Network Analysis
Final Report

Youhee Kil (r0768512)

Summer 2020

Contents

1 Introduction

Social Network Analysis (SNA) are used for modelling interaction in the classroom. In this report, we examined the evolution of a friendship and trust relationship network in one of classes of a secondary school with respect to their 'drink' behavior. Bokhove (2018) argued that SNA might be the most useful candidate methodology to explore classroom interaction. Normally students choose who to hang out or be friends with based on their similarity. Relationships with peers are dependent on the interaction, and it leads to the construction of the trust relationship between them. As Huang, Soto, Fujimoto, and Valente (2014) stated "adolescents are likely to form friendship around similarities in established risk behaviors". Therefore, we investigate whether the 'gender' and 'drink' affects the effects of friendship selection and influence as well as across developmental stages. Besides, network can be estimated based on cross-sectional or longitudinal time-series data (Hevey (2018)). In this case, a model based on random actors has been also implemented to examine changes in behavior as friendship and trust relationships change over time.

Veenstra, Dijkstra, Steglich, and Van Zalk (2013) presented that "Selection" and "Influence" are the two fundamental processes underlying network-behavior dynamics to understand adolescents' development. The contents of the paper is organized as follows. Section 2 briefly describes about the data and discusses related research questions. Section 3 presents the methods and results, (a) Selection and Influence Procedure and (b) QAP and community Detection. Section 4 introduces Stochastic Actor-Based Model, (a) Hypothesis and Methods, (b) Results, and (c) Goodness-of-fit, to examine how networks affect each individual student's behavior or how network structure changes over time. Lastly, short conclusion is followed.

2 Data & Research Questions

2.1 Data

We analyze the co-evolution of a friendship and trust relationship in a secondary school. The data were collected from students which friendship and trust networks as well as drink data were assessed over time. There were 33 students in the classroom, including 12 girls and 21 boys. Students were asked to assess the network data which they considered good friends and reliable friends. The average numbers of nominated 'good friend' is each 4.7 and 4.2 over the two waves, indicating a small decrease over time. Additionally, the average numbers of nominated 'reliable friend' are 2.3 and 1.9, showing a little decrease over time. However, the variance of the in-degree is different as the variance of the out-degrees. The variances of in-degree and out-degree measure how unequal the actors in a network Wasserman, Faust, et al. (1994). For example, as shown in Table 2, there is a significant difference in the degree of variance between friendship w1 and w2 indicating that friendship w1 has a higher degree of inequality between actors with receiving ties.

Name	Mean Indegree	Mean Outdegree	Variance Indegree	Variance outdegree
Friendship w1	4.6667	4.6667	7.5417	18.0417
Friendship w2	4.2424	4.2424	4.8769	20.6269
Trust w1	2.3333	2.3333	3.4167	7.6667
Trust w2	1.8788	1.8788	2.9849	2.9849

Table 1: Mean and Variance of In-degree and Out-degree

Some data were missing on account of pupils' absent at the each moment of data collection. A total of 9 students were absent, accounting for 27% of the total number of courses for friendship w1 (wave 1). A total of 6 students were absent, accounting for 19% of the total class size of friendship w2 (wave 2). Values in drink data range between 1 and 3, and a higher number means a higher frequency of alcohol consumption, there is the same number of missing values as friendship w1(wave 1) and w2 (wave 2). Drink data was categorized to four levels, (0) Absent, (1) Non Drinking, (2) Frequently Drinking, (3) Usually Drinking. Therefore, close to 20% of the total number of students were missing in this case which is problematic and might lead to an erroneous analysis of the data.

Name	Absent	Missing Proportion	Density
Friendship Wave 1	9	0.2727	0.2005
Friendship Wave 2	6	0.1884	0.1634
Trust Wave 1	9	0.2727	0.1002
Trust Wave 2	5	0.1515	0.0692

Table 2: Details of Data of the Friendship and Trust network

2.2 Research Questions

Acknowledge the importance of friendship is crucial for understanding behavior development in adolescence. In this paper, we focused on relations between peers. To understand how networks (friendship and trust relation) are shaped in the classroom, we examined network dynamics, peer selection and influence process, gender, and drinking behaviors, and detecting subgroup and community of the networks. Furthermore, we proceeded with the Stochastic Actor-Based Model to investigate more about selection and influence processes over time.

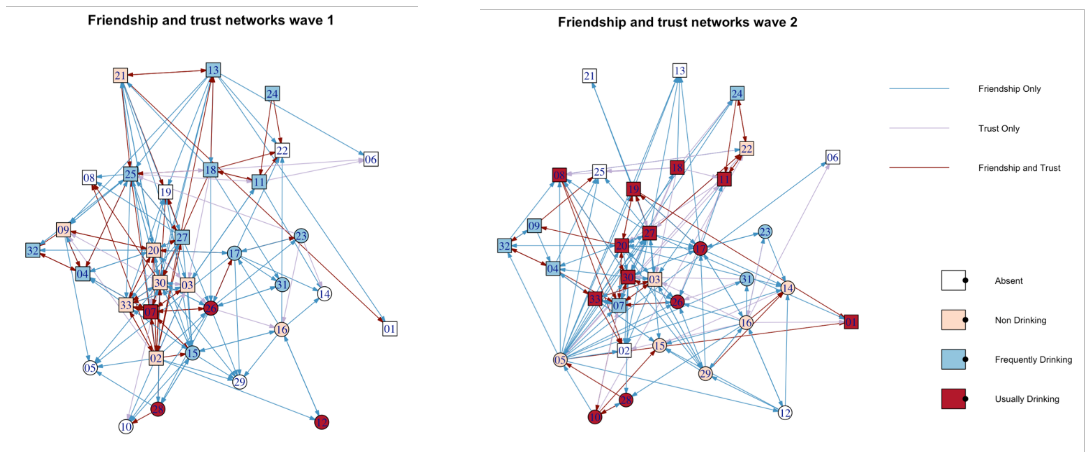
Based on the data and along with those procedures, following research questions are stated: (1) Does popularity with classmates result in trust relationship and drink behavior? or vice versa (2) Same gender will be more likely to form a mutual relationship (reciprocated dyads) compared to different gender? (3) Drinking behavior affects forming friendship and trust relationship? (4) Drinking behavior affects the peers' network and trust relationship over time?

3 The methods and results

3.1 Selection & Influence Procedure

In general, people would like to be friends with people who have the same gender, similar behavior, and characteristics.In addition, these features will affect the trust between friends. As Veenstra et al. (2013) stated the procedure of adolescents choosing whom to be friends with is "Selection Procedure" which is based on similarity. In this case, we assume 'gender' and 'drink' would be important features for selecting friends and forming a trust relation. In Figure 1, we could assume that students prefer friendship with same-sex alters and same drinking behaviors. In addition, students trust same-sex alters. In this case, along with the assumption and research questions, the hypothesis is mutual relationship will be more likely to be the same gender or same drinking behaviors.

The proportions of mutual friendship w1 and w2 are 0.4872 and 0.3488 respectively, indicating that the friendship w1 is higher than that of friendship w2. Similarly, trust w1 (wave 1) has a higher mutual trust relationship than trust w2 (wave 2), the proportions of mutual trust relationship for w1 and w2



(a) Wave 1 (b) Wave 2

Figure 1: Friendship and trust relationship network with frequencies of drinking
(Boy: square, Girl: Circle)

are 0.2857 and 0.2174. When we compare the proportion of mutual relationships between friendship and trust, trust waves have lower values which suggest that trustiness doesn't come as a friendship in this case. For example, peers might not trust their friends even though they are friends with each other. Based on the result of dyads, we can assume that, in this case, the number of mutual relations of friendship and trust is reduced from w1 and w2. However, it is difficult to draw any conclusions from mutual relationships, because the relationship in the class may rarely have really popular and active students. Social Network may be established in the center of these students.

In the social network, degrees, closeness, and betweenness can be of great interest to check centrality. In our case, the actor with higher out-degree is a more active person and nominates many others as friends. In addition, actors with higher in-degrees are more popular, and many others nominate him as a friend. Based on the results, Peer [30] was the most popular friend in wave 1 (in-degree: 30(w1)), but Peer [7] (in-degree: 9(w2)) became the most popular person in wave 2. In addition, Peer [7] was the most nominated one as a good friend and reliable person in the class both in wave 1 and wave 2 (Figure 1). Regarding out-degree, an actor with the largest number of out-degrees in friendship w1 and trust w1 was peer [2] which indicates peer [2] is most active student in the class in wave 1. Intimacy and intermediateness are other methods to checking centrality. Closeness focuses on how close each node is to every other node in a network. In addition, by checking betweenness, it is possible to know which peer is in a position to control the flow of information in the network. Based on the results of closeness and betweenness, In this classroom, the friendship is centered by peer [30], peer [20], peer [4], peer [27], and trust relationship is centered by peer [7], peer [25], peer [2]. Apparently, central peers are different in friendship networks and trust networks. By looking at the result, we can hypothesize there is a different relationship formed between friendship network and trust network. People who want to be friends are different from people who trust. Therefore, if those peers are missing in either wave 1 or wave 2, it may affect the analysis of the network. However, Peer [2] and Peer [25] are absent in Wave 2, these kinds of absence would lead to making huge differences between w1 and w2 since there are lots of absent in this data. Since our networks w1 and w2 are the same set of arcs, the degree and mean of the two networks are equal. Lastly, one thing we should focus is students who are in the center of the class in terms of friendship and trust relationship enjoy drinking in wave 2. As we mentioned earlier, students who are at the center in the class - peer [30], peer [20], peer [4], peer [27], peer [7], peer [25] - enjoy drinking ("usually-drinking" and "frequently-drinking") in wave 2 and all of them are male (Table 3). At the same time, Peer [2] and Peer [25] are absent at wave 2, this result may affect the analysis of the trust relationship in w2 . As seen in Figure 1, apparently wave 2 has more students who are categorized as "usually-drinking", and there are more relationship combining Friendship and Trust in wave 2. Noticeably, drinkers are more popular in wave 1 and wave 2.

Peer	Gender	Wave 1	Wave 2
30	male	Non-Drinking	Usually-Drinking
20	male	Non-Drinking	Usually-Drinking
4	male	Frequently-Drinking	Frequently-Drinking
27	male	Frequently-Drinking	Usually-Drinking
7	male	Usually-Drinking	Frequently-Drinking
25	male	Frequently-Drinking	**Absent**
2	male	Non-Drinking	**Absent**

Table 3: Popular Students drinking behaviors

		Girl	Boy
Friendship W1	**Girl**	1.53	0.11
	Boy	0.24	1.46
Friendship W2	**Girl**	1.55	0.25
	Boy	0.22	1.39
Trust Relationship W1	**Girl**	0.68	0.11
	Boy	0.10	1.90
Trust relationship W2	**Girl**	0.68	0.11
	Boy	0.10	1.90

Table 4: Percentage of each gender in each network

3.1.1 gender-based homophily

As we hypothesized earlier, girls tend to stick together slightly more than boys in both w1 and w2. Interestingly (Table 4), there is a slightly higher percentage for girls to hang out with girls in w2 compared to w1. In w1, percentage of 0.24 of boys think girls as a friend and 0.11 of girls think boys as a friend. Slightly fewer boys think girls as a friend in w2, however, a percentage of 0.25 of girls think boys as a friend in w2 which is a drastic increase compared to w1. We want to see how the trust network is separated by gender. There is a high percentage of trust between the same genders, a percentage of 0.68 for girls, and 1.90 for boys (Table 4). However, the trust between the boy and the girl is very low, and the trust of each relationship is about 0.1. Consequently, w1 tends to have a higher value of the assortative coefficient which measures the level of homophily of the graph. An interpretation of positive value of the assortative coefficient is that connected nodes have higher tendency to share similar properties. On the other hand, negative value of assortative coefficient means that the connected nodes tend to have different properties. As seen in Table 5, value of assortative coefficient of friendship is 0.49 at w1, and the highest value of assortative coefficient of friendship of trust is 0.45 at w1 as well. In total, we can confidently state that there is a high degree of gender segregation in both relationships. Girls tend to stick together, at the same time, boys tend to stick together and this trend for boys is captured in trust relationships more than girls. Besides, based on the result of assortative coefficient, both gender has moderately positive values. Therefore, we conclude that the students in the class social network show gender-based homophily. In other words, the students who are friends with each other tend to come from the same gender.

	Gender	Drink (three levels)	Drink (two levels)
Friendship W1	0.49	-0.01	0.10
Friendship W2	0.41	0.04	0.08
Trust W1	0.45	-0.01	0.10
Trust W2	0.38	0.10	0.21

Table 5: Assortativity coefficient to understand the level of homophily in the relationship of friendship and trust with gender and drinking behavior

3.1.2 Drinking behavior-based homophily

In addition, as we stated students who have the same drinking behavior will be more likely to have mutual friends, so that understanding the level of homophily in the relationship of friendship and trust relationship with the drinking network is necessary. Assortativity Coefficient allows us to test the association or measure the correlation between the characteristics of every pair of nodes that are connected. In Table 5, the result values for friendship w1 and w2 are -0.01 and 0.04 with drink w1 and drink w2, which is very close to 0 indicating no strong association of the property values between connected nodes. Similar to the trust relation, the values for w1 and w2 are 0.01 and 0.10.

Furthermore, we also tested assortative coefficient with drink data containing two levels (non-drinking, drinking) instead of three levels (non-drinking, frequently-drinking, usually-drinking) by combing frequently-drinking and usually drinking into "drinking". In table 5, the result values are slightly increased at each network. Similarly, Trsut W2 has the highest value of assortative coefficient (0.24) which indicates students who trust each other moderately tend to come from drinking behavior. In short, there is no close connection between friendship, trust, and drinking, but it is clear that the level of isomorphism increases to w2 with drinking behavior in each relationship between friendship and trust. Since it's not significantly increased, it's hard to conclude that students who have the same drinking behavior will be more likely to have mutual friends. But still, we can state that this drinking behavior can establish stronger mutual friendship and mutual trust relationship even with small degrees.

3.2 QAP and community detection

3.2.1 QAP

We expect the correlated relationship between each wave (w1 and w2). The further investigation of hypotheses that there is correlation between friendship w1 and w2, as well as, the relationship between trust w1 and w2, Hamming ratio, Jaccard distance, and QAP are used. Friendships w1 and w2 are very similar. The similarity is tested according to the Hamming ratio between the two friendship networks (0.9112), which is a super-stable friendship network. Trust network is also a super-stable network like friendship network (0.9460). Similar to the hamming distance, the Jaccard distance measures the difference between the two groups. The dissimilarity of two groups of two networks is 0.6039 and 0.7307. But sometimes the lack of values in the data set can make calculating similarity / dissimilarity challenging. Then, after filling the blank area with zeros to recalculate the difference, the difference value of the friendship and trust network of Jaccard distance increases to 0.7382 and 0.8264. Friendship w1 and w2 are moderately correlated (r=0.4802) and trust w1 and w2 are moderately correlated (0.4475) as well. Figure 1 presents overall network of friendship and trust relationship among students in the class room. It is clear to see which peer has a relationship with whom and how it changed as well as who is absent. QAP is used to analyze the dyadic data set which is a type of social network where there are two individuals that are linked. According to the QAP test results, we could conclude that there is a correlation between friendship w1 and w2, as well as, there is a relationship between trust w1 and w2. Also, there is a correlation between friendship w1 and w2 on the same gender and between trust w1 and w2 on the same gender.

3.2.2 Triad Census

"Many important theories about social relations can be tested by means of hypotheses about triad census in a social network" (Holland Leinhardt, 1977). Friendship w2 has more empty relations and asymmetric relations than friendship w1. We could assume that there are not as many circles as friendship w1. As a result of counting triads, friendship w2 have considerably more counts for '003'. '012', '021D', '021U', '021C', '111U', '030T'. In contrast, friendship w1 has more counts for '102'(slightly), '111D', '201', '120D', '210', '300' (Figure 3). The meaning of having '003', '012', '021U', '021C' triads have only asymmetric and null dyads, and only '111U' has 1 mutual. Therefore, compared with friendship w1, friendship w2 has more asymmetric and invalid binary relations. Trust w1 and w2 have only '003', '102', '201', '300' triads, and trust w2 has more of those triads except '300' (Figure 4). On the other hand, in the triad's friendship distribution, w1 and w2 in the random graph are very similar to the number of triads observed in the network. It is possible to compare the characteristics of our network with random graphs regarding size, density, and distribution of triads. Distribution of triads for Friendship relations in random graphs is more likely similar to the observed friendship networks except '030T', '030C', and '210'. There are more '030T' and less '030C', '210' triads captured in the random graph compared to the observed friendship networks. In short, we conclude that people, in this case, form stronger mutual trustiness between friends who have trusted the friend from w1, but, there are less complete triads ('300') which show a stable and strong shared norm 'trustiness' in the network. In addition, based on the result of triad census, the observed friendship network usually has more formed friendships based on active students.

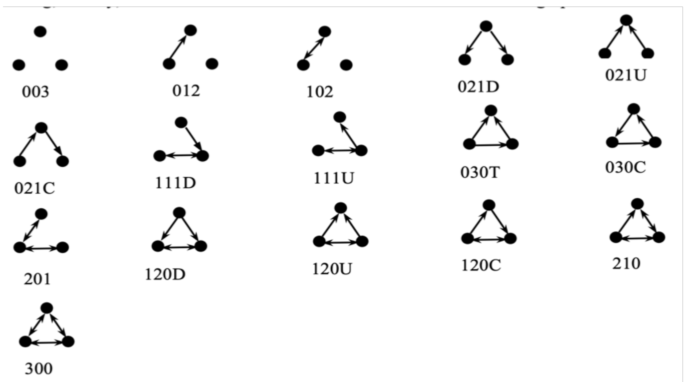
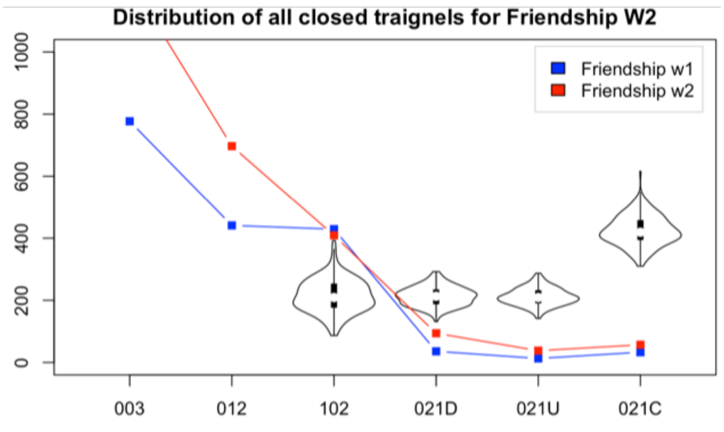


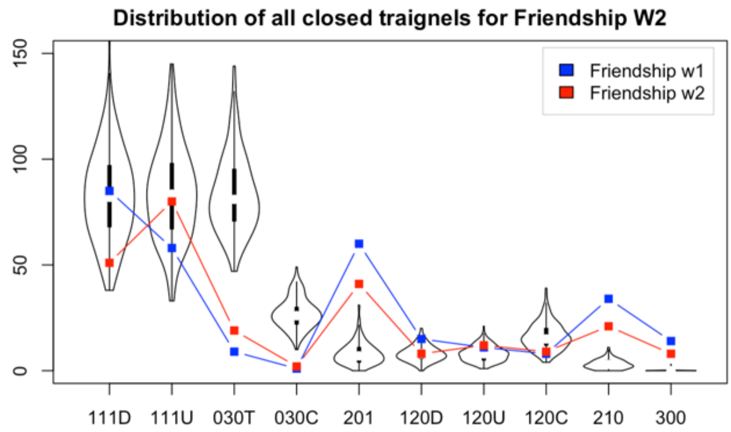
Figure 2: Triad Census

3.2.3 Subgroup

The number of subgroups of friendship w1 and w2 is the same, however, in the trust network, w2 has more subgroups compared to w1. The friendship network has 11 subgroups, and the trust networks w1 and w2 have 15 and 18 subgroups, respectively. It takes 5 to 7 steps to connect the two nodes that are situated the furthest apart in the friendship and trust network which suggests that these networks are not very compact. The largest groups in friendship w1 and w2 are 6 and 4. But k-core is easier to identify groups in the network compared to cliques because it is nested and doesn't overlap with other groups. Figure 5 shows that the center of the network is 6, which is the highest k-core. In this



(a)



(b)

Figure 3: Plot of Triad Distribution at Friendship W2 with Triad Counts at each waves

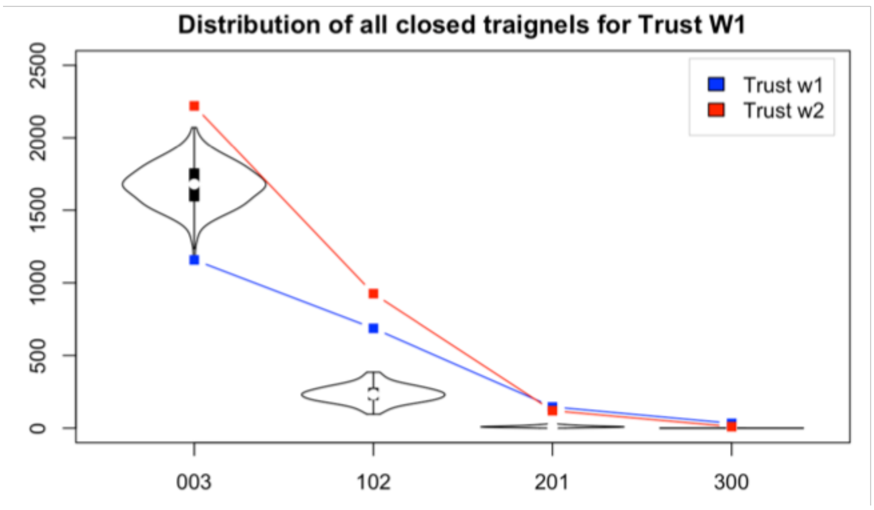


Figure 4: Plot of Triad Distribution at Trust W1 with Triad Counts at each waves

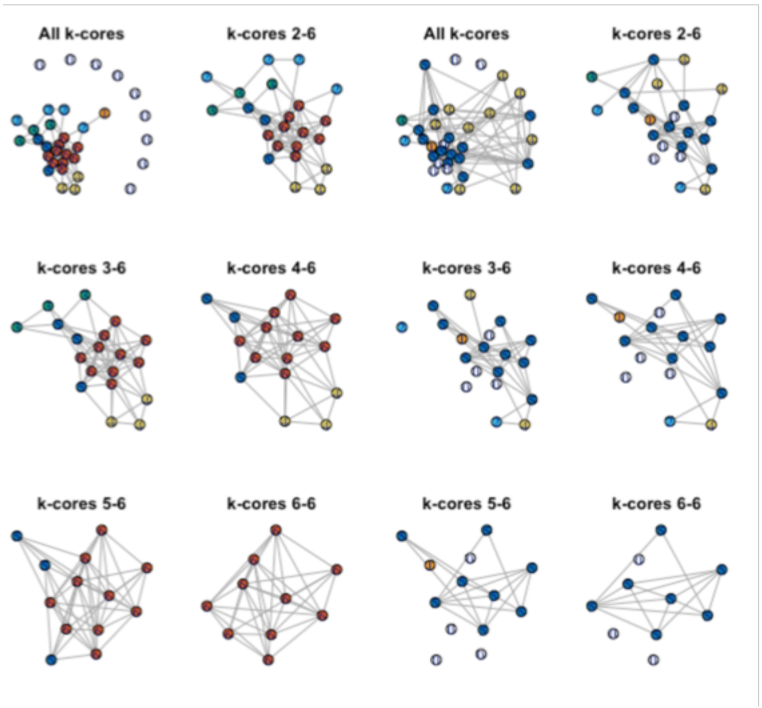


Figure 5: Different numbers of k-cores in Friendship W1 (left) and Friendship W2 (right)

case, the 6-core is composed of 12 of the 33 total nodes. To examine the pattern of the subgroups we used 'peeling away' progressively each of the lower k-cores in turn. Then we can see how its subgroups change. Obviously, each subgroup can be identified by a different number of k-cores and shows how the subgroups in friendship w1 and w2 are different, especially the influx of absent students in w1 may cause a difference in w2. We can interpret the number of subgroups and Figure 3, for friendship, the number of subgroups maintains in w2 and the core cliques are remained pretty much well in w2 by looking at Figure 3. But, the number of subgroups in trust relationships differs between w1 and w2. As we mentioned earlier, peer [25] and peer [2] were absent in w2. Maybe this is the reason why Trust w1 and w2 show a big difference, the range of k-cores in trust w1 is 1 to 5, but the highest k-core decreased to 3 in trust w2 (Figure 4). Besides, trust w1 has a total 11 of 0 k-core, which is not surprising, because most people cannot really trust others at the early stage in the classroom, which would be the reason why there are so many 0 k-cores in trust w1. For this reason, we assume that the k-core between trust w1 and trust w2 is very different. In total, finding subgroup and community is earlier in the Friendship network but in Trust network. We can interpret this circumstance as missing central peers such as peer [25], peer [2] in Trust leads to having different subgroups of the whole network. For friendship networks, since there is no missing popular and active peer so that there are no big changes within subgroups unlike trust relationship, but they're still some differences subgroups formed as time passed to w2.

3.2.4 Community Detection

The community detection algorithm aims to find densely connected areas in the network. But it is relatively loosely connected to other areas. Hence, it is easier to determine which student all of these community detection algorithms can be well detected in w1 (Figure 6). However, to determine the community in friendship w2, this detection becomes very complicated. As we've seen in Figure 7, the community in w1 has changed in w2, which may be due to gender segregation, drinking behavior, and the influx of absent peers in w1. The popular and active students in the class are listed in table 3. As Figure 6 and Figure 7 presented, those students (peer [30], [20], [4], [27], [7], [25], [2]) are mostly in the same group in friendship network (e.g. highest core in k-cores subgroup). Besides, the community group became more mixed-gender in w2 compared to w1 which slightly confirmed the result of the percentage of each gender in each network (Table 4). Additionally, the drinking behavior doesn't affect much for forming community based on the result of blockmodel and fast'n'greedy. But, still as we've confirmed the popular and active students enjoy drinking and those students formed biggest group in the classroom based on the k-core method. Therefore, We would like to investigate further why and how the community has changed in w2 more precisely based on the Stochastic Actor-Based Model. (Community detection of Trust Network W1 and W2 can be found at the Appendix Section (Figure 11, Figure 12))

4 Stochastic Actor-Based Model

4.1 Hypothesis & Method

How networks affect an individual's behavior such as peer influence, how network structure changes over time, and what are the endogenous associations between networks and behaviors. A Stochastic Actor-Based Model is used to answer how networks affect each individual peer's behavior or how network structure changes over time. Since ERGMs are generally limited to cross-sectional network data. But in this case, a model based on stochastic actor-based can be used to build models and test the last hypotheses - Drinking behavior affects the peers' networks - friendship and trust relationship over time - in the network. Huang et al. (2014) listed the assumption of Stochastic Actor-Based model distinguishing from model we used conventionally to estimate individual-level changes within the network. The following list is referenced by the paper "The Interplay of Friendship Networks and Social Networking Sites: Longitudinal Analysis of Selection and Influence Effects on Adolescent Smoking and Alcohol Use" (Huang et al. (2014)) (1) "Changes between measurement points are modeled according to a continuous-time Markov process to simulate likely unobserved developmental trajectories between the measurement time points.", (2) Each actor is assumed to independently make decisions about changes in friendship ties, trust relationship ties or behaviors, without conspiring with others about these decisions, (3) We tested the study aims as we specified a drinking behavior model to simultaneously estimate friend selection and influence and trust relationship of adolescent alcohol use.

The Jaccard index is a measure of similarity, the author of RSiena suggests that Jaccard values should be higher than 0.3 (T. Snijders (n.d.)). In this case, Jaccard values for friendship w1 and w2, values for friendship w1 and trust w1 are higher than 0.3 (0.3961, 0.3508) which indicates its stable changes over time. However, the Jaccard values for trust w1 and trust w2, values for friendship w1, and trust w2 are lower than 0.3 (0.2692, 0.1984). Values lower than 0.2 indicate that there might be difficulties in the estimation, but the reason why Jaccard value is low in this case, the network is mainly decreasing, then it wouldn't be a problem for the SIENA method (T. Snijders (n.d.)).

Sixteen different hypotheses will be tested with the sixteen effects listed in the table below (Table 6). As a categorize these effects on Gender Homophily effects, Drinking Homophily, Ego effects, Alter effects, Cross-production effects, and Reciprocity effects. First, Gender homophily effect, there appears to be a strong gender homophily effect, where students are much more likely to be friends with other students of the same gender, besides, students are much more likely to trust with other friends of the same gender. We hypothesize that the likelihood of an ego forming a new friendship tie is higher with an alter who has the same gender. Similarly, we hypothesize that the likelihood of an ego forming a new trust tie is higher with an alter who has the same gender. Secondly, the effect of drinking homogeneity is the same as the gender homogeneity, we assume that the likelihood of an ego forming a friendship and trust tie is higher with an alter who has the same drinking preference behavior. Third, for Ego /Alter gender effects on Friendship and Trust networks, we hypothesized that he or she will be more or less likely to

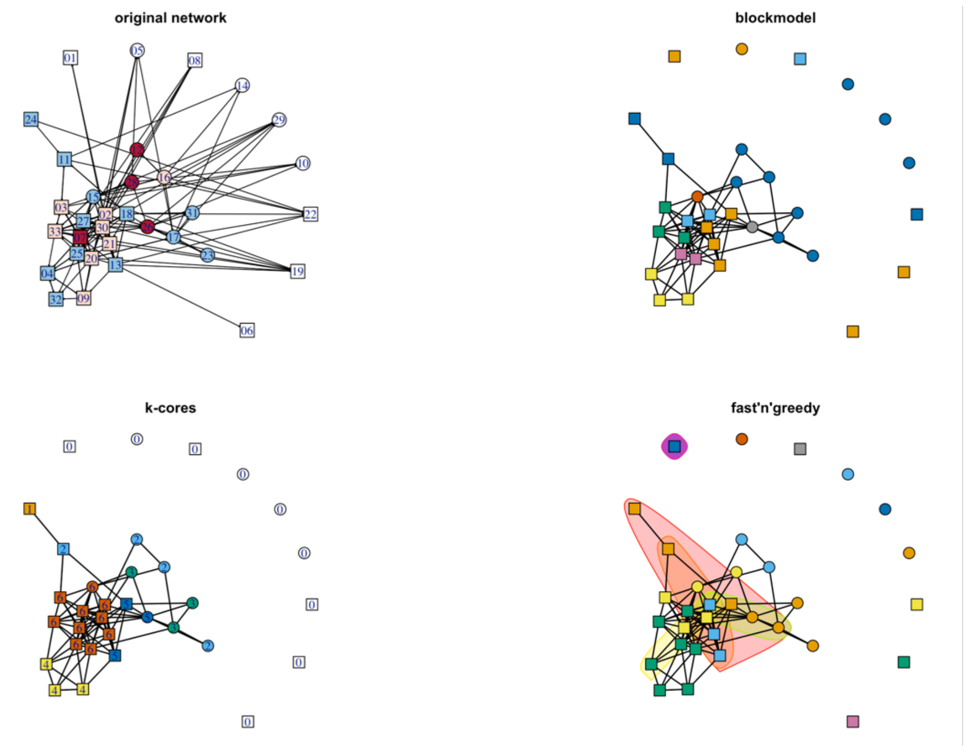


Figure 6: Community Detection Friendship w1

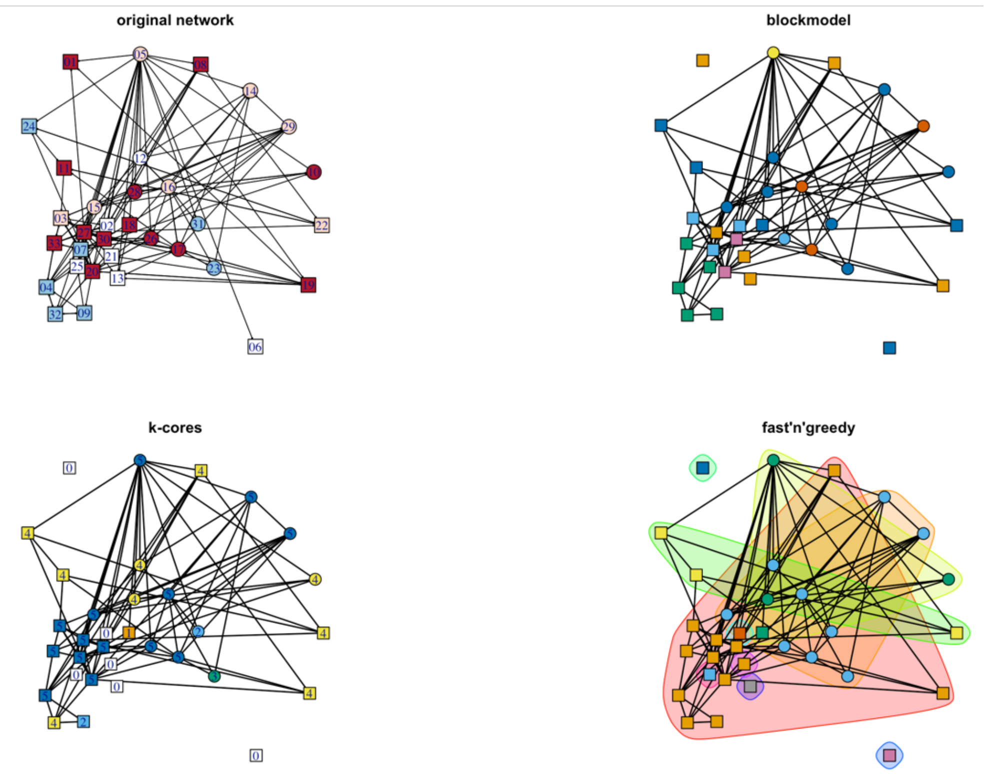


Figure 7: Community Detection Friendship w2

form a friendship and trust tie (egoX) based on the ego's gender, conversely, the likelihood of forming a friendship and trust might be related to the gender of altX (altX). Fourth, we hypothesize based on the drinking behavior of ego and alter, he or she is more or less form friendship and trust (egoX). On the contrary, the possibility of establishing friendship and trust is related to drinking behavior of altX (altX). In the latter case, a peer might want to be friends with someone because they drink, regardless of peer's drinking preference behavior. Lastly, we are interested in mutuality and reciprocity effects. The trend of reciprocity of friendship and trust relationships (recip) and the basic dyadic cross-production effect of friendship and trust are modeled as final effects.

4.2 Result

In the basic model, Outdegree, the reciprocity of friendship and trust networks, and the quadratic shape of drinking behavior have a small amount of default effects. The RSiena manual suggests that "absolute values less than 0.10 indicate excellent convergence, and absolute values less than 0.15 are reasonable". In this model, all of the network dynamics parameters have an excellent convergence with a maximum convergence ratio of 0.149. According to the table 7 , the basic rate parameter 'trust.dependent2' was only the one that appeared as significant which means average number of opportunities for changes in trust relationship over time is significant. Even though all hypothesized effects weren't significant in the result, coefficients of all effects are positive. For our example, we can see that friendship network formation is more likely with egos who have the same drinking behavior as an alter or the same gender as an altar, but those have large standard errors, so analysis would not be precise. The result shows that drinking behavior is in linear shape which means drinking behavior is increasing over time instead of quadratic shape, but it is not non-significant.

Effect	Type	ED name	ED shortName	ED Interaction1
1 - Gender Homophily	Selection	Friendship	sameX	gender
2 - Ego gender Effect	Selection	Friendship	egoX	gender
3 - Alter gender Effect	Selection	Friendship	altX	gender
4 - Gender Homophily	Selection	Trust	sameX	gender
5 - Ego gender Effect	Selection	Trust	egoX	gender
6 - Alter gender Effect	Selection	Trust	altX	gender
7 - Ego drinking Effect	Selection	Friendship	egoX	drinking
8 - Alter drinking Effect	Selection	Friendship	altX	drinking
9 - Drinking Homophily	Selection	Friendship	sameX	drinking
10- Ego drinking Effect	Selection	Trust	egoX	drinking
11- Alter drinking Effect	Selection	Trust	altX	drinking
12- Drinking Homophily	Selection	Trust	sameX	drinking
13- Cross-Production effect	Structural	Friendship	crprod	Trust
14- Cross-Production effect	Structural	Trust	crprod	Friendship
15- Reciprocity	Structural	Friendship	recip	NA
16- Reciprocity	Structural	Trust	recip	NA

Table 6: Hypothesized Effects on Stochastic Actor-Based Model

4.3 Goodness-of-Fit

Goodness-of-Fit allows us to assess the extent to which the model can produce simulated networks that 'look like' the observed network (Kolaczyk Csardi, 2014). T. A. Snijders (2020) described that

Table 7: Results of Effects on Stochastic Actor-Based Model

Function	**Effect**	**Coef**	**StEr**	**Conv**
friendship.dependent1	basic rate parameter friendship.dependent1	9.142	12.840	0.002
friendship.dependent1	friendship.dependent1: outdegree (density)	2.633	4.282	0.001
friendship.dependent1	friendship.dependent1: reciprocity	2.745	6.373	0.003
friendship.dependent1	friendship.dependent1: gender.coCovar alter	1.342	3.824	0.021
friendship.dependent1	friendship.dependent1: gender.coCovar ego	0.818	3.228	0.015
friendship.dependent1	friendship.dependent1: same gender.coCovar	0.956	2.842	0.012
friendship.dependent1	friendship.dependent1: drinking alter	0.498	3.109	0.015
friendship.dependent1	friendship.dependent1: drinking ego	0.953	1.228	0.020
friendship.dependent1	friendship.dependent1: same drinking	2.173	11.501	0.041
friendship.dependent1	friendship.dependent1: trust.dependent2	5.836	25.419	0.003
trust.dependent2	**basic rate parameter trust.dependent2**	**7.576**	**3.332**	**0.062**
trust.dependent2	trust.dependent2: outdegree (density)	3.109	3.852	0.013
trust.dependent2	trust.dependent2: reciprocity	0.093	0.600	0.002
trust.dependent2	trust.dependent2: gender.coCovar alter	0.978	0.634	0.025
trust.dependent2	trust.dependent2: gender.coCovar ego	0.468	2.233	0.034
trust.dependent2	trust.dependent2: same gender.coCovar	0.106	0.576	0.010
trust.dependent2	trust.dependent2: friendship.dependent1	2.503	4.623	0.030
drinking	rate drinking period 1	2.226	1.977	0.069
drinking	drinking linear shape	0.674	1.557	0.019
drinking	drinking quadratic shape	0.173	1.052	0.012

"The fit is good if the average values of the auxiliary statistics over many simulation runs are close to the values observed in the data" (T. A. Snijders (2020). The p-value is calculated by a Monte Carlo test based on the Maholanobis distance where the larger p-values indicate better model. In this case, our model does a good job of producing simulated networks that have the similar indegree distribution as the observed network for friendship.dependent and trust.dependent. In addition, for trust.dependent, it will also produce almost same in-degree distribution.

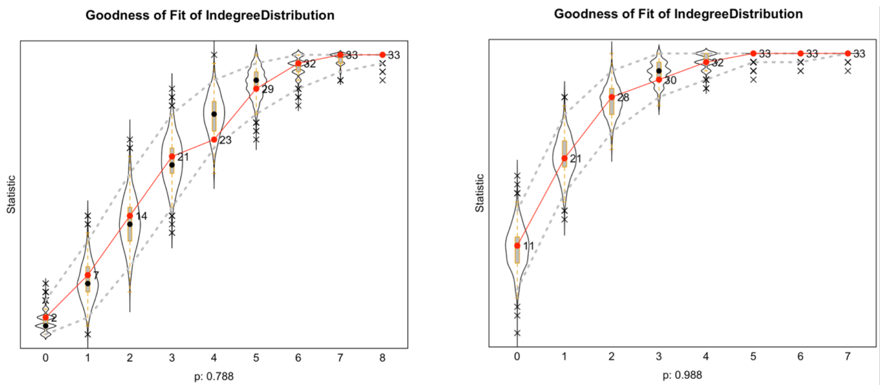
The in-degree variability of the descriptive statistics across the simulated network was presented by violin plots in figure 8. In this case, the circles fit inside the empirical 95% confidence interval which indicates as evidence of good fit. Therefore, we could conclude that our model does a good job of producing simulated networks that have the similar in-degree distribution as observed distribution. On the other hand, as Figure 10 presents, the circles are not perfectly fit inside the 95& confidence interval, but p-values are 0.159 and 0.434 in each friendship.depend and trust.depend which indicate moderately good fit ($p > 0.05$).

Our model did really successfully recreating the observed pattern of directed triads. In the case of friendship network, few cases are failed to recreate the observed patterns, 021D, 111D, 120D, 120U, 210 and 300. For example , the model overestimates the number of 111D type of triad with two directed ties. For 120D and 300 thype of triads, the model underestimates with three ties (120D - two directed ties and one reciprocal tie, 300 - three reciprocal ties) (Figure **??**). On the other hand, only two cases are failed in the case of trust network to recreate the observed patterns, 021D, 021C (two directed ties for both type of triads).

5 Conclusion

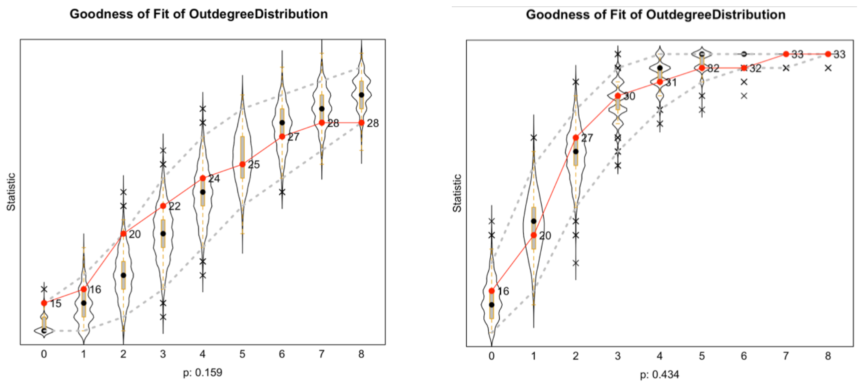
In this paper, we analyzed the relationship of friendship and trust between social relations and gender and drinking behavior. In addition, four hypotheses were tested along with the procedures of the examination. In the early stages, there are more mutual friendship and trust relationships. The interpersonal relationships in the class may have some really popular and active students, and those peers have established a stronger tendency in behavior dynamics. Check the different methods of centralization to analyze how relationships are formed. In addition, there is an obvious gender segregation in networks of friendship and trust. First three hypotheses were tested from those procedures. Furthermore, by conducting hamming, Jaccard distance and QAP, we confirmed the correlation between friendship network and trust network. In addition, triad censuses are a good way to test the social relations in the network. Detecting cliques and communities in the networks would be another way to see how the peers formed their relationship in the classroom as a group. It was clear that there is a relationship change over time, therefore, we conducted a stochastic actor-based model to investigate why it changed. The last hypothesis with many different hypothetical effects was tested through the Stochastic actor-based model. Initially, the hypothesis was proposed that there will be a significant effect of the drinking behavior of ego and alter. In other words, friendships and trust relationships will be formed based on his/her drinking habit and However, the results based on the Stochastic Actor-Based Model concluded the effect is not significant.

The aim of future research is at investigating further why drinking behavior does not have a significant effect on networks of friendship and trust relationships. This might be able to investigate by adding mediation effects, and negative ties. Veenstra et al. (2013) stated the examining mediation process to explain why adolescents change the network or their behavior is allowed for a longitudinal social network model. For the negative ties, traditionally, researchers focused on the positive meaning on the relationship, but Dijkstra, Lindenberg, Verhulst, Ormel, and Veenstra (2009) focused on antipathies in his multiple network approach, antipathies are explained as to "how popularity and friendships affect the



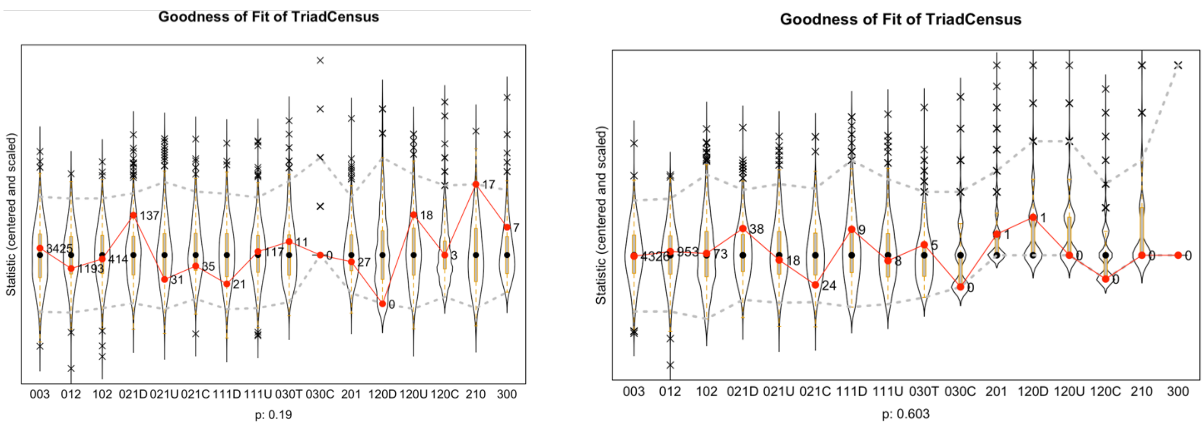
(a) Friendship (b) Trust

Figure 8: Plot of Goodness-of-fit In-degree Distribution



(a) Friendship (b) Trust

Figure 9: Plot of Goodness-of-fit Out-degree Distribution



(a) Friendship (b) Trust

Figure 10: Plot of Goodness-of-fit Triad Census

development of dislike relationships over time". In other words, there will be an effect of the dissimilarity which in popularity which may leads friends in the classroom tend to "agree upon which peers to dislike over time" (Veenstra et al. (2013)). Investigating other behavior effects that leading adolescents to changes in the network and behavior is going to be an additional goal for future research, for example, smoking behavior, using SNS, or consuming junk food, etc.

6 Appendix

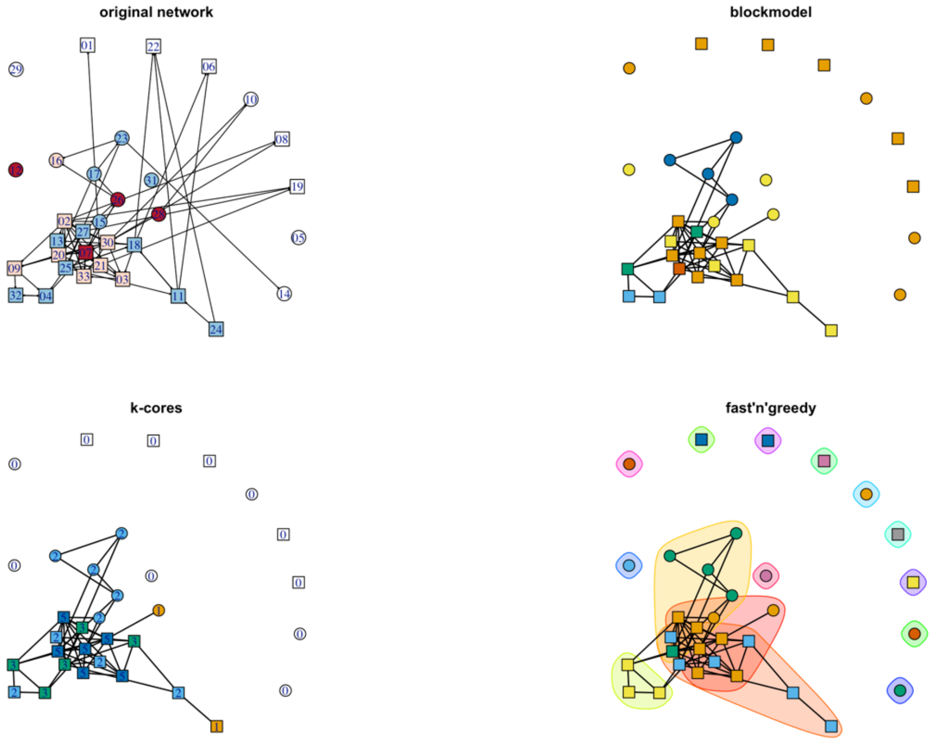


Figure 11: Community Detection Trust w1

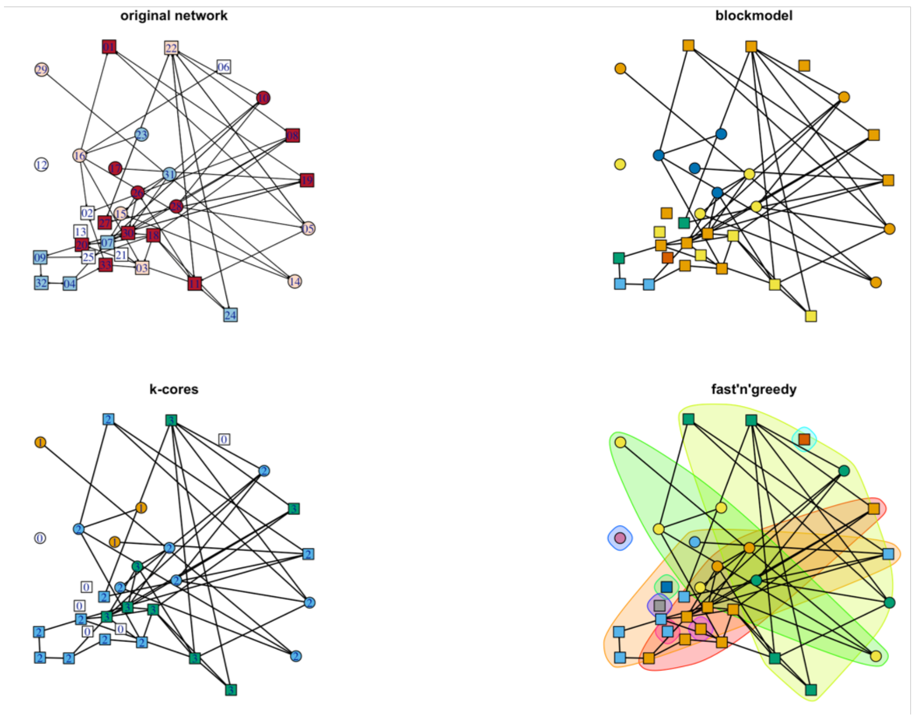


Figure 12: Community Detection Trust w2

References

Bokhove, C. (2018). Exploring classroom interaction with dynamic social network analysis. *International Journal of Research & Method in Education*, *41*(1), 17–37.

Dijkstra, J. K., Lindenberg, S., Verhulst, F. C., Ormel, J., & Veenstra, R. (2009). The relation between popularity and aggressive, destructive, and norm-breaking behaviors: Moderating effects of athletic abilities, physical attractiveness, and prosociality. *Journal of Research on Adolescence*, *19*(3), 401–413.

Hevey, D. (2018). Network analysis: a brief overview and tutorial. *Health Psychology and Behavioral Medicine*, *6*(1), 301–328.

Huang, G. C., Soto, D., Fujimoto, K., & Valente, T. W. (2014). The interplay of friendship networks and social networking sites: longitudinal analysis of selection and influence effects on adolescent smoking and alcohol use. *American journal of public health*, *104*(8), e51–e59.

Snijders, T. (n.d.). van, de, bunt, gg, steglich, ce (2010). *Introduction to stochastic actor-based models for network dynamics. Social Networks*, *32*(1), 44–60.

Snijders, T. A. (2020, August). *Goodness of fit testing in rsiena.* Retrieved from `https://www.stats.ox.ac.uk/~snijders/siena/SienaGOF_s.pdf`

Veenstra, R., Dijkstra, J. K., Steglich, C., & Van Zalk, M. H. (2013). Network–behavior dynamics. *Journal of Research on Adolescence*, *23*(3), 399–412.

Wasserman, S., Faust, K., et al. (1994). *Social network analysis: Methods and applications* (Vol. 8). Cambridge university press.

1 Code

```
# Most of other Codes are implemented and referenced from
# the course work of Network Analysis in KU LEUVEN (2020)
# Due to length of the pdf, not all codes are presented in this section.
# Only codes for visualization are presented in this section.
# Youhee Kil r0768512

# Visualization
rm(list=ls())

libraries <- c('dplyr', 'igraph', 'sna', 'ggplot2', 'vioplot',
               'statnet', 'UserNetR', 'RColorBrewer')
sapply(libraries, require, character.only = T)

# data base
setwd("/Users/youheekil/Downloads/stats/phase 2/Network Analysis/Week 1")

# data preparation
affective_w1 <- as.matrix(read.csv("RECENS_data/6300_affective_w1.csv",
                                   header=TRUE, row.names=1, sep=","))

affective_w2 <- as.matrix(read.csv("RECENS_data/6300_affective_w2.csv",
                                   header=TRUE, row.names=1, sep=","))




sex <- as.matrix(read.csv("RECENS_data/6300_sex.csv",
                          header=TRUE, row.names=1, sep=","))

drink <- as.matrix(read.csv("RECENS_data/6300_drink.csv",
                            header=TRUE, row.names=1, sep=","))

trust_w1 <- as.matrix(read.csv("RECENS_data/6300_trust_w1.csv",
                               header=TRUE, row.names=1, sep=","))

trust_w2 <- as.matrix(read.csv("RECENS_data/6300_trust_w2.csv",
                               header=TRUE, row.names=1, sep=","))

colnames(affective_w1) <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
                            "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                            "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
                            "31", "32", "33")
rownames(affective_w1) <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
                            "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                            "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
                            "31", "32", "33")
colnames(affective_w2) <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
                            "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                            "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
                            "31", "32", "33")
colnames(trust_w1) <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
                       "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                       "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
                       "31", "32", "33")
colnames(trust_w2) <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
                        "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                        "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
                        "31", "32", "33")
rownames(drink) <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10",
                     "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                     "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
```

```
                    "31", "32", "33")

friendship_w1 <- affective_w1
friendship_w2 <- affective_w2

friendship_w1[friendship_w1 %in% c(-2:1)] <- 0
friendship_w1[friendship_w1 == 2] <- 1

friendship_w2[friendship_w2 %in% c(-2:1)] <- 0
friendship_w2[friendship_w2 == 2] <- 1

friend_w1 <- friendship_w1
friend_w2 <- friendship_w2

drink_w1 <- drink[, 1]
drink_w2 <- drink[, 2]

drink_w1[drink_w1 %in% NA] <- 0
drink_w2[drink_w2 %in% NA] <- 0

#detach(package:sna)
#detach(package:statnet)
#library(igraph)
# Let's check how the networks look like!
g1 <- graph.adjacency(friend_w1)
g2 <- graph.adjacency(friend_w2)
g3 <- graph.adjacency(trust_w1)
g4 <- graph.adjacency(trust_w2)
g1234 <- graph.adjacency(friend_w1 + friend_w2 + trust_w1 + trust_w2)

myLayout <- layout.auto(g1234)



par(mfrow = c(1 , 1))
plot(g1,
     vertex.shape = ifelse(sex == 2,"square", "circle"),
     vertex.color = ifelse(drink_w1==3 | drink_w1==2, "red", "yellow"),
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
     main = "Friendship network wave 1")


plot(g2,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     #vertex.color = ifelse(drink_w2==0, "darkgreen", "white"),
     vertex.color = ifelse(drink_w1==3 | drink_w1==2, "red", "yellow"),
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
     main = "Friendship network wave 2")
plot(g3,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color = ifelse(drink_w1==3 | drink_w1==2, "red", "yellow"),
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
```

```
     main = "Trust network wave 1")
legend("topleft",
       c("drink daily", "drink not daily"),
       fill = c("red", "yellow"),
       inset=.02,
       horiz=F,
       cex=0.5,
       box.lty=2, box.lwd=0.2, box.col="black")

plot(g4,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color = ifelse(drink_w1==3 | drink_w1==2, "red", "yellow"),
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
     main = "Trust network wave 2")
legend("topleft",
       c("drink daily", "drink not daily"),
       fill = c("red", "yellow"),
       inset=.02,
       horiz=F,
       cex=0.5,
       box.lty=2, box.lwd=0.2, box.col="black")

par(mfrow = c(1, 1))

# now a combined view
friendtrust_w1 <- friend_w1 + 2*trust_w1
friendtrust_w2 <- friend_w2 + 2*trust_w2


ft1 <- graph.adjacency(friendtrust_w1, weighted=TRUE)
ft2 <- graph.adjacency(friendtrust_w2, weighted=TRUE)

my_pal <- brewer.pal(8,"Accent")
display.brewer.pal(n = 8, name = 'RdBu')
brewer.pal(n = 8, name = "RdBu")

par(mfrow = c(1,1))
plot(ft1,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color =  ifelse(drink_w1== 0, "white", ifelse(drink_w1 ==1 , "#FDDBC7",
    ifelse(drink_w1==2, "#92C5DE", "#B2182B"))),
     edge.color = ifelse(E(ft1)$weight == 1, "#4393C3", ifelse(E(ft1)$weight == 2, "#
    BEAED4", "darkred")), #green,purple, orange
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
     main = "Friendship and trust networks wave 1")
legend("topleft",
       legend = c("Friendship Only", "Trust Only", "Friendship and Trust"),
       col = c("#4393C3", "#BEAED4", "darkred"),
       lty=1,
       bty = "n",
#       pch = c(19,19,19),
       pt.cex = 1,
       cex = 0.8,
       text.col = "black",
       horiz = F)
legend("bottomleft",
       legend = c("Absent", "Non Drinking", "Frequently Drinking", "Usually Drinking"),
```

```
      fill = c("white", "#FDDBC7", "#92C5DE", "#B2182B"),
      bty = "n",
      pch = c(19,19,19),
      pt.cex = 1,
      cex = 0.8,
      text.col = "black")

plot(ft2,
    vertex.shape = ifelse(sex == 2,"square", "circle" ),
    vertex.color =  ifelse(drink_w2 == 0, "white", ifelse(drink_w2 ==1 , "#FDDBC7",
    ifelse(drink_w2==2, "#92C5DE", "#B2182B"))),
     # ifelse(drink_w1==3 | drink_w1==2, "red", "yellow"),
    edge.color = ifelse(E(ft2)$weight == 1, "#4393C3", ifelse(E(ft2)$weight == 2, "#
    BEAED4", "darkred")), #green,purple, orange
    edge.width = 1,
    edge.arrow.size = 0.1,
    vertex.size = 8,
    layout = myLayout,
    main = "Friendship and trust networks wave 2")
legend("topright",
      legend = c("Friendship Only", "Trust Only", "Friendship and Trust"),
      col = c("#4393C3", "#BEAED4", "darkred"),
      lty=1,
      bty = "n",
      #       pch = c(19,19,19),
      pt.cex = 1,
      cex = 0.8,
      text.col = "black",
      horiz = F)
legend("bottomright",
      legend = c("Absent", "Non Drinking", "Frequently Drinking", "Usually Drinking"),
      fill = c("white", "#FDDBC7", "#92C5DE", "#B2182B"),
      bty = "n",
      pch = c(19,19,19),
      pt.cex = 1,
      cex = 0.8,
      text.col = "black")


# community & Cliques


friend_w1 <-friendship_w1 + t(friendship_w1)
friend_w2 <-friendship_w2 + t(friendship_w2)
trust_w1 <- trust_w1 + t(trust_w1)
trust_w2 <- trust_w2 + t(trust_w2)
table(friend_w2)
table(trust_w1)
friend_w1[friend_w1 ==2 ] <- 1
friend_w2[friend_w2 ==2 ] <- 1
trust_w1[trust_w1==2] <- 1
trust_w2[trust_w2==2] <- 1


library(igraph)

friend1 <- graph.adjacency(friend_w1)
friend1 <- as.undirected(friend1)
components <- decompose.graph(friend1)
table(sapply(components, vcount))

friend2 <- graph.adjacency(friend_w2)
friend2 <- as.undirected(friend2)
components <- decompose.graph(friend2)
```

```
table(sapply(components, vcount))


trust1<- graph.adjacency(trust_w1)
trust1 <- as.undirected(trust1)
components <- decompose.graph(trust1)
table(sapply(components, vcount))

trust2 <- graph.adjacency(trust_w2)
trust2 <- as.undirected(trust2)
components <- decompose.graph(trust2)
table(sapply(components, vcount))
myLayout <- layout.fruchterman.reingold(friend1)
myLayout2 <- layout.fruchterman.reingold(friend2)

## previous plot with new layout
plot(ft1,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color =  ifelse(drink_w1== 0, "white", ifelse(drink_w1 ==1 , "#FDDBC7",
    ifelse(drink_w1==2, "#92C5DE", "#B2182B"))),
     edge.color = ifelse(E(ft1)$weight == 1, "#4393C3", ifelse(E(ft1)$weight == 2, "#
    BEAED4", "darkred")), #green,purple, orange
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
     main = "Friendship and trust networks wave 1")
legend("topleft",
       legend = c("Friendship Only", "Trust Only", "Friendship and Trust"),
       col = c("#4393C3", "#BEAED4", "darkred"),
       lty=1,
       bty = "n",
       #       pch = c(19,19,19),
       pt.cex = 1,
       cex = 0.8,
       text.col = "black",
       horiz = F)
legend("bottomleft",
       legend = c("Absent", "Non Drinking", "Frequently Drinking", "Usually Drinking"),
       fill = c("white", "#FDDBC7", "#92C5DE", "#B2182B"),
       bty = "n",
       pch = c(19,19,19),
       pt.cex = 1,
       cex = 0.8,
       text.col = "black")

library(statnet)
library(igraph)
# how many cliques are in the netowrk
cliques <- cliques(friend1)
length(cliques) # this is look like a long object, why ?
# of course each clique can be a subset of larger cliques
table(sapply(maximal.cliques(friend1), length))

#cliques _friend1
cliques <- cliques(friend1)
length(cliques)
table(sapply(maximal.cliques(friend1), length))

#cliques_friend2
cliques2 <- cliques(friend2)
length(cliques2)
table(sapply(maximal.cliques(friend2), length))

```

```
#cliques _trust
cliques3 <- cliques(trust1)
length(cliques3)
table(sapply(maximal.cliques(trust1), length))

#cliques_friend2
cliques4 <- cliques(trust2)
length(cliques4)
table(sapply(maximal.cliques(trust2), length))


#blocks_friend1
equiv.w1 <- equiv.clust(friendship_w1, cluster.method="ward.D2", method="hamming")
bm.w1 <- blockmodel(friendship_w1, equiv.w1, k=9)
(block.members <-bm.w1$block.membership[order(bm.w1$order.vector)])

# blocks_friend2

equiv.w2 <- equiv.clust(friendship_w2, cluster.method = "ward.D2", method = "hamming")
plot(equiv.w2)
bm.w2 <- blockmodel(friendship_w2, equiv.w2, k=7)
(block.members2 <-bm.w2$block.membership[order(bm.w2$order.vector)])

# blocks_trust1
equiv.tw1 <- equiv.clust(trust_w1, cluster.method = "ward.D2", method="hamming")
plot(equiv.tw1)
bm.tw1 <- blockmodel(trust_w1, equiv.tw1, k = 6)
(block.members3 <-bm.tw1$block.membership[order(bm.tw1$order.vector)])

equiv.tw2 <- equiv.clust(trust_w2, cluster.method = "ward.D2", method="hamming")
plot(equiv.tw2)
bm.tw2 <- blockmodel(trust_w2, equiv.tw1, k = 6)
(block.members4 <-bm.tw2$block.membership[order(bm.tw2$order.vector)])




# by applying the concept of k-cores
cores <- graph.coreness(friend1)
cores2 <- graph.coreness(friend2)
cores3 <- graph.coreness(trust1)
cores4 <- graph.coreness(trust2)
table(cores)
table(cores)
table(cores4)

library(grDevices)
library(RColorBrewer)
V(friend1)$name <- cores
V(friend1)$color <- cores

friend1_6 <- friend1
friend2_6 <- induced.subgraph(friend1,
                              vids = which(cores >1))
friend3_6 <- induced.subgraph(friend1,
                              vids = which(cores >2))
friend4_6 <- induced.subgraph(friend1,
                              vids = which(cores >3))
friend5_6 <- induced.subgraph(friend1,
                              vids = which(cores >4))
friend6_6 <- induced.subgraph(friend1,
                              vids = which(cores >5))

op <- par(mfrow =c(3,2), mar = c(c(3,0,2,0)))
```

```

plot(friend1_6,layout=myLayout,main="All k-cores")
plot(friend2_6,layout=myLayout[which(cores > 1),], main="k-cores 2-6")

plot(friend3_6,layout=myLayout[which(cores > 2),], main="k-cores 3-6")

plot(friend4_6,layout=myLayout[which(cores > 3),], main="k-cores 4-6")

plot(friend5_6,layout=myLayout[which(cores > 4),], main="k-cores 5-6")

plot(friend6_6,layout=myLayout[which(cores > 5),], main="k-cores 6-6")
par(op)


par(mfrow = c(1 , 1))

#plot the network with node colors showing k-cores
plot(friend1,
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 8,
     layout = myLayout,
     #edge.width = 1.5,
     #edge.arrow.size = 0.25,
     #vertex.size = 6,
     #vertex.label = cores,
     #vertex.color= cores,
     main="All k-cores")


# facst-greedy community detection algorithm

communities <- fastgreedy.community(friend1)
communities2 <- fastgreedy.community(friend2)
communities3 <- fastgreedy.community(trust1)
communities4 <- fastgreedy.community(trust2)
length(communities)
sizes(communities)
membership(communities)

plot(communities, friend1,edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.1,
     vertex.size = 10,
     #vertex.label = cores,
     #vertex.color= cores,
     #vertex.label = "",
     layout=myLayout)
par(mfrow=c(1,1))

g1
friend1


##====================================================
## summary of community detection                   ###
##===================================================

#(friendship w1)
library(sna)
par(mfrow = c(2,2), mar=c(3,0,2,0))
# the original friendship network

```

```
plot(g1,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color =  ifelse(drink_w1== 0, "white", ifelse(drink_w1 ==1 , "#FDDBC7",
    ifelse(drink_w1==2, "#92C5DE", "#B2182B"))),
     edge.color = "black",
     edge.width = 1,
    edge.arrow.size = 0.05,
     vertex.size = 10,
   #  vertex.label = "",
     vertex.color="skyblue",
     layout=myLayout,
     main="original network")

# the blockmodel (from the previous script)
plot(friend1,
     edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
    vertex.label = "",
    vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color= block.members,
     layout=myLayout,
     main="blockmodel")
# the k-cores
plot(friend1,
     edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
   #  vertex.label = "",
     vertex.color= cores,
     layout=myLayout,
     main="k-cores")
# communty detection
# the fast-greedy communities
plot(communities, friend1,edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.label = "",
     layout=myLayout,
     main="fast'n'greedy")
par(mfrow=c(1,1))

##=====================================================
## summary of community detection                    ###
##====================================================

#(friendship w2)
library(sna)
par(mfrow = c(2,2), mar=c(3,0,2,0))
# the original friendship network

plot(g2,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color =  ifelse(drink_w2== 0, "white", ifelse(drink_w2 ==1 , "#FDDBC7",
    ifelse(drink_w2==2, "#92C5DE", "#B2182B"))),
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.05,
     vertex.size = 10,
```

```
      #  vertex.label = "",
      vertex.color="skyblue",
      layout=myLayout,
      main="original network")

# the blockmodel (from the previous script)
plot(friend2,
      edge.color = "black",
      edge.width = 1.5,
      edge.arrow.size = 0.25,
      vertex.size = 8,
      vertex.label = "",
      vertex.shape = ifelse(sex == 2,"square", "circle" ),
      vertex.color= block.members2,
      layout=myLayout,
      main="blockmodel")
# the k-cores
plot(friend2,
      edge.color = "black",
      edge.width = 1.5,
      edge.arrow.size = 0.25,
      vertex.size = 8,
      vertex.shape = ifelse(sex == 2,"square", "circle" ),
      vertex.label =cores2,
      vertex.color= cores2,
      layout=myLayout,
      main="k-cores")
# communty detection
# the fast-greedy communities
plot(communities2, friend2,edge.color = "black",
      edge.width = 1.5,
      edge.arrow.size = 0.25,
      vertex.size = 8,
      vertex.shape = ifelse(sex == 2,"square", "circle" ),
      vertex.label = "",
      layout=myLayout,
      main="fast'n'greedy")

##=====================================================##
## summary of community detection(trust 1)            ##
##=====================================================##

#(friendship w2)
library(sna)
par(mfrow = c(2,2), mar=c(3,0,2,0))
# the original friendship network

plot(g3,
      vertex.shape = ifelse(sex == 2,"square", "circle" ),
      vertex.color =  ifelse(drink_w1== 0, "white", ifelse(drink_w1 ==1 , "#FDDBC7",
     ifelse(drink_w1==2, "#92C5DE", "#B2182B"))),
      edge.color = "black",
      edge.width = 1,
      edge.arrow.size = 0.05,
      vertex.size = 10,
      #  vertex.label = "",
      vertex.color="skyblue",
      layout=myLayout,
      main="original network")

# the blockmodel (from the previous script)
plot(trust1,
      edge.color = "black",
      edge.width = 1.5,
```

```r
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.label = "",
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color= block.members3,
     layout=myLayout,
     main="blockmodel")
# the k-cores
plot(trust1,
     edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.label =cores3,
     vertex.color= cores3,
     layout=myLayout,
     main="k-cores")
# communty detection
# the fast-greedy communities
plot(communities3, trust1,edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.label = "",
     layout=myLayout,
     main="fast'n'greedy")


##=====================================================##
## summary of community detection(trust 2)             ##
##=====================================================##

#(friendship w2)
library(sna)
par(mfrow = c(2,2), mar=c(3,0,2,0))
# the original friendship network

plot(g4,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color =  ifelse(drink_w2== 0, "white", ifelse(drink_w2 ==1 , "#FDDBC7",
    ifelse(drink_w2==2, "#92C5DE", "#B2182B"))),
     edge.color = "black",
     edge.width = 1,
     edge.arrow.size = 0.05,
     vertex.size = 10,
     #  vertex.label = "",
     vertex.color="skyblue",
     layout=myLayout,
     main="original network")

# the blockmodel (from the previous script)
plot(trust2,
     edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.label = "",
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.color= block.members4,
     layout=myLayout,
     main="blockmodel")
# the k-cores
```

```
plot(trust2,
     edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.label =cores4,
     vertex.color= cores4,
     layout=myLayout,
     main="k-cores")
# communty detection
# the fast-greedy communities
plot(communities4, trust2,edge.color = "black",
     edge.width = 1.5,
     edge.arrow.size = 0.25,
     vertex.size = 8,
     vertex.shape = ifelse(sex == 2,"square", "circle" ),
     vertex.label = "",
     layout=myLayout,
     main="fast'n'greedy")



vioplot(trust.random.triad_w1[,1], trust.random.triad_w1[,3], trust.random.triad_w1
    [,11], trust.random.triad_w1[,16],
        names=colnames(trust.random.triad_w1)[c(1, 3, 11, 16)],
        col="transparent",
        ylim=c(0, 2500),
        main = "Distribution of all closed traignels for Trust W1")
# and mark the observed numbers in each category
points(1:4,
       trust.triad.count_w1[c(1, 3, 11, 16)],
       col="blue",
       type="b",
       pch=15)
# and mark the observed numbers in each category
points(1:4,
       trust.triad.count_w2[c(1, 3, 11, 16)],
       col="red",
       type="b",
       pch=15)
legend("topright",
       c("Trust w1", "Trust w2"),
       fill = c("blue", "red"),
       inset=.02,
       horiz=F,
       cex=1,
       box.lty=1, box.lwd=0.2, box.col="black")



#=========================================================
#           Siena
#=========================================================


# the first dependent network - friendship
friendship.dependent1 <- sienaDependent(array(c(friend_w1, friend_w2), dim=c(33, 33, 2))
    )
# the second dependent network - trust
trust.dependent2 <- sienaDependent(array(c(trust_w1, trust_w2), dim=c(33, 33, 2)))
drinking <- sienaDependent(drink, type = "behavior")
#drinking <- sienaNet(drink,type="behavior")

```

```
# constant covariate - gender
gender.coCovar <- coCovar(sex[,1])

# create siena data object
sienaData1 <- sienaDataCreate(friendship.dependent1,
                              trust.dependent2,
                              drinking,
                              gender.coCovar)

# print initial report to file
print01Report(sienaData1, modelname="RECENS_multiplex_report")


####################################################
# Step 3: specify SIENA model
####################################################

sienaEffects <- getEffects(sienaData1)

# the default specification for multiplex data
sienaEffects

# let's look at the available effects
effectsDocumentation(sienaEffects)

# first it makes sense to include effects for the two networks separately

sienaEffects <- includeEffects(sienaEffects, egoX, altX, sameX,
                               interaction1="gender.coCovar", name = "friendship.dependent1")
sienaEffects <- includeEffects(sienaEffects, egoX, altX, sameX,
                               interaction1="gender.coCovar", name = "trust.dependent2")
sienaEffects <- includeEffects(sienaEffects, egoX, altX, sameX,
                               interaction1="drinking", name = "friendship.dependent1")
sienaEffects <- includeEffects(sienaEffects, egoX, altX, sameX,
                               interaction1="drinking", name = "trust.dependent2")

sienaEffects <- includeEffects(sienaEffects,name="drinking",
                               totSim,interaction1="friendship.dependent1",
                               include = FALSE)
sienaEffects <- includeEffects(sienaEffects,name="drinking",
                               totSim,interaction1="trust.dependent2",
                               include = FALSE)

sienaEffects <- includeEffects(sienaEffects,name="drinking",
                               effFrom,interaction1="gender.coCovar",
                               include = FALSE)

# then add the basic dyadic cross-production effect
sienaEffects <- includeEffects(sienaEffects, name="friendship.dependent1",
                               crprod,
                               interaction1="trust.dependent2")
sienaEffects <- includeEffects(sienaEffects, name="trust.dependent2",
                               crprod,
                               interaction1="friendship.dependent1")

# and control for degree correlations between the two networks
# (note: these are square root effects by default- they usually fit better and
#        are more stable)
sienaEffects <- includeEffects(sienaEffects, name="friendship.dependent1",
                               outActIntn,
                               interaction1="trust.dependent2",
                               include = FALSE)
sienaEffects <- includeEffects(sienaEffects, name="trust.dependent2",
```

```
                                outActIntn,
                                interaction1="friendship.dependent1",
                                include = FALSE)

# look at the specification
sienaEffects


####################################################
#####  Step 4: create estimation algorithm     #####
####################################################

sienaAlgorithm <- sienaAlgorithmCreate(projname="RECENS_multiplex_algo")


####################################################
######      Step 5: estimate SIENA model       #####
####################################################

# estimate model
set.seed(999)
model1 <- siena07(sienaAlgorithm, data = sienaData1, effects = sienaEffects, returnDeps
    = TRUE)
result <- model1



#If 'Overall maximum convergence ratio' is greater than .25 rerun the model

# rerun estimation if model has not converged

outTable <- function(x) {
   coef <- abs(x$theta)
   coefPretty <- sprintf("%.3f", round(coef,3))
   se <- diag(x$covtheta)**.5
   sePretty <- sprintf("%.3f", round(se,3))
   pval <- 2*pnorm(-abs(coef/se))
   symp <- symnum(pval, corr = FALSE,
                  cutpoints = c(0,  .001,.01,.05, .1, 1),
                  symbols = c("***","**","*","."," "))
   convPretty <- sprintf("%.3f", round(abs(x$tconv),3))
   out1 <- noquote(cbind(
      Function = x$effects[[1]],
      Effect = x$effects[[2]],
      Coef = coefPretty,
      StEr = sePretty,
      Sig = symp,
      Conv = convPretty))
   out2 <- paste("Maximum Convergence Ratio:", round(x$tconv.max,3))
   return(list(out1,out2))
}

outTable(model1)

## GOF
gof.id <- sienaGOF(result, IndegreeDistribution, verbose=TRUE, join=TRUE,
                   varName="friendship.dependent1")
plot(gof.id)
gof.id2 <- sienaGOF(result, IndegreeDistribution, verbose=TRUE, join=TRUE,
                    varName="trust.dependent2")
plot(gof.id2)

gof.od1 <- sienaGOF(result, OutdegreeDistribution, verbose=TRUE, join=TRUE,
                    varName="friendship.dependent1")
```

```
gof.od <- sienaGOF(result, OutdegreeDistribution, verbose=TRUE, join=TRUE,
                   varName="trust.dependent2")
plot(gof.od1)
plot(gof.od)

TriadCensus <- function(i, data, sims, wave, groupName, varName, levls=1:16){
   #unloadNamespace("igraph") # to avoid package clashes
   require(sna)
   require(network)
   x <- networkExtraction(i, data, sims, wave, groupName, varName)
   if (network.edgecount(x) <= 0){x <- symmetrize(x)}
   # because else triad.census(x) will lead to an error
   tc <- sna::triad.census(x)[1,levls]
   # names are transferred automatically
   tc
}

#detach(package:igraph)
goftc <- sienaGOF(result, TriadCensus, varName="friendship.dependent1",
                  verbose=TRUE, join=TRUE)
descriptives.sienaGOF(goftc)
plot(goftc)

# triad census
(gof.tc <- sienaGOF(result, verbose=TRUE, varName="friendship.dependent1", TriadCensus,
    join=T) )
plot(goftc, scale=TRUE, center=TRUE)
# Really good fit

BehaviorDistribution(result, BehaviorDistribution,
                     varName = "drinking", cumulative = FALSE)
```