**Prepare rules for the all the data sets**

**1) Try different values of support and confidence. Observe the change in number of rules for different support,confidence values**

**2) Change the minimum length in apriori algorithm**

**3) Visulize the obtained rules using different plots**

**# First data set I am using book.csv, create association rules related to different kind of books**

**# install the package**

install.packages("arules")

library("arules") # **invoke the package**

**#Visualizing Association Rules Package arulesViz supports visualization of**

**#association rules with scatter plot, balloon plot,**

**#graph, parallel coordinates plot, etc**

install.packages("arulesViz")

library("arulesViz")

**#importing data set**

Books<-read.csv(file.choose(),header = T)

View(Books)

head(Books)

ChildBks YouthBks CookBks DoItYBks RefBks ArtBks GeogBks ItalCook ItalAtlas ItalArt

1 0 1 0 1 0 0 1 0 0 0

2 1 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0 0

4 1 1 1 0 1 0 1 0 0 0

Florence

1 0

2 0

3 0

4 0

#check how many columns presents in data set

colnames(Books)

colnames(Books)

[1] "ChildBks" "YouthBks" "CookBks" "DoItYBks" "RefBks" "ArtBks" "GeogBks"

[8] "ItalCook" "ItalAtlas" "ItalArt" "Florence"

**# making rules using apriori algorithm**

**# Keep changing support and confidence values to obtain different rules**

Books\_rules <- apriori(as.matrix(Books,parameter=list(support=0.2,confidence=0.7)))

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 TRUE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 200

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

set transactions ...[11 item(s), 2000 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 done [0.00s].

writing ... [7 rule(s)] done [0.00s].

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

inspect(Books\_rules)

lhs rhs support confidence coverage lift count

[1] {ItalCook} => {CookBks} 0.1135 1.0000000 0.1135 2.320186 227

[2] {ChildBks,RefBks} => {CookBks} 0.1225 0.8085809 0.1515 1.876058 245

[3] {CookBks,RefBks} => {ChildBks} 0.1225 0.8032787 0.1525 1.899004 245

[4] {DoItYBks,ArtBks} => {CookBks} 0.1015 0.8218623 0.1235 1.906873 203

[5] {ArtBks,GeogBks} => {ChildBks} 0.1020 0.8000000 0.1275 1.891253 204

[6] {ArtBks,GeogBks} => {CookBks} 0.1035 0.8117647 0.1275 1.883445 207

[7] {DoItYBks,GeogBks} => {CookBks} 0.1085 0.8188679 0.1325 1.899926 217

**# here we got 7 rules**

**#sorting the rules by lift ratio**

inspect(sort(Books\_rules,by="lift"))

lhs rhs support confidence coverage lift count

[1] {ItalCook} => {CookBks} 0.1135 1.0000000 0.1135 2.320186 227

[2] {DoItYBks,ArtBks} => {CookBks} 0.1015 0.8218623 0.1235 1.906873 203

[3] {DoItYBks,GeogBks} => {CookBks} 0.1085 0.8188679 0.1325 1.899926 217

[4] {CookBks,RefBks} => {ChildBks} 0.1225 0.8032787 0.1525 1.899004 245

[5] {ArtBks,GeogBks} => {ChildBks} 0.1020 0.8000000 0.1275 1.891253 204

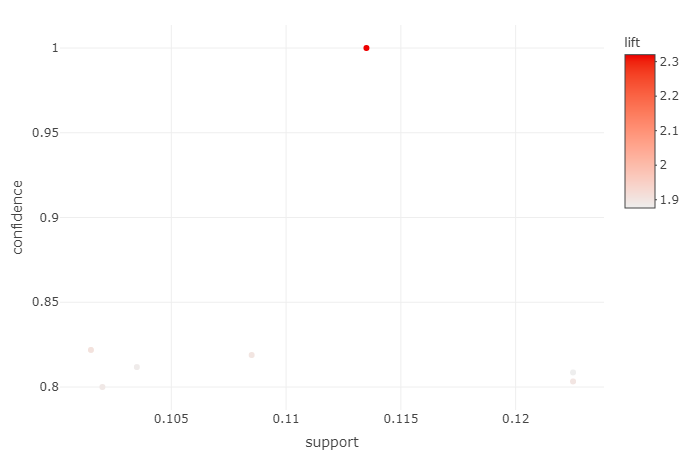
[6] {ArtBks,GeogBks} => {CookBks} 0.1035 0.8117647 0.1275 1.883445 207

[7] {ChildBks,RefBks} => {CookBks} 0.1225 0.8085809 0.1515 1.876058 245

**# here we consider top 5 rules based on highest lift ratio**

**#visualize the rules**

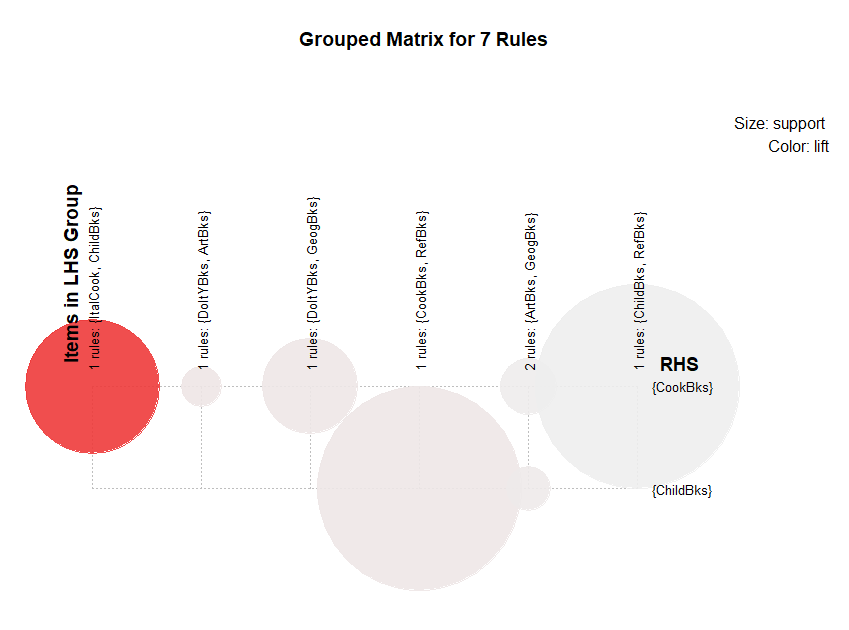
plotly\_arules(Books\_rules)



**# here the dark color dot indicate the highest lift ratio rules**

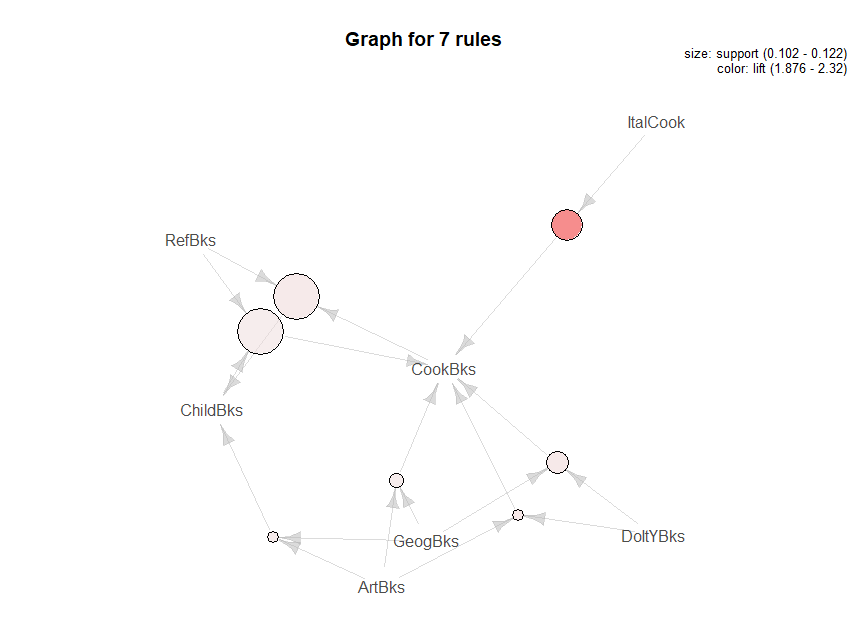
**# Different Ways of Visualizing Rules**

plot(Books\_rules,method="grouped")



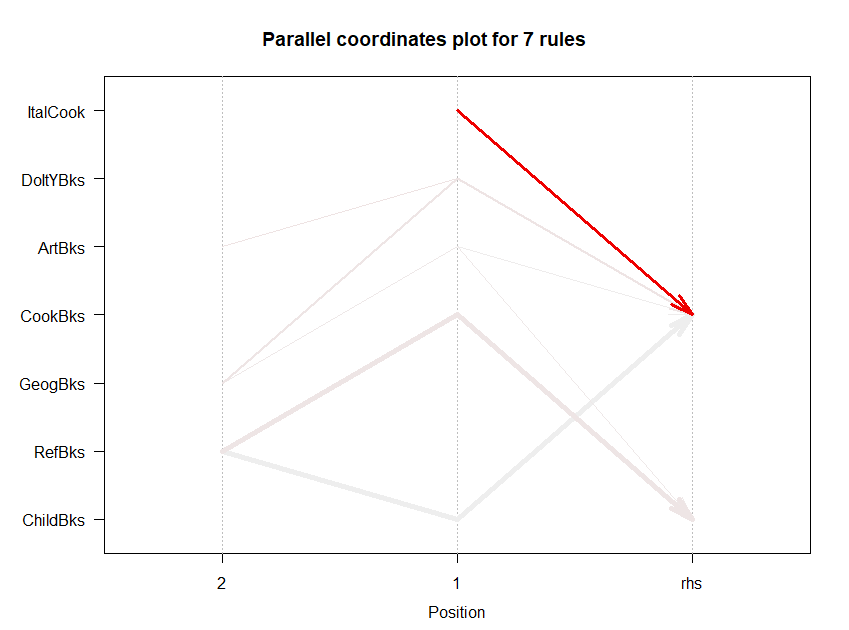
**#Graph**

plot(Books\_rules , method="graph", control=list(type="items"))



**#parallel coordinates plot**

plot(Books\_rules, method="paracoord", control=list(reorder=TRUE))



**#next we can try support=0.11, confidence=0.85**

Books\_rules2 <- apriori(as.matrix(Books,parameter=list(support=0.11,confidence=0.85)))

**#sorting the rules by lift ratio**

inspect(sort(Books\_rules2,by="lift"))

lhs rhs support confidence coverage lift count

[1] {ItalCook} => {CookBks} 0.1135 1.0000000 0.1135 2.320186 227

[2] {DoItYBks,ArtBks} => {CookBks} 0.1015 0.8218623 0.1235 1.906873 203

[3] {DoItYBks,GeogBks} => {CookBks} 0.1085 0.8188679 0.1325 1.899926 217

[4] {CookBks,RefBks} => {ChildBks} 0.1225 0.8032787 0.1525 1.899004 245

[5] {ArtBks,GeogBks} => {ChildBks} 0.1020 0.8000000 0.1275 1.891253 204

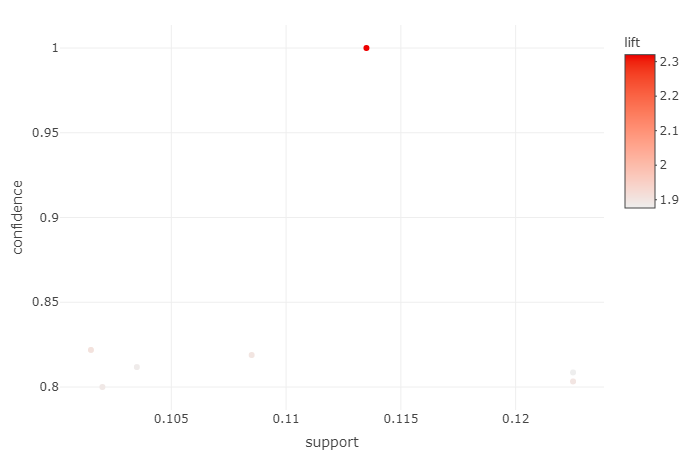
[6] {ArtBks,GeogBks} => {CookBks} 0.1035 0.8117647 0.1275 1.883445 207

[7] {ChildBks,RefBks} => {CookBks} 0.1225 0.8085809 0.1515 1.876058 245

**# here we consider top 5 rules based on highest lift ratio**

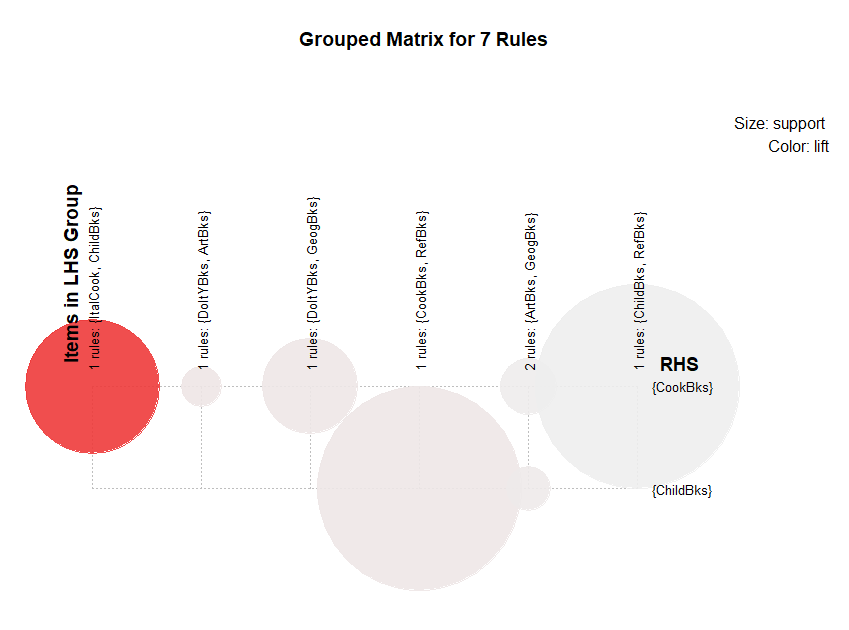
#visualize the rules

plotly\_arules(Books\_rules2)



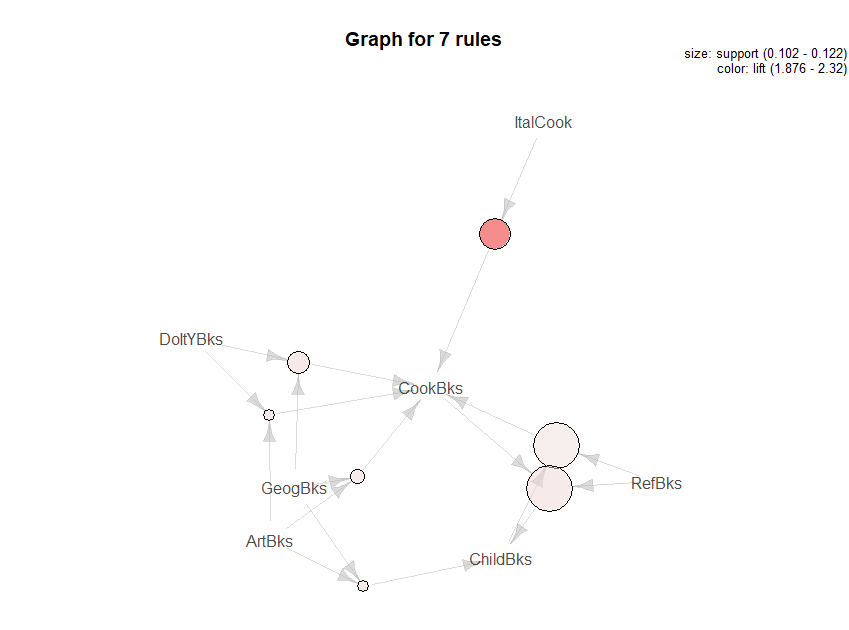
##grouped matrix for association rules

plot(Books\_rules2,method="grouped")



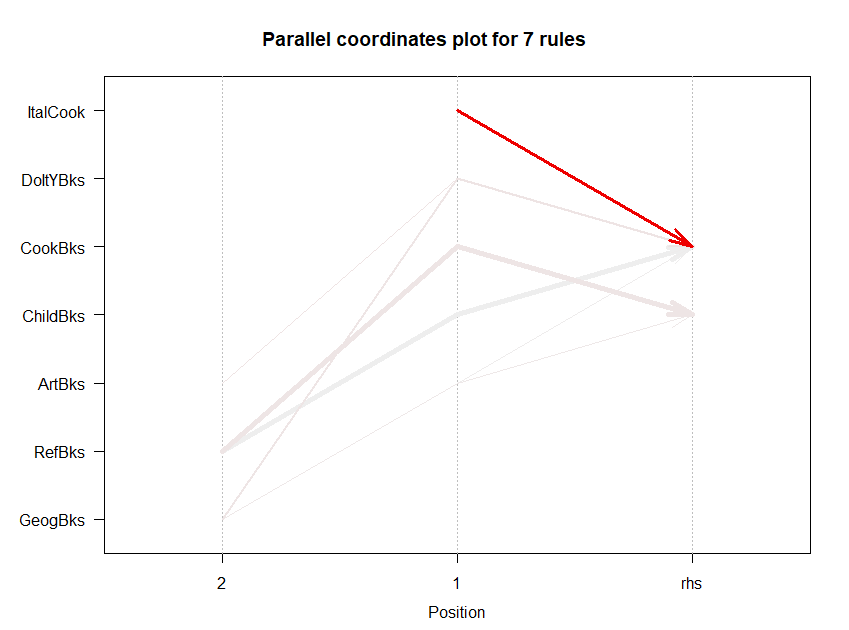
#graph

plot(Books\_rules2 , method="graph", control=list(type="items"))



**#parallel coordinates plot**

plot(Books\_rules2, method="paracoord", control=list(reorder=TRUE))



**#next we can try support=0.12, confidence=0.95**

Books\_rules3 <- apriori(as.matrix(Books,parameter=list(support=0.12,confidence=0.95)))

inspect(Books\_rules3)

inspect(sort(Books\_rules3,by="lift"))

**#visualize the rules**

plotly\_arules(Books\_rules3)

lhs rhs support confidence coverage lift count

[1] {ItalCook} => {CookBks} 0.1135 1.0000000 0.1135 2.320186 227

[2] {DoItYBks,ArtBks} => {CookBks} 0.1015 0.8218623 0.1235 1.906873 203

[3] {DoItYBks,GeogBks} => {CookBks} 0.1085 0.8188679 0.1325 1.899926 217

[4] {CookBks,RefBks} => {ChildBks} 0.1225 0.8032787 0.1525 1.899004 245

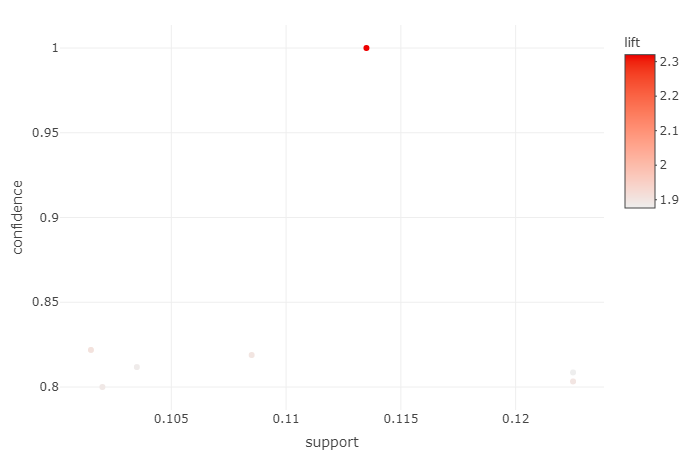
[5] {ArtBks,GeogBks} => {ChildBks} 0.1020 0.8000000 0.1275 1.891253 204

[6] {ArtBks,GeogBks} => {CookBks} 0.1035 0.8117647 0.1275 1.883445 207

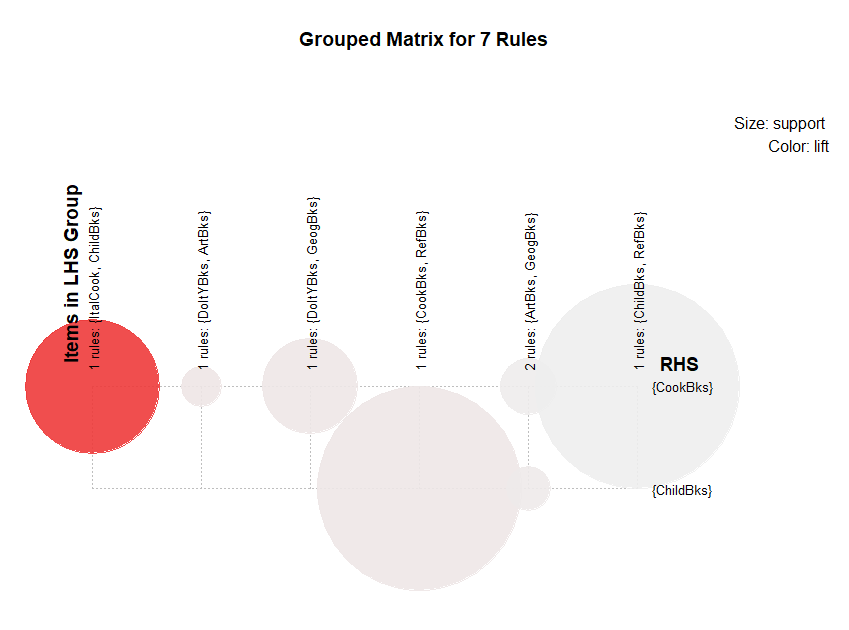
[7] {ChildBks,RefBks} => {CookBks} 0.1225 0.8085809 0.1515 1.876058 245

**# Different Ways of Visualizing Rules**

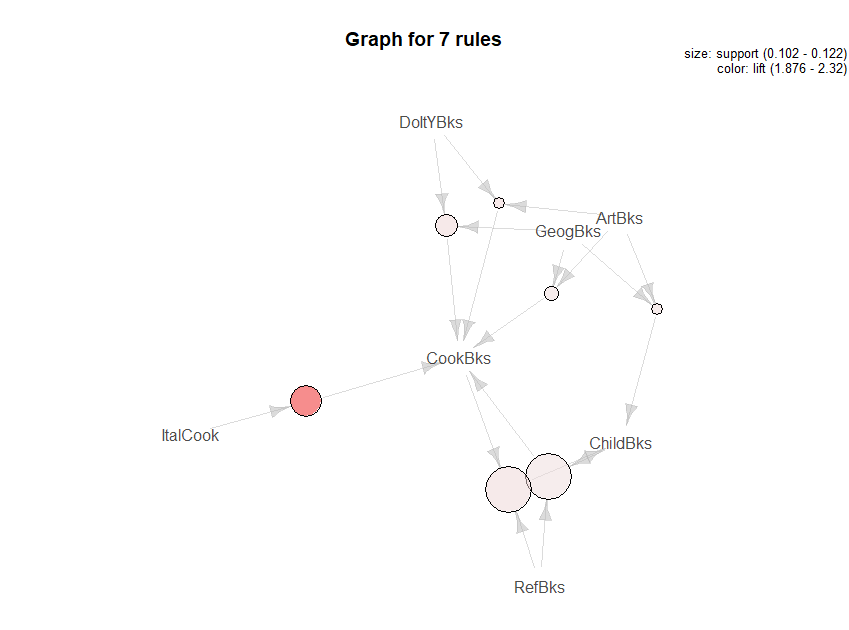
plot(Books\_rules3,method="grouped")



plot(Books\_rules3, method="paracoord", control=list(reorder=TRUE))



plot(Books\_rules3, method="graph", control=list(type="items"))



**#CONCLUSION:**

**Here we have applied three different value in support and confidence, In all case we got**

**This rules {ItalCook} => {CookBks} has highest lift ratio**

**#write the rules as csv files**

write(Books\_rules3, file="Book\_rules.csv",sep=",")

following are the final rules

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **rules** | **support** | **confidence** | **coverag** | **lift** | **count** |
| 1 | {ItalCook} => {CookBks} | 0.1135 | 1 | 0.1135 | 2.320186 | 227 |
| 2 | {ChildBks,RefBks} => {CookBks} | 0.1225 | 0.808580858 | 0.1515 | 1.876058 | 245 |
| 3 | {CookBks,RefBks} => {ChildBks} | 0.1225 | 0.803278689 | 0.1525 | 1.899004 | 245 |
| 4 | {DoItYBks,ArtBks} => {CookBks} | 0.1015 | 0.821862348 | 0.1235 | 1.906873 | 203 |
| 5 | {ArtBks,GeogBks} => {ChildBks} | 0.102 | 0.8 | 0.1275 | 1.891253 | 204 |
| 6 | {ArtBks,GeogBks} => {CookBks} | 0.1035 | 0.811764706 | 0.1275 | 1.883445 | 207 |
| 7 | {DoItYBks,GeogBks} => {CookBks} | 0.1085 | 0.818867925 | 0.1325 | 1.899926 | 217 |