Annex 3

Data extraction from Twitter's comments

Articles published on newspapers web page provide a link on Twitter feed in order to give the readers the possibility to comment them without the burden of managing the comment process.

Data extraction activity has to take in consideration the specificity of Twitter, first of all note that tweets are public, reader does not need to login in any private space in order to read tweets, comments are tweets that respond to a particular tweet. Tweets, since year 2017, are limited to 280 characters but often are written with less characters, they are direct real-time responds, the most similar form to a verbal comment. The relation between users is unidirectional, there is no need to be "friends" for opening a communication channel, as it happens with Facebook. Twitter relation model is based on "followers" and "followed", this relation is not mutual, a user can follow another user without being followed by the same user. Twitter specificity has a profound effect in its communication structure. Twitter design facilitates broadcasting of many to many, the term used is "multicasting" (Dhiraj, 2018, ch. 1).

Newspapers publish the article's link as title, in a space called "status", on top a title/link, often a picture with caption, then the hour and date of posting followed by the indication of the number of comments, retweets and emotional clicks of type "like", which means positive emotional clicks. A click is a clicked button. A retweet allow users to to forward tweets to audience known by the user forwarding the original tweet and attaching a comment on it. This is the mechanism that emulates the snow-ball effect, the best way to disseminate a tweet. Emotional click of type "like" works differently and is also conceptually different from the its Facebook counterpart, it is not limited to an emotional expression but allows to attach a comment on it. Finally, the comments are displayed and immediately readable after the above explained indicators.
Now a dilemma could rise, are the retweets and likes embedded texts to be considered as comments ? Of course they are, but they are of a particular nature. Retweets are a sort of very short introduction of the original article to a targeted audience, a way to give opportunity to the known audience of commenting themselves the original article, therefore should be considered as the attempt to open a new channel. Likes, some of them are not commented, are immediate emotional reactions that do not need comments, but are more a score on the original article. Moreover retweets and likes are not immediately displayed as the comments. In short they do not form a network but just a collection of reactions.
In order to be as much adherent as possible to the scope of the dissertation only the displayed comments will be considered as such.

Thanks to the above considerations in now possible to proceed with the analysis of the network created by Twitter mechanism around the specific newspaper article by extracting the relevant data from the set of comments. The best way to proceed is the use of Application Program Interface (API), a programmatic access to Twitter's tweets and users data through a programming language or scripts. R language makes available a library called "rtweet" (Kearney, 2018) which worked effectively under the tests done during the studies for this dissertation. Lamentably test were not successful due to a Twitter policy, Twitter API does not allow to get replies to a particular tweet, which was exactly what this study needs. (Twitter, 2019)
As it happened with Facebook it was necessary to scrape the texts from Twitter specific page in order to get the desired data. Another problem rose, Twitter pages are rendered by

a huge presence of Javascript which creates the HTML and CSS code, therefore they ae displayed differently depending on the browser. For the purpose of scraping the best rendered pages were observed using Opera web browser. The scraping has been performed manually because of the presence of high amount of Javascript code instead of straight HTML. The next picture is an example of part of Twitter comments page with names obfuscated for respecting users' privacy.

Scraping has been just a copy/paste exercise that resulted with a text including emojis and emoticons. Pictures where not included because they are elements outside the limits of this study.

The next text box shows the text scraped from the example above.

User screen name

@**username**
22 Jul 2018
More
Replying to @Telegraph
Deport them

2 replies 0 retweets 1 like
Reply 2 Retweet Like 1 Direct message
New conversation

User screen name

@**username**
22 Jul 2018
More
Read the article, the attackers was described as “a skinny white man” you daft racist bint.

0 replies 0 retweets 0 likes
Reply Retweet Like Direct message
End of conversation
New conversation

User screen name

@**username**
22 Jul 2018
More
Replying to @Telegraph
What the hell is this country coming to.

1 reply 0 retweets 0 likes
Reply 1 Retweet Like Direct message

User screen name

@**username**
22 Jul 2018
More
Cultural enrichment....

Translate Tweet
0 replies 0 retweets 0 likes
Reply Retweet Like Direct message
End of conversation

User screen name

@**username**
22 Jul 2018
More
Replying to @Telegraph
Those who attacked the child should deserve to put away for life!

0 replies 0 retweets 1 like
Reply Retweet Like 1 Direct message

The analysis resulted in the identification of the following pattern:

```
@username
 22 Jul 2018
More
Replying to @Telegraph
Those who attacked the child should deserve to put away for life!

0 replies 0 retweets 1 like
```

where all data can be extracted by means of a regular expression, which is the definition of the search pattern in a programmatic context. In order to simplify the extraction of data, all leading and trailing spaces of each line were trimmed. The expression is the following:

```
(\s@[A-z0-9]+)\n+([0-9]{1,2}\s[A-z]{3}\s[0-9]{4})\n+Replying\sto\s@[A-z0-9]+\n+
(.+\n+)(\d{1,4})\srepl.+\s(\d{1,4})\sretweet.*\s(\d{1,4})\slike.*
```

Each parenthesis encloses the capture group:

- username
  (\s@[A-z0-9]+)
- date
  ([0-9]{1,2}\s[A-z]{3}\s[0-9]{4})
- Comment
  (.+\n+)
- number of replies
  (\d{1,4})
- number of retweets
  (\d{1,4})
- number of likes
  (\d{1,4})

Extracted data are stored in a table with the following format

un	cm	nr	nt	nl

where:

un = user name
cm = comment
nr = number of replies
nt = number of retweets
nl = number of likes

The next step will assign an id number to each user for privacy purposes and draw the network diagram, by means of R programming, for each table corresponding to the set of comments.

```
for (ind in 1:length(df_files)) {
      df_file <- df_files[ind]
      df <- read_csv(paste0(dir_tw_csv, df_file))
      fqun <- as.data.frame(table(df$un)) # get unique records of user names and
calculates frequencies
      colnames(fqun)[1] <- 'name'
      fqun <- mutate(fqun, id = rownames(fqun)) # adds column id and populates
it
      df$un <- as.factor(df$un)     # otherwise grr::matches fires error
      dictionary_fqun_df <- grr::matches(fqun$name, df$un)
      dictionary_fqun_df_sorted <-
dictionary_fqun_df[order(dictionary_fqun_df[,2]),]
      df <- cbind(df,dictionary_fqun_df_sorted[,1])
      colnames(df)[6] <- 'id'
      df$ir <- '' # add blank column ir = user name id reply
      # ini build network 2 levels
      rows_df <- nrow(df)
      indf <- 1
      while (indf <= rows_df) {
            nr <- as.integer(df[indf,3])
            if (nr > 0) {
                  df[indf,7] <- 'A'
                  curid <- df[indf,6]
                  rp <- 0
                  while (nr > rp) {
                        rp <- rp + 1
                        newind <- indf + rp
                        df[newind,7] <- curid
                  }
                  indf <- indf + nr + 1
            } else {
                  df[indf,7] <- 'A'
                  indf <- indf + 1
            }
      }
      # end build network 2 levels
      # ini draw graphs
      ph <- vector('character');
      for (indf in 1:rows_df) {
            ph <- c(ph,c(df[indf,7],df[indf,6]))
      }
      gr <- graph(ph,directed = FALSE)
      di <- diameter(gr)
      deg <- degree(gr, mode="all")
      deg.dist <- degree_distribution(gr, cumulative=T, mode="all")
      png(filename = paste0(dir_graph,df_file,'_tw.png'),width = 1920, height =
1920)
      plot(gr,vertex.size = 12, vertex.label.cex = 6, edge.width = 2,
            main = paste0('Twitter users, comments n. ', df_file, ' - diameter =
', di))
      dev.off()
      # end draw graphs
}
```

The next pages contain network representation of almost all Twitter users comments, some of them were discarded because of incorrect scraping.

Diagram Twitter network, comment n. 167 – diameter = 3

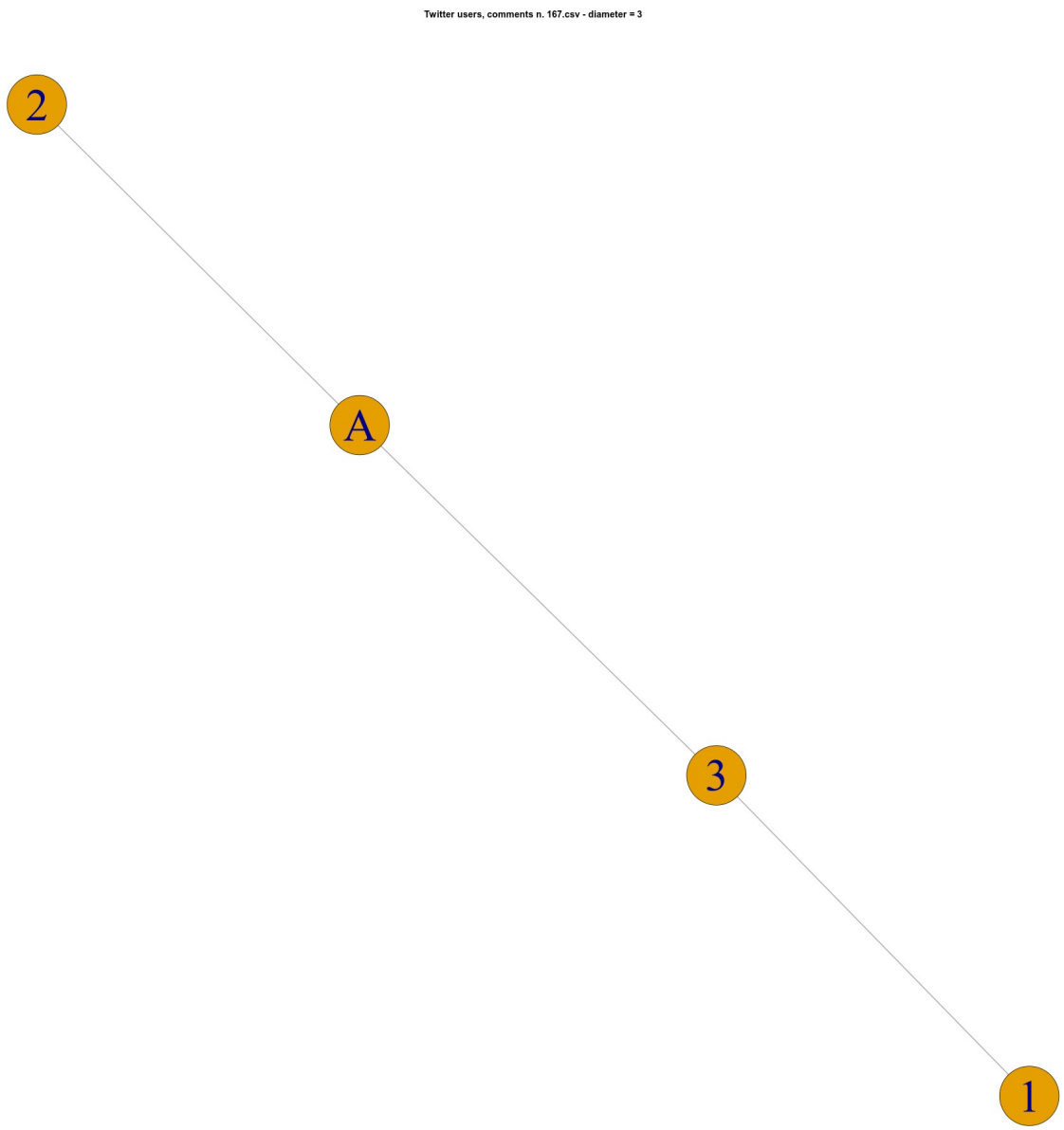


Diagram Twitter network, comment n. 169 – diameter = 4

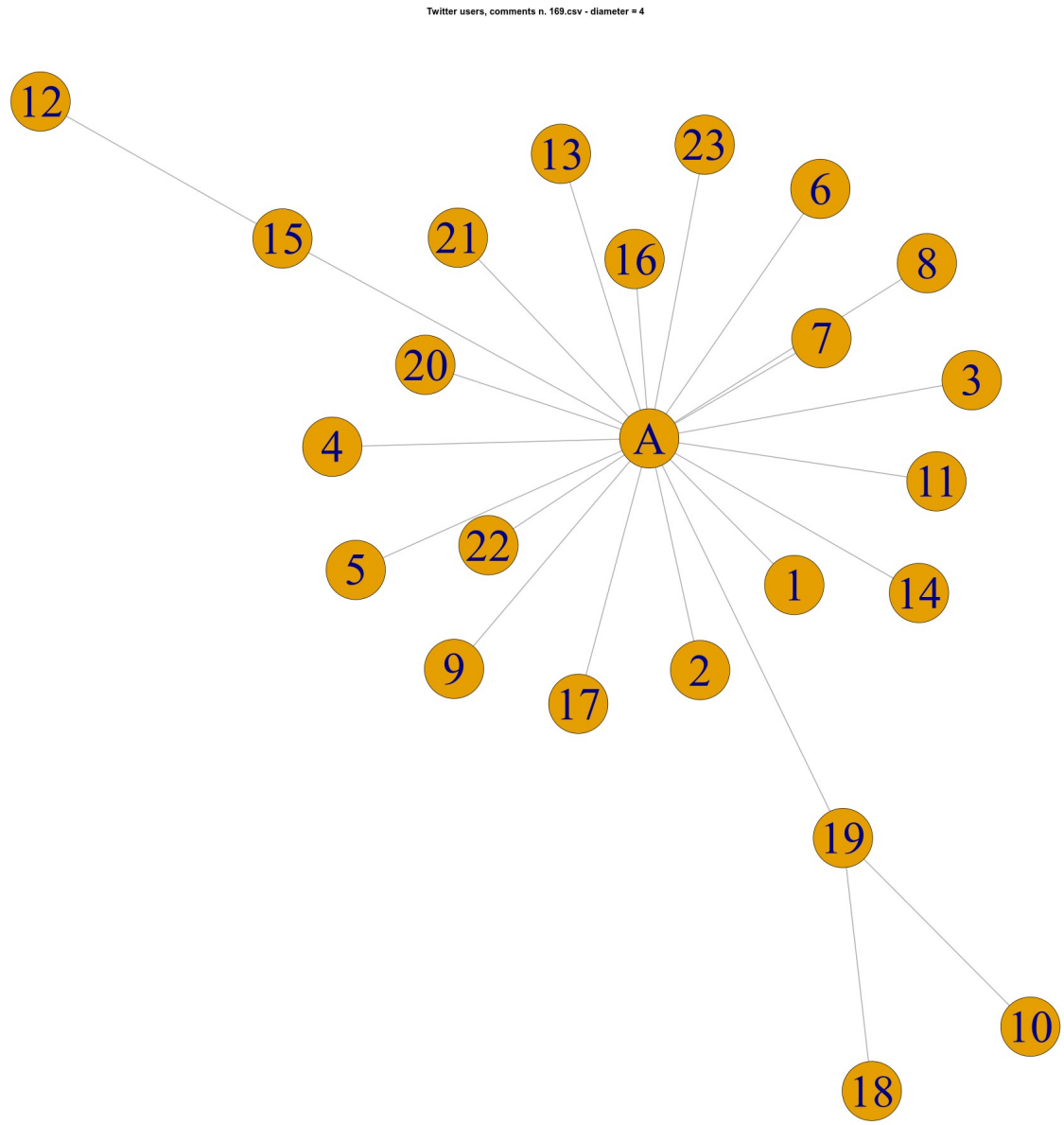


Diagram Twitter network, comment n. 170 – diameter = 1

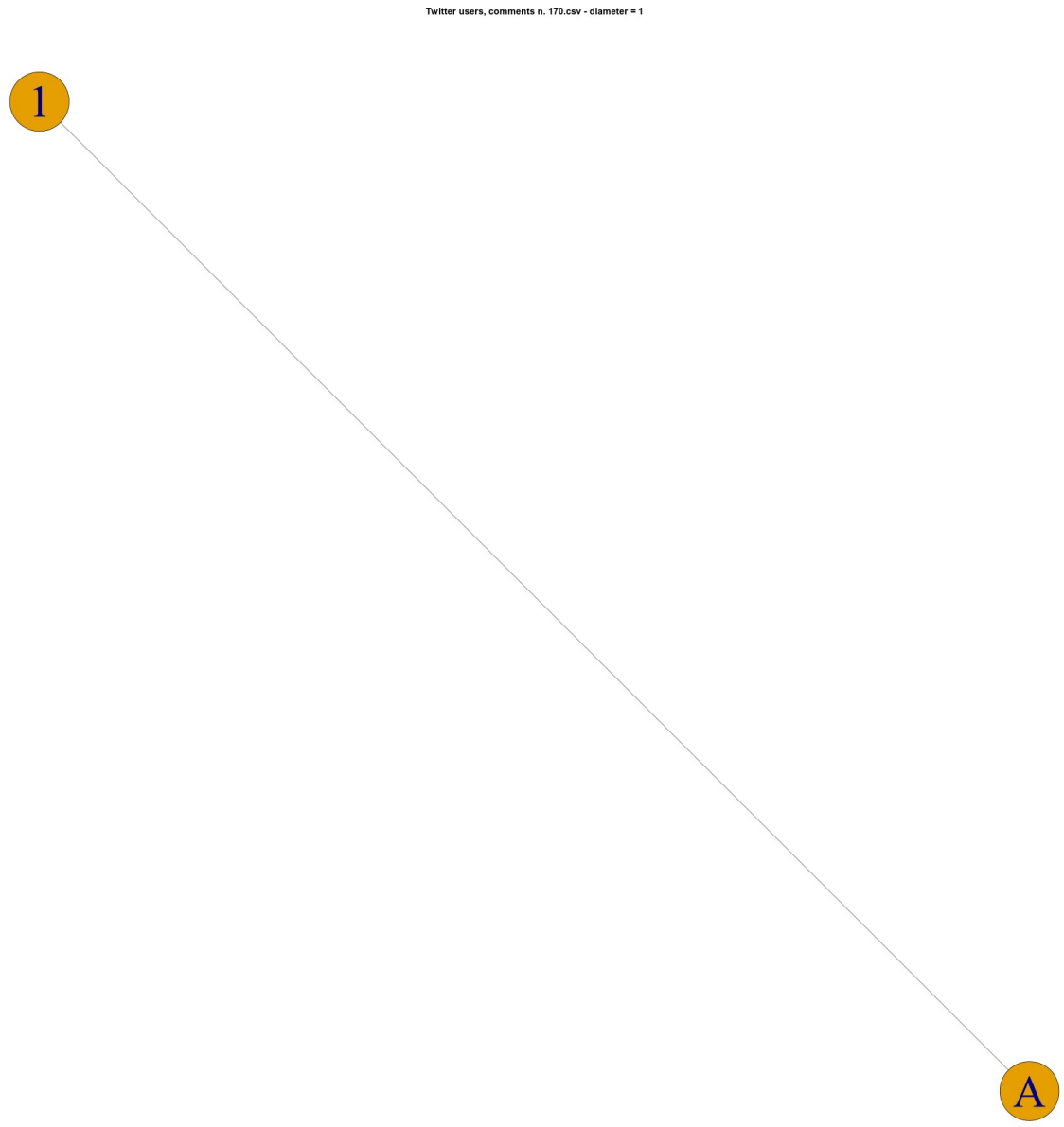


Diagram Twitter network, comment n. 171 – diameter = 2

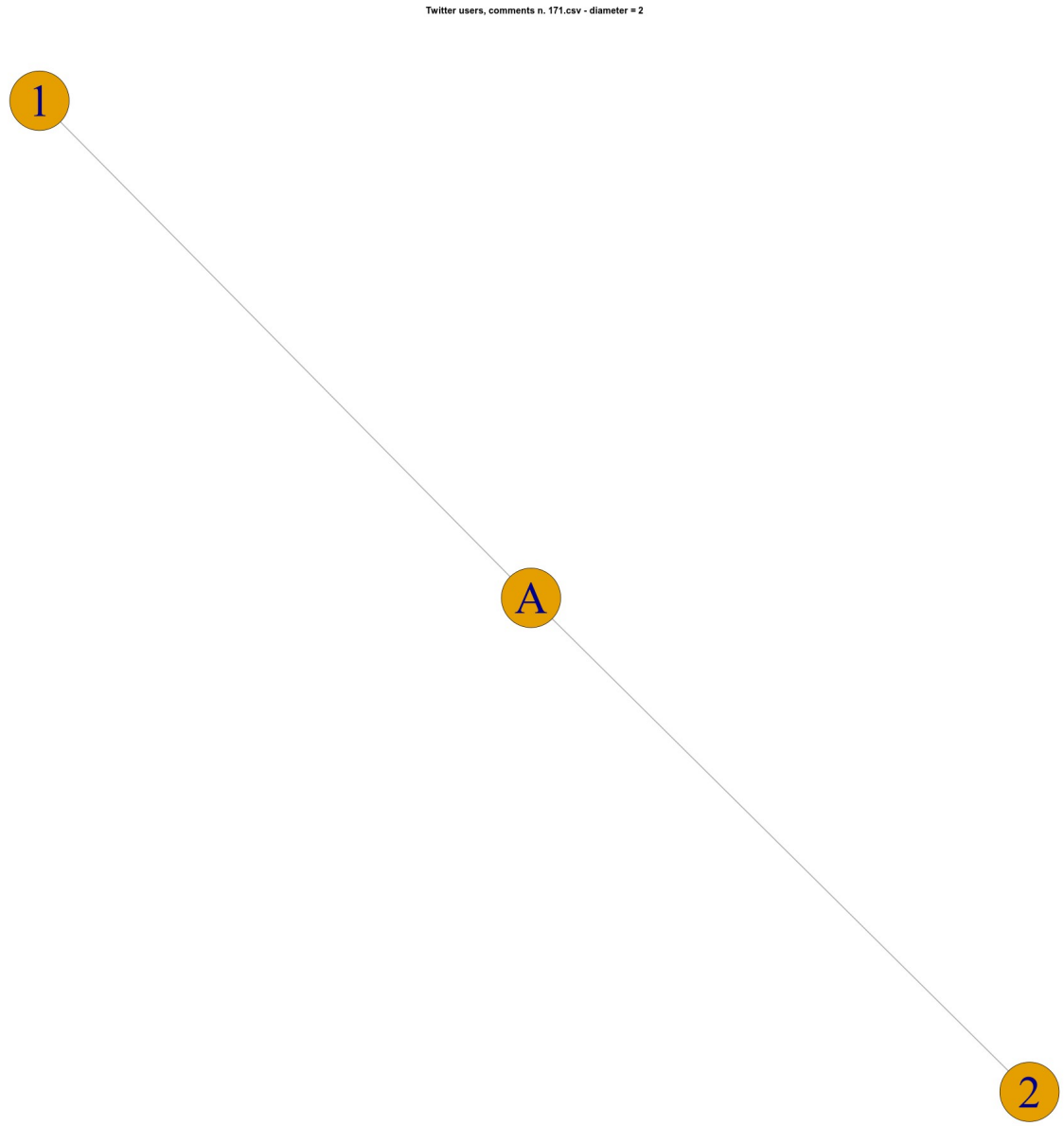


Diagram Twitter network, comment n. 172 – diameter = 1

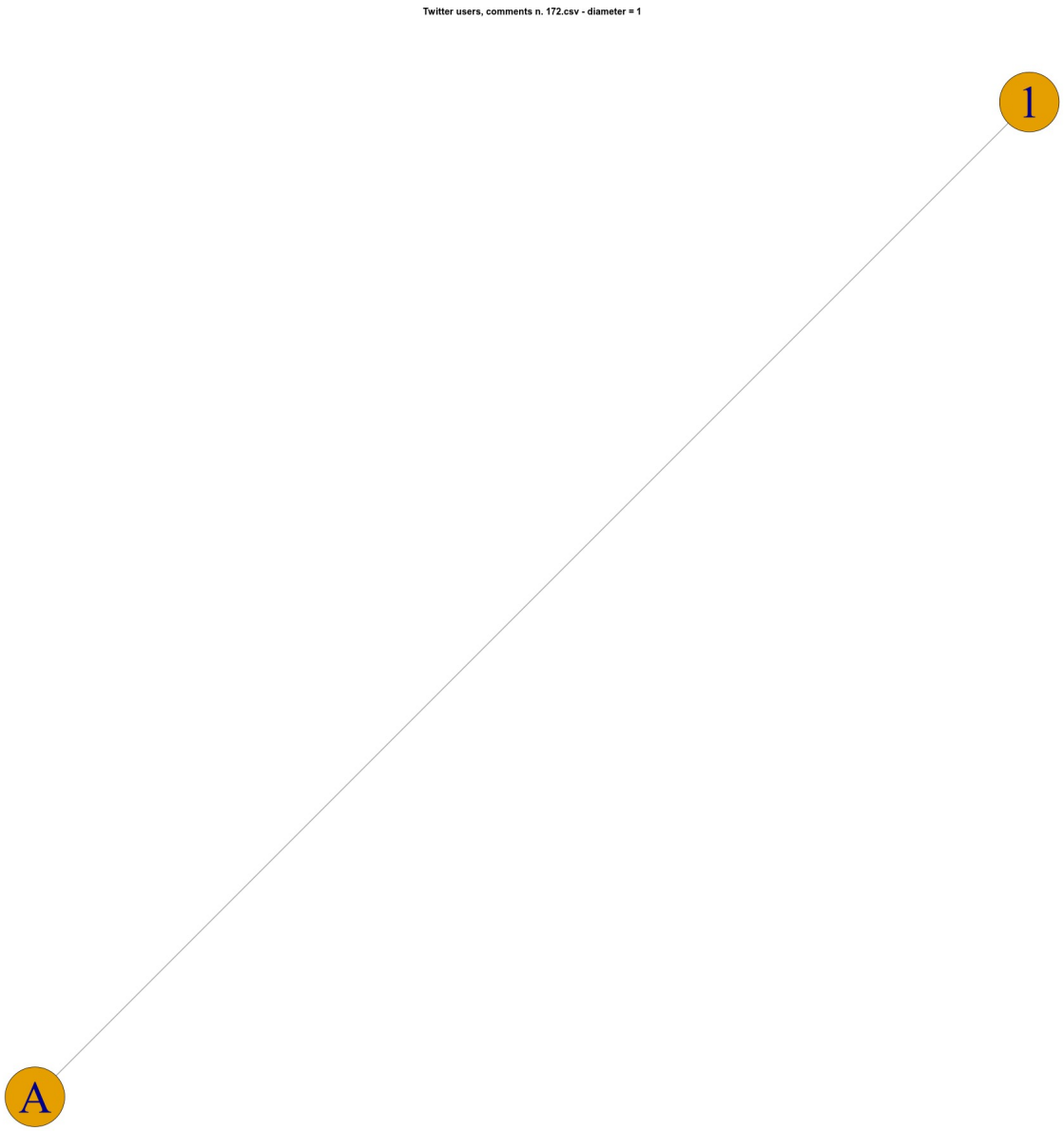


Diagram Twitter network, comment n. 173 – diameter = 2

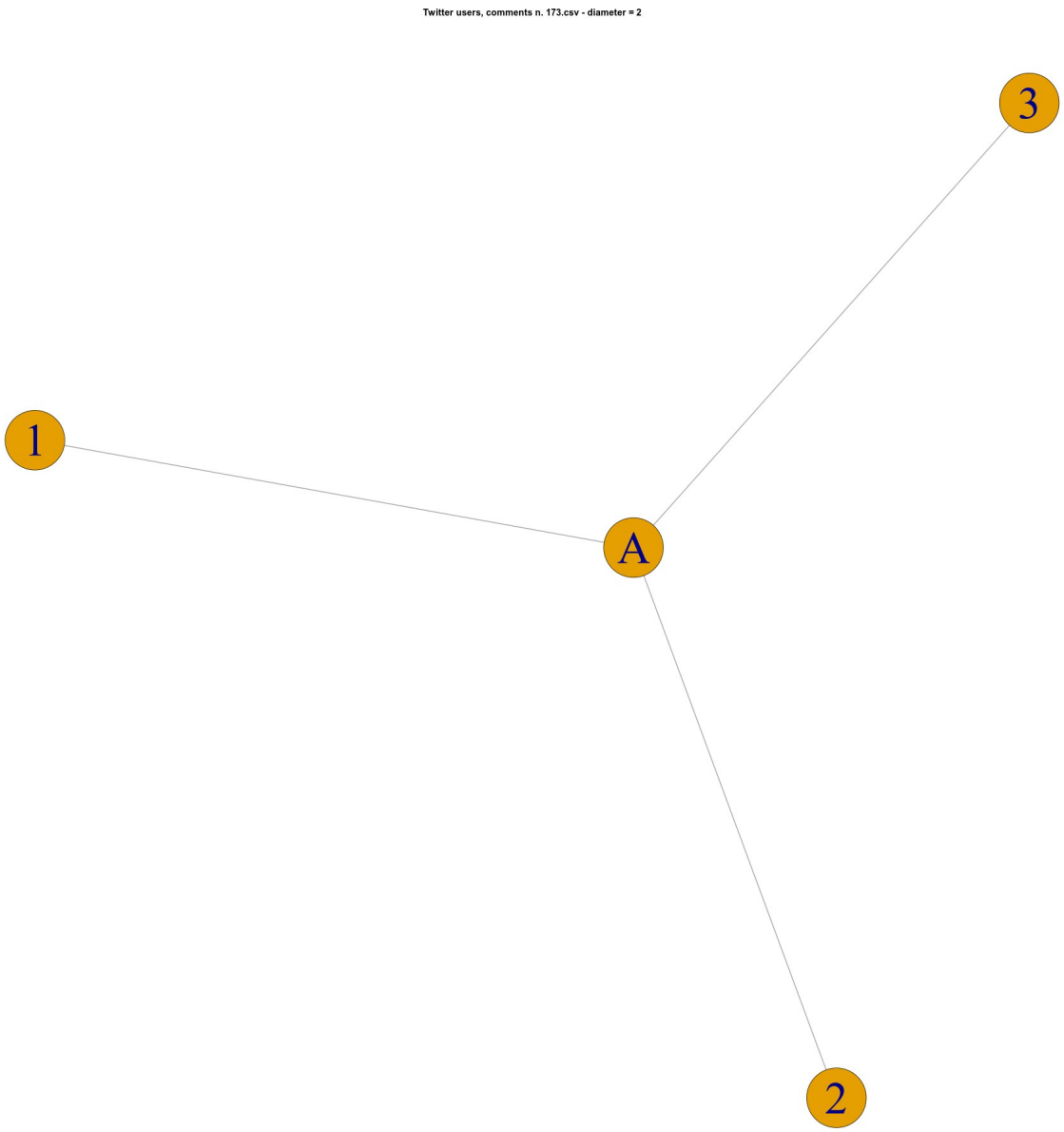


Diagram Twitter network, comment n. 175 – diameter = 1

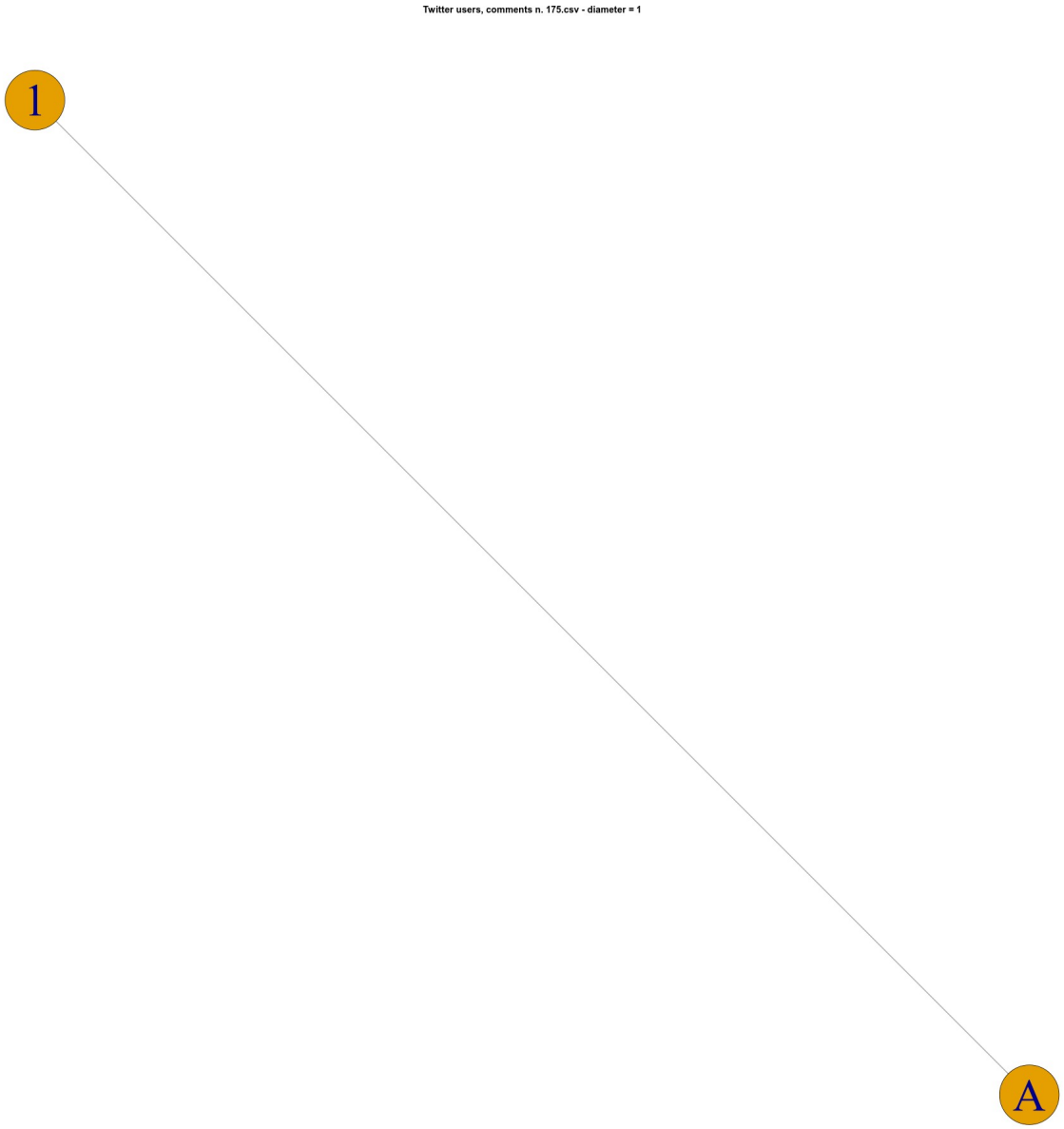


Diagram Twitter network, comment n. 176 – diameter = 1

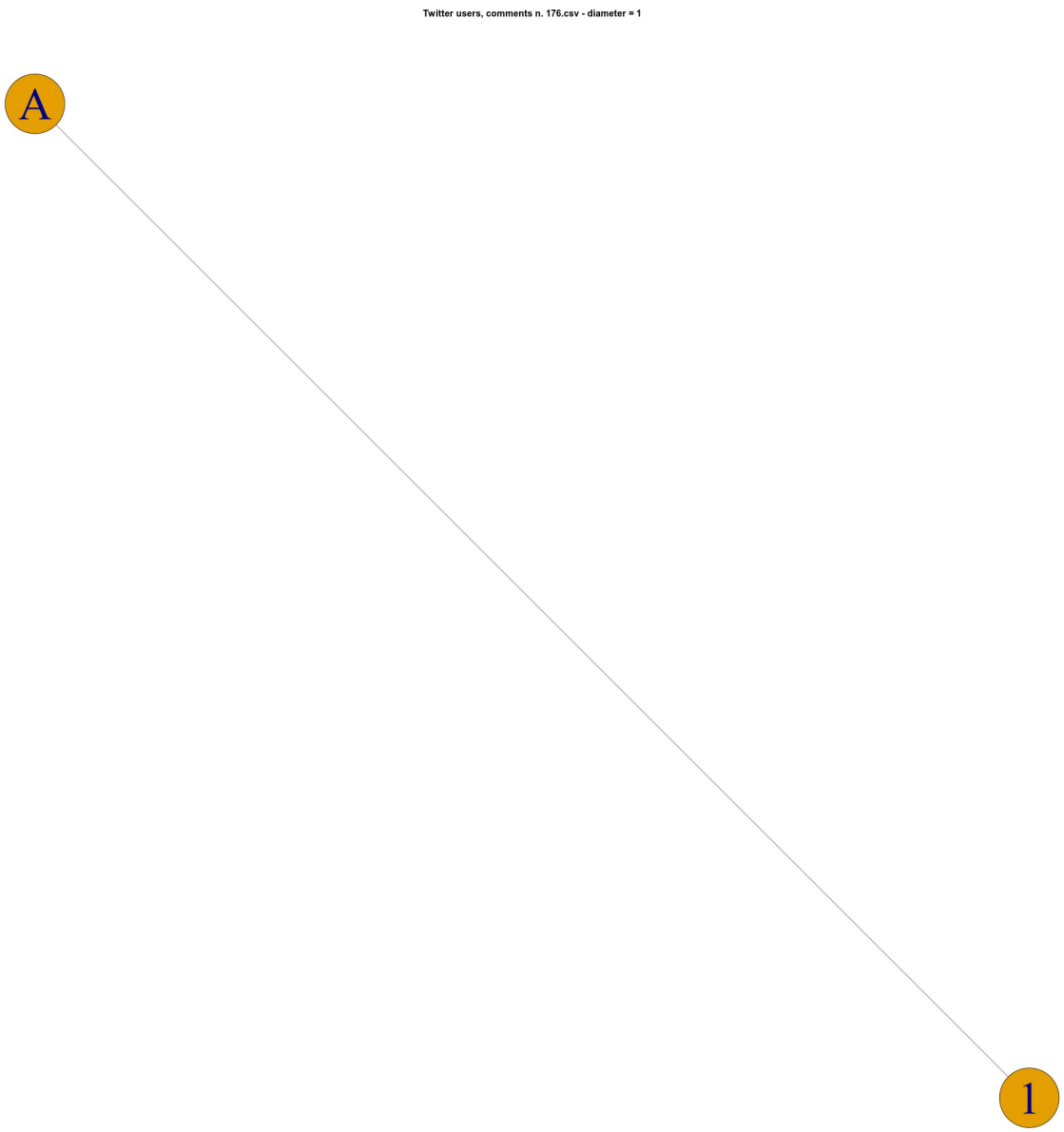


Diagram Twitter network, comment n. 178 – diameter = 2

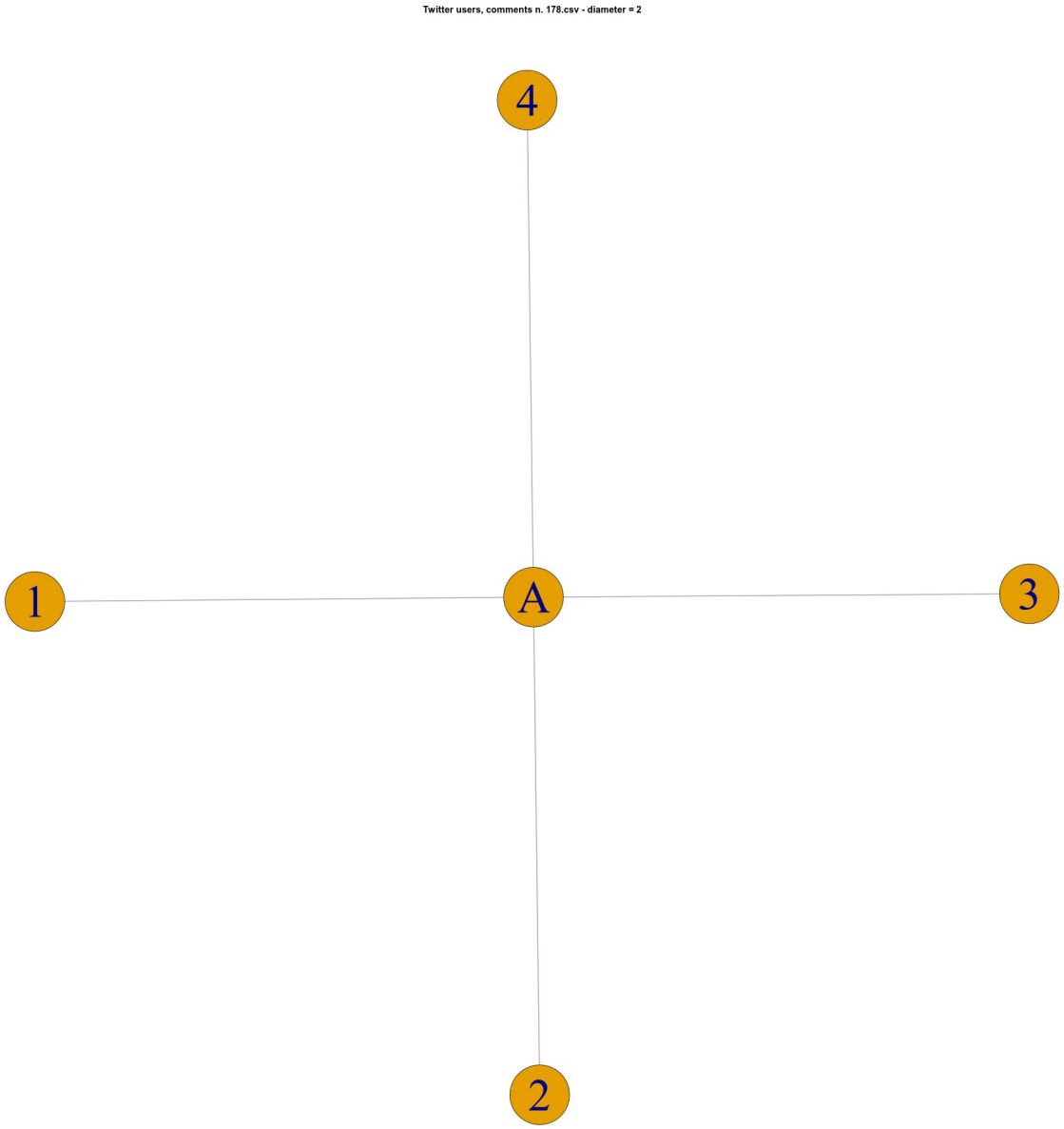


Diagram Twitter network, comment n. 179 – diameter = 3

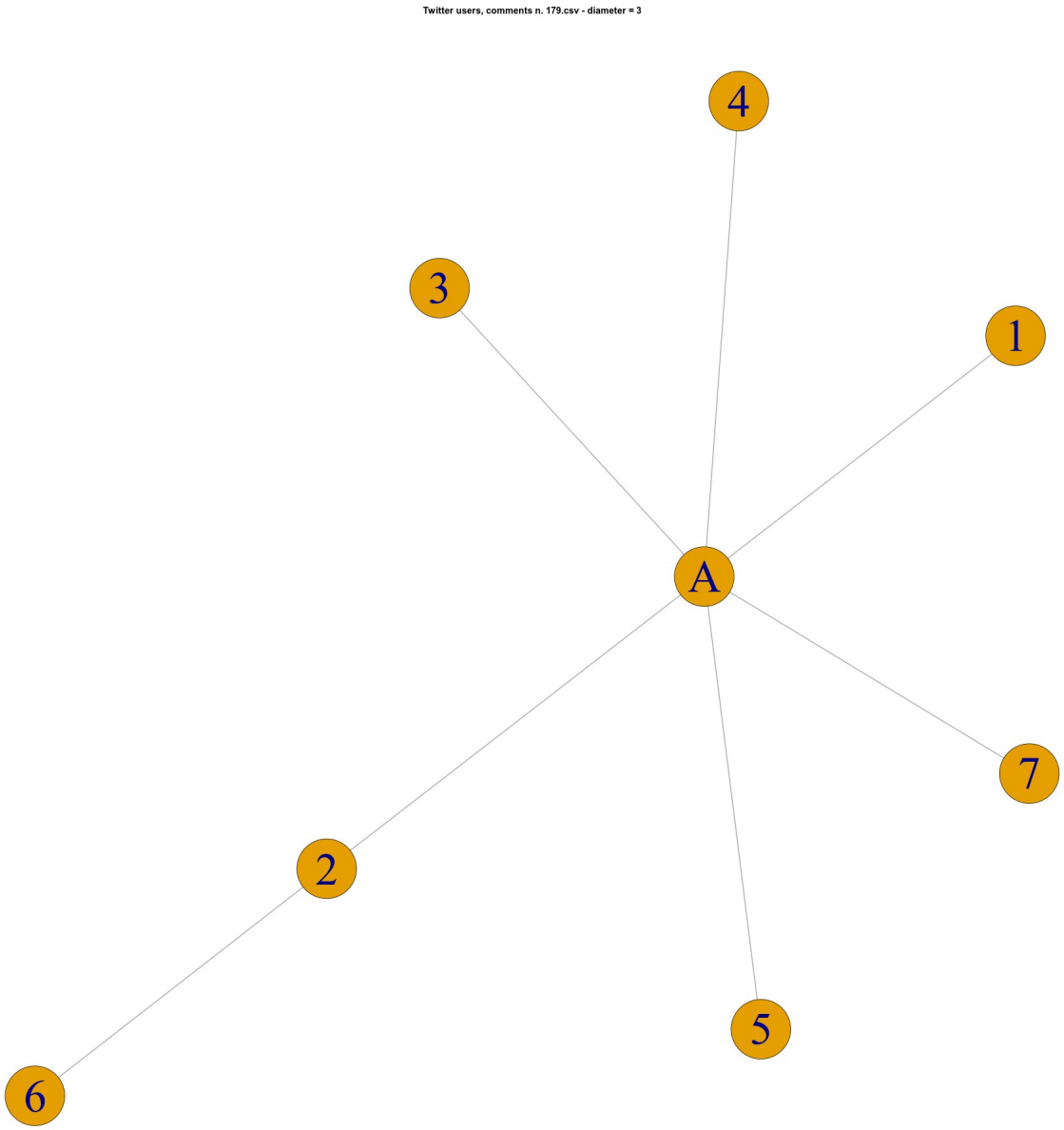


Diagram Twitter network, comment n. 180 – diameter = 1

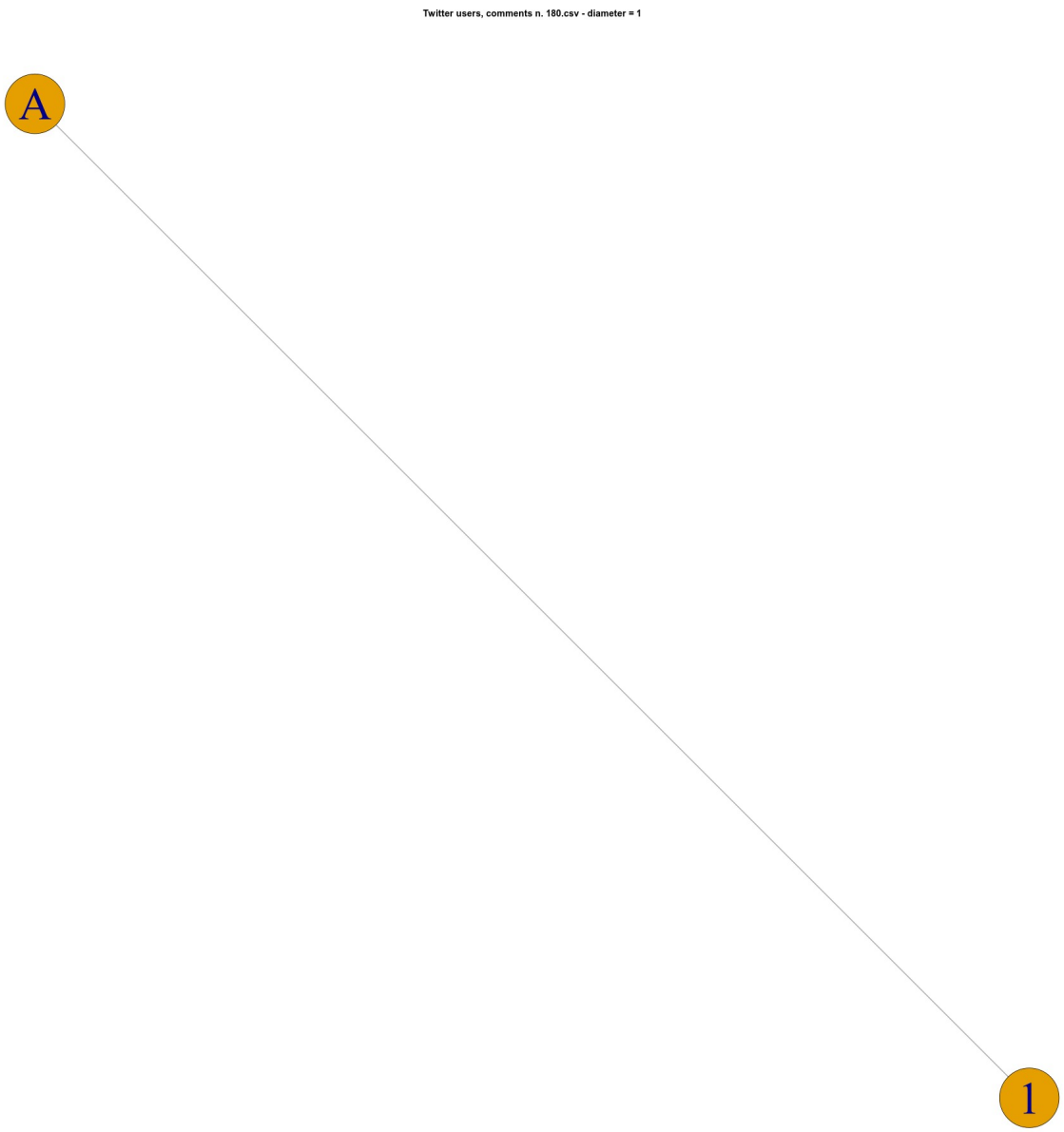


Diagram Twitter network, comment n. 181 – diameter = 1

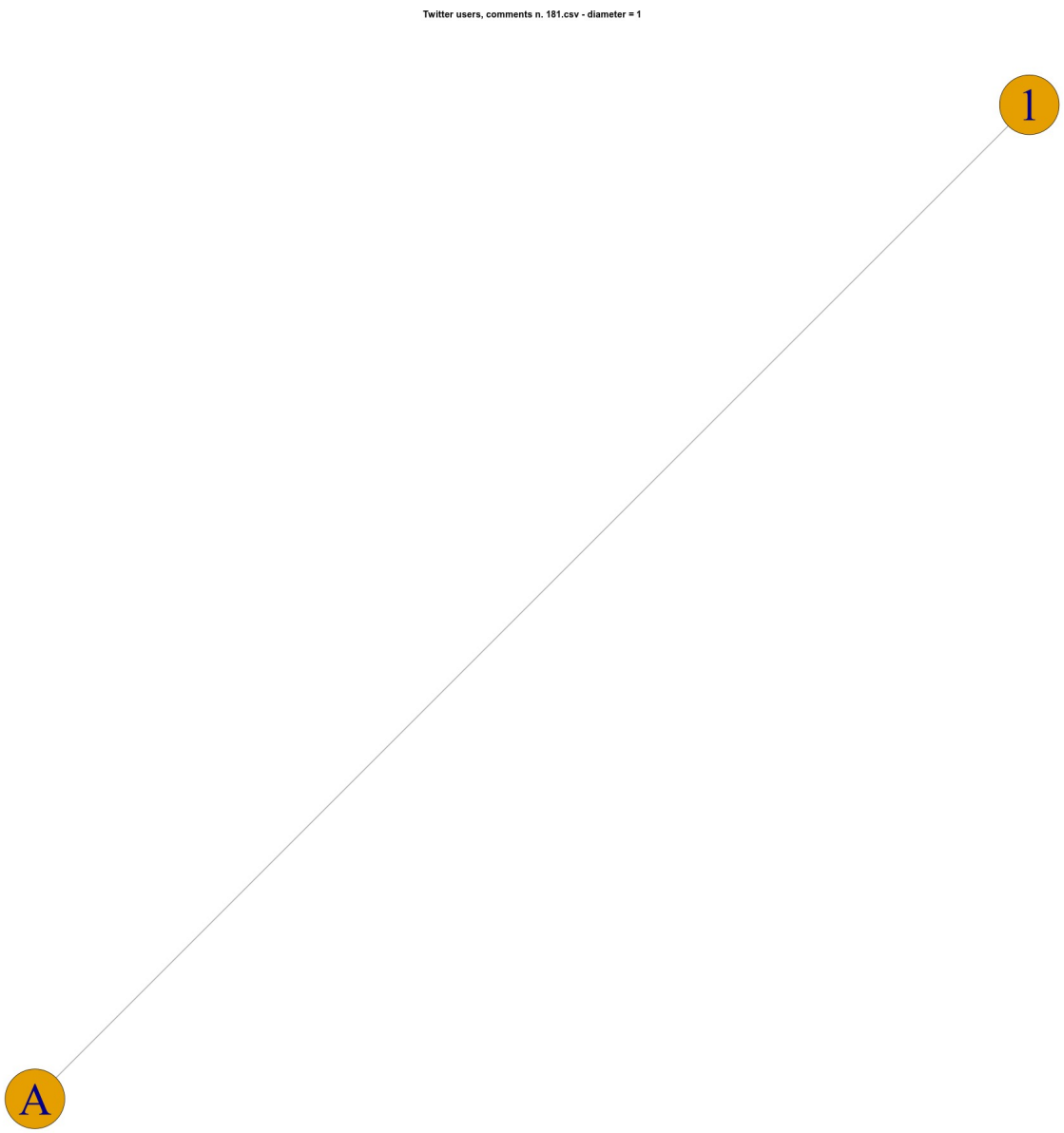


Diagram Twitter network, comment n. 182 – diameter = 2

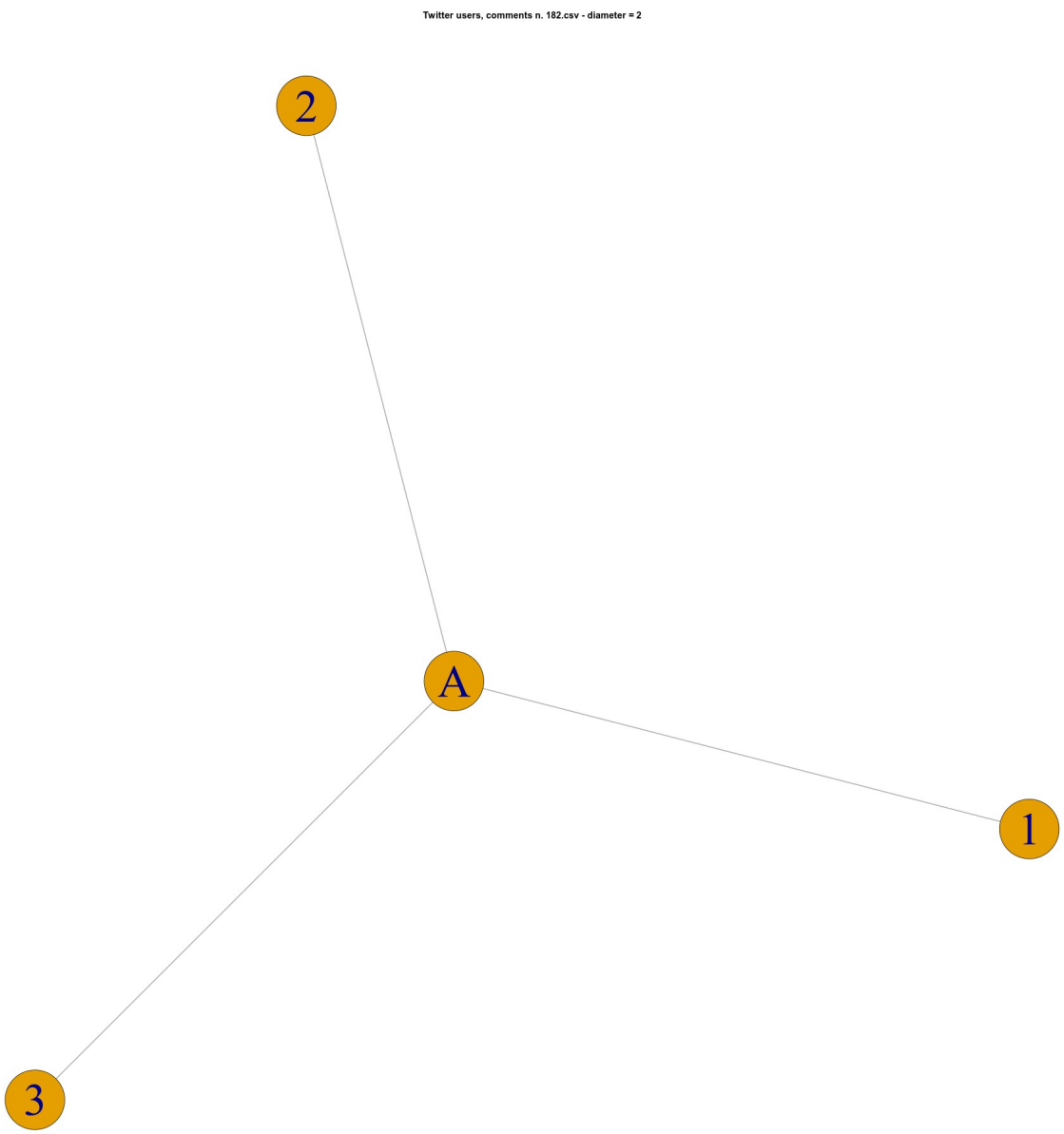


Diagram Twitter network, comment n. 183 – diameter = 2

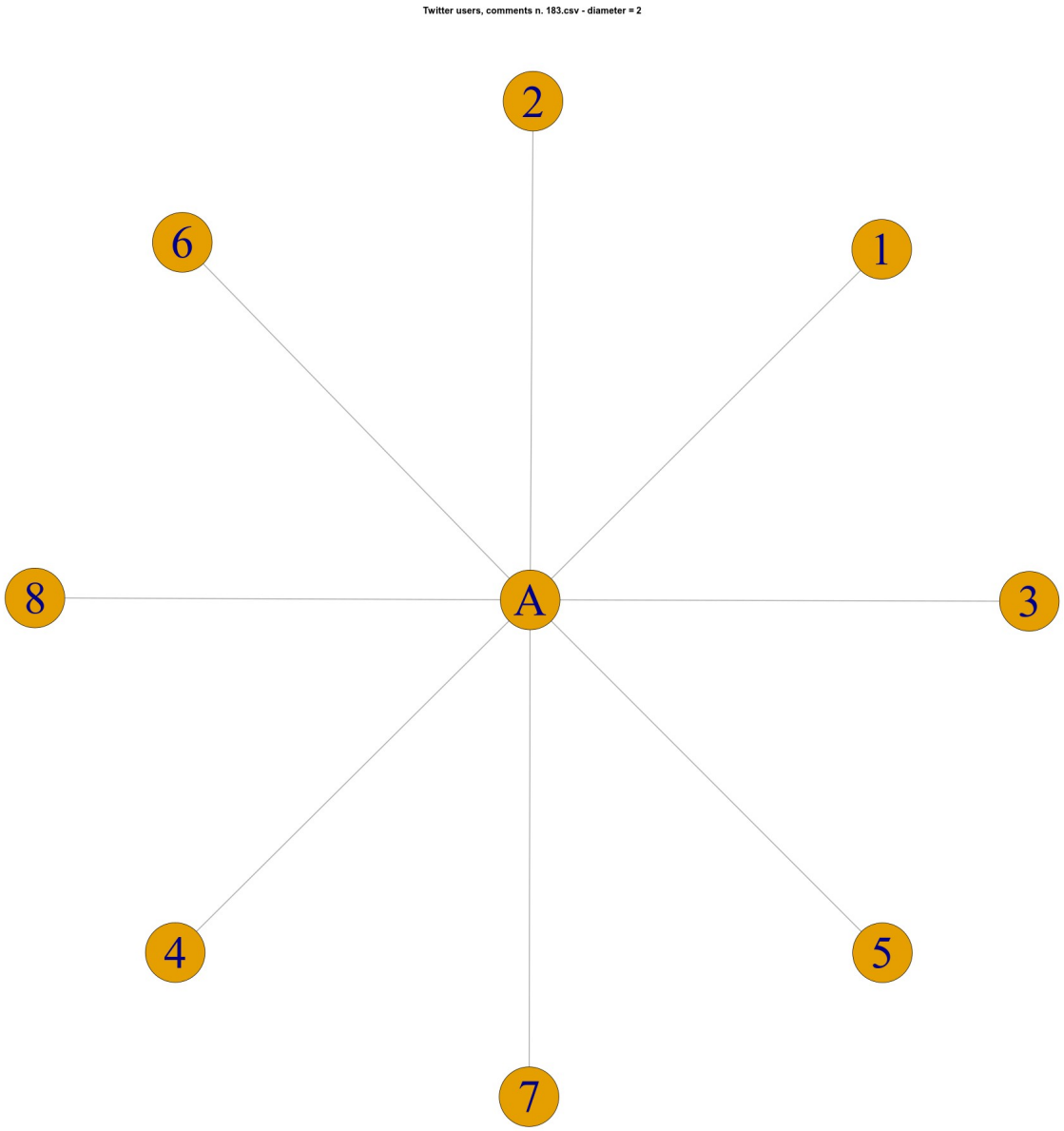


Diagram Twitter network, comment n. 184 – diameter = 3

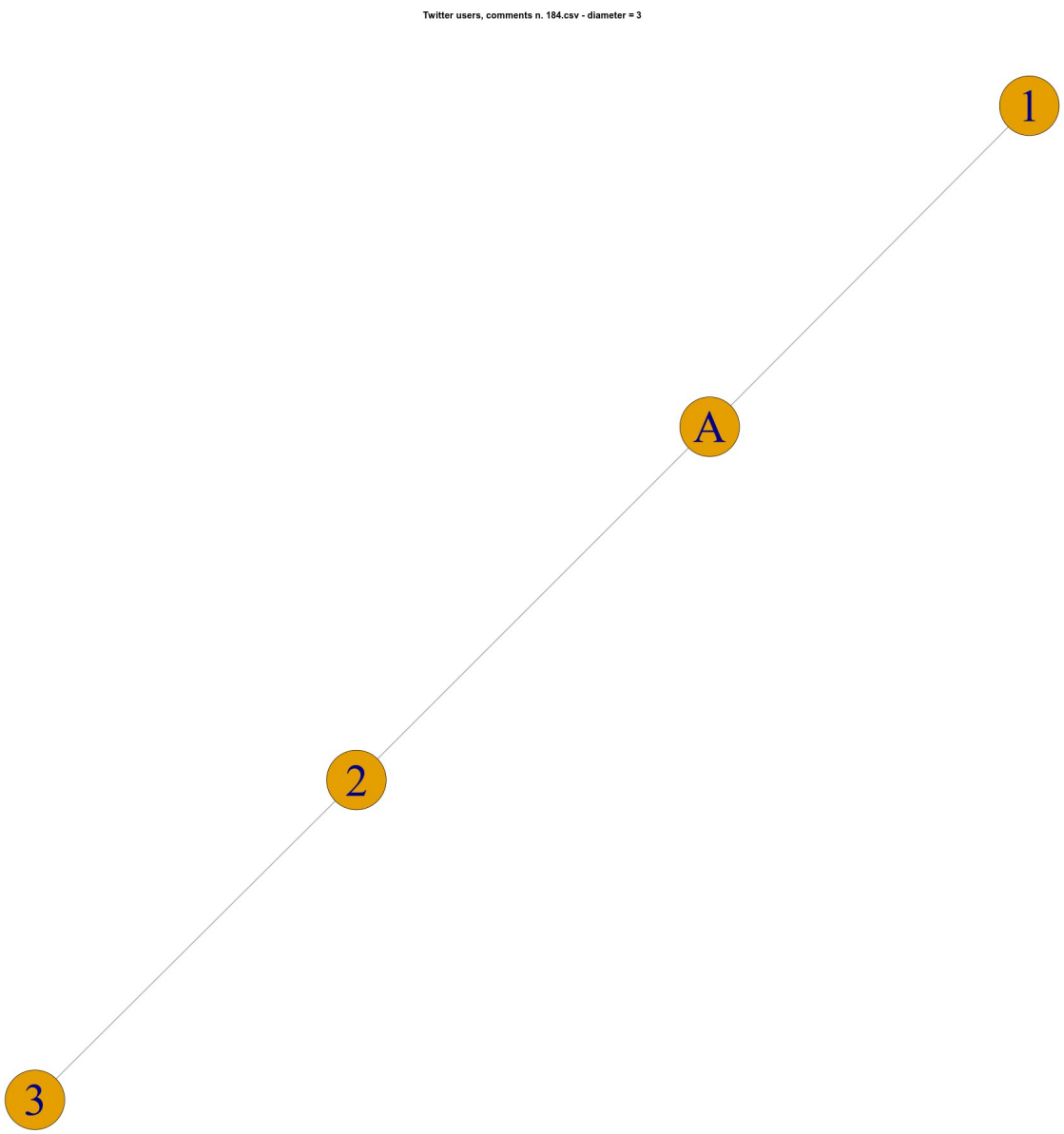


Diagram Twitter network, comment n. 186 – diameter = 1

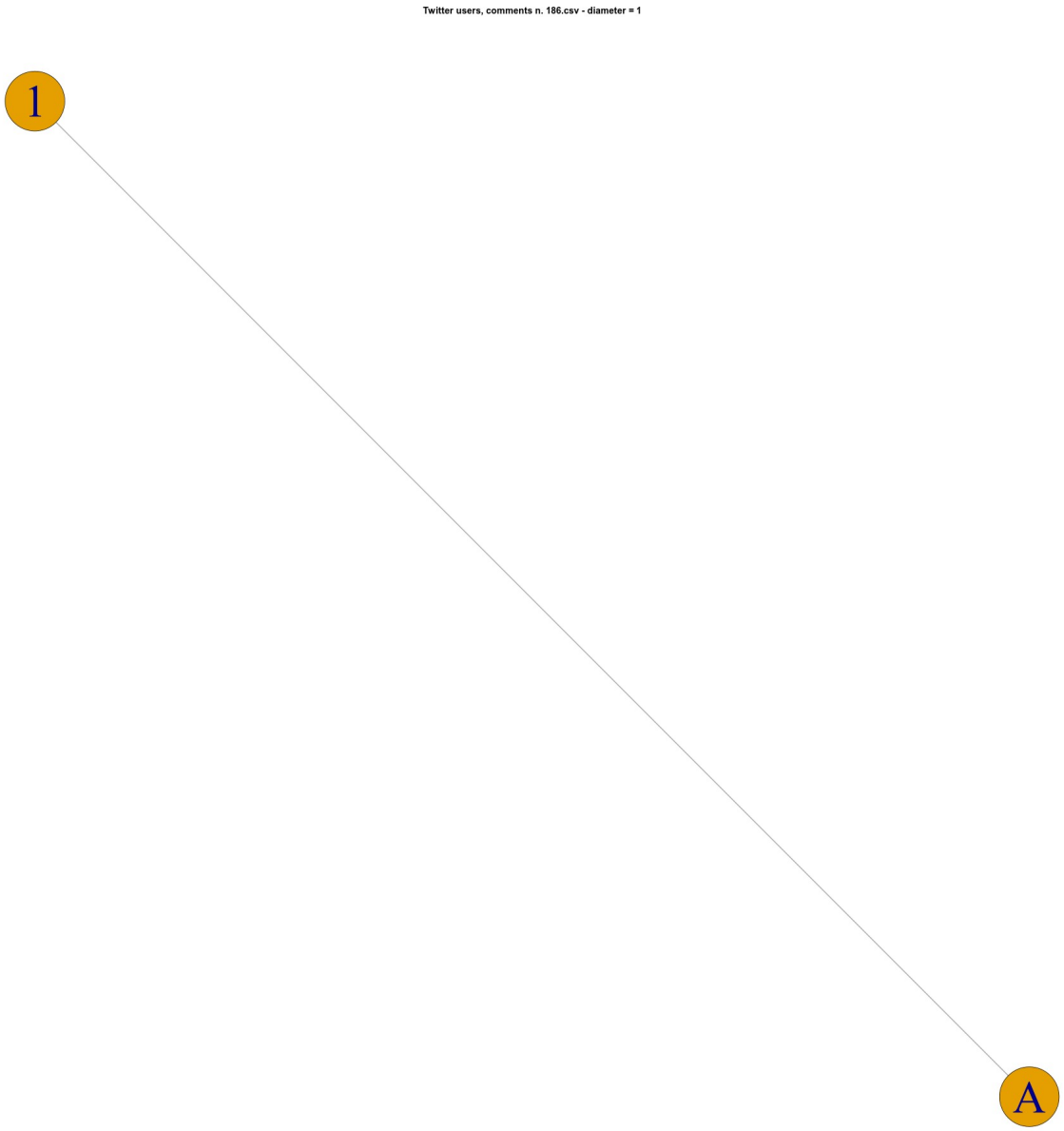


Diagram Twitter network, comment n. 187 – diameter = 1

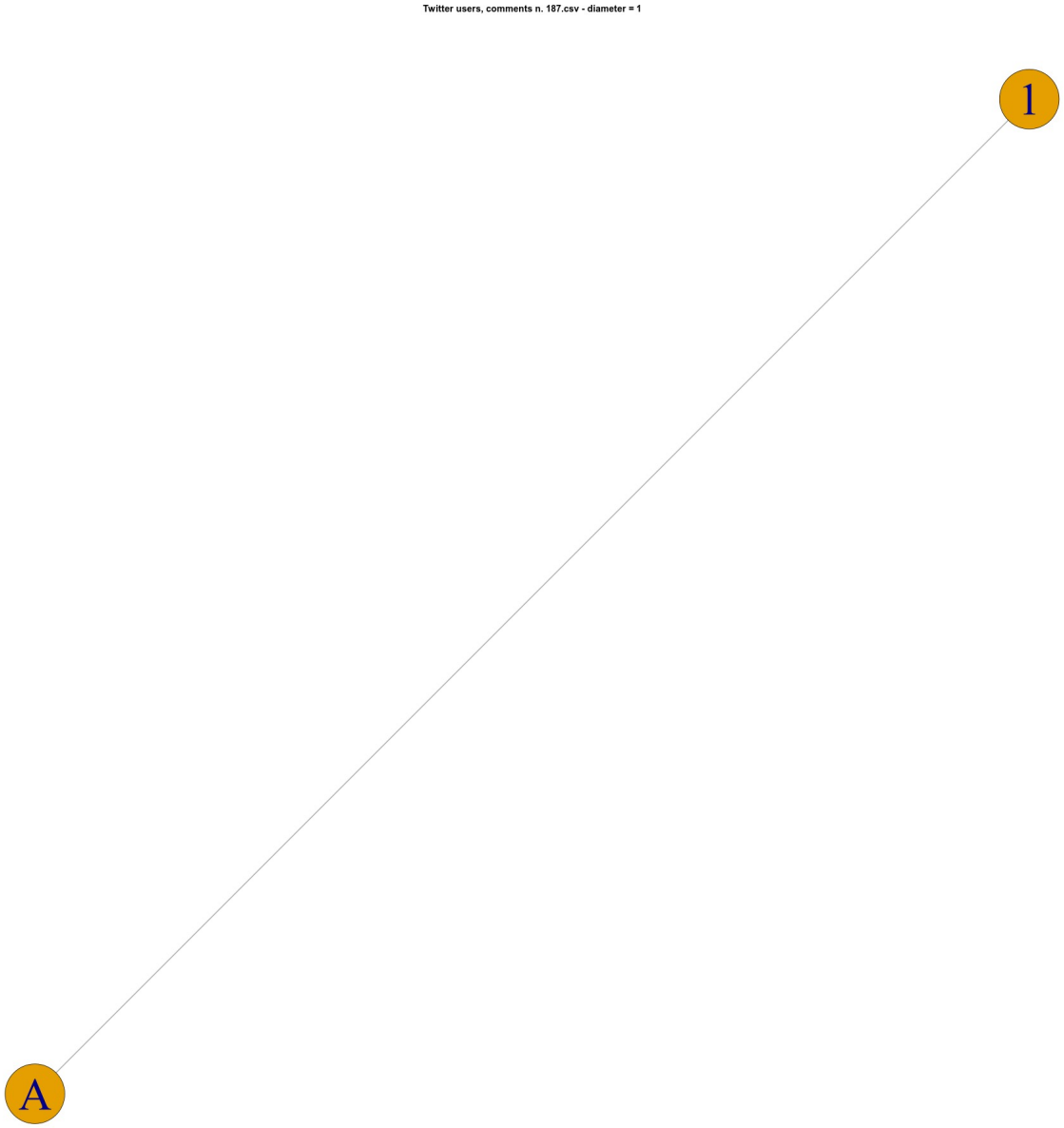


Diagram Twitter network, comment n. 188 – diameter = 4

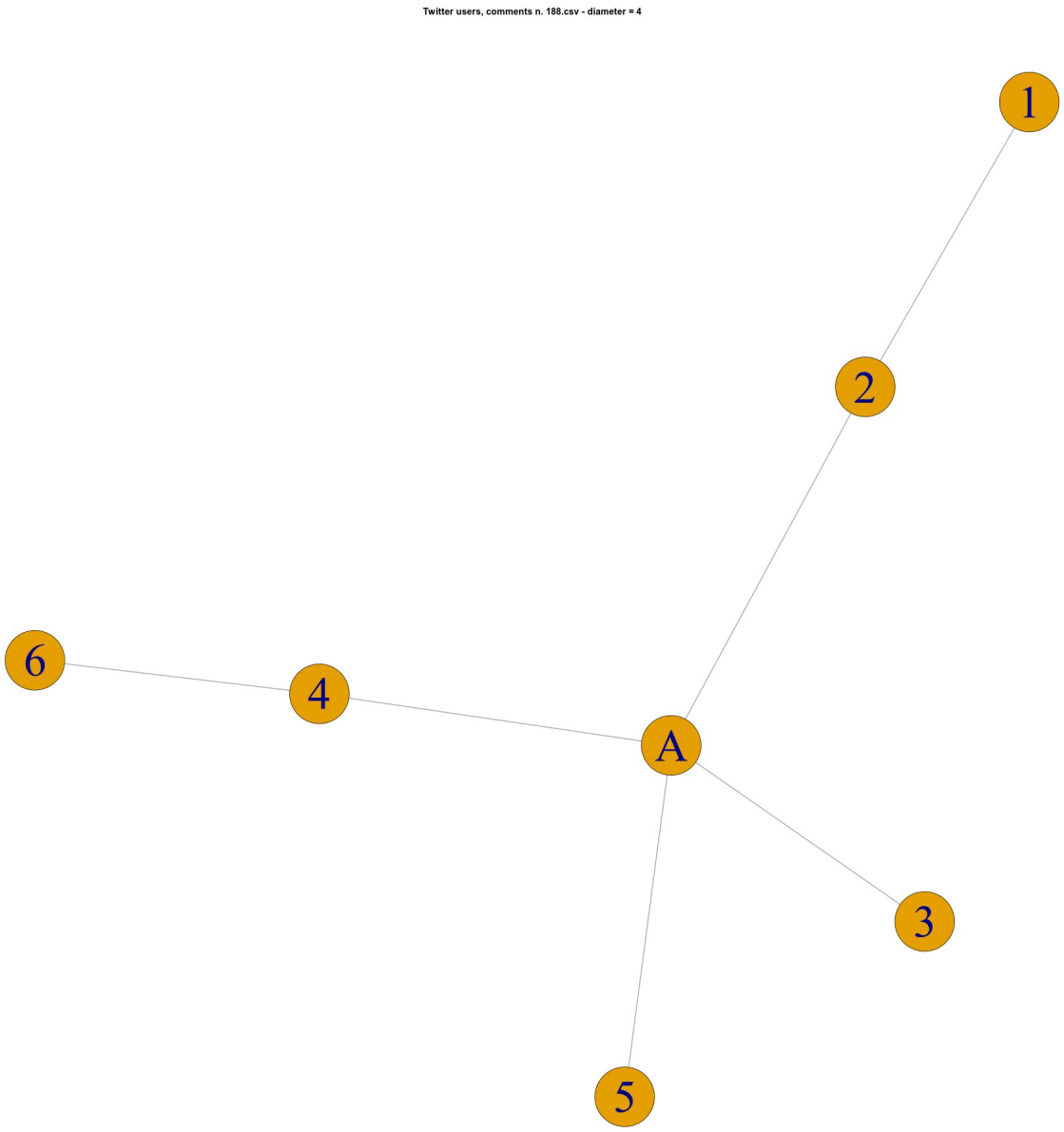


Diagram Twitter network, comment n. 189 – diameter = 2

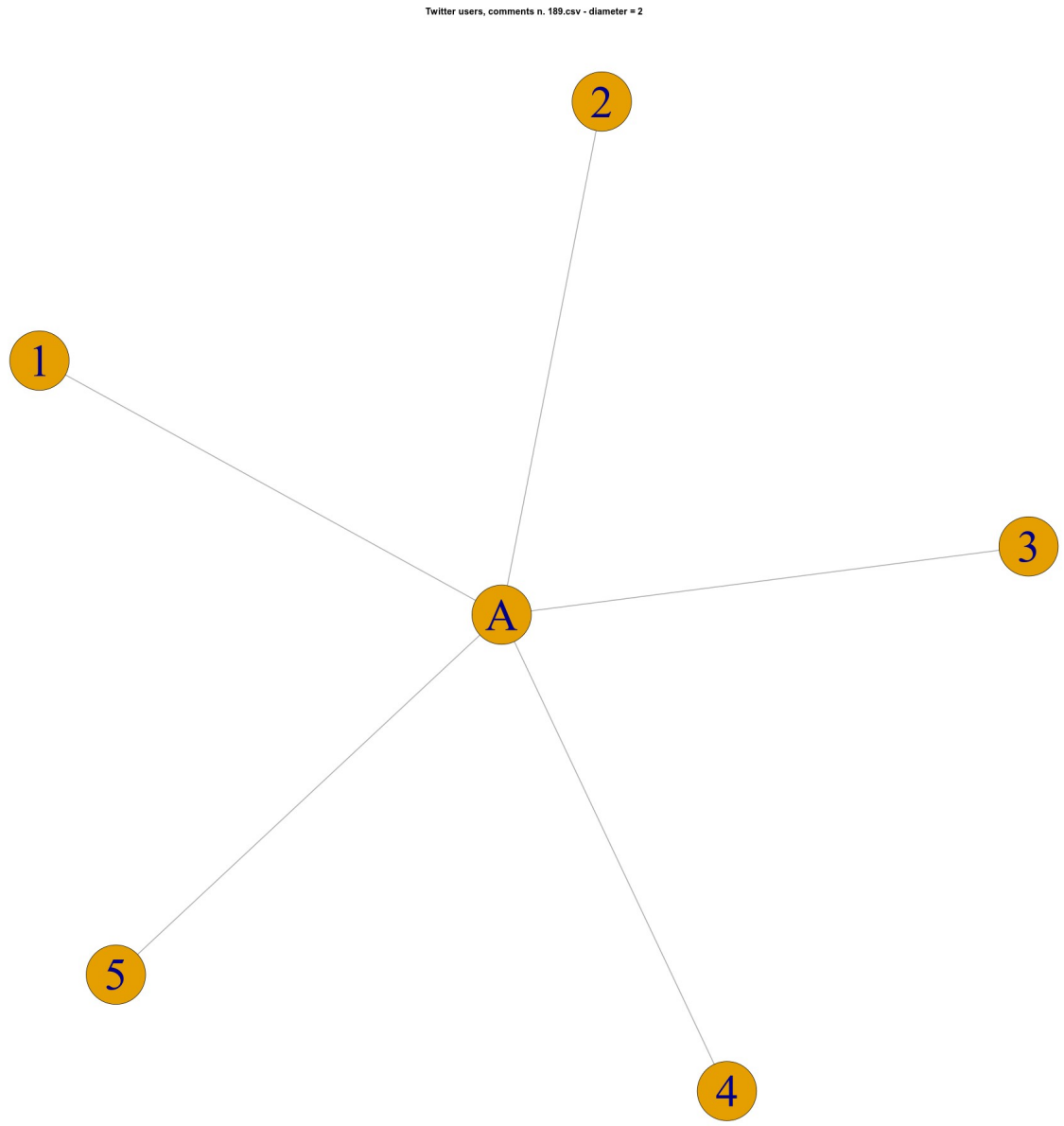


Diagram Twitter network, comment n. 190 – diameter = 1

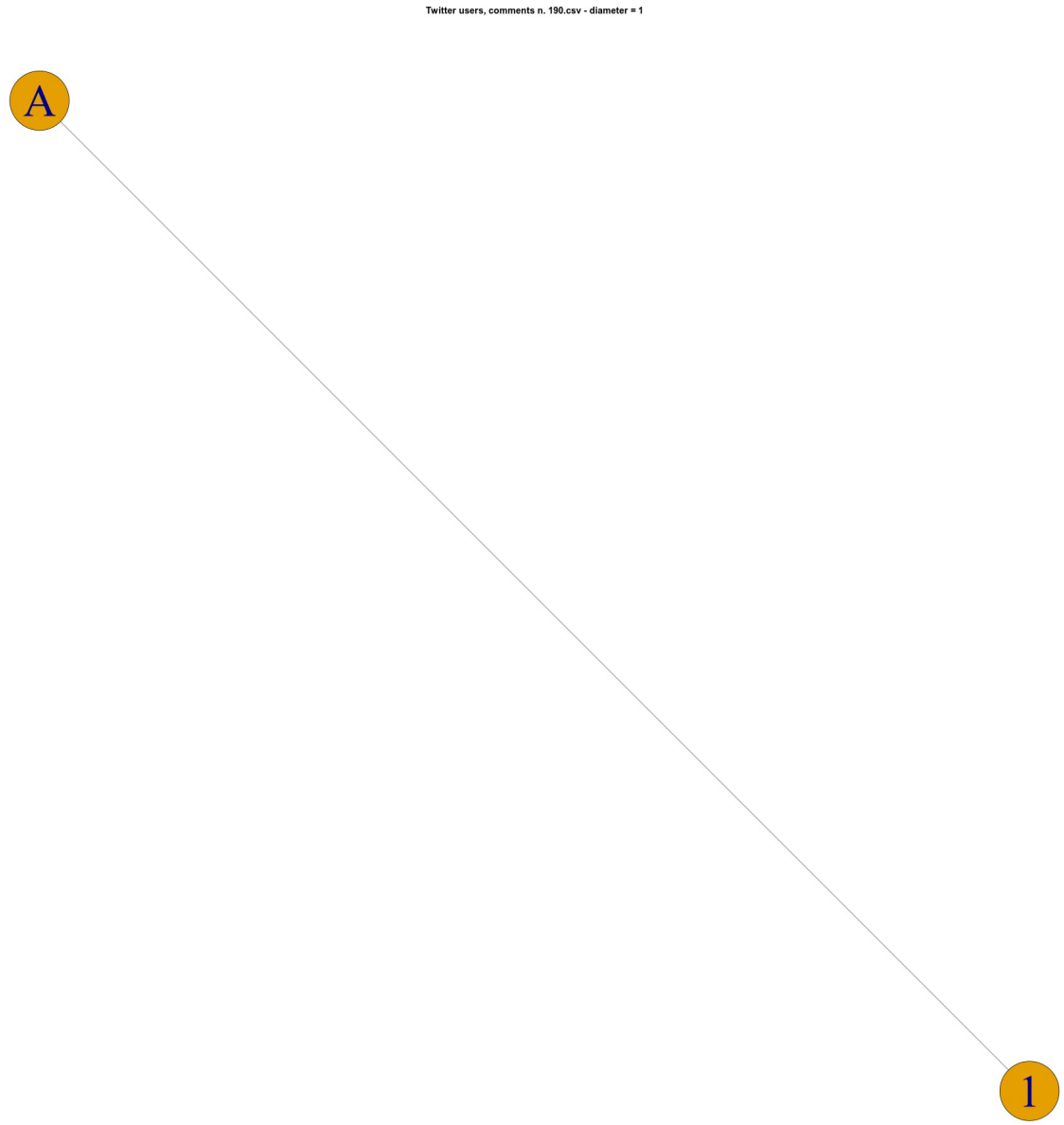


Diagram Twitter network, comment n. 192 – diameter = 2

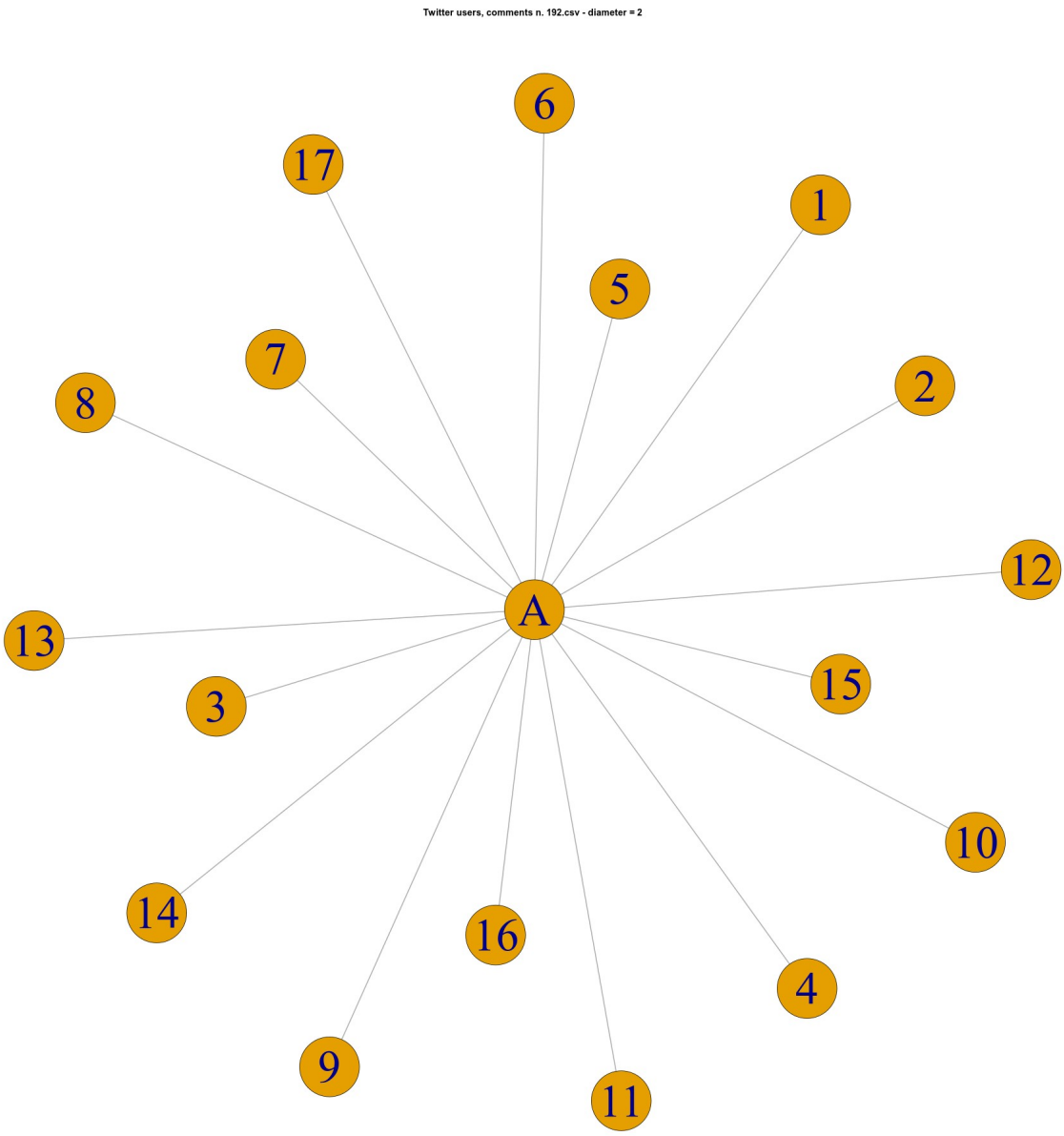


Diagram Twitter network, comment n. 193 – diameter = 2

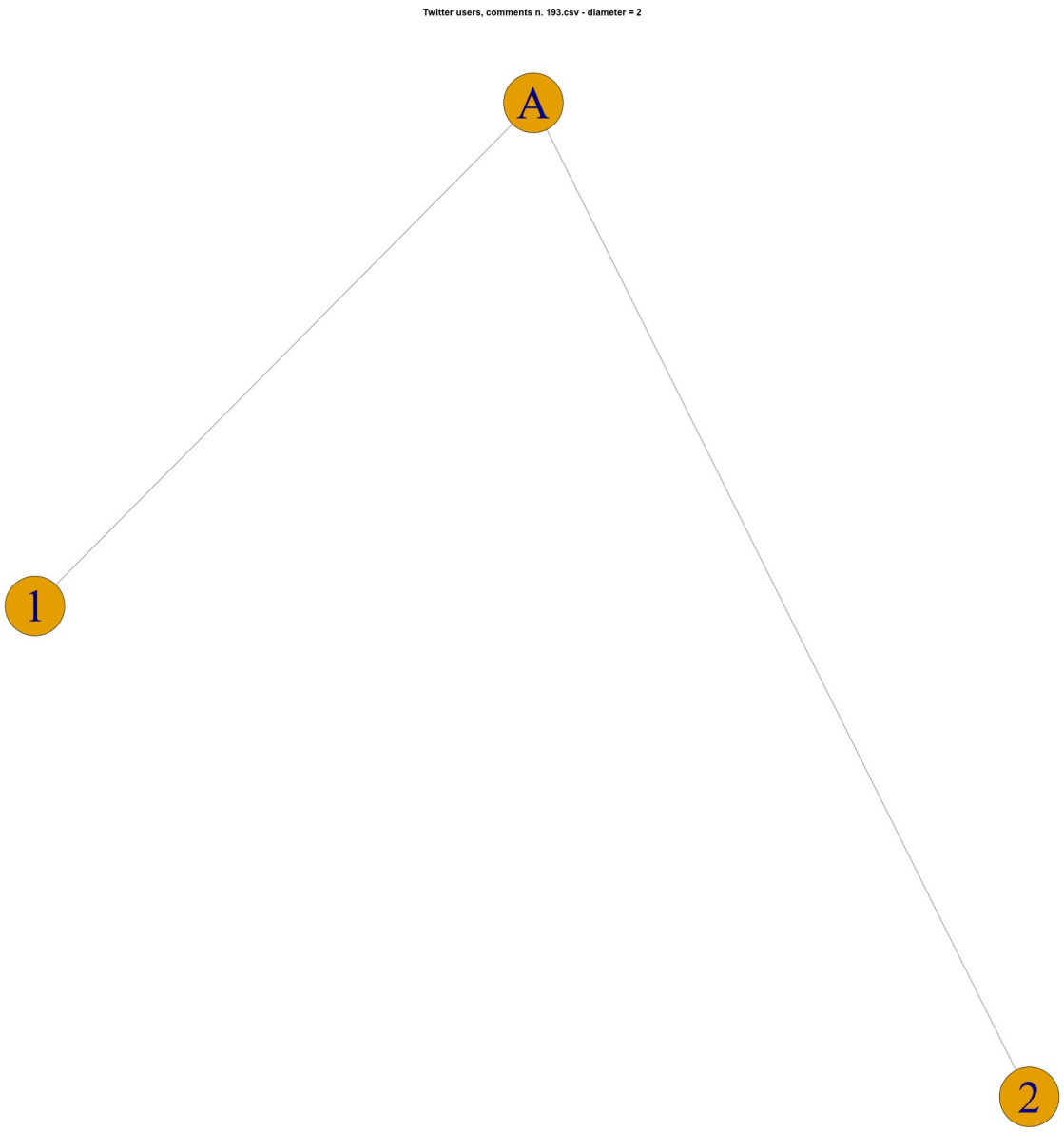


Diagram Twitter network, comment n. 194 – diameter = 2

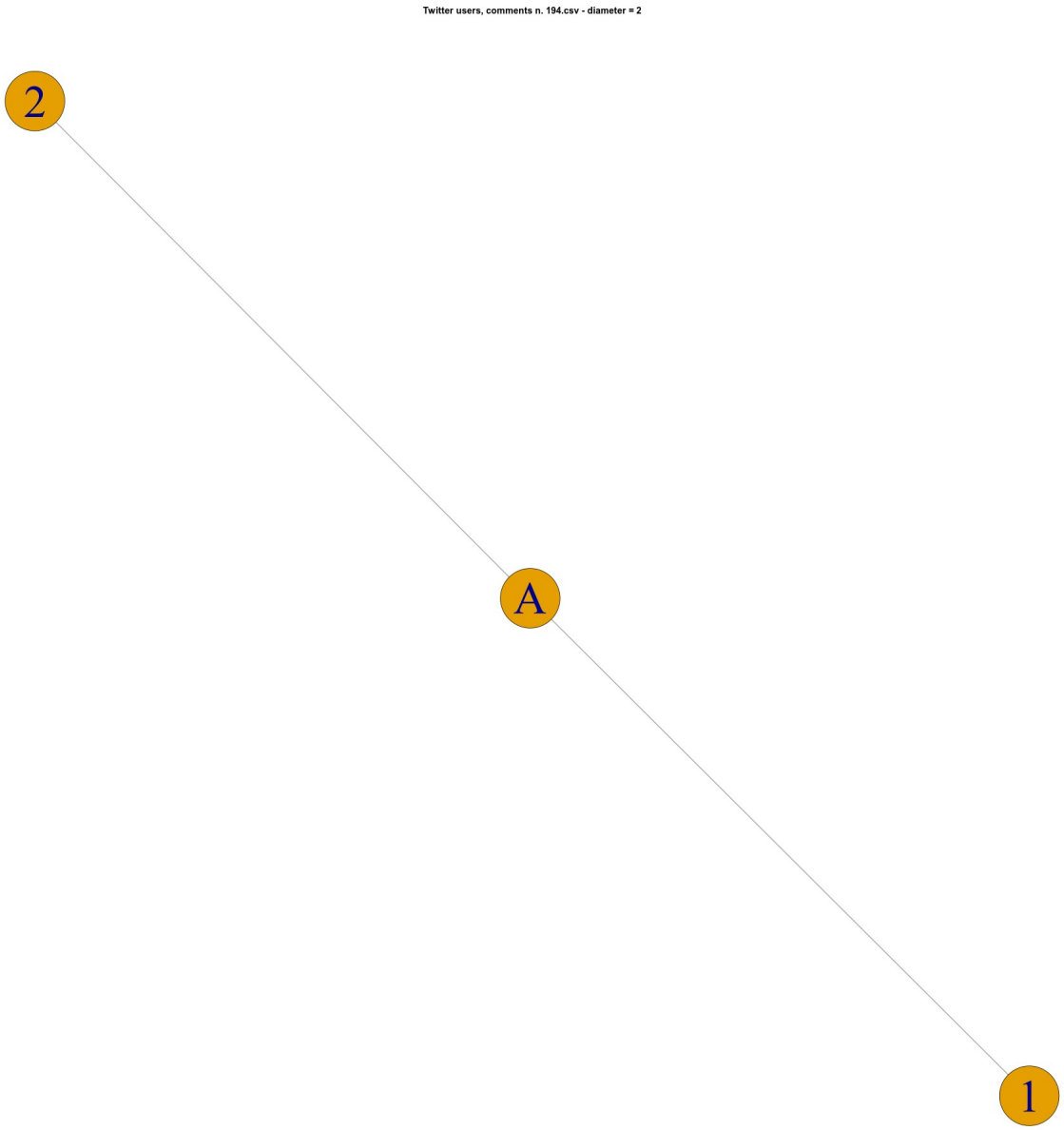


Diagram Twitter network, comment n. 198 – diameter = 3

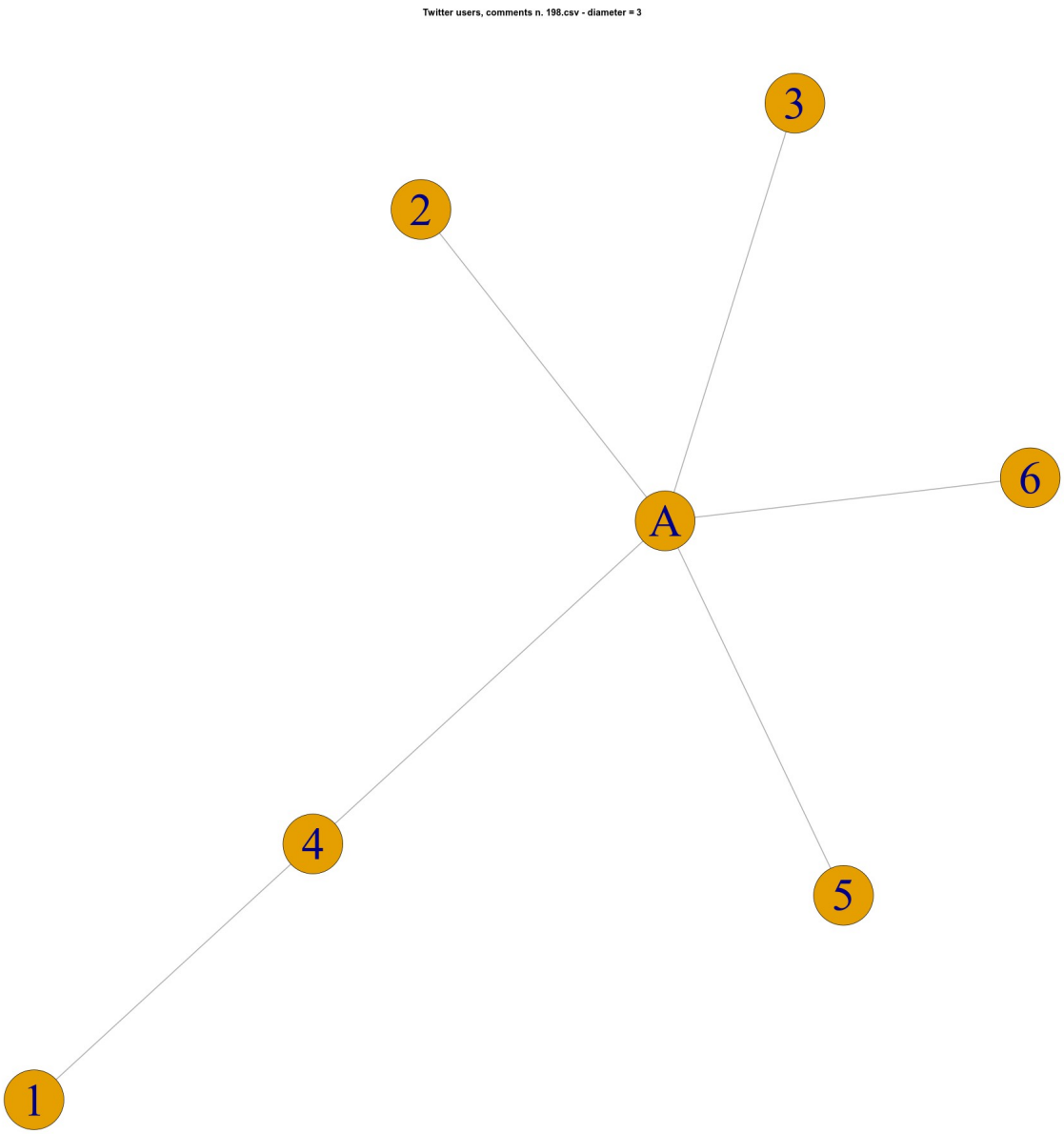


Diagram Twitter network, comment n. 199 – diameter = 2

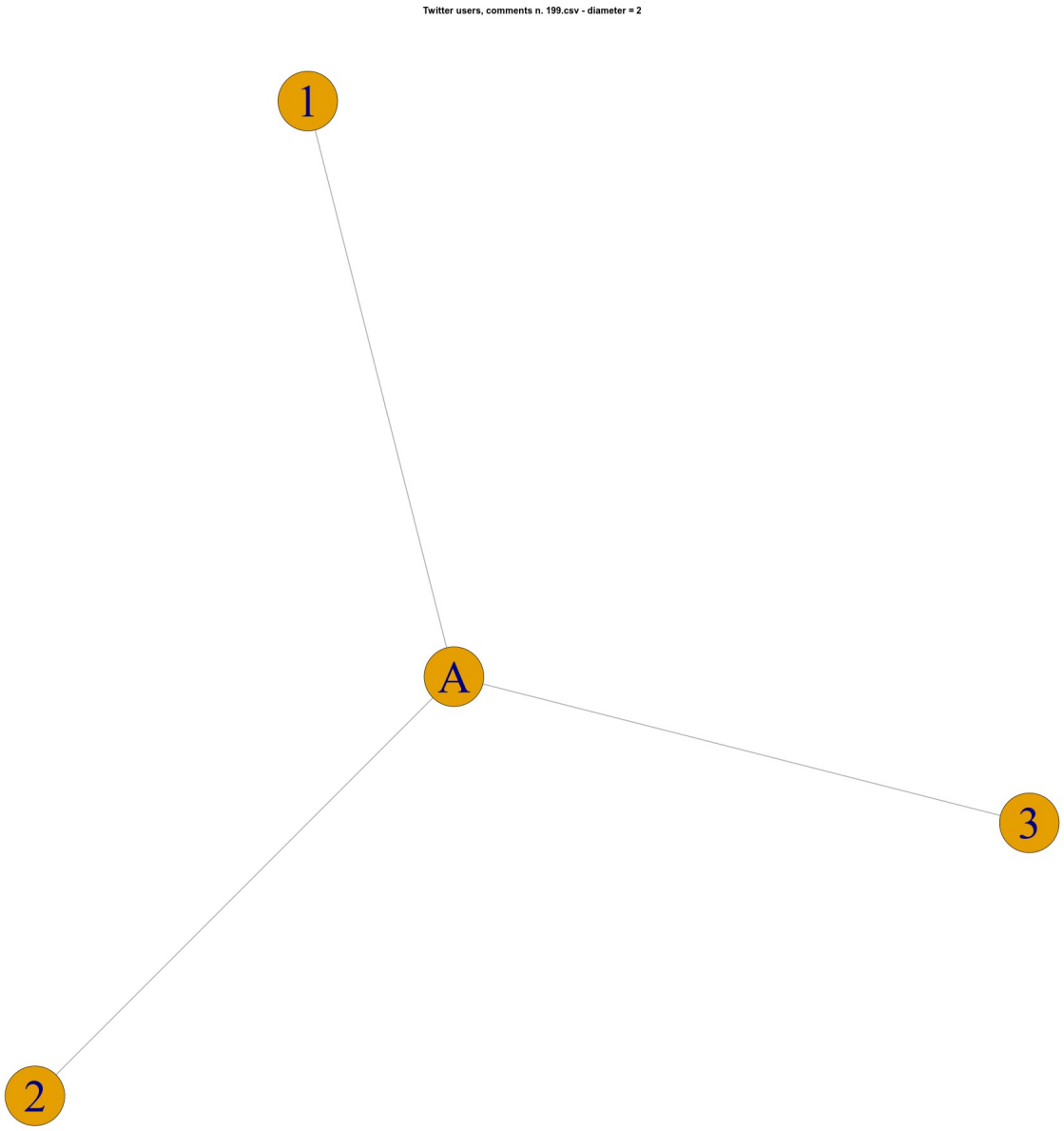


Diagram Twitter network, comment n. 201 – diameter = 3

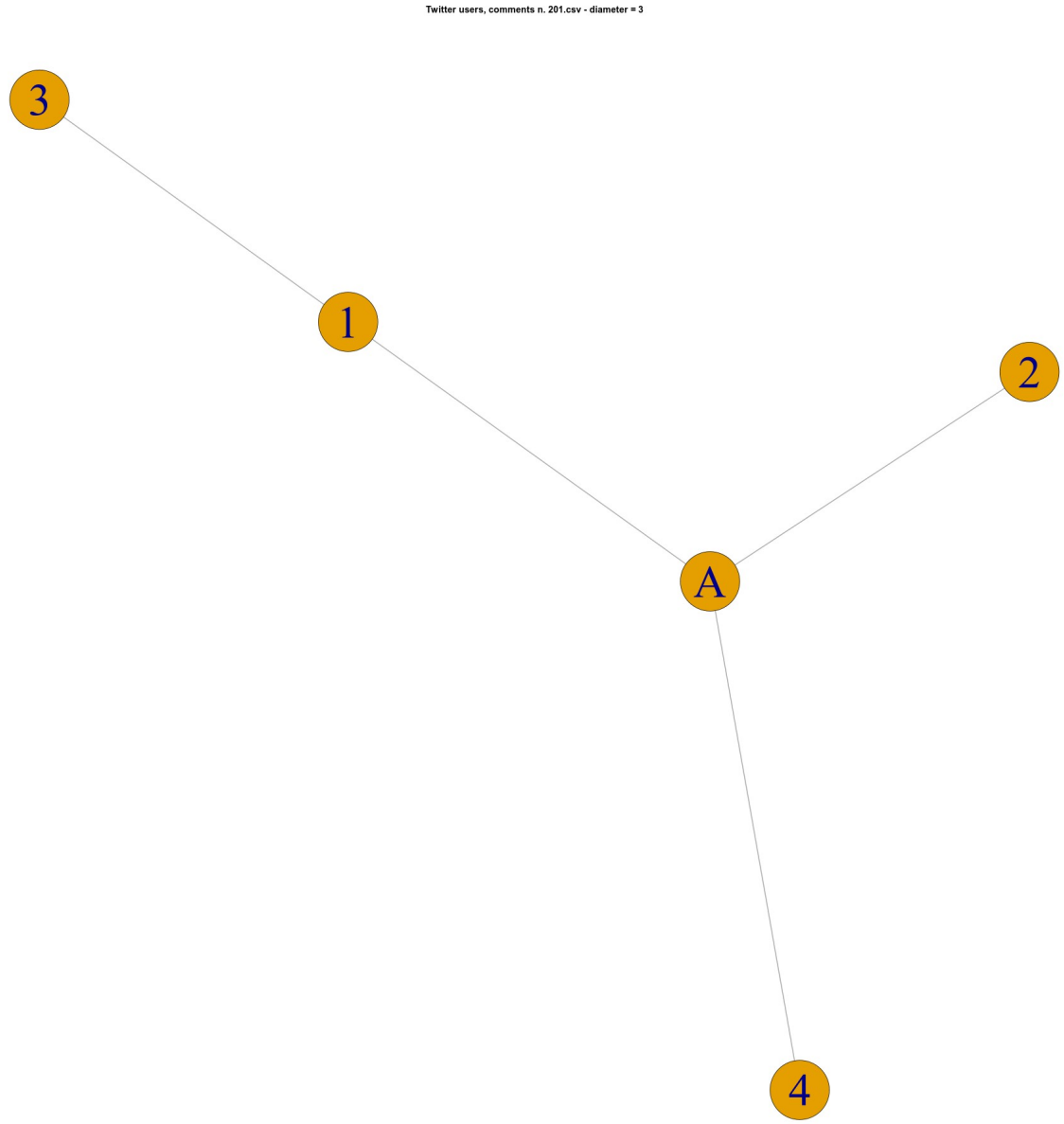


Diagram Twitter network, comment n. 203 – diameter = 1

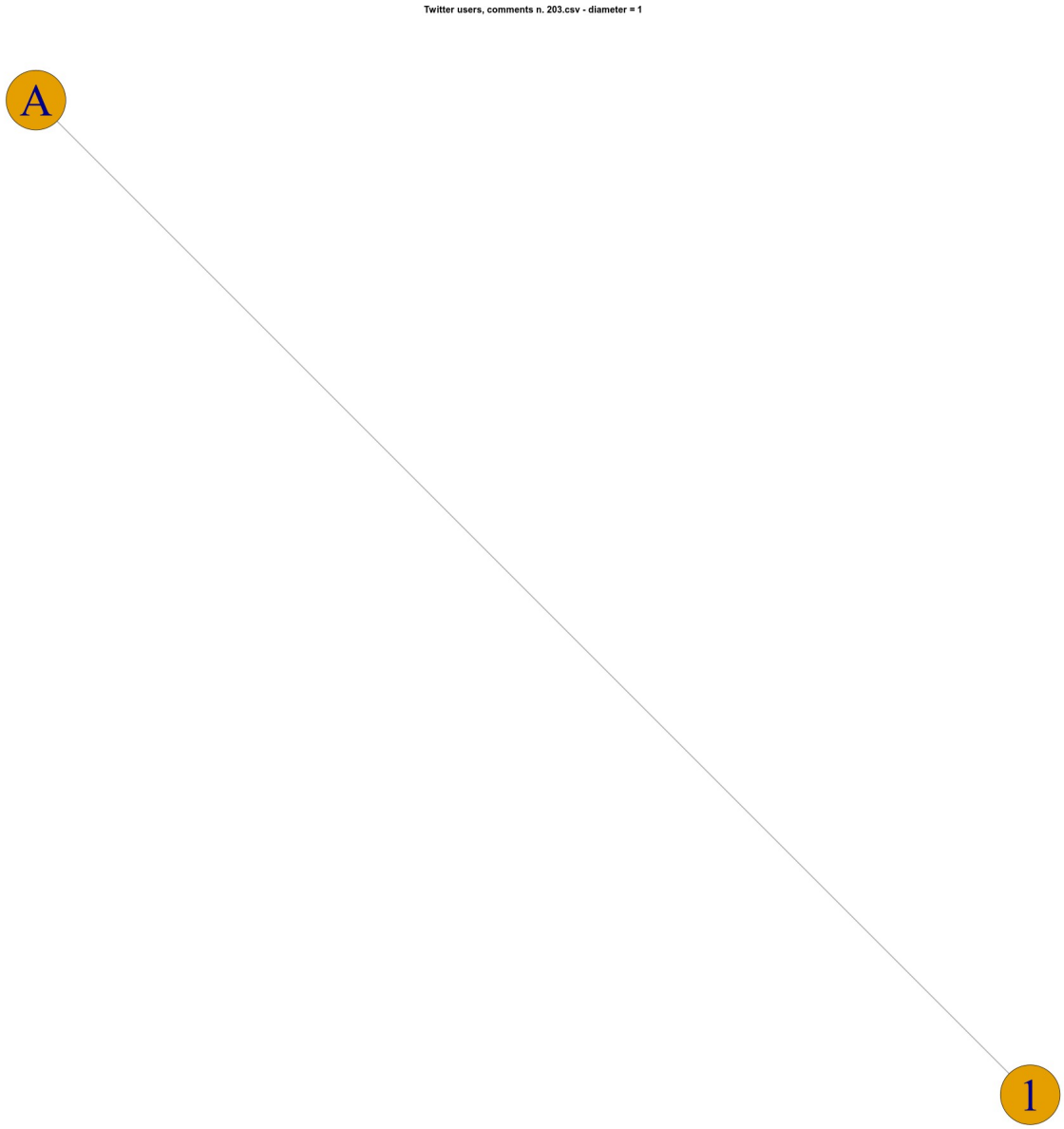


Diagram Twitter network, comment n. 204 – diameter = 1

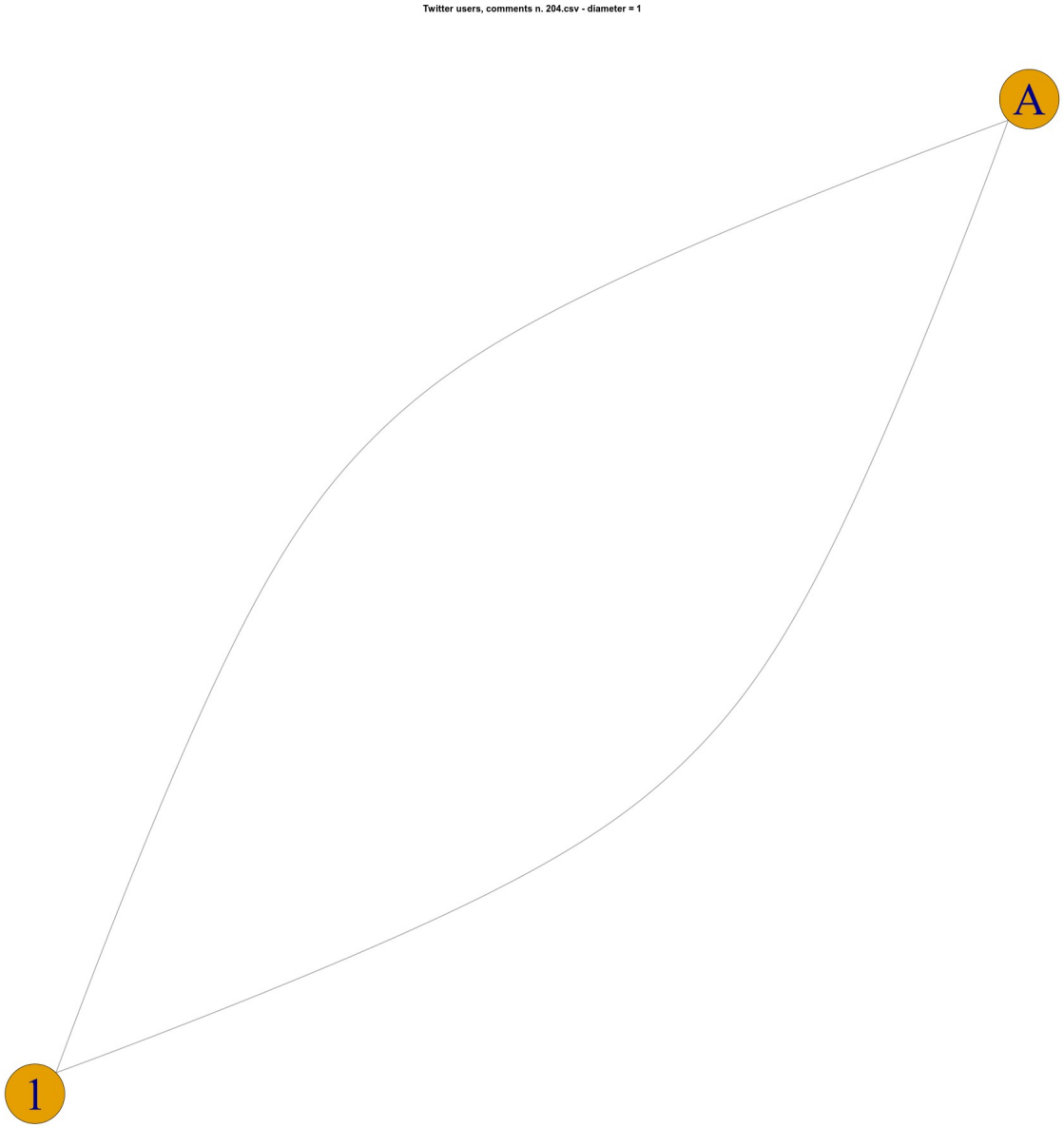


Diagram Twitter network, comment n. 206 – diameter = 2

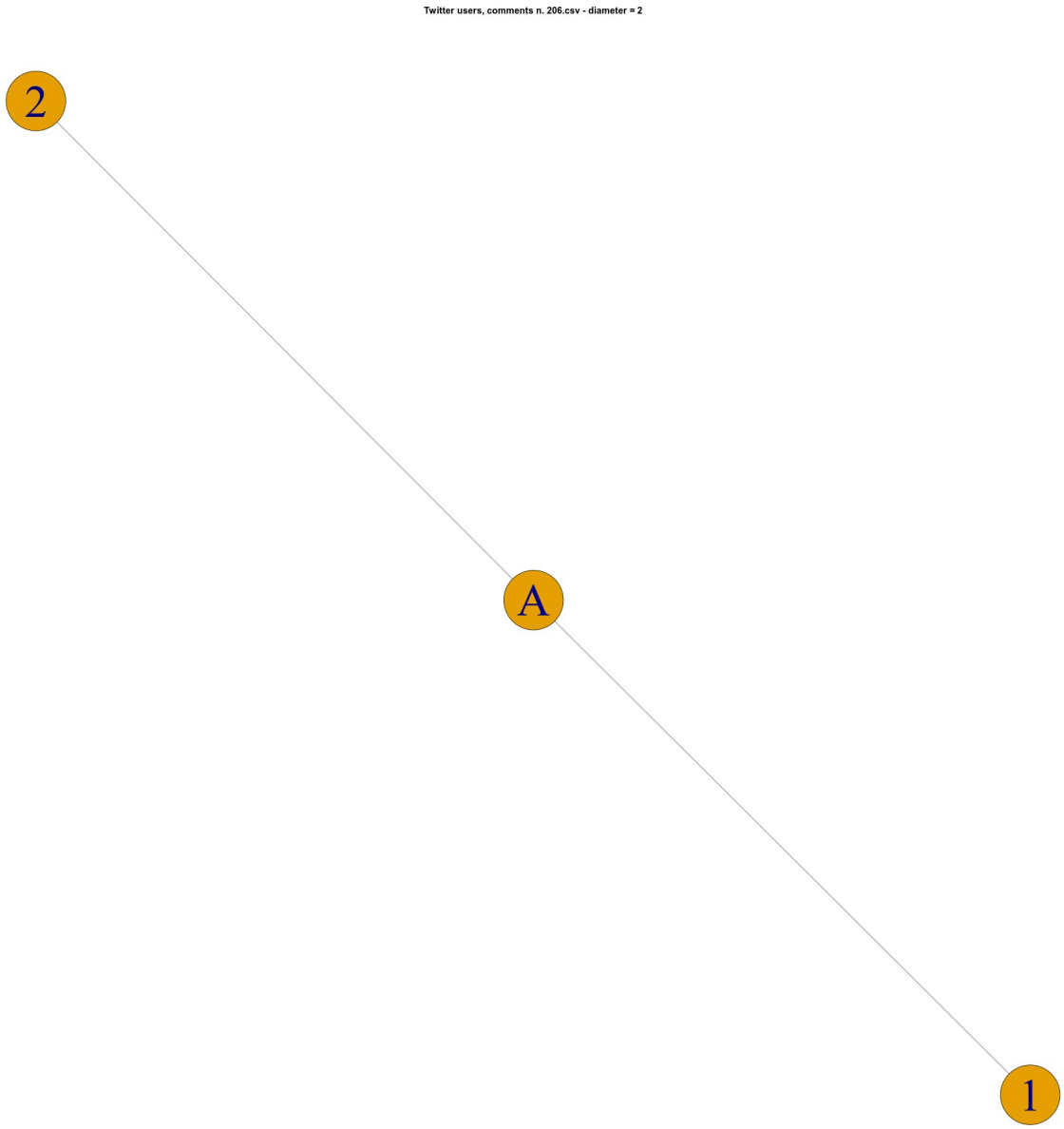


Diagram Twitter network, comment n. 207 – diameter = 2

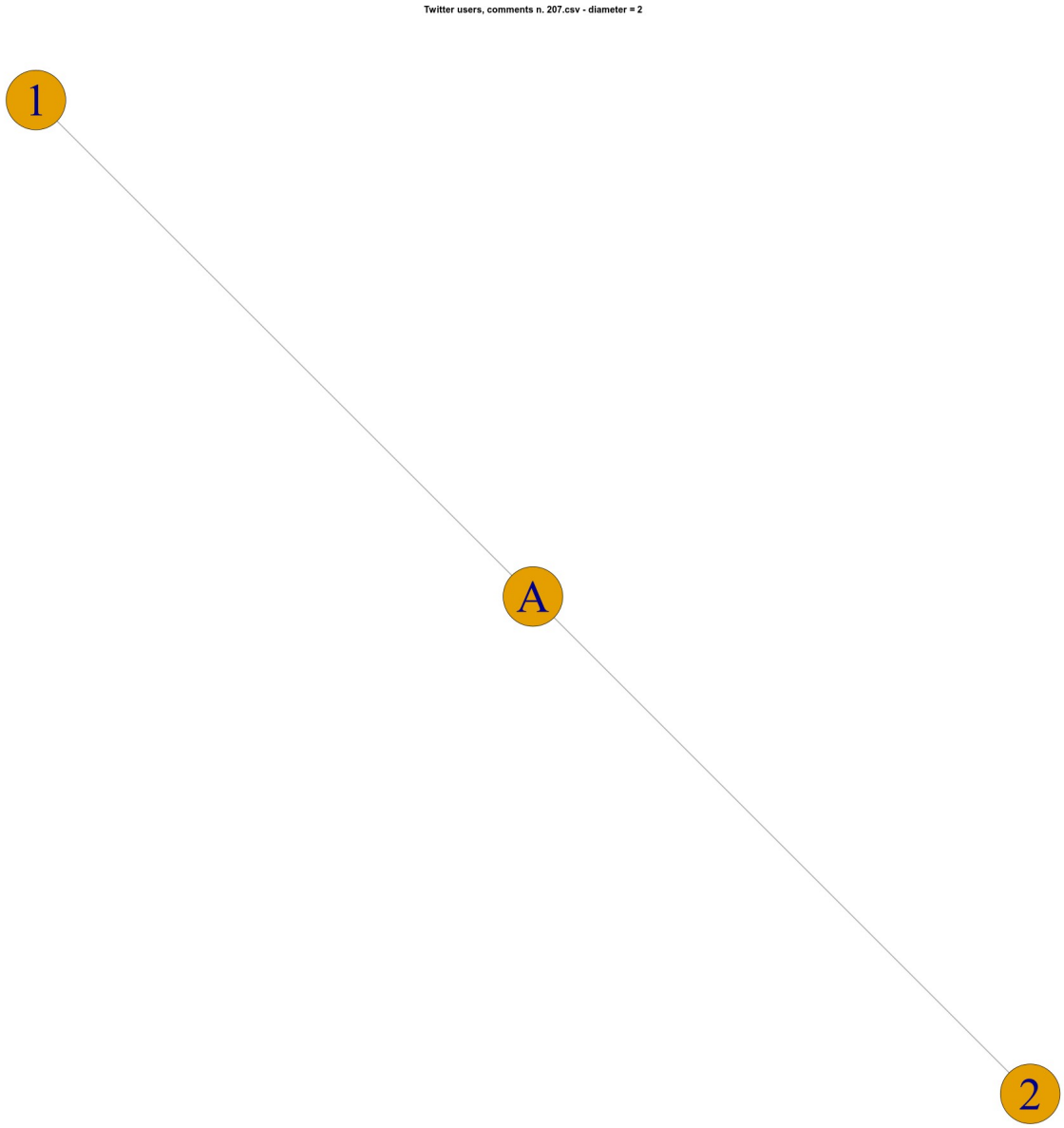


Diagram Twitter network, comment n. 213 – diameter = 3

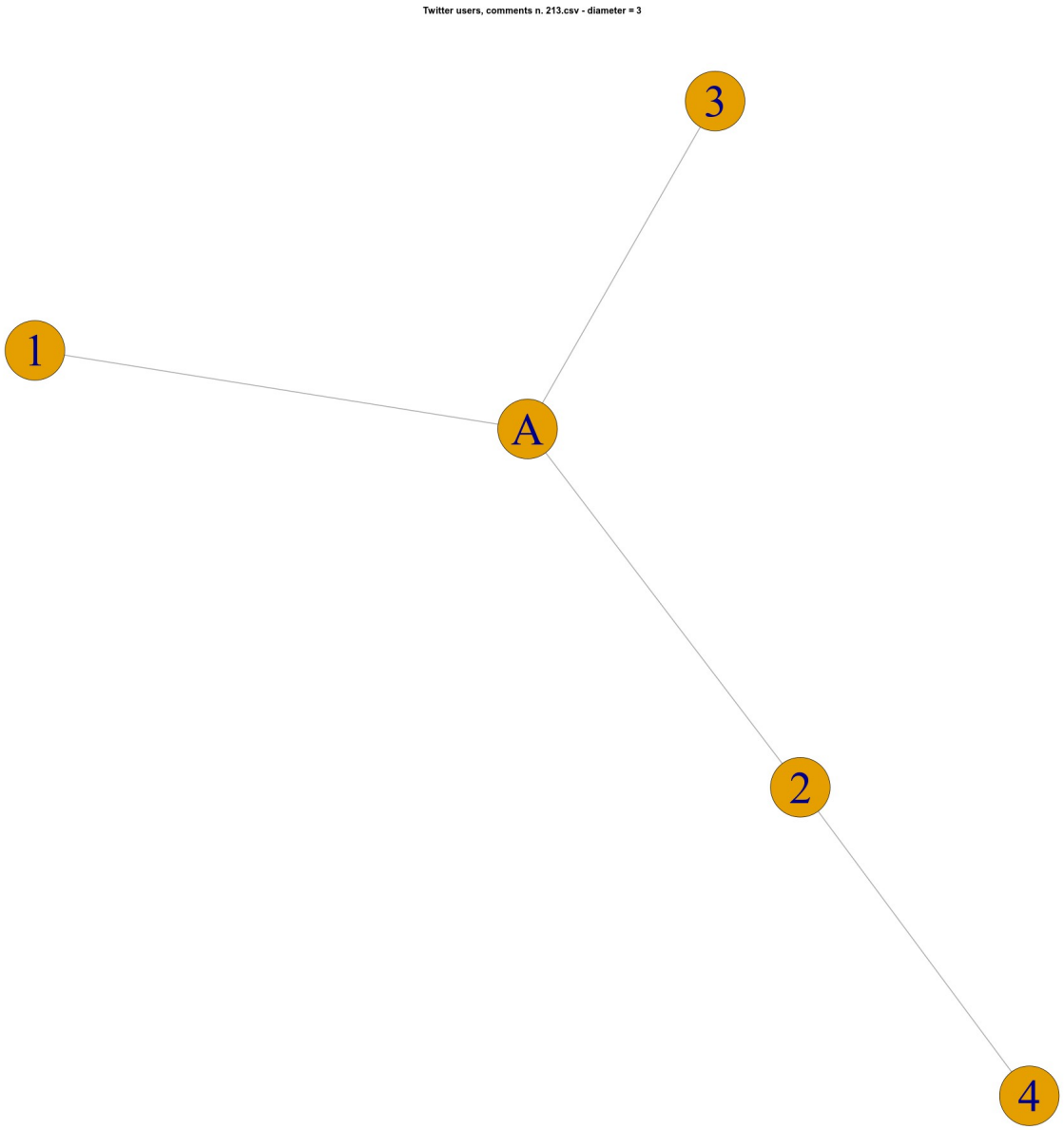


Diagram Twitter network, comment n. 216 – diameter = 2

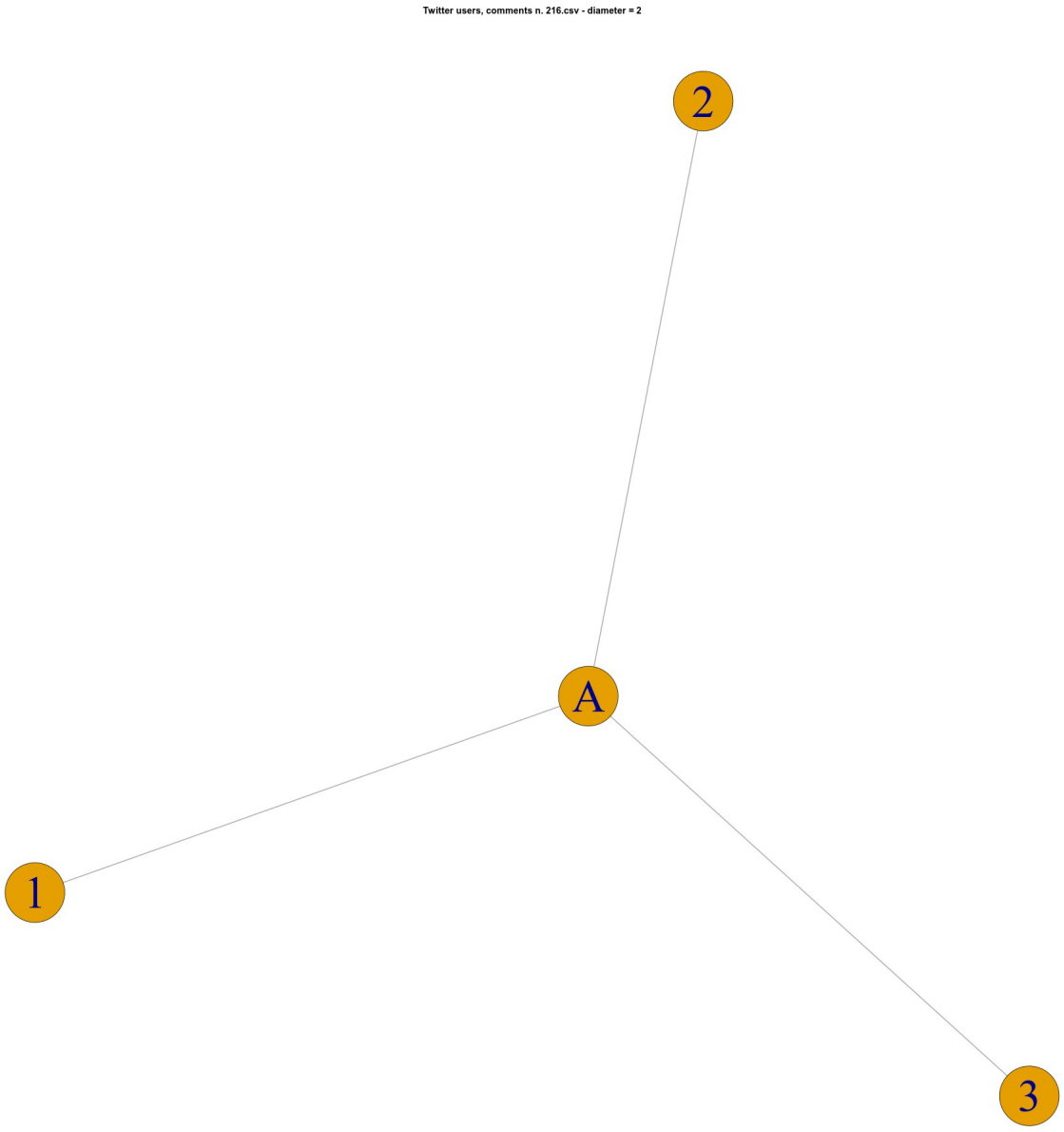


Diagram Twitter network, comment n. 219 – diameter = 2

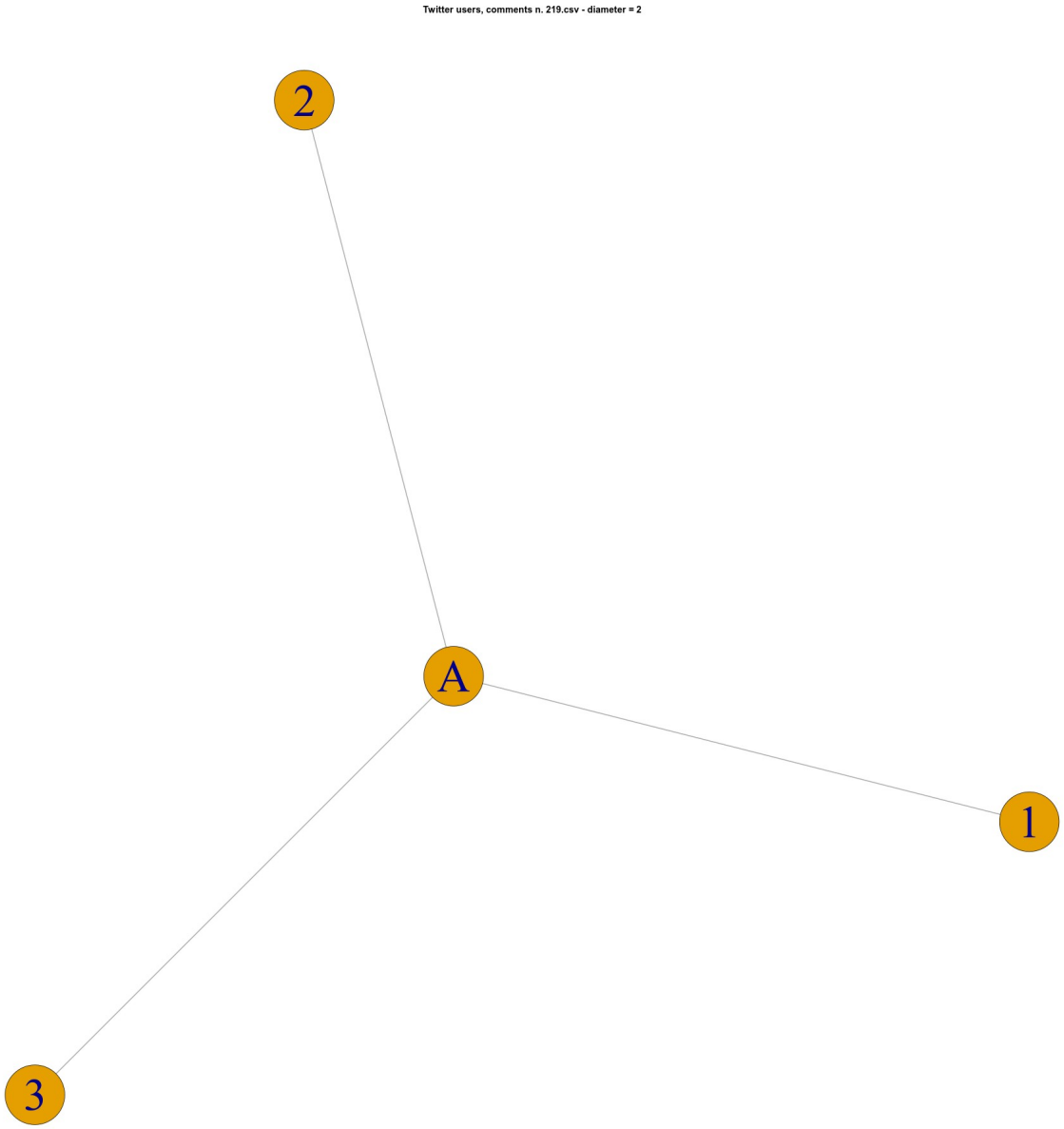


Diagram Twitter network, comment n. 222 – diameter = 2

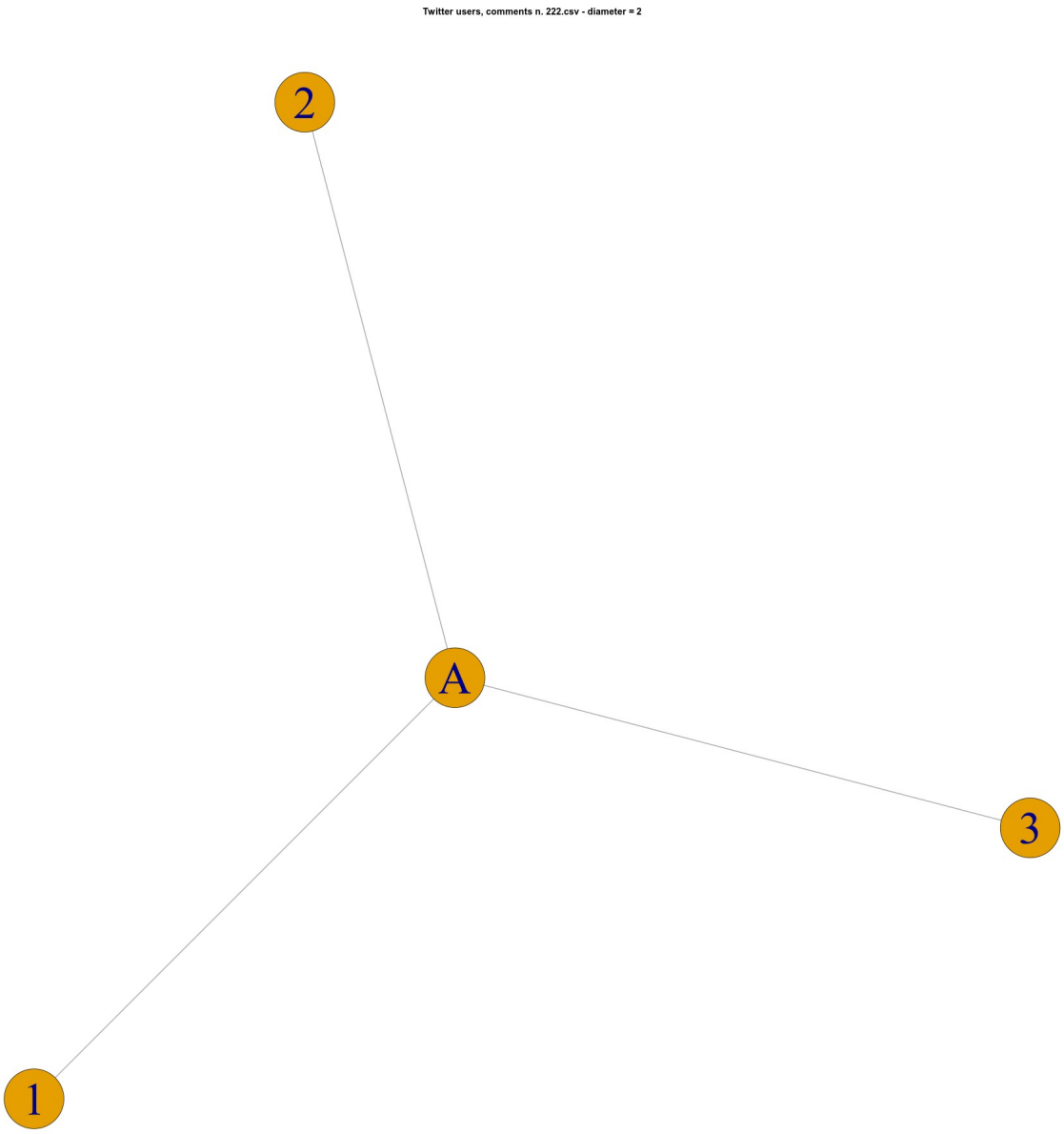
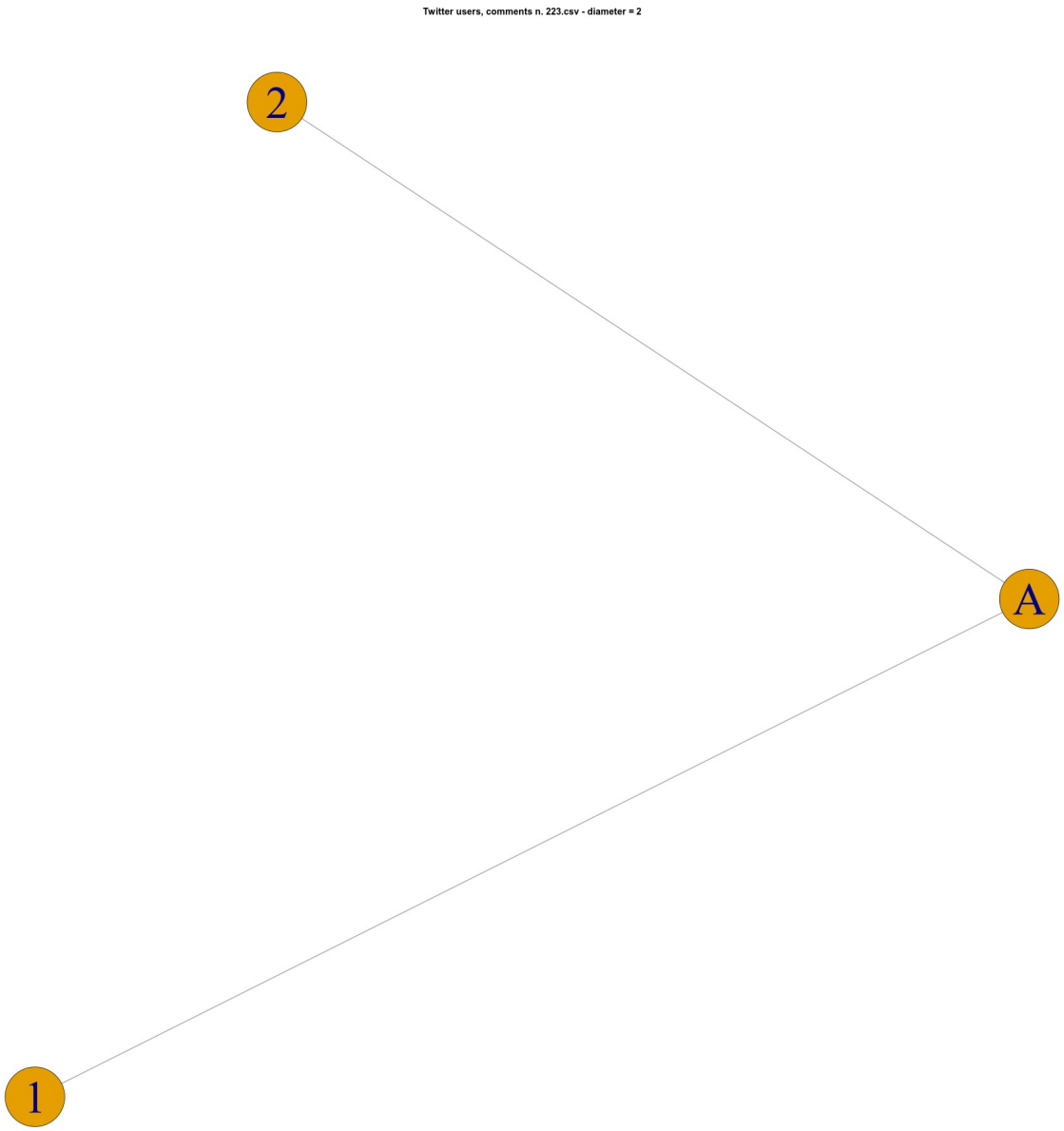


Diagram Twitter network, comment n. 223 – diameter = 2



References

Dhiraj M. (2018) , *Twitter, Social Communication in the Twitter Age* . Cambridge (UK): Polity Press

Kearney, M.W., (2018). rtweet: Collecting Twitter Data. Retrieved from https://cran.r-project.org/package=rtweet

Twitter, (2019). About Twitter's APIs. Retrieved from https://help.twitter.com/en/rules-and-policies/twitter-api