G <- read.csv(file.choose())

> View(G)

> summary(G)

citrus.fruit semi.finished.bread margarine ready.soups

Length:15295 Length:15295 Length:15295 Length:15295

Class :character Class :character Class :character Class :character

Mode :character Mode :character Mode :character Mode :character

> describe(G)

vars n mean sd median trimmed mad min max range

citrus.fruit\* 1 15295 89.28 51.86 99 89.78 65.23 1 167 166

semi.finished.bread\* 2 15295 70.00 57.72 68 67.43 84.51 1 162 161

margarine\* 3 15295 56.33 59.54 29 50.24 41.51 1 164 163

ready.soups\* 4 15295 41.56 55.00 1 32.78 0.00 1 162 161

skew kurtosis se

citrus.fruit\* -0.12 -1.31 0.42

semi.finished.bread\* 0.15 -1.48 0.47

margarine\* 0.52 -1.32 0.48

ready.soups\* 0.98 -0.61 0.44

> sum(is.na(G))

[1] 0

> grocerie\_rules <- apriori(groceries,parameter = list(support =0.005,confidence=0.2,minlen = 3))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen

0.2 0.1 1 none FALSE TRUE 5 0.005 3 10

target ext

rules TRUE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 76

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

set transactions ...[655 item(s), 15295 transaction(s)] done [0.01s].

sorting and recoding items ... [168 item(s)] done [0.00s].

creating transaction tree ... done [0.01s].

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

writing ... [116 rule(s)] done [0.00s].

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

Warning message:

Column(s) 1, 2, 3, 4 not logical or factor. Applying default discretization (see '? discretizeDF').

> grocerie\_rules

set of 116 rules

> grocerie\_rules <- apriori(G,parameter = list(support =0.005,confidence=0.2,minlen = 3))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen

0.2 0.1 1 none FALSE TRUE 5 0.005 3 10

target ext

rules TRUE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 76

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

set transactions ...[655 item(s), 15295 transaction(s)] done [0.01s].

sorting and recoding items ... [168 item(s)] done [0.00s].

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

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

writing ... [116 rule(s)] done [0.00s].

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

Warning message:

Column(s) 1, 2, 3, 4 not logical or factor. Applying default discretization (see '? discretizeDF').

> grocerie\_rules

set of 116 rules

> grocerie\_rules

set of 116 rules

> inspect(head(sort(grocerie\_rules,by="lift")))

lhs rhs support confidence coverage lift count

[1] {citrus.fruit=shopping bags,

margarine=} => {semi.finished.bread=} 0.01765283 1.0000000 0.01765283 4.388809 270

[2] {citrus.fruit=shopping bags,

ready.soups=} => {semi.finished.bread=} 0.01765283 1.0000000 0.01765283 4.388809 270

[3] {citrus.fruit=shopping bags,

margarine=,

ready.soups=} => {semi.finished.bread=} 0.01765283 1.0000000 0.01765283 4.388809 270

[4] {citrus.fruit=newspapers,

margarine=} => {semi.finished.bread=} 0.01203008 0.8932039 0.01346845 3.920101 184

[5] {citrus.fruit=newspapers,

ready.soups=} => {semi.finished.bread=} 0.01203008 0.8932039 0.01346845 3.920101 184

[6] {citrus.fruit=newspapers,

margarine=,

ready.soups=} => {semi.finished.bread=} 0.01203008 0.8932039 0.01346845 3.920101 184

> inspect(head(sort(grocerie\_rules,by="confidence")))

lhs rhs support confidence coverage lift count

[1] {citrus.fruit=hygiene articles,

margarine=} => {ready.soups=} 0.005230467 1 0.005230467 1.861385 80

[2] {citrus.fruit=salty snack,

margarine=} => {ready.soups=} 0.005557372 1 0.005557372 1.861385 85

[3] {semi.finished.bread=napkins,

margarine=} => {ready.soups=} 0.005099706 1 0.005099706 1.861385 78

[4] {citrus.fruit=chocolate,

margarine=} => {ready.soups=} 0.007191893 1 0.007191893 1.861385 110

[5] {citrus.fruit=napkins,

semi.finished.bread=} => {margarine=} 0.005230467 1 0.005230467 2.501226 80

[6] {citrus.fruit=napkins,

semi.finished.bread=} => {ready.soups=} 0.005230467 1 0.005230467 1.861385 80

> inspect(head(sort(grocerie\_rules,by="support")))

lhs rhs support

[1] {semi.finished.bread=,margarine=} => {ready.soups=} 0.22785224

[2] {semi.finished.bread=,ready.soups=} => {margarine=} 0.22785224

[3] {margarine=,ready.soups=} => {semi.finished.bread=} 0.22785224

[4] {citrus.fruit=soda,margarine=} => {ready.soups=} 0.02451782

[5] {citrus.fruit=soda,ready.soups=} => {margarine=} 0.02451782

[6] {citrus.fruit=canned beer,margarine=} => {ready.soups=} 0.02262177

confidence coverage lift count

[1] 1.0000000 0.22785224 1.861385 3485

[2] 1.0000000 0.22785224 2.501226 3485

[3] 0.5699101 0.39980386 2.501226 3485

[4] 1.0000000 0.02451782 1.861385 375

[5] 0.8012821 0.03059823 2.004188 375

[6] 1.0000000 0.02262177 1.861385 346

> inspect(head(sort(grocerie\_rules,by=c("count","lift")))) #max count = 228

lhs rhs support

[1] {semi.finished.bread=,ready.soups=} => {margarine=} 0.22785224

[2] {margarine=,ready.soups=} => {semi.finished.bread=} 0.22785224

[3] {semi.finished.bread=,margarine=} => {ready.soups=} 0.22785224

[4] {citrus.fruit=soda,ready.soups=} => {margarine=} 0.02451782

[5] {citrus.fruit=soda,margarine=} => {ready.soups=} 0.02451782

[6] {citrus.fruit=canned beer,ready.soups=} => {margarine=} 0.02262177

confidence coverage lift count

[1] 1.0000000 0.22785224 2.501226 3485

[2] 0.5699101 0.39980386 2.501226 3485

[3] 1.0000000 0.22785224 1.861385 3485

[4] 0.8012821 0.03059823 2.004188 375

[5] 1.0000000 0.02451782 1.861385 375

[6] 0.9226667 0.02451782 2.307798 346

> head(quality(grocerie\_rules))

support confidence coverage lift count

1 0.005230467 1.0000000 0.005230467 1.861385 80

2 0.005230467 0.8888889 0.005884276 2.223312 80

3 0.005557372 1.0000000 0.005557372 1.861385 85

4 0.005557372 0.8762887 0.006341942 2.191796 85

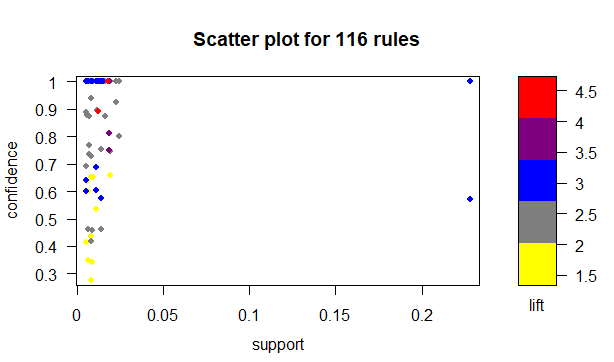
5 0.005099706 1.0000000 0.005099706 1.861385 78

6 0.005099706 0.6902655 0.007388035 1.726510 78

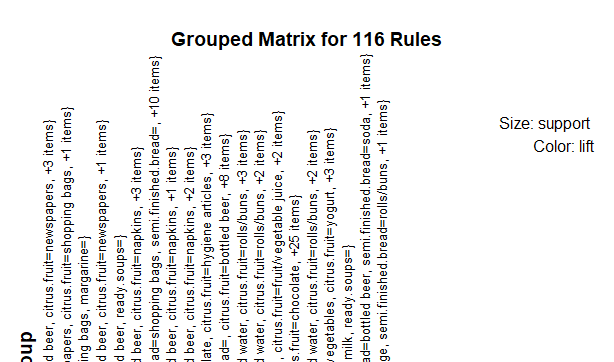
> #plot methods matrix , mosaic , doubledecker , graph , paracoord , scatterplot , groupedmatrix , two-key plot , matrix3D ,iplots

> #visualisation

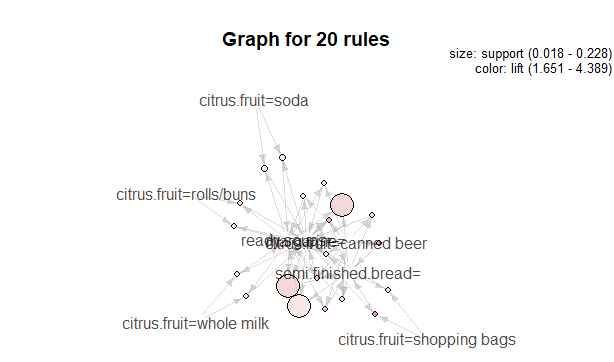
> plot(grocerie\_rules,method = "scatterplot",jitter=0,col=colfunction(5))

>

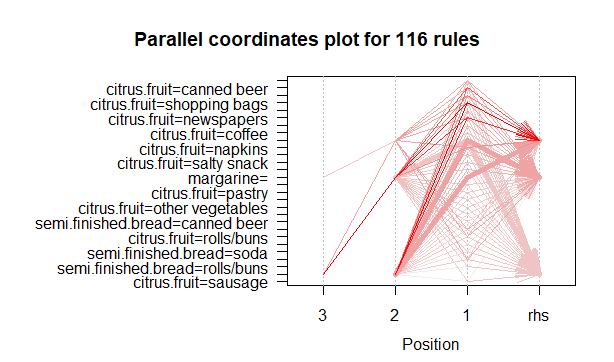
> plot(grocerie\_rules,method = "grouped matrix",col=colfunction(5))



> plot(head(sort(grocerie\_rules,by="support"),n=20),method="graph")



> plot(grocerie\_rules,method = "paracoord")



plot(grocerie\_rules,method = "matrix")

Itemsets in Antecedent (LHS)

[1] "{citrus.fruit=shopping bags,margarine=,ready.soups=}"

[2] "{citrus.fruit=newspapers,margarine=,ready.soups=}"

[3] "{citrus.fruit=canned beer,margarine=,ready.soups=}"

[4] "{citrus.fruit=shopping bags,ready.soups=}"

[5] "{citrus.fruit=newspapers,ready.soups=}"

[6] "{citrus.fruit=shopping bags,margarine=}"

[7] "{citrus.fruit=bottled beer,margarine=,ready.soups=}"

[8] "{citrus.fruit=newspapers,margarine=}"

[9] "{citrus.fruit=napkins,margarine=,ready.soups=}"

[10] "{citrus.fruit=canned beer,ready.soups=}"

[11] "{citrus.fruit=canned beer,margarine=}"

[12] "{citrus.fruit=soda,margarine=,ready.soups=}"

[13] "{semi.finished.bread=shopping bags,ready.soups=}"

[14] "{semi.finished.bread=,ready.soups=}"

[15] "{margarine=,ready.soups=}"

[16] "{citrus.fruit=napkins,semi.finished.bread=,ready.soups=}"

[17] "{citrus.fruit=newspapers,semi.finished.bread=,ready.soups=}"

[18] "{citrus.fruit=shopping bags,semi.finished.bread=,ready.soups=}"

[19] "{citrus.fruit=bottled beer,semi.finished.bread=,ready.soups=}"

[20] "{citrus.fruit=bottled water,semi.finished.bread=,ready.soups=}"

[21] "{citrus.fruit=canned beer,semi.finished.bread=,ready.soups=}"

[22] "{citrus.fruit=soda,semi.finished.bread=,ready.soups=}"

[23] "{citrus.fruit=rolls/buns,semi.finished.bread=,ready.soups=}"

[24] "{citrus.fruit=whole milk,semi.finished.bread=,ready.soups=}"

[25] "{citrus.fruit=napkins,ready.soups=}"

[26] "{citrus.fruit=bottled beer,margarine=}"

[27] "{citrus.fruit=bottled beer,ready.soups=}"

[28] "{citrus.fruit=napkins,margarine=}"

[29] "{semi.finished.bread=newspapers,ready.soups=}"

[30] "{citrus.fruit=hygiene articles,ready.soups=}"

[31] "{citrus.fruit=soda,margarine=}"

[32] "{citrus.fruit=salty snack,ready.soups=}"

[33] "{citrus.fruit=chocolate,ready.soups=}"

[34] "{citrus.fruit=napkins,semi.finished.bread=}"

[35] "{citrus.fruit=newspapers,semi.finished.bread=}"

[36] "{citrus.fruit=shopping bags,semi.finished.bread=}"

[37] "{citrus.fruit=bottled beer,semi.finished.bread=}"

[38] "{citrus.fruit=bottled water,semi.finished.bread=}"

[39] "{citrus.fruit=canned beer,semi.finished.bread=}"

[40] "{citrus.fruit=soda,semi.finished.bread=}"

[41] "{citrus.fruit=rolls/buns,semi.finished.bread=}"

[42] "{citrus.fruit=whole milk,semi.finished.bread=}"

[43] "{citrus.fruit=bottled water,margarine=,ready.soups=}"

[44] "{citrus.fruit=rolls/buns,margarine=,ready.soups=}"

[45] "{citrus.fruit=soda,ready.soups=}"

[46] "{citrus.fruit=bottled water,margarine=}"

[47] "{citrus.fruit=rolls/buns,margarine=}"

[48] "{citrus.fruit=coffee,ready.soups=}"

[49] "{citrus.fruit=hygiene articles,margarine=}"

[50] "{citrus.fruit=salty snack,margarine=}"

[51] "{semi.finished.bread=napkins,margarine=}"

[52] "{citrus.fruit=chocolate,margarine=}"

[53] "{semi.finished.bread=canned beer,margarine=}"

[54] "{citrus.fruit=coffee,margarine=}"

[55] "{citrus.fruit=fruit/vegetable juice,margarine=}"

[56] "{semi.finished.bread=newspapers,margarine=}"

[57] "{semi.finished.bread=bottled beer,margarine=}"

[58] "{semi.finished.bread=shopping bags,margarine=}"

[59] "{citrus.fruit=pastry,margarine=}"

[60] "{citrus.fruit=yogurt,margarine=}"

[61] "{semi.finished.bread=soda,margarine=}"

[62] "{semi.finished.bread=rolls/buns,margarine=}"

[63] "{citrus.fruit=other vegetables,margarine=}"

[64] "{citrus.fruit=sausage,margarine=}"

[65] "{semi.finished.bread=,margarine=}"

[66] "{citrus.fruit=napkins,semi.finished.bread=,margarine=}"

[67] "{citrus.fruit=newspapers,semi.finished.bread=,margarine=}"

[68] "{citrus.fruit=shopping bags,semi.finished.bread=,margarine=}"

[69] "{citrus.fruit=bottled beer,semi.finished.bread=,margarine=}"

[70] "{citrus.fruit=bottled water,semi.finished.bread=,margarine=}"

[71] "{citrus.fruit=canned beer,semi.finished.bread=,margarine=}"

[72] "{citrus.fruit=soda,semi.finished.bread=,margarine=}"

[73] "{citrus.fruit=rolls/buns,semi.finished.bread=,margarine=}"

[74] "{citrus.fruit=whole milk,semi.finished.bread=,margarine=}"

[75] "{citrus.fruit=whole milk,margarine=}"

[76] "{citrus.fruit=whole milk,margarine=,ready.soups=}"

[77] "{citrus.fruit=fruit/vegetable juice,ready.soups=}"

[78] "{citrus.fruit=pastry,ready.soups=}"

[79] "{semi.finished.bread=napkins,ready.soups=}"

[80] "{citrus.fruit=bottled water,ready.soups=}"

[81] "{citrus.fruit=rolls/buns,ready.soups=}"

[82] "{citrus.fruit=other vegetables,ready.soups=}"

[83] "{citrus.fruit=yogurt,ready.soups=}"

[84] "{semi.finished.bread=canned beer,ready.soups=}"

[85] "{semi.finished.bread=bottled beer,ready.soups=}"

[86] "{citrus.fruit=whole milk,ready.soups=}"

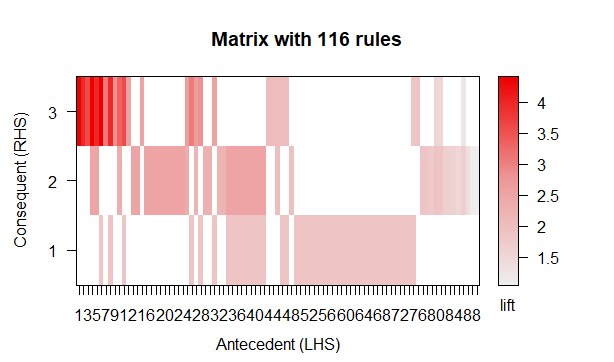
[87] "{semi.finished.bread=soda,ready.soups=}"

[88] "{semi.finished.bread=rolls/buns,ready.soups=}"

[89] "{citrus.fruit=sausage,ready.soups=}"

Itemsets in Consequent (RHS)

[1] "{ready.soups=}" "{margarine=}" "{semi.finished.bread=}"



# using different support and confidence

> grocerie\_rules2 <- apriori(groceries,parameter = list(support=0.005,confidence = 0.5,minlen=2))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen

0.5 0.1 1 none FALSE TRUE 5 0.005 2 10

target ext

rules TRUE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 76

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

set transactions ...[655 item(s), 15295 transaction(s)] done [0.01s].

sorting and recoding items ... [168 item(s)] done [0.00s].

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

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

writing ... [169 rule(s)] done [0.00s].

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

Warning message:

Column(s) 1, 2, 3, 4 not logical or factor. Applying default discretization (see '? discretizeDF').

> grocerie\_rules2

set of 169 rules

>> inspect(head(sort(grocerie\_rules2,by="lift")))

lhs rhs support confidence coverage lift count

[1] {semi.finished.bread=sausage} => {citrus.fruit=frankfurter} 0.006472703 1.0000000 0.006472703 26.370690 99

[2] {margarine=pip fruit} => {semi.finished.bread=tropical fruit} 0.005165087 0.5337838 0.009676365 22.741568 79

[3] {ready.soups=whole milk} => {margarine=other vegetables} 0.012160837 0.5454545 0.022294868 18.917749 186

[4] {citrus.fruit=shopping bags} => {semi.finished.bread=} 0.017652828 1.0000000 0.017652828 4.388809 270

[5] {citrus.fruit=shopping bags,

margarine=} => {semi.finished.bread=} 0.017652828 1.0000000 0.017652828 4.388809 270

[6] {citrus.fruit=shopping bags,

ready.soups=} => {semi.finished.bread=} 0.017652828 1.0000000 0.017652828 4.388809 270

> inspect(head(sort(grocerie\_rules2,by="confidence")))

lhs rhs support confidence coverage lift count

[1] {citrus.fruit=hygiene articles} => {ready.soups=} 0.005884276 1 0.005884276 1.861385 90

[2] {semi.finished.bread=sausage} => {citrus.fruit=frankfurter} 0.006472703 1 0.006472703 26.370690 99

[3] {citrus.fruit=newspapers} => {margarine=} 0.013468454 1 0.013468454 2.501226 206

[4] {citrus.fruit=newspapers} => {ready.soups=} 0.013468454 1 0.013468454 1.861385 206

[5] {semi.finished.bread=shopping bags} => {margarine=} 0.014972213 1 0.014972213 2.501226 229

[6] {semi.finished.bread=shopping bags} => {ready.soups=} 0.014972213 1 0.014972213 1.861385 229

> inspect(head(sort(grocerie\_rules2,by="support")))

lhs rhs support

[1] {margarine=} => {ready.soups=} 0.3998039

[2] {ready.soups=} => {margarine=} 0.3998039

[3] {semi.finished.bread=} => {margarine=} 0.2278522

[4] {margarine=} => {semi.finished.bread=} 0.2278522

[5] {semi.finished.bread=} => {ready.soups=} 0.2278522

[6] {semi.finished.bread=,margarine=} => {ready.soups=} 0.2278522

confidence coverage lift count

[1] 1.0000000 0.3998039 1.861385 6115

[2] 0.7441889 0.5372344 1.861385 6115

[3] 1.0000000 0.2278522 2.501226 3485

[4] 0.5699101 0.3998039 2.501226 3485

[5] 1.0000000 0.2278522 1.861385 3485

[6] 1.0000000 0.2278522 1.861385 3485

> inspect(head(sort(grocerie\_rules2,by=c("count","lift")))) #max count = 219

lhs rhs support

[1] {ready.soups=} => {margarine=} 0.3998039

[2] {margarine=} => {ready.soups=} 0.3998039

[3] {semi.finished.bread=} => {margarine=} 0.2278522

[4] {margarine=} => {semi.finished.bread=} 0.2278522

[5] {semi.finished.bread=,ready.soups=} => {margarine=} 0.2278522

[6] {margarine=,ready.soups=} => {semi.finished.bread=} 0.2278522

confidence coverage lift count

[1] 0.7441889 0.5372344 1.861385 6115

[2] 1.0000000 0.3998039 1.861385 6115

[3] 1.0000000 0.2278522 2.501226 3485

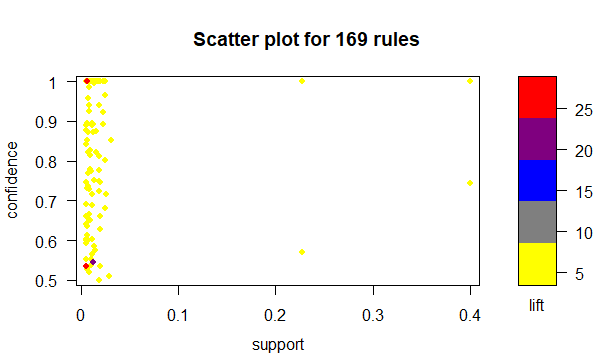
[4] 0.5699101 0.3998039 2.501226 3485

[5] 1.0000000 0.2278522 2.501226 3485

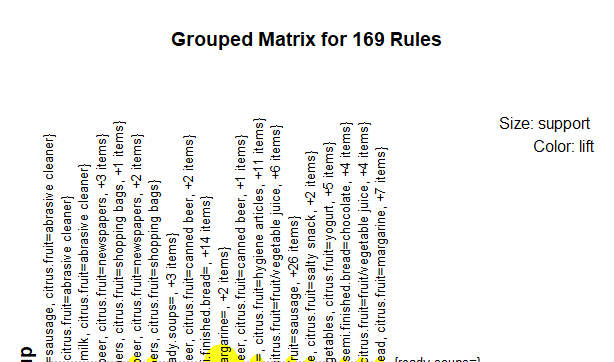
[6] 0.5699101 0.3998039 2.501226 3485

#visualisation

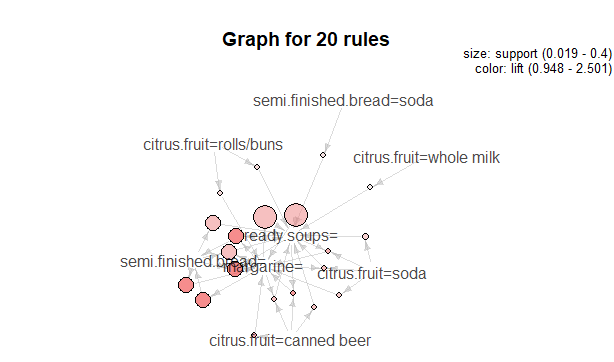
> plot(grocerie\_rules2,method = "scatterplot",jitter=0,col=colfunction(5))

>

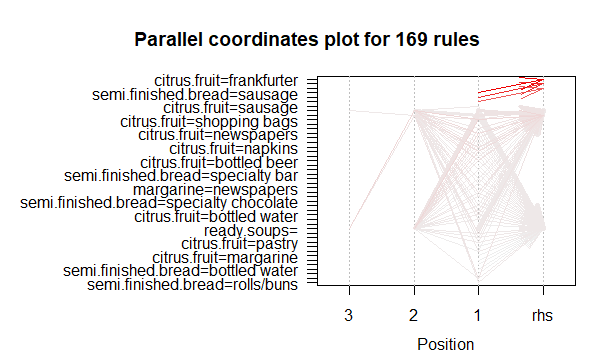
> plot(grocerie\_rules2,method = "grouped matrix",col=colfunction(5))



> plot(head(sort(grocerie\_rules2,by="support"),n=20),method="graph")



> plot(grocerie\_rules2,method = "paracoord")



plot(grocerie\_rules2,method = "matrix")

Itemsets in Antecedent (LHS)

[1] "{semi.finished.bread=sausage}"

[2] "{margarine=pip fruit}"

[3] "{ready.soups=whole milk}"

[4] "{citrus.fruit=shopping bags,margarine=,ready.soups=}"

[5] "{citrus.fruit=newspapers,margarine=,ready.soups=}"

[6] "{citrus.fruit=canned beer,margarine=,ready.soups=}"

[7] "{citrus.fruit=shopping bags,ready.soups=}"

[8] "{citrus.fruit=newspapers,ready.soups=}"

[9] "{citrus.fruit=shopping bags,margarine=}"

[10] "{citrus.fruit=bottled beer,margarine=,ready.soups=}"

[11] "{citrus.fruit=shopping bags}"

[12] "{citrus.fruit=newspapers,margarine=}"

[13] "{citrus.fruit=napkins,margarine=,ready.soups=}"

[14] "{citrus.fruit=canned beer,ready.soups=}"

[15] "{citrus.fruit=newspapers}"

[16] "{citrus.fruit=canned beer,margarine=}"

[17] "{citrus.fruit=soda,margarine=,ready.soups=}"

[18] "{semi.finished.bread=shopping bags,ready.soups=}"

[19] "{semi.finished.bread=,ready.soups=}"

[20] "{margarine=,ready.soups=}"

[21] "{citrus.fruit=napkins,semi.finished.bread=,ready.soups=}"

[22] "{citrus.fruit=newspapers,semi.finished.bread=,ready.soups=}"

[23] "{citrus.fruit=shopping bags,semi.finished.bread=,ready.soups=}"

[24] "{citrus.fruit=bottled beer,semi.finished.bread=,ready.soups=}"

[25] "{citrus.fruit=bottled water,semi.finished.bread=,ready.soups=}"

[26] "{citrus.fruit=canned beer,semi.finished.bread=,ready.soups=}"

[27] "{citrus.fruit=soda,semi.finished.bread=,ready.soups=}"

[28] "{citrus.fruit=rolls/buns,semi.finished.bread=,ready.soups=}"

[29] "{citrus.fruit=whole milk,semi.finished.bread=,ready.soups=}"

[30] "{citrus.fruit=napkins,ready.soups=}"

[31] "{citrus.fruit=bottled beer,margarine=}"

[32] "{citrus.fruit=bottled beer,ready.soups=}"

[33] "{citrus.fruit=canned beer}"

[34] "{citrus.fruit=napkins,margarine=}"

[35] "{citrus.fruit=napkins}"

[36] "{semi.finished.bread=newspapers,ready.soups=}"

[37] "{citrus.fruit=hygiene articles,ready.soups=}"

[38] "{citrus.fruit=soda,margarine=}"

[39] "{citrus.fruit=salty snack,ready.soups=}"

[40] "{citrus.fruit=chocolate,ready.soups=}"

[41] "{semi.finished.bread=shopping bags}"

[42] "{semi.finished.bread=}"

[43] "{margarine=}"

[44] "{citrus.fruit=napkins,semi.finished.bread=}"

[45] "{citrus.fruit=newspapers,semi.finished.bread=}"

[46] "{citrus.fruit=shopping bags,semi.finished.bread=}"

[47] "{citrus.fruit=bottled beer,semi.finished.bread=}"

[48] "{citrus.fruit=bottled water,semi.finished.bread=}"

[49] "{citrus.fruit=canned beer,semi.finished.bread=}"

[50] "{citrus.fruit=soda,semi.finished.bread=}"

[51] "{citrus.fruit=rolls/buns,semi.finished.bread=}"

[52] "{citrus.fruit=whole milk,semi.finished.bread=}"

[53] "{citrus.fruit=bottled beer}"

[54] "{citrus.fruit=hygiene articles}"

[55] "{semi.finished.bread=newspapers}"

[56] "{citrus.fruit=soda,ready.soups=}"

[57] "{citrus.fruit=coffee,ready.soups=}"

[58] "{citrus.fruit=chocolate}"

[59] "{citrus.fruit=bottled water,ready.soups=}"

[60] "{citrus.fruit=rolls/buns,ready.soups=}"

[61] "{ready.soups=}"

[62] "{margarine=shopping bags}"

[63] "{citrus.fruit=hygiene articles,margarine=}"

[64] "{citrus.fruit=salty snack,margarine=}"

[65] "{semi.finished.bread=napkins,margarine=}"

[66] "{citrus.fruit=chocolate,margarine=}"

[67] "{semi.finished.bread=canned beer,margarine=}"

[68] "{citrus.fruit=coffee,margarine=}"

[69] "{citrus.fruit=fruit/vegetable juice,margarine=}"

[70] "{semi.finished.bread=newspapers,margarine=}"

[71] "{semi.finished.bread=bottled beer,margarine=}"

[72] "{semi.finished.bread=shopping bags,margarine=}"

[73] "{citrus.fruit=pastry,margarine=}"

[74] "{citrus.fruit=bottled water,margarine=}"

[75] "{citrus.fruit=yogurt,margarine=}"

[76] "{semi.finished.bread=soda,margarine=}"

[77] "{citrus.fruit=rolls/buns,margarine=}"

[78] "{semi.finished.bread=rolls/buns,margarine=}"

[79] "{citrus.fruit=other vegetables,margarine=}"

[80] "{citrus.fruit=sausage,margarine=}"

[81] "{citrus.fruit=whole milk,margarine=}"

[82] "{semi.finished.bread=,margarine=}"

[83] "{citrus.fruit=napkins,semi.finished.bread=,margarine=}"

[84] "{citrus.fruit=newspapers,semi.finished.bread=,margarine=}"

[85] "{citrus.fruit=shopping bags,semi.finished.bread=,margarine=}"

[86] "{citrus.fruit=bottled beer,semi.finished.bread=,margarine=}"

[87] "{citrus.fruit=bottled water,semi.finished.bread=,margarine=}"

[88] "{citrus.fruit=canned beer,semi.finished.bread=,margarine=}"

[89] "{citrus.fruit=soda,semi.finished.bread=,margarine=}"

[90] "{citrus.fruit=rolls/buns,semi.finished.bread=,margarine=}"

[91] "{citrus.fruit=whole milk,semi.finished.bread=,margarine=}"

[92] "{citrus.fruit=fruit/vegetable juice,ready.soups=}"

[93] "{citrus.fruit=pastry,ready.soups=}"

[94] "{semi.finished.bread=napkins,ready.soups=}"

[95] "{citrus.fruit=salty snack}"

[96] "{semi.finished.bread=napkins}"

[97] "{citrus.fruit=waffles}"

[98] "{citrus.fruit=whole milk,ready.soups=}"

[99] "{citrus.fruit=soda}"

[100] "{semi.finished.bread=specialty bar}"

[101] "{citrus.fruit=other vegetables,ready.soups=}"

[102] "{citrus.fruit=yogurt,ready.soups=}"

[103] "{margarine=newspapers}"

[104] "{semi.finished.bread=canned beer,ready.soups=}"

[105] "{semi.finished.bread=specialty chocolate}"

[106] "{citrus.fruit=fruit/vegetable juice}"

[107] "{semi.finished.bread=bottled beer,ready.soups=}"

[108] "{citrus.fruit=coffee}"

[109] "{citrus.fruit=bottled water}"

[110] "{semi.finished.bread=bottled beer}"

[111] "{semi.finished.bread=canned beer}"

[112] "{semi.finished.bread=chocolate}"

[113] "{semi.finished.bread=soda,ready.soups=}"

[114] "{citrus.fruit=rolls/buns}"

[115] "{citrus.fruit=pastry}"

[116] "{semi.finished.bread=fruit/vegetable juice}"

[117] "{semi.finished.bread=soda}"

[118] "{citrus.fruit=margarine}"

[119] "{margarine=napkins}"

[120] "{semi.finished.bread=coffee}"

[121] "{semi.finished.bread=bottled water}"

[122] "{citrus.fruit=brown bread}"

[123] "{citrus.fruit=whole milk}"

[124] "{semi.finished.bread=rolls/buns}"

Itemsets in Consequent (RHS)

[1] "{ready.soups=}" "{margarine=}"

[3] "{semi.finished.bread=}" "{margarine=other vegetables}"

[5] "{semi.finished.bread=tropical fruit}" "{citrus.fruit=frankfurter}"

>