Homework 7

Kenigbolo Meya Stephen

March 23, 2016

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.

Use the data about transactions in a supermarket. Run FIM and Association rule generation algorithms to identify interesting itemsets and rules.

DATA file - Attach:supermarket.txt

1. Report which tools you decided to use, how you used them, what were the first results. Also report the running times for the tools chosen.

```
library(arules)

library(arulesViz)

supermarketDF <- read.csv("C:/Users/Kenigbolo PC/Desktop/Data Mining/supermar
ket.txt", header = F, sep = " ")


supermarket <- read.transactions("C:/Users/Kenigbolo PC/Desktop/Data Mining/s
upermarket.txt", format = "basket", sep=" ")

print(system.time(supermarket <- read.transactions("C:/Users/Kenigbolo PC/Des
ktop/Data Mining/supermarket.txt", format = "basket", sep=" ")))

##    user  system elapsed
##    3.91    0.00    3.91

rules <- apriori(supermarket, parameter = list(minlen=2, supp=0.003, conf=0.8
), control = list(verbose=F))

inspect(rules)

##    lhs                           rhs     support     confidence lift
## 1  {14438}                    => {13973} 0.003348830 1.0000000  12.793347
## 2  {7671}                     => {5330}  0.003151840 0.9876543   5.056200
## 3  {14381}                    => {5330}  0.003348830 0.9139785   4.679024
## 4  {5695}                     => {13973} 0.003821606 1.0000000  12.793347
## 5  {2740}                     => {3423}  0.003309432 1.0000000  16.343851
## 6  {10814}                    => {3423}  0.003191238 1.0000000  16.343851
## 7  {12382}                    => {5330}  0.003939800 1.0000000   5.119403
## 8  {9326}                     => {5330}  0.003545820 1.0000000   5.119403
```

```
## 9  {14083}                   => {5330}  0.004254984 1.0000000   5.119403
## 10 {5422}                    => {5330}  0.003585218 0.9285714   4.753731
## 11 {6174}                    => {5330}  0.004924750 1.0000000   5.119403
## 12 {3717}                    => {5330}  0.003664014 1.0000000   5.119403
## 13 {14744}                   => {5330}  0.004018596 1.0000000   5.119403
## 14 {12078}                   => {5330}  0.004294382 1.0000000   5.119403
## 15 {9630}                    => {13973} 0.005200536 1.0000000  12.793347
## 16 {13491}                   => {5330}  0.005003546 1.0000000   5.119403
## 17 {233}                     => {5330}  0.004688362 1.0000000   5.119403
## 18 {8282}                    => {5330}  0.003624616 0.9583333   4.906095
## 19 {5124}                    => {13973} 0.004688362 0.8095238  10.356519
## 20 {7466}                    => {5330}  0.005870302 1.0000000   5.119403
## 21 {2449}                    => {5330}  0.005239934 1.0000000   5.119403
## 22 {12456}                   => {5330}  0.007012844 1.0000000   5.119403
## 23 {14914}                   => {5330}  0.011110236 0.8924051   4.568581
## 24 {12562}                   => {5330}  0.016231975 0.8673684   4.440408
## 25 {5330,9630}               => {13973} 0.003151840 1.0000000  12.793347
## 26 {13491,14482}             => {5330}  0.003388228 1.0000000   5.119403
## 27 {13491,9108}              => {5330}  0.003348830 1.0000000   5.119403
## 28 {14482,14914}             => {5330}  0.003900402 1.0000000   5.119403
## 29 {14482,14914}             => {9108}  0.003348830 0.8585859   3.912500
## 30 {14754,14914}             => {5330}  0.003703412 0.9894737   5.065515
## 31 {14914,9108}              => {5330}  0.006303680 0.9142857   4.680597
## 32 {14482,6385}              => {14754} 0.005555118 0.8245614  12.638296
## 33 {14482,6385}              => {9108}  0.005633914 0.8362573   3.810751
## 34 {14482,14754}             => {9108}  0.006185486 0.8134715   3.706918
## 35 {12562,9108}              => {5330}  0.004294382 0.8861789   4.536707
## 36 {11995,9108}              => {5330}  0.007091640 0.8737864   4.473265
## 37 {14482,14914,5330}        => {9108}  0.003348830 0.8585859   3.912500
## 38 {14482,14914,9108}        => {5330}  0.003348830 1.0000000   5.119403
## 39 {14482,5330,6385}         => {14754} 0.003939800 0.8064516  12.360722
## 40 {14754,5330,6385}         => {14482} 0.003939800 0.8333333  54.939394
## 41 {14482,14754,6385}        => {9108}  0.004648964 0.8368794   3.813586
## 42 {14482,6385,9108}         => {14754} 0.004648964 0.8251748  12.647698
## 43 {14754,6385,9108}         => {14482} 0.004648964 0.8428571  55.567273
## 44 {14482,5330,6385}         => {9108}  0.004136790 0.8467742   3.858676
## 45 {14482,14754,5330}        => {9108}  0.004570168 0.8226950   3.748949
## 46 {14482,14754,5330,6385} => {9108}  0.003388228 0.8600000   3.918944
## 47 {14482,5330,6385,9108}  => {14754} 0.003388228 0.8190476  12.553784
## 48 {14754,5330,6385,9108}  => {14482} 0.003388228 0.9052632  59.681531
```

```
print(system.time(apriori(supermarket, parameter = list(minlen=2, supp=0.003,
conf=0.8), control = list(verbose=F))))
```

```
##    user  system elapsed
##    0.14    0.00    0.14
```

Tools Used => I used the Arules and ArulezViz library in R

```
library(arules)
library(arulesViz)
```

How Tools were used => First I loaded the data into R by reading it in via "read.transactions" => Then I tried to inspect the taransactions by calling the "inspect()" method on the supermarket transaction values but there were so many of them => Later used the inspect to check the rules => Continuation of how tools were used can be seen in the first results and execution times sub sections

```
supermarket <- read.transactions("C:/Users/Kenigbolo PC/Desktop/Data Mining/s
upermarket.txt", format = "basket", sep=" ")
```

First results

The first results obtained are the following;

=> First I ran the apriori algorithm provided by the arules package and I used a support of 0.002 and this gave me about 120 rules which was obviously a whole lot

=> I ran the apriori algorithm a second time and increased the support to 0.003 and this produced 48 rules, which is what I settled for as this task as this was quite reasonable for this first task in my opinion albeit 0.002 should be more suitable for the next task.

=> I proceeded to do a scatter plot for the rules that plotted confidence against support using lift as the gauge

=> I also explored plotting a Graph for 48 rules which in my opinion wasn't really clear enough which made me to proceed by

=> Plotting parallel coordinates plot for the 48 rules. At this point I realized the graphs weren't really interepretable so I adjusted my support a bit.

```
rules <- apriori(supermarket, parameter = list(minlen=2, supp=0.002, conf=0.8
), control = list(verbose=F))
inspect(rules)

##      lhs                            rhs     support     confidence lift
## 1    {15427}                     => {5330}  0.002324482 1.0000000   5.119403
## 2    {6614}                      => {7893}  0.002245686 1.0000000  13.742285
## 3    {8016}                      => {13973} 0.002088094 1.0000000  12.793347
## 4    {1839}                      => {5330}  0.002245686 1.0000000   5.119403
## 5    {10006}                     => {13973} 0.002679064 1.0000000  12.793347
## 6    {14105}                     => {3423}  0.002048696 1.0000000  16.343851
## 7    {3269}                      => {3423}  0.002521472 1.0000000  16.343851
## 8    {5145}                      => {5330}  0.002639666 1.0000000   5.119403
## 9    {9290}                      => {5330}  0.002048696 1.0000000   5.119403
## 10   {6917}                      => {13973} 0.002285084 1.0000000  12.793347
## 11   {15153}                     => {3723}  0.002285084 1.0000000  19.405199
## 12   {8078}                      => {5330}  0.002363880 1.0000000   5.119403
## 13   {13903}                     => {5330}  0.002639666 1.0000000   5.119403
## 14   {6676}                      => {3723}  0.002088094 1.0000000  19.405199
## 15   {6651}                      => {7893}  0.002836656 1.0000000  13.742285
## 16   {14474}                     => {3423}  0.002954850 1.0000000  16.343851
## 17   {14438}                     => {13973} 0.003348830 1.0000000  12.793347
```

```
## 18  {1508}                        => {5330}  0.002285084 1.0000000   5.119403
## 19  {3471}                        => {5330}  0.002954850 1.0000000   5.119403
## 20  {7671}                        => {5330}  0.003151840 0.9876543   5.056200
## 21  {14381}                       => {5330}  0.003348830 0.9139785   4.679024
## 22  {8971}                        => {3723}  0.002876054 1.0000000  19.405199
## 23  {5695}                        => {13973} 0.003821606 1.0000000  12.793347
## 24  {2740}                        => {3423}  0.003309432 1.0000000  16.343851
## 25  {10814}                       => {3423}  0.003191238 1.0000000  16.343851
## 26  {10797}                       => {5330}  0.002600268 1.0000000   5.119403
## 27  {14096}                       => {5330}  0.002363880 1.0000000   5.119403
## 28  {12382}                       => {5330}  0.003939800 1.0000000   5.119403
## 29  {9326}                        => {5330}  0.003545820 1.0000000   5.119403
## 30  {14083}                       => {5330}  0.004254984 1.0000000   5.119403
## 31  {5422}                        => {5330}  0.003585218 0.9285714   4.753731
## 32  {6174}                        => {5330}  0.004924750 1.0000000   5.119403
## 33  {3717}                        => {5330}  0.003664014 1.0000000   5.119403
## 34  {14744}                       => {5330}  0.004018596 1.0000000   5.119403
## 35  {12078}                       => {5330}  0.004294382 1.0000000   5.119403
## 36  {9630}                        => {13973} 0.005200536 1.0000000  12.793347
## 37  {13491}                       => {5330}  0.005003546 1.0000000   5.119403
## 38  {233}                         => {5330}  0.004688362 1.0000000   5.119403
## 39  {8282}                        => {5330}  0.003624616 0.9583333   4.906095
## 40  {5124}                        => {13973} 0.004688362 0.8095238  10.356519
## 41  {7466}                        => {5330}  0.005870302 1.0000000   5.119403
## 42  {2449}                        => {5330}  0.005239934 1.0000000   5.119403
## 43  {12456}                       => {5330}  0.007012844 1.0000000   5.119403
## 44  {14914}                       => {5330}  0.011110236 0.8924051   4.568581
## 45  {12562}                       => {5330}  0.016231975 0.8673684   4.440408
## 46  {5695,9108}                   => {13973} 0.002127492 1.0000000  12.793347
## 47  {5422,9108}                   => {5330}  0.002088094 0.9814815   5.024599
## 48  {13973,4435}                  => {5330}  0.002088094 0.8281250   4.239506
## 49  {14744,9108}                  => {5330}  0.002363880 1.0000000   5.119403
## 50  {5330,9630}                   => {13973} 0.003151840 1.0000000  12.793347
## 51  {9108,9630}                   => {13973} 0.002442676 1.0000000  12.793347
## 52  {13491,6385}                  => {14482} 0.002245686 0.8507463  56.087381
## 53  {13491,14754}                 => {14482} 0.002482074 0.8289474  54.650239
## 54  {13491,14482}                 => {5330}  0.003388228 1.0000000   5.119403
## 55  {13491,14482}                 => {9108}  0.002718462 0.8023256   3.656127
## 56  {13491,9108}                  => {14482} 0.002718462 0.8117647  53.517433
## 57  {13491,6385}                  => {14754} 0.002206288 0.8358209  12.810873
## 58  {13491,6385}                  => {5330}  0.002639666 1.0000000   5.119403
## 59  {13491,14754}                 => {5330}  0.002994248 1.0000000   5.119403
## 60  {13491,9108}                  => {5330}  0.003348830 1.0000000   5.119403
## 61  {2556,9108}                   => {5330}  0.002088094 0.8688525   4.448006
## 62  {11723,9108}                  => {5330}  0.002836656 0.8372093   4.286012
## 63  {7466,9108}                   => {5330}  0.002836656 1.0000000   5.119403
## 64  {2449,8233}                   => {5330}  0.002600268 1.0000000   5.119403
## 65  {11217,2449}                  => {5330}  0.002482074 1.0000000   5.119403
## 66  {12456,7595}                  => {5330}  0.002521472 1.0000000   5.119403
## 67  {12456,9108}                  => {5330}  0.002876054 1.0000000   5.119403
```

```
## 68  {14914,6385}             => {14482} 0.002206288 0.8358209  55.103392
## 69  {14482,14914}            => {5330}  0.003900402 1.0000000   5.119403
## 70  {14482,14914}            => {9108}  0.003348830 0.8585859   3.912500
## 71  {14914,6385}             => {5330}  0.002639666 1.0000000   5.119403
## 72  {14914,6385}             => {9108}  0.002245686 0.8507463   3.876776
## 73  {14754,14914}            => {5330}  0.003703412 0.9894737   5.065515
## 74  {13973,14914}            => {5330}  0.002954850 0.9868421   5.052042
## 75  {14914,9108}             => {5330}  0.006303680 0.9142857   4.680597
## 76  {14482,6385}             => {14754} 0.005555118 0.8245614  12.638296
## 77  {14482,6385}             => {9108}  0.005633914 0.8362573   3.810751
## 78  {14482,14754}            => {9108}  0.006185486 0.8134715   3.706918
## 79  {12562,14754}            => {5330}  0.002245686 0.8769231   4.489323
## 80  {12562,13973}            => {5330}  0.002679064 0.9315068   4.768759
## 81  {11217,12562}            => {5330}  0.002757860 0.8433735   4.317569
## 82  {12562,9108}             => {5330}  0.004294382 0.8861789   4.536707
## 83  {11026,15463}            => {7595}  0.002403278 0.8714286  14.619035
## 84  {11995,3423}             => {5330}  0.002048696 0.8253968   4.225539
## 85  {11995,9108}             => {5330}  0.007091640 0.8737864   4.473265
## 86  {13491,14482,6385}       => {5330}  0.002245686 1.0000000   5.119403
## 87  {13491,5330,6385}        => {14482} 0.002245686 0.8507463  56.087381
## 88  {13491,14482,14754}      => {5330}  0.002482074 1.0000000   5.119403
## 89  {13491,14754,5330}       => {14482} 0.002482074 0.8289474  54.650239
## 90  {13491,14482,5330}       => {9108}  0.002718462 0.8023256   3.656127
## 91  {13491,14482,9108}       => {5330}  0.002718462 1.0000000   5.119403
## 92  {13491,5330,9108}        => {14482} 0.002718462 0.8117647  53.517433
## 93  {13491,14754,6385}       => {5330}  0.002206288 1.0000000   5.119403
## 94  {13491,5330,6385}        => {14754} 0.002206288 0.8358209  12.810873
## 95  {13491,6385,9108}        => {5330}  0.002009298 1.0000000   5.119403
## 96  {13491,14754,9108}       => {5330}  0.002088094 1.0000000   5.119403
## 97  {14482,14914,6385}       => {5330}  0.002206288 1.0000000   5.119403
## 98  {14914,5330,6385}        => {14482} 0.002206288 0.8358209  55.103392
## 99  {14482,14754,14914}      => {5330}  0.002876054 1.0000000   5.119403
## 100 {14482,14754,14914}      => {9108}  0.002442676 0.8493151   3.870254
## 101 {14754,14914,9108}       => {14482} 0.002442676 0.8493151  55.993026
## 102 {14482,14914,5330}       => {9108}  0.003348830 0.8585859   3.912500
## 103 {14482,14914,9108}       => {5330}  0.003348830 1.0000000   5.119403
## 104 {14754,14914,6385}       => {5330}  0.002088094 1.0000000   5.119403
## 105 {14914,5330,6385}        => {9108}  0.002245686 0.8507463   3.876776
## 106 {14914,6385,9108}        => {5330}  0.002245686 1.0000000   5.119403
## 107 {14754,14914,9108}       => {5330}  0.002876054 1.0000000   5.119403
## 108 {14482,5330,6385}        => {14754} 0.003939800 0.8064516  12.360722
## 109 {14754,5330,6385}        => {14482} 0.003939800 0.8333333  54.939394
## 110 {14482,14754,6385}       => {9108}  0.004648964 0.8368794   3.813586
## 111 {14482,6385,9108}        => {14754} 0.004648964 0.8251748  12.647698
## 112 {14754,6385,9108}        => {14482} 0.004648964 0.8428571  55.567273
## 113 {14482,5330,6385}        => {9108}  0.004136790 0.8467742   3.858676
## 114 {14482,14754,5330}       => {9108}  0.004570168 0.8226950   3.748949
## 115 {14482,14754,14914,5330} => {9108}  0.002442676 0.8493151   3.870254
## 116 {14482,14754,14914,9108} => {5330}  0.002442676 1.0000000   5.119403
## 117 {14754,14914,5330,9108}  => {14482} 0.002442676 0.8493151  55.993026
```

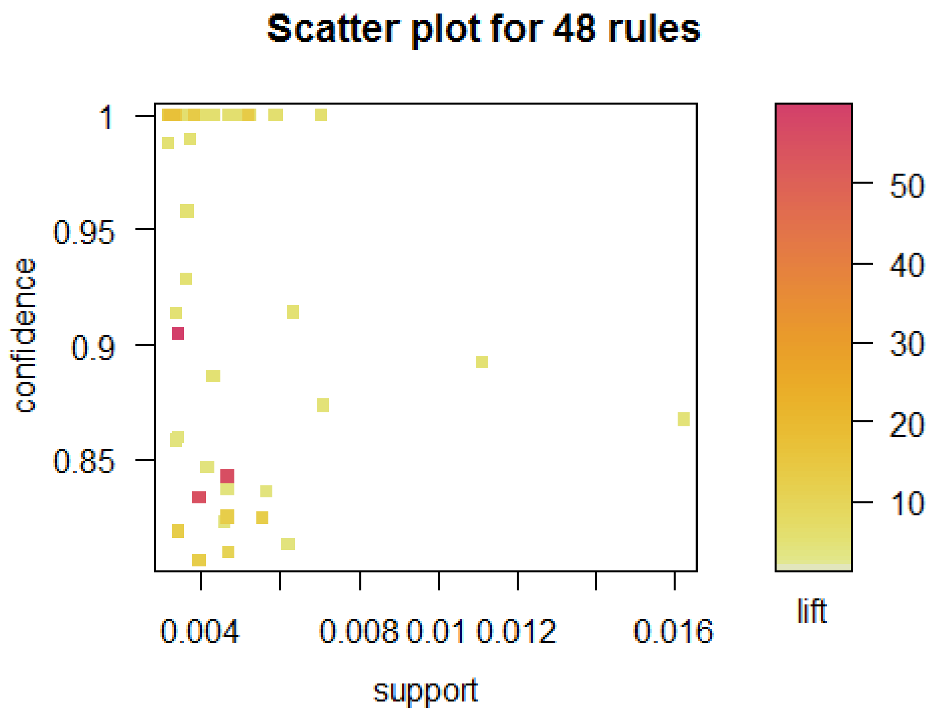
```
## 118 {14482,14754,5330,6385}  => {9108}  0.003388228 0.8600000   3.918944
## 119 {14482,5330,6385,9108}   => {14754} 0.003388228 0.8190476  12.553784
## 120 {14754,5330,6385,9108}   => {14482} 0.003388228 0.9052632  59.681531
```

```
rules <- apriori(supermarket, parameter = list(minlen=2, supp=0.003, conf=0.8
), control = list(verbose=F))
inspect(rules)
```

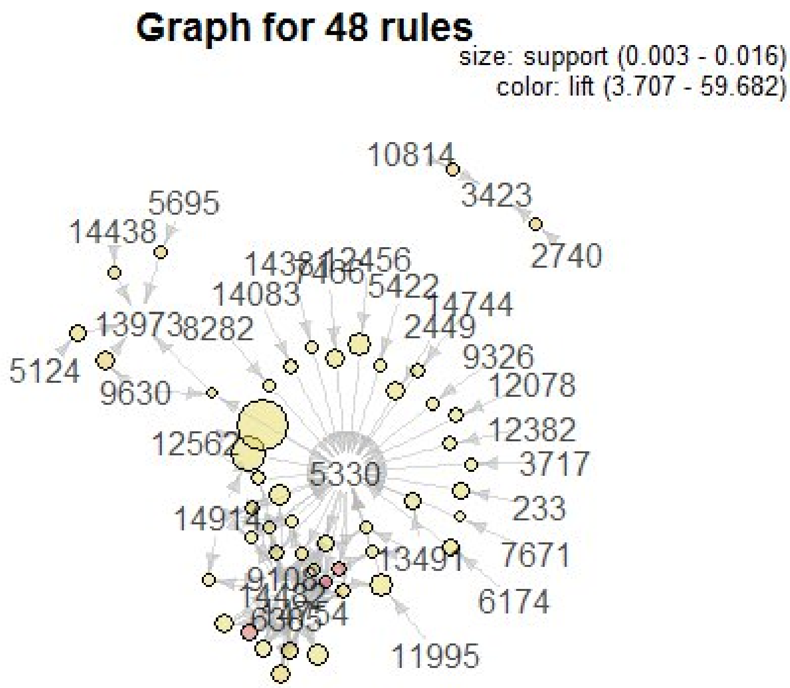
```
##    lhs                        rhs     support     confidence lift
## 1  {14438}                 => {13973} 0.003348830 1.0000000  12.793347
## 2  {7671}                  => {5330}  0.003151840 0.9876543   5.056200
## 3  {14381}                 => {5330}  0.003348830 0.9139785   4.679024
## 4  {5695}                  => {13973} 0.003821606 1.0000000  12.793347
## 5  {2740}                  => {3423}  0.003309432 1.0000000  16.343851
## 6  {10814}                 => {3423}  0.003191238 1.0000000  16.343851
## 7  {12382}                 => {5330}  0.003939800 1.0000000   5.119403
## 8  {9326}                  => {5330}  0.003545820 1.0000000   5.119403
## 9  {14083}                 => {5330}  0.004254984 1.0000000   5.119403
## 10 {5422}                  => {5330}  0.003585218 0.9285714   4.753731
## 11 {6174}                  => {5330}  0.004924750 1.0000000   5.119403
## 12 {3717}                  => {5330}  0.003664014 1.0000000   5.119403
## 13 {14744}                 => {5330}  0.004018596 1.0000000   5.119403
## 14 {12078}                 => {5330}  0.004294382 1.0000000   5.119403
## 15 {9630}                  => {13973} 0.005200536 1.0000000  12.793347
## 16 {13491}                 => {5330}  0.005003546 1.0000000   5.119403
## 17 {233}                   => {5330}  0.004688362 1.0000000   5.119403
## 18 {8282}                  => {5330}  0.003624616 0.9583333   4.906095
## 19 {5124}                  => {13973} 0.004688362 0.8095238  10.356519
## 20 {7466}                  => {5330}  0.005870302 1.0000000   5.119403
## 21 {2449}                  => {5330}  0.005239934 1.0000000   5.119403
## 22 {12456}                 => {5330}  0.007012844 1.0000000   5.119403
## 23 {14914}                 => {5330}  0.011110236 0.8924051   4.568581
## 24 {12562}                 => {5330}  0.016231975 0.8673684   4.440408
## 25 {5330,9630}             => {13973} 0.003151840 1.0000000  12.793347
## 26 {13491,14482}           => {5330}  0.003388228 1.0000000   5.119403
## 27 {13491,9108}            => {5330}  0.003348830 1.0000000   5.119403
## 28 {14482,14914}           => {5330}  0.003900402 1.0000000   5.119403
## 29 {14482,14914}           => {9108}  0.003348830 0.8585859   3.912500
## 30 {14754,14914}           => {5330}  0.003703412 0.9894737   5.065515
## 31 {14914,9108}            => {5330}  0.006303680 0.9142857   4.680597
## 32 {14482,6385}            => {14754} 0.005555118 0.8245614  12.638296
## 33 {14482,6385}            => {9108}  0.005633914 0.8362573   3.810751
## 34 {14482,14754}           => {9108}  0.006185486 0.8134715   3.706918
## 35 {12562,9108}            => {5330}  0.004294382 0.8861789   4.536707
## 36 {11995,9108}            => {5330}  0.007091640 0.8737864   4.473265
## 37 {14482,14914,5330}      => {9108}  0.003348830 0.8585859   3.912500
## 38 {14482,14914,9108}      => {5330}  0.003348830 1.0000000   5.119403
## 39 {14482,5330,6385}       => {14754} 0.003939800 0.8064516  12.360722
## 40 {14754,5330,6385}       => {14482} 0.003939800 0.8333333  54.939394
## 41 {14482,14754,6385}      => {9108}  0.004648964 0.8368794   3.813586
```

```
## 42 {14482,6385,9108}       => {14754} 0.004648964 0.8251748  12.647698
## 43 {14754,6385,9108}       => {14482} 0.004648964 0.8428571  55.567273
## 44 {14482,5330,6385}       => {9108}  0.004136790 0.8467742   3.858676
## 45 {14482,14754,5330}      => {9108}  0.004570168 0.8226950   3.748949
## 46 {14482,14754,5330,6385} => {9108}  0.003388228 0.8600000   3.918944
## 47 {14482,5330,6385,9108}  => {14754} 0.003388228 0.8190476  12.553784
## 48 {14754,5330,6385,9108}  => {14482} 0.003388228 0.9052632  59.681531
```

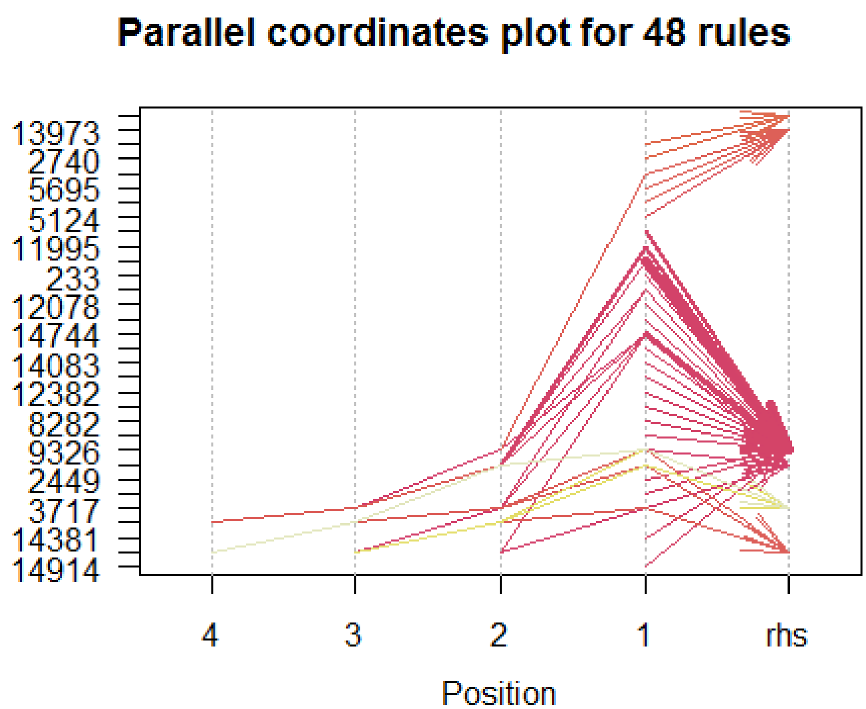
```
plot(rules)
```




```
plot(rules, method="graph", control=list(type="items"))
```




```
plot(rules, method="paracoord", control=list(reorder=TRUE))
```



=> I increased support to 0.004 in order to get a clearer picture and plots that are a bit understandable and this reduced my rules further to 24

```
rules <- apriori(supermarket, parameter = list(minlen=2, supp=0.004, conf=0.8
), control = list(verbose=F))
inspect(rules)
```

```
##    lhs                     rhs     support     confidence lift
## 1  {14083}              => {5330}  0.004254984 1.0000000   5.119403
## 2  {6174}               => {5330}  0.004924750 1.0000000   5.119403
## 3  {14744}              => {5330}  0.004018596 1.0000000   5.119403
## 4  {12078}              => {5330}  0.004294382 1.0000000   5.119403
## 5  {9630}               => {13973} 0.005200536 1.0000000  12.793347
## 6  {13491}              => {5330}  0.005003546 1.0000000   5.119403
## 7  {233}                => {5330}  0.004688362 1.0000000   5.119403
## 8  {5124}               => {13973} 0.004688362 0.8095238  10.356519
## 9  {7466}               => {5330}  0.005870302 1.0000000   5.119403
## 10 {2449}               => {5330}  0.005239934 1.0000000   5.119403
## 11 {12456}              => {5330}  0.007012844 1.0000000   5.119403
## 12 {14914}              => {5330}  0.011110236 0.8924051   4.568581
## 13 {12562}              => {5330}  0.016231975 0.8673684   4.440408
## 14 {14914,9108}         => {5330}  0.006303680 0.9142857   4.680597
## 15 {14482,6385}         => {14754} 0.005555118 0.8245614  12.638296
## 16 {14482,6385}         => {9108}  0.005633914 0.8362573   3.810751
## 17 {14482,14754}        => {9108}  0.006185486 0.8134715   3.706918
## 18 {12562,9108}         => {5330}  0.004294382 0.8861789   4.536707
## 19 {11995,9108}         => {5330}  0.007091640 0.8737864   4.473265
## 20 {14482,14754,6385} => {9108}  0.004648964 0.8368794   3.813586
## 21 {14482,6385,9108}  => {14754} 0.004648964 0.8251748  12.647698
## 22 {14754,6385,9108}  => {14482} 0.004648964 0.8428571  55.567273
## 23 {14482,5330,6385}  => {9108}  0.004136790 0.8467742   3.858676
## 24 {14482,14754,5330} => {9108}  0.004570168 0.8226950   3.748949
```

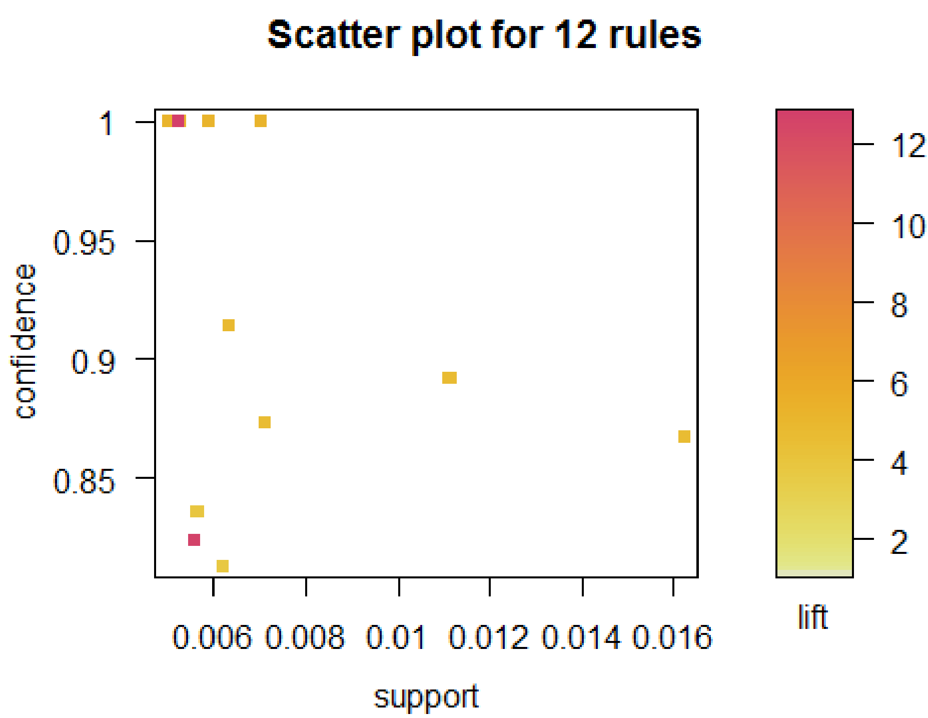
=> I further increased support to 0.005 in order to to see what the rule distribution will be like and I got 12 rules. => At this point it made sense to make plots again which I did by making a scatter plot and I noticed that majority of the rules had thier support between 0.006 and 0.008 and lift was high when support was <= 0.006 => I plotted the graph for 12 rules and the distribution gave me a clear view of the most frequent items => The parallel coordiates plot gave an interesting insight for 14482 as the arrow had a dissimilar direction wen compared to the other items

```
rules <- apriori(supermarket, parameter = list(minlen=2, supp=0.005, conf=0.8
), control = list(verbose=F))
inspect(rules)
```

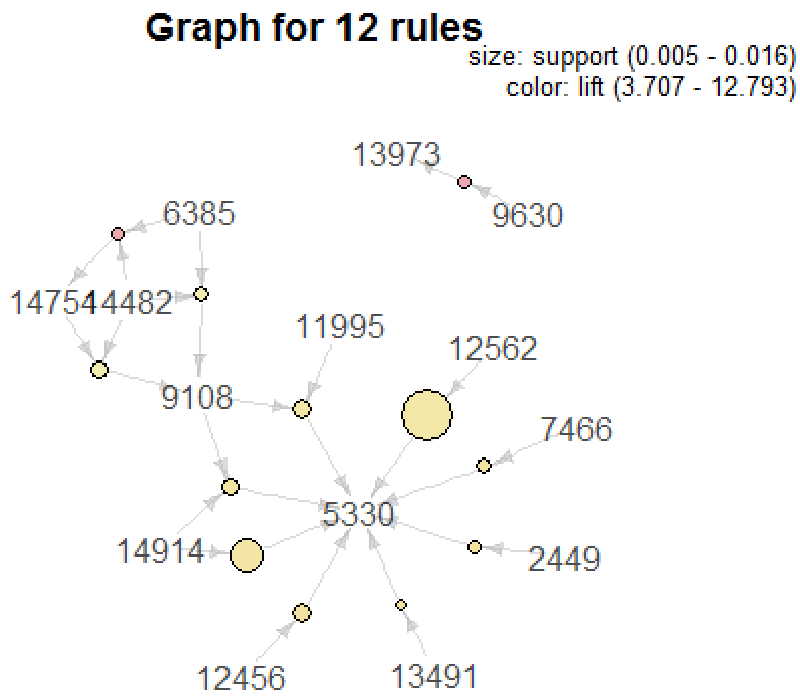
```
##    lhs               rhs     support     confidence lift
## 1  {9630}         => {13973} 0.005200536 1.0000000  12.793347
## 2  {13491}        => {5330}  0.005003546 1.0000000   5.119403
## 3  {7466}         => {5330}  0.005870302 1.0000000   5.119403
## 4  {2449}         => {5330}  0.005239934 1.0000000   5.119403
```

```
## 5  {12456}        => {5330}  0.007012844 1.0000000   5.119403
## 6  {14914}        => {5330}  0.011110236 0.8924051   4.568581
## 7  {12562}        => {5330}  0.016231975 0.8673684   4.440408
## 8  {14914,9108}   => {5330}  0.006303680 0.9142857   4.680597
## 9  {14482,6385}   => {14754} 0.005555118 0.8245614  12.638296
## 10 {14482,6385}   => {9108}  0.005633914 0.8362573   3.810751
## 11 {14482,14754} => {9108}  0.006185486 0.8134715   3.706918
## 12 {11995,9108}   => {5330}  0.007091640 0.8737864   4.473265
```

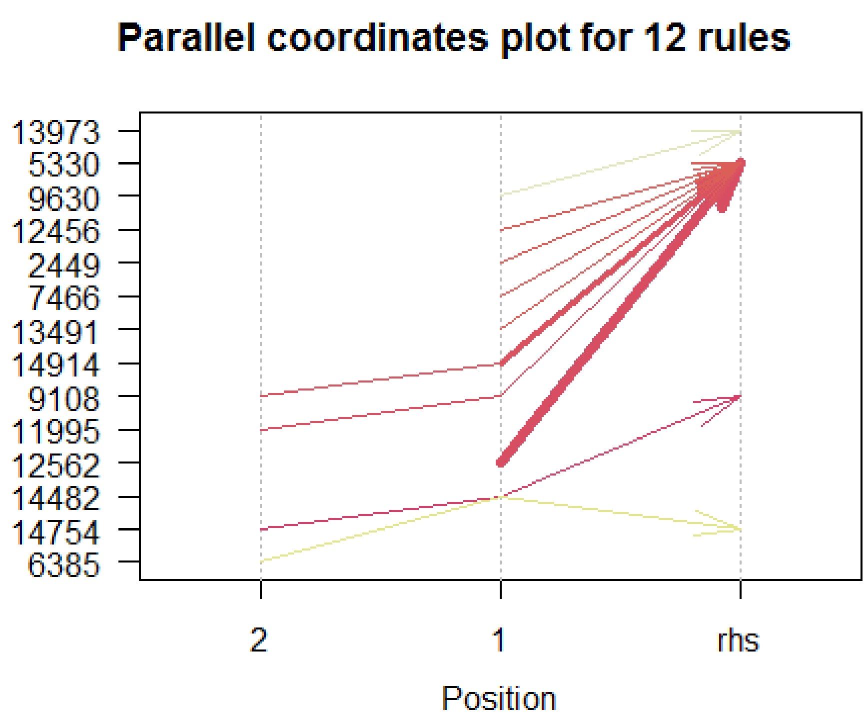
```
plot(rules)
```




```
plot(rules, method="graph", control=list(type="items"))
```

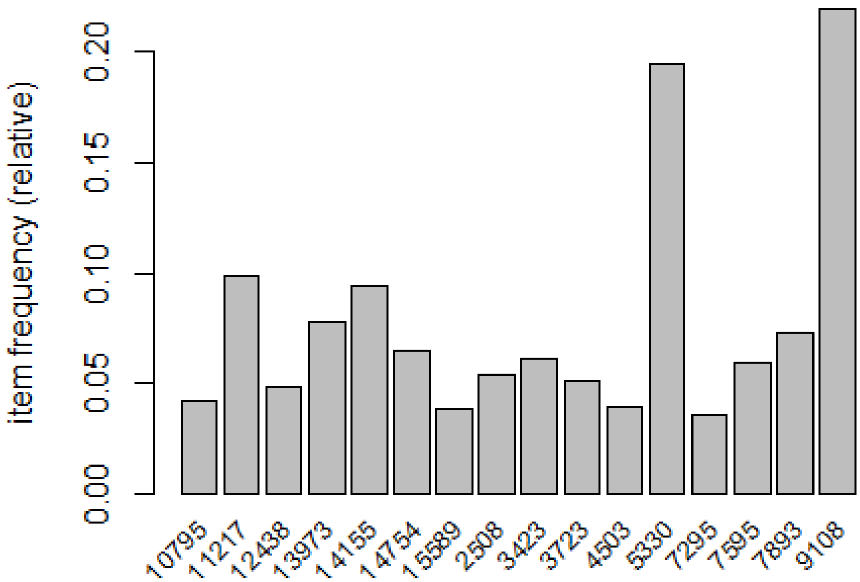



```
plot(rules, method="paracoord", control=list(reorder=TRUE))
```



=> Finally I decided to do a frequency plot for all items with a support of at least 0.035 in the data in order to enable me to compare about 15 items and identify which of those items had quite high frequency

```
itemFrequencyPlot(supermarket, support = 0.035, cex.names=0.8)
```



Running time for chosen tool => The running time of reading in the data itself is as follows user system elapsed 4.86 0.00 4.86

```
print(system.time(supermarket <- read.transactions("C:/Users/Kenigbolo PC/Desktop/Data Mining/supermarket.txt", format = "basket", sep=" ")))

##    user  system elapsed 
##    4.37    0.02    4.44
```

=> The running time for the rules is as follows user system elapsed 0.16 0.00 0.16

```
print(system.time(apriori(supermarket, parameter = list(minlen=2, supp=0.003, conf=0.8), control = list(verbose=F))))

##    user  system elapsed 
##    0.19    0.00    0.19
```

2. Report overall 5 different high-support, high-confidence, high-lift rules; provide the respective contingency tables and scores.

High Support - Top 5

```
## Mine frequent rules with top support
top.support <- sort(rules, decreasing = TRUE, na.last = NA, by = "support")

## Display the 10 rules with the highest support.
inspect(head(top.support, 5))
```

```
##    lhs             rhs    support     confidence lift
## 7  {12562}      => {5330} 0.016231975 0.8673684  4.440408
## 6  {14914}      => {5330} 0.011110236 0.8924051  4.568581
## 12 {11995,9108} => {5330} 0.007091640 0.8737864  4.473265
## 5  {12456}      => {5330} 0.007012844 1.0000000  5.119403
## 8  {14914,9108} => {5330} 0.006303680 0.9142857  4.680597
```

High Confidence - Top 10

```
## Mine frequent rules with top confidence.
top.confidence <- sort(rules, decreasing = TRUE, na.last = NA, by = "confiden
ce")

## Display the 10 itemsets with the highest confidence.
inspect(head(top.confidence, 5))
```

```
##   lhs        rhs     support     confidence lift
## 1 {9630}  => {13973} 0.005200536 1          12.793347
## 2 {13491} => {5330}  0.005003546 1           5.119403
## 3 {7466}  => {5330}  0.005870302 1           5.119403
## 4 {2449}  => {5330}  0.005239934 1           5.119403
## 5 {12456} => {5330}  0.007012844 1           5.119403
```

High Lift - Top 10

```
## Mine frequent rules with top lift.
top.lift <- sort(rules, decreasing = TRUE, na.last = NA, by = "lift")

## Display the 10 itemsets with the highest lift.
inspect(head(top.lift, 5))
```

```
##   lhs              rhs     support     confidence lift
## 1 {9630}       => {13973} 0.005200536 1.0000000  12.793347
## 9 {14482,6385} => {14754} 0.005555118 0.8245614  12.638296
## 2 {13491}      => {5330}  0.005003546 1.0000000   5.119403
## 3 {7466}       => {5330}  0.005870302 1.0000000   5.119403
## 4 {2449}       => {5330}  0.005239934 1.0000000   5.119403
```

The respective contingency tables

```
rules <- apriori(supermarket, parameter = list(minlen=2, supp=0.005, conf=0.8
), control = list(verbose=F))
cont_table <- inspect(rules)
```

```
##    lhs              rhs     support     confidence lift
## 1  {9630}        => {13973} 0.005200536 1.0000000  12.793347
```

```
## 2  {13491}       => {5330}  0.005003546 1.0000000   5.119403
## 3  {7466}        => {5330}  0.005870302 1.0000000   5.119403
## 4  {2449}        => {5330}  0.005239934 1.0000000   5.119403
## 5  {12456}       => {5330}  0.007012844 1.0000000   5.119403
## 6  {14914}       => {5330}  0.011110236 0.8924051   4.568581
## 7  {12562}       => {5330}  0.016231975 0.8673684   4.440408
## 8  {14914,9108}  => {5330}  0.006303680 0.9142857   4.680597
## 9  {14482,6385}  => {14754} 0.005555118 0.8245614  12.638296
## 10 {14482,6385}  => {9108}  0.005633914 0.8362573   3.810751
## 11 {14482,14754} => {9108}  0.006185486 0.8134715   3.706918
## 12 {11995,9108}  => {5330}  0.007091640 0.8737864   4.473265
```

```
#Outline Rules for contingency table
print(cont_table)
```

```
##               lhs        rhs     support confidence      lift
## 1          {9630} => {13973} 0.005200536  1.0000000 12.793347
## 2         {13491} =>  {5330} 0.005003546  1.0000000  5.119403
## 3          {7466} =>  {5330} 0.005870302  1.0000000  5.119403
## 4          {2449} =>  {5330} 0.005239934  1.0000000  5.119403
## 5         {12456} =>  {5330} 0.007012844  1.0000000  5.119403
## 6         {14914} =>  {5330} 0.011110236  0.8924051  4.568581
## 7         {12562} =>  {5330} 0.016231975  0.8673684  4.440408
## 8    {14914,9108} =>  {5330} 0.006303680  0.9142857  4.680597
## 9    {14482,6385} => {14754} 0.005555118  0.8245614 12.638296
## 10   {14482,6385} =>  {9108} 0.005633914  0.8362573  3.810751
## 11 {14482,14754} =>  {9108} 0.006185486  0.8134715  3.706918
## 12  {11995,9108} =>  {5330} 0.007091640  0.8737864  4.473265
```

```
library(dplyr)
```

Contingency Tables

From filtering, the data the following values will be sieved out F11(intersect of A and B in the DF), F1+(frequency of item A in the DF), F+1(Frequency of item B in the DF) and TOTAL (total number of transactions i.e. observables in the DF). The rest values will to be calculated will be done as stated below

F10 = F1+ - F11

F01 = F+1 - F11

F0+ = TOTAL - F1+

F+0 = TOTAL - F+1

F00 = F+0 - F10

RULE 9630 => 13973

```
#Filter the Rules Dataframe for where the item 9630 is present
subset9630 <- filter(supermarketDF, V1 == "9630" | V2 == "9630" | V3 == "9630
```

```
" |V4 == "9630" |V5 == "9630" |V6 == "9630" |V7 == "9630" |V8 == "9630" |V9 =
= "9630" |V10 == "9630" |V11 == "9630" )

#Total number of observables where
nrow(subset9630)

## [1] 132

#Filter the rules dataframe for where the  item 13973 is present
subset13973 <- filter(supermarketDF, V1 == "13973" | V2 == "13973" | V3 == "1
3973" |V4 == "13973" |V5 == "13973" |V6 == "13973" |V7 == "13973" |V8 == "139
73" |V9 == "13973" |V10 == "13973" |V11 == "13973" )

#Total number of observables where
nrow(subset13973)

## [1] 1984

#filter for where both 9630 and 13973 exists
subset9630_13973 <- filter(subset9630, V1 == "13973" | V2 == "13973" | V3 ==
"13973" |V4 == "13973" |V5 == "13973" |V6 == "13973" |V7 == "13973" |V8 == "1
3973" |V9 == "13973" |V10 == "13973" |V11 == "13973" )

#Total number of observables where both 9630 and 13973
nrow(subset9630_13973)

## [1] 95

#Total number in Data Frame
nrow(supermarket)

## [1] 25382

#Initialize the contingency table
contigencyTable <- matrix(c(95,37,132,1853,30575,32428,1948,30612,32560),ncol
=3,byrow=TRUE)
colnames(contigencyTable) <- c("13973","NOT 13973","TOTAL")
rownames(contigencyTable) <- c("9630","NOT 9630","TOTAL")

print(contigencyTable)

##          13973 NOT 13973 TOTAL
## 9630        95        37   132
## NOT 9630  1853     30575 32428
## TOTAL     1948     30612 32560
```

The contigency table for the rules is generatted by getting all the transactions where 9630 is present which is 132, where 13973 is present which is 1948 and where both 9630 and 13973 are present which is 95. The total number of the observables for the dataframe of the the supermarket.txt is 32560 observables.

F11 = 95, F1+ = 132, F+1 = 1948, TOTAL = 32560

RULE 13491 => 5330

```
#Filter the Rules Dataframe for where the item 13491 is present
subset13491 <- filter(supermarketDF, V1 == "13491" | V2 == "13491" | V3 == "1
3491" |V4 == "13491" |V5 == "13491" |V6 == "13491" |V7 == "13491" |V8 == "134
91" |V9 == "13491" |V10 == "13491" |V11 == "13491" )

#Total number of observables where
nrow(subset13491)

## [1] 127

#Filter the Rules Dataframe for where the item 5330 is present
subset5330 <- filter(supermarketDF, V1 == "5330" | V2 == "5330" | V3 == "5330
" |V4 == "5330" |V5 == "5330" |V6 == "5330" |V7 == "5330" |V8 == "5330" |V9 =
= "5330" |V10 == "5330" |V11 == "5330" )

#Total number of observables where
nrow(subset5330)

## [1] 4958

#Filter the Rules Dataframe for where the item 13491 and 5330 are present
subset13491_5330 <- filter(subset5330, V1 == "13491" | V2 == "13491" | V3 ==
"13491" |V4 == "13491" |V5 == "13491" |V6 == "13491" |V7 == "13491" |V8 == "1
3491" |V9 == "13491" |V10 == "13491" |V11 == "13491" )

#Total number of observables where 13491 and 5330 are present
nrow(subset13491_5330)

## [1] 97

#Initialize the contingency table
contigencyTable <- matrix(c(97,30,127,4861,27572,32433,4958,27602,32560),ncol
=3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("13491","NOT 13491","TOTAL")

print(contigencyTable)

##           5330 NOT 5330 TOTAL
## 13491       97       30   127
## NOT 13491 4861    27572 32433
## TOTAL     4958    27602 32560
```

F11 = 97, F1+ = 127, F+1 = 4958, TOTAL = 32560

RULE 7466 => 5330

```
#Filter the Rules Dataframe for where the item 7466 is present
subset7466 <- filter(supermarketDF, V1 == "7466" | V2 == "7466" | V3 == "7466
" |V4 == "7466" |V5 == "7466" |V6 == "7466" |V7 == "7466" |V8 == "7466" |V9 =
= "7466" |V10 == "7466" |V11 == "7466" )

#Total number of observables where
nrow(subset7466)

## [1] 149

#Filter the Rules Dataframe for where the item 7466 and 5330 are present
subset7466_5330 <- filter(subset5330, V1 == "7466" | V2 == "7466" | V3 == "74
66" |V4 == "7466" |V5 == "7466" |V6 == "7466" |V7 == "7466" |V8 == "7466" |V9
== "7466" |V10 == "7466" |V11 == "7466" )

#Total number of observables where
nrow(subset7466_5330)

## [1] 127

#Initialize the contingency table
contigencyTable <- matrix(c(127,22,149,4831,27580,32411,4958,27602,32560),nco
l=3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("7466","NOT 7466","TOTAL")

print(contigencyTable)

##          5330 NOT 5330 TOTAL
## 7466      127       22   149
## NOT 7466 4831    27580 32411
## TOTAL    4958    27602 32560
```

F11 = 127, F1+ = 149, F+1 = 4958, TOTAL = 32560

RULE 2449 => 5330

```
#Filter the Rules Dataframe for where the item 2449 is present
subset2449 <- filter(supermarketDF, V1 == "2449" | V2 == "2449" | V3 == "2449
" |V4 == "2449" |V5 == "2449" |V6 == "2449" |V7 == "2449" |V8 == "2449" |V9 =
= "2449" |V10 == "2449" |V11 == "2449" )

#Total number of observables where
nrow(subset2449)

## [1] 133
```

```r
#Filter the Rules Dataframe for where the item 2449 and 5330 are present
subset2449_5330 <- filter(subset5330, V1 == "2449" | V2 == "2449" | V3 == "24
49" |V4 == "2449" |V5 == "2449" |V6 == "2449" |V7 == "2449" |V8 == "2449" |V9
== "2449" |V10 == "2449" |V11 == "2449" )

#Total number of observables where
nrow(subset2449_5330)

## [1] 109

#Initialize the contingency table
contigencyTable <- matrix(c(109,24,133,4849,27578,32427,4958,27602,32560),nco
l=3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("2449","NOT 2449","TOTAL")

print(contigencyTable)

##          5330 NOT 5330 TOTAL
## 2449      109       24   133
## NOT 2449 4849    27578 32427
## TOTAL    4958    27602 32560
```

F11 = 109, F1+ = 133, F+1 = 4958, TOTAL = 32560

RULE 12456 => 5330

```r
#Filter the Rules Dataframe for where the item 12456 is present
subset12456 <- filter(supermarketDF, V1 == "12456" | V2 == "12456" | V3 == "1
2456" |V4 == "12456" |V5 == "12456" |V6 == "12456" |V7 == "12456" |V8 == "124
56" |V9 == "12456" |V10 == "12456" |V11 == "12456" )

#Total number of observables where
nrow(subset12456)

## [1] 178

#Filter the Rules Dataframe for where the item 12456 and 5330 are present
subset12456_5330 <- filter(subset5330, V1 == "12456" | V2 == "12456" | V3 ==
"12456" |V4 == "12456" |V5 == "12456" |V6 == "12456" |V7 == "12456" |V8 == "1
2456" |V9 == "12456" |V10 == "12456" |V11 == "12456" )

#Total number of observables where
nrow(subset12456_5330)

## [1] 86

#Initialize the contingency table
contigencyTable <- matrix(c(86,92,178,4872,27510,32382,4958,27602,32560),ncol
=3,byrow=TRUE)
```

```r
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("12456","NOT 12456","TOTAL")

print(contigencyTable)
```

```
##           5330 NOT 5330 TOTAL
## 12456       86       92   178
## NOT 12456 4872    27510 32382
## TOTAL     4958    27602 32560
```

F11 = 86 F1+ = 178, F+1 = 4958, TOTAL = 32560

RULE 14914 => 5330

```r
#Filter the Rules Dataframe for where the item 12456 is present
subset14914 <- filter(supermarketDF, V1 == "14914" | V2 == "14914" | V3 == "14914" |V4 == "14914" |V5 == "14914" |V6 == "14914" |V7 == "14914" |V8 == "14914" |V9 == "14914" |V10 == "14914" |V11 == "14914" )

#Total number of observables where
nrow(subset14914)
```

```
## [1] 316
```

```r
#Filter the Rules Dataframe for where the item 12456 and 5330 are present
subset14914_5330 <- filter(subset5330, V1 == "14914" | V2 == "14914" | V3 ==
"14914" |V4 == "14914" |V5 == "14914" |V6 == "14914" |V7 == "14914" |V8 == "14914" |V9 == "14914" |V10 == "14914" |V11 == "14914" )

#Total number of observables where
nrow(subset14914_5330)
```

```
## [1] 137
```

```r
#Initialize the contingency table
contigencyTable <- matrix(c(137,179,316,4821,27423,32244,4958,27602,32560),ncol=3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("14914","NOT 14914","TOTAL")

print(contigencyTable)
```

```
##           5330 NOT 5330 TOTAL
## 14914      137      179   316
## NOT 14914 4821    27423 32244
## TOTAL     4958    27602 32560
```

F11 = 137 F1+ = 316, F+1 = 4958, TOTAL = 32560

RULE 12562 => 5330

```
#Filter the Rules Dataframe for where the item 12456 is present
subset12562 <- filter(supermarketDF, V1 == "12562" | V2 == "12562" | V3 == "1
2562" |V4 == "12562" |V5 == "12562" |V6 == "12562" |V7 == "12562" |V8 == "125
62" |V9 == "12562" |V10 == "12562" |V11 == "12562" )

#Total number of observables where
nrow(subset12562)

## [1] 475

#Filter the Rules Dataframe for where the item 12456 and 5330 are present
subset12562_5330 <- filter(subset5330, V1 == "12562" | V2 == "12562" | V3 ==
"12562" |V4 == "12562" |V5 == "12562" |V6 == "12562" |V7 == "12562" |V8 == "1
2562" |V9 == "12562" |V10 == "12562" |V11 == "12562" )

#Total number of observables where
nrow(subset12562_5330)

## [1] 308

#Initialize the contingency table
contigencyTable <- matrix(c(308,167,475,4650,27435,32085,4958,27602,32560),nc
ol=3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("12562","NOT 12562","TOTAL")

print(contigencyTable)

##           5330 NOT 5330 TOTAL
## 12562      308      167   475
## NOT 12562 4650    27435 32085
## TOTAL     4958    27602 32560
```

F11 = 308 F1+ = 475, F+1 = 4958, TOTAL = 32560

RULE {14914,9108} => 5330

```
#Filter the Rules Dataframe for where the item 14914 is present
subset14914 <- filter(supermarketDF, V1 == "14914" | V2 == "14914" | V3 == "1
4914" |V4 == "14914" |V5 == "14914" |V6 == "14914" |V7 == "14914" |V8 == "149
14" |V9 == "14914" |V10 == "14914" |V11 == "14914" )

#Total number of observables where subset14914
nrow(subset14914)

## [1] 316
```

```
#Filter the Rules Dataframe for where the item 14914 and 9108 are present
subset9108_14914 <- filter(subset14914, V1 == "9108" | V2 == "9108" | V3 == "
9108" |V4 == "9108" |V5 == "9108" |V6 == "9108" |V7 == "9108" |V8 == "9108" |
V9 == "9108" |V10 == "9108" |V11 == "9108" )

#Total number of observables where subset9108_14914
nrow(subset9108_14914)

## [1] 85

subset9108_14914_5330 <- filter(subset9108_14914, V1 == "5330" | V2 == "5330"
| V3 == "5330" |V4 == "5330" |V5 == "5330" |V6 == "5330" |V7 == "5330" |V8 ==
"5330" |V9 == "5330" |V10 == "5330" |V11 == "5330"  )

nrow(subset9108_14914_5330)

## [1] 70

#Initialize the contingency table
contigencyTable <- matrix(c(70,15,85,4888,27587,32475,4958,27602,32560),ncol=
3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("{9108_14914}","NOT {9108_14914}","TOTAL")

print(contigencyTable)

##                  5330 NOT 5330 TOTAL
## {9108_14914}       70       15    85
## NOT {9108_14914} 4888    27587 32475
## TOTAL            4958    27602 32560
```

F11 = 70 F1+ = 85, F+1 = 4958, TOTAL = 32560

RULE {14482,6385} => 14754

```
#Filter the Rules Dataframe for where the item 14482 is present
subset14482 <- filter(supermarketDF, V1 == "14482" | V2 == "14482" | V3 == "1
4482" |V4 == "14482" |V5 == "14482" |V6 == "14482" |V7 == "14482" |V8 == "144
82" |V9 == "14482" |V10 == "14482" |V11 == "14482" )

#Total number of observables where subset14914
nrow(subset14482)

## [1] 385

#Filter the Rules Dataframe for where the item 14482 and 6385 are present
subset6385_14482 <- filter(subset14482, V1 == "6385" | V2 == "6385" | V3 == "
6385" |V4 == "6385" |V5 == "6385" |V6 == "6385" |V7 == "6385" |V8 == "6385" |
V9 == "6385" |V10 == "6385" |V11 == "6385" )
```

```r
#Total number of observables where subset6385_14482
nrow(subset6385_14482)
```

```
## [1] 121
```

```r
subset6385_14482_14754 <- filter(subset6385_14482, V1 == "14754" | V2 == "14754" | V3 == "14754" |V4 == "14754" |V5 == "14754" |V6 == "14754" |V7 == "14754" |V8 == "14754" |V9 == "14754" |V10 == "14754" |V11 == "14754"  )

#Total number of observables where subset6385_14482_14754
nrow(subset6385_14482_14754)
```

```
## [1] 83
```

```r
#Filter the Rules Dataframe for where the item 14482 is present
subset14754 <- filter(supermarketDF, V1 == "14754" | V2 == "14754" | V3 == "14754" |V4 == "14754" |V5 == "14754" |V6 == "14754" |V7 == "14754" |V8 == "14754" |V9 == "14754" |V10 == "14754" |V11 == "14754"  )

#Total number of observables where subset14754
nrow(subset14754)
```

```
## [1] 1656
```

```r
#Initialize the contingency table
contigencyTable <- matrix(c(83,38,121,1573,30866,32439,1656,30904,32560),ncol=3,byrow=TRUE)
colnames(contigencyTable) <- c("14754","NOT 14754","TOTAL")
rownames(contigencyTable) <- c("{14482,6385}","NOT {14482,6385}","TOTAL")

print(contigencyTable)
```

```
##                  14754 NOT 14754 TOTAL
## {14482,6385}        83        38   121
## NOT {14482,6385}  1573     30866 32439
## TOTAL             1656     30904 32560
```

F11 = 83 F1+ = 121, F+1 = 1656, TOTAL = 32560

RULE {14482,6385} => 9108

```r
#Filter the Rules Dataframe for where the item 9108 is present
subset9108 <- filter(supermarketDF, V1 == "9108" | V2 == "9108" | V3 == "9108" |V4 == "9108" |V5 == "9108" |V6 == "9108" |V7 == "9108" |V8 == "9108" |V9 == "9108" |V10 == "9108" |V11 == "9108"  )

nrow(subset9108)
```

```
## [1] 5570

subset6385_14482_9108 <- filter(subset6385_14482,V1 == "9108" | V2 == "9108"
| V3 == "9108" |V4 == "9108" |V5 == "9108" |V6 == "9108" |V7 == "9108" |V8 ==
"9108" |V9 == "9108" |V10 == "9108" |V11 == "9108"  )

#Total number of observables where subset6385_14482_14754
nrow(subset6385_14482_9108)

## [1] 103

#Initialize the contingency table
contigencyTable <- matrix(c(103,18,121,5467,26972,32439,5570,26990,32560),nco
l=3,byrow=TRUE)
colnames(contigencyTable) <- c("9108","NOT 9108","TOTAL")
rownames(contigencyTable) <- c("{14482,6385}","NOT {14482,6385}","TOTAL")

print(contigencyTable)

##                  9108 NOT 9108 TOTAL
## {14482,6385}      103       18   121
## NOT {14482,6385} 5467    26972 32439
## TOTAL            5570    26990 32560
```

F11 = 103 F1+ = 121, F+1 = 5570, TOTAL = 32560

3. Discuss whether some other scores studied last week or in the lecture slides would help identify "more interesting" and different rules?

I believe the Odds ratios (f11.f00)/(f10.f01) will help identify more interesting and different rules because Odds ratios are used to compare the relative odds of the occurrence of the outcome of interestingness, given exposure to the variable of interest. It is a way to quantify how strongly the presence or absence of an item (e.g. 5330) is associated with the presence or absence of another item (e.g. 12456) in the supermarket.txt dataset

4. Given the ability to discover frequent itemsets and association rules, propose a strategy to use these tools to study different customer segments, shops, shopping times, or specific products.

Proposing a strategy will depend to a large extent on the labels of the data however the first step will be to find the frequency of products in the data set after which we match the frequency of the products in each shop. It will be sensible that after matching the frequency of the products in each shop we check out the times when these products were bought in each shop. We can also mine for the different times some specific products (with high frequency) were bought in shops. For the customer segment it will then make sense to mine the frequent item sets in order to understand what group of items were bought (This should give us an idea of the different types of customers). Furthermore, we can also mine for which combinations were bought more in the shops.

5. Select some relatively high-support high-confidence rule (A->B) and based on that example describe the conditional probabilities P(A|B) and P(B|A), as well as the Bayes rule.

From my top 5 high confidence and high support, the rule with the highest confidence and support is

lhs	rhs	support	confidence	lift
{12456} => {5330}		0.007012844	1.0000000	5.119403

Considering the above let A = lhs and B = rhs hence A = 12456 B = 5330

Analyzing the contingency table for the rule "RULE 12456 => 5330"

```
#Filter the Rules Dataframe for where the item 12456 is present
subset12456 <- filter(supermarketDF, V1 == "12456" | V2 == "12456" | V3 == "1
2456" |V4 == "12456" |V5 == "12456" |V6 == "12456" |V7 == "12456" |V8 == "124
56" |V9 == "12456" |V10 == "12456" |V11 == "12456" )

#Total number of observables where
nrow(subset12456)

## [1] 178

#Filter the Rules Dataframe for where the item 12456 and 5330 are present
subset12456_5330 <- filter(subset5330, V1 == "12456" | V2 == "12456" | V3 ==
"12456" |V4 == "12456" |V5 == "12456" |V6 == "12456" |V7 == "12456" |V8 == "1
2456" |V9 == "12456" |V10 == "12456" |V11 == "12456" )

#Total number of observables where
nrow(subset12456_5330)

## [1] 86
```

```r
#Initialize the contingency table
contigencyTable <- matrix(c(86,92,178,4872,27510,32382,4958,27602,32560),ncol
=3,byrow=TRUE)
colnames(contigencyTable) <- c("5330","NOT 5330","TOTAL")
rownames(contigencyTable) <- c("12456","NOT 12456","TOTAL")

print(contigencyTable)
```

```
##           5330 NOT 5330 TOTAL
## 12456       86       92   178
## NOT 12456 4872    27510 32382
## TOTAL     4958    27602 32560
```

To calcultae p(A) P(A) = n(A)/n(S) where n(A) refers to number of A present n(A) == n(12456)

where n(S) total number in Sample space n(S) == n(Total)

From Contingency table n(12456) = 178

n(Total) = 32560

p(12456) = 178/32560 p(12456) = 0.00546683

To calcultae p(B) P(B) = n(B)/n(S) where n(B) refers to number of B present n(B) == n(5330)

where n(S) total number in Sample space n(S) == n(Total)

From Contingency table n(5330) = 4958

n(Total) = 32560

p(5330) = 4958/32560 p(5330) = 0.1522727

Now from the contingency table we have N(AnB) = 86

P(AnB) = n(AnB)/n(S) P(12456n5330) = 86/32560 P(12456n5330) = 0.002641278

Now we can calculate for P(A|B) and P(B|A)

P(A|B) = P(AnB)/P(B) = 0.002641278/0.1522727 = 0.01734571

P(B|A) = P(AnB)/P(A) = 0.002641278/0.00546683 = 0.4831462

Using the Bayes Rule P(A|B) = (P(B|A)P(A))/P(B)

Hence for Bayes Rule Have gotten the following

P(B|A) = 0.4831462 P(A) = 0.00546683 P(B) = 0.1522727

P(A|B) = (0.4831462*0.00546683)/0.1522727 = 0.01734571

From the above we can see that my earlier values for P(A|B) calculated without using the bayes rule corresponds with the P(A|B) using the Bayes rule

6. (Bonus 2p) Run Krimp on same data, provide commands and describe your findings and compare to FIM+Association rules. (link to Krimp documentation)