

Association Rule Mining and Twitter

Twitter API Setup

To access a Twitter API you will need to set up an account and receive a consumerKey, the consumerSecret, the access_Token, and the access_Secret. A popular library and API: “twitterR”.

Once you have your keys, you can set up the API.

```
requestURL='https://api.twitter.com/oauth/request_token'
accessURL='https://api.twitter.com/oauth/access_token'
authURL='https://api.twitter.com/oauth/authorize'

### NOTES: rtweet is another excellent option
## https://mkearney.github.io/blog/2017/06/01/intro-to-rtweet/
### https://rtweet.info/

### Install the needed packages...
#install.packages("twitterR")
#install.packages("ROAuth")
# install.packages("rtweet")
library(arules)

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##      abbreviate, write

library(rtweet)
library(twitterR)

##
## Attaching package: 'twitterR'

## The following object is masked from 'package:rtweet':
##
##      lookup_statuses

library(ROAuth)
library(jsonlite)

##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:rtweet':
##
##      flatten

#install.packages("streamR")
#library(streamR)
#install.packages("rjson")
library(rjson)
```

```

##
## Attaching package: 'rjson'

## The following objects are masked from 'package:jsonlite':
##
##      fromJSON, toJSON

#install.packages("tokenizers")
library(tokenizers)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2      v purrr  0.3.4
## v tibble  3.0.3      v dplyr  1.0.2
## v tidyr   1.1.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x tidyr::expand()    masks Matrix::expand()
## x dplyr::filter()    masks stats::filter()
## x purrr::flatten()   masks jsonlite::flatten(), rtweet::flatten()
## x rjson::fromJSON()  masks jsonlite::fromJSON()
## x dplyr::id()        masks twitterR::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::location()  masks twitterR::location()
## x tidyr::pack()      masks Matrix::pack()
## x dplyr::recode()    masks arules::recode()
## x rjson::toJSON()    masks jsonlite::toJSON()
## x tidyr::unpack()    masks Matrix::unpack()

library(plyr)

## -----

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##      arrange, count, desc, failwith, id, mutate, rename, summarise,
##      summarize

## The following object is masked from 'package:purrr':
##
##      compact

## The following object is masked from 'package:twitterR':
##
##      id

library(dplyr)
library(ggplot2)
#install.packages("syuzhet") ## sentiment analysis

```

```
## Loading required package: grid
## Registered S3 method overwritten by 'seriation':
##   method      from
##   reorder.hclust gclus
```

```
##### Read in the tweet transactions
TweetTrans <- read.transactions(TransactionTweetsFile,
                                rm.duplicates = FALSE,
                                format = "basket",
                                sep=",",
                                ## cols =
                                )
inspect(head(TweetTrans))
```

```
##      items
## [1] {<U+304C>,
##      <U+3053><U+3068>,
##      <U+3053><U+3068><U+306B>,
##      <U+3053><U+306E>,
##      <U+3055>,
##      <U+305D><U+306E>,
##      <U+305F>,
##      <U+3063>,
##      <U+3067>,
##      <U+3068>,
##      <U+3068><U+304D>,
##      <U+306A>,
##      <U+306B>,
##      <U+306E>,
##      <U+306F>,
##      <U+308C>,
##      <U+3092>,
##      <U+30B2><U+30FC><U+30E0>,
##      <U+30CB><U+30E5><U+30FC><U+30B9>,
##      <U+30E9><U+30A4><U+30B9><U+30DC><U+30A6><U+30EB>,
##      <U+50D5>,
##      <U+53F2><U+4E0A>,
##      <U+5468><U+5E74>,
##      <U+5909><U+3048><U+308B>,
##      <U+59FF>,
##      <U+5E74>,
##      <U+601D>,
##      <U+6700><U+9AD8>,
##      <U+89E6><U+308C>,
##      <U+9078><U+51FA>,
##      <U+982D>,
##      aqp3i87iv1,
##      https,
##      pcmpevm2pp,
##      t.co}
## [2] {2021,
##      49ers,
##      around,
##      domclare,
##      garoppolo,
##      https,
##      jimmy,
##      keeping,
##      mikegarafolo,
```

```

##      network,
##      nfl,
##      per,
##      rt,
##      season,
##      szxdz,
##      t.co}
## [3] {cowhercbs,
##      embodied,
##      every,
##      impacted,
##      life,
##      lived,
##      lost,
##      matriarch,
##      nfl,
##      pat,
##      pittsburgh,
##      rooney,
##      rt,
##      spirit}
## [4] {7u,
##      bowl,
##      champions,
##      congratulations,
##      ducks,
##      flag,
##      https,
##      maui,
##      nfl,
##      nflflagbowl,
##      t.co,
##      tisbo7vynr}
## [5] {56,
##      a9ultydtmi,
##      calvin,
##      good,
##      https,
##      johnson,
##      matthew,
##      passes,
##      remember,
##      rt,
##      sharpfootball,
##      stafford,
##      t.co,
##      td,
##      times}
## [6] {barrysanders,
##      done,
##      family,
##      fans,
##      just,
##      mattstafford,

```

```
##      note,
##      quick,
##      rt,
##      send,
##      team,
##      thanking,
##      wanted}
```

```
## See the words that occur the most
Sample_Trans <- sample(TweetTrans, 50)
summary(Sample_Trans)
```

```
## transactions as itemMatrix in sparse format with
## 50 rows (elements/itemsets/transactions) and
## 698 columns (items) and a density of 0.0169914
##
## most frequent items:
##      nfl      https      rt      t.co matthew (Other)
##      26       24       24       24       8       487
##
## element (itemset/transaction) length distribution:
## sizes
## 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18
## 1 2 1 1 4 2 4 2 8 6 7 3 4 2 3
##
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      3.00   9.00   12.50   11.86   14.00   18.00
##
## includes extended item information - examples:
##      labels
## 1 __andrew_powell
## 2  _codystewart
## 3    _tjwatt
```

```
## Read the transactions data into a dataframe
TweetDF <- read.csv(TransactionTweetsFile, header = FALSE, sep = ",")
head(TweetDF)
```

```
##
##      V1      V2      V3
## 1 <U+30E9><U+30A4><U+30B9><U+30DC><U+30A6><U+30EB> <U+3068> <U+5E74>
## 2      rt      domclare 49ers
## 3      rt      cowhercbs nfl
## 4      congratulations nfl flag
## 5      rt sharpfootball 56
## 6      rt barrysanders just
##      V4      V5      V6      V7
## 1 <U+306E> <U+53F2><U+4E0A> <U+6700><U+9AD8> <U+306E>
## 2 keeping      jimmy      garoppolo around
## 3      lost      matriarch      pat rooney
## 4      bowl      7u      champions maui
## 5      td      passes      matthew stafford
## 6      wanted      send      quick note
##      V8      V9      V10      V11
## 1 <U+30B2><U+30FC><U+30E0>      https      t.co pcmpevm2pp
## 2      2021      nfl      season      per
## 3      embodied      spirit pittsburgh      lived
```

```

## 4          ducks nflflagbowl      https      t.co
## 5          calvin      johnson      remember      good
## 6          thanking mattstafford      family      done
##          V12      V13
## 1 <U+30E9><U+30A4><U+30B9><U+30DC><U+30A6><U+30EB> <U+304C>
## 2          mikegarafolo      nfl
## 3          life impacted
## 4          tisbo7vynr
## 5          times      https
## 6          team      fans
##          V14          V15          V16          V17
## 1 <U+305D><U+306E> <U+59FF> <U+3092> <U+5909><U+3048><U+308B>
## 2          network      https      t.co          szxdz
## 3          every
## 4
## 5          t.co a9ultydtmi
## 6
##          V18          V19          V20          V21          V22
## 1 <U+3053><U+3068><U+306B> <U+306A> <U+3063> <U+305F> <U+3053><U+306E>
## 2
## 3
## 4
## 5
## 6
##          V23          V24          V25          V26
## 1 <U+30CB><U+30E5><U+30FC><U+30B9> <U+306B> <U+89E6><U+308C> <U+305F>
## 2
## 3
## 4
## 5
## 6
##          V27          V28          V29          V30          V31
## 1 <U+3068><U+304D> <U+50D5> <U+306F> <U+5468><U+5E74> <U+3067>
## 2
## 3
## 4
## 5
## 6
##          V32          V33          V34          V35          V36
## 1 <U+9078><U+51FA> <U+3055> <U+308C> <U+305F> <U+53F2><U+4E0A>
## 2
## 3
## 4
## 5
## 6
##          V37          V38          V39          V40
## 1 <U+6700><U+9AD8> <U+306E> <U+30B2><U+30FC><U+30E0> <U+306E>
## 2
## 3
## 4
## 5
## 6
##          V41          V42          V43          V44          V45          V46          V47
## 1 <U+3053><U+3068> <U+3092> <U+982D> <U+306B> <U+601D> https t.co

```

```
## 2
## 3
## 4
## 5
## 6
##      V48 V49
## 1 aqp3i87iv1 NA
## 2           NA
## 3           NA
## 4           NA
## 5           NA
## 6           NA
```

```
str(TweetDF)
```

```
## 'data.frame': 100 obs. of 49 variables:
## $ V1 : Factor w/ 42 levels "__andrew_powell",...: 2 35 35 14 35 35 35 35 12 35 ...
## $ V2 : Factor w/ 69 levels "<U+3068>","adamschefter",...: 1 18 13 45 59 4 47 10 8 25 ...
## $ V3 : Factor w/ 64 levels "_tjwatt","<U+5E74>","...: 2 6 36 19 7 28 31 14 40 16 ...
## $ V4 : Factor w/ 67 levels "", "<U+306E>","...: 2 30 35 10 56 62 55 64 5 7 ...
## $ V5 : Factor w/ 73 levels "", "<U+53F2><U+4E0A>","...: 2 35 41 4 46 58 28 45 49 33 ...
## $ V6 : Factor w/ 68 levels "", "<U+6700><U+9AD8>","...: 2 27 48 12 44 52 45 39 43 55 ...
## $ V7 : Factor w/ 70 levels "", "<U+306E>","...: 2 8 49 36 53 41 37 3 40 13 ...
## $ V8 : Factor w/ 62 levels "", "<U+30B2><U+30FC><U+30E0>","...: 2 4 20 19 12 54 37 40 5 30 ...
## $ V9 : Factor w/ 61 levels "", "_codystewart",...: 23 37 52 38 26 33 21 37 28 2 ...
## $ V10: Factor w/ 62 levels "", "2yoona2zeroblitz",...: 57 54 43 29 49 20 29 28 29 2 ...
## $ V11: Factor w/ 52 levels "", "100","67.22",...: 31 32 23 42 17 14 42 2 42 43 ...
## $ V12: Factor w/ 46 levels "", "<U+30E9><U+30A4><U+30B9><U+30DC><U+30A6><U+30EB>","...: 2 24 20 40 39 ...
## $ V13: Factor w/ 42 levels "", "<U+304C>","...: 2 24 20 1 19 15 1 35 1 1 ...
## $ V14: Factor w/ 35 levels "", "<U+305D><U+306E>","...: 2 24 9 1 31 1 1 3 1 1 ...
## $ V15: Factor w/ 26 levels "", "<U+59FF>","...: 2 13 1 1 4 1 1 20 1 1 ...
## $ V16: Factor w/ 20 levels "", "<U+3092>","...: 2 17 1 1 1 1 1 18 1 1 ...
## $ V17: Factor w/ 18 levels "", "<U+5909><U+3048><U+308B>","...: 2 14 1 1 1 1 1 6 1 1 ...
## $ V18: Factor w/ 11 levels "", "<U+3053><U+3068><U+306B>","...: 2 1 1 1 1 1 1 1 1 1 ...
## $ V19: Factor w/ 5 levels "", "<U+306A>","...: 2 1 1 1 1 1 1 1 1 1 ...
## $ V20: Factor w/ 4 levels "", "<U+3063>","...: 2 1 1 1 1 1 1 1 1 1 ...
## $ V21: Factor w/ 2 levels "", "<U+305F>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V22: Factor w/ 2 levels "", "<U+3053><U+306E>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V23: Factor w/ 2 levels "", "<U+30CB><U+30E5><U+30FC><U+30B9>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V24: Factor w/ 2 levels "", "<U+306B>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V25: Factor w/ 2 levels "", "<U+89E6><U+308C>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V26: Factor w/ 2 levels "", "<U+305F>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V27: Factor w/ 2 levels "", "<U+3068><U+304D>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V28: Factor w/ 2 levels "", "<U+50D5>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V29: Factor w/ 2 levels "", "<U+306F>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V30: Factor w/ 2 levels "", "<U+5468><U+5E74>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V31: Factor w/ 2 levels "", "<U+3067>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V32: Factor w/ 2 levels "", "<U+9078><U+51FA>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V33: Factor w/ 2 levels "", "<U+3055>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V34: Factor w/ 2 levels "", "<U+308C>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V35: Factor w/ 2 levels "", "<U+305F>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V36: Factor w/ 2 levels "", "<U+53F2><U+4E0A>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V37: Factor w/ 2 levels "", "<U+6700><U+9AD8>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V38: Factor w/ 2 levels "", "<U+306E>":": 2 1 1 1 1 1 1 1 1 1 ...
## $ V39: Factor w/ 2 levels "", "<U+30B2><U+30FC><U+30E0>":": 2 1 1 1 1 1 1 1 1 1 ...
```



```
## $ V40: Factor w/ 2 levels "", "<U+306E>": 2 1 1 1 1 1 1 1 1 1 ...
## $ V41: Factor w/ 2 levels "", "<U+3053><U+3068>": 2 1 1 1 1 1 1 1 1 1 ...
## $ V42: Factor w/ 2 levels "", "<U+3092>": 2 1 1 1 1 1 1 1 1 1 ...
## $ V43: Factor w/ 2 levels "", "<U+982D>": 2 1 1 1 1 1 1 1 1 1 ...
## $ V44: Factor w/ 2 levels "", "<U+306B>": 2 1 1 1 1 1 1 1 1 1 ...
## $ V45: Factor w/ 2 levels "", "<U+601D>": 2 1 1 1 1 1 1 1 1 1 ...
## $ V46: Factor w/ 2 levels "", "https": 2 1 1 1 1 1 1 1 1 1 ...
## $ V47: Factor w/ 2 levels "", "t.co": 2 1 1 1 1 1 1 1 1 1 ...
## $ V48: Factor w/ 2 levels "", "aqp3i87iv1": 2 1 1 1 1 1 1 1 1 1 ...
## $ V49: logi NA NA NA NA NA NA ...
```

Cleaning the text data

Note that cleaning the text data is very important in text mining applications. Tweets are especially “messy”. We will remove “rt”, “http”, etc and any other strings of no importance.

```
## Convert all columns to char
```

```
TweetDF<-TweetDF %>%
  mutate_all(as.character)
str(TweetDF)
```

```
## 'data.frame': 100 obs. of 49 variables:
## $ V1 : chr "<U+30E9><U+30A4><U+30B9><U+30DC><U+30A6><U+30EB>" "rt" "rt" "congratulations" ...
## $ V2 : chr "<U+3068>" "domclare" "cowhercbs" "nfl" ...
## $ V3 : chr "<U+5E74>" "49ers" "nfl" "flag" ...
## $ V4 : chr "<U+306E>" "keeping" "lost" "bowl" ...
## $ V5 : chr "<U+53F2><U+4E0A>" "jimmy" "matriarch" "7u" ...
## $ V6 : chr "<U+6700><U+9AD8>" "garoppolo" "pat" "champions" ...
## $ V7 : chr "<U+306E>" "around" "rooney" "maui" ...
## $ V8 : chr "<U+30B2><U+30FC><U+30E0>" "2021" "embodied" "ducks" ...
## $ V9 : chr "https" "nfl" "spirit" "nflflagbowl" ...
## $ V10: chr "t.co" "season" "pittsburgh" "https" ...
## $ V11: chr "pcmpevm2pp" "per" "lived" "t.co" ...
## $ V12: chr "<U+30E9><U+30A4><U+30B9><U+30DC><U+30A6><U+30EB>" "mikegarafolo" "life" "tisbo7vynr" .
## $ V13: chr "<U+304C>" "nfl" "impacted" "" ...
## $ V14: chr "<U+305D><U+306E>" "network" "every" "" ...
## $ V15: chr "<U+59FF>" "https" "" "" ...
## $ V16: chr "<U+3092>" "t.co" "" "" ...
## $ V17: chr "<U+5909><U+3048><U+308B>" "szxdz" "" "" ...
## $ V18: chr "<U+3053><U+3068><U+306B>" "" "" "" ...
## $ V19: chr "<U+306A>" "" "" "" ...
## $ V20: chr "<U+3063>" "" "" "" ...
## $ V21: chr "<U+305F>" "" "" "" ...
## $ V22: chr "<U+3053><U+306E>" "" "" "" ...
## $ V23: chr "<U+30CB><U+30E5><U+30FC><U+30B9>" "" "" "" ...
## $ V24: chr "<U+306B>" "" "" "" ...
## $ V25: chr "<U+89E6><U+308C>" "" "" "" ...
## $ V26: chr "<U+305F>" "" "" "" ...
## $ V27: chr "<U+3068><U+304D>" "" "" "" ...
## $ V28: chr "<U+50D5>" "" "" "" ...
## $ V29: chr "<U+306F>" "" "" "" ...
## $ V30: chr "<U+5468><U+5E74>" "" "" "" ...
## $ V31: chr "<U+3067>" "" "" "" ...
## $ V32: chr "<U+9078><U+51FA>" "" "" "" ...
## $ V33: chr "<U+3055>" "" "" "" ...
```

```
## $ V34: chr "<U+308C>" "" "" "" ...
## $ V35: chr "<U+305F>" "" "" "" ...
## $ V36: chr "<U+53F2><U+4E0A>" "" "" "" ...
## $ V37: chr "<U+6700><U+9AD8>" "" "" "" ...
## $ V38: chr "<U+306E>" "" "" "" ...
## $ V39: chr "<U+30B2><U+30FC><U+30E0>" "" "" "" ...
## $ V40: chr "<U+306E>" "" "" "" ...
## $ V41: chr "<U+3053><U+3068>" "" "" "" ...
## $ V42: chr "<U+3092>" "" "" "" ...
## $ V43: chr "<U+982D>" "" "" "" ...
## $ V44: chr "<U+306B>" "" "" "" ...
## $ V45: chr "<U+601D>" "" "" "" ...
## $ V46: chr "https" "" "" "" ...
## $ V47: chr "t.co" "" "" "" ...
## $ V48: chr "aqp3i87iv1" "" "" "" ...
## $ V49: chr NA NA NA NA ...
```

```
# We can now remove certain words
```

```
TweetDF[TweetDF == "t.co"] <- ""
TweetDF[TweetDF == "rt"] <- ""
TweetDF[TweetDF == "http"] <- ""
TweetDF[TweetDF == "https"] <- ""
```

```
## Clean with grepl - every row in each column
```

```
MyDF<-NULL
```

```
for (i in 1:ncol(TweetDF)){
```

```
  MyList=c() # each list is a column of logicals ...
```

```
  MyList=c(MyList,grepl("[[:digit:]]", TweetDF[[i]]))
```

```
  MyDF<-cbind(MyDF,MyList) ## create a logical DF
```

```
  ## TRUE is when a cell has a word that contains digits
```

```
}
```

```
## For all TRUE, replace with blank
```

```
TweetDF[MyDF] <- ""
```

```
head(TweetDF,10)
```

```
##           V1           V2           V3           V4           V5
## 1
## 2           domclare           keeping           jimmy
## 3           cowhercbs           nfl           lost matriarch
## 4 congratulations           nfl           flag           bowl
## 5           sharpfootball           td           passes
## 6           barrysanders           just           wanted           send
## 7           nfl_memes           matthew           stafford           greatest
## 8           cbssportshq           deshaun           watson           one
## 9           check           broadcast playstation
## 10          glivingood           dmoorenfl aliccscouting jaberuski
##           V6           V7           V8           V9           V10          V11
## 1
## 2          garoppolo           around           nfl           season           per
## 3           pat           rooney embodied           spirit pittsburgh lived
## 4          champions           maui           ducks           nflflagbowl
## 5           matthew           stafford calvin           johnson           remember           good
## 6           quick           note thanking mattstafford           family           done
## 7           mic'd           moments           nfl           history
## 8           kind           player           nfl           history
```

```
## 9          madden          nfl          live
## 10 sethmurphybbd chasecameron_          _codystewart
##          V12          V13          V14 V15 V16 V17 V18 V19 V20 V21 V22 V23
## 1
## 2 mikegarafolo          nfl network          szxdz
## 3          life impacted          every
## 4
## 5          times
## 6          team          fans
## 7
## 8          pass          tds          rush tds          fir
## 9
## 10
## V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40 V41
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## V42 V43 V44 V45 V46 V47 V48 V49
## 1 <NA>
## 2 <NA>
## 3 <NA>
## 4 <NA>
## 5 <NA>
## 6 <NA>
## 7 <NA>
## 8 <NA>
## 9 <NA>
## 10 <NA>
```

```
# Now we save the dataframe using the write table command
write.table(TweetDF, file = "UpdatedChocolate.csv", col.names = FALSE,
            row.names = FALSE, sep = ",")
TweetTrans <- read.transactions("UpdatedChocolate.csv", sep = ",",
                                format("basket"), rm.duplicates = TRUE)
```

```
## distribution of transactions with duplicates:
## items
## 1 2 3
## 10 5 1
```

```
inspect(head(TweetTrans))
```

```
## items
## [1] {}
## [2] {around,
##      domclare,
##      garoppolo,
##      jimmy,
```

```

##      keeping,
##      mikegarafolo,
##      network,
##      nfl,
##      per,
##      season,
##      szxdz}
## [3] {cowhercbs,
##      embodied,
##      every,
##      impacted,
##      life,
##      lived,
##      lost,
##      matriarch,
##      nfl,
##      pat,
##      pittsburgh,
##      rooney,
##      spirit}
## [4] {bowl,
##      champions,
##      congratulations,
##      ducks,
##      flag,
##      maui,
##      nfl,
##      nflflagbowl}
## [5] {calvin,
##      good,
##      johnson,
##      matthew,
##      passes,
##      remember,
##      sharpfootball,
##      stafford,
##      td,
##      times}
## [6] {barrysanders,
##      done,
##      family,
##      fans,
##      just,
##      mattstafford,
##      note,
##      quick,
##      send,
##      team,
##      thanking,
##      wanted}

```

ARM

Next we will apply the apriori algorithm to find the associations including computing the support, confidence and lift. Read more on the arules library to tweak / tune the following code to achieve desired results.

```
# So that you do not have an enormous amount of rules, you can thresholds for  
# support, confidence and lift ... also minlength for the rules.
```

```
TweetTrans_rules = arules::apriori(TweetTrans,  
  parameter = list(support=.025, confidence=.45, minlen=3))  
  
## Apriori  
##  
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
##      0.45      0.1      1 none FALSE          TRUE      5  0.025      3  
## maxlen target ext  
##      10 rules TRUE  
##  
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
##      0.1 TRUE TRUE  FALSE TRUE      2      TRUE  
##  
## Absolute minimum support count: 2  
##  
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[584 item(s), 100 transaction(s)] done [0.00s].  
## sorting and recoding items ... [61 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.01s].  
## writing ... [77305 rule(s)] done [0.05s].  
## creating S4 object ... done [0.08s].
```

```
inspect(head(TweetTrans_rules))
```

```
##      lhs                                rhs      support confidence  
## [1] {bleacherreport,brgridiron} => {mcclain_on_nfl} 0.03      1  
## [2] {bleacherreport,mcclain_on_nfl} => {brgridiron}  0.03      1  
## [3] {send,thankings}              => {quick}      0.03      1  
## [4] {quick,thankings}              => {send}      0.03      1  
## [5] {quick,send}                  => {thankings}  0.03      1  
## [6] {send,thankings}              => {note}      0.03      1  
##      coverage lift      count  
## [1] 0.03      6.666667 3  
## [2] 0.03     12.500000 3  
## [3] 0.03     33.333333 3  
## [4] 0.03     33.333333 3  
## [5] 0.03     33.333333 3  
## [6] 0.03     33.333333 3
```

```
## sorted
```

```
SortedRules_conf <- sort(TweetTrans_rules, by="confidence", decreasing=TRUE)  
inspect(head(SortedRules_conf))
```

```
##      lhs                                rhs      support confidence  
## [1] {bleacherreport,brgridiron} => {mcclain_on_nfl} 0.03      1  
## [2] {bleacherreport,mcclain_on_nfl} => {brgridiron}  0.03      1
```

```
## [3] {send,thanking}      => {quick}      0.03      1
## [4] {quick,thanking}     => {send}      0.03      1
## [5] {quick,send}        => {thanking}   0.03      1
## [6] {send,thanking}     => {note}      0.03      1
##      coverage lift      count
## [1] 0.03      6.666667 3
## [2] 0.03     12.500000 3
## [3] 0.03     33.333333 3
## [4] 0.03     33.333333 3
## [5] 0.03     33.333333 3
## [6] 0.03     33.333333 3
```

```
SortedRules_sup <- sort(TweetTrans_rules, by="support", decreasing=TRUE)
inspect(head(SortedRules_sup))
```

```
##      lhs      rhs      support confidence coverage lift
## [1] {matthew,stafford} => {nfl}      0.12      0.8571429 0.14      1.530612
## [2] {matthew,nfl}     => {stafford} 0.12      1.0000000 0.12      6.250000
## [3] {nfl,stafford}    => {matthew} 0.12      0.8571429 0.14      6.122449
## [4] {mic'd,moments}   => {greatest} 0.11      1.0000000 0.11      9.090909
## [5] {greatest,moments} => {mic'd}    0.11      1.0000000 0.11      9.090909
## [6] {greatest,mic'd} => {moments} 0.11      1.0000000 0.11      9.090909
##      count
## [1] 12
## [2] 12
## [3] 12
## [4] 11
## [5] 11
## [6] 11
```

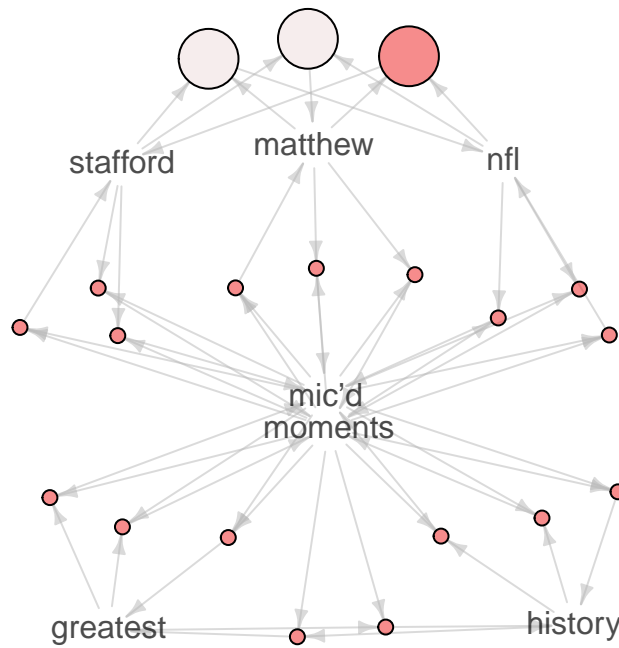
Displaying Results

The results will be displayed as an interactive graph.

```
plot (SortedRules_sup[1:20],method="graph",shading="confidence")
```

Graph for 20 rules

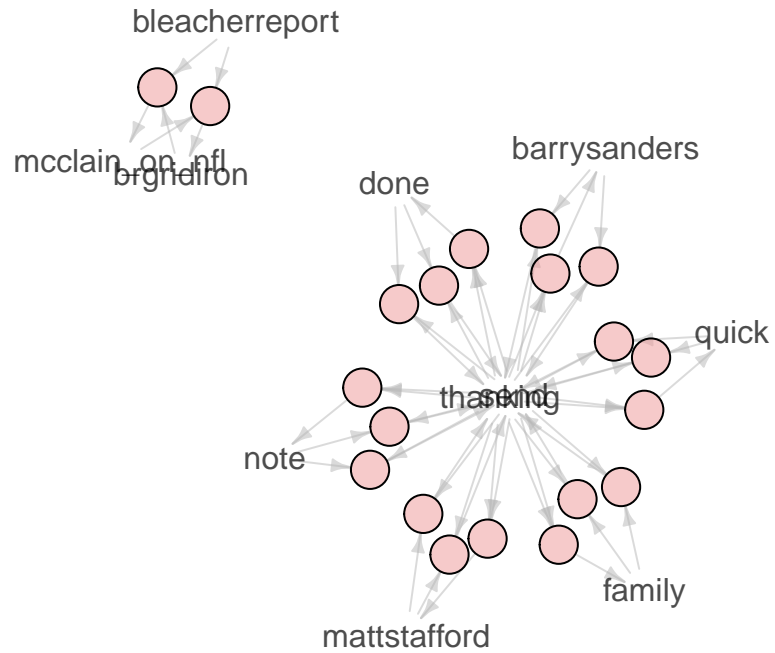
size: support (0.11 – 0.12)
color: confidence (0.857 – 1)



```
plot (SortedRules_conf[1:20],method="graph",shading="confidence")
```

Graph for 20 rules

size: support (0.03 – 0.03)
color: confidence (1 – 1)



```
plot (SortedRules_sup[1:20],method="graph",interactive=TRUE,shading="confidence")  
plot (SortedRules_conf[1:20],method="graph",interactive=TRUE,shading="confidence")
```