> library(arules)

> library(data.table)

> library(readr)

> library(arulesViz)

Loading required package: grid

Warning message:

package ‘arulesViz’ was built under R version 3.5.2

> library(RColorBrewer)

Warning message:

package ‘RColorBrewer’ was built under R version 3.5.2

> trnsact <- read\_csv("trnsact.csv", col\_names = FALSE,

+ col\_types = cols(X10 = col\_skip(), X11 = col\_skip(),

+ X12 = col\_skip(), X13 = col\_skip(),

+ X14 = col\_skip(), X3 = col\_skip(),

+ X5 = col\_skip(), X8 = col\_skip(),

+ X9 = col\_skip()))

|==============================================================================| 100% 10839 MB

> # change the colum names

> names(trnsact) <- c("SKU","store","trannum","date","stype")

> #randomly choose data record

> transaction <- trnsact[sample(row.names(trnsact), 5000000),]

> #modify the data

> transaction <- subset(transaction,stype == "P" )

> transaction <- transaction[,-c(5)]

> transaction$SKU <- as.factor(transaction$SKU)

> transaction$store <- as.numeric(transaction$store)

> transaction$trannum <- as.numeric(transaction$trannum)

> transaction$date <- as.Date(transaction$date)

> transaction$basket<- paste(transaction$store,transaction$trannum,transaction$date,collapse = NULL, sep = ',')

> transaction$basket<- as.factor(transaction$basket)

> transaction1<- data.frame(transaction$basket,transaction$SKU)

> # output the transaction data and then import as transaction form

> write.csv(transaction1,file="D:/data science/Project2-Association Rules/project 2/transaction1.csv",row.names = F)

> transaction2 <- read.transactions("./transaction1.csv",cols=c(1,2), format="single",rm.duplicates=TRUE, sep=',')

> summary(transaction2)

transactions as itemMatrix in sparse format with

3186329 rows (elements/itemsets/transactions) and

455339 columns (items) and a density of 3.166175e-06

most frequent items:

4628597 4108011 3524026 5528349 3978011 (Other)

21589 12798 10745 9668 8316 4530560

element (itemset/transaction) length distribution:

sizes

1 2 3 4 5 6 7 8 9 10 11

2192480 694418 215230 61884 16586 4374 1021 245 68 14 7

12

2

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

1.000 1.000 1.000 1.442 2.000 12.000

includes extended item information - examples:

labels

1 1000002

2 1000007

3 1000057

includes extended transaction information - examples:

transactionID

1 1002,100,2005-07-16

2 1002,100,2005-08-19

3 1002,1000,2004-08-01

> # head(itemFreq[order(-itemFreq)],10)

> itemFrequencyPlot(transaction1, topN=10, horiz=T)

> # association rules

> rules=apriori(transaction2,parameter=list(support=0.000008,confidence=0.01))

Apriori

Parameter specification:

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

0.01 0.1 1 none FALSE TRUE 5 8e-06 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: 25

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

set transactions ...[455339 item(s), 3186329 transaction(s)] done [4.29s].

sorting and recoding items ... [29149 item(s)] done [0.10s].

creating transaction tree ... done [1.88s].

checking subsets of size 1 2 3 done [3.21s].

writing ... [83 rule(s)] done [1.23s].

creating S4 object ... done [0.92s].

> #exclude redundant rules

> subset.matrix<-is.subset(rules,rules,sparse = FALSE)

> subset.matrix[lower.tri(subset.matrix,diag = T)]<-NA

> redundant<-colSums(subset.matrix,na.rm=T)>=1

> which(redundant)

{2587335,2657335} {6782521,6792521} {6420353,6600353} {4732521,4762521} {8402644,8412644}

2 4 6 8 10

{8520723,8540723} {4722472,4772472} {1543503,1563503} {6500353,6520353} {6560353,6570353}

12 14 16 18 20

{6570353,6580353} {6320353,6340353} {6372521,6402521} {768635,828635} {8132644,8142644}

22 24 26 28 30

{6972521,7222521} {8522644,8542644} {6402521,6412521} {6480353,6490353} {6470353,6480353}

32 34 36 38 40

{8122644,8142644} {4462521,4512521} {6972521,7232521} {4142521,4512521} {6642521,6742521}

42 44 46 48 50

{6642521,6752521} {6300353,6340353} {6742521,6752521} {6560353,6580353} {6032521,6062521}

52 54 56 58 60

{6032521,6072521} {4318362,5938362} {6470353,6490353} {6062521,6072521} {5109905,5749904}

62 64 66 68 70

{5189905,6949904} {3348362,5278362} {3898011,3968011}

73 75 81

> rules.pruned<- rules[!redundant]

> ordered\_rules <- sort(rules.pruned, by="lift")

> inspect(ordered\_rules)

lhs rhs support confidence lift count

[1] {2657335} => {2587335} 8.473701e-06 0.13432836 1685.095840 27

[2] {6782521} => {6792521} 8.473701e-06 0.09782609 1078.567813 27

[3] {6600353} => {6420353} 8.159860e-06 0.09219858 1076.098953 26

[4] {4732521} => {4762521} 8.473701e-06 0.09121622 870.194237 27

[5] {4722472} => {4772472} 8.473701e-06 0.09030100 822.082016 27

[6] {6520353} => {6500353} 8.159860e-06 0.08099688 726.993585 26

[7] {6372521} => {6402521} 1.098443e-05 0.09408602 725.881401 35

[8] {8412644} => {8402644} 8.787542e-06 0.09491525 704.967895 28

[9] {8540723} => {8520723} 8.159860e-06 0.08666667 686.936600 26

[10] {1543503} => {1563503} 9.415224e-06 0.09036145 683.898563 30

[11] {768635} => {828635} 8.159860e-06 0.07142857 573.286974 26

[12] {6412521} => {6402521} 9.101383e-06 0.07323232 564.993403 29

[13] {8132644} => {8142644} 8.159860e-06 0.07027027 547.443032 26

[14] {6570353} => {6560353} 9.729064e-06 0.09198813 516.028958 31

[15] {7222521} => {6972521} 8.473701e-06 0.07068063 501.585157 27

[16] {4462521} => {4512521} 1.223979e-05 0.08904110 497.744256 39

[17] {6742521} => {6752521} 1.318131e-05 0.08123791 493.048975 42

[18] {8542644} => {8522644} 9.729064e-06 0.07730673 478.300361 31

[19] {6972521} => {7232521} 1.067059e-05 0.07572383 468.506870 34

[20] {6320353} => {6340353} 9.101383e-06 0.08605341 465.525439 29

[21] {8142644} => {8122644} 9.415224e-06 0.07334963 463.722348 30

[22] {6480353} => {6470353} 1.098443e-05 0.08293839 384.111908 35

[23] {6480353} => {6490353} 1.004291e-05 0.07582938 374.020689 32

[24] {6642521} => {6752521} 9.415224e-06 0.06097561 370.073055 30

[25] {4142521} => {4512521} 9.415224e-06 0.06355932 355.299844 30

[26] {6032521} => {6072521} 1.537820e-05 0.07596899 354.410253 49

[27] {6642521} => {6742521} 8.787542e-06 0.05691057 350.746222 28

[28] {6570353} => {6580353} 8.787542e-06 0.08308605 344.712893 28

[29] {6062521} => {6072521} 1.506436e-05 0.07017544 327.382189 48

[30] {6560353} => {6580353} 1.380899e-05 0.07746479 321.391019 44

[31] {6300353} => {6340353} 9.101383e-06 0.05753968 311.273954 29

[32] {6032521} => {6062521} 1.286747e-05 0.06356589 296.113806 41

[33] {4318362} => {5938362} 1.192595e-05 0.05928237 289.713403 38

[34] {6490353} => {6470353} 1.098443e-05 0.05417957 250.921401 35

[35] {3348362} => {5278362} 2.699031e-05 0.07078189 155.540964 86

[36] {5109905} => {5749904} 9.415224e-06 0.03452244 84.941969 30

[37] {5189905} => {6949904} 9.101383e-06 0.02543860 47.736006 29

[38] {3968011} => {3898011} 1.694740e-05 0.01322557 9.725598 54

[39] {3690654} => {3968011} 1.192595e-05 0.01138747 8.886661 38

[40] {3898011} => {3978011} 1.851661e-05 0.01361643 5.217224 59

[41] {5618966} => {4628597} 1.035675e-05 0.03210117 4.737824 33

[42] {8798636} => {5528349} 9.101383e-06 0.01080477 3.560979 29

[43] {3898011} => {3524026} 1.631972e-05 0.01200092 3.558761 52

[44] {264715} => {4628597} 1.255363e-05 0.01169933 1.726708 40

[45] {4112626} => {4628597} 8.473701e-06 0.01000000 1.475904 27

> #rules visualization

> plot(ordered\_rules[1:20],

+ control=list(jitter=2,col=rev(brewer.pal(9, "Greens")[4:9])),

+ shading = 'lift')