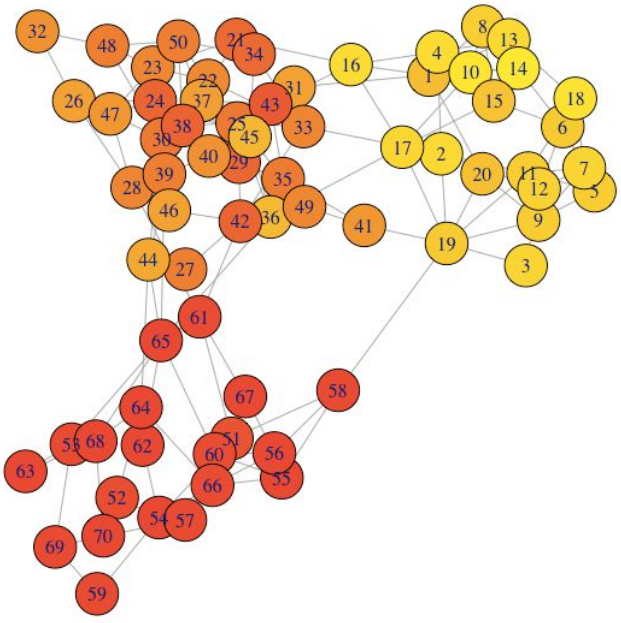
GTDMO - 4th assignment
Group 3

Abstract

The assignment will be divided into two parts. First, we will examine the Social Network in order to identify if there are communities across the population and understand if their members share common characteristics or play similar roles within the graph. After clustering the network, we will evaluate if there are hubs, that is any node with a high degree of connection, as that would make them a focal point in the community.
After analysing the population, we are interested in simulating the spread of a fake news starting from one single person, that we consider "infected by the fake news" at time 0, and in particular to understand if there would be any differences in terms of mean infection time if the spread begins from different communities.

Here we display the provided graph colored by age; this representation allows us to get a first feeling of what the clusters may look like.



Methodology

The proposed problem concerns a sample of 70 people of different ages and gender, which are in contact with each other through many means (in person, social media, videoconference,…), and they thus create what we call a *Social Network,* represented by a graph. In fact, a *Social Network* is made by three essential characteristics:

- There is a collection of *entities* who take part in the network, in our case the people, which are going to represent the nodes of the graph;
- the *entities* are connected by a *relationship* that we can identify, in our case whether they spend together at least 5 hours a week, which are the edges of the graph;

- the *relationship* is non-random, but there are local communities i.e. there is a structure behind the graph (*locality assumption*).

In this case, we have an unoriented simple graph with a single node type.
In mining *Social Network* graphs we are interested in finding local communities, which are groups of vertices which probably share common properties or play similar roles within the graph, thus they will be strongly related to themselves and will have a weak connection with other communities. Given the variety of individuals (age ranges from 7 to 89), in our problem we expect to find communities of people which are maybe connected because of attending the same school, or working in the same building, or simply chatting with each other without having ever met, maybe because they share the same interests and visit the same blogs.

In a simple undirected graph it would be particularly easy to identify the components, especially by looking at the *adjacency matrix*, which is the matrix $D=(d_{ij})$ having as entries the number of edges (relations) between two vertices (people) v_i and v_j. Given that the graph is simple, there will only be entries corresponding to 0 or 1, where the diagonal will be made by all zeros (there are no loops), and the communities will be identified by the submatrices of 1 corresponding to the components of the graph. Outside the submatrices, there will be only zeros. However, it would be difficult to identify the communities in this way when the rows and the columns are not in the right order; for this reason, we will rely on the "*strength of weak ties*" and exploit the ***Girvan-Newman algorithm*** for undirected and unweighted graphs.

The goal is to find densely linked clusters of nodes which could be progressively splitted into smaller communities, to get a hierarchical decomposition of the *Network*. Before doing so, let us first define *edge betweenness* (*btw* from now on) as the number of shortest paths passing over an edge, between any two nodes. Thus, *btw* is an indicator of the centrality of an edge with respect to a graph, and the edges with the highest btw are those connecting clusters which are more far from one another. Going back to *Girvan* and *Newman,* they created an algorithm which recursively computes the btw and removes the edges displaying the highest values; while edges are being removed, we can identify the communities in the subgraphs which are being generated. At this point, the algorithm provides a way of computing the *betweennesses* based on a credit mechanism, which requires first to level the nodes according to the steps needed to arrive there from the starting node, and then to compute the shortest paths. The btw will be the sum of the credits given to all the edges by repeating this procedure for each node.

Then, to select the right number of clusters, we first define *modularity* (Q) as a measure of how well a network is partitioned into communities; hence, given a partitioning of the *Network* into groups $s \in S$, we can compute modularity, which will be a function of the original graph and of its division in clusters, as:

$$Q(G,S) = (1/2m)\,\Sigma_{s\in S}\Sigma_{i\in S}\Sigma_{j\in S}(A_{ij} - (k_i k_j)/2m)$$

where k_i and k_j are the degrees of the nodes i and j, m is the number of edges, and A_{ij} is the number of edges between i and j (being a simple graph, A_{ij} will take value 1 if $i \to j$ and 0 else). Notice that we are multiplying by a *proportionality constant*, which is a *normalizing cost* allowing us to always keep Q between -1 and 1, so that we can interpret it in a more agile way. Given the formula, Q is proportional to the difference between the number of edges

within group S and its expected value, which is computed by generating a rewired network G' with the same degree distribution but random connections.
This means that if Q is positive the connections are certainly non random, and, as a rule of thumb, if $Q \geq 0.3\text{-}0.7$ we are observing a significant community structure.
Having defined what modularity is, we can identify the best splitting of the graph as the one presenting higher Q, and the so formed clusters will be the *communities* of the *Social Network.*

Once the communities are identified, it is interesting to analyze them looking for *similarities among the members*; we can evaluate the composition of the clusters in terms of gender distribution, mean age and variance, and also by choosing a similarity index and finding the mean and maximum similarity within each cluster. In this report, we decided to exploit the ***Inverse Log-Weighted Similarity***, because it accounts for the fact that sharing a hub may be also happening by chance i.e. a hub with higher degree provides less information about similarity between vertices. According to this Index, two vertices are more similar if they share low-degree common neighbours.

When mining a *Social Network,* it is crucial to check if there are any *hubs*, which are nodes with a particularly high degree (number of edges having the node as end vertex); they are the "*centers*" of the communication between individuals. For instance, if a *fake new* gets to a hub, we expect it to spread faster, as hubs connect a high number of individuals.

Indeed, *fake news* are something really common nowadays, and it is well known that, given that our world is becoming increasingly connected, they tend to spread really fast; this happens because individuals tend to share information before even checking its reliability.
It would be really interesting then to analyze what would be the *mean speed* at which *fake news* would spread, in relation to where they start from.

In order to do so we would have to simulate a random walk in the given population exploiting ***Markov Chains***; however, the properties of the MC should be discussed.

- Given that the probability of getting "infected" by *fake news* changes depends on the number of infected neighbours, it follows that it also depends on time, and thus we will be dealing with an *inhomogeneous* MC, whose transitional matrix is changing at each step.
- The MC will also be *periodic*: once infected, an individual remains infected, thus remains in its state, but a "healthy" individual has a positive probability to go back to its state only until 5 or more of its neighbours are infected. After that, any individual will have 0 probability to go back to their initial state in a finite number of steps.
- Moreover, the MC will be *reducible*, as not all of its states intercommunicate, since once "infected" there is no possibility to go back to the initial state.

The simulation is discussed in the next sections.

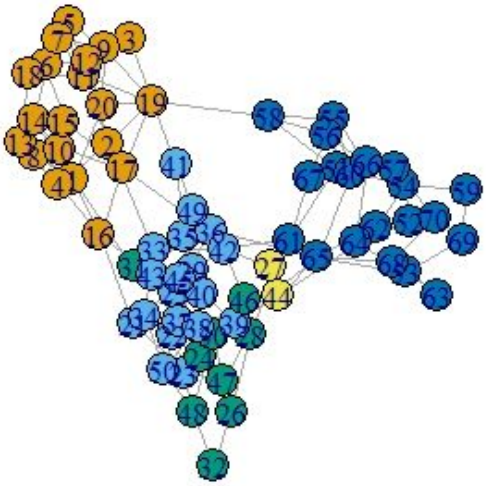
Computation and Empirical Results

We will use R as a tool for computation, since we can automatically plot graphs, compute easily betweenness and modularity and analyze the characteristics of each group found, if any. Before plotting our graph we must first fix the position of the nodes using a particular algorithm. Otherwise, since the graph is unweighted and unoriented, every time we would observe a different positioned graph: this wouldn't change the modularity or the betweenness, since at every rearrangement the degrees of each node are held fixed, but would cause confusion, and could cause misinterpretation about which node belongs to which community if we choose to plot a graph that shows the division in clusters depicted with shadow regions. The population's distribution is:

```
> mean(vertex_attr(g)$age)
[1] 47.87143
> table(vertex_attr(g)$gender)
 F  M
35 35
```

To Identify if there are communities in the graph, we follow the Girvan-Newman algorithm, and get that the maximum modularity for this Network, which provides the optimal clustering, is:

```
modularity(eb)
[1] 0.569237
```



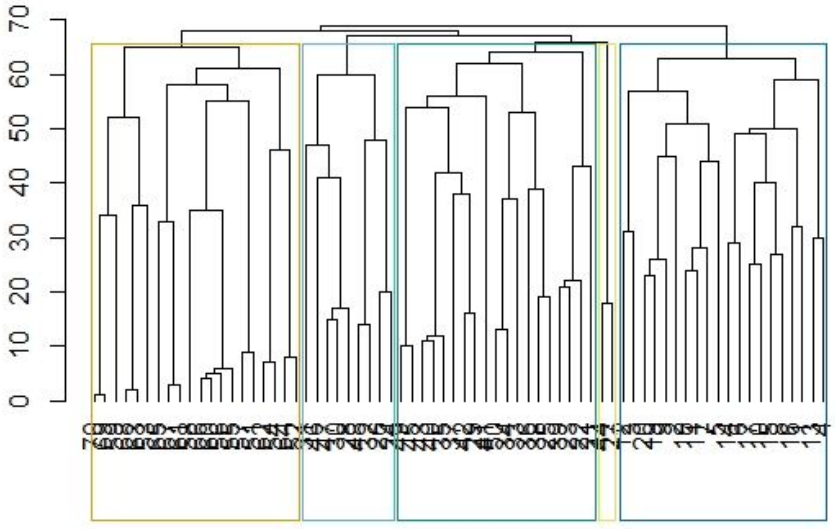
corresponding to 5 clusters.
We then proceed to analyze these communities, to see if there is similarity between the members,
and to check if we can find patterns in terms of the qualities attributed to them. To do so, first we assign each node to its subgraph through an algorithm based on *membership*, which is an index of the belonging of each individual to its cluster.
We can now look for common attributes by exploring the communities' descriptive statistics, and by computing the *Inverse Log-Weighted Similarity Index*; we only report here the mean and the maximum index value for each group, since the output of the function would be a matrix with entries the similarity indices for all the individuals. The results are summarized in the following table:

	AGE				GENDER		Number of members	SIMILARITY	
	mean	min	max	variance	relative frequence			mean	max
					F	M			
C1	80	72	89	22.46	50%	50%	20	0.3964363	2.189697
C2	46	27	71	146.88	63%	37%	19	0.3947169	2.164043
C3	51	33	60	88.75	22.22%	77.78%	9	0.4625577	2.352.934
C4	54	45	63	162	50%	50%	2	0.3176277	2.152.909
C5	15	7	23	18.21	50%	50%	20	0.3518975	2.152.909

Using the *Girvan-Newman method*, we were able to identify 5 communities:

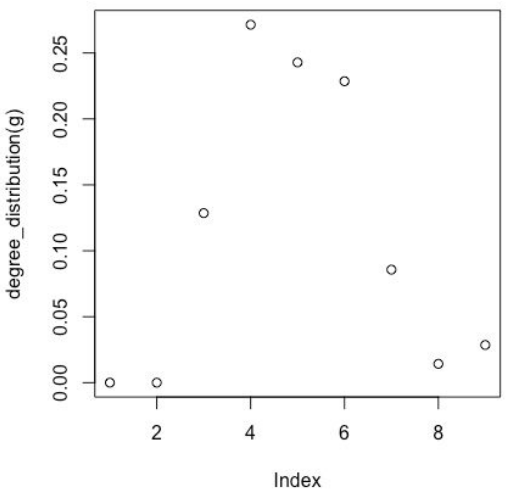
- C1 - *"Elderly people"* : the average age is 80 years old, the age variance is pretty low, and there is an equal number of females and males, meaning that these individuals are well clustered by age and form an heterogeneous community gender-wise. The group also displays the second higher mean similarity.
- C2 - *"Heterogeneous mainly females" :* this community displays the higher variance, with an age that ranges from 27 to 71; it is also not balanced in terms of gender, showing a prevalence of female members. Apparently, this is the most heterogeneous community.
- C3 - *"Half age mainly males" :* this group displays the higher mean and maximum similarity within its members, and is the second smaller one. Females and males are not balanced, with a prevalence of males.
- C4 - *"Half age"* : this cluster only contains two individuals, which have a mean age really close to the one of C3, and a really high variance in age, equal to 162.
- C5 - *"Young people"* : this community is the one with the lowest variance, and the higher numerosity (together with C1); it also shows the lower mean and minimum age, having 7 years old children as members.

We also display the dendrogram where the identified communities are highlighted.



Now that we identified the communities and discussed their common characteristics, we can proceed by looking for *hubs*. In order to do so, we compare the degree of all the nodes, keeping the division into clusters. In this way we can easily select the most connected nodes and thus identify which would represent *hubs*, if any. There are two nodes (19 and 65) with degree 8 (maximum degree) and one with degree 7 (42),while the most frequent degrees are 3, 4, and 5. Considering that there are 3 out of 70 nodes with degrees higher than 7, these three nodes could be considered the *hubs* of the *Network.*

```
> table(degree(g))
 2  3  4  5  6  7  8
 9 19 17 16  6  1  2
```



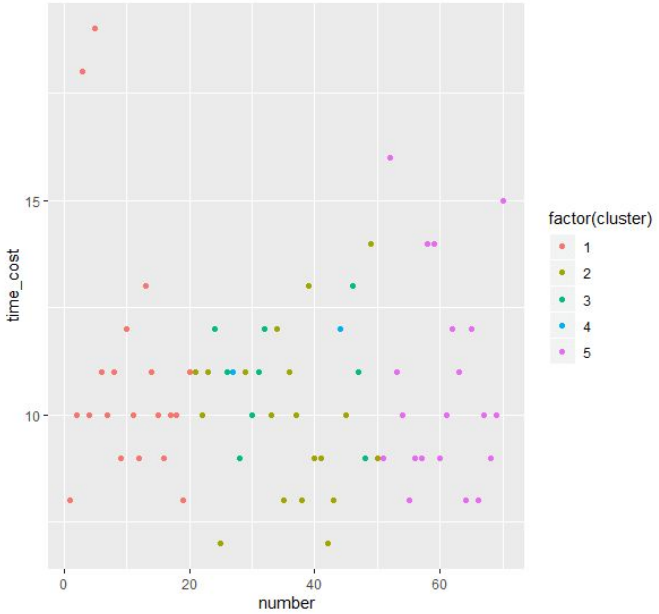
For the sake of brevity here we only reported the distribution table and plot of the nodes' degrees.

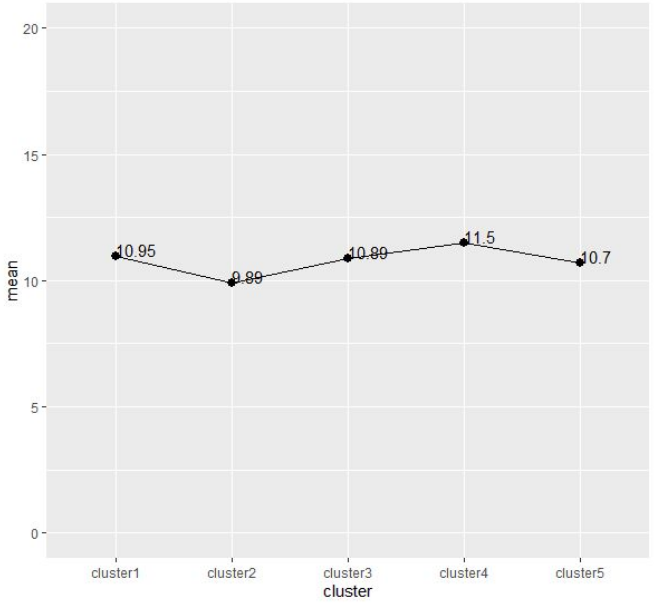
Simulation

In this case, the fake news spreads in the population with the probability P. From P, we can know that, at time 1, the infected neighbor has 20% of the probability to get infected. In following steps, the probability of an uninfected person is 0.2 times the number of infected neighbors. Thus, once reaching a step where an uninfected person has more than or equal to 5 infected neighbors, he will certainly get infected. Meanwhile, once one person gets infected from other people, in the next step, he starts infecting as the other person did with him. In this mechanism, the more people get infected, the faster the fake news spreads since once 5 neighbors of one person get infected then this person 100% gets infected and starts to infect others.

To better understand this procedure, we build a function in R to simulate the model. In this function, the arguments are the graph and the starting point and we compute the steps needed to infect the whole population from the starting person. This function can be split into two parts. The first part is how the infected person infects his neighbors. And the second part is that after one step, we have to update the states of the population so that in the next step, everyone has the updated states and the probability one person gets will affect the next step. In fact, the spread of the fake news is a random walk, which means that the state of present step only depends on the last step, so it is necessary to update after infection action. The loop inside the function retakes the two parts until all the people get infected, and returns the steps it takes.

Using this function, we can get all the steps taken starting from every person. Plotting the steps according to the clusters, we can find that the speed of the spread starting from each of the identified communities doesn't differ a lot. When we compute the mean speeds of the spread of each cluster, the results are quite similar.





Conclusions

Through our *Social Network* analysis we are able to find the optimal clustering according to and the *Girvan-Newman method,* thus to identify the main 5 communities within it, based on the individuals' attributes.

From the inspection of the nodes' degree we found two nodes (19 and 65) with degree 8, and one node (42) with degree 7; considering that these are the only 3 nodes out of 70 with a degree higher than 7, they represent the *hubs* of the *Network* i.e. they are the centres of the communication.

After simulating the spread of the fake news in this population, we find that there is no great difference in the mean speed of the spread among people from different clusters as the starting point.

What we noticed however is that C3 and C4 are similar in terms of mean age, and by looking at the dendrogram we also observe that these two groups have a common edge right above the Q threshold we used for clustering. Given these facts, we considered that we could actually cut the clusters at a higher level and only keep 4 of them, with a gain in terms of *within* variance. In fact, when including those two individuals in C3, we get a total C(3-4) variance equal to 88.67273, which is only slightly higher than the variance of C3 alone, and we don't have the great variance displayed by C4 anymore.
Thus, considering the splitting made by the *Girvan-Newman method*, we could instead consider clustering the *Social Network* in 4 more balanced clusters only, given by:

- Elderly people
- Mainly females with mixed age
- Mainly men at half age
- Young people