,tomatoes}
## [5]  {asparagus,beans,broccoli}
## [6]  {asparagus,beans,squash,tomatoes}
## [7]  {corn,tomatoes}
## [8]  {broccoli,green peppers,tomatoes}
## [9]  {asparagus,beans,squash}
## [10] {beans,corn}
## [11] {beans,broccoli,green peppers,squash}
## [12] {asparagus,beans,squash}
## [13] {asparagus,beans,corn,squash}
## [14] {beans,corn,green peppers,tomatoes}
```

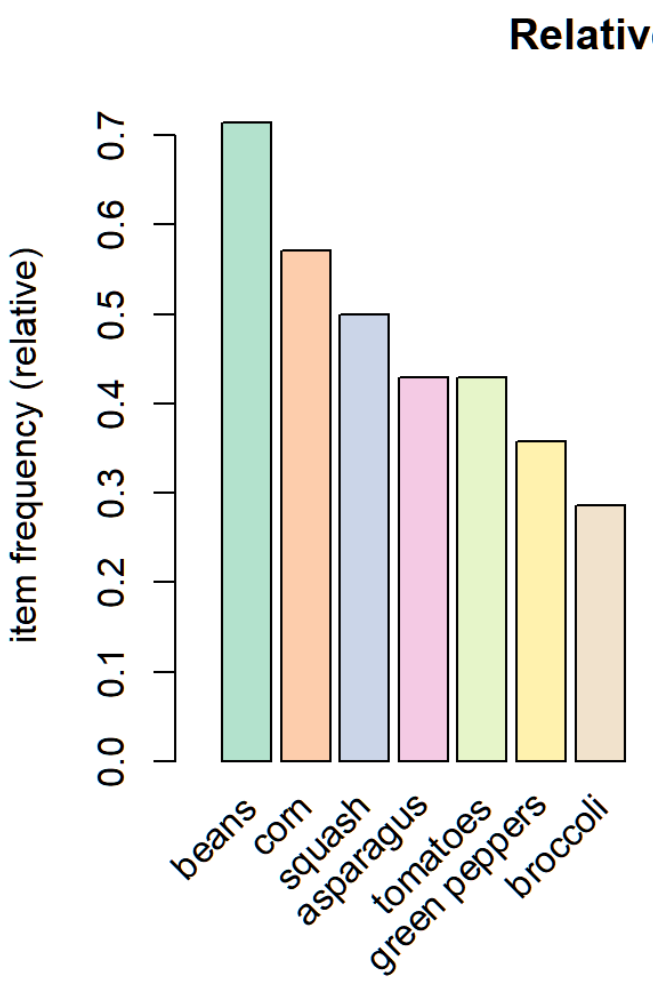
```
# Create an item frequency plot for the top 20 items
if (!require("RColorBrewer")) {
  # install color package of R
  # install.packages("RColorBrewer")
  #include library RColorBrewer
  library(RColorBrewer)
}
```

```
## Loading required package: RColorBrewer
```

```
itemFrequencyPlot(tr,topN=20,type="absolute",col=brewer.pal(8,
'Pastel2'), main="Absolute Item Frequency Plot")
```




```
itemFrequencyPlot(tr,topN=20,type="relative",col=brewer.pal(8,
'Pastel2'), main="Relative Item Frequency Plot")
```



3、建立关联规则。利用频繁项集建立路边蔬菜摊中隐含的关联规则（支持度>50%，可信度>80%），并列表表示所有候选关联规则支持度和可信度的值。给出该案例挖掘的最终关联规则结果。

```
association.rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8,maxlen=10))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime s
upport minlen
##         0.8    0.1    1 none FALSE            TRUE       5
0.001      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: 0
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[7 item(s), 14 transaction(s)] done [0.0
0s].
## sorting and recoding items ... [7 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].
```

```
summary(association.rules)
```

```
## set of 27 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4
##  3 13 11
##
##    Min. 1st Qu.   Median    Mean 3rd Qu.    Max.
##   2.000   3.000   3.000   3.296   4.000   4.000
##
## summary of quality measures:
##     support          confidence        coverage             l
ift
##  Min.   :0.07143   Min.   :0.8000   Min.   :0.07143   Min.
:1.120
##  1st Qu.:0.07143   1st Qu.:1.0000   1st Qu.:0.07143   1st Q
u.:1.400
##  Median :0.07143   Median :1.0000   Median :0.07143   Median
:1.667
##  Mean   :0.13492   Mean   :0.9675   Mean   :0.14815   Mean
:1.912
##  3rd Qu.:0.14286   3rd Qu.:1.0000   3rd Qu.:0.14286   3rd Q
u.:2.167
##  Max.   :0.42857   Max.   :1.0000   Max.   :0.50000   Max.
:3.500
##      count
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.889
##  3rd Qu.:2.000
##  Max.   :6.000
```

```
##
## mining info:
##  data ntransactions support confidence
##    tr            14   0.001        0.8
```

```
inspect(association.rules[1:10])
```

```
##      lhs                          rhs              support    co
nfidence coverage
## [1]  {asparagus}                  => {squash}       0.35714286 0.
8333333  0.42857143
## [2]  {asparagus}                  => {beans}        0.35714286 0.
8333333  0.42857143
## [3]  {squash}                     => {beans}        0.42857143 0.
8571429  0.50000000
## [4]  {broccoli,tomatoes}          => {green peppers} 0.07142857 1.
0000000  0.07142857
## [5]  {broccoli,squash}            => {green peppers} 0.07142857 1.
0000000  0.07142857
## [6]  {green peppers,squash}       => {broccoli}     0.07142857 1.
0000000  0.07142857
## [7]  {broccoli,corn}              => {green peppers} 0.07142857 1.
0000000  0.07142857
## [8]  {asparagus,broccoli}         => {beans}        0.07142857 1.
0000000  0.07142857
## [9]  {broccoli,squash}            => {beans}        0.07142857 1.
0000000  0.07142857
## [10] {green peppers,squash}       => {beans}        0.07142857 1.
0000000  0.07142857
##      lift     count
## [1]  1.666667 5
## [2]  1.166667 5
## [3]  1.200000 6
## [4]  2.800000 1
## [5]  2.800000 1
## [6]  3.500000 1
## [7]  2.800000 1
## [8]  1.400000 1
```

```
## [9]  1.400000 1
## [10] 1.400000 1
```