**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**

**Ans:**

> movies <- read.csv(file.choose())

> View(movies)

> str(movies)

'data.frame': 10 obs. of 15 variables:

$ V1 : Factor w/ 4 levels "Gladiator","Harry Potter1",..: 4 1 3 1 1 1 2 1 1 4

$ V2 : Factor w/ 5 levels "Harry Potter2",..: 3 5 4 5 5 5 1 5 5 2

$ V3 : Factor w/ 5 levels "","Braveheart",..: 4 2 1 5 5 5 1 1 5 3

$ V4 : Factor w/ 2 levels "","Green Mile": 2 1 1 1 1 1 1 1 1 2

$ V5 : Factor w/ 2 levels "","LOTR2": 2 1 1 1 1 1 1 1 1 1

$ Sixth.Sense : int 1 0 0 1 1 1 0 0 1 1

$ Gladiator : int 0 1 0 1 1 1 0 1 1 1

$ LOTR1 : int 1 0 1 0 0 0 0 0 0 0

$ Harry.Potter1: int 1 0 0 0 0 0 1 0 0 0

$ Patriot : int 0 1 0 1 1 1 0 1 1 0

$ LOTR2 : int 1 0 1 0 0 0 0 0 0 0

$ Harry.Potter2: int 0 0 0 0 0 0 1 0 0 0

$ LOTR : int 0 0 0 0 0 0 0 0 0 1

$ Braveheart : int 0 1 0 0 0 0 0 0 0 0

$ Green.Mile : int 1 0 0 0 0 0 0 0 0 1

**converting everything into character format**

> movies[] <- lapply(movies,as.character)

> str(movies)

'data.frame': 10 obs. of 15 variables:

$ V1 : chr "Sixth Sense" "Gladiator" "LOTR1" "Gladiator" ...

$ V2 : chr "LOTR1" "Patriot" "LOTR2" "Patriot" ...

$ V3 : chr "Harry Potter1" "Braveheart" "" "Sixth Sense" ...

$ V4 : chr "Green Mile" "" "" "" ...

$ V5 : chr "LOTR2" "" "" "" ...

$ Sixth.Sense : chr "1" "0" "0" "1" ...

$ Gladiator : chr "0" "1" "0" "1" ...

$ LOTR1 : chr "1" "0" "1" "0" ...

$ Harry.Potter1: chr "1" "0" "0" "0" ...

$ Patriot : chr "0" "1" "0" "1" ...

$ LOTR2 : chr "1" "0" "1" "0" ...

$ Harry.Potter2: chr "0" "0" "0" "0" ...

$ LOTR : chr "0" "0" "0" "0" ...

$ Braveheart : chr "0" "1" "0" "0" ...

$ Green.Mile : chr "1" "0" "0" "0" ...

**Creating a custom fucntion to collapse all the items in a transaction into a single sentence**

>paste\_fun <- function(i){

return (paste(as.character(i),collapse=" "))

}

> movies["new\_col"] <- apply(movies,1,paste\_fun)

> View(movies)

**Now Selecting the new column which contains all items of a transaction in a single sentence**

> x <- Corpus(VectorSource(movies$new\_col))

**Creating a TDM matrix**

> x <- tm\_map(x,stripWhitespace)

**Converting TDM matrix to data frame**

>dtm0\_df <- data.frame(as.matrix(dtm0))

> head(dtm0\_df)

green harry lotr1 lotr2 mile potter1 sense sixth braveheart gladiator patriot potter2 lotr

1 1 1 1 1 1 1 1 1 0 0 0 0 0

2 0 0 0 0 0 0 0 0 1 1 1 0 0

3 0 0 1 1 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 1 1 0 1 1 0 0

5 0 0 0 0 0 0 1 1 0 1 1 0 0

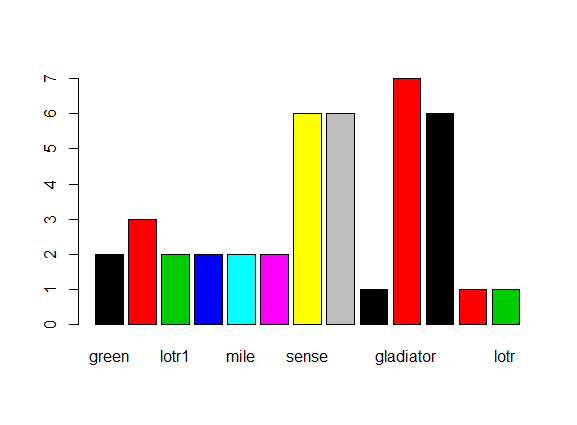
6 0 0 0 0 0 0 1 1 0 1 1 0 0

**Association Rules**

>library(arules)

>library(arulesViz)

>barplot(sapply(dtm0\_df,sum),col=1:10)



**Applying apriori algorithm**

1. **For support=0.002,confidence=0.5,minlen=2**

>rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.002,confidence=0.5,minlen=2))

Apriori

Parameter specification:

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

0.5 0.1 1 none FALSE TRUE 5 0.002 2 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

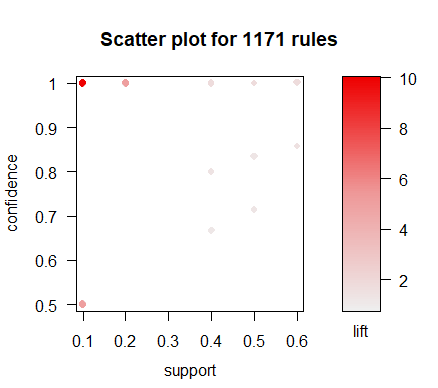
checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**1171 rule(s)] done [0.06s**].

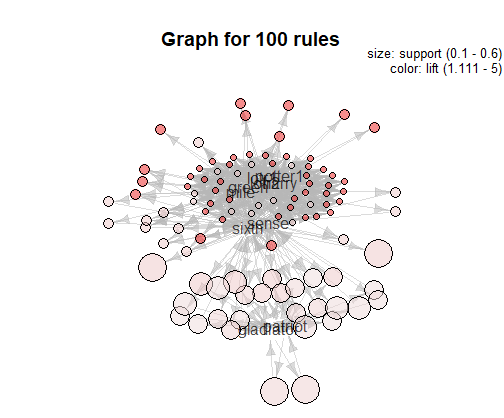
creating S4 object ... done [0.00s].

>inspect(rules)

>plot(rules)



>plot(rules,method = "graph")



**2. For support=0.0045,confidence=0.7,minlen=4**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.0045,confidence=0.7,minlen=4))

Apriori

Parameter specification:

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

0.7 0.1 1 none FALSE TRUE 5 0.0045 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: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**855 rule(s)] done [0.11s**].

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

**3. For support=0.0055,confidence=0.8,minlen=4**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.0055,confidence=0.8,minlen=4))

Apriori

Parameter specification:

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

0.8 0.1 1 none FALSE TRUE 5 0.0055 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: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**855 rule(s)] done [0.00s**].

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

**4. For support=0.0075,confidence=0.35,minlen=5**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.0075,confidence=0.35,minlen=5))

Apriori

Parameter specification:

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

0.35 0.1 1 none FALSE TRUE 5 0.0075 5 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**548 rule(s)] done [0.13s**].

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

**5. For support=0.00555,confidence=0.75,minlen=5**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.00555,confidence=0.75,minlen=5))

Apriori

Parameter specification:

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

0.75 0.1 1 none FALSE TRUE 5 0.00555 5 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**542 rule(s)] done [0.02s].**

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

**6. For support=0.005,confidence=0.5,minlen=6**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.005,confidence=0.5,minlen=6))

Apriori

Parameter specification:

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

0.5 0.1 1 none FALSE TRUE 5 0.005 6 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**238 rule(s)] done [0.03s**].

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

**7. For support=0.005,confidence=0.5,minlen=7**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.005,confidence=0.5,minlen=7))

Apriori

Parameter specification:

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

0.5 0.1 1 none FALSE TRUE 5 0.005 7 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [**64 rule(s)] done [0.03s**].

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

**8. For support=0.006,confidence=0.8,minlen=8**

> rules <- apriori(as.matrix(dtm0\_df),parameter = list(support=0.006,confidence=0.8,minlen=8))

Apriori

Parameter specification:

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

0.8 0.1 1 none FALSE TRUE 5 0.006 8 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 0

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

set transactions ...[13 item(s), 10 transaction(s)] done [0.00s].

sorting and recoding items ... [13 item(s)] done [0.00s].

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

checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].

writing ... [8 rule(s)] done [0.00s].

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

**So From Above 8 cases we observed that,**

**1. Lower the Confidence level Higher the no. of rules.**

**2. Lower the minlen(minimum length), Higher the no. of Rules are getting generated.**

**3. Higher the Support, lower the no. of rules.**

**4. So, we can conclude that these 3 parameters different combinations can generate different rules.**