# Exercise one: Analyzing an offline and online social networks

## Question 1 (3 points):

* Find out the node ID of

1. highest degree = S54
2. highest betweenness = S37
3. highest closeness = S37
4. highest eigenvector in the Highschool network = S110

* Afbeelding met diagram

  Automatisch gegenereerde beschrijvingAfbeelding met diagram

  Automatisch gegenereerde beschrijvingHighlight the above nodes in the Highschool network;

Afbeelding met grafiek, diagram

Automatisch gegenereerde beschrijving

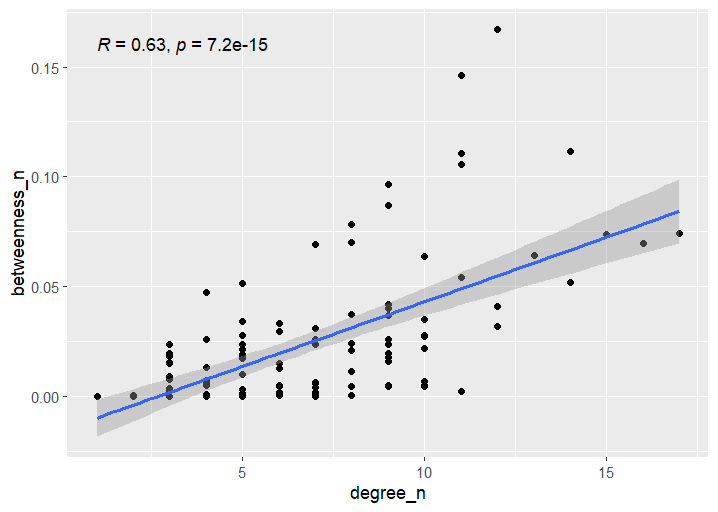
* Explain why these metrics identify the same node or different nodes as the most central one.

All these metrics are calculated in a different way which causes different outputs for the most central node. However according to the betweenness and closeness S37 is the most central. This is probably because these 2 methods both use the shortest path in deciding which node is the most central.

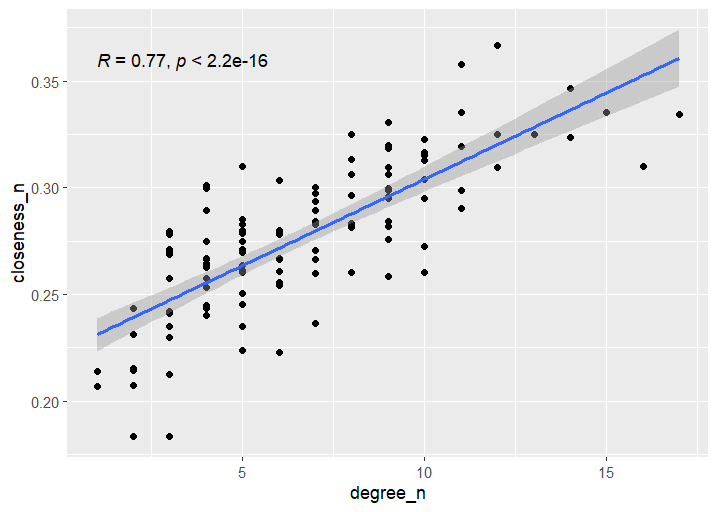
## Question 2 (5 points):

* Study the correlations for all the nodes in the Highschool network between

1. degree and betweenness,



1. degree and closeness,



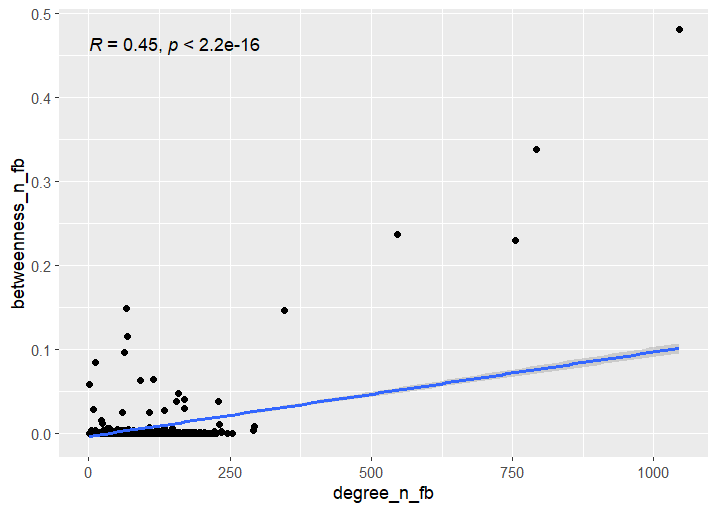
1. degree and eigenvector

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

* Study the correlations for all the nodes in the Facebook network between;

1. degree and betweenness,

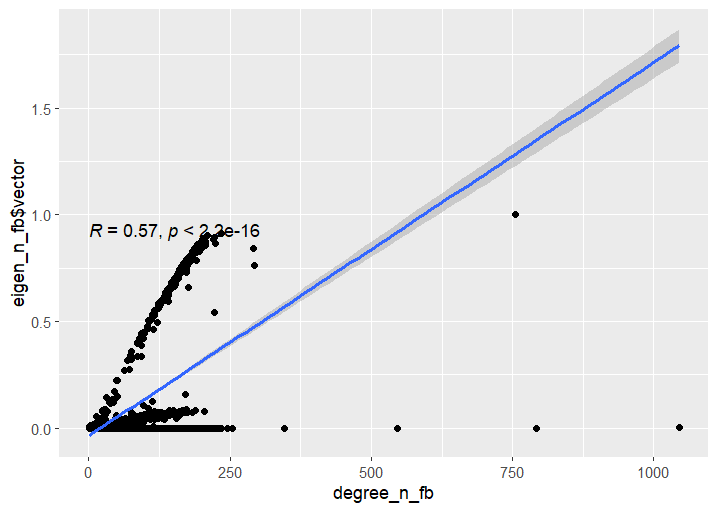


1. degree and closeness,

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

1. degree and eigenvector



* From the above results, how well do different metrics correlate with each other? Which centrality metric will you use and why? Overall the correlations in the high school network is higher. For the high school network its best to use the closeness centrality since it correlates the best with the degree. For the Facebook network this would be the eigenvector centrality, however this might differ if the outliers are removed from the network.

## Question 3 (5 points):

* For both the Highschool and Facebook networks, calculate the shortest path lengths between every pair of two nodes. How many percentage of nodes can be reached within 6 path lengths? Does “six degree of separation” apply to each network?
* Study the degree distribution of these two networks, are they similar? Then use degree distribution to explain the degree of separation you answered above.

## Question 4, 4 points:

Now you will analyze the meso-scale structure of the Highschool network. First, check out the node attributes of Highschool network. You can find 1) the gender and 2) the residential hall of each student. A hypothesis can be formed as: If two students share some common characteristics, such as living in the same residential hall or of the same gender, their chance of being friends are higher. Test the above hypothesis by the following steps

1. Visualize the network and color the nodes by gender and residential hall, respectively.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

1. Build 8 subgraphs of the original network according to gender and residential hall: 1 subgraph for female student, 1 subgraph for male student, 1 subgraph for students with unknown gender, and 5 subgraphs for students living in residential hall from 1501 to 1505, respectively. For example, to build a subgraph of all female students, you should keep all the nodes of female students and the edges between them. Other nodes and edges are removed.
2. Study the edge density of all the subgraph and compare them to the edge density of the original network. What is your conclusion for the hypothesis?

* Density for female friends is 0.0551786521935776
* Density for male friends is 0.0514285714285714
* Density for unknown friends is 0.1
* Density for 1501 friends is 0.12987012987013
* Density for 1502 friends is 0.0980392156862745
* Density for 1503 friends is 0.152046783625731
* Density for 1504 friends is 0.0758620689655173
* Density for 1505 friends is 0.0965909090909091
* Density for the whole network is 0.0536512667660209

Comparing the network density with all the subgraphs partly confirms the hypothesis for the study halls. However for gender this is not the case only for the unknown gender, but these are small in size.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure Unknow gender graph

## Question 5 (4 points):

To better understand the meso-scale structure, we will study community detection algorithm. One important stand of community detection algorithm is based on modularity, which tries to maximize the difference between the actual number of edges in a community and the expected number of edges in the community. However optimizing modularity in a network is NP-hard, therefore have to use heuristics.

1. Calculate the modularity of the Highschool network if community is merely identified by
   * 1. gender and
     2. residential hall, respectively.

gender modularity 4.78267523722059e-05

hall modularity 0.17559113865932

1. Search the Louvain Community Detection and explain the algorithm in your own words. The Louvain algorithm is an unsupervised community detection algorithmdivided in 2 phases: Modularity Optimization and Community Aggregation. The algorithm will start by randomly ordering all the nodes in the network in the modularity optimization phase. Then it will optimize modularity by merging communities of nodes until no significant increase in modularity is reached. After this phase all nodes belonging to the same community are merged in one big node to build a new network. In this network nodes represent communities from the previous phase and edges represent the sum of the weights of the edges between nodes in those communities. (Rita, 2020)
2. Use the Louvain Community Detection to identify communities in the Highschool network. Compare the modularity value produced by the Louvain algorithm to those in 1), and explain the reasons for the differences.

Modularity by the Louvain algorithm: 0.701644

The modularity of the Louvain algorithm is way higher compared to the networks identified by gender or residential hall. This is because the Louvain algorithm tries to create communities which improves the modularity.

**Self-exercise:** we did not cover all the network metrics in the above exercise, such as local and global clustering coefficient, diameter and component. But some of them might appear in the exam. Within your own group, explain to each other the meaning and possible application of other network metrics that are not covered in the above exercise. Check if your understanding are aligned. Reach your tutor if you have unsolved issues.

# Exercise two: Network formation models

## Question 6 (3 points):

1. Develop three networks with the same number of vertices (n), but different probability (p); Name them as ER1, ER2, and ER3. Develop the plots of ER1, ER2 and ER3, describe how these three graphs look differently as p increase and explain why.

The value for P indicates how likely a node is connected another node. So as the value of P gets higher more connections/edges appear. This also increases the density of the graph.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure ER1

Afbeelding met cravatte, accessoire

Automatisch gegenereerde beschrijving

Figure ER2

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure ER3

1. For a large n (e.g., n=1000), study the relation between clustering coefficient of the network and p, and explain the reason for such a relation. (You can use the function of transitivity (graph.object) to calculate clustering coefficient).

Transitivity: 0.2001041 and the chosen P value was 0.2.

As the probability of connection p increases, the transitivity of the network also increases. This is because as more edges are added to the network, nodes become more likely to form triangles, and therefore the transitivity increases.

## Question 7 (2 points):

Check the clustering coefficient and average path length of the Regular, SW1, SW2 and SW3. Describe the trend of clustering coefficient and average path length as p increase. Which graph does mimic the desirable attributes of a small world network?

SW1\_clustering\_coef: 0.6777076

SW1\_avg\_path\_length: 8.394292

SW2\_clustering\_coef: 0.6387643

SW2\_avg\_path\_length : 4.26903

SW3\_clustering\_coef: 0.3706598

SW3\_avg\_path\_length: 2.919309

The clustering coefficient and average path length decreases as P gets higher. Small world networks are a type of network that have both local clustering and short average path lengths between nodes. They are characterized by a few highly connected hubs that are interconnected to many less connected nodes. So in this case the SW2 graph represents the small world network the best since the average path length is way shorter than in SW1 and the clustering coefficient did not decrease that much.

## Question 8 (5 points):

You might realize not every value of p can return you a small-world network that you are looking for. Then a question arises as how can one find the range of p. In the Figure 2 of Watts and Strogatz (1998) (<https://www-nature-com.proxy.library.uu.nl/articles/30918>), it explains how can one decide the range of p by looking at the dynamics between path length and clustering coefficient.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure Figure 2: Characteristic path length L(p) and clustering coefficient C(p) for the family of randomly rewired graphs described in Fig. 1.

Here *L* is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices. The clustering coefficient *C*(*p*) is defined as follows. Suppose that a vertex *v* has *kv* neighbours; then at most *kv*(*kv* − 1)/2 edges can exist between them (this occurs when every neighbour of *v* is connected to every other neighbour of *v*). Let *Cv* denote the fraction of these allowable edges that actually exist. Define *C* as the average of *Cv* over all *v*. For friendship networks, these statistics have intuitive meanings: *L* is the average number of friendships in the shortest chain connecting two people; *Cv* reflects the extent to which friends of *v* are also friends of each other; and thus *C* measures the cliquishness of a typical friendship circle. The data shown in the figure are averages over 20 random realizations of the rewiring process described in [Fig. 1](https://www-nature-com.proxy.library.uu.nl/articles/30918#Fig1), and have been normalized by the values *L*(0), *C*(0) for a regular lattice. All the graphs have *n* = 1,000 vertices and an average degree of *k* = 10 edges per vertex. We note that a logarithmic horizontal scale has been used to resolve the rapid drop in *L*(*p*), corresponding to the onset of the small-world phenomenon. During this drop, *C*(*p*) remains almost constant at its value for the regular lattice, indicating that the transition to a small world is almost undetectable at the local level.

1. Start with a regular network of size=300, nei=6, first reproduce the Figure 2 of Watts and Strogatz (1998). Then provide the range of p which can turn this regular network (size=300, nei=6) into a small-world network.

Afbeelding met tekst, hemel

Automatisch gegenereerde beschrijving

1. Do you need to rewire significant amount of connections to make the network smallworld-like?
2. In the paper of Watts and Strogatz (1998), they pointed out that the value of p has two important implications:

“The idealized construction above reveals the key role of short cuts. It suggests that the small-world phenomenon might be common in sparse networks with many vertices, as even a tiny fraction of short cuts would suffice.” “Thus, infectious diseases are predicted to spread much more easily and quickly in a small world; the alarming and less obvious point is how few short cuts are needed to make the world small.”

Use your own words to explain these two implications. For the second implication, connect it with the spread of COVID.

## (Question 10, 2 points)

For the Highschool network, identify five edges which after deletion, there will be significant gain of the average path lengths of the network. In other words, if such five edges did not exist, the average path length of the network would increase significant. Provide your answer in the format of A-B, in which A and B are the node ID. Are they weak ties or strong ties?

Afbeelding met tekst

Automatisch gegenereerde beschrijving

By calculating the current average path length and doing this again after deleting a edge the following edges would increase the average path length the most:

"avg path length:3.69096328410784"

"deleted edge: S4 - S37"

"new\_avg\_path\_length:3.78458203495461"

"deleted edge: S24 - S49"

"new\_avg\_path\_length:3.75247256469313"

"deleted edge: S28 - S97"

"new\_avg\_path\_length:3.78187237501694"

"deleted edge: S36 - S88"

"new\_avg\_path\_length:3.73892426500474"

"deleted edge: S44 - S49"

"new\_avg\_path\_length:3.74881452377727"

S4-S37: This edge connects two nodes that have a high betweenness centrality, meaning they act as important bridges or connectors between different parts of the network. If this edge is removed, it disrupts the network's overall connectivity and increase the average path length. This edge connects two nodes that are part of different clusters or communities in the network. This edge is a weak tie.

S24-S49: This edge connects two communities that are separate from each other except for one node S44. This is the other edge S44-S49 that would increase the average path length significantly. S49 is the bridge between a small community and 2 larger ones. This is a weak tie.

S28 - S97: S97 connects to the same community as S49, but connects to a different bigger community than S24 or S44. This is also a weak tie.

S36-S88: S36 connects to two different bigger communities and is the bridge between those and 2 nodes that are otherwise separate from the whole social network. This a weak tie.

## (Question 11, 6 points):

Simulate the spread of simple contagion in the Highschool network.

stopifnot(require(data.table))

stopifnot(require(Matrix))

calculate\_value <- function(node, each\_neighbors,Pprob){

return(each\_neighbors[[node]][ which(runif(length(each\_neighbors[[node]]), 0, 1)<=Pprob)])

#'runif' is a function to generate random number in R

}

#This function:

#1) searches the neighbors of contagious node;

#2) To those who are connected to a contagious node, generates a random number and compare to the

#probability of p, if random number<p, this node will be infected and return the value of 1

IC<-function(node\_seed,network,Pprob){

#prepare input for the 'calculate\_value' function#

adj\_matrix <- igraph::as\_adjacency\_matrix(network, type = 'both')

each\_neighbors <- which(adj\_matrix > 0, arr.ind = TRUE)

each\_neighbors <- split(each\_neighbors[, 2], each\_neighbors[, 1]) #get the neigbhour list of each node

nNode<-vcount(network)

node\_status <- rep.int(0, nNode) #start from a healthy population

day\_infected<-vector()#Total number of infected population

new\_infected <- list() # Record the ID of person getting infected at each time step

day<-1

node\_status[as.numeric(node\_seed)] <- 1 # infected(value=1) health(value=0)

day\_infected[day] <- sum(node\_status )

new\_infected[[day]]<-node\_seed #The ID of the person infected in Day 1 (Patient Zero)

#simulate the spread of virus within 4 weeks##

for (day in c(2:28)){

ContagiousID<-which(node\_status == 1)

infectedID<-unlist(lapply(ContagiousID,calculate\_value,each\_neighbors,Pprob))

newinfectedID<- setdiff(infectedID, which(node\_status == 1))

#Update the node status and other variables

node\_status[newinfectedID] <- 1

day\_infected[day] <- length(newinfectedID)

new\_infected[[day]]<-newinfectedID

day=day+1

}

return(day\_infected) #return the number of newly infected people by day

#return(list(day\_infected,new\_infected)) #if you want to see the ID of infected ppl in each day,use this command instead

tel = 0

for (i in 0:99){

tel = tel + IC("5", Highschool, 0.15)

}

as.integer(tel/100)

number of newly infected people by day: 1 0 1 1 2 3 4 5 5 6 7 8 9 9 9 8 7 6 5 4 3 2 1 1 1 0 0 0

## (Question 12, 6 points):

Now you are going to test the “strength of weak ties” in the simple contagion:

1. Delete the 5 edges that you have identified in Q11 from the Highschool network and form a new network (Highschool 2); The following edges are deleted edge: S4 - S37, S24 - S49, S28 - S97, S36 - S88 and S44 - S49.
2. Delete 5 strong ties from the Highschool network and form a new network (Highschool 3); To determine the strongest ties an algorithm is used called k-core decomposition, which identifies the most densely connected subgraphs in the network. A k-core of a graph is a maximal connected subgraph in which every vertex is connected to at least k

vertices in the subgraph. This algorithm takes a range of values for k and returns the vertex set that constitutes the k-core with the highest possible value of k within the range. To obtain the k-core of a graph, the algorithm first deletes the vertices whose outdegree is less than k. It then updates the outdegree of the neighbors of the deleted vertices, and if that causes a vertex’s outdegree to fall below k, it will also delete that vertex. The algorithm repeats this operation until every vertex left in the subgraph has an outdegree of at least k. (k-Core Decomposition, 2023). When running this algorithm on the high school network it returned nodes of only one subgraph so some strong ties will be manually examined and removed.

Removed nodes with k-core decomposition S20-S112, S112-S1, S110-S52

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure S20

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure S112

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure S102

Manually removed nodes: S107-S28, S64-S58Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure S107

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure S64

1. Apply the IC models you developed in Q12 on the original Highschool network, Highschool2 and Highschool3. Record the number of newly infected people by day.

(1 0 1 1 2 3 4 4 5 6 8 8 8 9 9 8 8 6 5 4 3 2 1 1 0 0 0 0)

1. Generate a plot (with x-axis as Day, y-axis as the number of newly infected people by day) to compare the results from Step 3.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure IC model

1. Recall the “strength of weak ties” from the lecture, do the results in Step 3&4 support such a claim and why? The strength of weak ties is a concept that suggests that relationships with weak ties, can be more valuable in terms of providing opportunities and information than relationships with strong ties like close friends. This is also displayed in the graph by looking at the difference between the graphs highschool and highschool2. The weak ties act as bridges between different subgraphs, increasing the likelihood more people get infected every day.

When comparing the network highschool2 and highschool3 there are not many differences after removing strong ties. Strong ties won’t increase the amount of new people to be infected that much since losing one connection of many does not prevent the person to be infected. An example is a social group of 5 friends, if one person cuts of his relationship with one of the friends he can still infect that person through the other 3 friends.

## Question 13 (8 points):

In the above exercises, the “strength of weak ties” are tested in a simplified IC model with a specific probability p. Do you think your observation in Q13 holds regardless of the contagiousness of the virus? To find out,

* 1. Play around the probability p in the IC model. Change the value of p to high and low ends, run the IC model again on Highschool, Highschool 2 and Highschool 3, and see if you will observe different things (2 points). The same observation made in Q12 is seen in both graphs, however the graphs have a higher peak if the disease is less infectious and vice versa for a more infectious disease. Also the peak amount of people being infected happens earlier if the disease is more infectious and later if the disease is not that infectious.

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure IC model with P of 0.9

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure IC model with P value of 0.1

* 1. The above IC model is a simplified version of the SIR model. In the SIR model, node have three status: S, I, or R, (Susceptible, Infectious, or Recovered). Modify the IC model to a SIR model with the following characteristics:
     + - Each node in the network has three statuses: Susceptible, Infectious, or Recovered.
       - At Day 0, all the nodes in the network are Susceptible;
       - At Day 1, an infectious node (N0, node ID= S5) is introduced to the network;
       - At the following days, all the nodes connecting to the infectious node will have a chance of 0.15 (p=0.15) being infected.
       - Every infected node will remain infectious for 3 days, i.e., only the infected nodes activated from the past 3 days can transmit the virus to their neighbours. After that, their status becomes Recovered, which cannot be either Infectious or Susceptible again.
       - Model the contagion process for 4 weeks.

## (Question 14, 5 points):

An arbitrary assumption about the thresholds of each node in the Highschool network has been made, which can be found in the “Highschool\_network\_att.csv”. Build a threshold model according to the above model description and the predefined thresholds of each node, answer the following questions

1. By seeding 5 nodes (ID=59,63,91,92,99), how many people in the network can be activated? In total 52 people can be activated.

threshold\_model <- function(network, node\_seed, threshold, n\_day){

nNode <- vcount(network)

node\_status <- rep.int(0, nNode) # start from a healthy population

adj\_matrix <- igraph::as\_adjacency\_matrix(network, type = 'both')

each\_neighbors <- which(adj\_matrix > 0, arr.ind = TRUE)

each\_neighbors <- split(each\_neighbors[, 2], each\_neighbors[, 1]) # get the neighbour list of each node

day <- 1

for (seed in node\_seed){

node\_status[as.numeric(node\_seed)] <- 1 # adopt(value=1) dont adopt(value=0)

}

for (day in 2:n\_day) {

for (node in 1:nNode) {

if (node\_status[node] == 0) {

neighbours <- each\_neighbors[[node]]

n\_neighbors <- length(neighbours)

n\_adopters <- sum(node\_status[neighbours] == 1)

if(n\_adopters > n\_neighbors \* threshold[node]){

node\_status[node] <- 1

}

}

}

}

return(sum(node\_status==1))

}

threshold\_model(Highschool, list('59', '63', '91', '92', '99'), highschool\_att$Threshold, 10000)

52

1. Use the “width of a bridge” from the lecture to explain why the contagion fails to reach the following two communities:
   1. the one consisted of Node 55, 107, 93, 109, 80, 28;
   2. the one consisted of Node 110, 39, 10, 1, 50, 106.

In a social network, a bridge is a tie between two nodes (or individuals) that belong to different clusters or communities. The width of a bridge between two communities is defined as the number of overlapping ties between them. So a bridge with a low width has few ties connecting them. The contagion fails for these two communities because the majority of the neighbors needs to be adopting the campaign. This is not possible if the connections to these communities are connect by small bridges with weak ties.

## Question 15 (6 points) :

Apply the empirical distribution of the threshold of students in this class to the Highschool network, and answer the following questions:

1. How are you going to do it? Explain your method into steps. I used the distribution in the csv file, but if I would do it myself I would create a distribution that follows a sigmoid function and put more values on the lower and higher end of the sigmoid function. It would be similar to figure 14, but with more of a curve.   
   Afbeelding met grafiek

   Automatisch gegenereerde beschrijving

Figure Empirical distribution

Afbeelding met grafiek

Automatisch gegenereerde beschrijving

Figure Distrubtion of the original threshold

After achieving this distribution it needs to be assigned to the different students. I did this by randomizing the order of values in the distribution and assigned that list to the thresholds of students.

1. What are the limitations of your method? What procedures are you going to take to address such limitations? This distribution is still based on a sample and not the original values. Another limitation is how the new thresholds get assigned to the students. I randomize the list of thresholds and assign it to each student, but since there are a lot of ones in the distribution it can occur that the ties with the starting students have a lot of thresholds of 1 which makes it harder to spread the ‘’once a beef” campaign. This can be resolved by running the model multiple times with different orders of the distribution assigned to every student, but still won’t resolve the issue fully. Also the same can be said about there being a lot of zeroes in the new thresholds. A lot of students can join the campaign if the correct students have a low threshold which will also not be representative of reality.

Another limitation is that a lot of thresholds are the same which is not representative to reality. I would resolve this by creating a distribution that follows a sigmoid function and put more values on the lower and higher end of the sigmoid function. This would make it so the influence of the campaign will process a bit more smoothly.

Another limitation is that a threshold of 1 will almost never be reached which is unrealistic. I would suggest reducing the amount of people of having a threshold of 1 and put it between 0.9 and 1. This would make the spread of the campaign also more realistic since a person is the average of the five people he/she spends the most time with.

1. After you apply the empirical threshold distribution to the Highschool network, by using Node ID=59,63,91,92,99 as seeds, how many people in the network can be activated?

This depends fully on how the new distribution is assigned, but if I would run the model a 1000 times with different assigned thresholds it would be an average of: 13

# Bibliografie

Rita, L. (2020, April 9). *Louvain Algorithm.* Opgehaald van towardsdatascience: https://towardsdatascience.com/louvain-algorithm-93fde589f58c