From raw data to temporal graph structure exploration

MSc in Business Analytics - AUEB
Student: Vretteas Stylianos – p2822003
Course: Social Network Analysis
Homework 2: From raw data to temporal graph structure exploration

Task1 - DBLP co-authorship graph

The initial file “authors csv” consists of 2.793.928 rows and 4 columns. The columns are given the names “YEAR_PUBLISHED”, “PAPER_TITLE”, “PLACE_PRESENTED” and “AUTHORS_LIST”. The file is a list of academic papers and provides information regarding the the year of the paper, the conference where the paper was presented and the authors that wrote it. Our task is to manipulate the given data and construct edge lists dataframes for the last 5 years in order to analyze these relations as graphs. Regarding the weight of each relation, we have to find the frequency of how many times an author has co-wrote a paper with another author.

For the above ETL (extraction, transformation, loading) procedures, Python language was used and Jupyter notebook as a compiler. After the loading of the file, the first filter that was implemented was at the “PLACE_PRESENTED” column, papers that were not presented in the «CIKM", "KDD", "ICWSM", "WWW", "IEEE BigData" conferences were excluded.
Afterwards, the column “YEAR_PUBLISHED” was filtered at these values 2016, 2017, 2018, 2019, 2020, 2021. Given the fact that after this action there is no paper presented at 2021, the 5-year period that was chosen was 2016-2020. Finally, after the removal of NAs values (1 row for the above) the final dataframe consists of 8.720 rows and 4 columns.

In order to construct the edge list objects whose structure is a three column dataframe which describes the citations from an author to another author and the frequency of this relation as a pair, column “AUTHORS_LIST” has to be further manipulated.

A subset dataframe of each year and then a dictionary containing as keys the number of the row and as values the splitted names of the authors with the comma “, “as delimeter is created from the “AUTHORS_LIST” column. Then, a list for each year is created with the values of these dictionaries in order to count the frequencies of each pair. The general idea is to create a list of lists and compare each element of each list with all other elements to count the frequencies of relations and obtain the weight of each row. After the calculation of the most frequent pairs of elements, the final consolidated dataframes are constructed with structure “From”, “To” and “Weight” columns. After this, a simple text manipulation is implemented to remove the parentheses and the commas from the author names that were have been left from the raw data. Dataframes consists of 9666, 10908, 12622, 18071, 18966 rows for years 2016 to 2020 respectively.
Finally, I export these cleaned dataframes as csv files in order to load them into R-Studio and begin the graph analysis.

Task2 - Average degree over time

For the 2nd task the creation of plots that will describe the evolution of different metrics of the graphs is requested. Specifically:

- **Number of vertices**

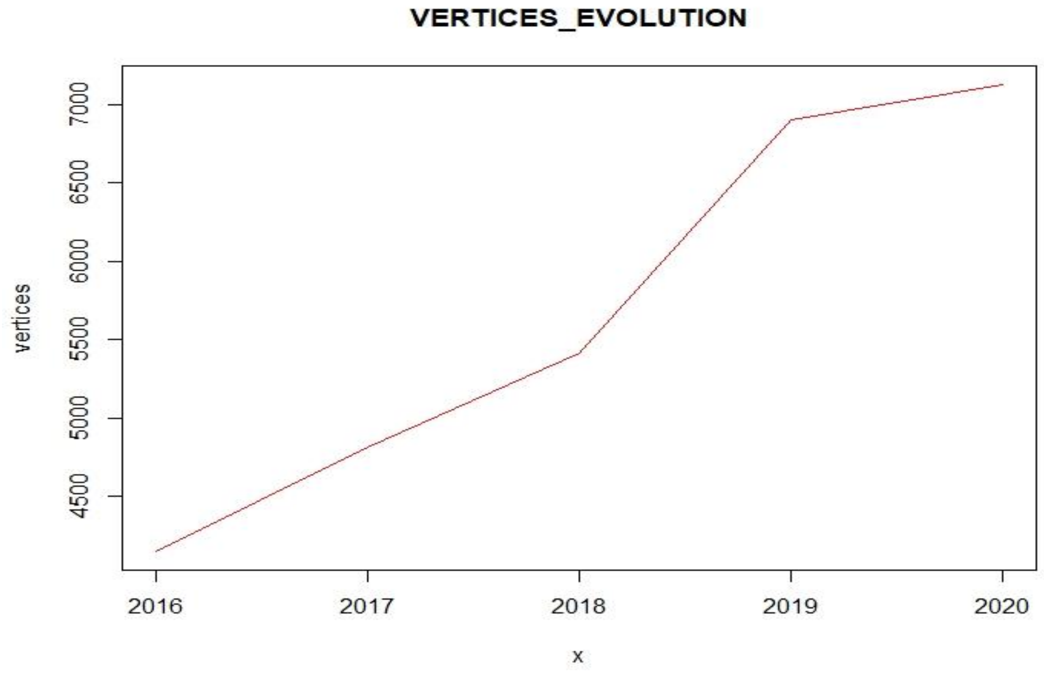


Figure 1

- **Number of edges**

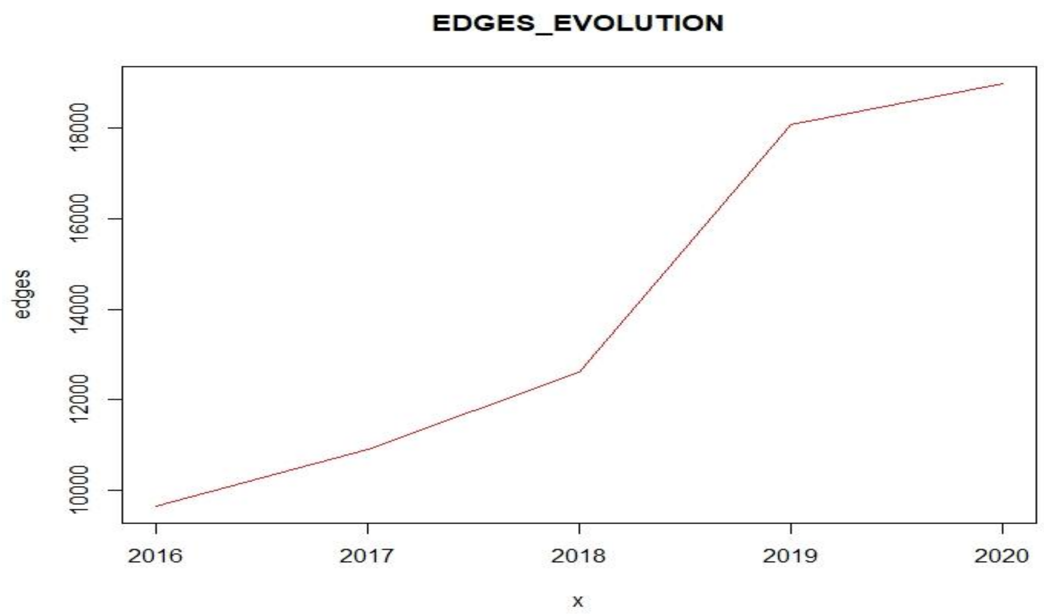


Figure 2

- **Diameter of the graph**

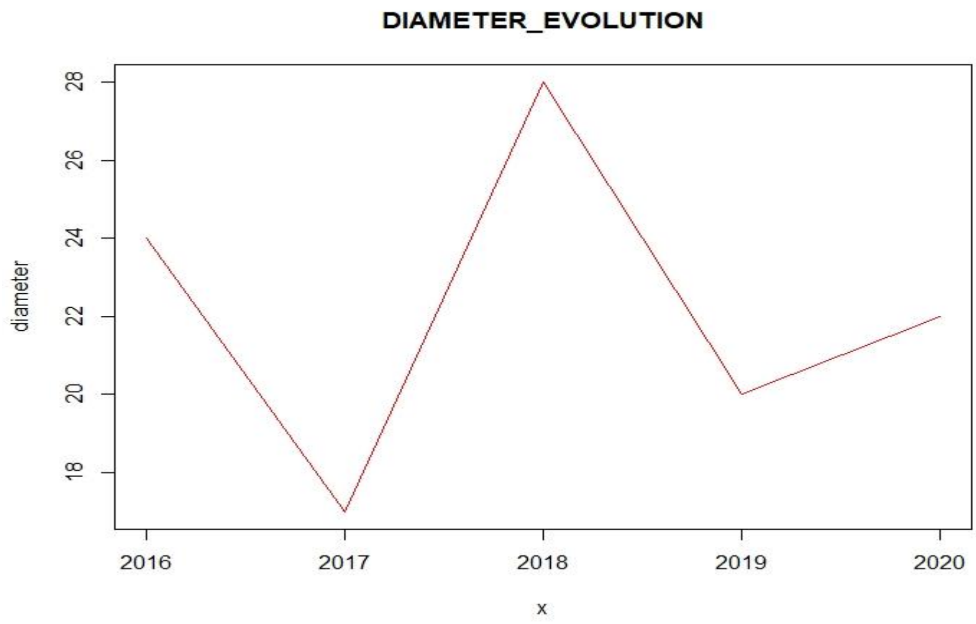


Figure 3

- **Average degree (simple, not weighted)**

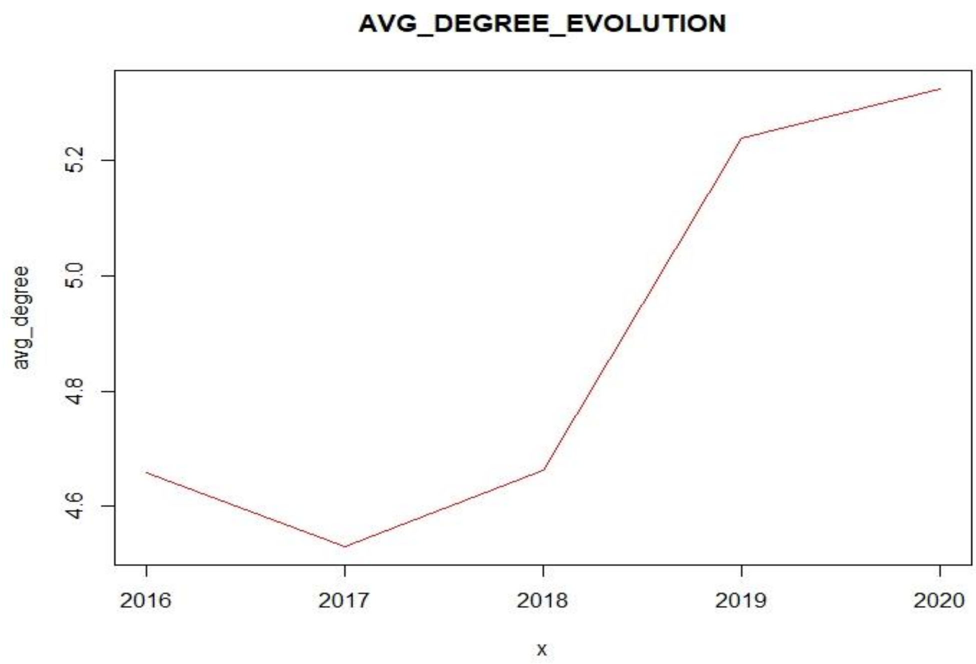


Figure 4

From the above plots, we clearly see that there is an increase into the number of vertices, edges and average degree of the networks across these 5 years. Regarding the diameter of the networks which is the shortest longest paths between the two most distant nodes of the network there is a great variation into the values each year.

Specifically, for the nodes of the graphs the range varies from 4150 to 7125,
for the edges varies from 9666 to 18966 while the average degree ranges from value 4.65 to value 5.32. Regarding the diameter the maximum value is 28 while the minimum is 17 but there is no time proportional but random.

Task3 - Important nodes

For the 3rd task we have to find the top 10 authors each year taking into consideration two different metrics, first their degree and then their PageRank values.

Top 10 authors based in degree each year

2016

```
> print(degrees_top10_2016)
     Philip S. Yu    Jiawei Han 0001     Hui Xiong 0001        Jieping Ye Naren Ramakrishnan      Yi Chang 0001
               46                 41                 39                32                 32                 31
        Jiebo Luo        Rayid Ghani      Chang-Tien Lu    Yannis Kotidis
               29                 28                 25                25
```

Table 1

2017

```
> print(degrees_top10_2017)
     Philip S. Yu    Jiawei Han 0001     Hui Xiong 0001      Yi Chang 0001 Claudio Rossi 0003      Clemens Mewald
               44                 42                 38                 32                 32                 31
   Heng-Tze Cheng       Martin Wicke       Mustafa Ispir      Zakaria Haque
               31                 31                  31                 31
```

Table 2

2018

```
> print(degrees_top10_2018)
   Philip S. Yu Jiawei Han 0001         Kun Gai  Wenwu Zhu 0001   Jure Leskovec  Jing Gao 0004 Chao Zhang 0014  Xing Xie 0001
             70              37              35              28              27             27              27             26
   Enhong Chen     Qi Liu 0003
             25              25
```

Table 3

2019

```
> print(degrees_top10_2019)
   Philip S. Yu Weinan Zhang 0001     Hui Xiong 0001        Jieping Ye     Jie Tang 0001   Jiawei Han 0001      Yong Li 0008
             69                59                 49                41                39                37                36
    Enhong Chen      Jingren Zhou           Jian Pei
             36                35                 35
```

Table 4

2020

```
> print(degrees_top10_2020)
  Jiawei Han 0001      Hongxia Yang      Hui Xiong 0001       Xiuqiang He          Ji Zhang      Peng Cui 0001
               69                43                42                41                40                 39
Christos Faloutsos     Wei Wang 0010        Jieping Ye       Ruiming Tang
               38                38                37                35
```

Table 5

Top 10 authors based in PageRank each year alongside their PageRank rate

2016

```
> p_rank_df_2016[1:10,]
            Character    Page_Rank
1        Philip S. Yu 0.0017288334
2      Hui Xiong 0001 0.0014581015
3     Jiawei Han 0001 0.0014119510
4          Jiebo Luo 0.0013099364
5         Jieping Ye 0.0010027077
6       Yi Chang 0001 0.0009601005
7       Hanghang Tong 0.0009272920
8  Christos Faloutsos 0.0009216757
9     Maarten de Rijke 0.0009158533
10       Jiliang Tang 0.0009155034
```

Table 6

2017

```
> p_rank_df_2017[1:10,]
           Character    Page_Rank
1       Philip S. Yu 0.0014558956
2    Jiawei Han 0001 0.0013585699
3     Hui Xiong 0001 0.0010997688
4      Jure Leskovec 0.0010681579
5          Jiebo Luo 0.0009454158
6      Hanghang Tong 0.0009285808
7       Jiliang Tang 0.0007750644
8      Yi Chang 0001 0.0007711858
9    Chao Zhang 0014 0.0007510406
10     Ingmar Weber 0.0007208090
```

Table 7

2018

```
> p_rank_df_2018[1:10,]
           Character    Page_Rank
1      Philip S. Yu 0.0019809631
2  Jiawei Han 0001 0.0009301987
3     Jure Leskovec 0.0008753490
4    Wenwu Zhu 0001 0.0007842984
5  Chao Zhang 0014 0.0006775310
6     Xing Xie 0001 0.0006263373
7     Jing Gao 0004 0.0006259877
8      Martin Ester 0.0006201636
9    Yiqun Liu 0001 0.0006143691
10          Kun Gai 0.0006129884
> 
```

Table 8

2019

```
> p_rank_df_2019[1:10,]
            Character    Page_Rank
1        Philip S. Yu 0.0015871036
2      Hui Xiong 0001 0.0009633261
3   Weinan Zhang 0001 0.0008767308
4          Jieping Ye 0.0007255196
5       Hanghang Tong 0.0007021244
6     Jiawei Han 0001 0.0006855583
7       Peng Cui 0001 0.0006574207
8       Jie Tang 0001 0.0006517701
9        Enhong Chen 0.0006377621
10    Gerhard Weikum 0.0006257373
> 
```

Table 9

2020

```
> p_rank_df_2020[1:10,]
                Character    Page_Rank
1         Jiawei Han 0001 0.0010753255
2          Hui Xiong 0001 0.0007594661
3            Hongxia Yang 0.0007284981
4  Elke A. Rundensteiner 0.0006983864
5            Yong Li 0008 0.0006821198
6              Jieping Ye 0.0006800497
7           Peng Cui 0001 0.0006533883
8             Xiuqiang He 0.0006465968
9             Ji-Rong Wen 0.0006450074
10          Jiliang Tang 0.0006423610
> 
```

Table 10

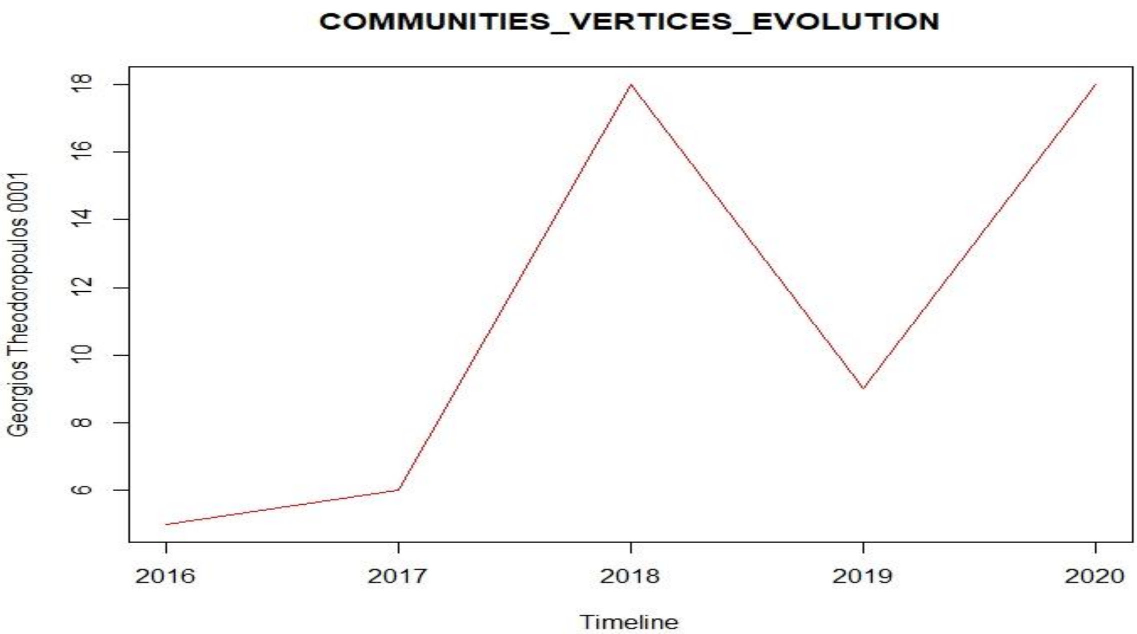
Both method rankings have similarities despite they differ in their calculation procedure.

Specifically for every year the first author is the same. Moreover, regarding the degree method there is no intersection into the top 10 authors every year on the other hand If we take into consideration their PageRank, author Jiawei Han 0001 is in the intersection into top 10 based on Pagerank (this means that he is included in the top 10 authors every year).

Task4 – Communities

For the 4th task we have to detect communities by applying clustering algorithms “fast_greedy”, “infomap” and “louvain”. We apply each algorithm into all networks and compare the system time of each process. Below, (table 11) the detailed R output of each algorithm run. It is clear that the Louvain algorithm is the most efficient in time while the infomap is the slowest. From, these algorithms I will choose the Louvain algorithm for community detection due to the fact that is time-efficient and compared to the fast greedy algorithm is better because fast greedy can ignore some communities due to its nature of processing the graphs.

For the 2nd part of the 4th task, we construct a vector which contains all authors that belong to a community each year. Randomly we choose the author "Georgios Theodoropoulos 0001" from this vector, then we find the communities where the author belonged to each year and then we plot the size of these communities each year.



System time – Algorithm Comparison

```
C:/Users/svret/py_files/SNA - project2 - ETL/
> system.time(cluster_fast_greedy(network_2016))
   user  system elapsed 
   0.03    0.00    0.03 
> system.time(cluster_infomap(network_2016))
   user  system elapsed 
   0.71    0.00    0.71 
> system.time(cluster_louvain(network_2016))
   user  system elapsed 
   0.02    0.00    0.01 
> # 2017
> system.time(cluster_fast_greedy(network_2017))
   user  system elapsed 
   0.03    0.00    0.03 
> system.time(cluster_infomap(network_2017))
   user  system elapsed 
   0.82    0.02    0.83 
> system.time(cluster_louvain(network_2017))
   user  system elapsed 
   0.01    0.01    0.03 
> # 2018
> system.time(cluster_fast_greedy(network_2018))
   user  system elapsed 
   0.03    0.00    0.04 
> system.time(cluster_infomap(network_2018))
   user  system elapsed 
   1.03    0.01    1.05 
> system.time(cluster_louvain(network_2018))
   user  system elapsed 
   0.01    0.00    0.01 
> # 2019
> system.time(cluster_fast_greedy(network_2019))
   user  system elapsed 
   0.06    0.00    0.06 
> system.time(cluster_infomap(network_2019))
   user  system elapsed 
   1.50    0.03    1.55 
> system.time(cluster_louvain(network_2019))
   user  system elapsed 
   0.03    0.00    0.03 
> # 2020
> system.time(cluster_fast_greedy(network_2020))
   user  system elapsed 
   0.04    0.02    0.06 
> system.time(cluster_infomap(network_2020))
   user  system elapsed 
   1.52    0.02    1.53 
> system.time(cluster_louvain(network_2020))
   user  system elapsed 
   0.03    0.00    0.03 
```

Table 11 – System time for each algorithm run

We see that the at 2016 and 2017 George Theodoropoulos' community consisted of 5 authors and 6 authors respectively. Then there is a big increase at 18 authors in 2018 which is also the maximum value followed by a "valley" at 10 authors in 2019 and finally at 2020 there are 18 authors again.

Regarding the similarities in each year, we compute the intersection of each year vector and found out that the below authors were included each year.

Similar authors

```
> example_similarities
[1] "Andrew Stephen McGough"      "Georgios Theodoropoulos 0001" "Ibad Kureshi"                 "John Brennan"
[5] "Stephen Bonner"
>
```

Table 12

For this task, R code was created in such way that the user can compare the above for each one author that is included in the intersection vector "instersect_author". The user can choose an author of his choice from the vector and see the evolution of the communities the author was part of and his co-authors in these communities in all the years. (Eg. George Theodoropoulos was part in the same communities every year with the five (5) authors from *table 12*.

Finally, we plot the mid-sized communities which are the communities that consists between 50 & 90 authors, we set the color for each community and we make an induced subgraph in order to visualize better the results. Below are the outputs for the communities that were formed each year.

Communities of 2016

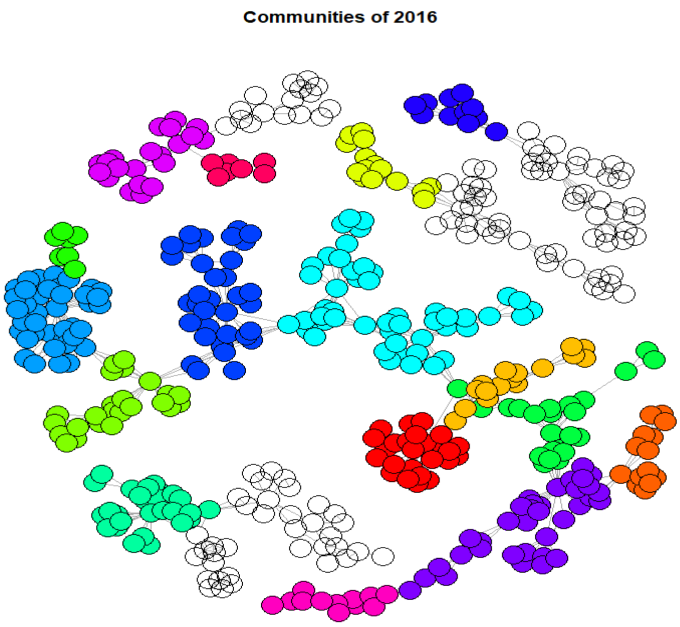


Figure 5

Communities of 2017

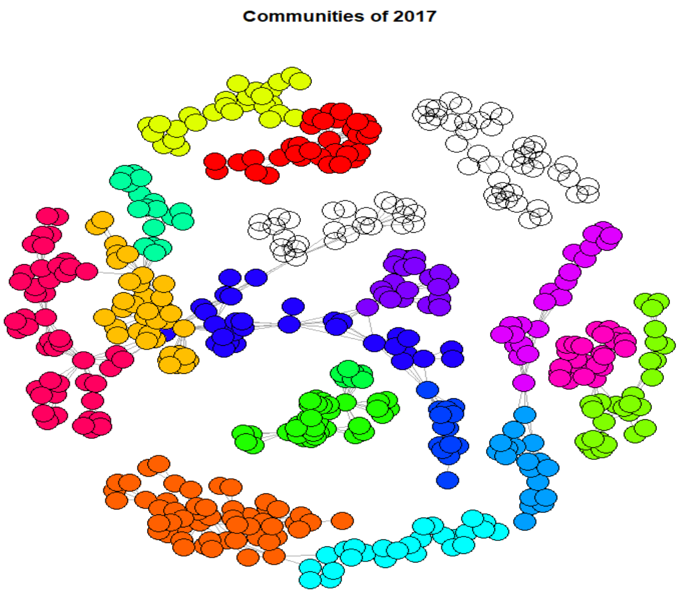


Figure 6

Communities of 2018

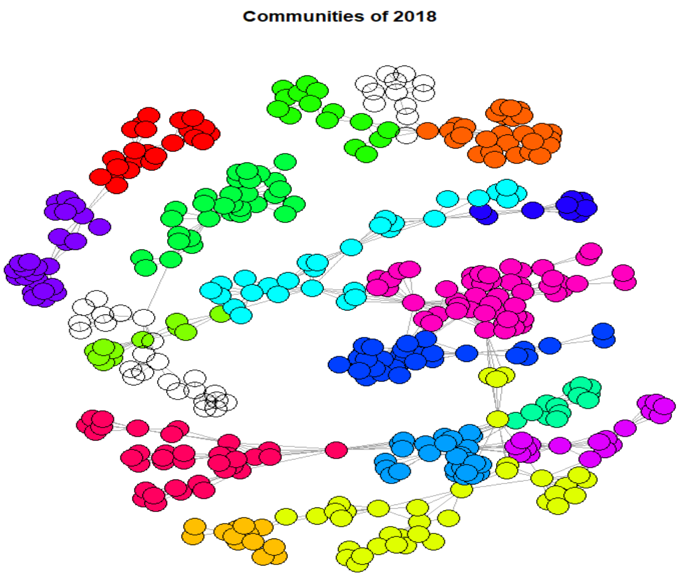


Figure 7

Communities of 2019

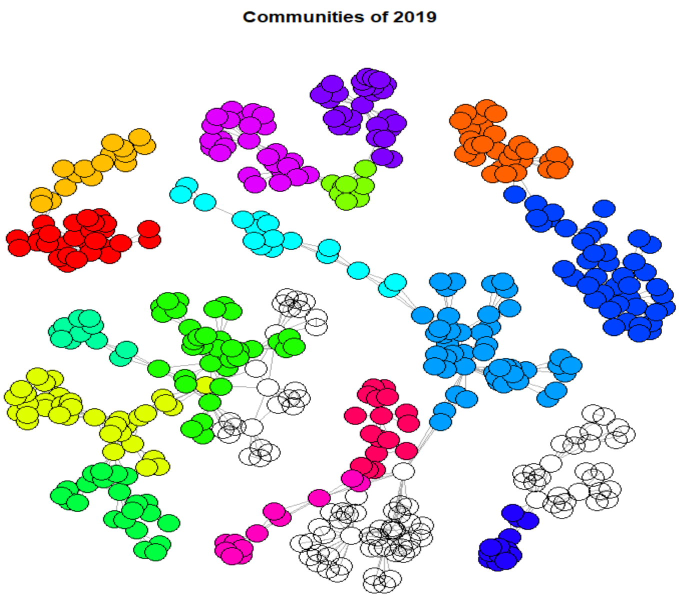


Figure 8

Communities of 2020

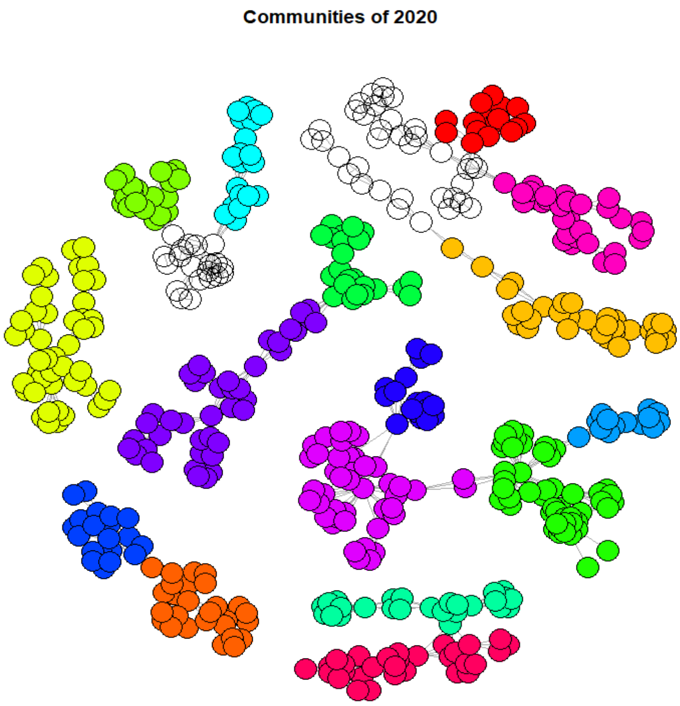


Figure 9