COMP Assignment 2

Denecian Dennis

3/31/2021

Preliminary

Let us first import all the necessary libraries.

```
library('arules')
```

```
## Warning: package 'arules' was built under R version 4.0.4

## Loading required package: Matrix

##
## Attaching package: 'arules'

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

```
library('backports')
library('zeallot')
```

```
## Warning: package 'zeallot' was built under R version 4.0.4
```

```
library('arulesViz')
```

```
## Warning: package 'arulesViz' was built under R version 4.0.4
```

```
library('chron')
```

```
## Warning: package 'chron' was built under R version 4.0.4
```

Importing the online retail data set.

```
ORData <- read.csv(file.choose(),stringsAsFactors = TRUE)
```

Looking at the summary of the data below.

```
summary(ORData)
```

```
##    InvoiceNo        StockCode                               Description
##  573585 :  1114   85123A :  2313   WHITE HANGING HEART T-LIGHT HOLDER:  2369
##  581219 :   749   22423  :  2203   REGENCY CAKESTAND 3 TIER          :  2200
##  581492 :   731   85099B :  2159   JUMBO BAG RED RETROSPOT           :  2159
##  580729 :   721   47566  :  1727   PARTY BUNTING                     :  1727
##  558475 :   705   20725  :  1639   LUNCH BAG RED RETROSPOT           :  1638
##  579777 :   687   84879  :  1502   ASSORTED COLOUR BIRD ORNAMENT     :  1501
##  (Other):537202   (Other):530366   (Other)                           :530315
##     Quantity                   InvoiceDate       UnitPrice
##  Min.   :-80995.00   10/31/2011 14:41:  1114   Min.   :-11062.06
```

```
##  1st Qu.:     1.00   12/8/2011 9:28  :   749   1st Qu.:     1.25
##  Median :     3.00   12/9/2011 10:03 :   731   Median :     2.08
##  Mean   :     9.55   12/5/2011 17:24 :   721   Mean   :     4.61
##  3rd Qu.:    10.00   6/29/2011 15:58 :   705   3rd Qu.:     4.13
##  Max.   : 80995.00   11/30/2011 15:13:   687   Max.   : 38970.00
##                      (Other)         :537202
##    CustomerID               Country
##  Min.   :12346    United Kingdom:495478
##  1st Qu.:13953    Germany       :  9495
##  Median :15152    France        :  8557
##  Mean   :15288    EIRE          :  8196
##  3rd Qu.:16791    Spain         :  2533
##  Max.   :18287    Netherlands   :  2371
##  NA's   :135080   (Other)       : 15279
```

Looking at the summary of the `country` attribute of the data set.

```
summary(ORData$Country)
```

```
##            Australia              Austria              Bahrain
##                 1259                  401                   19
##              Belgium               Brazil               Canada
##                 2069                   32                  151
##      Channel Islands               Cyprus       Czech Republic
##                  758                  622                   30
##              Denmark                 EIRE   European Community
##                  389                 8196                   61
##              Finland               France              Germany
##                  695                 8557                 9495
##               Greece            Hong Kong              Iceland
##                  146                  288                  182
##               Israel                Italy                Japan
##                  297                  803                  358
##              Lebanon            Lithuania                Malta
##                   45                   35                  127
##          Netherlands               Norway               Poland
##                 2371                 1086                  341
##             Portugal                  RSA         Saudi Arabia
##                 1519                   58                   10
##            Singapore                Spain               Sweden
##                  229                 2533                  462
##          Switzerland United Arab Emirates       United Kingdom
##                 2002                   68               495478
##          Unspecified                  USA
##                  446                  291
```

Data Exploration

I was assigned the country Australia to identify association rules that can be used by the country manager to understand the buying patterns of their customers. Let us first extract the data relating to the country Australia.

```
Aust <- ORData[ORData$Country=='Australia',]
head(Aust)
```

```
##     InvoiceNo StockCode                        Description Quantity
## 198    536389     22941       CHRISTMAS LIGHTS 10 REINDEER        6
## 199    536389     21622   VINTAGE UNION JACK CUSHION COVER        8
## 200    536389     21791 VINTAGE HEADS AND TAILS CARD GAME       12
## 201    536389    35004C     SET OF 3 COLOURED  FLYING DUCKS        6
## 202    536389    35004G          SET OF 3 GOLD FLYING DUCKS        4
## 203    536389    85014B              RED RETROSPOT UMBRELLA        6
##         InvoiceDate UnitPrice CustomerID   Country
## 198 12/1/2010 10:03      8.50      12431 Australia
## 199 12/1/2010 10:03      4.95      12431 Australia
## 200 12/1/2010 10:03      1.25      12431 Australia
## 201 12/1/2010 10:03      5.45      12431 Australia
## 202 12/1/2010 10:03      6.35      12431 Australia
## 203 12/1/2010 10:03      5.95      12431 Australia
```

Removing the unwanted attributes

For this analysis we do not require the `Invoice Date` field and given that all the data is relating to `Australia`, we can also remove this attribute.

```
Aust$InvoiceDate <-NULL
Aust$Country <- NULL
rownames(Aust) <- NULL
```

Missing Values

```
apply(Aust,2,function(k) sum(is.na(k)))
```

```
##   InvoiceNo   StockCode Description    Quantity   UnitPrice  CustomerID
##           0           0           0           0           0           0
```

Here we see that there are no missing values in the data set.

Noise

```
str(Aust)
```

```
## 'data.frame':    1259 obs. of  6 variables:
##  $ InvoiceNo  : Factor w/ 25900 levels "536365","536366",..: 23 23 23 23 23 23 23 23 23 23 ...
##  $ StockCode  : Factor w/ 4070 levels "10002","10080",..: 1845 713 839 2511 2512 3413 3411 1144 1637
##  $ Description: Factor w/ 4224 levels ""," 4 PURPLE FLOCK DINNER CANDLES",..: 814 3927 3892 3274 327
##  $ Quantity   : int  6 8 12 6 4 6 3 2 4 4 ...
##  $ UnitPrice  : num  8.5 4.95 1.25 5.45 6.35 5.95 5.95 8.5 3.75 3.75 ...
##  $ CustomerID : int  12431 12431 12431 12431 12431 12431 12431 12431 12431 12431 ...
```

We can observe leading white spaces in the `Description` attribute. Let us remove these now.

```
Aust$Description <- trimws(Aust$Description)
Aust$Description<-gsub(" ","_",Aust$Description)
str(Aust$Description)
```

```
##  chr [1:1259] "CHRISTMAS_LIGHTS_10_REINDEER" ...
```

Outliers

```
summary(Aust[,c('Quantity', 'UnitPrice')])
```

```
##     Quantity         UnitPrice
##  Min.   :-120.00   Min.   :  0.000
##  1st Qu.:   6.00   1st Qu.:  1.250
##  Median :  24.00   Median :  1.790
##  Mean   :  66.44   Mean   :  3.221
##  3rd Qu.:  96.00   3rd Qu.:  3.750
##  Max.   :1152.00   Max.   :350.000
```

We have seen that there exist no missing values in all the attributes. Given that, quantity and price are the two numerical variables left to check for outliers.

We see that there exist negative values for quantity. Since the data contain transactions, we can assume that these negative values are refunds made by customers.

Importing Data as Transaction Objects

```r
write.csv(Aust, file='2021-clean-australia.csv', row.names = FALSE)

AustData <-read.transactions('2021-clean-australia.csv', format = c('single'),header = TRUE, rm.duplica
```

Association Rules

```r
summary(AustData)
```

```
## transactions as itemMatrix in sparse format with
##  69 rows (elements/itemsets/transactions) and
##  600 columns (items) and a density of 0.03036232
##
## most frequent items:
##   22720   20725   21731   22090   22138 (Other)
##      10       9       9       8       8    1213
##
## element (itemset/transaction) length distribution:
## sizes
##   1   2   3   4   5   6   7   8   9  10  13  14  16  17  19  20  22  23  24  26
##  15  10   2   2   2   1   1   4   1   3   1   1   1   2   2   2   2   3   1   1
##  27  34  35  46  57  69  73  81  82  97 138
##   1   1   1   1   2   1   1   1   1   1   1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.00    8.00   18.22   22.00  138.00
##
## includes extended item information - examples:
##    labels
## 1   15036
## 2 15056BL
## 3  16161P
##
## includes extended transaction information - examples:
##   transactionID
## 1        536389
## 2        537676
## 3        539419
```

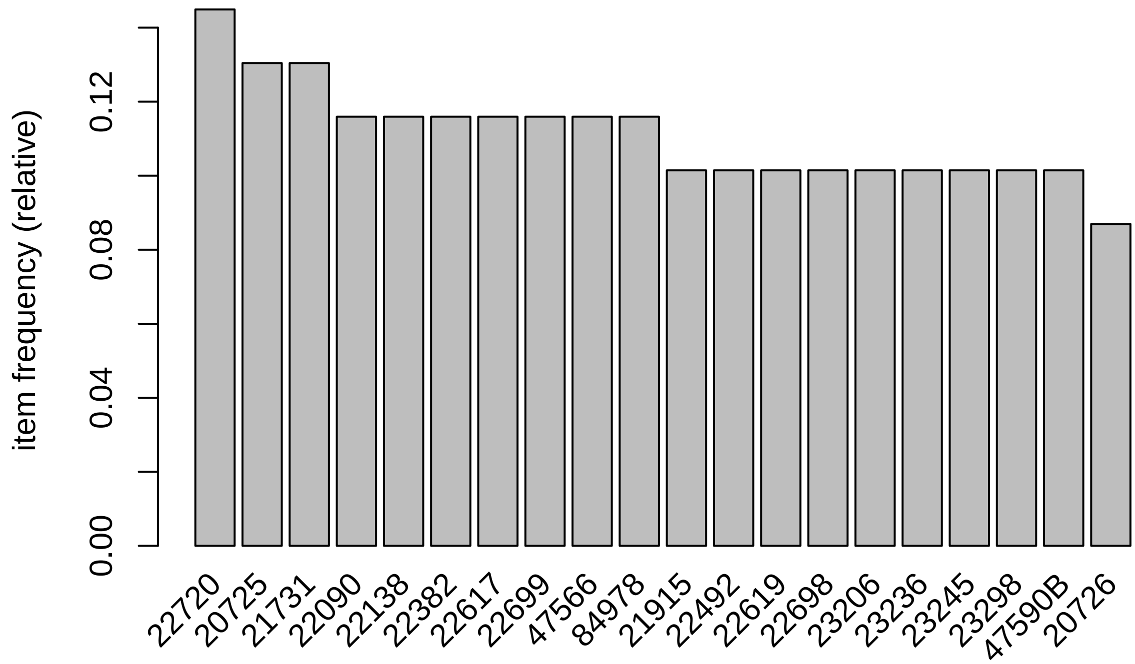
```r
s_lst <- c(22720, 20725, 21731, 22090, 22138)
d_lst<-list()

for (i in 1:length(s_lst)){
  print(Aust[Aust$StockCode==s_lst[i],]$Description)

}
```

```
##  [1] "SET_OF_3_CAKE_TINS_PANTRY_DESIGN" "SET_OF_3_CAKE_TINS_PANTRY_DESIGN"
##  [3] "SET_OF_3_CAKE_TINS_PANTRY_DESIGN" "SET_OF_3_CAKE_TINS_PANTRY_DESIGN"
##  [5] "SET_OF_3_CAKE_TINS_PANTRY_DESIGN" "SET_OF_3_CAKE_TINS_PANTRY_DESIGN"
##  [7] "SET_OF_3_CAKE_TINS_PANTRY_DESIGN" "SET_OF_3_CAKE_TINS_PANTRY_DESIGN"
##  [9] "SET_OF_3_CAKE_TINS_PANTRY_DESIGN" "SET_OF_3_CAKE_TINS_PANTRY_DESIGN"
## [1] "LUNCH_BAG_RED_RETROSPOT" "LUNCH_BAG_RED_RETROSPOT"
## [3] "LUNCH_BAG_RED_RETROSPOT" "LUNCH_BAG_RED_RETROSPOT"
## [5] "LUNCH_BAG_RED_RETROSPOT" "LUNCH_BAG_RED_RETROSPOT"
## [7] "LUNCH_BAG_RED_RETROSPOT" "LUNCH_BAG_RED_RETROSPOT"
## [9] "LUNCH_BAG_RED_RETROSPOT"
## [1] "RED_TOADSTOOL_LED_NIGHT_LIGHT" "RED_TOADSTOOL_LED_NIGHT_LIGHT"
## [3] "RED_TOADSTOOL_LED_NIGHT_LIGHT" "RED_TOADSTOOL_LED_NIGHT_LIGHT"
## [5] "RED_TOADSTOOL_LED_NIGHT_LIGHT" "RED_TOADSTOOL_LED_NIGHT_LIGHT"
## [7] "RED_TOADSTOOL_LED_NIGHT_LIGHT" "RED_TOADSTOOL_LED_NIGHT_LIGHT"
## [9] "RED_TOADSTOOL_LED_NIGHT_LIGHT"
## [1] "PAPER_BUNTING_RETROSPOT" "PAPER_BUNTING_RETROSPOT"
## [3] "PAPER_BUNTING_RETROSPOT" "PAPER_BUNTING_RETROSPOT"
## [5] "PAPER_BUNTING_RETROSPOT" "PAPER_BUNTING_RETROSPOT"
## [7] "PAPER_BUNTING_RETROSPOT" "PAPER_BUNTING_RETROSPOT"
## [1] "BAKING_SET_9_PIECE_RETROSPOT" "BAKING_SET_9_PIECE_RETROSPOT"
## [3] "BAKING_SET_9_PIECE_RETROSPOT" "BAKING_SET_9_PIECE_RETROSPOT"
## [5] "BAKING_SET_9_PIECE_RETROSPOT" "BAKING_SET_9_PIECE_RETROSPOT"
## [7] "BAKING_SET_9_PIECE_RETROSPOT" "BAKING_SET_9_PIECE_RETROSPOT"
```

We can see that the top 5 most frequent items bought are:

Frequent Item	No of times Bought
SET_OF_3_CAKE_TINS_PANTRY_DESIGN	10
LUNCH_BAG_RED_RETROSPOT	9
RED_TOADSTOOL_LED_NIGHT_LIGHT	9
PAPER_BUNTING_RETROSPOT	8
BAKING_SET_9_PIECE_RETROSPOT	8

```r
itemFrequencyPlot(AustData, topN=20)
```



Here we see the top 20 most frequent item bought in Australia by stick code.

```
Aust.rules <- apriori(AustData, parameter = list(conf=0.90, supp=0.07, minlen=2, maxlen=3, target='rules
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5    0.07      2
##  maxlen target  ext
##       3  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 4
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[600 item(s), 69 transaction(s)] done [0.00s].
## sorting and recoding items ... [54 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(AustData, parameter = list(conf = 0.9, supp = 0.07, minlen
## = 2, : Mining stopped (maxlen reached). Only patterns up to a length of 3
## returned!
```

```
##  done [0.00s].
## writing ... [54 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(Aust.rules)
```

```
##      lhs                rhs       support    confidence coverage   lift      count
## [1]  {47590A}        => {47590B} 0.07246377 1          0.07246377  9.857143 5
## [2]  {22726}         => {22727}  0.08695652 1          0.08695652 11.500000 6
## [3]  {22727}         => {22726}  0.08695652 1          0.08695652 11.500000 6
## [4]  {23296}         => {23295}  0.07246377 1          0.07246377 13.800000 5
## [5]  {23295}         => {23296}  0.07246377 1          0.07246377 13.800000 5
## [6]  {23296}         => {23294}  0.07246377 1          0.07246377 13.800000 5
## [7]  {23294}         => {23296}  0.07246377 1          0.07246377 13.800000 5
## [8]  {23296}         => {23293}  0.07246377 1          0.07246377 13.800000 5
## [9]  {23293}         => {23296}  0.07246377 1          0.07246377 13.800000 5
## [10] {23295}         => {23294}  0.07246377 1          0.07246377 13.800000 5
## [11] {23294}         => {23295}  0.07246377 1          0.07246377 13.800000 5
## [12] {23295}         => {23293}  0.07246377 1          0.07246377 13.800000 5
## [13] {23293}         => {23295}  0.07246377 1          0.07246377 13.800000 5
## [14] {23294}         => {23293}  0.07246377 1          0.07246377 13.800000 5
## [15] {23293}         => {23294}  0.07246377 1          0.07246377 13.800000 5
## [16] {22662}         => {22382}  0.07246377 1          0.07246377  8.625000 5
## [17] {23174}         => {22138}  0.07246377 1          0.07246377  8.625000 5
## [18] {22631}         => {22630}  0.07246377 1          0.07246377 11.500000 5
## [19] {22631}         => {22629}  0.07246377 1          0.07246377 11.500000 5
## [20] {21843}         => {20979}  0.07246377 1          0.07246377 13.800000 5
## [21] {20979}         => {21843}  0.07246377 1          0.07246377 13.800000 5
## [22] {21843}         => {22720}  0.07246377 1          0.07246377  6.900000 5
## [23] {20979}         => {22720}  0.07246377 1          0.07246377  6.900000 5
## [24] {22630}         => {22629}  0.08695652 1          0.08695652 11.500000 6
## [25] {22629}         => {22630}  0.08695652 1          0.08695652 11.500000 6
## [26] {23295,23296} => {23294}  0.07246377 1          0.07246377 13.800000 5
## [27] {23294,23296} => {23295}  0.07246377 1          0.07246377 13.800000 5
## [28] {23294,23295} => {23296}  0.07246377 1          0.07246377 13.800000 5
## [29] {23295,23296} => {23293}  0.07246377 1          0.07246377 13.800000 5
## [30] {23293,23296} => {23295}  0.07246377 1          0.07246377 13.800000 5
## [31] {23293,23295} => {23296}  0.07246377 1          0.07246377 13.800000 5
## [32] {23294,23296} => {23293}  0.07246377 1          0.07246377 13.800000 5
## [33] {23293,23296} => {23294}  0.07246377 1          0.07246377 13.800000 5
## [34] {23293,23294} => {23296}  0.07246377 1          0.07246377 13.800000 5
## [35] {23294,23295} => {23293}  0.07246377 1          0.07246377 13.800000 5
## [36] {23293,23295} => {23294}  0.07246377 1          0.07246377 13.800000 5
## [37] {23293,23294} => {23295}  0.07246377 1          0.07246377 13.800000 5
## [38] {22630,22631} => {22629}  0.07246377 1          0.07246377 11.500000 5
## [39] {22629,22631} => {22630}  0.07246377 1          0.07246377 11.500000 5
## [40] {22720,23245} => {23236}  0.07246377 1          0.07246377  9.857143 5
## [41] {22720,23236} => {23245}  0.07246377 1          0.07246377  9.857143 5
## [42] {20979,21843} => {22720}  0.07246377 1          0.07246377  6.900000 5
## [43] {21843,22720} => {20979}  0.07246377 1          0.07246377 13.800000 5
## [44] {20979,22720} => {21843}  0.07246377 1          0.07246377 13.800000 5
## [45] {22423,22630} => {22629}  0.07246377 1          0.07246377 11.500000 5
## [46] {22423,22629} => {22630}  0.07246377 1          0.07246377 11.500000 5
## [47] {22630,22699} => {22629}  0.07246377 1          0.07246377 11.500000 5
## [48] {22629,22699} => {22630}  0.07246377 1          0.07246377 11.500000 5
```

```
## [49] {22423,22630} => {22699}  0.07246377 1          0.07246377  8.625000 5
## [50] {22630,22699} => {22423}  0.07246377 1          0.07246377 11.500000 5
## [51] {22423,22699} => {22630}  0.07246377 1          0.07246377 11.500000 5
## [52] {22423,22629} => {22699}  0.07246377 1          0.07246377  8.625000 5
## [53] {22629,22699} => {22423}  0.07246377 1          0.07246377 11.500000 5
## [54] {22423,22699} => {22629}  0.07246377 1          0.07246377 11.500000 5
```

With a confidence level of 90% and support of support of 7% we have generated 54 association rules.

```
summary(Aust.rules)
```

```
## set of 54 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3
## 25 29
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   3.000   2.537   3.000   3.000
##
## summary of quality measures:
##     support          confidence    coverage            lift
##  Min.   :0.07246   Min.   :1    Min.   :0.07246   Min.   : 6.90
##  1st Qu.:0.07246   1st Qu.:1    1st Qu.:0.07246   1st Qu.:11.50
##  Median :0.07246   Median :1    Median :0.07246   Median :13.80
##  Mean   :0.07354   Mean   :1    Mean   :0.07354   Mean   :12.13
##  3rd Qu.:0.07246   3rd Qu.:1    3rd Qu.:0.07246   3rd Qu.:13.80
##  Max.   :0.08696   Max.   :1    Max.   :0.08696   Max.   :13.80
##      count
##  Min.   :5.000
##  1st Qu.:5.000
##  Median :5.000
##  Mean   :5.074
##  3rd Qu.:5.000
##  Max.   :6.000
##
## mining info:
##      data ntransactions support confidence
##  AustData            69    0.07        0.9
```

From the summary statistics we see that with the minimun and maximun confidence and support levels we have generated 25 rules with length of 2 and 29 rules with length of 3.

We can also see the statistics for the lift of the 54 rules generated. Given that the median and maximum lift values and the lift value within the 3rd quartile are all the same; we can assume that the distribution of the life values are skewed to the right.

Let us filter the rules for this lift value in an effort to narrow down on the strongest rules.

```
subset.rules <- Aust.rules[quality(Aust.rules)$lift>13]
inspect(subset.rules)
```

```
##      lhs              rhs     support    confidence coverage   lift count
## [1]  {23296}       => {23295} 0.07246377 1          0.07246377 13.8 5
## [2]  {23295}       => {23296} 0.07246377 1          0.07246377 13.8 5
## [3]  {23296}       => {23294} 0.07246377 1          0.07246377 13.8 5
## [4]  {23294}       => {23296} 0.07246377 1          0.07246377 13.8 5
```

```
## [5]  {23296}       => {23293} 0.07246377 1          0.07246377 13.8 5
## [6]  {23293}       => {23296} 0.07246377 1          0.07246377 13.8 5
## [7]  {23295}       => {23294} 0.07246377 1          0.07246377 13.8 5
## [8]  {23294}       => {23295} 0.07246377 1          0.07246377 13.8 5
## [9]  {23295}       => {23293} 0.07246377 1          0.07246377 13.8 5
## [10] {23293}       => {23295} 0.07246377 1          0.07246377 13.8 5
## [11] {23294}       => {23293} 0.07246377 1          0.07246377 13.8 5
## [12] {23293}       => {23294} 0.07246377 1          0.07246377 13.8 5
## [13] {21843}       => {20979} 0.07246377 1          0.07246377 13.8 5
## [14] {20979}       => {21843} 0.07246377 1          0.07246377 13.8 5
## [15] {23295,23296} => {23294} 0.07246377 1          0.07246377 13.8 5
## [16] {23294,23296} => {23295} 0.07246377 1          0.07246377 13.8 5
## [17] {23294,23295} => {23296} 0.07246377 1          0.07246377 13.8 5
## [18] {23295,23296} => {23293} 0.07246377 1          0.07246377 13.8 5
## [19] {23293,23296} => {23295} 0.07246377 1          0.07246377 13.8 5
## [20] {23293,23295} => {23296} 0.07246377 1          0.07246377 13.8 5
## [21] {23294,23296} => {23293} 0.07246377 1          0.07246377 13.8 5
## [22] {23293,23296} => {23294} 0.07246377 1          0.07246377 13.8 5
## [23] {23293,23294} => {23296} 0.07246377 1          0.07246377 13.8 5
## [24] {23294,23295} => {23293} 0.07246377 1          0.07246377 13.8 5
## [25] {23293,23295} => {23294} 0.07246377 1          0.07246377 13.8 5
## [26] {23293,23294} => {23295} 0.07246377 1          0.07246377 13.8 5
## [27] {21843,22720} => {20979} 0.07246377 1          0.07246377 13.8 5
## [28] {20979,22720} => {21843} 0.07246377 1          0.07246377 13.8 5
```

The next thing we want to do is filter the data for redundant rules.

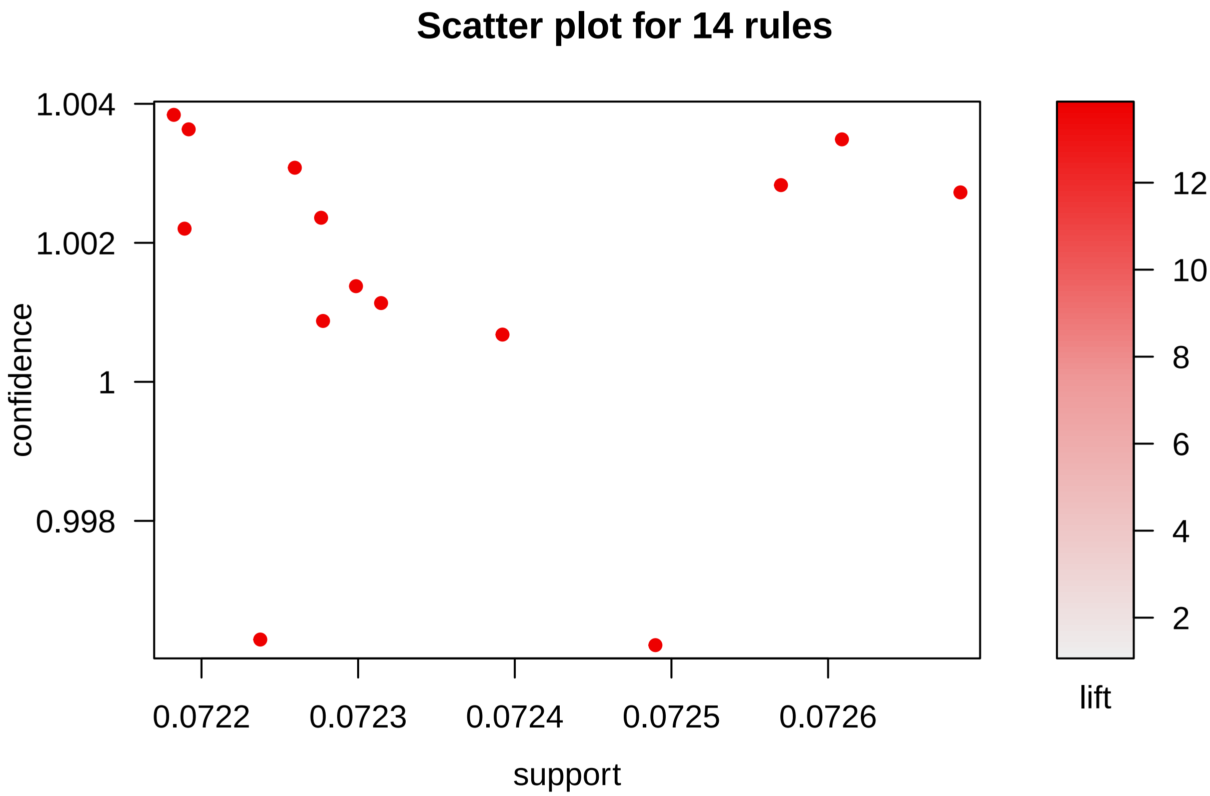
```
subset.rules <- subset.rules[!is.redundant(subset.rules)]
inspect(subset.rules)
```

```
##      lhs        rhs     support    confidence coverage   lift count
## [1]  {23296} => {23295} 0.07246377 1          0.07246377 13.8 5
## [2]  {23295} => {23296} 0.07246377 1          0.07246377 13.8 5
## [3]  {23296} => {23294} 0.07246377 1          0.07246377 13.8 5
## [4]  {23294} => {23296} 0.07246377 1          0.07246377 13.8 5
## [5]  {23296} => {23293} 0.07246377 1          0.07246377 13.8 5
## [6]  {23293} => {23296} 0.07246377 1          0.07246377 13.8 5
## [7]  {23295} => {23294} 0.07246377 1          0.07246377 13.8 5
## [8]  {23294} => {23295} 0.07246377 1          0.07246377 13.8 5
## [9]  {23295} => {23293} 0.07246377 1          0.07246377 13.8 5
## [10] {23293} => {23295} 0.07246377 1          0.07246377 13.8 5
## [11] {23294} => {23293} 0.07246377 1          0.07246377 13.8 5
## [12] {23293} => {23294} 0.07246377 1          0.07246377 13.8 5
## [13] {21843} => {20979} 0.07246377 1          0.07246377 13.8 5
## [14] {20979} => {21843} 0.07246377 1          0.07246377 13.8 5
```

From this step we have reduced the number of rules from 28 to 14.

```
plot(subset.rules)
```

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



Here we see a scatter plot of the 14 association rules.

```
plot(subset.rules, method = "graph",  engine = "htmlwidget")
```

The visualization above shows a mapping of each rules and how each items rules connect to each rule.

```
plot(subset.rules, method = "matrix",  engine = "htmlwidget")
```

Using the association rules above, we can note the following:

1. There exist strong association rules developing around six (6)items.
2. StockNo 21843 and 20979 have strong association rules between the stocks. Additional, no rules exists between these two stocks and the other four (4) stocks.
3. All the fourteen (14) rules have a confidence of one (1) which means 100
4. All fourteen (14) rules suggest high co-occurrence and should be looked into to gain a better understanding of the buying patterns of the customers in Australia.