**R-Pattern-Rec**

**GOAL**

* Analyze a subset of transaction data of a time interval (ex: hour) of all “BUY” transactions. Each row is a “Buy Side” (wealth mgt, hedge fund, even market maker) trader of same firm into your execution broker business
* Identify key trends for “actionable insight”.
* Such “insight” may be a list of securities that trade at the exact same time by same trader –ex: an algo may combine multiple securities that may be related –ex: pairs trading or trading an industry sector or combining 2 industry sectors
* Action from this insight may include providing customers with analytics of the securities they traded together. Why?
  + To track their performance
  + get real time ratings of the securities to traders so they will best assess how long to keep these securities or increase their holdings
    - (real time alpha seeking results may serve this purpose)
    - Hence, custom real time alpha output to these traders may be a service they will desire (and pay the Exec Broker for)

Why subset?

For a quick example of what is possible in R-Studio. One can also do this for sells and other strategies. This example will highlight what is possible for securities trading using Machine Learning (ML) Pattern Recognition (PR).

\*\*\* in the end, we found that buying defense and energy stocks together was a major trend. As an Exec Broker, with this insight, my immediate action would be to provide real time alpha seeking to the buy side firm.

How do we get there?

**Let’s START**

Instead of using actual securities, I made up securities. The first letter of each security identifies industry segment of “group”, as follows:

A – Auto

B – Banks

C – Consumer

D – Defense

E – Energy

F – Food

H – Health-Care

So DA is a defense stock.

Imagine a time period (indeed a possibility) when Defense Spending is expected to increase, along with increased tension in the Middle East, oil production problems in Russia, and a revision to Government Health Care that will allow consumers to choose insurance companies across state lines, and a slowdown in GDP from 2% to 1%.. Such a scenario may result in Bullish expectations of Defense and Energy stocks, possibly increase for Health Care stocks, neutral for Food, slightly bearish for Consumer products, and more bearish for auto and bank stocks. How can we confirm this and identify core trends for actionable insight?

We have ML PR software that can automate this.

Her is a snippet of data (each row a trader transaction)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DA | DB | DC | DD | DE | EA |
| DA | DB | EA | EB | EC | ED |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DA | DB | DC | DD | DE | DF | DG | EA |
| DA | DB | EA | EB | EC | ED |  |  |
| DA | DB | DC | DD | DE | EA | EB |  |
| DA | DB | EA | EB | EC | ED | EE |  |
| DA | DB | DC | DD | DE | EA |  |  |
| DA | DB | EA | EB | EC | ED |  |  |
| DA | DB | DC | DD | DE | EA | EB | EC |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EA | EB | EC |  |  |  |
| EA | EB | EC | ED |  |  |
| EA | EB | EC | ED | DA |  |
| EA | EB | EC |  |  |  |
| EA | EB | EC | ED | DA | DB |

Oops – unexpected – last line below indicates some buying of auto, bank, consumer product stocks. Our analysis will determine whether that is rare.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DA | DB | HA | HB |  |  |  |
| DA | DB | HA | HB | CA | BA | AA |

Data has 1500 rows

1 challenge is for this data to be formatted in this way – EZ via Perl or Python …

So we have this data. What software can do this?

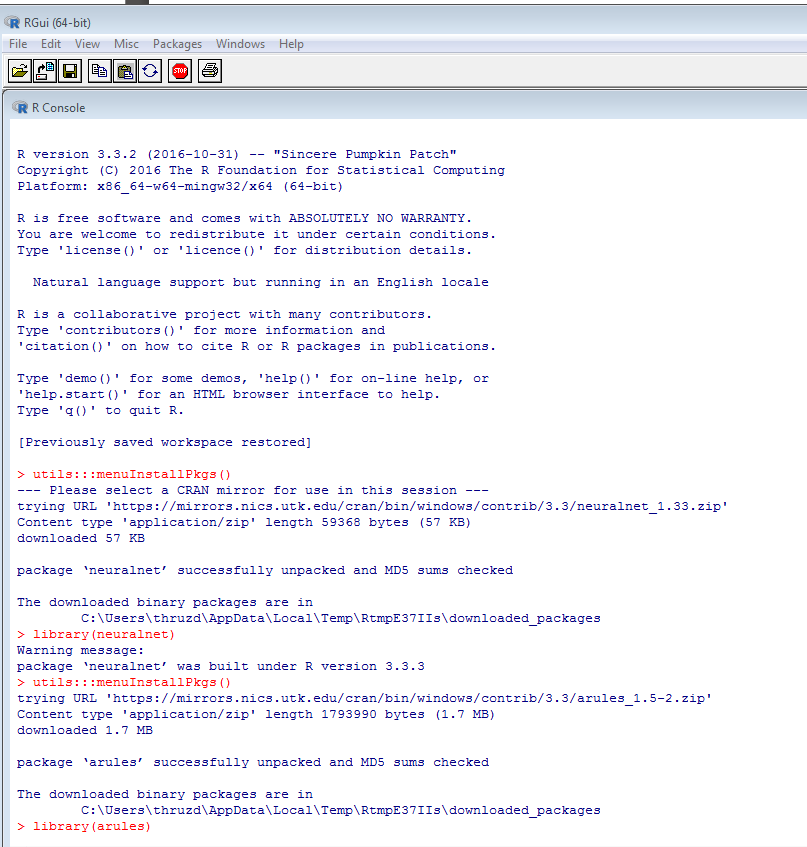
RStudio with “arules” package.

**START THE SOFTWARE, ADD THE PACKAGE, CODE A SOLUTION FOR ACTIONABLE INSIGHT:**

In R-Studio ….

Hit the Packages Menu, choose “arules”. I also choose neuralnet but I think I do not need it yet – in the example. Or type in

* **Install.packages(“arules”)**



In RED are what you add; note my comments and output from commands.

**> library(arules)**

Loading required package: Matrix

Attaching package: ‘arules’

The following objects are masked from ‘package:base’:

abbreviate, write

Warning message:

package ‘arules’ was built under R version 3.3.3

READ IN YOUR DATA

**> g<-read.transactions("H:\\R\\G.csv",sep=",")**

**\*\*\*\*\* NOTE: G.csv is the file I provided; save it wherever and modify the command above.**

VERIFY YOUR DATA WAS READ

**> summary(g**)

transactions as itemMatrix in sparse format with

1500 rows (elements/itemsets/transactions) and

19 columns (items) and a density of 0.2821754

\*\*\*\* “density” is proportion of non zero cells. A “sparse matrix” is a matrix with all possible inputs (columns or all 19 securities per data I provided) to each entity (row). Hence many columns will have nothing or “0”. Those with securities will have 1. Add all 1’s then divide by rows\*columns for “density”.

most frequent items:

EA EB EC DA DB (Other)

1232 1072 1024 1012 964 2738

**\*\*\* BELOW the #’s 3 to 9 are columns for # of trades per trader algo (ex: 3 means 3 trades @ 1 time; 9 means 9trades @ 1 time**

element (itemset/transaction) length distribution:

sizes

3 4 5 6 7 8 9

268 272 100 484 264 110 2

Min. 1st Qu. Median Mean 3rd Qu. Max.

3.000 4.000 6.000 5.361 7.000 9.000

includes extended item information - examples:

labels

1 AA

2 BA

3 CA

>

**JUST LOOK AT 1st 5 (from your Data …)**

**> inspect(g[1:5])**

items

[1] {DA,DB,DC,DD,DE,EA}

[2] {DA,DB,EA,EB,EC,ED}

[3] {DA,DB,DC,DD,DE}

[4] {DA,DB,EA,EB,EC,ED}

[5] {DA,DB,DC,DD,DE,DF,DG,EA}

>

In all 19 columns what % per column is in **sparse matrix**

> itemFrequency(g[1:19])

AA BA CA DA DB DC DD DE DF DG

0.00000000 0.00000000 0.00000000 1.00000000 1.00000000 0.52631579 0.52631579 0.52631579 0.10526316 0.10526316

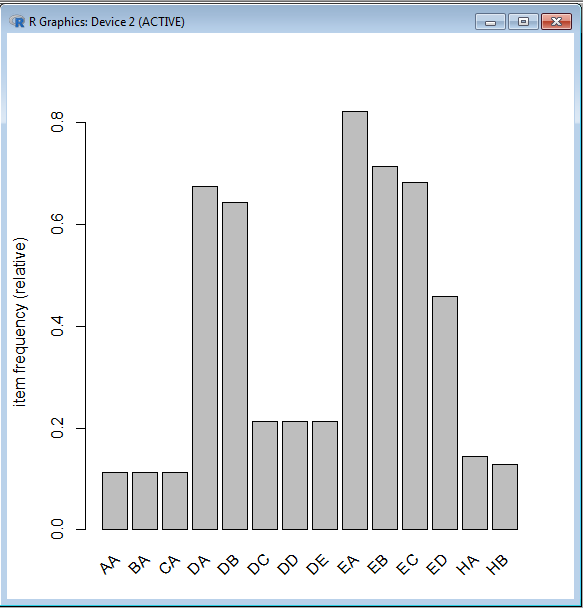
EA EB EC ED EE F1 FA HA HB

0.89473684 0.63157895 0.52631579 0.47368421 0.05263158 0.00000000 0.00000000 0.00000000 0.00000000

**NEXT GRAPH**

**Support=0.05 means include securities that were in at least 5% of the 1500 transactions.**

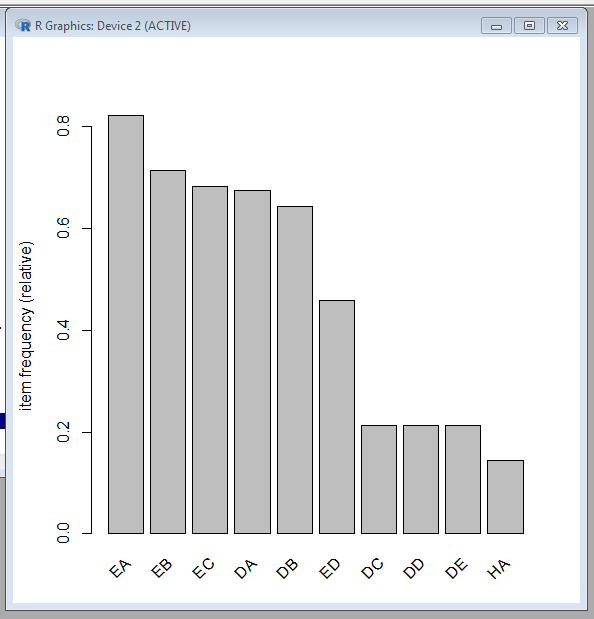
* **itemFrequencyPlot(g,support=0.05)**



**Graph Top 10**

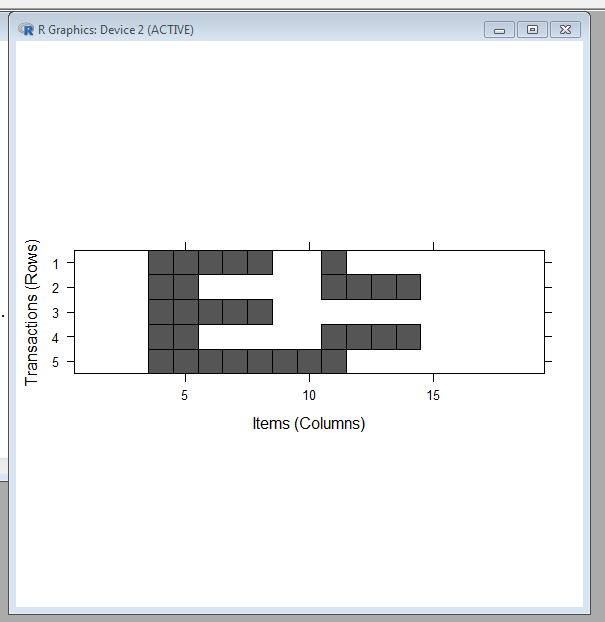
**> itemFrequencyPlot(g,topN=10)**

>



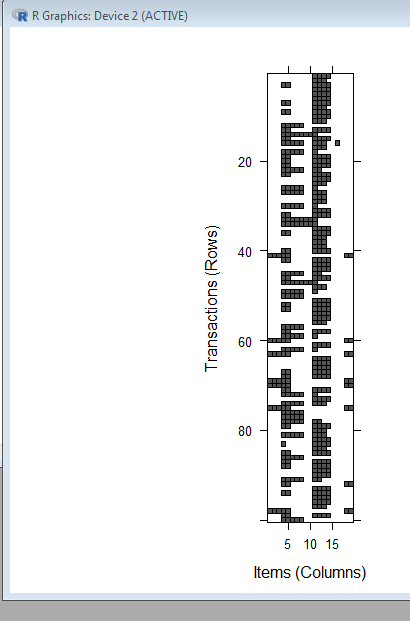
**GRAPH 1st 5 transactions**

**> image(g[1:5])**



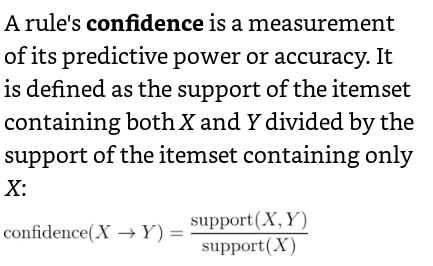
**Sorry – manually (or more R code) one would need map all columns #’s with (securities) in the spares matrix.**

**How about at random – 100 transactions?**



**How many rules does our sparse matrix have? we attempt to use the default settings of support = 0.1 and confidence = 0.8,**

**Define confidence as:**



**> apriori(g)**

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

0.8 0.1 1 none FALSE TRUE 5 0.1 1 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 150

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[19 item(s), 1500 transaction(s)] done [0.00s].

sorting and recoding items ... [14 item(s)] done [0.00s].

creating transaction tree ... done [0.00s].

checking subsets of size 1 2 3 4 5 6 7 done [0.00s].

writing ... [691 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

**set of 691 rules**

**it found 691 rules such as:**

**if I buy A and B, I have some C … (this is an example of a rule**

**all 691 rules cover these possibilities**

**if I buy A and B and have no Z then that does not comprise a “rule”**

**LETS TAILOR THIS; I WANT LESS # OF RULES. I ONLY WANT TO ATTAIN ACTIONABLE INSIGHT ON ITEMS WITH HIGHER CONFIDENCE (85%, NOT 80%, GREATER SUPPORT 20%, NOT PRIOR 10%, AND TOTAL GROUPINGS OF 4 TO 9; NOT INTERESTED IN 3 …..**

**> grules <-apriori(g,parameter=list(support=0.2, confidence=0.85, minlen=4))**

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

0.85 0.1 1 none FALSE TRUE 5 0.2 4 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 300

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[19 item(s), 1500 transaction(s)] done [0.00s].

sorting and recoding items ... [9 item(s)] done [0.00s].

creating transaction tree ... done [0.00s].

checking subsets of size 1 2 3 4 5 6 done [0.00s].

**writing ... [103 rule(s)] done [0.00s].**

creating S4 object ... done [0.00s].

>

**SO 103 rules**

**> grules**

set of 103 rules

>

**Inspect 1st 5**

**> inspect(grules[1:5])**

lhs rhs support confidence lift

[1] {DC,DD,DE} => {DB} 0.2133333 1 1.556017

[2] {DB,DC,DD} => {DE} 0.2133333 1 4.687500

[3] {DB,DC,DE} => {DD} 0.2133333 1 4.687500

[4] {DB,DD,DE} => {DC} 0.2133333 1 4.687500

[5] {DC,DD,DE} => {DA} 0.2133333 1 1.482213

>

**The lift of a rule measures how much more likely one item or itemset is accessed (or bought in our example) relative to its typical rate of purchase (standalone), given that you know another item or itemset has been purchased. This is defined by the following equation:**

**LIFT(x->Y) = confidence (X->Y)/support(X)**

**> summary(grules)**

set of 103 rules

rule length distribution (lhs + rhs):sizes

4 5 6

65 32 6

Min. 1st Qu. Median Mean 3rd Qu. Max.

4.000 4.000 4.000 4.427 5.000 6.000

summary of quality measures:

support confidence lift

Min. :0.2133 Min. :0.8525 Min. :1.218

1st Qu.:0.2773 1st Qu.:1.0000 1st Qu.:1.395

Median :0.2773 Median :1.0000 Median :1.465

Mean :0.2825 Mean :0.9722 Mean :1.921

3rd Qu.:0.3093 3rd Qu.:1.0000 3rd Qu.:1.556

Max. :0.4587 Max. :1.0000 Max. :4.688

mining info:

data ntransactions support confidence

g 1500 0.2 0.85

**sort by highest lift – these can be actionable insights (especially those lifts over 4 – 4 times more likely with the other securities than standalone. These are rows 1-15; however, rows 16-25 indicated ‘lift’ of almost 2 so defense / energy stocks together should also lead to actionable insight.**

**> inspect(sort(grules,by="lift")[1:40])**

lhs rhs support confidence lift

**[1] {DB,DC,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[2] {DB,DC,DE} => {DD} 0.2133333 1.0000000 4.687500**

**[3] {DB,DD,DE} => {DC} 0.2133333 1.0000000 4.687500**

**[4] {DA,DC,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[5] {DA,DC,DE} => {DD} 0.2133333 1.0000000 4.687500**

**[6] {DA,DD,DE} => {DC} 0.2133333 1.0000000 4.687500**

**[7] {DA,DB,DC} => {DD} 0.2133333 1.0000000 4.687500**

**[8] {DA,DB,DD} => {DC} 0.2133333 1.0000000 4.687500**

**[9] {DA,DB,DC} => {DE} 0.2133333 1.0000000 4.687500**

**[10] {DA,DB,DE} => {DC} 0.2133333 1.0000000 4.687500**

**[11] {DA,DB,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[12] {DA,DB,DE} => {DD} 0.2133333 1.0000000 4.687500**

**[13] {DA,DB,DC,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[14] {DA,DB,DC,DE} => {DD} 0.2133333 1.0000000 4.687500**

**[15] {DA,DB,DD,DE} => {DC} 0.2133333 1.0000000 4.687500**

**[16] {DA,EB,EC} => {ED} 0.3093333 0.8656716 1.887365**

**[17] {DA,EA,EC} => {ED} 0.3093333 0.8656716 1.887365**

**[18] {DA,EA,EB,EC} => {ED} 0.3093333 0.8656716 1.887365**

**[19] {DB,EB,EC} => {ED} 0.2773333 0.8524590 1.858559**

**[20] {DA,DB,EC} => {ED} 0.2773333 0.8524590 1.858559**

**[21] {DB,EA,EC} => {ED} 0.2773333 0.8524590 1.858559**

**[22] {DA,DB,EB,EC} => {ED} 0.2773333 0.8524590 1.858559**

**[23] {DB,EA,EB,EC} => {ED} 0.2773333 0.8524590 1.858559**

**[24] {DA,DB,EA,EC} => {ED} 0.2773333 0.8524590 1.858559**

**[25] {DA,DB,EA,EB,EC} => {ED} 0.2773333 0.8524590 1.858559**

[26] {DC,DD,DE} => {DB} 0.2133333 1.0000000 1.556017

[27] {DA,DC,DD} => {DB} 0.2133333 1.0000000 1.556017

[28] {DA,DC,DE} => {DB} 0.2133333 1.0000000 1.556017

[29] {DA,DD,DE} => {DB} 0.2133333 1.0000000 1.556017

[30] {DA,DC,DD,DE} => {DB} 0.2133333 1.0000000 1.556017

[31] {DC,DD,DE} => {DA} 0.2133333 1.0000000 1.482213

[32] {DB,DC,DD} => {DA} 0.2133333 1.0000000 1.482213

[33] {DB,DC,DE} => {DA} 0.2133333 1.0000000 1.482213

[34] {DB,DD,DE} => {DA} 0.2133333 1.0000000 1.482213

[35] {DB,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[36] {DB,EB,ED} => {DA} 0.2773333 1.0000000 1.482213

[37] {DB,EA,ED} => {DA} 0.2773333 1.0000000 1.482213

[38] {DB,EB,EC} => {DA} 0.3253333 1.0000000 1.482213

[39] {DB,EA,EC} => {DA} 0.3253333 1.0000000 1.482213

[40] {DB,EA,EB} => {DA} 0.3573333 1.0000000 1.482213

>

**Take subsets and just focus on what may be 2 popular securities – EA and DA ….**

> EArules<-subset(grules,items %in% "EA")

> inspect(EArules)

lhs rhs support confidence lift

[1] {EB,EC,ED} => {EA} 0.4586667 1.0000000 1.217532

[2] {EA,EC,ED} => {EB} 0.4586667 1.0000000 1.399254

[3] {EA,EB,ED} => {EC} 0.4586667 1.0000000 1.464844

[4] {DB,EC,ED} => {EA} 0.2773333 1.0000000 1.217532

[5] {DB,EA,ED} => {EC} 0.2773333 1.0000000 1.464844

[6] {DB,EA,EC} => {ED} 0.2773333 0.8524590 1.858559

[7] {DA,EC,ED} => {EA} 0.3093333 1.0000000 1.217532

[8] {DA,EA,ED} => {EC} 0.3093333 1.0000000 1.464844

[9] {DA,EA,EC} => {ED} 0.3093333 0.8656716 1.887365

[10] {DB,EB,ED} => {EA} 0.2773333 1.0000000 1.217532

[11] {DB,EA,ED} => {EB} 0.2773333 1.0000000 1.399254

[12] {DA,EB,ED} => {EA} 0.3093333 1.0000000 1.217532

[13] {DA,EA,ED} => {EB} 0.3093333 1.0000000 1.399254

[14] {DA,DB,ED} => {EA} 0.2773333 1.0000000 1.217532

[15] {DB,EA,ED} => {DA} 0.2773333 1.0000000 1.482213

[16] {DA,EA,ED} => {DB} 0.2773333 0.8965517 1.395049

[17] {DB,EB,EC} => {EA} 0.3253333 1.0000000 1.217532

[18] {DB,EA,EC} => {EB} 0.3253333 1.0000000 1.399254

[19] {DB,EA,EB} => {EC} 0.3253333 0.9104478 1.333664

[20] {DA,EB,EC} => {EA} 0.3573333 1.0000000 1.217532

[21] {DA,EA,EC} => {EB} 0.3573333 1.0000000 1.399254

[22] {DA,EA,EB} => {EC} 0.3573333 0.9178082 1.344446

[23] {DA,DB,EC} => {EA} 0.3253333 1.0000000 1.217532

[24] {DB,EA,EC} => {DA} 0.3253333 1.0000000 1.482213

[25] {DA,EA,EC} => {DB} 0.3253333 0.9104478 1.416672

[26] {DA,DB,EB} => {EA} 0.3573333 1.0000000 1.217532

[27] {DB,EA,EB} => {DA} 0.3573333 1.0000000 1.482213

[28] {DA,EA,EB} => {DB} 0.3573333 0.9178082 1.428125

[29] {DB,EB,EC,ED} => {EA} 0.2773333 1.0000000 1.217532

[30] {DB,EA,EC,ED} => {EB} 0.2773333 1.0000000 1.399254

[31] {DB,EA,EB,ED} => {EC} 0.2773333 1.0000000 1.464844

[32] {DB,EA,EB,EC} => {ED} 0.2773333 0.8524590 1.858559

[33] {DA,EB,EC,ED} => {EA} 0.3093333 1.0000000 1.217532

[34] {DA,EA,EC,ED} => {EB} 0.3093333 1.0000000 1.399254

[35] {DA,EA,EB,ED} => {EC} 0.3093333 1.0000000 1.464844

[36] {DA,EA,EB,EC} => {ED} 0.3093333 0.8656716 1.887365

[37] {DA,DB,EC,ED} => {EA} 0.2773333 1.0000000 1.217532

[38] {DB,EA,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[39] {DA,EA,EC,ED} => {DB} 0.2773333 0.8965517 1.395049

[40] {DA,DB,EA,ED} => {EC} 0.2773333 1.0000000 1.464844

[41] {DA,DB,EA,EC} => {ED} 0.2773333 0.8524590 1.858559

[42] {DA,DB,EB,ED} => {EA} 0.2773333 1.0000000 1.217532

[43] {DB,EA,EB,ED} => {DA} 0.2773333 1.0000000 1.482213

[44] {DA,EA,EB,ED} => {DB} 0.2773333 0.8965517 1.395049

[45] {DA,DB,EA,ED} => {EB} 0.2773333 1.0000000 1.399254

[46] {DA,DB,EB,EC} => {EA} 0.3253333 1.0000000 1.217532

[47] {DB,EA,EB,EC} => {DA} 0.3253333 1.0000000 1.482213

[48] {DA,EA,EB,EC} => {DB} 0.3253333 0.9104478 1.416672

[49] {DA,DB,EA,EC} => {EB} 0.3253333 1.0000000 1.399254

[50] {DA,DB,EA,EB} => {EC} 0.3253333 0.9104478 1.333664

[51] {DA,DB,EB,EC,ED} => {EA} 0.2773333 1.0000000 1.217532

[52] {DB,EA,EB,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[53] {DA,EA,EB,EC,ED} => {DB} 0.2773333 0.8965517 1.395049

[54] {DA,DB,EA,EC,ED} => {EB} 0.2773333 1.0000000 1.399254

[55] {DA,DB,EA,EB,ED} => {EC} 0.2773333 1.0000000 1.464844

[56] {DA,DB,EA,EB,EC} => {ED} 0.2773333 0.8524590 1.858559

> DArules<-subset(grules,items %in% "DA")

> inspect(DArules)

lhs rhs support confidence lift

[1] {DC,DD,DE} => {DA} 0.2133333 1.0000000 1.482213

**[2] {DA,DC,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[3] {DA,DC,DE} => {DD} 0.2133333 1.0000000 4.687500**

**[4] {DA,DD,DE} => {DC} 0.2133333 1.0000000 4.687500**

[5] {DB,DC,DD} => {DA} 0.2133333 1.0000000 1.482213

[6] {DA,DC,DD} => {DB} 0.2133333 1.0000000 1.556017

[**7] {DA,DB,DC} => {DD} 0.2133333 1.0000000 4.687500**

**[8] {DA,DB,DD} => {DC} 0.2133333 1.0000000 4.687500**

[9] {DB,DC,DE} => {DA} 0.2133333 1.0000000 1.482213

[10] {DA,DC,DE} => {DB} 0.2133333 1.0000000 1.556017

[**11] {DA,DB,DC} => {DE} 0.2133333 1.0000000 4.687500**

**[12] {DA,DB,DE} => {DC} 0.2133333 1.0000000 4.687500**

[13] {DB,DD,DE} => {DA} 0.2133333 1.0000000 1.482213

[14] {DA,DD,DE} => {DB} 0.2133333 1.0000000 1.556017

[**15] {DA,DB,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[16] {DA,DB,DE} => {DD} 0.2133333 1.0000000 4.687500**

[17] {DA,EC,ED} => {EB} 0.3093333 1.0000000 1.399254

[18] {DA,EB,ED} => {EC} 0.3093333 1.0000000 1.464844

[19] {DA,EB,EC} => {ED} 0.3093333 0.8656716 1.887365

[20] {DB,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[21] {DA,EC,ED} => {DB} 0.2773333 0.8965517 1.395049

[22] {DA,DB,ED} => {EC} 0.2773333 1.0000000 1.464844

[23] {DA,DB,EC} => {ED} 0.2773333 0.8524590 1.858559

[24] {DA,EC,ED} => {EA} 0.3093333 1.0000000 1.217532

[25] {DA,EA,ED} => {EC} 0.3093333 1.0000000 1.464844

[26] {DA,EA,EC} => {ED} 0.3093333 0.8656716 1.887365

[27] {DB,EB,ED} => {DA} 0.2773333 1.0000000 1.482213

[28] {DA,EB,ED} => {DB} 0.2773333 0.8965517 1.395049

[29] {DA,DB,ED} => {EB} 0.2773333 1.0000000 1.399254

[30] {DA,EB,ED} => {EA} 0.3093333 1.0000000 1.217532

[31] {DA,EA,ED} => {EB} 0.3093333 1.0000000 1.399254

[32] {DA,DB,ED} => {EA} 0.2773333 1.0000000 1.217532

[33] {DB,EA,ED} => {DA} 0.2773333 1.0000000 1.482213

[34] {DA,EA,ED} => {DB} 0.2773333 0.8965517 1.395049

[35] {DB,EB,EC} => {DA} 0.3253333 1.0000000 1.482213

[36] {DA,EB,EC} => {DB} 0.3253333 0.9104478 1.416672

[37] {DA,DB,EC} => {EB} 0.3253333 1.0000000 1.399254

[38] {DA,DB,EB} => {EC} 0.3253333 0.9104478 1.333664

[39] {DA,EB,EC} => {EA} 0.3573333 1.0000000 1.217532

[40] {DA,EA,EC} => {EB} 0.3573333 1.0000000 1.399254

[41] {DA,EA,EB} => {EC} 0.3573333 0.9178082 1.344446

[42] {DA,DB,EC} => {EA} 0.3253333 1.0000000 1.217532

[43] {DB,EA,EC} => {DA} 0.3253333 1.0000000 1.482213

[44] {DA,EA,EC} => {DB} 0.3253333 0.9104478 1.416672

[45] {DA,DB,EB} => {EA} 0.3573333 1.0000000 1.217532

[46] {DB,EA,EB} => {DA} 0.3573333 1.0000000 1.482213

[47] {DA,EA,EB} => {DB} 0.3573333 0.9178082 1.428125

[48] {DB,DC,DD,DE} => {DA} 0.2133333 1.0000000 1.482213

[49] {DA,DC,DD,DE} => {DB} 0.2133333 1.0000000 1.556017

[**50] {DA,DB,DC,DD} => {DE} 0.2133333 1.0000000 4.687500**

**[51] {DA,DB,DC,DE} => {DD} 0.2133333 1.0000000 4.687500**

**[52] {DA,DB,DD,DE} => {DC} 0.2133333 1.0000000 4.687500**

[53] {DB,EB,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[54] {DA,EB,EC,ED} => {DB} 0.2773333 0.8965517 1.395049

[55] {DA,DB,EC,ED} => {EB} 0.2773333 1.0000000 1.399254

[56] {DA,DB,EB,ED} => {EC} 0.2773333 1.0000000 1.464844

[57] {DA,DB,EB,EC} => {ED} 0.2773333 0.8524590 1.858559

[58] {DA,EB,EC,ED} => {EA} 0.3093333 1.0000000 1.217532

[59] {DA,EA,EC,ED} => {EB} 0.3093333 1.0000000 1.399254

[60] {DA,EA,EB,ED} => {EC} 0.3093333 1.0000000 1.464844

[61] {DA,EA,EB,EC} => {ED} 0.3093333 0.8656716 1.887365

[62] {DA,DB,EC,ED} => {EA} 0.2773333 1.0000000 1.217532

[63] {DB,EA,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[64] {DA,EA,EC,ED} => {DB} 0.2773333 0.8965517 1.395049

[65] {DA,DB,EA,ED} => {EC} 0.2773333 1.0000000 1.464844

[66] {DA,DB,EA,EC} => {ED} 0.2773333 0.8524590 1.858559

[67] {DA,DB,EB,ED} => {EA} 0.2773333 1.0000000 1.217532

[68] {DB,EA,EB,ED} => {DA} 0.2773333 1.0000000 1.482213

[69] {DA,EA,EB,ED} => {DB} 0.2773333 0.8965517 1.395049

[70] {DA,DB,EA,ED} => {EB} 0.2773333 1.0000000 1.399254

[71] {DA,DB,EB,EC} => {EA} 0.3253333 1.0000000 1.217532

[72] {DB,EA,EB,EC} => {DA} 0.3253333 1.0000000 1.482213

[73] {DA,EA,EB,EC} => {DB} 0.3253333 0.9104478 1.416672

[74] {DA,DB,EA,EC} => {EB} 0.3253333 1.0000000 1.399254

[75] {DA,DB,EA,EB} => {EC} 0.3253333 0.9104478 1.333664

[76] {DA,DB,EB,EC,ED} => {EA} 0.2773333 1.0000000 1.217532

[77] {DB,EA,EB,EC,ED} => {DA} 0.2773333 1.0000000 1.482213

[78] {DA,EA,EB,EC,ED} => {DB} 0.2773333 0.8965517 1.395049

[79] {DA,DB,EA,EC,ED} => {EB} 0.2773333 1.0000000 1.399254

[80] {DA,DB,EA,EB,ED} => {EC} 0.2773333 1.0000000 1.464844

[81] {DA,DB,EA,EB,EC} => {ED} 0.2773333 0.8524590 1.858559

**Save the rules to a file for potential additional analysis; convert the grules into an R data frame to examine it in R-Studio**

**> write(grules,file="H:\\R\\grules.csv",sep=",",quote=TRUE,row.names=FALSE)**

**> grules\_df<-as(grules,"data.frame")**

**> str(grules\_df)**

'data.frame': 103 obs. of 4 variables:

$ rules : Factor w/ 103 levels "{DA,DB,DC,DD} => {DE}",..: 100 68 70 72 99 35 37 39 67 34 ...

$ support : num 0.213 0.213 0.213 0.213 0.213 ...

$ confidence: num 1 1 1 1 1 1 1 1 1 1 ...

$ lift : num 1.56 4.69 4.69 4.69 1.48 ...

>

|  |  |  |  |
| --- | --- | --- | --- |
| rules | support | confidence | lift |
| {DC,DD,DE} => {DB} | 0.213333333 | 1 | 1.556016598 |
| {DB,DC,DD} => {DE} | 0.213333333 | 1 | 4.6875 |
| {DB,DC,DE} => {DD} | 0.213333333 | 1 | 4.6875 |
| {DB,DD,DE} => {DC} | 0.213333333 | 1 | 4.6875 |
| {DC,DD,DE} => {DA} | 0.213333333 | 1 | 1.482213439 |
| {DA,DC,DD} => {DE} | 0.213333333 | 1 | 4.6875 |
| {DA,DC,DE} => {DD} | 0.213333333 | 1 | 4.6875 |
| {DA,DD,DE} => {DC} | 0.213333333 | 1 | 4.6875 |
| {DB,DC,DD} => {DA} | 0.213333333 | 1 | 1.482213439 |
| {DA,DC,DD} => {DB} | 0.213333333 | 1 | 1.556016598 |
| {DA,DB,DC} => {DD} | 0.213333333 | 1 | 4.6875 |
| {DA,DB,DD} => {DC} | 0.213333333 | 1 | 4.6875 |
| {DB,DC,DE} => {DA} | 0.213333333 | 1 | 1.482213439 |
| {DA,DC,DE} => {DB} | 0.213333333 | 1 | 1.556016598 |
| {DA,DB,DC} => {DE} | 0.213333333 | 1 | 4.6875 |
| {DA,DB,DE} => {DC} | 0.213333333 | 1 | 4.6875 |
| {DB,DD,DE} => {DA} | 0.213333333 | 1 | 1.482213439 |
| {DA,DD,DE} => {DB} | 0.213333333 | 1 | 1.556016598 |
| {DA,DB,DD} => {DE} | 0.213333333 | 1 | 4.6875 |
| {DA,DB,DE} => {DD} | 0.213333333 | 1 | 4.6875 |
| {DB,EC,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DB,EB,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DB,EB,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
| {DA,EC,ED} => {EB} | 0.309333333 | 1 | 1.399253731 |
| {DA,EB,ED} => {EC} | 0.309333333 | 1 | 1.46484375 |
| {DA,EB,EC} => {ED} | 0.309333333 | 0.865671642 | 1.887365498 |
| {EB,EC,ED} => {EA} | 0.458666667 | 1 | 1.217532468 |
| {EA,EC,ED} => {EB} | 0.458666667 | 1 | 1.399253731 |
| {EA,EB,ED} => {EC} | 0.458666667 | 1 | 1.46484375 |
| {DB,EC,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EC,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DA,DB,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DA,DB,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
| {DB,EC,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DB,EA,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
| {DA,EC,ED} => {EA} | 0.309333333 | 1 | 1.217532468 |
| {DA,EA,ED} => {EC} | 0.309333333 | 1 | 1.46484375 |
| {DA,EA,EC} => {ED} | 0.309333333 | 0.865671642 | 1.887365498 |
| {DB,EB,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EB,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DA,DB,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DB,EB,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DA,EB,ED} => {EA} | 0.309333333 | 1 | 1.217532468 |
| {DA,EA,ED} => {EB} | 0.309333333 | 1 | 1.399253731 |
| {DA,DB,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EA,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DB,EB,EC} => {DA} | 0.325333333 | 1 | 1.482213439 |
| {DA,EB,EC} => {DB} | 0.325333333 | 0.910447761 | 1.416671828 |
| {DA,DB,EC} => {EB} | 0.325333333 | 1 | 1.399253731 |
| {DA,DB,EB} => {EC} | 0.325333333 | 0.910447761 | 1.333663713 |
| {DB,EB,EC} => {EA} | 0.325333333 | 1 | 1.217532468 |
| {DB,EA,EC} => {EB} | 0.325333333 | 1 | 1.399253731 |
| {DB,EA,EB} => {EC} | 0.325333333 | 0.910447761 | 1.333663713 |
| {DA,EB,EC} => {EA} | 0.357333333 | 1 | 1.217532468 |
| {DA,EA,EC} => {EB} | 0.357333333 | 1 | 1.399253731 |
| {DA,EA,EB} => {EC} | 0.357333333 | 0.917808219 | 1.344445634 |
| {DA,DB,EC} => {EA} | 0.325333333 | 1 | 1.217532468 |
| {DB,EA,EC} => {DA} | 0.325333333 | 1 | 1.482213439 |
| {DA,EA,EC} => {DB} | 0.325333333 | 0.910447761 | 1.416671828 |
| {DA,DB,EB} => {EA} | 0.357333333 | 1 | 1.217532468 |
| {DB,EA,EB} => {DA} | 0.357333333 | 1 | 1.482213439 |
| {DA,EA,EB} => {DB} | 0.357333333 | 0.917808219 | 1.428124822 |
| {DB,DC,DD,DE} => {DA} | 0.213333333 | 1 | 1.482213439 |
| {DA,DC,DD,DE} => {DB} | 0.213333333 | 1 | 1.556016598 |
| {DA,DB,DC,DD} => {DE} | 0.213333333 | 1 | 4.6875 |
| {DA,DB,DC,DE} => {DD} | 0.213333333 | 1 | 4.6875 |
| {DA,DB,DD,DE} => {DC} | 0.213333333 | 1 | 4.6875 |
| {DB,EB,EC,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EB,EC,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DA,DB,EC,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DA,DB,EB,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DA,DB,EB,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
| {DB,EB,EC,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,EC,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DB,EA,EB,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DB,EA,EB,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
| {DA,EB,EC,ED} => {EA} | 0.309333333 | 1 | 1.217532468 |
| {DA,EA,EC,ED} => {EB} | 0.309333333 | 1 | 1.399253731 |
| {DA,EA,EB,ED} => {EC} | 0.309333333 | 1 | 1.46484375 |
| {DA,EA,EB,EC} => {ED} | 0.309333333 | 0.865671642 | 1.887365498 |
| {DA,DB,EC,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,EC,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EA,EC,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DA,DB,EA,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DA,DB,EA,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
| {DA,DB,EB,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,EB,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EA,EB,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DA,DB,EA,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DA,DB,EB,EC} => {EA} | 0.325333333 | 1 | 1.217532468 |
| {DB,EA,EB,EC} => {DA} | 0.325333333 | 1 | 1.482213439 |
| {DA,EA,EB,EC} => {DB} | 0.325333333 | 0.910447761 | 1.416671828 |
| {DA,DB,EA,EC} => {EB} | 0.325333333 | 1 | 1.399253731 |
| {DA,DB,EA,EB} => {EC} | 0.325333333 | 0.910447761 | 1.333663713 |
| {DA,DB,EB,EC,ED} => {EA} | 0.277333333 | 1 | 1.217532468 |
| {DB,EA,EB,EC,ED} => {DA} | 0.277333333 | 1 | 1.482213439 |
| {DA,EA,EB,EC,ED} => {DB} | 0.277333333 | 0.896551724 | 1.395049363 |
| {DA,DB,EA,EC,ED} => {EB} | 0.277333333 | 1 | 1.399253731 |
| {DA,DB,EA,EB,ED} => {EC} | 0.277333333 | 1 | 1.46484375 |
| {DA,DB,EA,EB,EC} => {ED} | 0.277333333 | 0.852459016 | 1.858558902 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**CONCUSION:**

**Securities DA, DB, DC, DD DE and**

**EA, EB, EC, ED were frequently bought together**

**Generate custom alpha seeking strategies and provide for customer**