Market Basket Analysis

Jennifer Brosnahan

7/20/2020

The purpose of this project is to conduct a Market Basket Analysis on 30 days worth of Electronidex transactions and try to find patterns on Electronidex item purchases. I will also provide recommendations (cross-selling, recommenders, email promotions, etc) based on any insights gained from this analysis.

Loading packages

```
library(tidyverse)
library(openxlsx)
library(knitr)
library(ggplot2)
library(arules)
library(arulesViz)
library(dplyr)
library(kableExtra)
```

Importing data using read.transactions() function for Market Basket analysis

```
trans <- read.transactions(file.path('C:/Users/jlbro/OneDrive/C3T4', 'ElectronidexTransactions.csv'), format = 'basket', sep=',', rm.duplicates=TRUE)
```

```
## distribution of transactions with duplicates:
## items
##   1   2
## 191  10
```

Reviewing summary of data

```
summary(trans)
```

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  125 columns (items) and a density of 0.03506172
##
## most frequent items:
##                     iMac                HP Laptop CYBERPOWER Gamer Desktop
##                     2519                     1909                     1809
##            Apple Earpods         Apple MacBook Air                   (Other)
##                     1715                     1530                    33622
##
## element (itemset/transaction) length distribution:
## sizes
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##    2 2163 1647 1294 1021  856  646  540  439  353  247  171  119   77   72   56
##   16   17   18   19   20   21   22   23   25   26   27   29   30
##   41   26   20   10   10   10    5    3    1    1    3    1    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   4.383   6.000  30.000
##
## includes extended item information - examples:
##                             labels
## 1 1TB Portable External Hard Drive
## 2 2TB Portable External Hard Drive
## 3                   3-Button Mouse
```

Viewing 5 transaction baskets

```
inspect(trans[1:5])
```

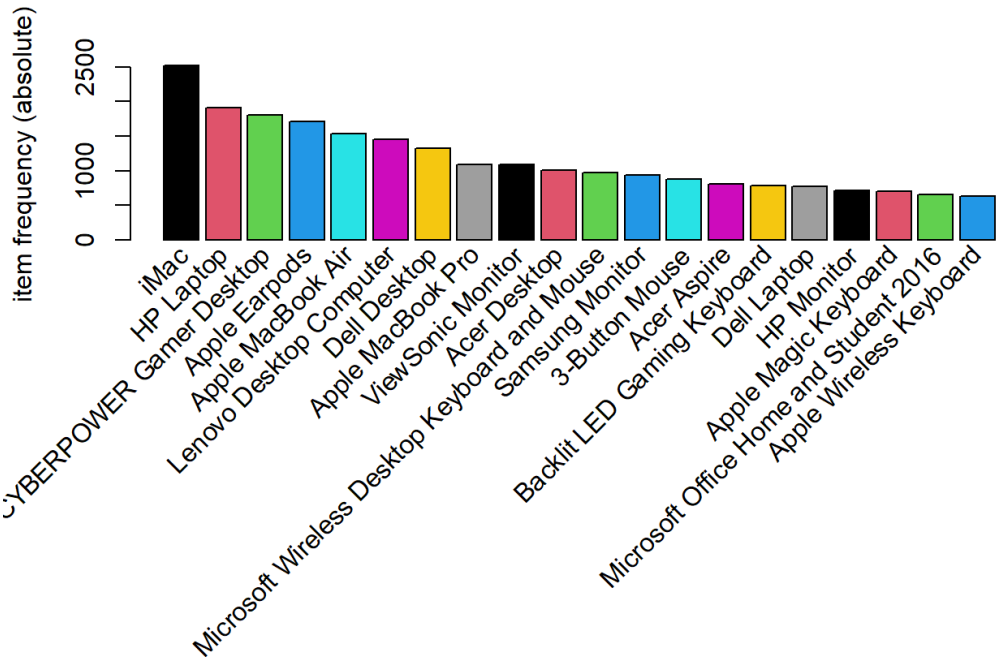
```
##     items
## [1] {Acer Aspire,
##      Belkin Mouse Pad,
##      Brother Printer Toner,
##      VGA Monitor Cable}
## [2] {Apple Wireless Keyboard,
##      Dell Desktop,
##      Lenovo Desktop Computer}
## [3] {iMac}
## [4] {Acer Desktop,
##      Intel Desktop,
##      Lenovo Desktop Computer,
##      XIBERIA Gaming Headset}
## [5] {ASUS Desktop,
##      Epson Black Ink,
##      HP Laptop,
##      iMac}
```

Viewing item frequency. Output defaults to alpha order.

```
itemFrequency(trans, type = 'absolute')
```

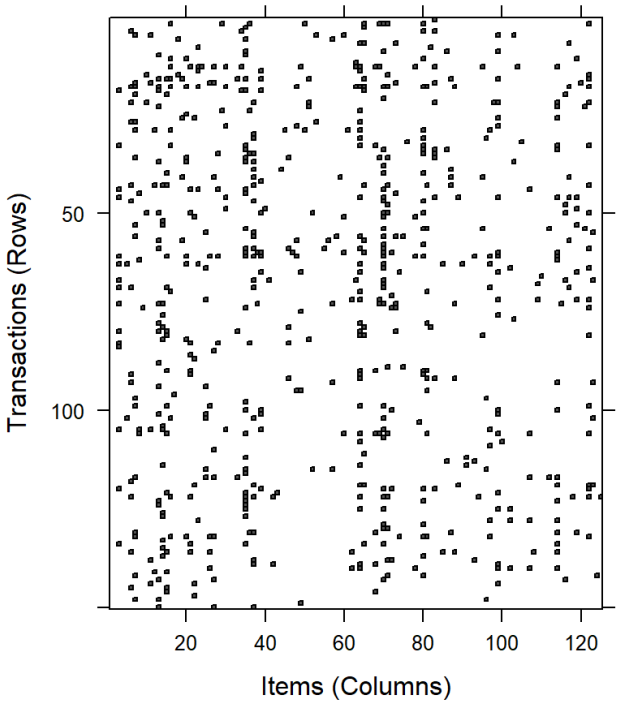
Plotting top 20 most frequently purchased items

```
itemFrequencyPlot(trans, topN=20, type = 'absolute', col = 1:20)
```



Now viewing image of a sample of purchases. Graph reveals clusters of products purchased, however, you cannot see actual product numbers

```
image(sample(trans, 150))
```



Viewing bottom frequency items for possible liquidation purposes

```
Item_frequency <- data.frame(sort(itemFrequency(trans, type = 'absolute'), decreasing = FALSE))

colnames(Item_frequency) <- c('Frequency')

kable(Item_frequency, caption = 'Viewing Frequency from Bottom on Up') %>%
  kable_styling(bootstrap_options = c('striped','hover'))
```

Viewing Frequency from Bottom on Up

	Frequency
Logitech Wireless Keyboard	22
VGA Monitor Cable	22
Panasonic On-Ear Stereo Headphones	23
1TB Portable External Hard Drive	27
Canon Ink	27
Logitech Stereo Headset	30
Ethernet Cable	32
Canon Office Printer	35
Gaming Mouse Professional	35
Audio Cable	36
Logitech Multimedia Speakers	38
5TB Desktop Hard Drive	41
Roku Express	41
XIBERIA Gaming Headset	42
Philips Flexible Earhook Headphone	44
Samsung Galaxy Tablet	45
HP Notebook Touchscreen Laptop PC	51
Kindle	52
HDMI Adapter	53
EagleTec Wireless Combo Keyboard and Mouse	55
Generic Black 3-Button	56
DYMO Label Manker	57
Multi Media Stand	59
2TB Portable External Hard Drive	60
USB Cable	62
DOSS Touch Wireless Bluetooth	64

	Frequency
Kensington Headphones	64
3TB Portable External Hard Drive	68
Microsoft Wireless Comfort Keyboard and Mouse	71
Mackie CR Speakers	75
Samsung Charging Cable	76
Large Mouse Pad	78
Logitech MK360 Wireless Keyboard and Mouse Combo	79
Logitech Desktop MK120 Mouse and keyboard Combo	80
Dell Monitor	83
Google Home	84
HP USB Keyboard	84
Sceptre Monitor	88
Fire TV Stick	89
Logitech ClearChat Headset	89
Height-Adjustable Standing Desk	90
Rokono Mini Speaker	91
APIE Bluetooth Headphone	93
Halter Mesh Metal Monitor Stand	102
Full Motion Monitor Mount	106
Apple Wired Keyboard	110
DYMO Labeling Tape	112
Fire HD Tablet	118
Logitech Wireless Mouse	128
Rii LED Keyboard	130
Monster Beats By Dr Dre	132
PC Gaming Headset	140
Ailihen Stereo Headphones	144
Cyber Acoustics	148
Halter Acrylic Monitor Stand	148
Apple TV	151
HP Black & Tri-color Ink	160

	Frequency
Dell Wired Keyboard	163
Koss Home Headphones	168
Rii LED Gaming Keyboard & Mouse Combo	171
JBL Splashproof Portable Bluetooth Speaker	173
Brother Printer Toner	174
Smart Light Bulb	174
Microsoft Basic Optical Mouse	176
Epson Black Ink	181
iPhone Charger Cable	187
LG Touchscreen Laptop	187
Dell KM117 Wireless Keyboard & Mouse	189
Microsoft Headset	207
Logitech MK270 Wireless Keyboard and Mouse Combo	220
Zombie Gaming Headset	220
Redragon Gaming Mouse	229
Logitech MK550 Wireless Wave Keyboard and Mouse Combo	241
Panasonic In-Ear Headphone	245
Sonos	246
Otium Wireless Sports Bluetooth Headphone	254
HDMI Cable 6ft	256
Slim 2TB Portable External Hard Drive	256
HP Desktop	269
Logitech Keyboard	272
Alienware Laptop	275
Cambridge Bluetooth Speaker	279
Etekcity Power Extension Cord Cable	302
Brother Printer	324
Dell 2 Desktop	327
Bose Companion Speakers	329
iPad	336
Eluktronics Pro Gaming Laptop	351

	Frequency
HP Wireless Printer	365
ASUS Desktop	368
Slim Wireless Mouse	372
Wireless Portable Mouse	378
Intel Desktop	390
Computer Game	413
AOC Monitor	414
HP Wireless Mouse	428
Epson Printer	473
iPad Pro	515
ASUS Chromebook	516
ASUS Monitor	545
ASUS 2 Monitor	567
LG Monitor	567
Belkin Mouse Pad	576
Acer Monitor	580
Apple Wireless Keyboard	638
Logitech 3-button Mouse	638
Microsoft Office Home and Student 2016	654
Apple Magic Keyboard	705
HP Monitor	711
Dell Laptop	776
Backlit LED Gaming Keyboard	785
Acer Aspire	814
3-Button Mouse	875
Samsung Monitor	941
Microsoft Wireless Desktop Keyboard and Mouse	969
Acer Desktop	1002
ViewSonic Monitor	1085
Apple MacBook Pro	1087
Dell Desktop	1318

	Frequency
Lenovo Desktop Computer	1456
Apple MacBook Air	1530
Apple Earpods	1715
CYBERPOWER Gamer Desktop	1809
HP Laptop	1909
iMac	2519

Initial observations

- 9835 transactions
- 125 total products
- 43,151 total items were purchased
- Top 5: iMac, HP Laptop, Cyberpower Gamer Desktop, Apple Earpods, Apple MacBook Air
- Items/transaction range is 0-30
- Customers purchase 1 item the most
- The average items purchased is 4.4
- The distribution of items purchased is skewed right
- The top 7 products stand out, after that, item frequency seems to level off

Now, we will start applying rules. We are aiming for useful amount of support, confidence, and lifts greater than 1 for stronger rules

```
Rules1 <- apriori(trans, parameter = list(supp = 0.1, conf = 0.8, minlen = 3)) #Covers 10% of transactions (N=983), 80%
 correct
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      3
##  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: 983
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [0 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules1, by='lift')) #Generated No rules
```

```
Rules2 <- apriori(trans, parameter = list(supp = 0.01, conf = 0.8, minlen = 3)) #Covers 1% of transactions (N=98), 80% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      3
##  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: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [82 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [0 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules2, by='lift')) #Still no rules
```

```
Rules3 <- apriori(trans, parameter = list(supp = 0.005, conf = 0.8, minlen = 3)) #Covers .05% transactions (N=49), 80% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.005      3
##  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: 49
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [109 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [1 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules3, by='lift')) #1 rule, lift = 4
```

```
##     lhs                    rhs             support confidence    coverage     lift count
## [1] {Acer Aspire,
##      Dell Desktop,
##      ViewSonic Monitor} => {HP Laptop} 0.005287239     0.8125 0.006507372 4.185928    52
```

```
Rules4 <- apriori(trans, parameter = list(supp = 0.005, conf = 0.7, minlen = 3)) #Covers .05% transactions (N=49), 70% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5   0.005      3
##  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: 49
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [109 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [3 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules4, by='lift')) #3 rules, lift = 2.7-4.1
```

```
##     lhs                          rhs             support confidence    coverage     lift count
## [1] {Acer Aspire,
##      Dell Desktop,
##      ViewSonic Monitor}       => {HP Laptop} 0.005287239  0.8125000 0.006507372 4.185928    52
## [2] {ASUS 2 Monitor,
##      Dell Desktop,
##      Lenovo Desktop Computer} => {iMac}      0.005185562  0.7391304 0.007015760 2.885807    51
## [3] {ASUS 2 Monitor,
##      ASUS Monitor}            => {iMac}      0.005083884  0.7142857 0.007117438 2.788805    50
```

```
Rules5 <- apriori(trans, parameter = list(supp = 0.005, conf = 0.6, minlen = 3)) #Covers .05% transactions (N=49), 60% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.005      3
##  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: 49
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [109 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [28 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules5, by='lift')) #28 rules, lift = 2.3-4.8
```

```
##      lhs                                        rhs             support confidence    coverage     lift count
## [1]  {Acer Aspire,
##       Dell Desktop,
##       ViewSonic Monitor}                     => {HP Laptop} 0.005287239  0.8125000 0.006507372 4.185928    52
## [2]  {Acer Aspire,
##       iMac,
##       ViewSonic Monitor}                     => {HP Laptop} 0.006202339  0.6630435 0.009354347 3.415942    61
## [3]  {Acer Desktop,
##       iMac,
##       ViewSonic Monitor}                     => {HP Laptop} 0.006405694  0.6363636 0.010066090 3.278489    63
## [4]  {Dell Desktop,
##       Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {HP Laptop} 0.006202339  0.6224490 0.009964413 3.206802    61
## [5]  {Computer Game,
##       ViewSonic Monitor}                     => {HP Laptop} 0.007422471  0.6186441 0.011997966 3.187200    73
## [6]  {Computer Game,
##       Dell Desktop}                          => {HP Laptop} 0.005693950  0.6086957 0.009354347 3.135946    56
## [7]  {Acer Aspire,
##       ViewSonic Monitor}                     => {HP Laptop} 0.010777834  0.6022727 0.017895272 3.102856   106
## [8]  {ASUS 2 Monitor,
##       Dell Desktop,
##       Lenovo Desktop Computer}               => {iMac}      0.005185562  0.7391304 0.007015760 2.885807    51
## [9]  {ASUS 2 Monitor,
##       ASUS Monitor}                          => {iMac}      0.005083884  0.7142857 0.007117438 2.788805    50
## [10] {ASUS 2 Monitor,
##       Microsoft Office Home and Student 2016} => {iMac}      0.005185562  0.6986301 0.007422471 2.727681    51
## [11] {Dell Desktop,
##       Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {iMac}      0.006914082  0.6938776 0.009964413 2.709125    68
## [12] {Apple Magic Keyboard,
##       Dell Desktop,
##       Lenovo Desktop Computer}               => {iMac}      0.005287239  0.6842105 0.007727504 2.671382    52
## [13] {Apple Magic Keyboard,
##       ASUS Monitor}                          => {iMac}      0.006812405  0.6700000 0.010167768 2.615899    67
## [14] {Acer Desktop,
##       HP Laptop,
##       ViewSonic Monitor}                     => {iMac}      0.006405694  0.6562500 0.009761057 2.562215    63
## [15] {Acer Desktop,
##       ASUS 2 Monitor}                        => {iMac}      0.006405694  0.6428571 0.009964413 2.509925    63
## [16] {ASUS Monitor,
##       ViewSonic Monitor}                     => {iMac}      0.008235892  0.6377953 0.012913066 2.490161    81
## [17] {ASUS Monitor,
##       Dell Desktop}                          => {iMac}      0.007930859  0.6341463 0.012506355 2.475915    78
## [18] {Acer Desktop,
##       HP Laptop,
##       Lenovo Desktop Computer}               => {iMac}      0.006304016  0.6326531 0.009964413 2.470085    62
## [19] {ASUS Monitor,
##       Lenovo Desktop Computer}               => {iMac}      0.009761057  0.6315789 0.015455008 2.465891    96
## [20] {ASUS 2 Monitor,
##       Dell Desktop}                          => {iMac}      0.009049314  0.6312057 0.014336553 2.464433    89
## [21] {Acer Desktop,
##       Apple Magic Keyboard}                  => {iMac}      0.006710727  0.6226415 0.010777834 2.430996    66
## [22] {ASUS Monitor,
##       Microsoft Office Home and Student 2016} => {iMac}      0.005998983  0.6145833 0.009761057 2.399534    59
## [23] {Belkin Mouse Pad,
##       Microsoft Office Home and Student 2016} => {iMac}      0.005490595  0.6136364 0.008947636 2.395837    54
## [24] {Apple MacBook Pro,
##       ASUS Monitor}                          => {iMac}      0.005388917  0.6022727 0.008947636 2.351470    53
## [25] {HP Laptop,
##       HP Monitor,
##       Lenovo Desktop Computer}               => {iMac}      0.005388917  0.6022727 0.008947636 2.351470    53
## [26] {HP Laptop,
##       Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {iMac}      0.008439248  0.6014493 0.014031520 2.348255    83
## [27] {Acer Desktop,
##       ASUS Monitor}                          => {iMac}      0.005795628  0.6000000 0.009659380 2.342596    57
```

```
## [28] {Dell Desktop,
##       Microsoft Office Home and Student 2016} => {iMac}      0.009456024  0.6000000 0.015760041 2.342596    93
```

```
Rules6 <- apriori(trans, parameter = list(supp = 0.006, conf = 0.6, minlen = 3)) #Covers .06% transactions (N=59), 60% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.006      3
##  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: 59
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [102 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [17 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules6, by='lift')) #17 rules, lift = 2.3-3.1
```

```
##      lhs                                        rhs             support confidence    coverage     lift count
## [1]  {Acer Aspire,
##       iMac,
##       ViewSonic Monitor}                     => {HP Laptop} 0.006202339  0.6630435 0.009354347 3.415942    61
## [2]  {Acer Desktop,
##       iMac,
##       ViewSonic Monitor}                     => {HP Laptop} 0.006405694  0.6363636 0.010066090 3.278489    63
## [3]  {Dell Desktop,
##       Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {HP Laptop} 0.006202339  0.6224490 0.009964413 3.206802    61
## [4]  {Computer Game,
##       ViewSonic Monitor}                     => {HP Laptop} 0.007422471  0.6186441 0.011997966 3.187200    73
## [5]  {Acer Aspire,
##       ViewSonic Monitor}                     => {HP Laptop} 0.010777834  0.6022727 0.017895272 3.102856   106
## [6]  {Dell Desktop,
##       Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {iMac}      0.006914082  0.6938776 0.009964413 2.709125    68
## [7]  {Apple Magic Keyboard,
##       ASUS Monitor}                          => {iMac}      0.006812405  0.6700000 0.010167768 2.615899    67
## [8]  {Acer Desktop,
##       HP Laptop,
##       ViewSonic Monitor}                     => {iMac}      0.006405694  0.6562500 0.009761057 2.562215    63
## [9]  {Acer Desktop,
##       ASUS 2 Monitor}                        => {iMac}      0.006405694  0.6428571 0.009964413 2.509925    63
## [10] {ASUS Monitor,
##       ViewSonic Monitor}                     => {iMac}      0.008235892  0.6377953 0.012913066 2.490161    81
## [11] {ASUS Monitor,
##       Dell Desktop}                          => {iMac}      0.007930859  0.6341463 0.012506355 2.475915    78
## [12] {Acer Desktop,
##       HP Laptop,
##       Lenovo Desktop Computer}               => {iMac}      0.006304016  0.6326531 0.009964413 2.470085    62
## [13] {ASUS Monitor,
##       Lenovo Desktop Computer}               => {iMac}      0.009761057  0.6315789 0.015455008 2.465891    96
## [14] {ASUS 2 Monitor,
##       Dell Desktop}                          => {iMac}      0.009049314  0.6312057 0.014336553 2.464433    89
## [15] {Acer Desktop,
##       Apple Magic Keyboard}                  => {iMac}      0.006710727  0.6226415 0.010777834 2.430996    66
## [16] {HP Laptop,
##       Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {iMac}      0.008439248  0.6014493 0.014031520 2.348255    83
## [17] {Dell Desktop,
##       Microsoft Office Home and Student 2016} => {iMac}      0.009456024  0.6000000 0.015760041 2.342596    93
```

```
Rules7 <- apriori(trans, parameter = list(supp = 0.007, conf = 0.6, minlen = 3)) #Covers .07% transactions (N=68), 60% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.007      3
##  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: 68
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [97 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [8 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules7, by='lift')) #8 rules, lift = 2.3-3.1
```

```
##     lhs                                        rhs             support confidence   coverage     lift count
## [1] {Computer Game,
##      ViewSonic Monitor}                     => {HP Laptop} 0.007422471  0.6186441 0.01199797 3.187200    73
## [2] {Acer Aspire,
##      ViewSonic Monitor}                     => {HP Laptop} 0.010777834  0.6022727 0.01789527 3.102856   106
## [3] {ASUS Monitor,
##      ViewSonic Monitor}                     => {iMac}      0.008235892  0.6377953 0.01291307 2.490161    81
## [4] {ASUS Monitor,
##      Dell Desktop}                          => {iMac}      0.007930859  0.6341463 0.01250635 2.475915    78
## [5] {ASUS Monitor,
##      Lenovo Desktop Computer}               => {iMac}      0.009761057  0.6315789 0.01545501 2.465891    96
## [6] {ASUS 2 Monitor,
##      Dell Desktop}                          => {iMac}      0.009049314  0.6312057 0.01433655 2.464433    89
## [7] {HP Laptop,
##      Lenovo Desktop Computer,
##      ViewSonic Monitor}                     => {iMac}      0.008439248  0.6014493 0.01403152 2.348255    83
## [8] {Dell Desktop,
##      Microsoft Office Home and Student 2016} => {iMac}      0.009456024  0.6000000 0.01576004 2.342596    93
```

```
Rules8 <- apriori(trans, parameter = list(supp = 0.008, conf = 0.6, minlen = 3)) #Covers .08% transactions (N=78), 60% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.008      3
##  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: 78
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [93 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules8, by='lift')) #6 rules, lift = 2.3-3.1
```

```
##     lhs                                        rhs             support confidence   coverage     lift count
## [1] {Acer Aspire,
##      ViewSonic Monitor}                     => {HP Laptop} 0.010777834  0.6022727 0.01789527 3.102856   106
## [2] {ASUS Monitor,
##      ViewSonic Monitor}                     => {iMac}      0.008235892  0.6377953 0.01291307 2.490161    81
## [3] {ASUS Monitor,
##      Lenovo Desktop Computer}               => {iMac}      0.009761057  0.6315789 0.01545501 2.465891    96
## [4] {ASUS 2 Monitor,
##      Dell Desktop}                          => {iMac}      0.009049314  0.6312057 0.01433655 2.464433    89
## [5] {HP Laptop,
##      Lenovo Desktop Computer,
##      ViewSonic Monitor}                     => {iMac}      0.008439248  0.6014493 0.01403152 2.348255    83
## [6] {Dell Desktop,
##      Microsoft Office Home and Student 2016} => {iMac}      0.009456024  0.6000000 0.01576004 2.342596    93
```

```
Rules9 <- apriori(trans, parameter = list(supp = 0.009, conf = 0.6, minlen = 3)) #Covers .09% transactions (N=88), 60% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.009      3
##  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: 88
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [87 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [4 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules9, by='lift')) #6 rules, lift = 1.5-2.2
```

```
##     lhs                                        rhs             support confidence   coverage     lift count
## [1] {Acer Aspire,
##      ViewSonic Monitor}                     => {HP Laptop} 0.010777834  0.6022727 0.01789527 3.102856   106
## [2] {ASUS Monitor,
##      Lenovo Desktop Computer}               => {iMac}      0.009761057  0.6315789 0.01545501 2.465891    96
## [3] {ASUS 2 Monitor,
##      Dell Desktop}                          => {iMac}      0.009049314  0.6312057 0.01433655 2.464433    89
## [4] {Dell Desktop,
##      Microsoft Office Home and Student 2016} => {iMac}      0.009456024  0.6000000 0.01576004 2.342596    93
```

```
Rules10 <- apriori(trans, parameter = list(supp = 0.01, conf = 0.6, minlen = 3)) #Covers 1% transactions (N=98), 60% cor
rect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.01      3
##  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: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [82 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [1 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules10, by='lift')) #1 rule, lift = 3.1
```

```
##     lhs                                rhs         support    confidence
## [1] {Acer Aspire,ViewSonic Monitor} => {HP Laptop} 0.01077783 0.6022727
##     coverage   lift     count
## [1] 0.01789527 3.102856 106
```

```
# This one seems best
Rules11 <- apriori(trans, parameter = list(supp = 0.01, conf = 0.5, minlen = 3)) ###Covers 1% transactions (N=98), 50% c
orrect
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      3
##  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: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[125 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [82 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [19 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(Rules11[1:10], by='lift')) #Yields 19 total association rules, lift = 1.9-3.1
```

```
##      lhs                                        rhs            support confidence   coverage     lift count
## [1]  {Acer Aspire,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01077783  0.6022727 0.01789527 3.102856   106
## [2]  {ASUS 2 Monitor,
##       Lenovo Desktop Computer}               => {iMac}      0.01087951  0.5911602 0.01840366 2.308083   107
## [3]  {Apple Magic Keyboard,
##       Dell Desktop}                          => {iMac}      0.01016777  0.5847953 0.01738688 2.283232   100
## [4]  {ASUS Monitor,
##       HP Laptop}                             => {iMac}      0.01179461  0.5829146 0.02023386 2.275889   116
## [5]  {ASUS 2 Monitor,
##       HP Laptop}                             => {iMac}      0.01108287  0.5828877 0.01901373 2.275784   109
## [6]  {HP Laptop,
##       Microsoft Office Home and Student 2016} => {iMac}      0.01291307  0.5521739 0.02338587 2.155868   127
## [7]  {Acer Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01006609  0.5439560 0.01850534 2.123782    99
## [8]  {Apple Magic Keyboard,
##       Lenovo Desktop Computer}               => {iMac}      0.01138790  0.5161290 0.02206406 2.015137   112
## [9]  {Apple Magic Keyboard,
##       HP Laptop}                             => {iMac}      0.01474326  0.5105634 0.02887646 1.993406   145
## [10] {HP Laptop,
##       HP Monitor}                            => {iMac}      0.01057448  0.5024155 0.02104728 1.961594   104
```

```
# Further inspection by support
support11 <- data.frame(inspect(sort(Rules11, by = 'support', decreasing = TRUE))) #Dell desktop and ViewSonic Monitor
 (#2) had greatest support at 1.5% (n=150 count)
```

```
##      lhs                                        rhs            support confidence   coverage     lift count
## [1]  {HP Laptop,
##       Lenovo Desktop Computer}                => {iMac}      0.02308083  0.5000000 0.04616167 1.952164   227
## [2]  {Dell Desktop,
##       Lenovo Desktop Computer}                => {iMac}      0.01860702  0.5069252 0.03670564 1.979202   183
## [3]  {Acer Desktop,
##       HP Laptop}                              => {iMac}      0.01596340  0.5114007 0.03121505 1.996675   157
## [4]  {Lenovo Desktop Computer,
##       ViewSonic Monitor}                      => {iMac}      0.01576004  0.5555556 0.02836807 2.169071   155
## [5]  {Dell Desktop,
##       ViewSonic Monitor}                      => {HP Laptop} 0.01525165  0.5747126 0.02653787 2.960869   150
## [6]  {Apple Magic Keyboard,
##       HP Laptop}                              => {iMac}      0.01474326  0.5105634 0.02887646 1.993406   145
## [7]  {Dell Desktop,
##       ViewSonic Monitor}                      => {iMac}      0.01474326  0.5555556 0.02653787 2.169071   145
## [8]  {HP Laptop,
##       Microsoft Office Home and Student 2016} => {iMac}      0.01291307  0.5521739 0.02338587 2.155868   127
## [9]  {CYBERPOWER Gamer Desktop,
##       ViewSonic Monitor}                      => {iMac}      0.01281139  0.5271967 0.02430097 2.058348   126
## [10] {Acer Desktop,
##       Lenovo Desktop Computer}                => {iMac}      0.01230300  0.5307018 0.02318251 2.072033   121
## [11] {CYBERPOWER Gamer Desktop,
##       ViewSonic Monitor}                      => {HP Laptop} 0.01220132  0.5020921 0.02430097 2.586734   120
## [12] {ASUS Monitor,
##       HP Laptop}                              => {iMac}      0.01179461  0.5829146 0.02023386 2.275889   116
## [13] {Apple Magic Keyboard,
##       Lenovo Desktop Computer}                => {iMac}      0.01138790  0.5161290 0.02206406 2.015137   112
## [14] {ASUS 2 Monitor,
##       HP Laptop}                              => {iMac}      0.01108287  0.5828877 0.01901373 2.275784   109
## [15] {ASUS 2 Monitor,
##       Lenovo Desktop Computer}                => {iMac}      0.01087951  0.5911602 0.01840366 2.308083   107
## [16] {Acer Aspire,
##       ViewSonic Monitor}                      => {HP Laptop} 0.01077783  0.6022727 0.01789527 3.102856   106
## [17] {HP Laptop,
##       HP Monitor}                             => {iMac}      0.01057448  0.5024155 0.02104728 1.961594   104
## [18] {Apple Magic Keyboard,
##       Dell Desktop}                           => {iMac}      0.01016777  0.5847953 0.01738688 2.283232   100
## [19] {Acer Desktop,
##       ViewSonic Monitor}                      => {iMac}      0.01006609  0.5439560 0.01850534 2.123782    99
```

```
# Further inspection by confidence
confidence11 <- data.frame(inspect(sort(Rules11, by = 'confidence', decreasing = TRUE))) #Acer Aspire and ViewSonic Moni
tor (#1) had greatest confidence at 60% (although 2nd rule 57%)
```

```
##      lhs                                        rhs            support confidence   coverage     lift count
## [1]  {Acer Aspire,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01077783  0.6022727 0.01789527 3.102856   106
## [2]  {ASUS 2 Monitor,
##       Lenovo Desktop Computer}               => {iMac}      0.01087951  0.5911602 0.01840366 2.308083   107
## [3]  {Apple Magic Keyboard,
##       Dell Desktop}                          => {iMac}      0.01016777  0.5847953 0.01738688 2.283232   100
## [4]  {ASUS Monitor,
##       HP Laptop}                             => {iMac}      0.01179461  0.5829146 0.02023386 2.275889   116
## [5]  {ASUS 2 Monitor,
##       HP Laptop}                             => {iMac}      0.01108287  0.5828877 0.01901373 2.275784   109
## [6]  {Dell Desktop,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01525165  0.5747126 0.02653787 2.960869   150
## [7]  {Dell Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01474326  0.5555556 0.02653787 2.169071   145
## [8]  {Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {iMac}      0.01576004  0.5555556 0.02836807 2.169071   155
## [9]  {HP Laptop,
##       Microsoft Office Home and Student 2016} => {iMac}      0.01291307  0.5521739 0.02338587 2.155868   127
## [10] {Acer Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01006609  0.5439560 0.01850534 2.123782    99
## [11] {Acer Desktop,
##       Lenovo Desktop Computer}               => {iMac}      0.01230300  0.5307018 0.02318251 2.072033   121
## [12] {CYBERPOWER Gamer Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01281139  0.5271967 0.02430097 2.058348   126
## [13] {Apple Magic Keyboard,
##       Lenovo Desktop Computer}               => {iMac}      0.01138790  0.5161290 0.02206406 2.015137   112
## [14] {Acer Desktop,
##       HP Laptop}                             => {iMac}      0.01596340  0.5114007 0.03121505 1.996675   157
## [15] {Apple Magic Keyboard,
##       HP Laptop}                             => {iMac}      0.01474326  0.5105634 0.02887646 1.993406   145
## [16] {Dell Desktop,
##       Lenovo Desktop Computer}               => {iMac}      0.01860702  0.5069252 0.03670564 1.979202   183
## [17] {HP Laptop,
##       HP Monitor}                            => {iMac}      0.01057448  0.5024155 0.02104728 1.961594   104
## [18] {CYBERPOWER Gamer Desktop,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01220132  0.5020921 0.02430097 2.586734   120
## [19] {HP Laptop,
##       Lenovo Desktop Computer}               => {iMac}      0.02308083  0.5000000 0.04616167 1.952164   227
```

```
# Further inspection by lift
lift11 <- data.frame(inspect(sort(Rules11, by = 'lift', decreasing = TRUE)))
```

```
##      lhs                                        rhs            support confidence   coverage     lift count
## [1]  {Acer Aspire,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01077783  0.6022727 0.01789527 3.102856   106
## [2]  {Dell Desktop,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01525165  0.5747126 0.02653787 2.960869   150
## [3]  {CYBERPOWER Gamer Desktop,
##       ViewSonic Monitor}                     => {HP Laptop} 0.01220132  0.5020921 0.02430097 2.586734   120
## [4]  {ASUS 2 Monitor,
##       Lenovo Desktop Computer}               => {iMac}      0.01087951  0.5911602 0.01840366 2.308083   107
## [5]  {Apple Magic Keyboard,
##       Dell Desktop}                          => {iMac}      0.01016777  0.5847953 0.01738688 2.283232   100
## [6]  {ASUS Monitor,
##       HP Laptop}                             => {iMac}      0.01179461  0.5829146 0.02023386 2.275889   116
## [7]  {ASUS 2 Monitor,
##       HP Laptop}                             => {iMac}      0.01108287  0.5828877 0.01901373 2.275784   109
## [8]  {Dell Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01474326  0.5555556 0.02653787 2.169071   145
## [9]  {Lenovo Desktop Computer,
##       ViewSonic Monitor}                     => {iMac}      0.01576004  0.5555556 0.02836807 2.169071   155
## [10] {HP Laptop,
##       Microsoft Office Home and Student 2016} => {iMac}      0.01291307  0.5521739 0.02338587 2.155868   127
## [11] {Acer Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01006609  0.5439560 0.01850534 2.123782    99
## [12] {Acer Desktop,
##       Lenovo Desktop Computer}               => {iMac}      0.01230300  0.5307018 0.02318251 2.072033   121
## [13] {CYBERPOWER Gamer Desktop,
##       ViewSonic Monitor}                     => {iMac}      0.01281139  0.5271967 0.02430097 2.058348   126
## [14] {Apple Magic Keyboard,
##       Lenovo Desktop Computer}               => {iMac}      0.01138790  0.5161290 0.02206406 2.015137   112
## [15] {Acer Desktop,
##       HP Laptop}                             => {iMac}      0.01596340  0.5114007 0.03121505 1.996675   157
## [16] {Apple Magic Keyboard,
##       HP Laptop}                             => {iMac}      0.01474326  0.5105634 0.02887646 1.993406   145
## [17] {Dell Desktop,
##       Lenovo Desktop Computer}               => {iMac}      0.01860702  0.5069252 0.03670564 1.979202   183
## [18] {HP Laptop,
##       HP Monitor}                            => {iMac}      0.01057448  0.5024155 0.02104728 1.961594   104
## [19] {HP Laptop,
##       Lenovo Desktop Computer}               => {iMac}      0.02308083  0.5000000 0.04616167 1.952164   227
```

```
head(quality(Rules11))
```

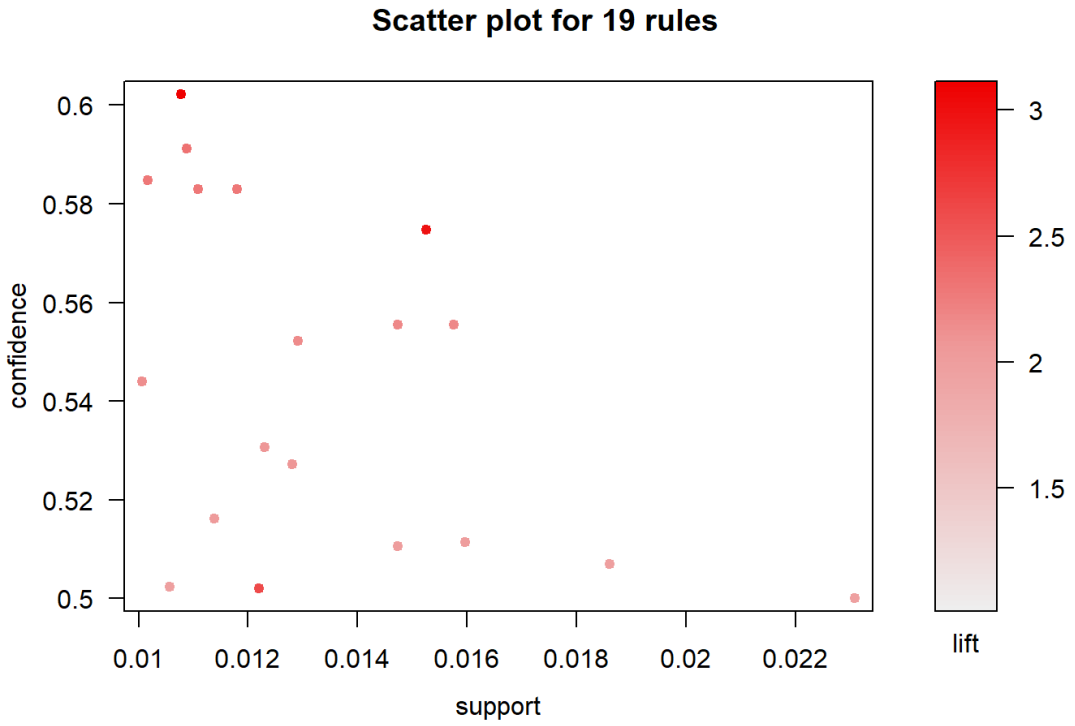
	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
1	0.01087951	0.5911602	0.01840366	2.308083	107
2	0.01108287	0.5828877	0.01901373	2.275784	109
3	0.01179461	0.5829146	0.02023386	2.275889	116
4	0.01291307	0.5521739	0.02338587	2.155868	127
5	0.01057448	0.5024155	0.02104728	1.961594	104
6	0.01016777	0.5847953	0.01738688	2.283232	100

6 rows

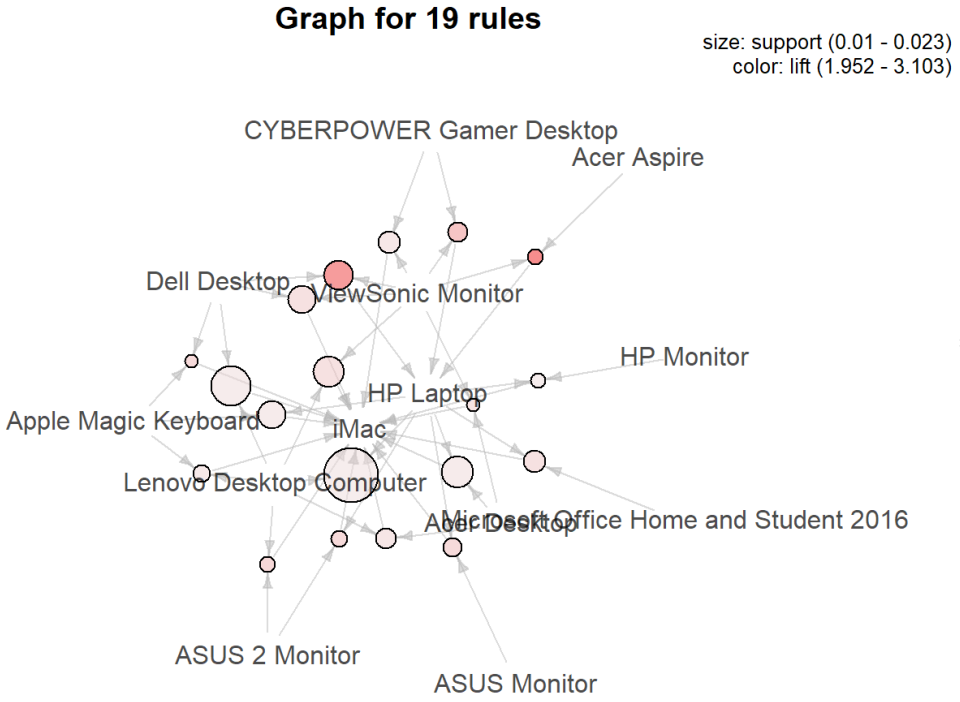
```
inspect(Rules11[is.redundant(Rules11)]) #No redundant rules
```

Plotting our best set of rules (11)

```
plot(Rules11)
```




```
plot(Rules11, method = 'graph')
```



Summary of our best set of rules (11)

```
summary(Rules11)
```

```
## set of 19 rules
##
## rule length distribution (lhs + rhs):sizes
##  3
## 19
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       3       3       3       3       3       3
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.01007   Min.   :0.5000   Min.   :0.01739   Min.   :1.952
##  1st Qu.:0.01098   1st Qu.:0.5110   1st Qu.:0.01962   1st Qu.:2.006
##  Median :0.01230   Median :0.5440   Median :0.02339   Median :2.156
##  Mean   :0.01343   Mean   :0.5439   Mean   :0.02495   Mean   :2.234
##  3rd Qu.:0.01500   3rd Qu.:0.5788   3rd Qu.:0.02745   3rd Qu.:2.280
##  Max.   :0.02308   Max.   :0.6023   Max.   :0.04616   Max.   :3.103
##      count
##  Min.   : 99.0
##  1st Qu.:108.0
##  Median :121.0
##  Mean   :132.1
##  3rd Qu.:147.5
##  Max.   :227.0
##
## mining info:
##   data ntransactions support confidence
##  trans          9835    0.01        0.5
```

Interesting discoveries within Electronidex's transactions (Top Product Relationships)

1. Customers who purchase a ViewSonic Monitor with either an Acer Aspire laptop, Dell Desktop, or a CYBERPOWER Gamer Desktop also purchase an HP Laptop.

- These itemset combinations would serve as an excellent recommendation of items frequently bought together to further boost sales of higher cost items.

2. Monitors (ViewSonic, ASUS 2, ASUS, HP Monitor) are a popular product purchased alongside desktop computers, as seen in 10 of 19 rules.

- ViewSonic and ASUS 2 are top and would be excellent recommendation items with any desktop computer brand.

3. iMac Desktops are often purchased with 1 or more other desktop computer brands (Lenovo, Dell, Acer), as seen in 1406 total transactions.

- It is unclear if customers are buying different brands and keeping all, or buying to test out different brands, only to eventually return less desired desktops.
- In this situation, it would be prudent to first investigate returns of desktops purchased alongside other desktop computer brands listed above. If customers are more often keeping all desktops, then these higher sale transactions could serve as excellent recommendation opportunities of associated products frequently bought together.

Recommendations for Blackwell, if they acquire Electronidex:

1. Initiate recommender systems for the following products listed within these item sets frequently bought together:

- Acer Aspire, ViewSonic Monitor, HP Laptop
- Dell Desktop, ViewSonic Monitor, HP Laptop
- CYBERPOWER Gamer Desktop, ViewSonic Monitor, HP Laptop
- ASUS Monitor, HP Laptop, iMac
- ASUS 2 Monitor, HP Laptop, iMac
- HP Laptop, Microsoft Office Home and Student 2016, iMac
- CYBERPOWER Gamer Desktop, ViewSonic Monitor, iMac
- HP Laptop, HP Monitor, iMac

2. Send promotional emails to customers who buy specific products within any of the above item sets a-h, informing them of products likely to be interesting to them.

3. Recommend ViewSonic and ASUS 2 Monitors as items customers also view each time a desktop computer is viewed or added to cart.

4. Investigate returns of desktops purchased alongside other desktop brands. If customers are more often keeping all desktops, then initiate recommender systems and email promotions as items customers frequently purchase together.

5. Discount or bundle the 20 lowest selling Electronidex products alongside other items recommended to customers in an effort to liquidate products not selling.