

## Annex 2

### Data extraction from Facebook's comments

Copies of the articles published on newspapers web page are also published in the newspapers Facebook page in order to give the readers the possibility to comment them without the burden of managing the comment process.

After the Cambridge Analytica data scandal Facebook imposed serious restrictions on API usage. Given the above restrictions it has been more convenient to adopt web scraping techniques. All elements of the scraped text, like for example keywords, and attributes, like for example indents, are carefully analysed for correct interpretation with the aim of modelling a network that describes the connecting structure. R language has been used for this purpose, in particular the igraph package for network construction and measurements.

The first step consists in extracting all information contained in the comments section of the Facebook page related to the specific article. Next picture shows part of the section. User names are obfuscated in order to respect the privacy of users.



The image shows a screenshot of a Facebook comments section. The comments are as follows:

- Comment 1:** User [obfuscated] says "I tell you what The world hasn't just gone mad It has become increasingly cruel er and sadistic... See more". It has 59 likes and 1 reply.
- Comment 2:** User [obfuscated] says "This should carry a death penalty.". It has 57 likes and 1 reply.
- Comment 3:** User [obfuscated] says "Think we should bring back capital punishment or national service this country is broken". It has 5 likes and 1 reply.
- Comment 4:** User [obfuscated] says "Alan Turner both.". It has 2 likes and 1 reply.
- Reply Input:** A text box with "Write a reply..." and icons for emojis, photos, GIFs, and stickers.
- Comment 5:** User [obfuscated] says "i have 60% burns so i know this young boy has years and years of pain ahead. The 3 b#stards that have done this should be put through the same pain for the same amount of time". It has 38 likes and 1 reply.
- Comment 6:** User [obfuscated] says "Mincing machine would see the not do it again". It has 2 likes and 1 reply.
- Reply Input:** A text box with "Write a reply..." and icons for emojis, photos, GIFs, and stickers.
- Comment 7:** User [obfuscated] says "An imported crime. That's cultural enrichment for you.". It has 23 likes and 1 reply.

As above explained APIs able to extract information in a suitable format are not available. Web page scraping is a tedious exercise because the HTML code behind the page is extremely complex. The most convenient method is a simple copy/paste of the section that results with the following text related to the above picture.

```
User
User I tell you what
The world hasn't just gone mad
It has become increasingly cruel er and sadistic...See more
59

Like
· Reply · 49w

User
User i have 60% burns so i know this young boy has years and
years of pain ahead. The 3 b#stards that have done this should
be put through the same pain for the same amount of time
38

Like
· Reply · 49w

User
User Mincing machine would see the not do it again
2

Like
· Reply · 49w

Oreste Parlatano
Write a reply...

User
User An imported crime. That's cultural enrichment for you.
23

Like
· Reply · 49w
```

Text pattern analysis allows to recognise the difference between a comment and a reply and the number of emotional clicks, it resulted in a software program here detailed and explained.

First of all the R libraries needed:

```
stringr, gdata, readr, dplyr, grr, igraph
```

then the creation of an empty table that will contain the comments and their attributes

```
df_fb_comments <- data.frame(
  'sn' = integer(),      # serial number
  'un' = character(),    # user name
  'cm' = character(),    # comment
  'nl' = integer(),      # number of likes
  'rp' = logical(),      # is a reply ?
  stringsAsFactors=FALSE
)
```

Each line of pasted comments is evaluated for populating the above table

```
sn <- 0
un <- ''
cm <- ''
nl <- 0
rp <- FALSE
file_name <- paste0(dir_fb_comments,fb_comment_number,".txt")
file_connection <- file(file_name)
txt_fb_comments <- readLines(file_connection)
len_comments = length(txt_fb_comments)
rep <- ''
ffound <- 0
is_indented <- FALSE
wr <- FALSE
for (ind in 1:len_comments){
  cur_row_1 <- txt_fb_comments[ind]
  cur_row_2 <- txt_fb_comments[ind + 1]
  cur_row_3 <- txt_fb_comments[ind + 2]
  is_indented <- startsWith( cur_row_1, ' ') & startsWith( cur_row_2, ' ') &
ffound > 0
  un <- str_trim(cur_row_1, side = "both")
  maybe_len <- str_length(un)
  maybe_after <- cur_row_2
  if (grepl(un, maybe_after, fixed=TRUE) & maybe_len>5 & maybe_len<30) {
    sn <- sn + 1
    wr <- TRUE
    nl <- cur_row_3
    cm <- str_trim(str_remove(maybe_after, un), side = "both")
    if ( (!grepl("\\D", nl) & str_length(nl) > 0) ) {
      nl <- '0'
    }
    if (is_indented) {
      rep <- '- start of reply(ies) -'
      rp <- TRUE
      print(rep)
    }
    print(paste(sn, ',', un, ',', cm, ',', nl))
    ffound <- ffound + 1
  }
  if (grepl('Write a reply', un, fixed=TRUE)) {
    rep <- '- end of reply(ies) -'
    rp <- FALSE
    print(rep)
  }
  if (wr) {
    df_fb_comments[nrow(df_fb_comments) + 1,] = list(sn,un,cm,nl,rp)
  }
  is_indented <- FALSE
  rep <- ''
  wr <- FALSE
}
close(file_connection)
```

In order to comply with user privacy an additional table with all names is created, an id number is attributed to each name in a way that later only ids will be displayed instead of names.

```
fqun <- as.data.frame(table(df_fb_comments$un)) # get unique records of user
names and calculates frequencies
fqun <- mutate(fqun, id = rownames(fqun)) # adds column id and populates it
colnames(fqun)[1] <- 'name'
# prevent grr::matches error
df_fb_comments$un <- as.factor(df_fb_comments$un)
dictionary_fqun_df_fb_comments <- grr::matches(fqun$name, df_fb_comments$un)
dictionary_fqun_df_fb_comments_sorted <-
dictionary_fqun_df_fb_comments[order(dictionary_fqun_df_fb_comments[,2]),]
df_fb_comments <-
cbind(df_fb_comments,dictionary_fqun_df_fb_comments_sorted[,1])
colnames(df_fb_comments)[6] <- 'id'
```

Eventually two table are available for further processing, these tables are saved in CSV format that can be used by any other application program, for example Excel.

```
write.csv(df_fb_comments,paste0(dir_results,'fb_',fb_comment_number,'.csv'),
row.names = FALSE)
write.csv(fqun,paste0(dir_results,'fb_',fb_comment_number,'_names_frequency.csv'
), row.names = FALSE)
```

Next figure shows the table corresponding to FB comments picture above

sn	un	cm	nl	rp	id
53	obfuscated	I tell you what	59	FALSE	45
54	obfuscated	i have 60% burns so i know this young boy has years and years	38	FALSE	99
55	obfuscated	Mincing machine would see the not do it again	2	TRUE	125
56	obfuscated	An imported crime. That's cultural enrichment for you.	23	FALSE	56

Next figure shows the table containing user names and attributed ids

name	Freq	id
obfuscated	1	1
obfuscated	1	2
obfuscated	1	3
obfuscated	1	4
obfuscated	2	5
obfuscated	1	6
obfuscated	1	7
obfuscated	1	8
obfuscated	1	9
obfuscated	1	10
obfuscated	1	11
obfuscated	1	12
obfuscated	1	13
obfuscated	1	14
obfuscated	2	15
obfuscated	6	16
obfuscated	2	17
obfuscated	1	18
obfuscated	1	19
obfuscated	2	20

Now data are organized for further processing, the next step aims to compose the network representation of users comments.

Library "igraph", within R programming environment, contains all tools for implementation of graph algorithms.

Network graph is composed by iterating the table containing the comments, the first and central node is identified as "A", meaning the article's author, the other nodes are identified by an id number corresponding to the user in the table containing user names. Comments are represented by ties.

```
ph <- vector('character');
len <- nrow(df_fb_comments)
for (row in 1:len){
  cond <- df_fb_comments[row,5]
  if (cond) {
    ph <- c(ph,c(id,df_fb_comments[row,6]))
  } else {
    id <- as.character(df_fb_comments[row,6])
    ph <- c(ph,c('A',df_fb_comments[row,6]))
  }
}
gr <- graph(ph,directed = FALSE)
```

Facebook allow undirected ties, it means that friendship, or link, is mutual, if you are my friend, also I am your friend. This is the reason way the last instruction of the above script creates a graph with the option `directed = FALSE`.

Once the net graph is composed, then is possible to measure the diameter

```
di <- diameter(gr)
```

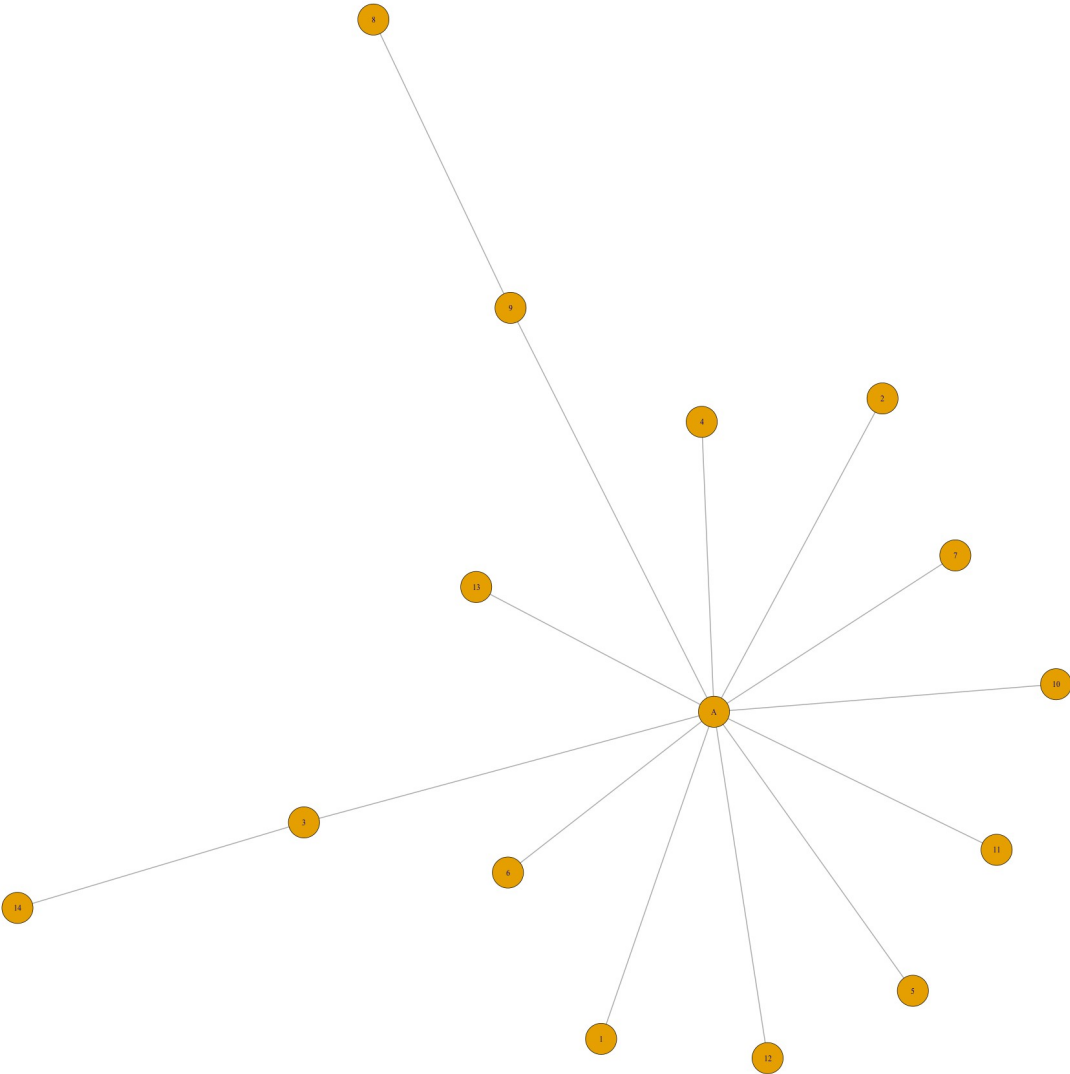
Finally is possible to draw the net graph and saving the image on disk

```
png(filename =
paste0(dir_results,"fb_net_",fb_comment_number,".png"),width =
1920, height = 1920)
plot(gr, vertex.size = 6,
vertex.label.cex = 1,
edge.width = 2,
main = paste0('FB users, article n. ',
fb_comment_number, ' - diameter = ', di)
)
dev.off()
```

The next pages contain network representation of all Facebook users comments.

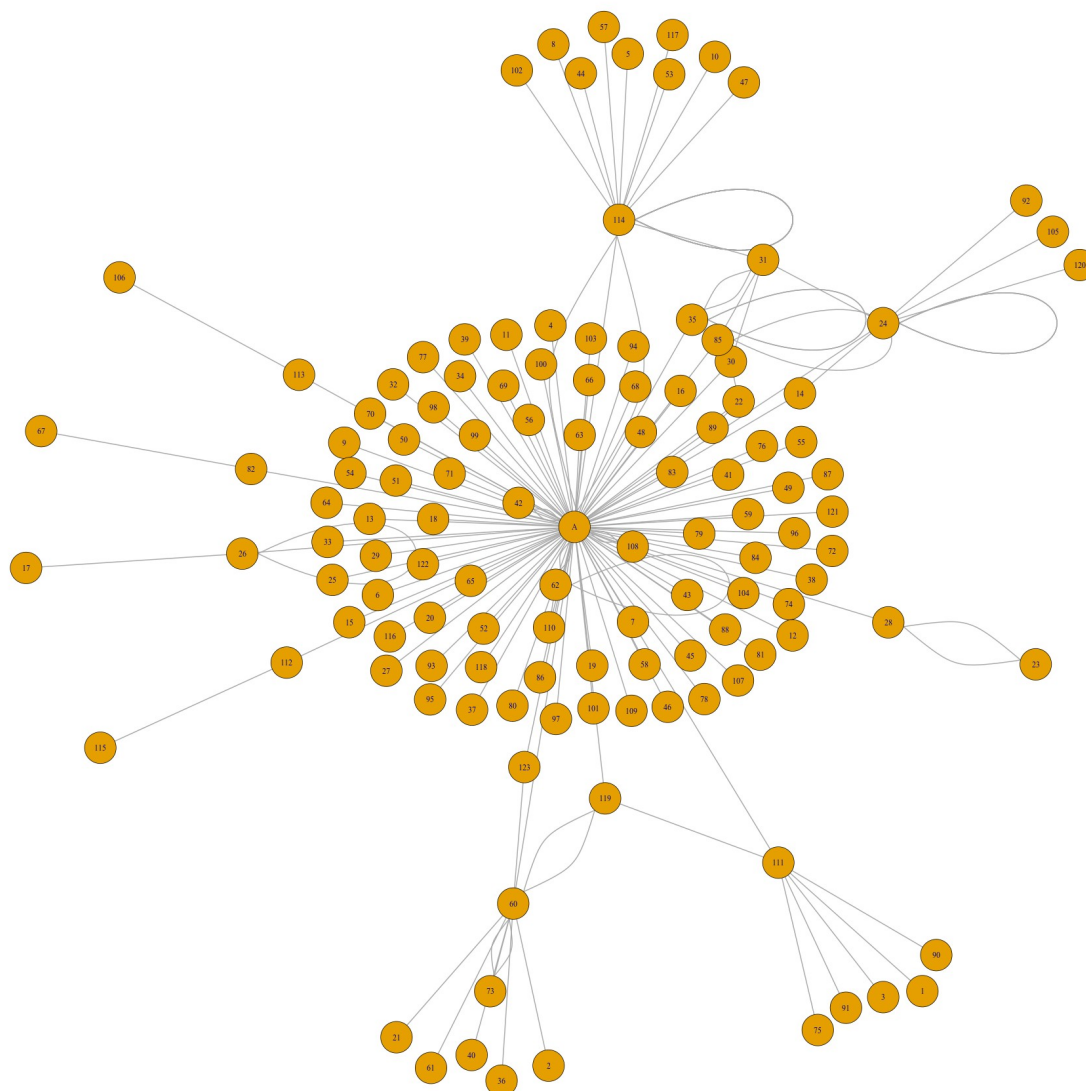
# Diagram Facebook network, article n. 145 – diameter = 4

FB users, article n. 145 - diameter = 4



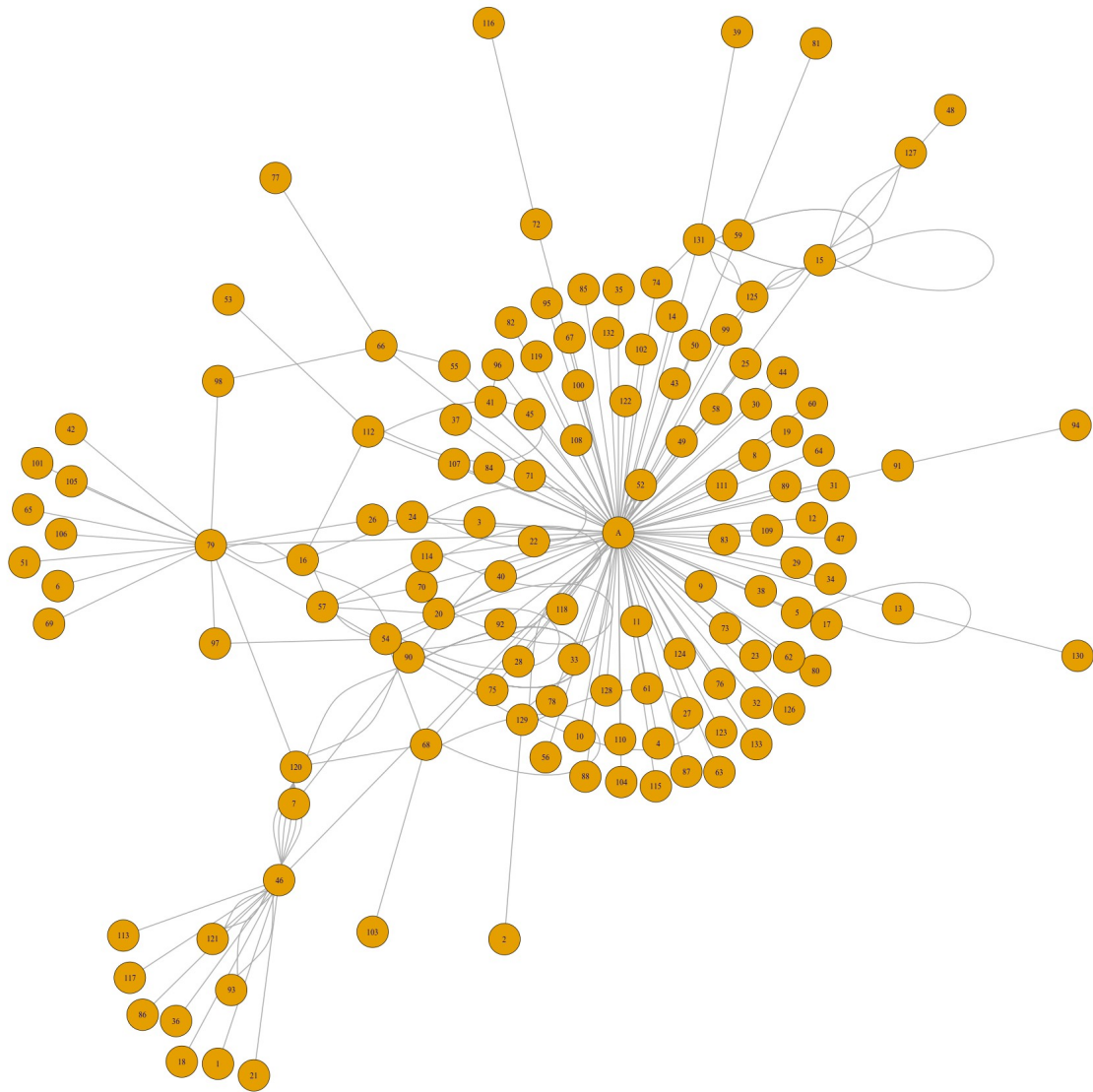
# Diagram Facebook network, comment n. 155 – diameter = 4

FB users, article n. 155 - diameter = 4



# Diagram Facebook network, comment n. 160 – diameter = 4

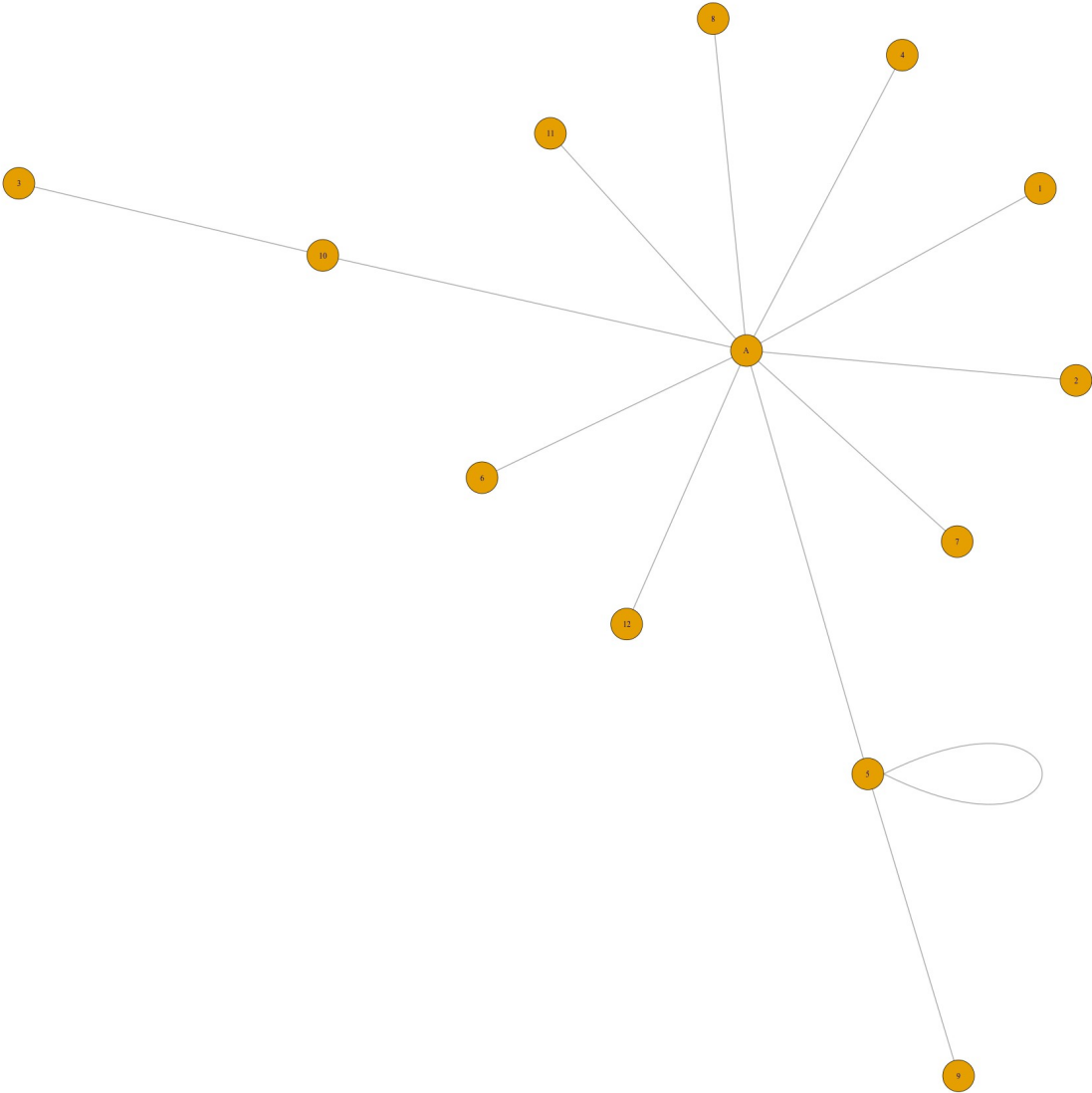
FB users, article n. 160 - diameter = 4





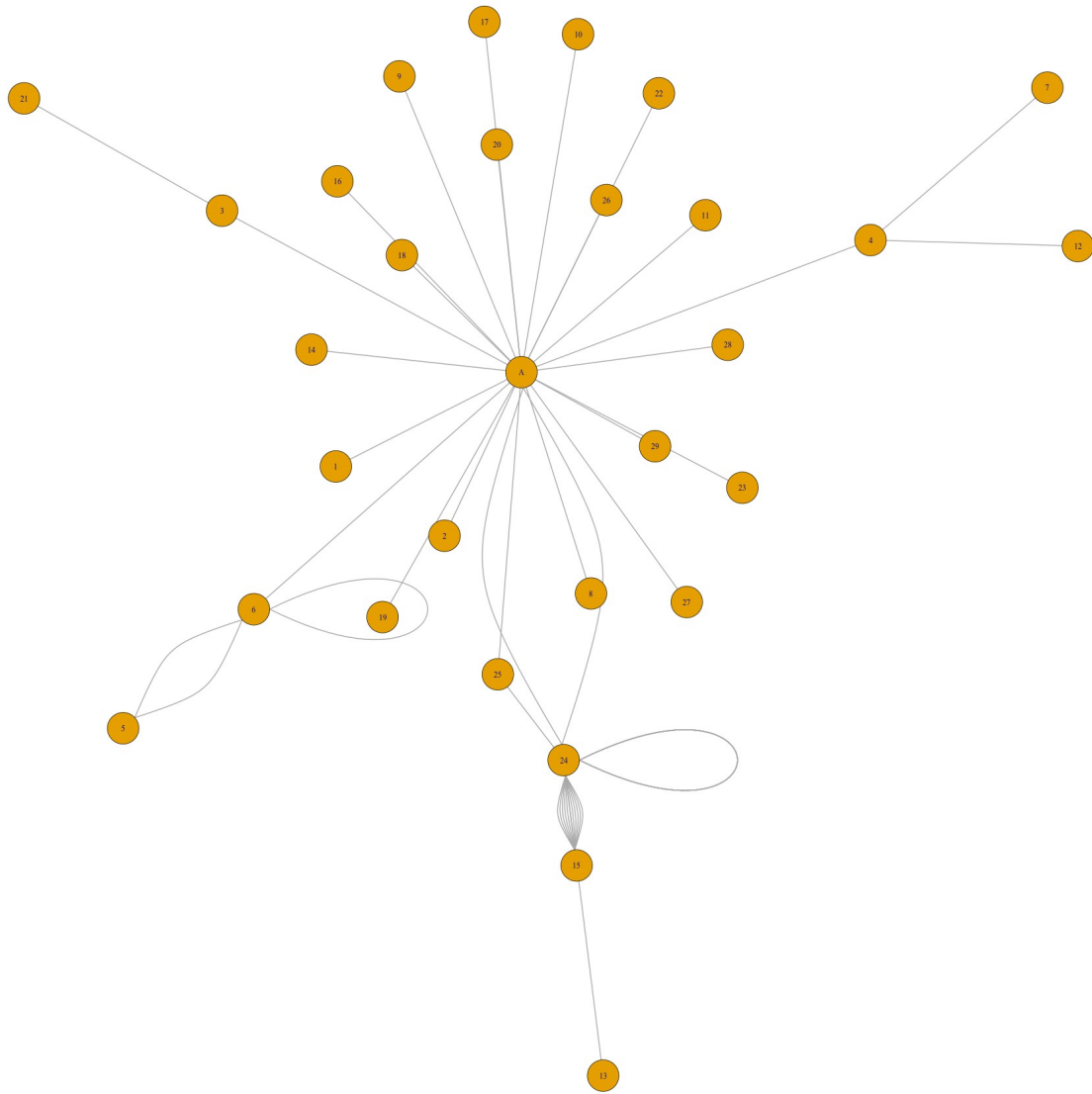
# Diagram Facebook network, comment n. 162 – diameter = 4

FB users, article n. 162 - diameter = 4



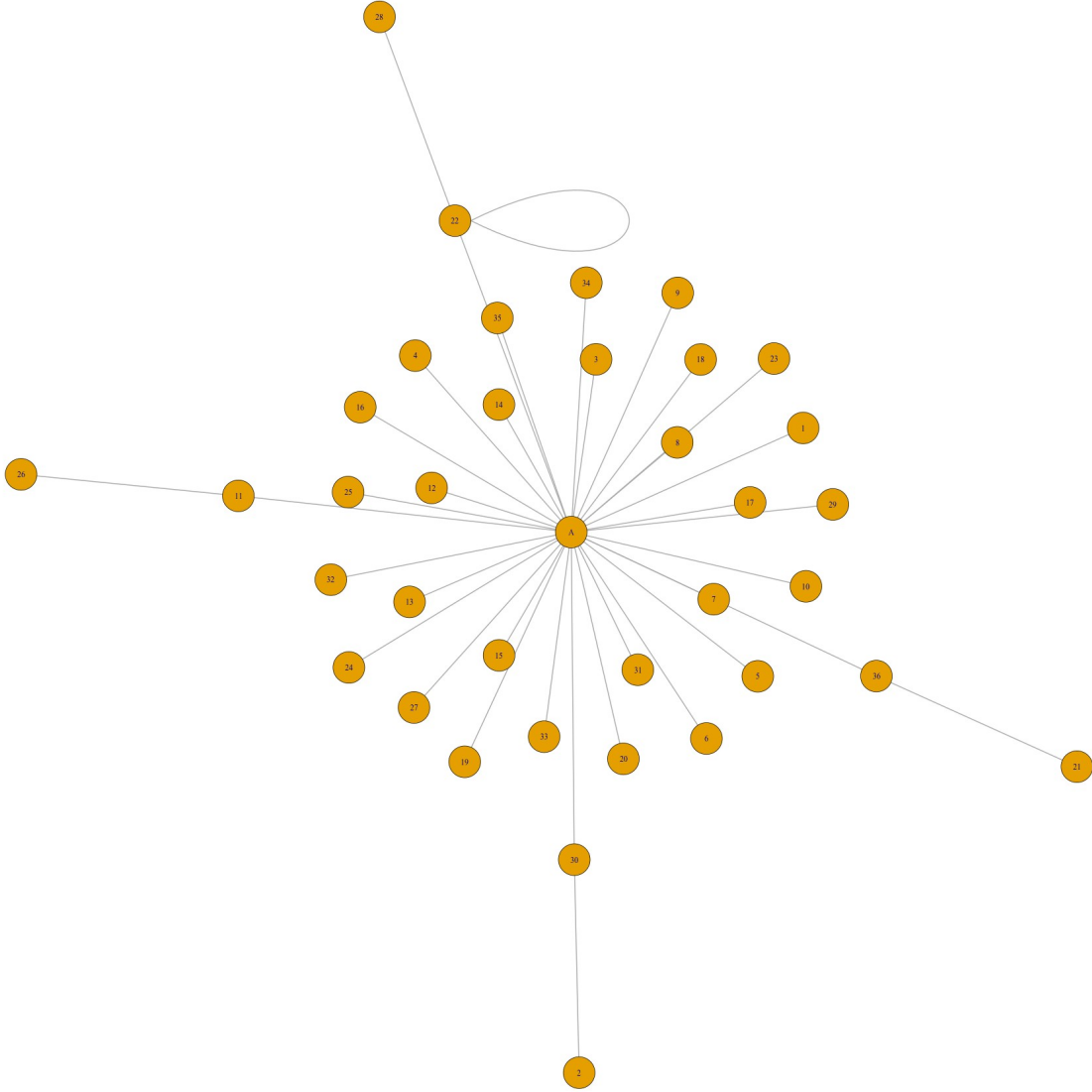
# Diagram Facebook network, comment n. 163 – diameter = 4

FB users, article n. 163 - diameter = 4



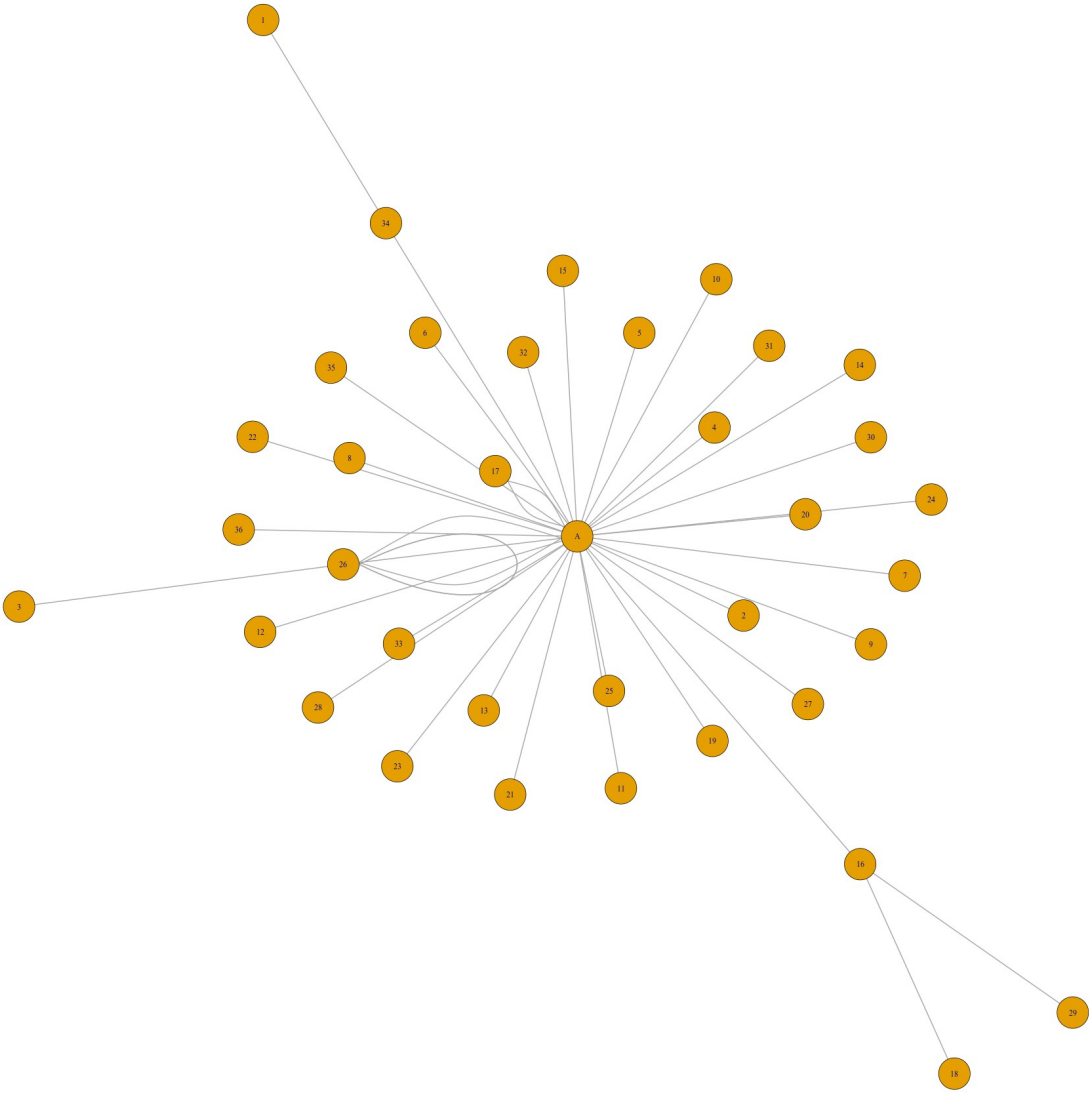
# Diagram Facebook network, comment n. 164 – diameter = 4

FB users, article n. 164 - diameter = 4



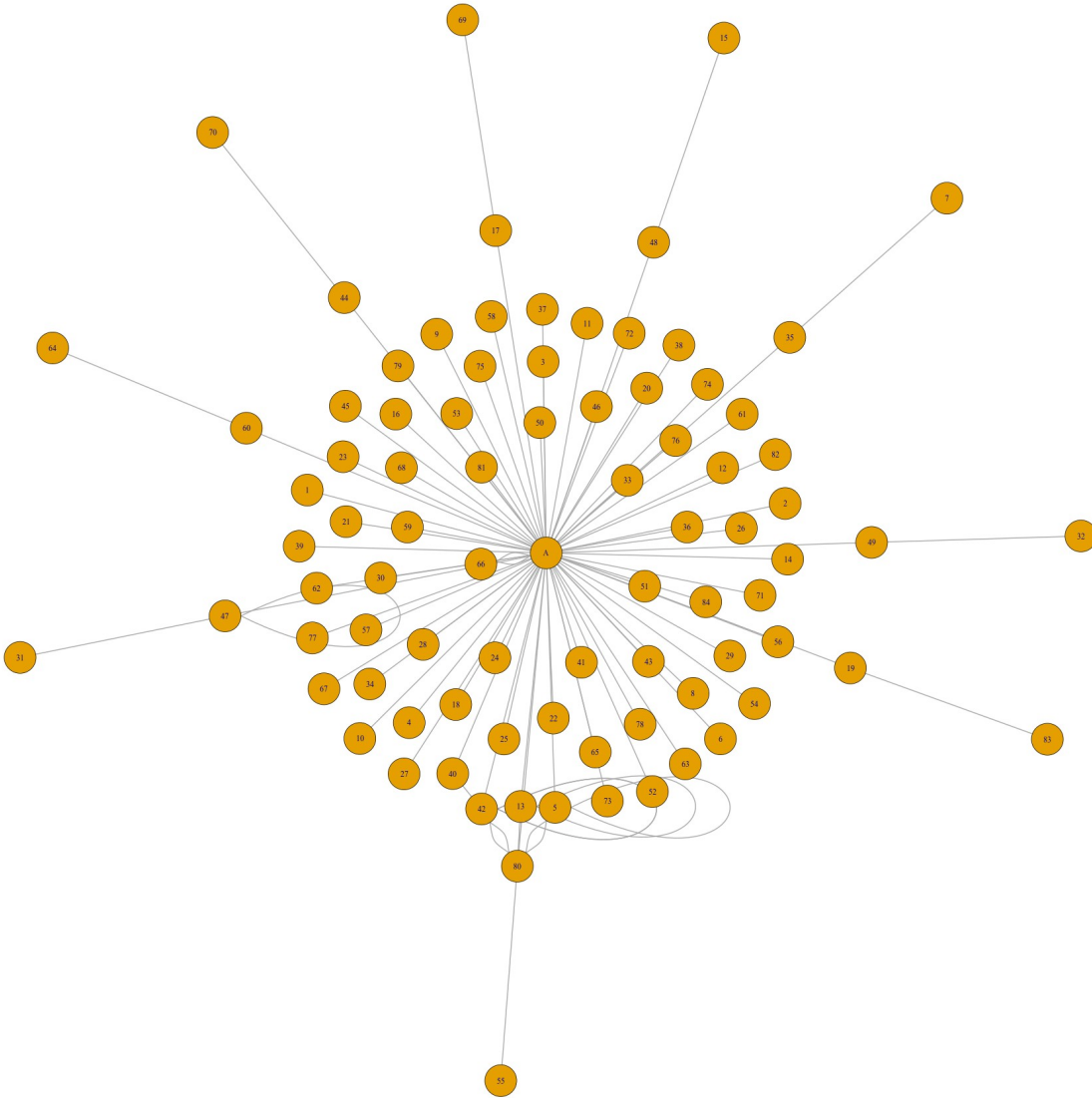
# Diagram Facebook network, comment n. 146 – diameter = 4

FB users, article n. 146 - diameter = 4



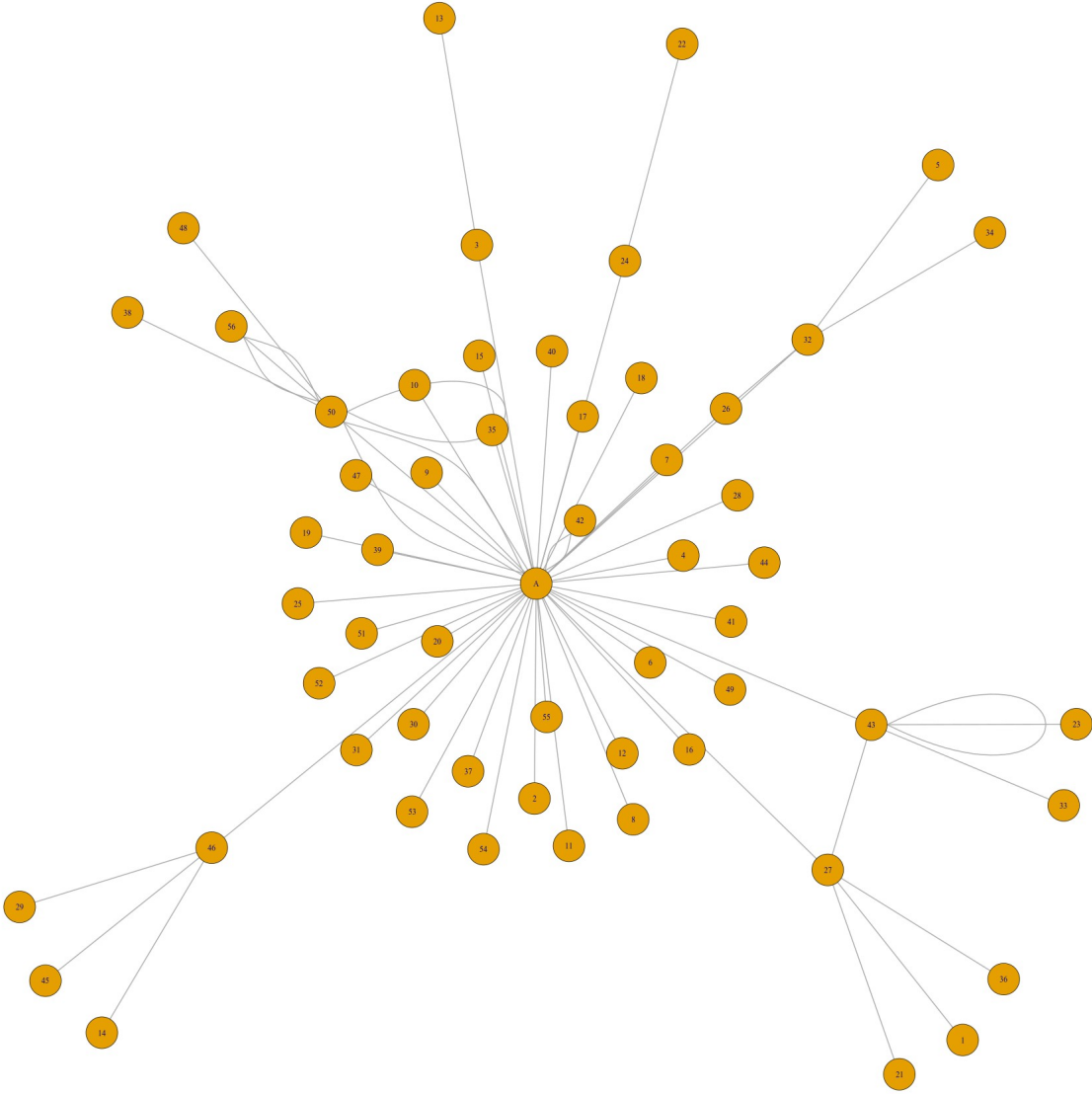
# Diagram Facebook network, comment n. 147 – diameter = 4

FB users, article n. 147 - diameter = 4



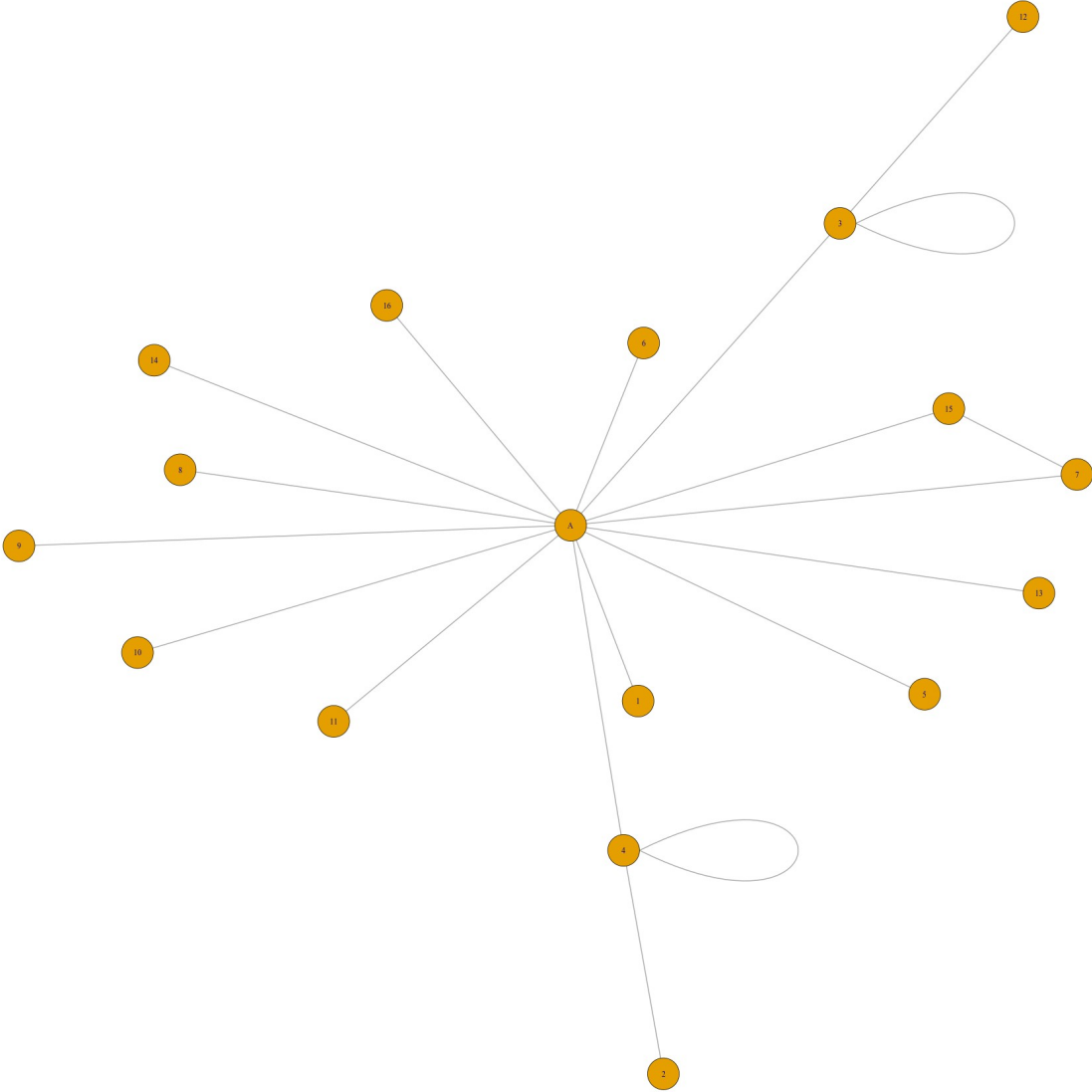
# Diagram Facebook network, comment n. 148 – diameter = 4

FB users, article n. 148 - diameter = 4



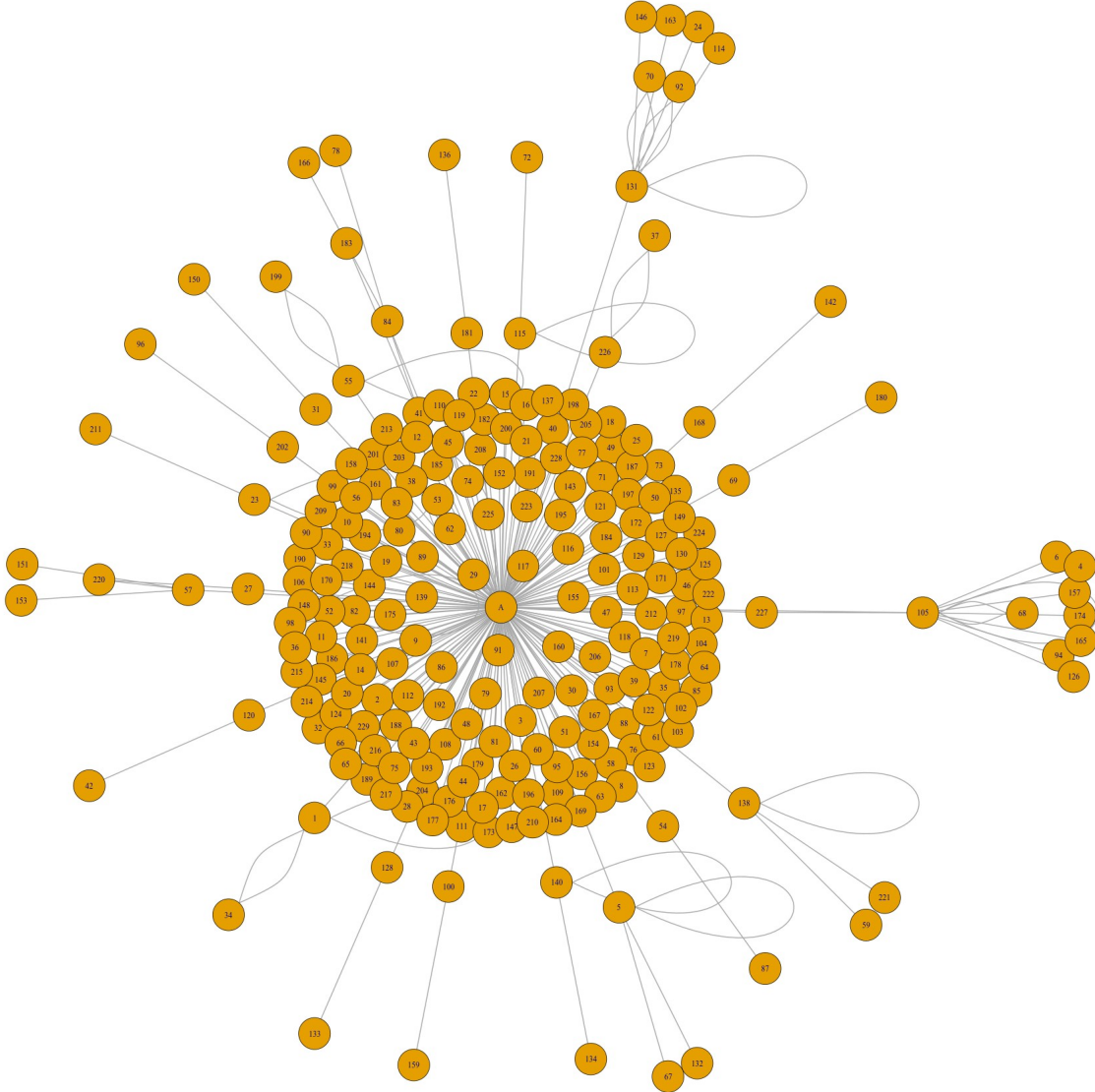
# Diagram Facebook network, comment n. 149 – diameter = 4

FB users, article n. 149 - diameter = 4



# Diagram Facebook network, comment n. 150 – diameter = 4

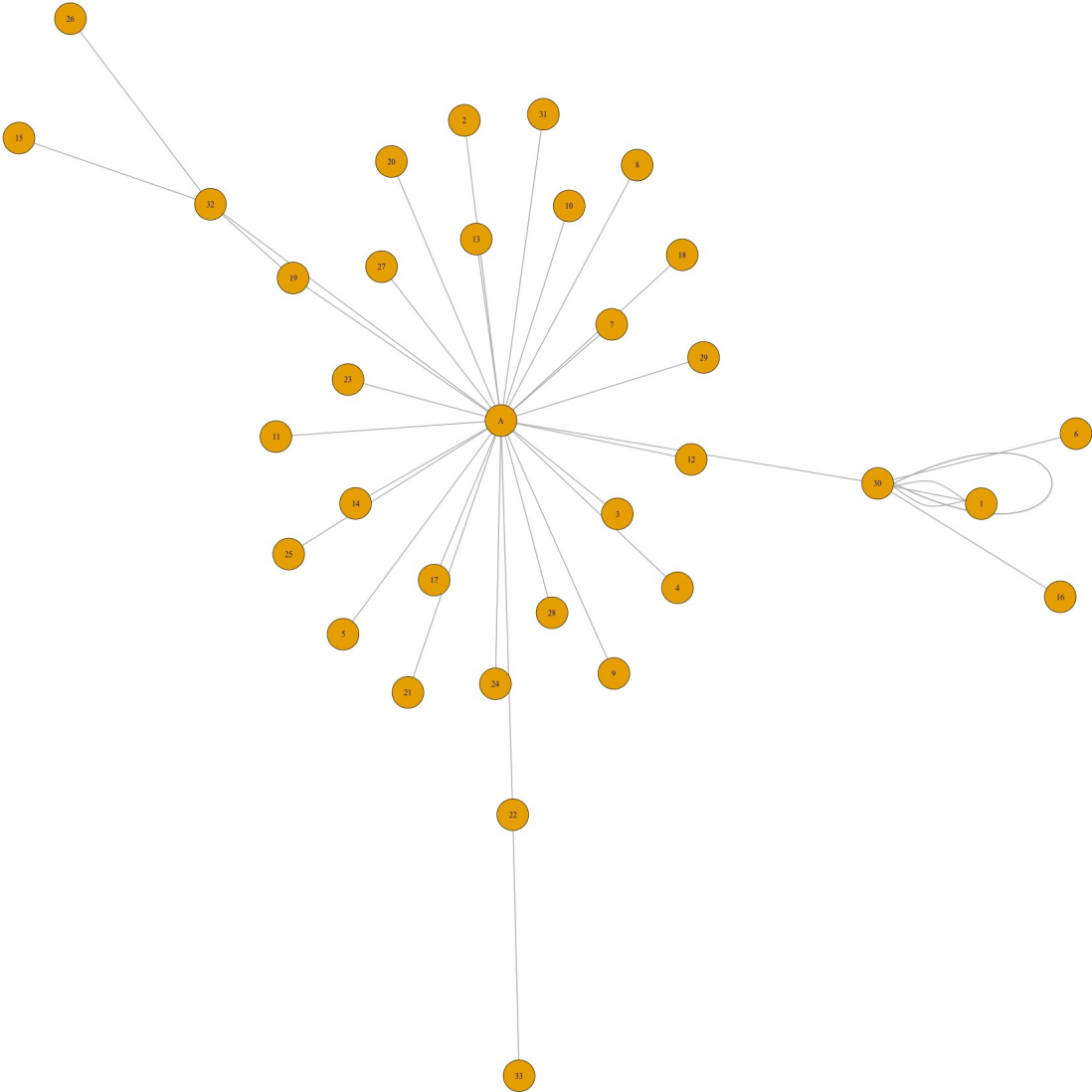
FB users, article n. 150 - diameter = 4





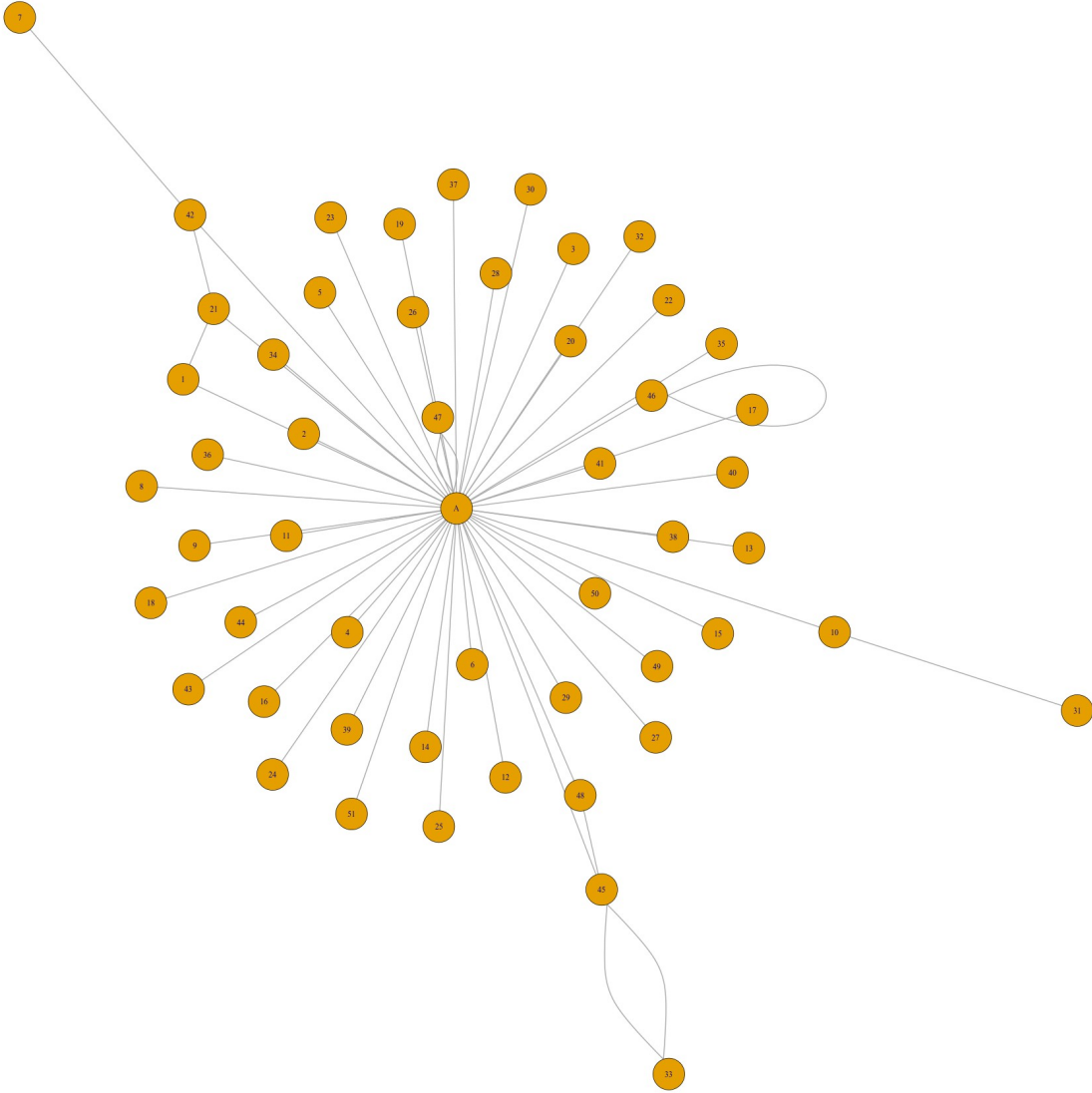
# Diagram Facebook network, comment n. 153 – diameter = 4

FB users, article n. 153 - diameter = 4



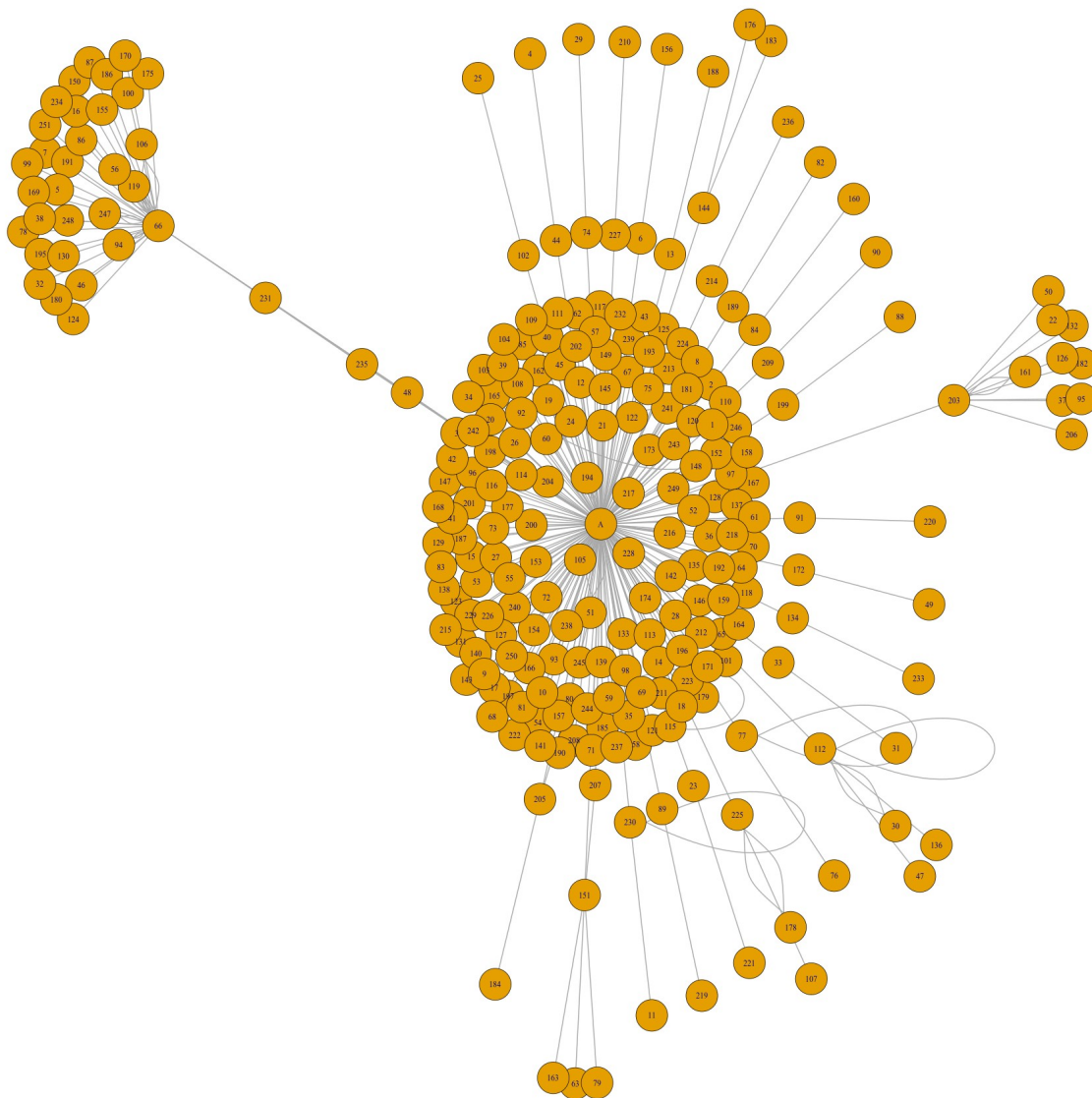
# Diagram Facebook network, comment n. 154 – diameter = 4

FB users, article n. 154 - diameter = 4



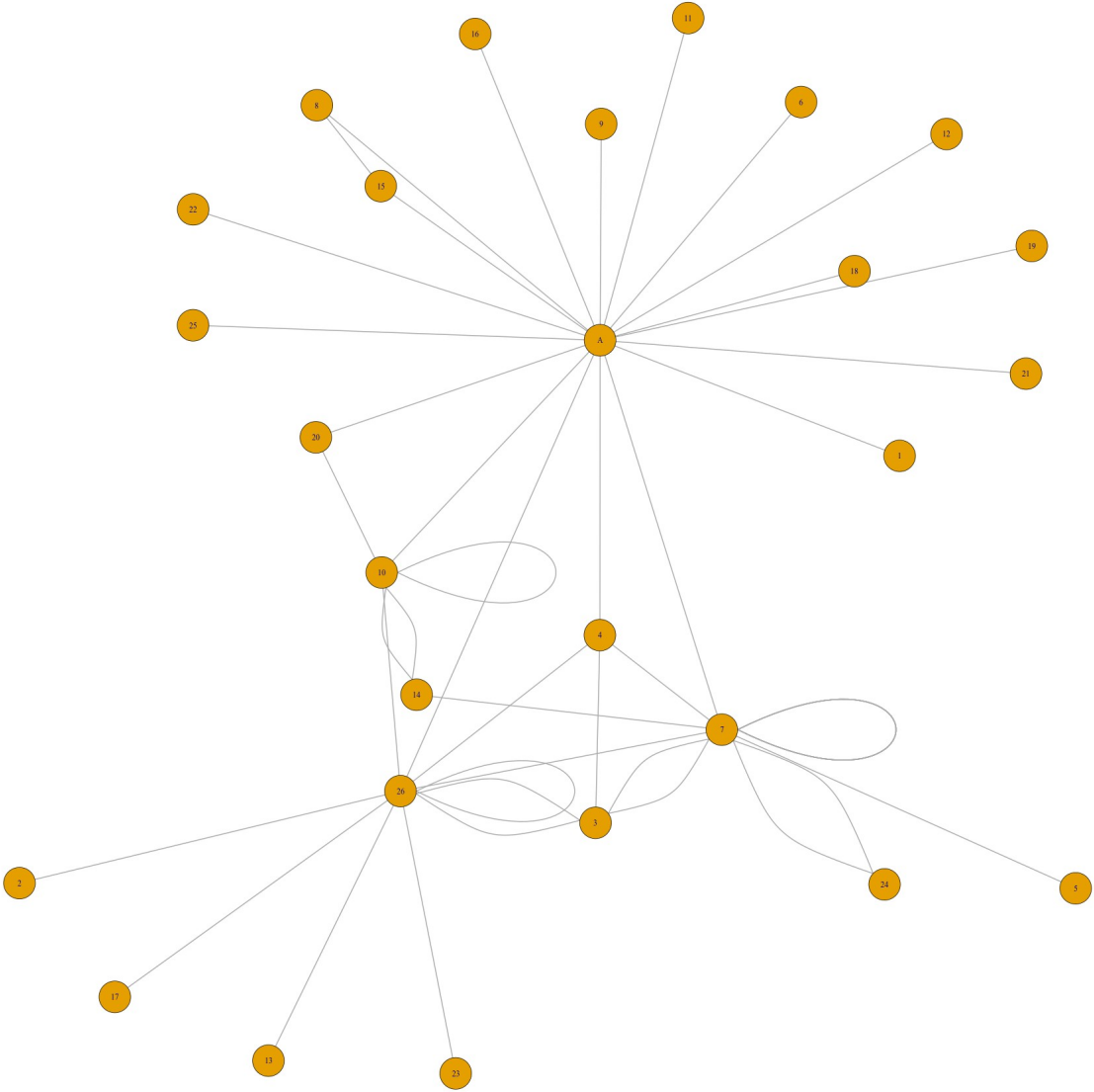
# Diagram Facebook network, comment n. 157 – diameter = 4

FB users, article n. 157 - diameter = 4



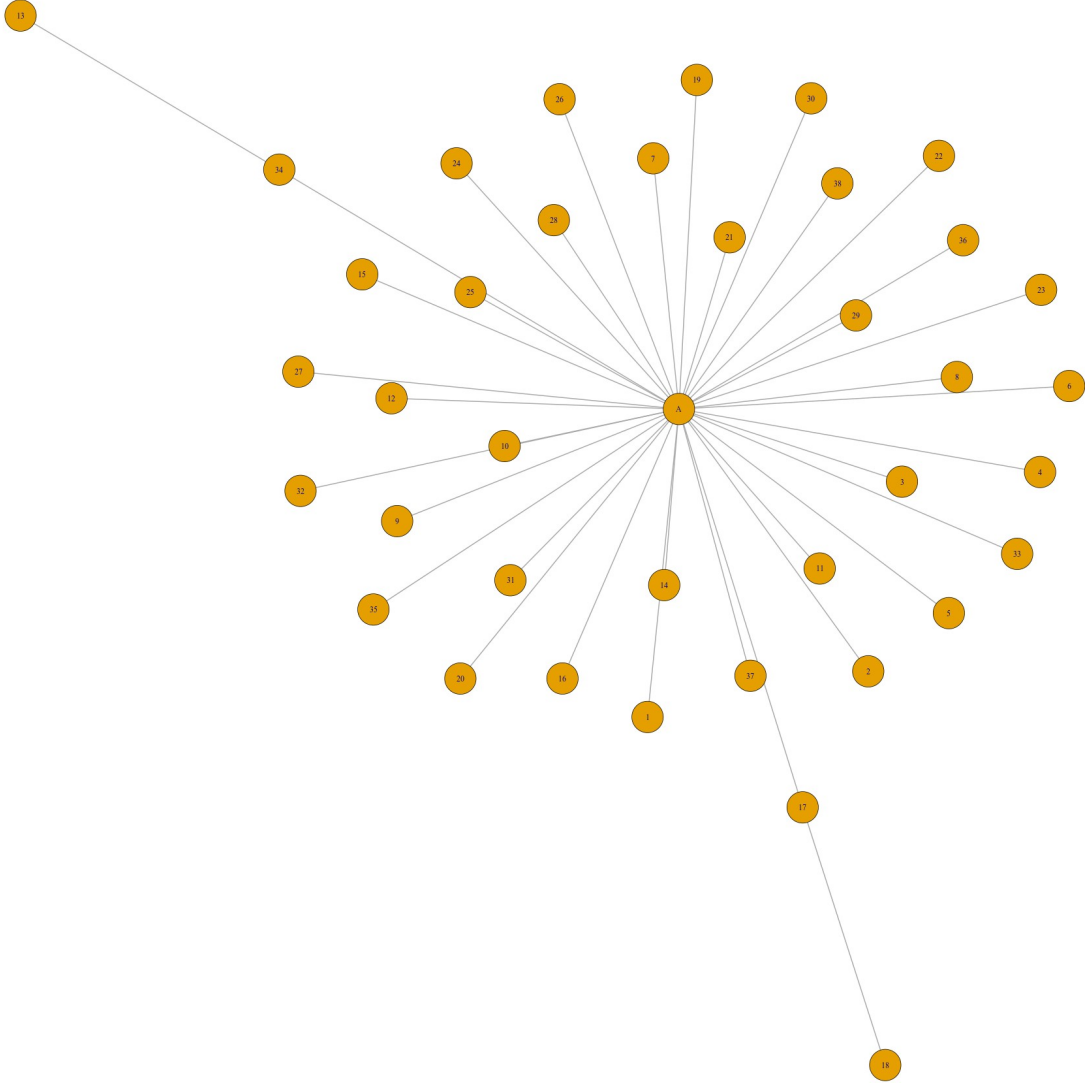
# Diagram Facebook network, comment n. 159 – diameter = 3

FB users, article n. 159 - diameter = 3



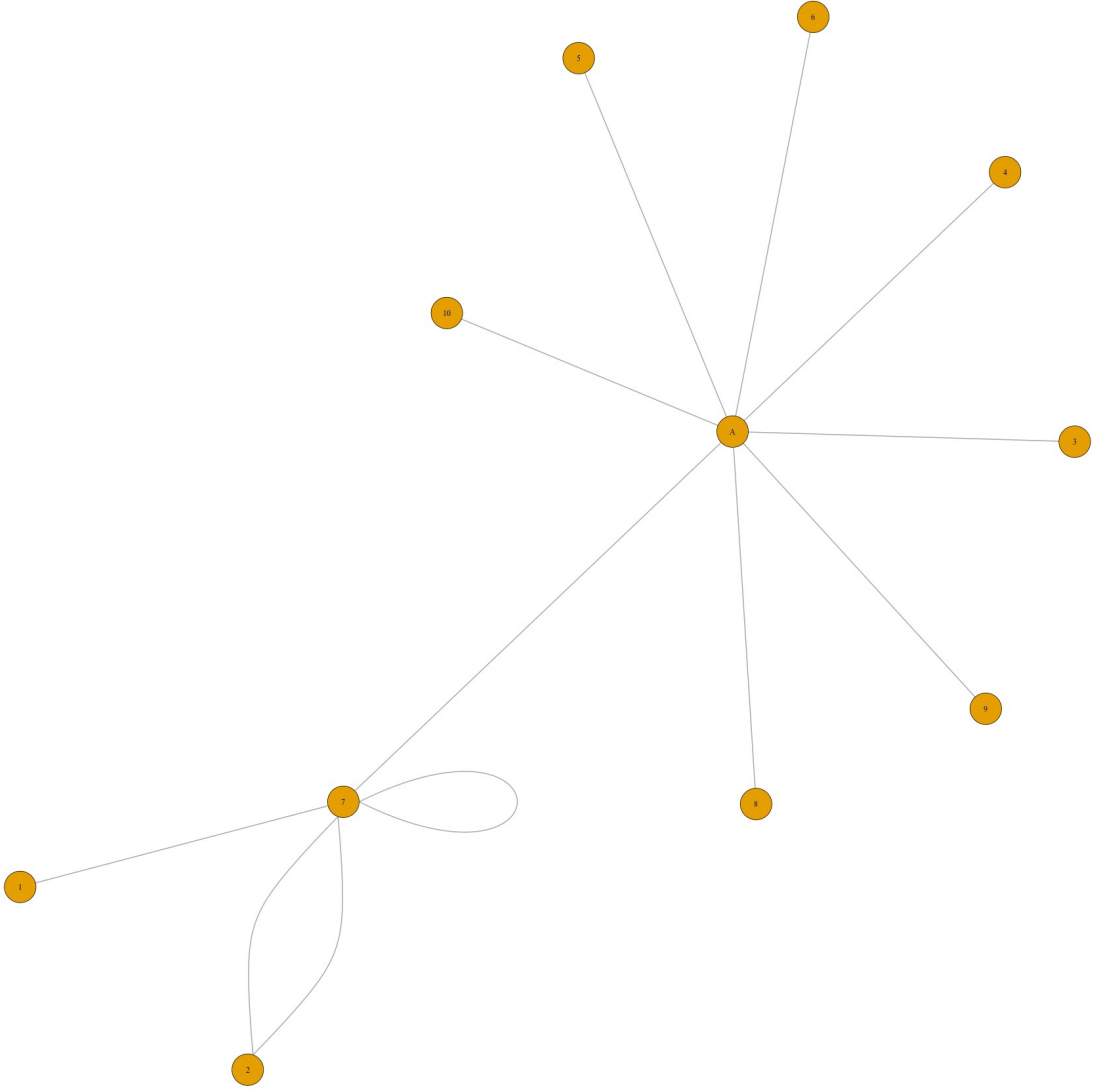
# Diagram Facebook network, comment n. 161 – diameter = 4

FB users, article n. 161 - diameter = 4



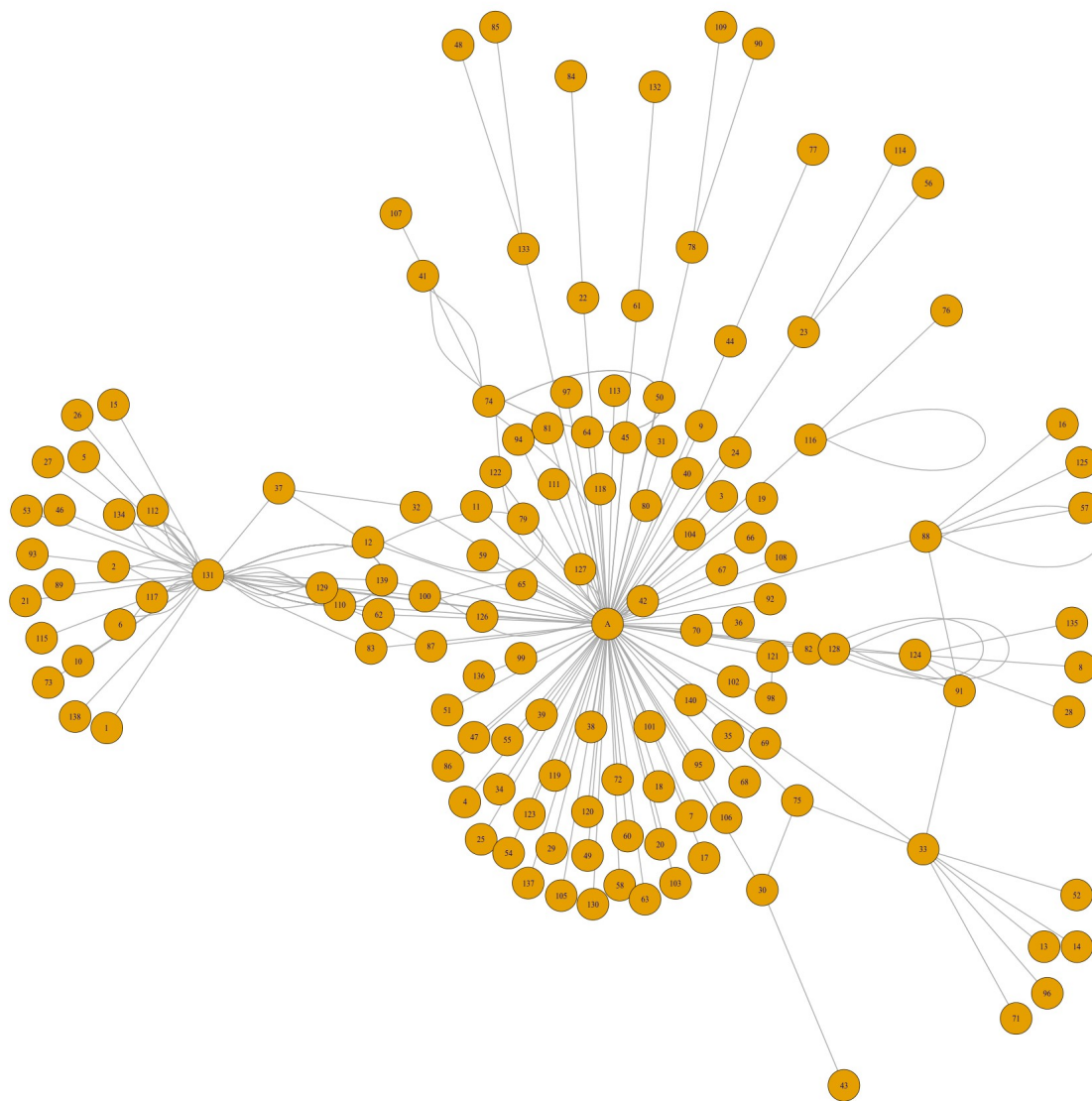
# Diagram Facebook network, comment n. 165 – diameter = 3

FB users, article n. 165 - diameter = 3



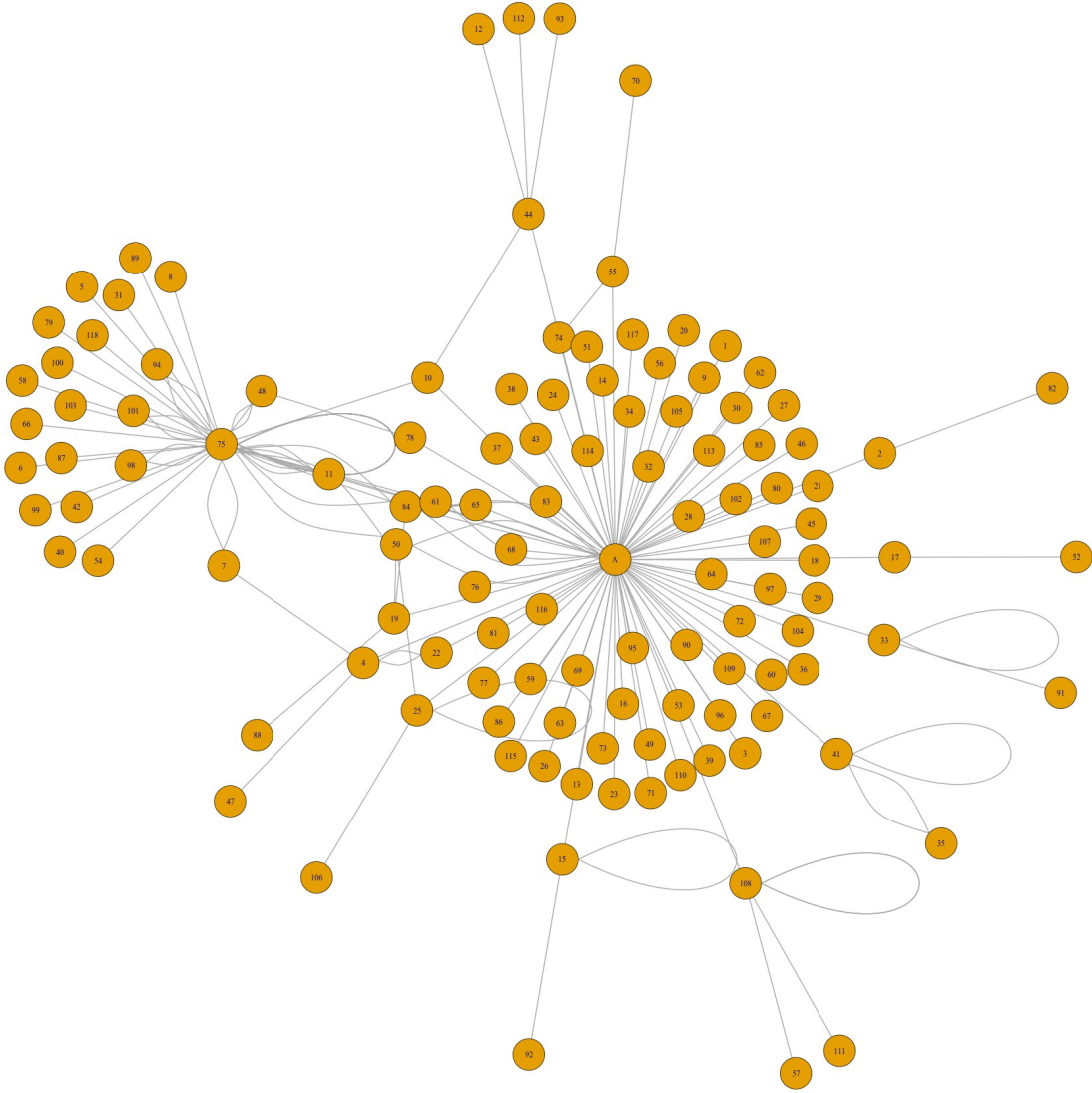
# Diagram Facebook network, comment n. 228 – diameter = 4

FB users, article n. 228 - diameter = 4



# Diagram Facebook network, comment n. 229 – diameter = 4

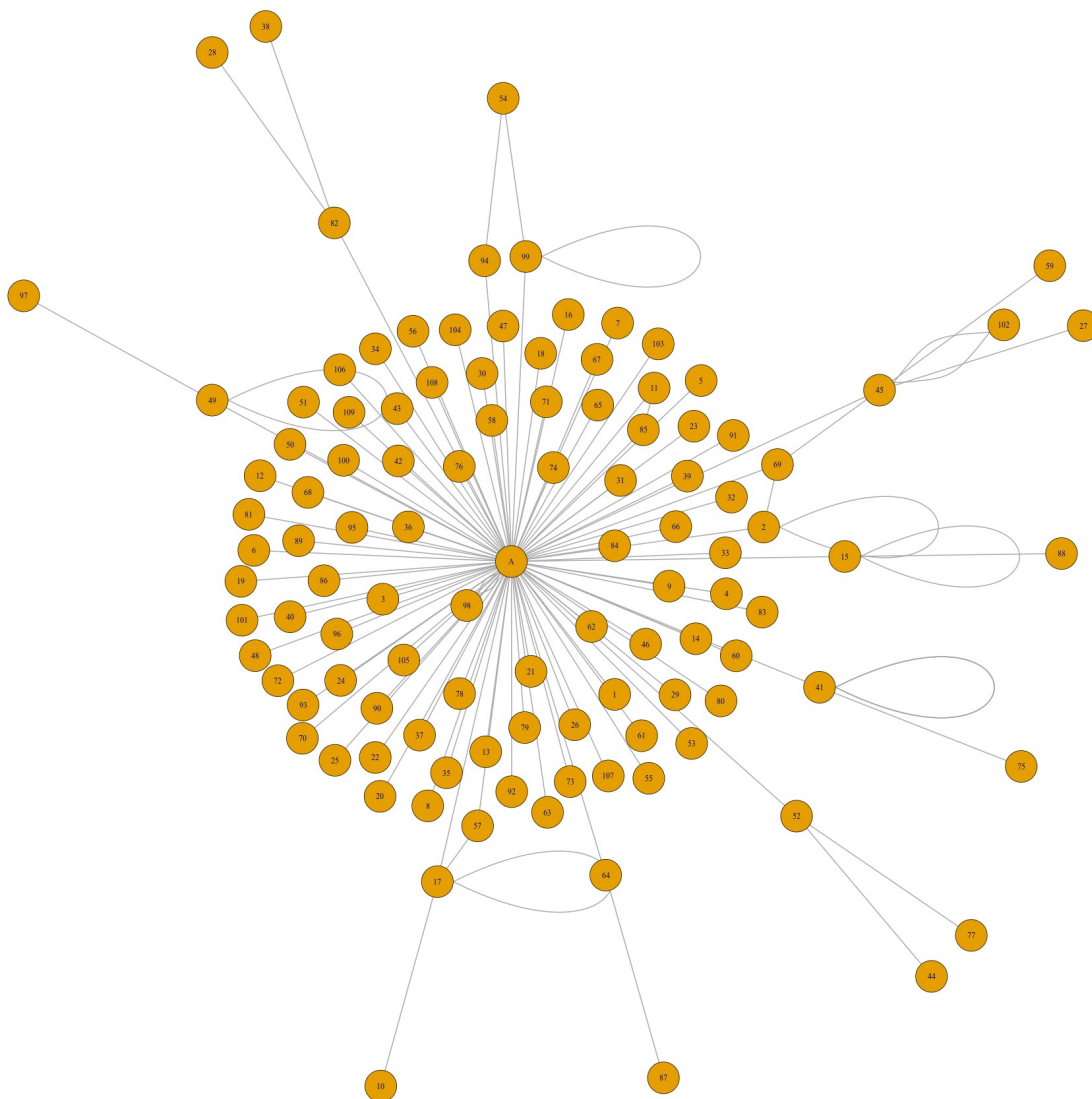
FB users, article n. 229 - diameter = 4





# Diagram Facebook network, comment n. 230 – diameter = 4

FB users, article n. 230 - diameter = 4



# Diagram Facebook network, comment n. 158 – diameter = 4

FB users, article n. 158 - diameter = 4

