Association Rules Mining Assingment

Knowledge Discovery and Data Analytics I

Win Phyo

April 3, 2021

```
rm(list=ls())  # Clear environment
```

Import necessary packages

```
library('arules')
library('backports')
library('zeallot')
library('arulesViz')
library('dplyr')
library('stringr')
library('chron')
```

```
df <- read.csv('OnlineRetail.csv', stringsAsFactors = FALSE)
```

Data exploration

```
str(df)
```

```
## 'data.frame':    541909 obs. of  8 variables:
##  $ InvoiceNo  : chr  "536365" "536365" "536365" "536365" ...
##  $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
##  $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREA
M CUPID HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
##  $ Quantity   : int  6 6 8 6 6 2 6 6 6 32 ...
##  $ InvoiceDate: chr  "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:
26" ...
##  $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : int  17850 17850 17850 17850 17850 17850 17850 17850 17850 13047 ...
##  $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingd
om" ...
```

```
summary(df)
```

```
##   InvoiceNo          StockCode          Description          Quantity
##  Length:541909      Length:541909      Length:541909      Min.   :-80995.00
##  Class :character   Class :character   Class :character   1st Qu.:     1.00
##  Mode  :character   Mode  :character   Mode  :character   Median :     3.00
##                                                           Mean   :     9.55
##                                                           3rd Qu.:    10.00
##                                                           Max.   : 80995.00
##
##  InvoiceDate          UnitPrice            CustomerID        Country
##  Length:541909      Min.   :-11062.06   Min.   :12346    Length:541909
##  Class :character   1st Qu.:     1.25   1st Qu.:13953    Class :character
##  Mode  :character   Median :     2.08   Median :15152    Mode  :character
##                     Mean   :     4.61   Mean   :15288
##                     3rd Qu.:     4.13   3rd Qu.:16791
##                     Max.   : 38970.00   Max.   :18287
##                                         NA's   :135080
```

Data cleaning

```
sum(is.null(df$InvoiceNo))  # Get sum of all records with null InvoiceNo
```

```
## [1] 0
```

No null InvoiceNo values observed in dataframe

It is observed that there are invoice numbers that begin with the character ‘C’. Drop ‘C’ character.

```
xrows <- 0
for(i in 1:nrow(df)){
  if(grepl('C', df[i, 'InvoiceNo'])){
    xrows[i] <- i
  }
}

xrows <- xrows[!is.na(xrows)]  # Drop rows with NA; keep only valid row numbers

df <- df[-xrows, ]  # Remove these rows from dataframe
```

Replace spaces in item description field with underscores

```
df$Description <- trimws(df$Description)  # Remove trailing and leading spaces
df$Description <- gsub(" ", "_", df$Description)
```

Convert date field to native R datetime object

```
df$InvoiceDate <- as.Date(df$InvoiceDate, format = "%m/%d/%Y")
```

Filter dataframe for Irish transactions

```
eire <- df[df$Country == 'EIRE', ]
```

Write cleaned Irish dataset to file for ease of use and as a checkpoint

```
write.csv(eire, file = '2021-cleaned-eire.csv', row.names = FALSE)
```

Assocation Rule Mining

Import Irish dataset as transactions

```
eire <- read.transactions(
  '2021-cleaned-eire.csv',
  format = c('single'),
  header = TRUE,
  rm.duplicates = FALSE,
  cols = c('InvoiceNo', 'StockCode'),
  sep = ','
)
```

Check what this object looks like

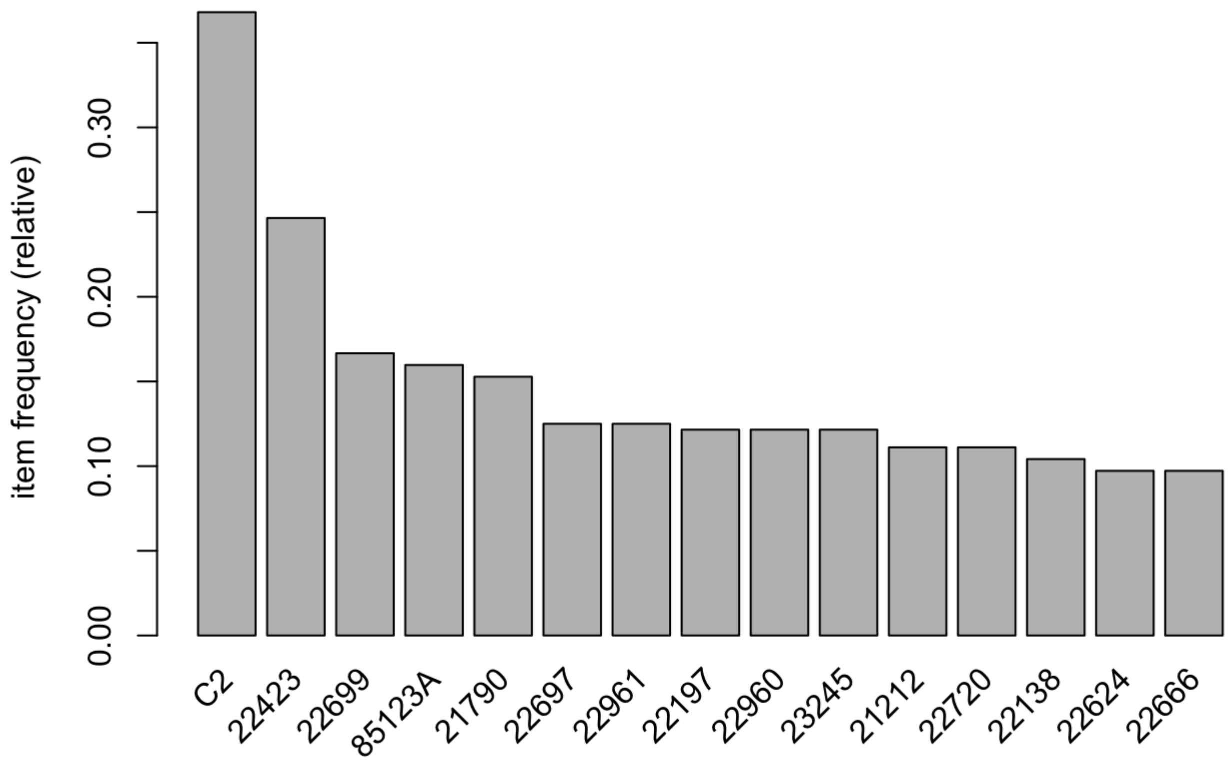
```
inspect(eire[1:2])
```

```
##     items    transactionID
## [1] {21055,
##      21056,
##      21576,
##      21579,
##      21833,
##      21889,
##      21891,
##      22147,
##      22150,
##      22355,
##      22492,
##      22493,
##      22622,
##      22968,
##      85071A,
##      85071C,
##      85135B,
##      85136A,
##      85136C,
##      C2}            536540
## [2] {21915}         536541
```

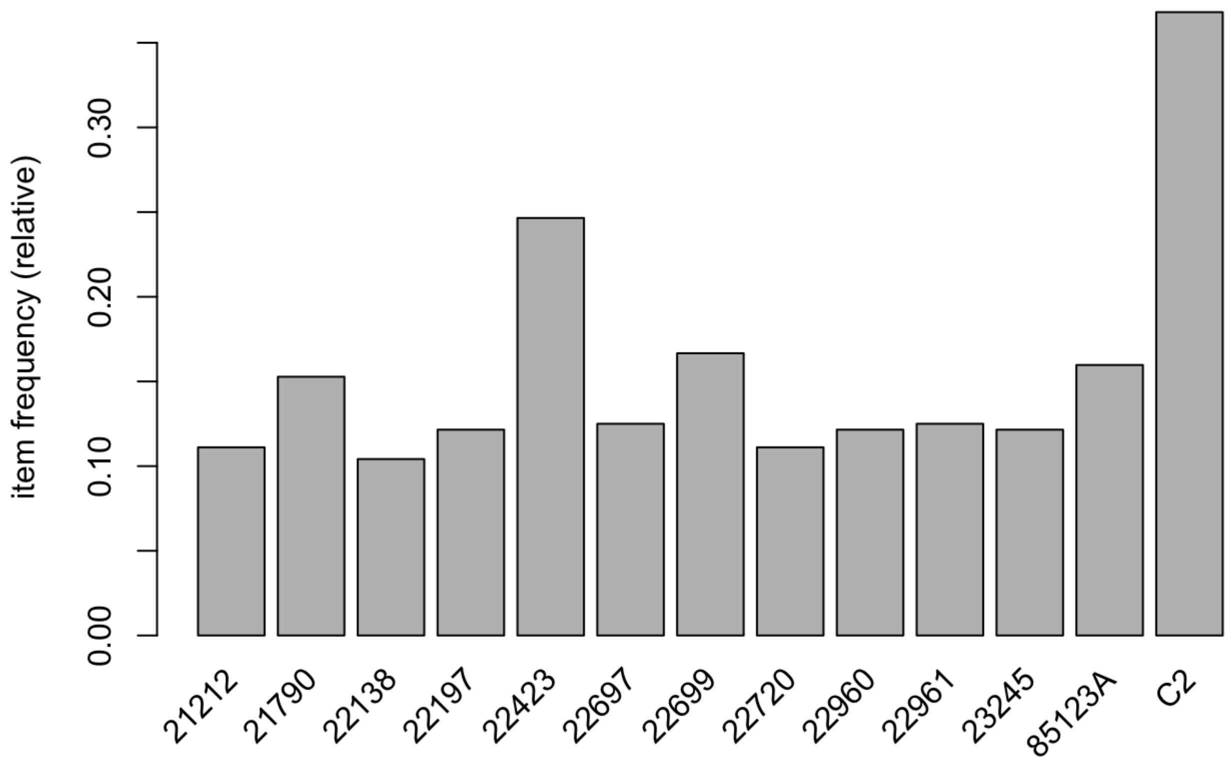
Items are grouped by invoice number; each record in this dataset corresponds to a particular, unique invoice number.

Explore support counts to determine minimum value/cutoff point

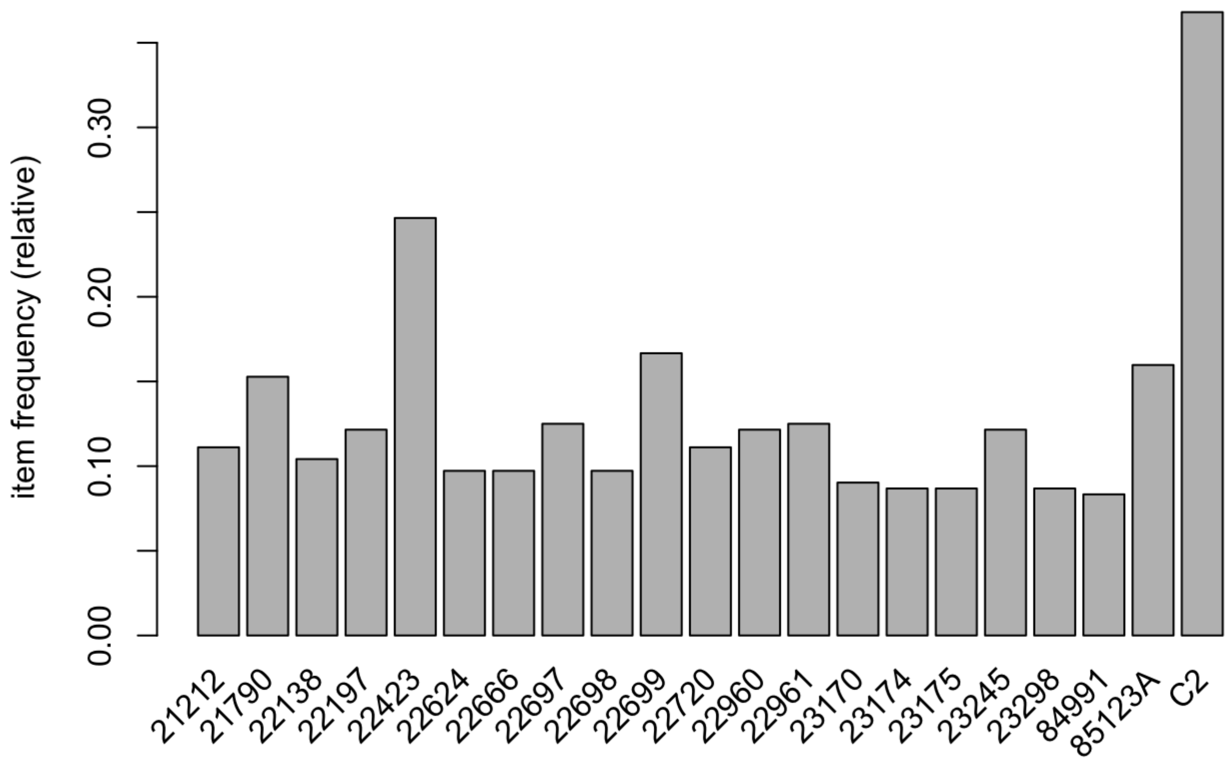
```
itemFrequencyPlot(eire, topN = 15)  # 15 most frequently occuring items
```




```
itemFrequencyPlot(eire, support = 0.1)
```




```
itemFrequencyPlot(eire, support = 0.08)
```



There are 13 items that appear in at least 10% of all transactions. There are 21 items that appear in at least 8% of all transactions.

Get association rules

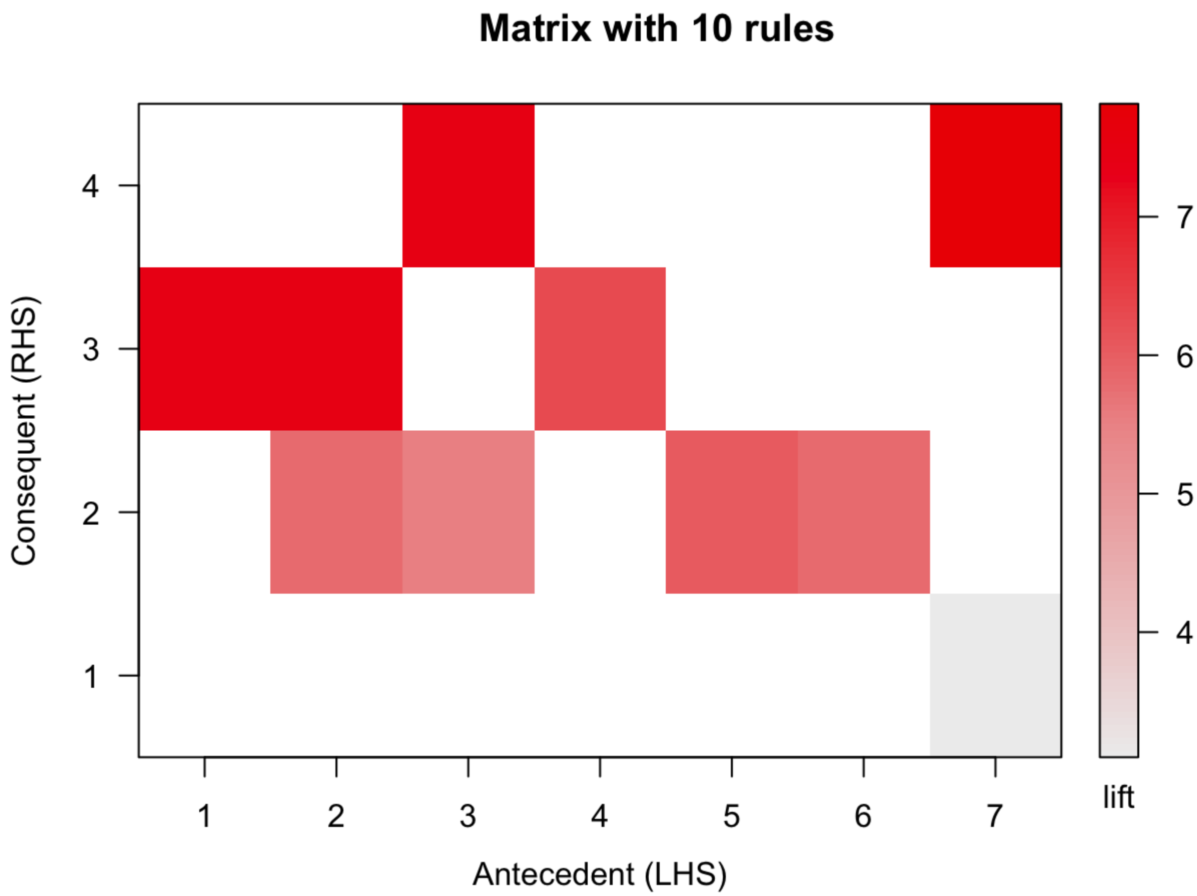
Let the minimum confidence value be 0.7, and the minimum support value be 0.08

```
eire.rules <- apriori(
  eire,
  parameter = list(
    confidence = 0.7,
    support = 0.08,
    minlen = 2
  )
)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.08      2
##  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: 23
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[1968 item(s), 288 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 done [0.00s].
## writing ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
plot(
  eire.rules,
  method = 'matrix',
)
```

```
## Itemsets in Antecedent (LHS)
## [1] "{22698,22699}" "{22698}"       "{22697}"       "{22423,22699}"
## [5] "{22423,22697}" "{22697,22698}" "{22697,22699}"
## Itemsets in Consequent (RHS)
## [1] "{22423}" "{22699}" "{22697}" "{22698}"
```



Explore rules

```
inspect(eire.rules)
```

```
##      lhs              rhs     support    confidence coverage   lift     count
## [1]  {22698}       => {22697} 0.09027778 0.9285714  0.09722222 7.428571 26
## [2]  {22697}       => {22698} 0.09027778 0.7222222  0.12500000 7.428571 26
## [3]  {22698}       => {22699} 0.09375000 0.9642857  0.09722222 5.785714 27
## [4]  {22697}       => {22699} 0.11458333 0.9166667  0.12500000 5.500000 33
## [5]  {22697,22698} => {22699} 0.08680556 0.9615385  0.09027778 5.769231 25
## [6]  {22698,22699} => {22697} 0.08680556 0.9259259  0.09375000 7.407407 25
## [7]  {22697,22699} => {22698} 0.08680556 0.7575758  0.11458333 7.792208 25
## [8]  {22697,22699} => {22423} 0.08680556 0.7575758  0.11458333 3.072983 25
## [9]  {22423,22697} => {22699} 0.08680556 1.0000000  0.08680556 6.000000 25
## [10] {22423,22699} => {22697} 0.08680556 0.7812500  0.11111111 6.250000 25
```

```
summary(eire.rules)
```

```
## set of 10 rules
##
## rule length distribution (lhs + rhs):sizes
## 2 3
## 4 6
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     2.0     3.0     2.6     3.0     3.0
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.08681   Min.   :0.7222   Min.   :0.08681   Min.   :3.073
##  1st Qu.:0.08681   1st Qu.:0.7635   1st Qu.:0.09462   1st Qu.:5.773
##  Median :0.08681   Median :0.9213   Median :0.10417   Median :6.125
##  Mean   :0.09097   Mean   :0.8716   Mean   :0.10556   Mean   :6.243
##  3rd Qu.:0.09028   3rd Qu.:0.9533   3rd Qu.:0.11458   3rd Qu.:7.423
##  Max.   :0.11458   Max.   :1.0000   Max.   :0.12500   Max.   :7.792
##      count
##  Min.   :25.0
##  1st Qu.:25.0
##  Median :25.0
##  Mean   :26.2
##  3rd Qu.:26.0
##  Max.   :33.0
##
## mining info:
##  data ntransactions support confidence
##  eire           288    0.08        0.7
```

Return only rules with a lift value greater than 5

```
subset.rules = eire.rules[quality(eire.rules)$lift > 5]
inspect(subset.rules)
```

```
##     lhs              rhs     support    confidence coverage   lift     count
## [1] {22698}       => {22697} 0.09027778 0.9285714  0.09722222 7.428571 26
## [2] {22697}       => {22698} 0.09027778 0.7222222  0.12500000 7.428571 26
## [3] {22698}       => {22699} 0.09375000 0.9642857  0.09722222 5.785714 27
## [4] {22697}       => {22699} 0.11458333 0.9166667  0.12500000 5.500000 33
## [5] {22697,22698} => {22699} 0.08680556 0.9615385  0.09027778 5.769231 25
## [6] {22698,22699} => {22697} 0.08680556 0.9259259  0.09375000 7.407407 25
## [7] {22697,22699} => {22698} 0.08680556 0.7575758  0.11458333 7.792208 25
## [8] {22423,22697} => {22699} 0.08680556 1.0000000  0.08680556 6.000000 25
## [9] {22423,22699} => {22697} 0.08680556 0.7812500  0.11111111 6.250000 25
```

These rules have at least 8% support, 72% confidence, and 5.5 lift.

There corresponding item descriptions for the relevant stock codes are as follows:

StockCode	Item Description
22423	REGENCY_CAKESTAND_3_TIER

StockCode	Item Description
22697	GREEN_REGENCY_TEACUP_AND_SAUCER
22698	PINK_REGENCY_TEACUP_AND_SAUCER
22699	ROSES_REGENCY_TEACUP_AND_SAUCER

3/4 items are teacup and saucer pairs. Transactions that involved the Regency Cakestand and Green Regency Teacup and Saucer also included the Roses Regency Teacup and Saucer 100% of the time. The rules with the highest lift value are rules 7, 1, and 2.