Social network analysis

Javier Monedero

Social network analysis tutorial

1. Preparation:

- Download: http://www.rdatamining.com/data/termDocMatrix.rdata?attredirects=0&d=1
- Download: http://www.rdatamining.com/data/rdmTweets.RData?attredirects=0&d=1
- load("./rdmTweets-201306.RData")
- load("./termDocMatrix.rdata")#A dataset of rminingtweets prepared by Yanchang Zhao

```
head(termDocMatrix)
```

```
##                Docs
## Terms          1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
##   analysis     0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  1  0  1  1  1  1  0  0
##   applications 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   code         0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   computing    0 0 1 1 0 1 1 1 1  1  0  1  0  0  0  0  0  0  0  0  0  0  0
##   data         1 1 0 0 2 0 0 0 0  0  1  2  1  1  1  0  1  0  0  0  0  0  0
##   examples     0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##                Docs
## Terms          24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
##   analysis      1  0  0  1  1  1  1  0  0  0  0  0  0  0  1  0  0  0  0  0
##   applications  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0
##   code          0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   computing     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
##   data          0  0  0  0  1  0  0  0  1  0  0  1  1  0  0  0  0  0  1  0
##   examples      0  0  0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0
##                Docs
## Terms          44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
##   analysis      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   applications  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
##   code          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   computing     0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
##   data          0  1  1  1  0  0  1  0  0  0  1  1  0  0  1  1  0  1  0  1
##   examples      0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##                Docs
## Terms          64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
##   analysis      0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0
##   applications  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
##   code          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
##   computing     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   data          0  1  1  0  0  0  1  0  0  0  1  0  0  0  2  0  0  0  0  1
##   examples      0  0  0  0  0  0  0  0  0  1  1  0  0  1  0  0  0  0  0  0
##                Docs
## Terms          84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102
##   analysis      0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0   0   1   0
##   applications  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0   0   0   0
##   code          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0   0   0
##   computing     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0   0   0
##   data          0  0  0  0  1  0  1  0  0  1  0  0  0  1  0  0   0   0   0
##   examples      0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0   0   0   0
##                Docs
## Terms           103 104 105 106 107 108 109 110 111 112 113 114 115 116 117
##   analysis        0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
##   applications    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   code            0   0   1   0   0   1   0   0   0   1   0   0   0   0   0
##   computing       0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   data            0   3   0   1   0   0   1   0   0   0   0   1   0   0   1
##   examples        0   0   1   0   0   1   0   0   0   1   0   0   0   0   1
##                Docs
## Terms           118 119 120 121 122 123 124 125 126 127 128 129 130 131 132
##   analysis        0   0   0   1   0   0   0   0   1   1   1   0   0   0   0
##   applications    1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   code            0   1   0   0   0   0   0   0   0   0   0   1   1   0   0
##   computing       0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   data            0   1   0   1   0   1   1   0   0   0   0   0   0   0   0
##   examples        0   1   0   1   0   0   0   0   0   0   0   1   0   0   0
##                Docs
## Terms           133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
##   analysis        0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
##   applications    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   code            0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
##   computing       0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   data            1   0   0   0   0   1   2   0   0   0   2   0   2   1   0
##   examples        0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
##                Docs
## Terms           148 149 150 151 152 153 154
##   analysis        0   0   0   0   0   1   0
##   applications    0   0   0   0   0   0   1
##   code            0   0   0   0   0   0   0
##   computing       0   0   0   0   0   0   0
##   data            0   2   0   1   1   0   2
##   examples        0   0   0   0   0   0   0
```

Change to a boolean matrix:

```
termDocMatrix[termDocMatrix>=1] <- 1
```

Transform into a term-term adjacent matrix by using the expression termMatrix <- termDocMatrix %% *t(termDocMatrix).* %% means multiply the two matrices and t is the transposed matrix.

```
head(termMatrix)
```

```
##                Terms
## Terms           analysis applications code computing data examples
##   analysis            23            0    1         0    4        4
##   applications         0            9    0         0    7        0
##   code                 1            0    9         0    1        6
##   computing            0            0    0        10    1        0
##   data                 4            7    1         1   53        5
##   examples            4            0    6         0    5       17
##                Terms
## Terms           introduction mining network package parallel positions
##   analysis                 2      4      12       2        0         2
##   applications             0      6       0       1        0         0
##   code                     0      3       1       0        0         0
##   computing                0      1       0       2        7         0
##   data                     2     34       0       7        1         5
##   examples                 2      5       2       2        0         0
##                Terms
## Terms           postdoctoral  r research series slides social time tutorial
##   analysis                 3 11        1      4      3      9    4        4
##   applications             0  4        1      0      0      0    0        0
##   code                     0  8        0      2      0      0    2        1
##   computing                1  9        0      0      1      0    0        1
##   data                     5 22        6      1      4      0    1        4
##   examples                 0 14        0      2      1      1    2        3
##                Terms
## Terms           users
##   analysis          5
##   applications      1
##   code              0
##   computing         2
##   data              4
##   examples          3
```

2. Network of terms

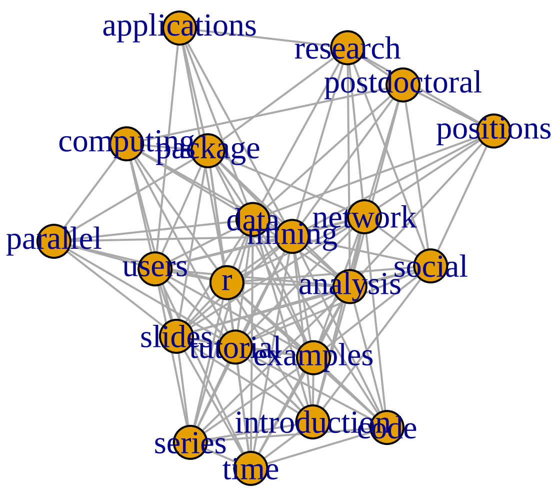
```r
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```r
graph <- graph.adjacency(termMatrix, weighted=T, mode="undirected")
graph <- simplify(graph)#remove loops
V(graph)$label <- V(graph)$name#Set labels
V(graph)$degree <- degree(graph)#Set degrees of vertices
layout1 <- layout.fruchterman.reingold(graph)
plot(graph, layout=layout1)
```



You can play with it. For example use:

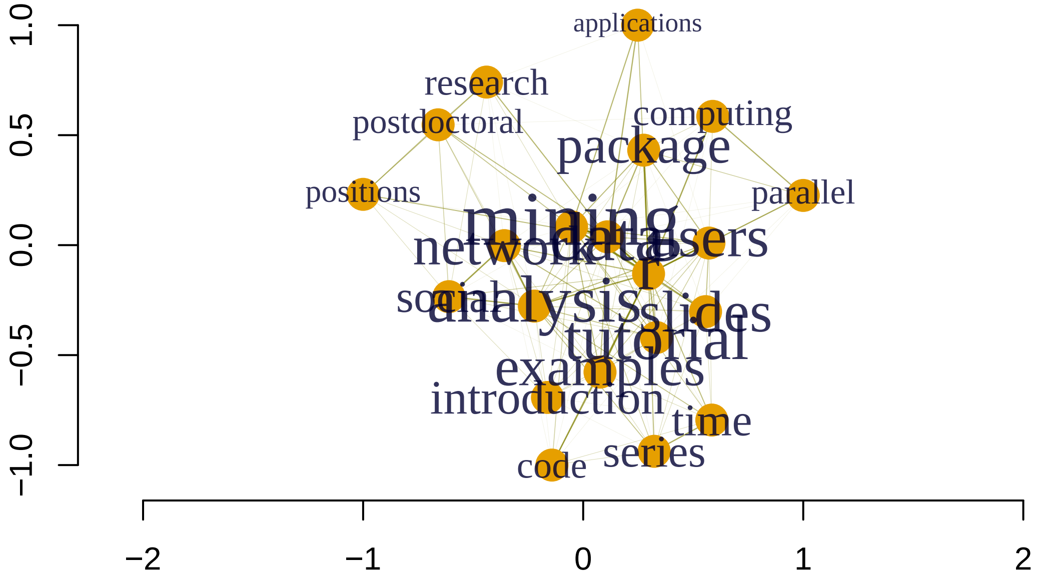
```
layout=layout.kamada.kawai
tkplot(graph, layout=layout.kamada.kawai)
```

```
## [1] 1
```

Set the label size of vertices based on their degrees:

```
V(graph)$label.cex <- 2.2 * V(graph)$degree / max(V(graph)$degree)+ .2
V(graph)$label.color <- rgb(0, 0, .2, .8)
V(graph)$frame.color <- NA
egam <- (log(E(graph)$weight)+.4) / max(log(E(graph)$weight)+.4)
E(graph)$color <- rgb(.5, .5, 0, egam)
E(graph)$width <- egam
plot(graph, layout1)
```

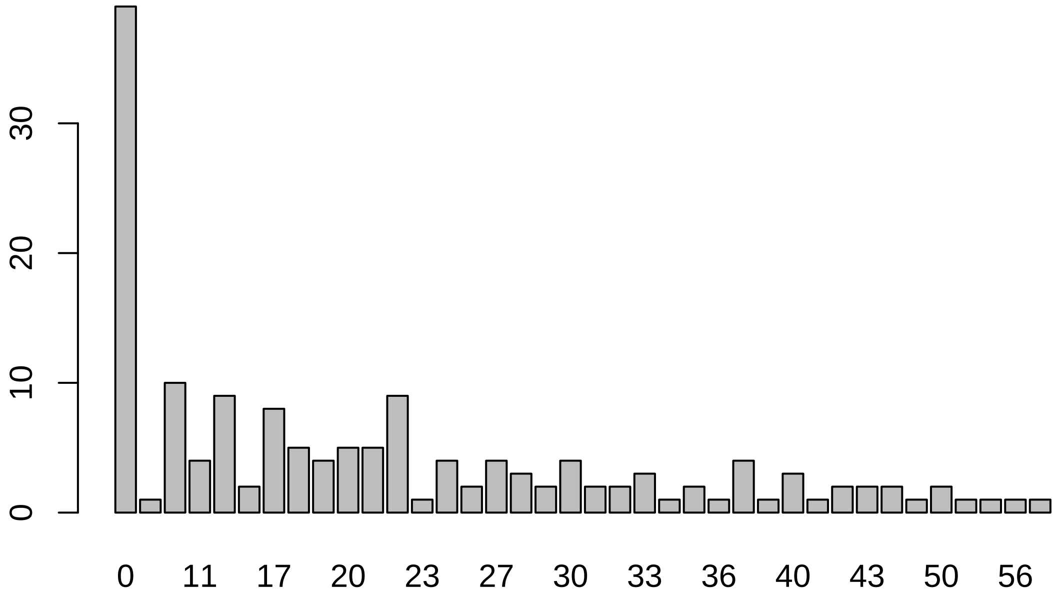
```
## Warning in if (axes) {: the condition has length > 1 and only the first
## element will be used
```



3. Network of tweets

Due to the fact that most tweets in the termDocMatrix are related with the words r, data and mining, first remove them.

```
idx <- which(dimnames(termDocMatrix)$Terms %in% c("r", "data", "mining"))
M <- termDocMatrix[-idx,]
tweetMatrix <- t(M) %*% M#tweet-tweet adjacent matrix as before
graph <- graph.adjacency(tweetMatrix, weighted=T, mode = "undirected")
V(graph)$degree <- degree(graph)
graph <- simplify(graph)
# set labels of vertices to tweet IDs
V(graph)$label <- V(graph)$name
V(graph)$label.cex <- 1
V(graph)$label.color <- rgb(.4, 0, 0, .7)
V(graph)$size <- 2
V(graph)$frame.color <- NA
#Distribution of degree of vertices
barplot(table(V(graph)$degree))
```

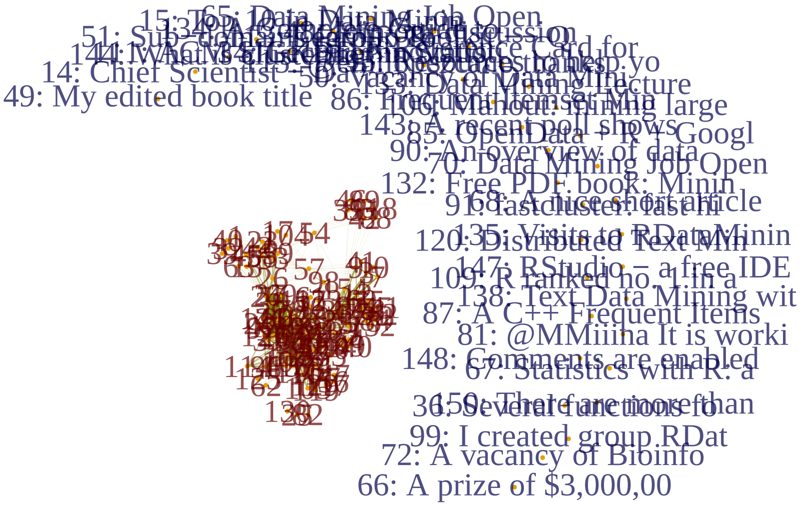


Something less than 40 vertices area isolated probably because of removing r, data and mining terms.

Let's complicate it a little more. First, we set vertex colors based on degree. Then we set labels of isolated vertices to tweet IDs and the first 20 characters of every tweet. The labels of other vertices are set to tweet IDs only in order to avoid the graph to be overcrowded with labels. We also set the color and width of edges based on their weights.

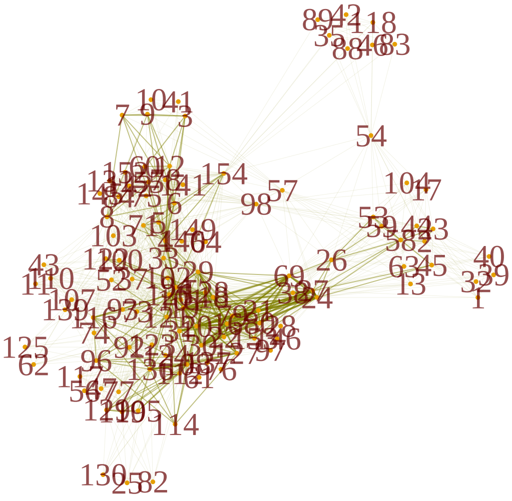
```
library(twitteR)
idx <- V(graph)$degree == 0
V(graph)$label.color[idx] <- rgb(0, 0, .3, .7)
load(file = "./rdmTweets.RData")
# convert tweets to a data frame
df <- do.call("rbind", lapply(rdmTweets, as.data.frame))#Convert the tweets to a data frame
# set labels to the IDs and the first 20 characters of tweets
V(graph)$label[idx] <- paste(V(graph)$name[idx], substr(df$text[idx], 1, 20), sep=": ")
```

```
egam <- (log(E(graph)$weight)+.2) / max(log(E(graph)$weight)+.2)
E(graph)$color <- rgb(.5, .5, 0, egam)
E(graph)$width <- egam
layout2 <- layout.fruchterman.reingold(graph)
plot(graph, layout=layout2)
```



Next, remove isolated vertices fromteh graph, and plot:

```
graph2 <- delete.vertices(graph, V(graph)[degree(graph)==0])
plot(graph2, layout=layout.fruchterman.reingold)
```

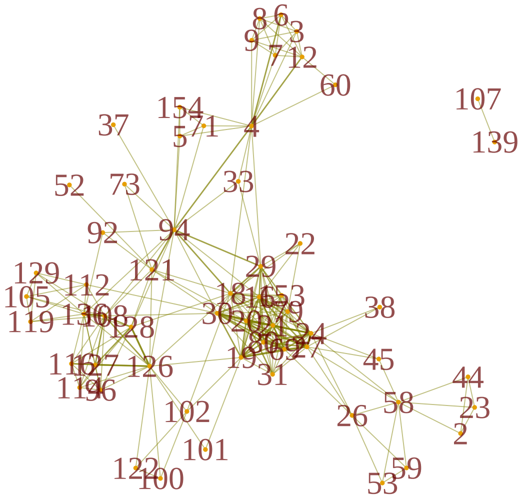


We can also remove edges with low degrees to appreciate the other edges better:

```
graph3 <- delete.edges(graph, E(graph)[E(graph)$weight <= 1])
graph3 <- delete.vertices(graph3, V(graph3)[degree(graph3) == 0])
#You can focus on specific groups. Here, I choose the group in the middle left:
df$text[c(7,12,6,9,8,3,4)]
```

```
## [1] "State of the Art in Parallel Computing with R http://t.co/zmClglqi"
## [2] "The R Reference Card for Data Mining is updated with functions & packages for handling big data
## [3] "Parallel Computing with R using snow and snowfall http://t.co/nxp8EZpv"
## [4] "R with High Performance Computing: Parallel processing and large memory http://t.co/XZ3ZZBRF"
## [5] "Slides on Parallel Computing in R http://t.co/AdDVxbOY"
## [6] "Easier Parallel Computing in R with snowfall and sfCluster http://t.co/BPcinvzK"
## [7] "Tutorial: Parallel computing using R package snowfall http://t.co/CHBCyr76"
```

```r
plot(graph3, layout=layout.fruchterman.reingold)
```



4. Two-mode network

It is a network with two types of vertices: tweets and terms. Term and tweet vertices are distinguished by colors and sizes.

```r
graph <- graph.incidence(termDocMatrix, mode=c("all"))
```

Get indexes for term vertices and tweet vertices:

```r
nTerms <- nrow(M)
nDocs <- ncol(M)
idx.terms <- 1:nTerms
idx.docs <- (nTerms+1):(nTerms+nDocs)
```

Set colors and sizes for vertices:

```r
V(graph)$degree <- degree(graph)
V(graph)$color[idx.terms] <- rgb(0, 1, 0, .5)
```

```
## Warning in vattrs[[name]][index] <- value: number of items to replace is
## not a multiple of replacement length
```

```r
V(graph)$size[idx.terms] <- 6
```

```
## Warning in vattrs[[name]][index] <- value: number of items to replace is
## not a multiple of replacement length
```

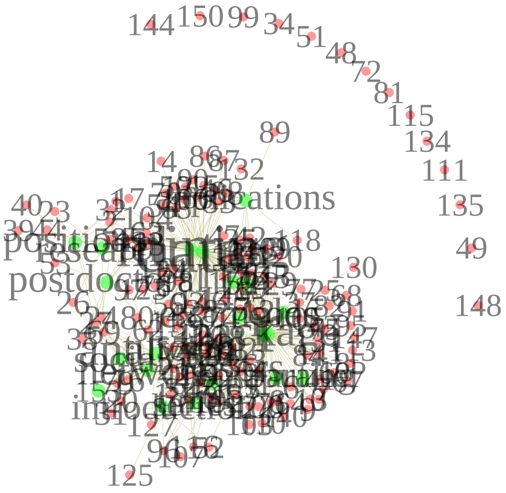
```r
V(graph)$color[idx.docs] <- rgb(1, 0, 0, .4)
V(graph)$size[idx.docs] <- 4
V(graph)$frame.color <- NA
```

Set vertex labels and their colors and sizes:

```r
V(graph)$label <- V(graph)$name
V(graph)$label.color <- rgb(0, 0, 0, 0.5)
V(graph)$label.cex <- 1.4*V(graph)$degree/max(V(graph)$degree) + 1
```

Set edge width and color:

```r
E(graph)$width <- .3
E(graph)$color <- rgb(.5, .5, 0, .3)
plot(graph, layout=layout.fruchterman.reingold)
```



“r”, “data” and “mining” represent the three centers with most tweets. Which vertices deal with “r”?

```r
V(graph)[nei("r")]
```

```
## + 70/175 vertices, named:
##  [1] 3   4   5   6   7   8   9   10  12  19  21  22  25  28  30  33  35
## [18] 36  41  42  55  64  67  68  73  74  75  77  78  82  84  85  91  92
## [35] 94  95  100 101 102 105 108 109 110 112 113 114 117 118 119 120 121
## [52] 122 126 128 129 131 136 137 138 140 141 142 143 145 146 147 149 151
## [69] 152 154
```

One can also determine which tweets contain the three terms:

```r
(rdmVertices <- V(graph)[nei("r") & nei("data") & nei("mining")])
```

```
## + 14/175 vertices, named:
##  [1] 12  35  36  42  55  78  117 119 138 143 149 151 152 154
```

```
df$text[as.numeric(rdmVertices$label)]#See what tweets correspond to such numbers
```

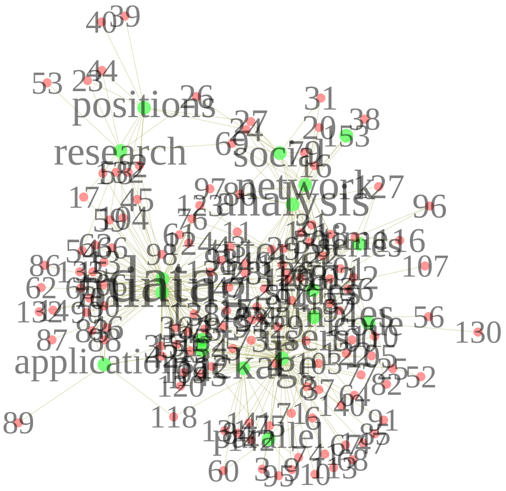
```
##  [1] "The R Reference Card for Data Mining is updated with functions & packages for handling big data
##  [2] "Call for reviewers: Data Mining Applications with R. Pls contact me if you have experience on t
##  [3] "Several functions for evaluating performance of classification models added to R Reference Card
##  [4] "Call for chapters: Data Mining Applications with R, an edited book to be published by Elsevier
##  [5] "Some R functions and packages for outlier detection have been added to R Reference Card for Dat
##  [6] "Access large amounts of Twitter data for data mining and other tasks within R via the twitteR p
##  [7] "My document, R and Data Mining - Examples and Case Studies, is scheduled to be published by Els
##  [8] "Lecture Notes on data mining course at CMU, some of which contain R code examples. http://t.co/
##  [9] "Text Data Mining with Twitter and R.  http://t.co/a50ySNq"
## [10] "A recent poll shows that R is the 2nd popular tool used for data mining. See Poll: Data Mining/
## [11] "RDataMining group: to share your experience on using R for data mining with other data miners.
## [12] "R Reference Card for Data Mining also available at mirrors: www2.rdatamining.com, www3.rdatamin
## [13] "TraMineR is an excellent R package for mining and visualizing sequence data. Its function seqe
## [14] "An R Reference Card for Data Mining is now available on CRAN. It lists many useful R functions
```

Now, delete “r”, “data” and “mining” as well as vertices to see the relationships between tweets and other words.

```
idx <- which(V(graph)$name %in% c("r", "data", "mining"))
graph2 <- delete.vertices(graph, V(graph)[idx-1])
graph2 <- delete.vertices(graph2, V(graph2)[degree(graph2)==0])
df$text[as.numeric(rdmVertices$label)]#See what tweets correspond to such numbers
```

```
##  [1] "The R Reference Card for Data Mining is updated with functions & packages for handling big data
##  [2] "Call for reviewers: Data Mining Applications with R. Pls contact me if you have experience on t
##  [3] "Several functions for evaluating performance of classification models added to R Reference Card
##  [4] "Call for chapters: Data Mining Applications with R, an edited book to be published by Elsevier
##  [5] "Some R functions and packages for outlier detection have been added to R Reference Card for Dat
##  [6] "Access large amounts of Twitter data for data mining and other tasks within R via the twitteR p
##  [7] "My document, R and Data Mining - Examples and Case Studies, is scheduled to be published by Els
##  [8] "Lecture Notes on data mining course at CMU, some of which contain R code examples. http://t.co/
##  [9] "Text Data Mining with Twitter and R.  http://t.co/a50ySNq"
## [10] "A recent poll shows that R is the 2nd popular tool used for data mining. See Poll: Data Mining/
## [11] "RDataMining group: to share your experience on using R for data mining with other data miners.
## [12] "R Reference Card for Data Mining also available at mirrors: www2.rdatamining.com, www3.rdatamin
## [13] "TraMineR is an excellent R package for mining and visualizing sequence data. Its function seqe
## [14] "An R Reference Card for Data Mining is now available on CRAN. It lists many useful R functions
```

```
plot(graph2, layout=layout.fruchterman.reingold)#Groups of tweets and their keywords
```



Thank you for your time!