Social Media Intelligence Assignment 2022

Project Report

Elisabeth Putri – 20306250

3rd June 2022

By including this statement, we the authors of this work, verify that:

- I hold a copy of this assignment that we can produce if the original is lost or damaged.

- I hereby certify that no part of this assignment/product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.

- No part of this assignment/product has been written/produced for us by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned.

- I am aware that this work may be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism (which may retain a copy on its database for future plagiarism checking).

- I hereby certify that we have read and understand what the School of Computing and Mathematics defines as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

The aim of this project is giving insights for Elon Musk about his decision to buy Twitter. There are several tasks to be completed, then the results are to be presented to the board of directors of Twitter. The tasks comprise: Gathering the Network, Mention Graph, Graph Statistics, Information Flow, Account Popularity, and Account Selection. The analysis code is done by using R, those are attached in this report.

1. Gathering the Network

Twitter is one of the trending social media which has been used world-wide. It gives a freedom for everybody to share their own mind, even their opinions regarding to an event. Because of this ability, we can gather public opinions related into Elon Musk's decision to purchase Twitter. There are 25,000 tweets data to be used for this analysis. The completed data contains 90 information each tweet, such as user id, status id, created at, screen name, text, source, display text width, reply to status id, reply to user id, reply to screen name, is quote, is retweet, favourite count, retweet count, quote count, reply count, hashtags, symbols, url, media url and media type, mention user id, mention screen name, language, quoted information (such as status id, text, etc.), and retweeted information.

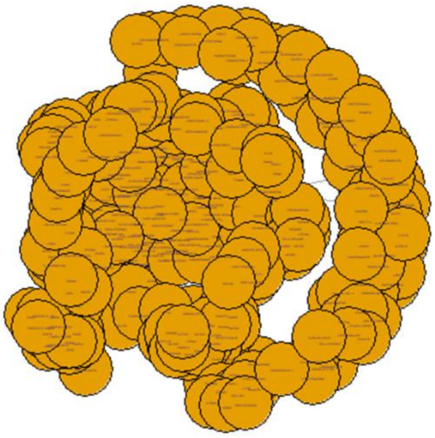
In this task, the part of data which be used is the text. The table below shows several tweets from the data we use.

User name	Tweets
CunningStunt19	`goddammit I was promised an edit button. fuck all #elonmusktwitter`

AChristhope	Dear Social Media Users:\n\n#Twitter #instagram #Facebook #socialmediamanager #YouTube #YouTubers #socialjusticewarrior #politician #Leaders #followers #walkthetalk @elonmusk #elonmusktwitter https://t.co/rMynzvjzmx
Gaurav_GK99	Am creating a illustrated photo of you and going to make it as nft . Will you buy it @elonmusk #ElonMusk #ElonMuskTwitter if you retweet or like this. I will give you the nft link. Guys please to make a change in my life. Am going to post this daily untill @elonmusk shares.

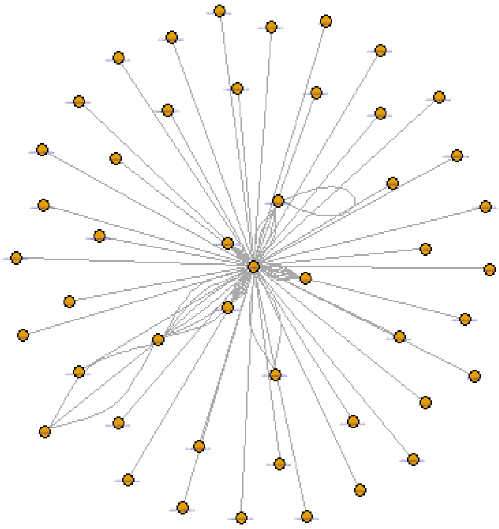
2. Mention Graph

From the observed data, we are going to examine the relation of the users by creating graph. This graph assesses the relation from each tweet by using their mention. These graphs depict the relation between each account from the data. The difference between both graphs is the direction of the edge between each account. The directed graph has arrow in the end of the edge, otherwise.



From the data, we found that there are 52 number of components. Component means a subset of nodes (users) which link to each other, and this subset is separated from other larger connected set of nodes. The size difference between each component is distinct. The largest component has 354 connected users, while the others are just 11 and below.

Considering about the size component, the largest component is chosen to be plotted. The plot is depicted below.



3. Graph Statistics

This section is mainly examining properties of the graph (largest component subgraph). First, we will investigate the diameter of the graph. The graph diameter means the maximum shortest path between each node. In our graph, the diameter is 6.

Next, the density of the graph is based on the edge's connectivity. It is basically a ratio between the edge in the graph and the maximum possible edge for that graph. In this case, the density of our graph is 0.024. It means the network connection in our graph is only 2.4% compared to how connected it might be, thus the information will not transmit efficiently.

Other properties which is important to be investigated in a graph are degree and degree distribution. Degree is accessing the edge of each node, while degree distribution is calculating the probability of distribution in each degree in the complete network. As this graph is an undirected graph, degree can be calculated once without separation between in-link and out-link of each node. From this graph, the highest number of edges in a node is 178. The histogram plot of degree distribution shows that the number of edges in each node is not well distributed. From the analysis, it rejects the null hypothesis of Kolmogorov-Smirnov test as its result is so small. The Kolmogorov-Smirnov hypothesis is the original data could have been drawn from the fitted power-law distribution. It is supported by the power law coefficient which is only 1.026638. Therefore, this graph is not following the form of power law distribution.

4. Information Flow

When individual has relation with each other, they commonly pass their information to someone else, especially when their relation is a strong relationship. They might not pass the information directly, however, most likely people will react whenever they receive something about person that they know. For example, when one person makes a problem, the community around him which knows him will know and gives reaction.

The same logic happens in the graph. The community surround him is called neighbours which can give reaction, either clarification, insight, or rejection. In our graph, we investigate what people opinions are about Elon Musk purchases Twitter. The neighbourhood overlap is used in this analysis.

This analysis only uses the largest component; thus, it is specific into 146 twitter users only. These tweets contain people opinion, they only have relation with Elon Musk by mentioning his twitter account. They were not talking about Elon Musk (it might not detect because we are only focusing on mention graph, I think we should use retweet graph to see their discussion), it makes each of the twitter accounts do not have any relation to each other. It is proved by the highest number of related accounts is between the user id of Elon Musk (id for reply to user, not user id which means it is mentioned not the one who has the tweet) and DeDludla who mentioned him twice.

5. Account Popularity

In the web search, it is affected by the rank of each page. The rank is evaluated by voting, which each in-link into the page works as an endorsement into it. To calculate this into the graph, there is Page Rank method. Assuming that each page (in this analysis is tweet) has reference into another page, it will link to each other to give more details information. This PageRank value does not change, it will always sum the score to 1. More popular the account, their Page Rank score will get higher.

In our graph analysis, the PageRank is calculated by Scaled PageRank, and it is compared with eigen value. The ten most popular account and their PageRank score is detailed below.

Id number	Twitter Account User Name	Page Rank score
"6182852"	Elon Musk	0.308518527
"128372940"	Bill Gates	0.039552618
"51827346"	-	0.026090744
"1513921541918138372"	milgrim369	0.020782265
"1316181534"	GHWStewart	0.017171617
"1021241294"	Lady_Williams84	0.013849264
"1450520650335072262"	Omeletela	0.012462664
"1452304054206488576"	Saitejatalari1	0.010517661
"1510000127745970186"	PatariaAmada1022	0.010349100
"1423355927302975489"	zoellaawww	0.009315947

Aligned with the previous analysis, the most popular account is Elon Musk (by mentioned). The second popular account is Bill Gates (by mentioned). These two accounts become popular because there is one account, milgrim369, who mentioned them in his tweets several times. There is unknown vertex in the graph which we can track the id number, but the twitter account can not be found. The rest of the popular account is getting popular by mentioning Elon Musk in their tweet.

6. Account Selection

From the given pay off table, we can obtain which strategy should be used by Elon in monitoring his company for each day. The mixed strategy nash equilibrium is used in this analysis. This technique is chosen because there is no dominant strategy, no information about competitor chosen strategy.

			Competitor		
			0.315	0.28	0.405
			SpaceX	Tesla	Boring
Monitor	0.553	SpaceX	1, 0.3	0.1, 3	0.3, 2
	0.298	Tesla	0.2, 1	1, 0.2	0.3, 3
	0.149	Boring	0.1, 3	0.1, 2	1, 0.1

The calculation shows that in this equilibrium of probability. In this stage of equilibrium, the probability of SpaceX, makes the competitor indifferent between their strategies. It makes a guarantee that Elon has a better chance no matter what the competitor does. Because Elon can only monitor one account each day, assumed that this probability is given for the whole week (7 days), the calculation will be as shown.

Account	Day(s) per week
SpaceX	3.871 ≈ 4
Tesla	2.086 ≈ 2
Boring	1.043 ≈ 1

Social Media Intelligence Project

Elisabeth Putri - 20306250

27/05/2022

By including this statement, we the authors of this work, verify that: • I hold a copy of this assignment that we can produce if the original is lost or damaged. • I hereby certify that no part of this assignment/product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment. • No part of this assignment/product has been written/produced for us by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned. • I am aware that this work may be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism (which may retain a copy on its database for future plagiarism checking). • I hereby certify that we have read and understand what the School of Computing and Mathematics defines as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

Before we start the coding, importing all the library.

```
library('igraph')

## Warning: package 'igraph' was built under R version 4.1.3

##
## Attaching package: 'igraph'

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

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

library('rtweet')

## Warning: package 'rtweet' was built under R version 4.1.3

library('tm')

## Warning: package 'tm' was built under R version 4.1.3

## Loading required package: NLP

library('dplyr')

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1. Gathering the Network

```
em <- search_tweets(
  "#elonmusktwitter", n=25000, include_rts = FALSE, retryonratelimit = TRUE)

names(em)
```

```
##  [1] "user_id"                 "status_id"
##  [3] "created_at"              "screen_name"
##  [5] "text"                    "source"
##  [7] "display_text_width"      "reply_to_status_id"
##  [9] "reply_to_user_id"        "reply_to_screen_name"
## [11] "is_quote"                "is_retweet"
## [13] "favorite_count"          "retweet_count"
## [15] "quote_count"             "reply_count"
## [17] "hashtags"                "symbols"
## [19] "urls_url"                "urls_t.co"
## [21] "urls_expanded_url"       "media_url"
## [23] "media_t.co"              "media_expanded_url"
## [25] "media_type"              "ext_media_url"
## [27] "ext_media_t.co"          "ext_media_expanded_url"
## [29] "ext_media_type"          "mentions_user_id"
## [31] "mentions_screen_name"    "lang"
## [33] "quoted_status_id"        "quoted_text"
## [35] "quoted_created_at"       "quoted_source"
## [37] "quoted_favorite_count"   "quoted_retweet_count"
## [39] "quoted_user_id"          "quoted_screen_name"
## [41] "quoted_name"             "quoted_followers_count"
## [43] "quoted_friends_count"    "quoted_statuses_count"
## [45] "quoted_location"         "quoted_description"
## [47] "quoted_verified"         "retweet_status_id"
## [49] "retweet_text"            "retweet_created_at"
## [51] "retweet_source"          "retweet_favorite_count"
## [53] "retweet_retweet_count"   "retweet_user_id"
## [55] "retweet_screen_name"     "retweet_name"
## [57] "retweet_followers_count" "retweet_friends_count"
## [59] "retweet_statuses_count"  "retweet_location"
## [61] "retweet_description"     "retweet_verified"
## [63] "place_url"               "place_name"
## [65] "place_full_name"         "place_type"
## [67] "country"                 "country_code"
## [69] "geo_coords"              "coords_coords"
## [71] "bbox_coords"             "status_url"
## [73] "name"                    "location"
## [75] "description"             "url"
## [77] "protected"               "followers_count"
## [79] "friends_count"           "listed_count"
## [81] "statuses_count"          "favourites_count"
## [83] "account_created_at"      "verified"
## [85] "profile_url"             "profile_expanded_url"
## [87] "account_lang"            "profile_banner_url"
## [89] "profile_background_url"  "profile_image_url"
```

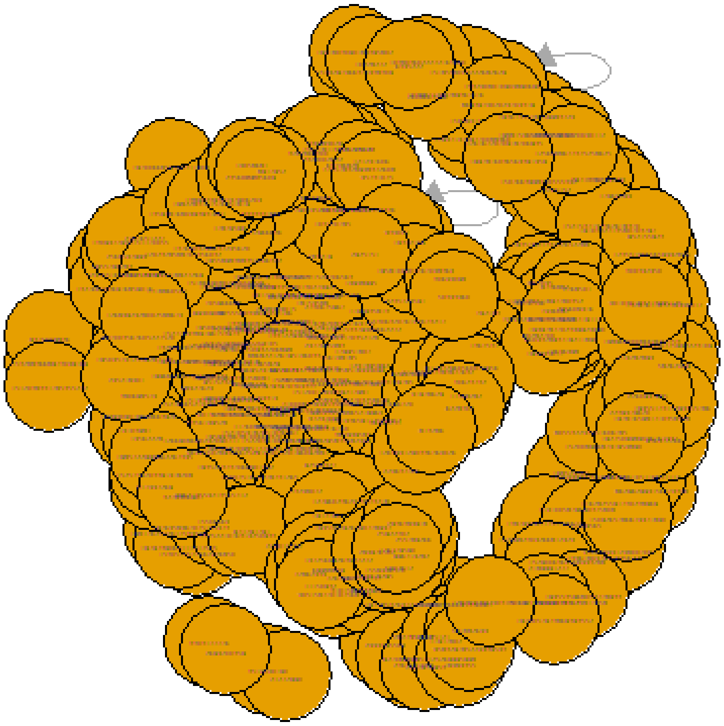
```r
tweettext = as.data.frame(em$text, em$screen_name)
print(head(tweettext))
```

```
##
em$text
## CunningStunts19
goddammit I was promised an edit button. fuck all #elonmusktwitter
## AChristhope
Dear Social Media Users:\n\n#Twitter #instagram #Facebook #socialmediamanager
#YouTube #YouTubers #socialjusticewarrior #politician #Leaders #followers
#walkthetalk @elonmusk #elonmusktwitter https://t.co/rMynzvjzmx
## Gaurav_GK99     Am creating a illustrated photo of you and going to make
it as nft . Will you buy it @elonmusk #ElonMusk #ElonMuskTwitter if you
retweet or like this. I will give you the nft link. Guys please to make a
change in my life. Am going to post this daily untill @elonmusk shares.
## Gaurav_GK99.1   Am creating a illustrated photo of you and going to make
it as nft . Will you buy it @elonmusk #ElonMusk #ElonMuskTwitter if you
retweet or like this. I will give you the nft link. Guys please to make a
change in my life. Am going to post this daily untill @elonmusk shares.
## Gaurav_GK99.2   Am creating a illustrated photo of you and going to make
it as nft . Will you buy it @elonmusk #ElonMusk #ElonMuskTwitter if you
retweet or like this. I will give you the nft link. Guys please to make a
change in my life. Am going to post this daily untill @elonmusk shares.
## Gaurav_GK99.3   Am creating a illustrated photo of you and going to make
it as nft . Will you buy it @elonmusk #ElonMusk #ElonMuskTwitter if you
retweet or like this. I will give you the nft link. Guys please to make a
change in my life. Am going to post this daily untill @elonmusk shares.
```

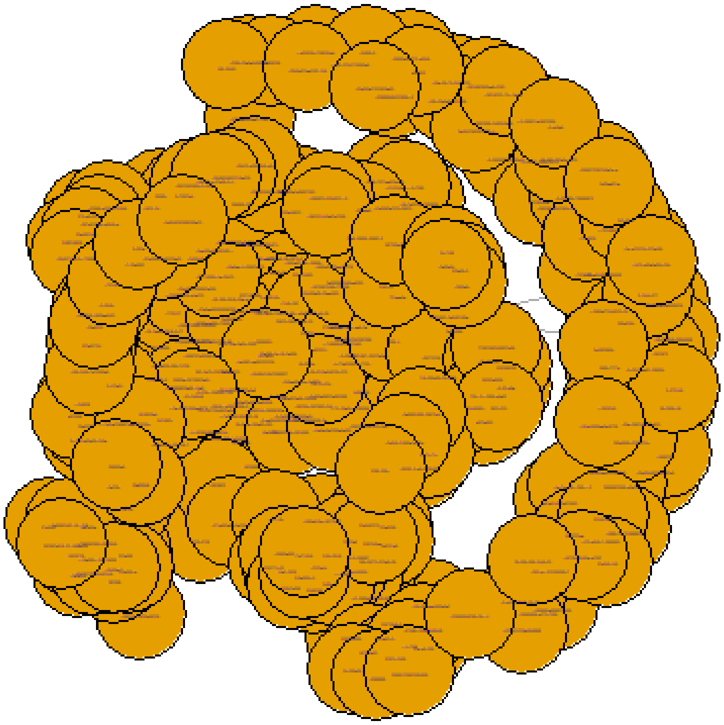
2. Mention Graph

```r
# Create graph on the mentions in each tweet
datatw = network_data(em, "mention")

gnet <- graph_from_data_frame(datatw, directed = TRUE)

par(mar = c(0, 0, 0, 0))
```

```
V(gnet)$label.cex=0.2
plot(gnet, layout = layout.fruchterman.reingold, vertex.size = 30)
```

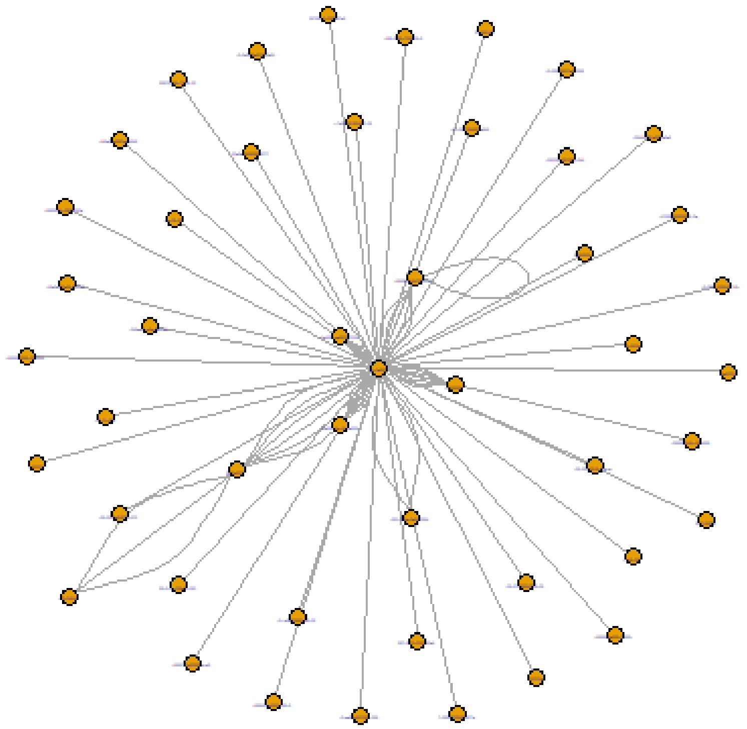


```
# Convert the directed graph into undirected graph
convertem <- graph_from_data_frame(datatw, directed = FALSE)

par(mar = c(0, 0, 0, 0))
V(convertem)$label.cex=0.1
plot(convertem, layout = layout.fruchterman.reingold, vertex.size = 30)
```

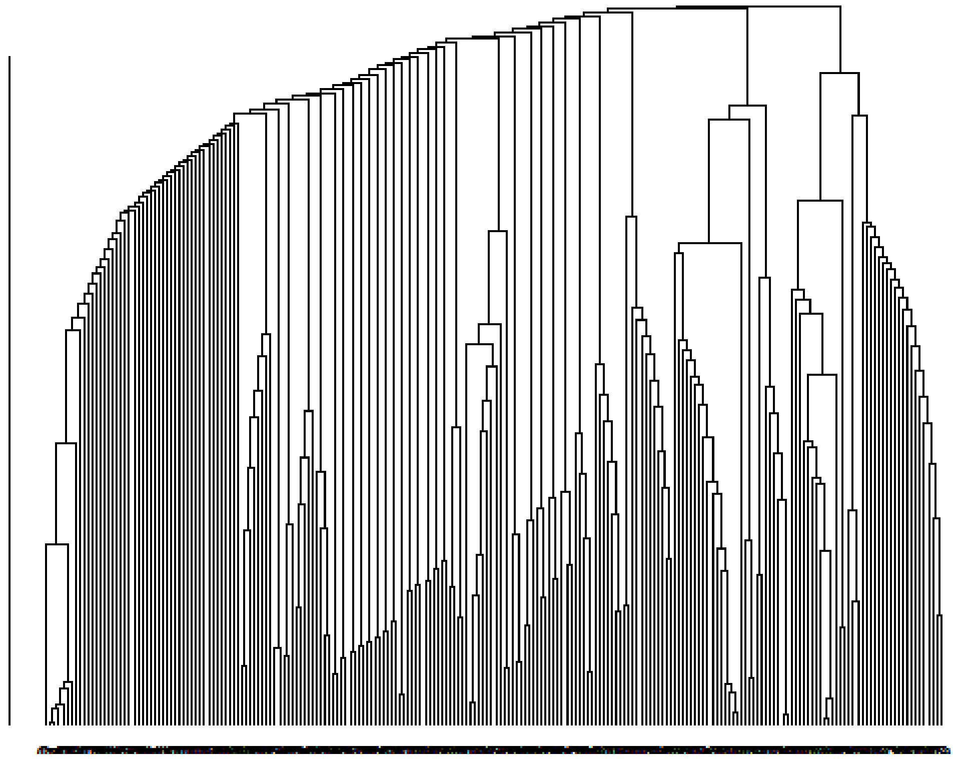


```
# Compute the number of components and size of each components
# number of components
count_components(convertem)

## [1] 30

# size of each components
aa = components(convertem)
aa$csize

##  [1] 216   2   2   2   2   3   3   2   2   2   2   3   2   1   2   2   2
2   3
## [20]   2   2   2   2   4   2   2   2   6   2   2

# Plot the largest component of the graph
## First split the components of the graph
deco = decompose(convertem)

## nodes number in each component
nc = sapply(deco, function(x) {length(V(x))})

## index of largest component
lc = which(nc == max(nc))

## largest component's edges
lce = deco[[lc]]
```

```
## largest component graph
lcgraph = cluster_edge_betweenness(lce, directed = FALSE)

# check all partition
print(lcgraph$membership)

##   [1]  1  1  2  3  4  5  1  6  7  1  1  1  1  1  1  8  1  9  7  1  1  2  1
1  1
##  [26]  5 10  1  1 11 10  1  2 12 13  2 14  1 15  4  4  7 16  1 17  1  1  1
3  1
##  [51]  1  1  1  1  4  1  1 18  7  1 17  1 19  2  1 14 20 20 21  1 22  7 21
1 20
##  [76] 23 20 24  1  1 25  1  1 26  2  7 27  1 28  1  1 29 30 23  1 31  1 32
7  2
## [101]  7  1 33 34  5  1  2  3  3  3  3  3  3  3  3  3  3  4 23  5  6  7  8
8  9
## [126]  1  5  5  5  5  5  5  5  5  5  5  5  5  5  5 10 11 11 10 10 10 10 10
12 13
## [151] 14 15  7 16 16 17 17 18 18 18 18 18 18 20 17 17 17 17 17 17 17 17 17
17 17
## [176] 17 17 17 17 17 17 17 17 19 14 20 20 20 20 20 21 22 20 20 20 20 24 25
25 25
## [201] 26 27 28 29 30 31 32 32  7 33 33 33 33 34  5  5

# check all the edges
table(lcgraph$membership)

##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26
## 47  8 12  5 20  2 11  3  2  8  3  2  2  4  2  3 23  7  2 14  3  2  3  2  4
2
## 27 28 29 30 31 32 33 34
##  2  2  2  2  2  3  5  2

## By checking all the edges, the highest number is 1, so we use 1 as the
largest component
lcno = 1
indexlc = which(lcgraph$membership == lcno)

## Taking the nodes from largest component to be plotted
lcgraphed = subgraph(deco[[lc]], indexlc)

## Warning in subgraph(deco[[lc]], indexlc): At
## structural_properties.c:2051 :igraph_subgraph is deprecated from igraph
0.6, use
## igraph_induced_subgraph instead

plot(lcgraphed, vertex.size=5)
```



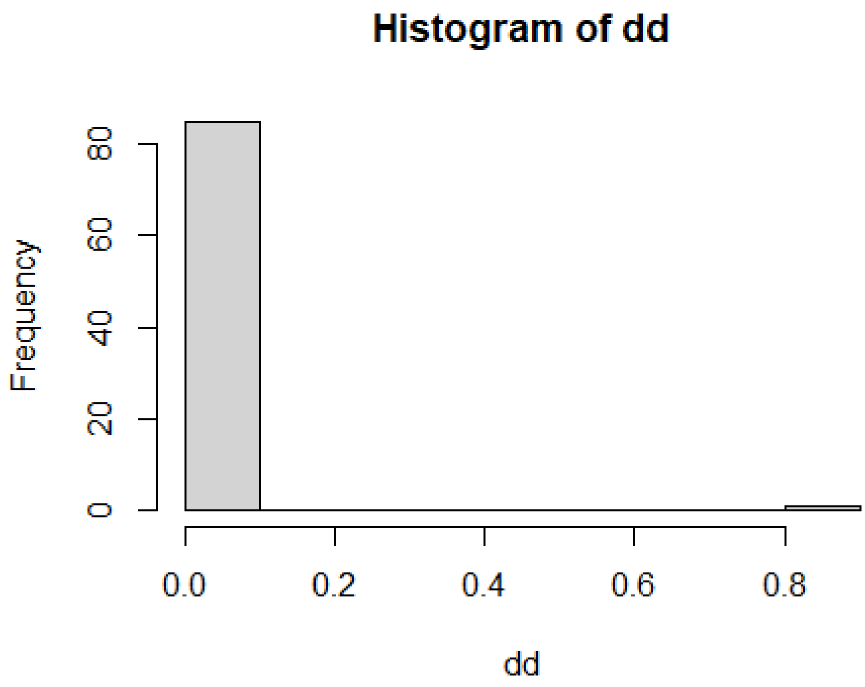
```
## Graph partition by hierarchical relationship
plot(as.dendrogram(lcgraph))
```



3. Graph Statistics

```
# graph diameter
diameter(lcgraphed)

## [1] 3

# graph density
graph.density(lcgraphed)

## [1] 0.08233117

# plotting the degree distribution of the graph
degree(lcgraphed)

## 1334440824608706562 1464529166557237253  710008378945355776
1523178461560418304
##                   1                  18                   1
2
##  886822126145196033           158553324 1176088924894257152
1436568756331839492
##                   8                   7                   1
1
##          3165824401            97297046 1331224968407891970
1316630499738021889
##                   1                   1                   1
1
##           363762075 1519396939736829952 1483070768980316160
1475554039890952196
##                   9                   1                   1
1
## 1455186931759910918 1484854743470309377            45152326
901313610806251520
##                   1                   7                   1
1
## 1450820441803829256  837126781282852864          2977485983
908701589170352128
##                   1                   1                   1
1
##            16222574 1480481160820109319           156198185
40628724
##                   1                   1                   1
1
## 1488327858518999044           174451807 1522034108788281344
1445628693376684040
##                   1                   1                   1
1
## 1408056463331758081 1512608806575898624 1512001189034217481
1521059182451036161
##                   1                   1                   1
1
## 1531579300947759105 1256866059098955777 1345668916261965824
```

```
1241807750432206848
##                    1                   1                   1
1
##  927032191233568768 1399302045413294085          2648989015
1511636208325169155
##                    1                   1                   1
1
##           1021241294            44196397          1291945442
##                    1                  85                   3

max(degree(lcgraphed))

## [1] 85

dd = degree.distribution(lcgraphed)
hist(dd)
```



```
# estimating the Power Law coefficient(c) from the degree distribution
pl = fit_power_law(dd, xmin = 0.0000000000000000001)
print(pl)

## $continuous
## [1] TRUE
##
## $alpha
## [1] 1.026219
##
## $xmin
## [1] 1e-18
##
## $logLik
## [1] -10.68584
##
## $KS.stat
## [1] 0.6268302
##
## $KS.p
## [1] 0.003721941
```

```r
print(pl)$alpha
```

```
## $continuous
## [1] TRUE
##
## $alpha
## [1] 1.026219
##
## $xmin
## [1] 1e-18
##
## $logLik
## [1] -10.68584
##
## $KS.stat
## [1] 0.6268302
##
## $KS.p
## [1] 0.003721941
```

```
## [1] 1.026219
```

~~~
4. Information Flow


```r
# neighourhood overlap between each pair of connected nodes in the twitter
graph
# changing the name of the vertex to easier analysis process
bb = set.vertex.attribute(lcgraphed, "name", value = paste("A",1:146, sep =
""))
~~~

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

```r
en = ends(bb, E(bb), names = FALSE)
```

```r
ne.over = function(no1, no2, graphed) {
  i = intersection(neighbors(graphed, no1), neighbors(graphed, no2))
```

```
  u = union(neighbors(graphed, no1), neighbors(graphed, no2))
  a = length(i)/ (length(u))
  return(a)
}

nodes = list()

for (i in seq(nrow(en))){
  node1 = en[i, 1]
  node2 = en[i, 2]
  nover = ne.over(no1 = node1, no2 = node2, graphed = bb)
  nodes[i] = nover
}


# identify the pair with the greatest and least neighborhood overlap
d = c()

for (i in 1:length(nodes)){
  d[i] = nodes[[i]][1]
}

l = d[order(d,decreasing=TRUE)]
#l
```

5. Account Popularity

```
# Measuring popularity of each account by Scaled PageRank


hu = rep(1, length(names(V(bb))))

names(hu) = names(V(bb))

au = hu


M = as_adjacency_matrix(bb)

M = as.matrix(M)


## iterate K times

ik = 100

for (i in 1:ik) {
  au = t(M) %*% hu
  hu = M %*% au
}

au = au/sum(au)
hu = hu/sum(hu)

# eigenvalue solution
ed1 = eigen(M %*% t(M))
hub.ed = ed1$vectors[,1]

ed2 = eigen(t(M) %*% M)
au.ed = ed2$vectors[,1]

hub.ed = hub.ed/sum(hub.ed)
au.ed = au.ed/sum(au.ed)

# Result checking
au - au.ed
hu - hub.ed

## PageRank

P = M %*% diag(1/colSums(M))
nonodes = nrow(P)
R = matrix(1/nonodes, nonodes, nonodes)
```

```
ld = 0.8
S = ld*P + (1-ld)*R
colSums(S)

iter = rep(1/nonodes, nonodes)
for (k in 1:ik) {
  iter = S %*% iter
}

e.val = eigen(S)

pr = e.val$vector[,1]/sum(e.val$vector[,1])
pr = Re(pr)

#checking method
iter - pr

# Ten highest page rank
pro = sort(pr, index.return=TRUE)
pro$ix[1:10]
[1] 141 142 143   1   8   9  10  11  13  16

prorder = pr[order(pr,decreasing=TRUE)]
prorder[1:10]
[1] 0.308518527 0.039552618 0.026090744 0.020782265 0.017171617 0.013849264
0.012462664 0.010517661 0.010349100 0.009315947
```

~~~
V(lcgraphed)[141, 142, 143, 1, 8, 9, 10, 11, 13, 16]

+ 10/146 vertices, named, from 6efdbdc:

  [1] 6182852           128372940         51827346          1513921541918138372
1316181534        1021241294        1450520650335072262

[8] 1452304054206488576 1510000127745970186 1423355927302975489



em %>% filter_at(vars(user_id, reply_to_user_id), any_vars(. %in% c("6182852",
"128372940", "51827346", "1513921541918138372", "1316181534", "1021241294",
"1450520650335072262", "1452304054206488576", "1510000127745970186",
"1423355927302975489")))

```
~~~

6. Account Selection

			Competitor		
			q_1	q_2	1-q_1-q_2
			SpaceX	Tesla	Boring
Monitor	p_1	SpaceX	1, 0.3	0.1, 3	0.3, 2
	p_2	Tesla	0.2, 1	1, 0.2	0.3, 3
	1- p_1- p_2	Boring	0.1, 3	0.1, 2	1, 0.1

Monitor

$$E_{competitor}(SpaceX) = \; q_1 + 0.1q_2 + 0.3(1 - q_1 - q_2) = \; 0.7q_1 - 0.2q_2 + 0.3$$

$$E_{competitor}(Tesla) = \; 0.2q_1 + q_2 + 0.3(1 - q_1 - q_2) = \; -0.1q_1 + 0.7q_2 + 0.3$$

$$E_{competitor}(Boring) = \; 0.1q_1 + 0.1q_2 + (1 - q_1 - q_2) = \; -0.9q_1 - 0.9q_2 + 1$$

Competitor

$$E_{monitor}(SpaceX) = \; 0.3p_1 + p_2 + 3(1 - p_1 - p_2) = \; -2.7p_1 - 2p_2 + 3$$

$$E_{monitor}(Tesla) = \; 3p_1 + 0.2p_2 + 2(1 - p_1 - p_2) = \; p_1 - 1.8p_2 + 2$$

$$E_{monitor}(Boring) = \; 2p_1 + 3p_2 + 0.1(1 - p_1 - p_2) = \; 1.9p_1 + 2.9p_2 + 0.1$$

From those equations, the probability distribution for each strategy is given below.

			Competitor		
			0.315	0.28	0.405
			SpaceX	Tesla	Boring
Monitor	0.553	SpaceX	1, 0.3	0.1, 3	0.3, 2
	0.298	Tesla	0.2, 1	1, 0.2	0.3, 3
	0.149	Boring	0.1, 3	0.1, 2	1, 0.1