DSC 480 SOCIAL NETWORK ANALYSIS

FINAL REPORT

XUYANG JI, ANGELO KELVAKIS, MIMIDOO GYOH

JUNE 3RD, 2023

VI. Executive Summary

Network Analysis Corp (NAC) was tasked with compiling company-level network data on the employees at Real Shade Sunglasses company (RSS). Members of the NAC data science team interviewed employees from each department of RSS: Executive, Marketing, Sales, HR, Distribution, Manufacturing, and Finance. The employees that were interviewed were asked basic questions about everyday life at their company in an effort to compile three different networks. The networks that were developed were based on the three research questions:

1. Which coworkers do you notice spending time together during lunch break or outside of work?
2. Who do you observe your coworkers interacting with each other on work-related matters?
3. Who do you observe your coworkers giving advice to?

From those three questions we were able to generate three networks: Friendship, Work, and Advice. The Friendship Network houses information about employees who talked about hobbies, discussed their personal lives, and had informal discussions about work-related topics in a non-professional capacity. The Work Network housed all of the data on who was described working together on projects, giving updates on material, and engaged in meetings. The Advice Network has information on who gives professional advice to each other and can give insights into the hierarchy of the company and who feels like they need information and who gives it. The Advice Network also includes data from a mentorship program where employees were paired with each other based on seniority and a pre-filled personal survey in order to promote cross-department cohesion within the company. This data was added on top of the research question data.

VII. Research Scope

The data was collected in an interview style with 3 members of NAC interviewing one employee at a time and asking the three research questions above in a randomized pattern with randomized selection of the employees so that no single department was sampled more than another. The response from the interviewee was recorded in third person with the names of the employees recorded and then encoded in the final data.

Along with the research questions and interviews, a post-interview survey was sent out asking everyone who was interviewed about their physical appearance (hair color), gender, corrective lenses usage, and hobbies. This data was stored within the code book where survey questions were joined to company data on employee names, age, years worked, and department. You can find keys to each numeric attribute within this codebook as well.

The three networks were compiled from edgelists where question data was converted into interaction data for the edgelist. This was then formatted using R to compile adjacency matrices for analysis. The R packages “xUCINET”, “sna”, and “igraph”, were used to collect statistical data on the networks in order to compare metrics like density, transitivity, dyadic and triadic

relationships, and centrality. The networks were also visualized and information on how the network related to the information from the research questions was analyzed.

VIII. Governance

The Real Shade Sunglasses company is headed by a team of executives a1, a2, a3, and a4. These four participants guide the daily work of all the departments and make sure that the Shade is indeed real. They are supported by senior staff of each department and were all included in the interview process. At this company, there are contracts and agreements in place. Some data may be restricted and some data will need to be encrypted and anonymized. A mentorship program was set up to ensure that all members of the departments have access to information and can better form interpersonal relationships with their colleagues at the organization. The mentorship is structured to have a lead mentor from each department and several employees from various departments. NAC was given access to some select documents by the executive teams including the mentorship roster , basic demographics and employee engagement programs (hobbies). Overall, while the hierarchy can be clearly distinguished by seniority, based on the data below, you will find that this network is mostly sparse, with weak interpersonal relationships, stronger working relationships, and contains mild advice relationships. The governance in this study is that each employee was asked by a governing participant (executive department and the heads of their departments) to talk to NAC. They were given a 10 minute time allotment and 70 members complied.

IX. Network Data Cleaning & Preprocessing

The dataset comes from the week-long interviews of 70 employees cross Executive, Marketing, Sales, Human Resources, Distribution, Manufacturing, and Finance departments in the sunglass company, in which we ask all employees to list out the relevant employee interactions in certain scenarios, the location where it happened, and the names of involved employees. Employees' personal information was collected and encoded as below, including gender, need for corrective lenses, hair color, seniority level (years spent working for the company), hobbies, and mentorship status. Following negotiations with management, the names are encoded and anonymous as a combination of letter and sequence. NAC coded each partner with the name a# where # represented a numeric increment which serves as the identification for the participants in this study.

Department	Code	Gender	Code	Lenses	Code	Hair	Code	Seniority	Code	Hobby	Code	Mentorship	Code
Executive	0	Female	0	No need	0	Blonde	0	Entry: 0-6	0	Poker	0	Mentor	1
Marketing	1	Male	1	Needs	1	Brown	1	Junior: 6-10	1	Tennis	1	Mentee	0
Sales	2					Black	2	Senior: 10+	2	Cooking	2		

Human Resources	3					Red	3	*yrs have worked		Camping	3		
Distribution	4									Golf	4		
Manufacture	5									Coffee Roasting	5		
Finance	6									Video Games	6		

To better understand the relationships within the company, we have observed various types of interactions, including mentorship, friendship, work-related, and etc. Our dataset is stored as comp.network where all friendship, working, and advice ties are in edgelist format in csv file. In the undirected working and friendship edgelist, a value of 1 indicates there is a mutual relationship presents between the vertices; whereas the a value range from 1 to 4 in the directed and weighted advices network indicates the quality and importance of the advices given from the source vertex to the receiver based on their seniority level. To ensure a smooth conversion from the edgelist to adjacency matrix, all possible vertices' names are appended with a weight value of 0; then sorted the rows into appropriate sequences.

Meantime, given the fact there were intragroup interactions within the departments and intergroups interactions between cross-functional departments; we first use matrix multiplication to get the adjacency matrix of the intragroup interactions with the mmult formula in Excel, and then use R to transform the intergroup interactions which stored as a edgelist into an adjacency matrix. Finally, these two matrices are merged with the condition that if two vertices have a relationship present in the intergroup adjacency matrix, then fill the value of 1; otherwise, fill the value as in the intragroup adjacency matrix.

**Details about the formula can be found in Merged_working workbook.*

After importing the relevant CSV file into R, additional rows are firstly trimmed to keep data consistent and clean. Finally igraph objects are created for each network with appropriate flags, such as directed and weighted in the 'advices' network. The graph.data.frame function would simultaneously import the vertex attributes and edge attributes if there are more than 2 columns. An important step is to filter out any edges with a weight of 0, where *simplify* is used. Due to the way the data is imported, there are self-loops with a weight of 0 along the diagonal of the matrix, which would cause problems for sociomatrix transformation. To retain multiple edges while removing loops, the *remove.loops* parameter is set to true. Finally, the edgelist matrices are transformed into adjacency/sociomatrix with *get.adjacency.* For the sociomatrix of 'advices', the attribute weights is added simultaneously by setting *attr= 'weights*'.

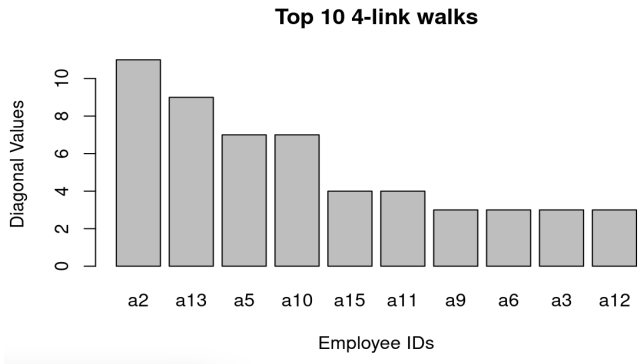
```r
library(igraph)
friendship <- read.csv('EdgeList_ds/friendship_edgelist.csv')
friends <- graph.data.frame(as.matrix(friendship), directed = FALSE)
friends2 <- simplify(friends, remove.multiple = FALSE, remove.loops = TRUE)
friend_adjacency <- get.adjacency(friends2, sparse = FALSE)
class(friend_adjacency)
friend_adjacency
write.csv(friend_adjacency,"friend_adjacency.csv")

working <- read.csv('EdgeList_ds/working_edgeList.csv', header=FALSE)
work <- graph.data.frame(as.matrix(working), directed = FALSE)
work2 <- simplify(work, remove.multiple = TRUE, remove.loops = TRUE)
work_adjacency <- get.adjacency(work2, sparse = FALSE)
      You, 21 hours ago • uploaded working_adjacency matrix …
work_adjacency
write.csv(work_adjacency,"work_adjacency.csv")

advices <- read.csv('EdgeList_ds/advice-Edgelist.csv', header = TRUE)
advices <- advices[c(1:135),]
advices <- graph_from_data_frame(advices, directed = TRUE)
advices2<- simplify(advices, remove.multiple = TRUE, remove.loops = TRUE)
advice_adjacency <- get.adjacency(advices2, sparse = FALSE, attr = 'weight' )
write.csv(advice_adjacency,"advice_adjacency.csv")
```

V. Data Analysis

Friendship Network

I. I. Network Shape And Basic Descriptions

One method of calculating who is the most well connected within a friendship network, the adjacency matrix was multiplied 4 times to see which employees had paths back to themselves. As seen by the graph of the first 10 sorted values, a2 has 11 different 4-vertex paths to get back to themselves, a13 has 9, a5 and a10 have 7, a15 and a11 have 4, and several others have 3, 2, and 1. The conclusion can be drawn that a2 and a11 are the most popular employees at the sunglasses company. They have the highest amount of connections in a friendship network and have been observed to have the most performing social interactions. a5 and a10 are close behind them with slightly fewer. Only the top ten are shown as trailing values are all equal to 1 or 0.



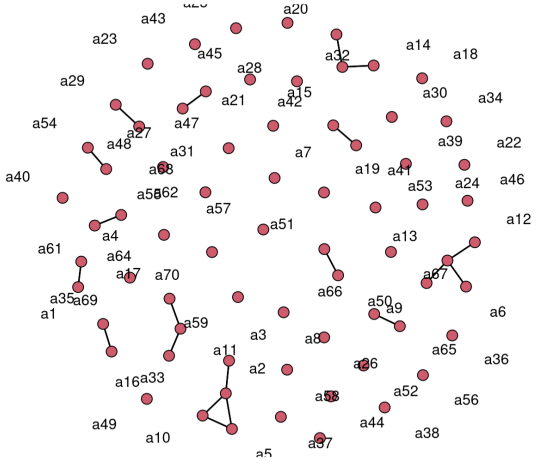
In order to gather metrics on this network, density is compared to the transitivity and the number of isolates. In this network there are 70 nodes. This network is very sparse as the density is only 0.0083. This means that the network is very spread out with many more nodes than links and people who are friends only happen between a handful of individuals. There are no wide-spread networks of friendships. The Transitivity of this network is equal to 0.3 meaning that there are more triads than dyads. This can be interpreted as the people who have friendships are connected to another employee and are not simply one on one. With almost half of the network containing isolates, it may be concluded that either the interviewers did not collect data on the actual friendships present in the company, or this is a business culture where friendships are not openly displayed / not present.

II. Measure Of Centrality

The three methods of centrality used to compare the network are degree, betweenness, and eigenvector. As seen by the ordered outputs of the centralized data, the outcome from the matrix multiplication is similar with the degree mapping to the order of the 4-connection output from above. The between data starts to diminish quickly after the first couple entries and the EV values follow somewhat of a similar pattern as the degree values. Our data does not show a strong correlation between any of the metrics where a change in degree explains only 65% of the variation in between and only 52% of variation in ev. The correlation between ev and between is even lower with only 28% explained.

III. Graphing Of The Network

The numeric outputs from calculating transitivity and density can be more clearly seen with the graph output where the sparsity is easy to see and the lack of complex networks shows why the density is so low. This is again shown through the histogram of geodesics where the lengths of all the connections are zero, meaning that the distance between employees cannot be calculated due to the lack of linkages between nodes.



DSC 480 Network Analysis
Xuyang JI, Angelo Kelvakis, Mimidoo Gyoh

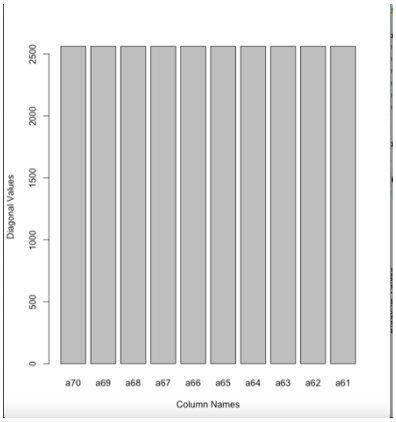
Working Network

I. Clustering coefficient and Basic Description

In this network there are 70 nodes. The network is relatively dense with a density of 0.1602 suggesting that the network is more dense than sparse. There are closely connected networks of work. It has a clustering coefficient (transitivity) of 96% meaning this network has a high level of transitivity. This means individuals are more likely to have connections to other individuals who are also connected to each other therefore forming clusters within the network. This can have important implications for diffusion of information.
There are no isolates. It may be concluded that every participant in the working network has at least one connection.

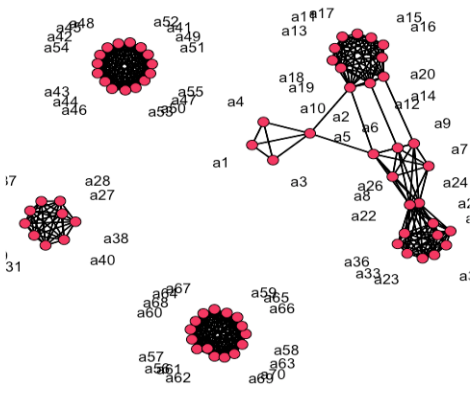
II. Network Paths (4-Link path)

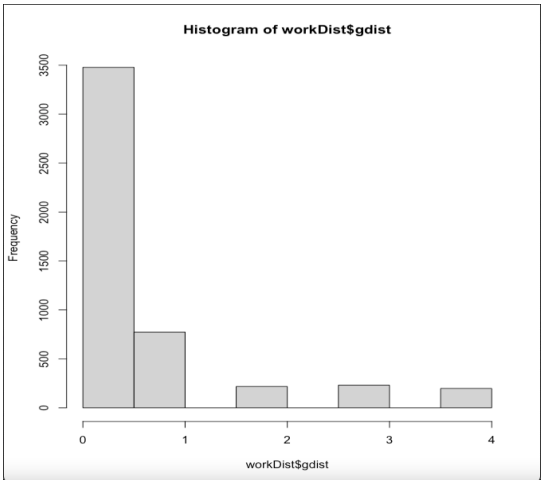
One method of calculating who is the most well connected within a working network, the adjacency matrix was multiplied 4 times to see which employees had paths back to themselves.
As seen by the graph of the last ten sorted values, they all have about 2600 4-vertex paths to get to themselves. This means that they have a similar and high level of connectivity within a work setting in the sunglass company. They all have a high amount of connections within a working network.



III. Graph Of The Network

The numeric output from calculating the clustering coefficient (transitivity) and density can be seen clearly with the graph output. This is a left skewed distribution. This suggests that there are many pairs of nodes with relatively small geodesic distances but only a few pair of nodes with large geodesic distances. It also suggests that these nodes are closely connected and that information or influence can flow easily between them.





DSC 480 Network Analysis
Xuyang JI, Angelo Kelvakis, Mimidoo Gyoh

Advising Network

I. Data Transformation - Dichotomization

While the weights in the directed advising network indicates the importance and quality of the advices and mentorship had given by the source vertices, with 1 being very little and 5 being a great deal. Dichotomization technique is used to transform the asymmetric raw data, which refers to converting valued data into binary data. The reason behind this is that graph-theoretic methods are only applicable to binary data. We take the valued adjacency matrix and set all cells with a tie strength greater than the threshold value of 0 as 1, ,and set al the remaining cells to 0. After the transformation, there are approximately 2.4% of the vertice pairs have a tie (i.e. density=0.024), with a transitivity score of 0.26 and 4830 possible dyads. The 26% transitivity score shows the ratio of connected triples existed in the network, further indicating the company's top-down management/advising system.

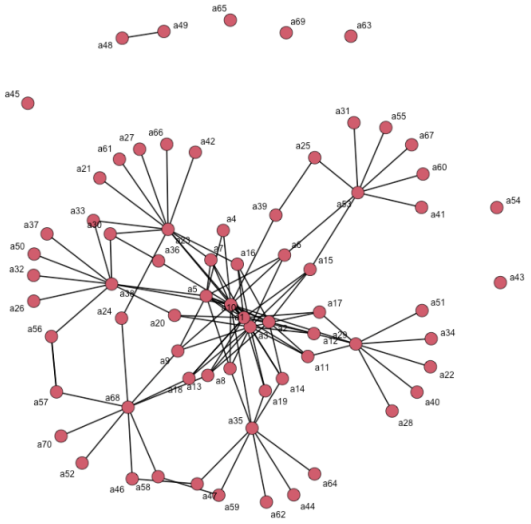
II. Component Size & Distribution

Within the advising relationship, one initial assumption is that coworkers with higher socially significant attributes are more likely to mentor/advise those with lower attribute values, such as seniority, age, and status. For seniority, we construct employees with the same seniority level as a sociomatrix and convert into a relative seniority difference matrix. For age, we might use absolute difference in age; however, considering the diversity and company policy, we decided to focus on seniority level only. Looking at the distribution of the directed advising network, the false output for is_connected command indicates it does not have a directed path from each vertex to all other vertices, hence our network is rather sparse.

As listed below, the membership indicates the cluster id to which each vertex belongs, whereas most employees are advice-seekers as in cluster 1 with 62 employees while a few other coworkers with higher seniority level are mentor represented in higher vector. Looking at the plotted largest component on the right, it can be inferred that most employees can receive advice/mentorship from coworkers with higher seniority level, either directly or indirectly, since the "weak" attribute also considers semi-paths. Upon investigation, there are 6 isolated employees who are enabled to get mentorship from others. On the other hand, based on the closeness centrality measures, employees including a48, a49, a1, and a10 require most steps to get connected to. In conclusion, while most employees are well-connected in the mentorship/advising system, some entry level employees might be afraid to reach out for help. Although the CEO of the company seems hard to get connected with at the first glance, she is closely connected with other executives and department leads, as shown in the *Figure.Advice Network Graph* below.

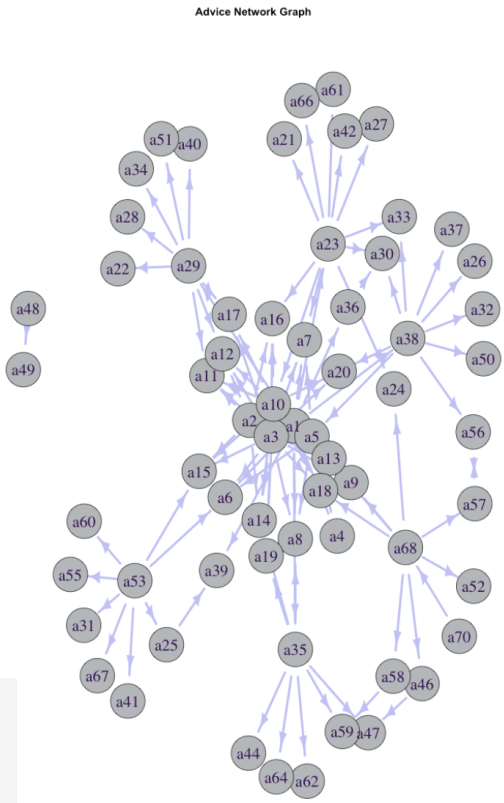
DSC 480 Network Analysis
Xuyang JI, Angelo Kelvakis, Mimidoo Gyoh

```
> library(igraph)
> ##convert into graph object
> adviceGraph <- graph_from_adjacency_matrix(sunglass_network$Dichotimized_advice,
+                                            mode = 'directed',
+                                            weighted = TRUE,
+                                            diag = FALSE)
> components(adviceGraph, mode = c('weak'))
$membership
 a1  a2  a3  a4  a5  a6  a7  a8  a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
a21 a22 a23 a24 a25 a26 a27 a28 a29 a30 a31 a32 a33 a34 a35 a36 a37 a38 a39 a40
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
a41 a42 a43 a44 a45 a46 a47 a48 a49 a50 a51 a52 a53 a54 a55 a56 a57 a58 a59 a60
  1   1   2   1   3   1   1   4   4   1   1   1   1   5   1   1   1   1   1   1
a61 a62 a63 a64 a65 a66 a67 a68 a69 a70
  1   1   6   1   7   1   1   1   8   1

$csize
[1] 62  1  1  2  1  1  1  1

$no
[1] 8

> lgc<- component.largest(adviceDN, connected="weak",
+                         result = 'graph',
+                         return.as.edgelist = FALSE)
> gplot(adviceDN,vertex.col=2+lgc)  #Plot with component membership
> #Plot largest component itself
> isolates(adviceDN)
[1] 43 45 54 63 65 69
> closeness(adviceGraph, mode='all')%>% sort(decreasing = TRUE)%>%.[1:4]
       a48         a49          a1         a10
1.000000000 1.000000000 0.008547009 0.007518797
```



The *Figure.Advice Network Graph* uses the Frucheterman-Reingold method models, representing the vertices as a collection of magnets and springs. As shown, employees with ID number from a1 to a5 are the executive/management team within the sunglass company, hence often play a role as advice-giver to department leads, while employees with higher ID number are most likely to be the advice-seeker, receiving mentorship/advices from their respective department leads.

To classify dyads in the directed graph, the relationships between each pair of vertices is measured, across three states: mutual, asymmetric or non-existent. Based on the output, there are 2 pairs with mutual connections, 112 pairs with asymmetric connections, and 2301 pairs with no connection between them. Hence, it could be concluded that mentorship and the advising systems are extremely exclusive with employees across various functional departments, where often the department leads give high-quality feedback and mentorships to employees within the same teams.




```
> ## Vignette #4 Dyads and Triags
> dyad.census(adviceGraph)
$mut
[1] 2

$asym
[1] 112

$null
[1] 2301

> triad.census(adviceGraph)
 [1] 47708  6234   119   466    55   101     7     9    40     0     0     1
[13]     0     0     0     0
```

VI. Positional Analysis in Network

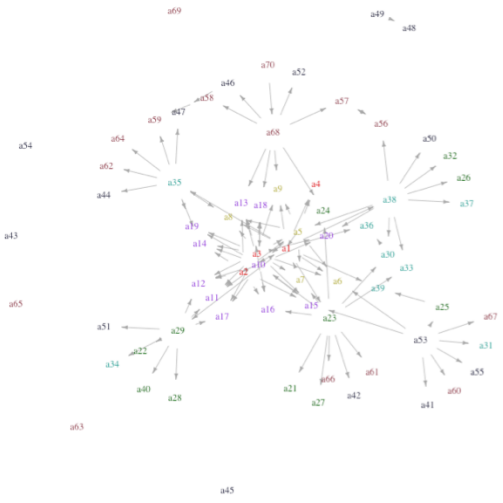
Positioning is the process of determining what positions an individual has in a social network (group or community). In social network analysis, there are certain strategies/tools we use to determine how individuals are positioned. This includes but not limited centralization, inbound (indegree) versus outbound (outdegree) connections and other similar measures. In positional analysis, nodes are grouped based on the similarity in their pattern of connection.

The positions can be seen as roles as in social settings, individuals in similar positions tend to perform similar functions/tasks within the network thus having a similar connection pattern. However, this is not always the case. The data used in this analysis is extracted from the directed advice network for the midterm project. For the simplicity of the plot, self-loop are omitted as it's more meaningful to focus on the direction of information flow. Given that positional analysis is based on identifying structurally equivalent nodes, there is a more generalized notion of structural equivalence.

The nodes are considered to share a position or role when they have generally similar patterns of relations, rather than exact ones i.e. the nodes don't have to connect to the exact same node but should share a similar structure. For example, a teacher who connects to 4 students and a principal is structurally equivalent to another teacher who also connects to 4 students and the principal even if they do not connect to the exact same students. There are many strategies for identifying roles in the networks which include, Brokerage Typology, Structural Equivalence, Isomorphic Local Graphs, Block Modeling with CONCOR, and Stochastic block modeling. A more cohesive and supportive advice network is generated with the name and position generator.

Baseline Network Plot – categorized by Department

In order to visually distinguish the departments, the network is plotted with vertex labels colored according to their respective departments, as shown below. The attribute dataset is derived by selecting specific rows from the complete attributes.csv dataset, based on matching IDs with the names of vertices in the network. Furthermore, it is reordered to match the node order within the network. This process generates the 'eAttr' dataset, which exclusively includes attributes relevant to the network nodes. The 'Department' attribute of each network node is then assigned the corresponding values from 'eAttr'.

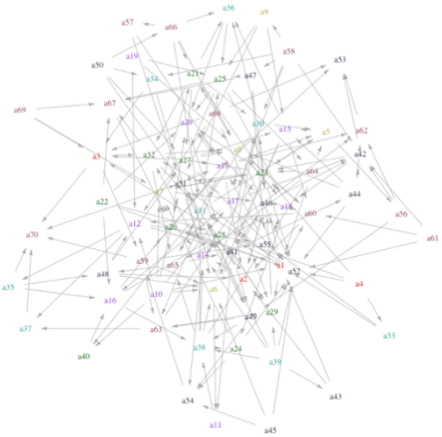


It is observed that the executive members occupy a central position within the network, indicating their influential role in the overall structure. This centrality

suggests that executive members are likely to play a crucial role in facilitating information flow and decision-making processes within the organization. On the other hand, department managers tend to play the role of advice-givers, providing guidance and support to employees within their respective departments. This pattern suggests that department managers are more directly involved in fostering communication and sharing knowledge within their specific teams. These findings highlight the distinct roles and positions of executive members and department managers, shedding light on the hierarchical dynamics and communication patterns within the organization.

Random Network Generated by Name & Position Generator

Using randomly ordered name generators, the effect of name generators' relative position on the likelihood of respondents' satisficing in their response is tested. The goal is to increase the probability of edges in the random graph by modifying the probability vector generated by the runif function.



To enhance visual differentiation between departments, the network visualization incorporates vertex labels colored according to their respective departments. This color-coded representation provides a clear visual distinction and aids in identifying different organizational units. Finally, the Department' attribute of each network node is assigned the corresponding values extracted from 'eAttr', serving as the true class and enabling accurate association of departments with
their respective nodes.

Algorithm 1. Brokerage

Unlike Burt's measure, the Gould-Fernandez measure below uses the department attribute to measure brokerage roles, sample output is shown below. Based on the output, node 'a1' has no incoming or outgoing ties and is not involved in any brokerage relationships. Therefore, it does not fill any of the mentioned roles (Coordinator, Itinerant Broker/Consultant, Representative, Gatekeeper, or Liaison). It serves as a standalone node within the network, in our case, the CEO of the company.

Node 'a2' has no coordination brokerage ties weighted by advice, 1 itinerant brokerage tie, initiated no gatekeeper brokerage ties, 3 representative brokerage ties and 2 liason brokerage ties. This individual is involved in 6 brokerage ties in total.

```
> #Vignette 1. Brokerage roles&position
> bNet <- as.network(adviceD, loops =TRUE, multiple=TRUE, directed=TRUE)
> bNet %v% 'department' <- as.character(attributes$Department)
> adviceBro<-  brokerage(bNet, 'department')
> head(adviceBro$raw.nli)
   w_I w_O b_IO b_OI b_O  t
a1   0   0    0    0   0  0
a2   0   1    0    3   2  6
a3   0   1    0    0   9 10
a4   0   0    0    0   0  0
a5   0   4    0    0   8 12
a6   0   0    0    0   0  0
```

Fig 2.0 Brokerage (Original)

Node 'a3' on the other hand, has no coordination brokerage ties weighted by advice, 1 itinerant brokerage tie, initiated no gatekeeper brokerage ties, no representative brokerage ties and 9 liason brokerage ties. This individual is involved in 10 brokerage ties in total.

For the randomized data, Node 'a10' has no coordination brokerage ties weighted by advice, 1 itinerant brokerage tie, initiated no gatekeeper brokerage ties, no representative brokerage ties and 10 liason brokerage ties. This individual involved in 11 brokerage ties in total

```
> bNet <- as.network(random[,1:2], loops =TRUE, multiple=TRUE, directed=TRUE)
> bNet %v% 'department' <- as.character(eAttr$Department)
> adviceBro<-  brokerage(bNet, 'department')
> head(adviceBro$raw.nli)
    w_I w_O b_IO b_OI b_O  t
a1    0   0    0    0   0  0
a10   0   1    0    0  10 11
a11   0   2    0    0  17 19
a12   0   0    0    0   3  3
a13   0   0    0    0   6  6
a14   0   1    0    0   5  6
```

Fig 2.1 Brokerage (Randomized)

Node 'a11' on the other hand, has no coordination brokerage ties weighted by advice, 2 itinerant brokerage ties, initiated no gatekeeper brokerage ties, no representative brokerage ties and 17 liason brokerage ties. This individual involved in 10 brokerage ties in total. The output provides insights into the brokerage roles and positions within the random network, shedding light on the patterns interdepartmental connections and the influence of specific nodes in facilitating the flow of communication and collaboration within and between departments.

The other nodes can be explained in a similar manner by considering their respective values for `w_I`, `w_O`, `b_IO`, `b_OI`, `b_O`, and `t`. We can utilize these nodes to gain a clearer understanding of the specific roles and positions that each node fulfills within the network.

Algorithm 2. Structural Equivalence

In the concept of structural equivalence, two nodes are considered structurally equivalent when they share identical relationship patterns with all other nodes in the network. This can be assessed by comparing the sets and patterns of neighbors for each node or by calculating the absolute difference between the row values of a matrix which represents the relationships of the two nodes. When the absolute difference between nodes is 0, those nodes are considered to be structurally equivalent. By examining these criteria, we can determine whether two nodes exhibit structurally equivalent properties in the network.

For example, for nodes ‘a1’ and ‘a2’, the sum of their absolute differences is 18, This means that they are not structurally equivalent.

```
> a_row <- adj_mat["a1", !colnames(adj_mat) %in% c("a1", "a2")]
> b_row <- adj_mat["a2", !colnames(adj_mat) %in% c("a1", "a2")]
> sum(abs(a_row-b_row))
[1] 18
```

Fig 3.0 Structural Equivalence

For the randomized data, the sum of the absolute differences of nodes a1 and a2 is 5. They are not structurally equivalent but are perceived to be more similar than in the original data.

```
> a_row <- adj_mat["a1", !colnames(adj_mat) %in% c("a1", "a2")]
> b_row <- adj_mat["a2", !colnames(adj_mat) %in% c("a1", "a2")]
> sum(abs(a_row-b_row))
[1] 5
>
```

Fig 3.1 Structural Equivalence

By extrapolating every pair of nodes in the advice network then convert into a similarity matrix, sets of similar actors are identified using K-Means Clustering algorithm. The following plots show the actors who are proximately similar when set the number of clusters equal to 4 and 5.

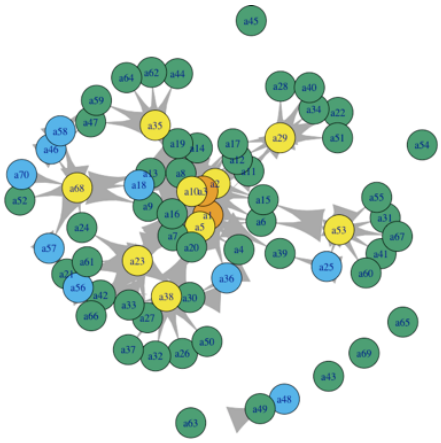


Fig 3.3 Cluster of 5

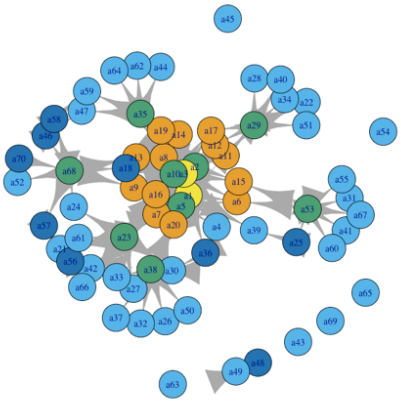


Fig 3.2 Clusters of 4

The output of the randomized data is below:

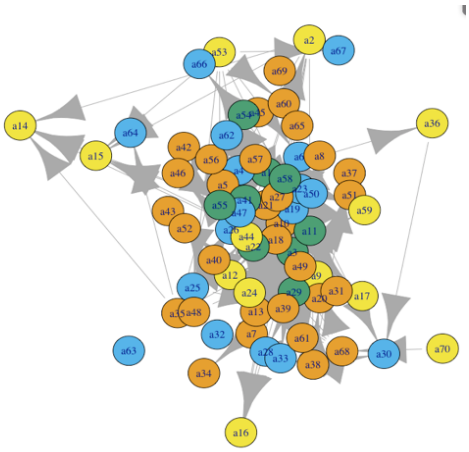


Fig 3.4 Clusters of 4

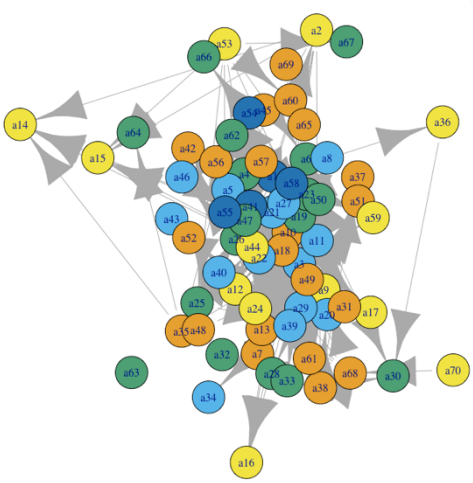


Fig 3.5 Clusters of 5

Recall that a2 represents an executive member of the sunglass company grouped together as occupying the same positions as the employees when set the number of clusters equals to 5 in the original data The reason for this phenomenon is that there is a strict definition of structural equivalence which relies on comparing the precise set of actors that each node is connected to. As a result, this method confuses similarity with closeness. This means that some nodes that we deem structurally inequivalent may be regarded as similar in actuality. On the other hand, the cluster of 4 makes more logical sense as it groups most executive and management members into the same clusters.

Algorithm 3. Block Modeling with CONCOR

The CONCOR algorithm operates on the basis of a similar concept of equivalence as described in structural equivalence. This strategy identifies structurally equivalent nodes based on the patterns in their connections or relationships. In addition to that, CONCOR takes advantage of correlation and stacks matrices which is used to create a block model that incorporates multiple relationships at the same time.

Based on the output, there are some entry and medium-level employees are classified as the same as those in the executive groups. The main aim of the CONCOR strategy is converging the data to only -1s and 1s. This is done by repeatedly running correlation on the results of the initial correlation, the data will eventually converge to only -1s and 1s.

The output of the final blocks is 4 clusters as shown below.

```
> split_results_corred <- lapply(split_results, cor_many_times)
> groups_2 <- lapply(split_results_corred, function(x) x[, 1] > 0)
>
> split_results_again <- lapply(groups_2,
+                               function(x) list(adj_mat[, names(x[x])], adj_mat[, names(x[!x])]))
> split_results_again <- unlist(split_results_again, recursive = F)
>
> final_blocks <- lapply(split_results_again, colnames)
> final_blocks
[[1]]
 [1] "a1"  "a22" "a25" "a28" "a31" "a34" "a40" "a41" "a43" "a45" "a48" "a49" "a51" "a53" "a54" "a55" "a60"
[18] "a63" "a65" "a67" "a68" "a69" "a70"

[[2]]
 [1] "a21" "a24" "a26" "a27" "a30" "a32" "a33" "a37" "a42" "a46" "a50" "a52" "a56" "a57" "a58" "a61" "a66"

[[3]]
 [1] "a2"  "a3"  "a4"  "a5"  "a6"  "a7"  "a8"  "a9"  "a10" "a23" "a29" "a35" "a36" "a38" "a39"

[[4]]
 [1] "a11" "a12" "a13" "a14" "a15" "a16" "a17" "a18" "a19" "a20" "a44" "a47" "a59" "a62" "a64"
```

Fig 4.2 Original Data Output (Final Blocks)

```
[[1]]
 [1] "a70" "a67" "a64" "a44" "a36" "a69" "a40" "a24" "a23" "a11" "a29" "a22" "a48" "a33" "a41"

[[2]]
 [1] "a21" "a51" "a63" "a61" "a20" "a59" "a54" "a10" "a2"  "a43" "a8"  "a6"  "a17" "a35" "a3"

[[3]]
 [1] "a16" "a47" "a68" "a65" "a52" "a49" "a30" "a19" "a53" "a27" "a37" "a56" "a12" "a45"

[[4]]
 [1] "a26" "a66" "a62" "a39" "a60" "a58" "a13" "a57" "a50" "a55" "a14" "a28" "a38" "a5"  "a15" "a25" "a46"
[18] "a31" "a42" "a34" "a32" "a18" "a9"  "a7"  "a4"  "a1"
```

Fig 4.3 Randomized data output (Final Blocks)

This is then plotted and the nodes are colored by positions. Based on the output, there are some entry and medium-level employees are classified as the same as those in the executive groups.

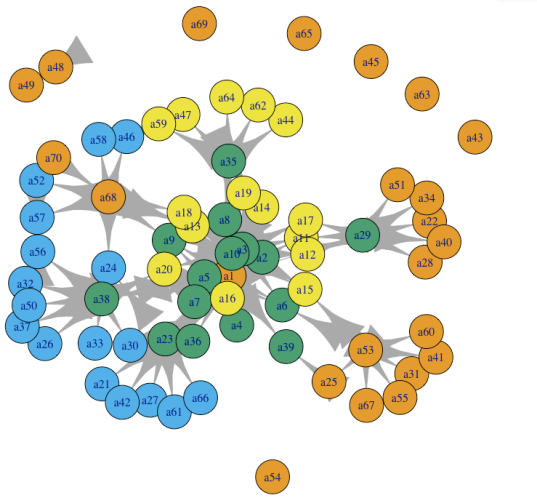


Fig 4.5 Original Data Output

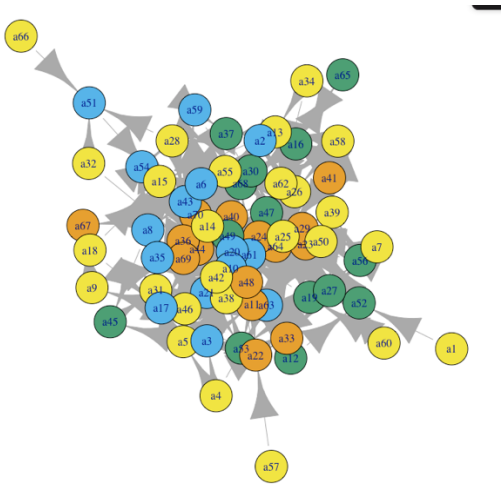


Fig 4.6 Randomized Data Output

Algorithm 4. Isomorphic Local Graphs

To identify the individuals who are precise structural equivalent, isomorphic is used. This method works by relaxing the condition that requires the nodes be tied to precisely the same set of nodes by defining them as structurally equivalent as long as their local neighborhoods are automorphic. The underlying algorithm, bliss, essentially permutes the matrices of the two networks it is comparing to see if, under any of the different permutations, the two matrices are equivalent. Isomorphic local graph analysis tries to identify similar sub-graphs within a network (For example, the cliques within a school). These subgraphs are smaller sections of the network that exhibit identical structural characteristics.

The neighborhood size was set to 3 and 4, and looped through the different local neighborhoods to evaluate whether they are automorphic, the output plot shows nodes who share isomorphic neighborhoods. Using 4 clustering which seems to be more logical.

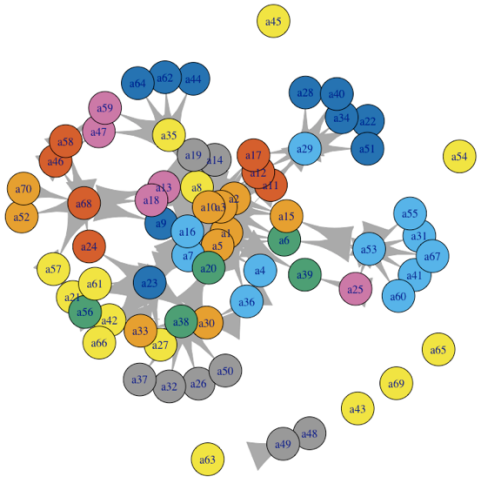


Fig 4.7.1 Neighborhood size = 3

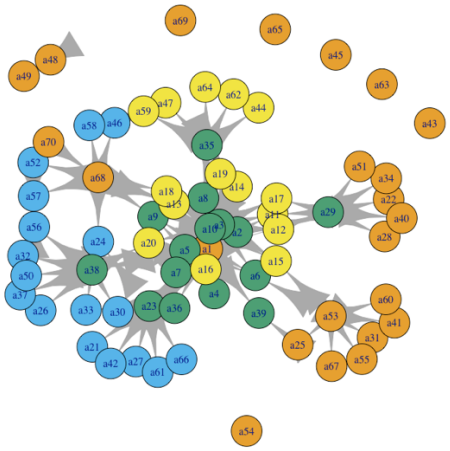


Fig 4.7.2 Neighborhood size = 4

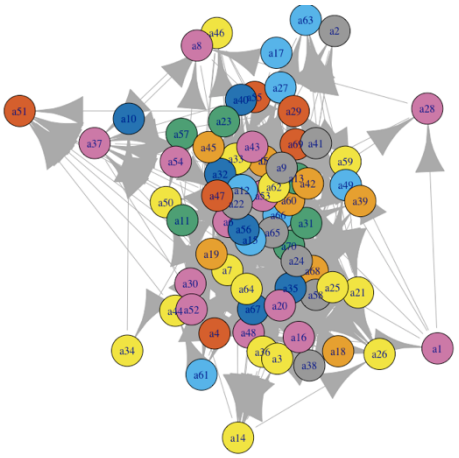


Fig 4.8.1 Neighborhood size = 3

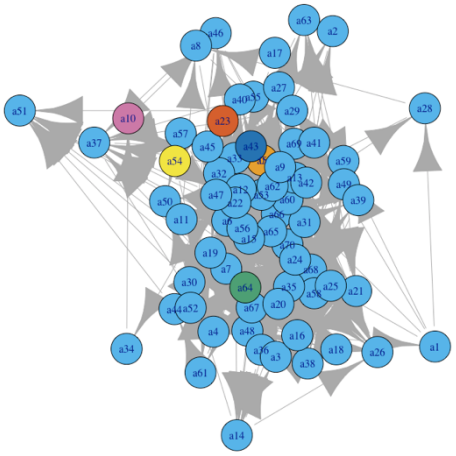


Fig 4.8.2 Neighborhood size = 4

Algorithm 5. Stochastic Block Models

In a Stochastic block modeling, the network is divided into a predefined number of blocks. Each node is assigned to one of the blocks. The main objective of this is to uncover the underlying block structure and the connections between them. The connections within a block are typically denser compared to connections between blocks. Instead of being dependents on the individuals that share the same set of nodes, the individuals who share a role or position will have the same probability of being attached to all other alters in the network. With the use of this method, equivalence is not absolute, but based on probability.

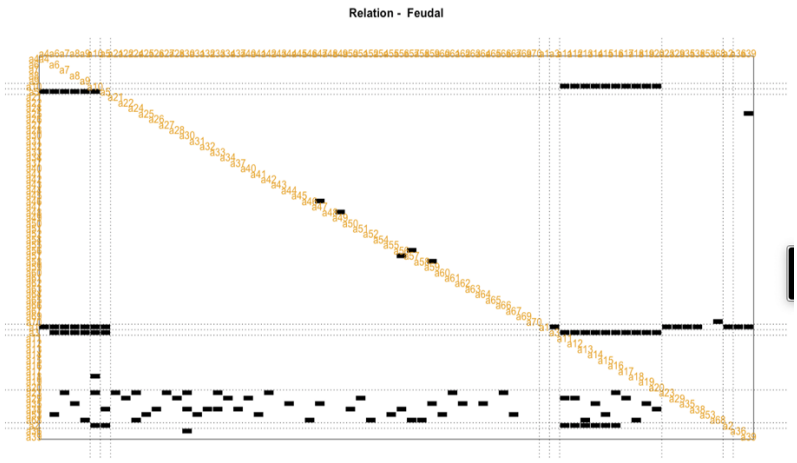


Fig 5.0 Stochastic Block Model

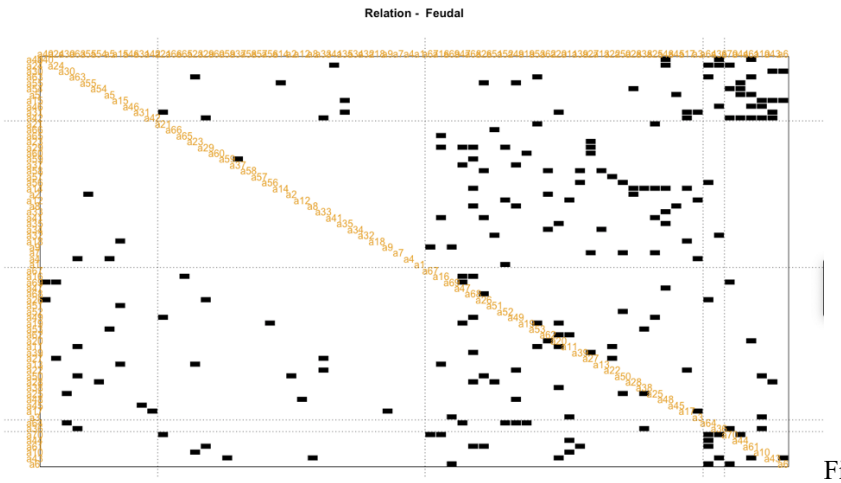


Fig 5.1 Randomized data output

Based on the output of true (original) scenarios, it can be concluded that the community structure captured by the model accurately represents the underlying social dynamics in the advice network. The output below shows the SBMs output using random network, which serves as a benchmark for validation. By analyzing SBMs on random networks with known properties, we can assess the performance and effectiveness of the SBM inference methods. In this case, the

SBM failed to accurately recovers the block structure and connection probabilities in the random network, it demonstrates the model's validity and the inefficacy of the inference technique.

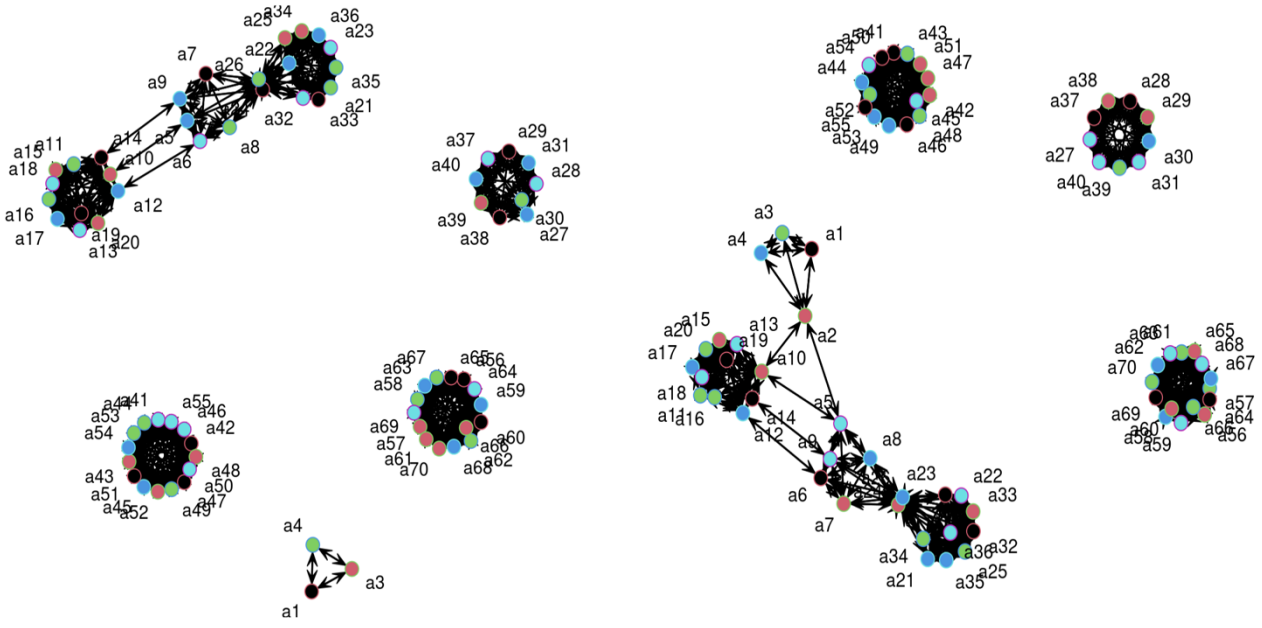
VII. Subgroups and Communities Detection

I. Cutpoints And Bridges

Cutpoints in a graph refer to vertices whose removal would increase the number of components in the network. They are significant when analyzing the flow of a network, as their removal affects the connectivity of the graph. Cutpoints often occupy crucial positions, connecting different parts of the network. In the case of the undirected working network, a weak component rule is applied to identify cutpoints, given the network's low density.

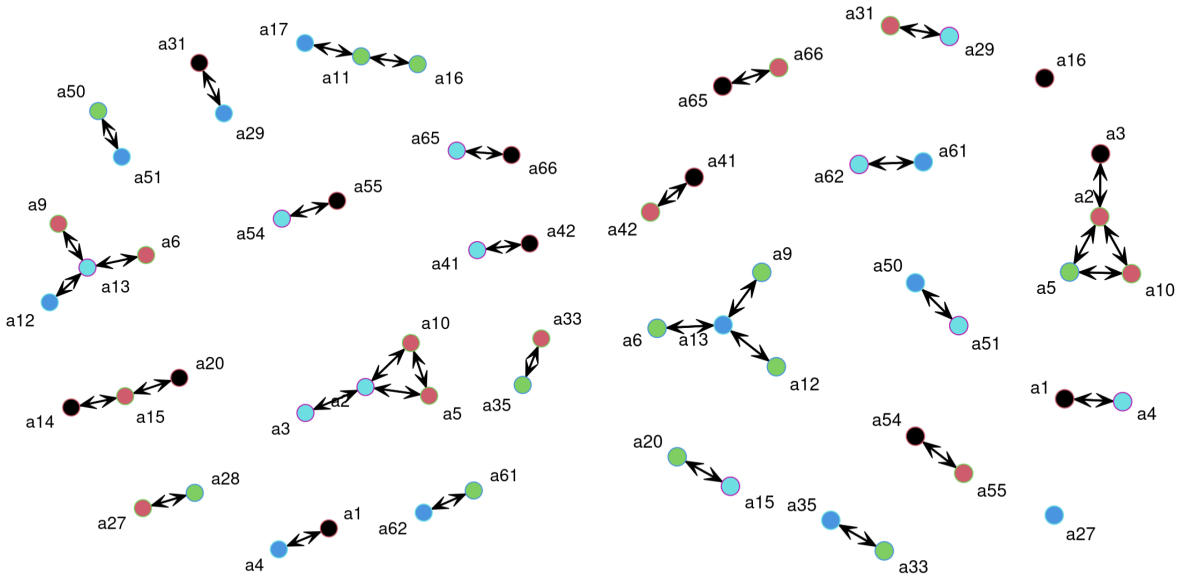
Working Network

The Working network has only one cutpoint as it has clusters of highly connected nodes. The figures above show how the removal of the cutpoint ‘a2’ leads to the isolation of the a1-a4-a3 subgroup.



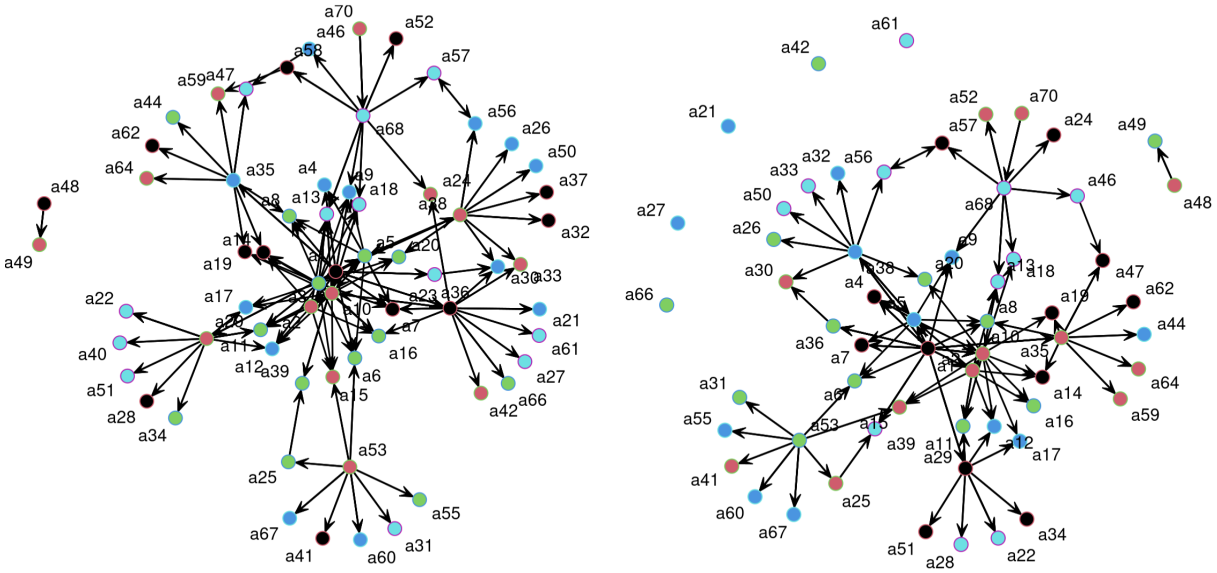
DSC 480 Network Analysis
Xuyang JI, Angelo Kelvakis, Mimidoo Gyoh

Friendship Network



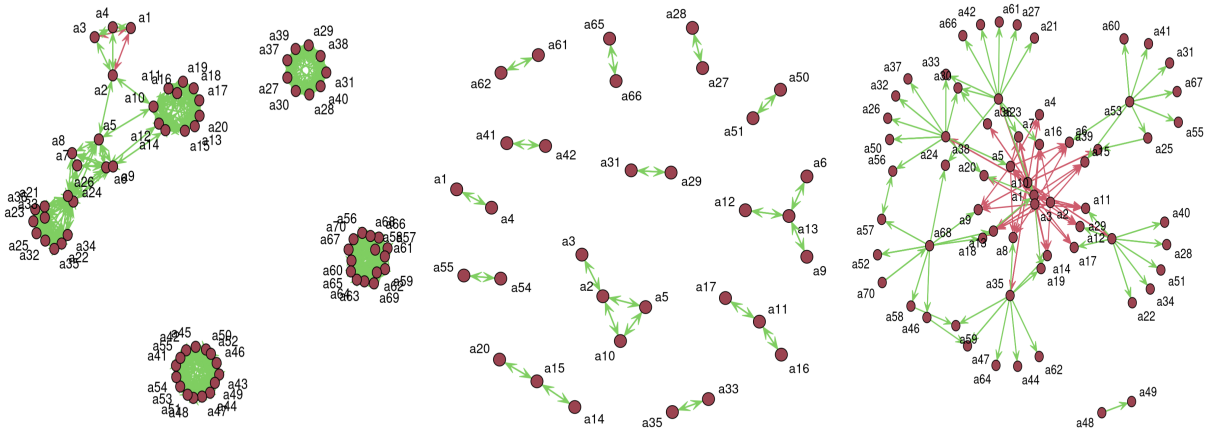
With the Friendship network, there are far more dyads to be broken up into isolates. In this network, there were 4 cutpoints identified and removed. Because this network is even less dense than the working network, the cutpoints do not show a significant change in the overall structure, however, it can be seen how some triads are broken up leading to identification of brokerages between friendships.

Advice Network



The Advice network has a far more distributed network compared to the Working and Friendship networks, leading to a more drastic change in the structure of the network when removing cutpoints. Six cutpoints were removed from the network and resulted in the outermost nodes becoming isolated.

Network Bridges



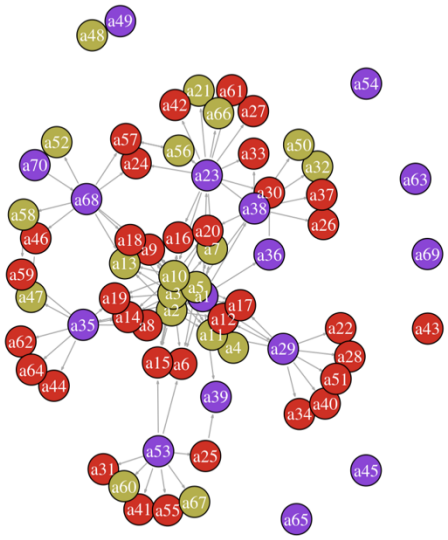
The bridges of the network serve as the cutpoints of the edges. They showcase the connections between nodes that hold the most structure within the network. The Working network shows bridges to the removed cutpoint that was removed. The Friendship network shows no bridges as the network is not dense and has low connectivity. The Advice network shows several bridges in the center of the network with the employees giving the most advice having the highest number of edges.

II. Clique Analysis

A clique in a network is a maximally complete subgraph where every node is directly connected to every other node within the subgraph. It represents a subset of nodes that have all possible ties among them. In this network analysis, the vertices represent employees, and in order to visually distinguish them based on their attributes, the network is plotted with vertex labels colored according to their respective attributes.

Advice Network: Seniority Attribute vs Clique Analysis

The Advice network has data on which employees share advice with other employees. In order to analyze the accuracy of the clique identification, the graph has been coded by the attribute Seniority, which ranges from 0-2 and indicates how the employee is positioned in the company with 0 being entry level (red) , 1 being Junior level (gold), and 2 being senior level (purple). Seeing as the mentorship program provides employees with lower seniority a mentor with higher seniority, it should be a strong predictor of cliques.



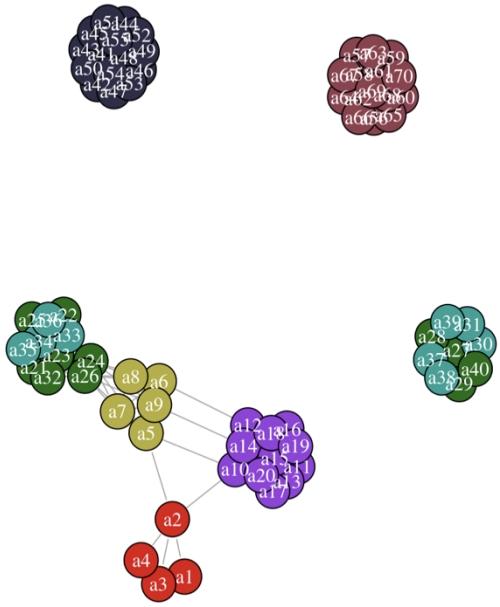
When performing clique analysis, the top three cliques were the following:

1. a10 a1 a5 a3
2. a23 a1 a10
3. a7 a1 a23

The highest ranked clique contains the members at the center of the graph which have top seniority. This is true for the other two top cliques, and the resulting schema of the network where senior level employees often mentor junior employees, who in turn mentor entry level employees.

Working Network: Department Attribute vs Clique Analysis

Similar to the selection of attribute for analysis, the working network cliques were compared to the Department attribute which makes logical sense as most people work mainly within their own department. The graph shows all 7 departments: Executive (red), Marketing (gold), Sales (purple), Human Resources (black), Distribution (brown), Manufacturing (green), and Finance (blue).



The clique analysis showed the following cliques:

1. a42 a41 a55 a54 a53 a52 a51 a50 a49 a48 a47 a46 a45 a44 a43
2. a70 a56 a69 a68 a67 a66 a65 a64 a63 a62 a61 a60 a59 a58 a57
3. a10 a11 a20 a19 a18 a17 a16 a15 a14 a13 a12
4. a32 a21 a36 a35 a34 a33 a26 a25 a24 a23 a22
5. a31 a27 a40 a39 a38 a37 a30 a29 a28
6. a6 a12
7. a9 a14
8. a7 a5 a26 a24 a9 a8 a6
9. a2 a5 a10
10. a1 a2 a4 a3

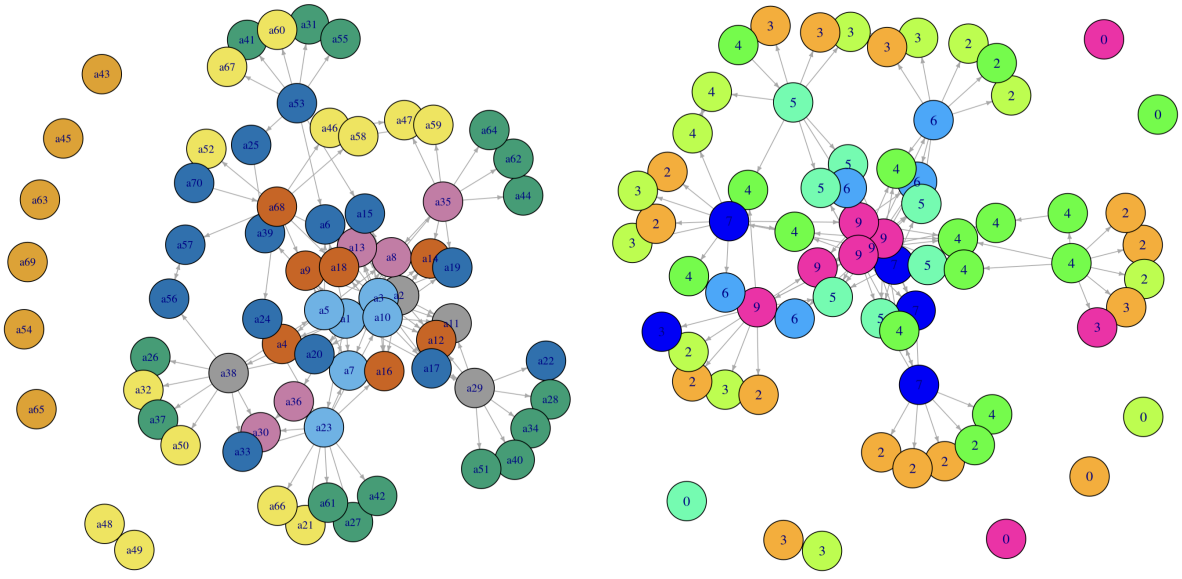
These cliques show a strong correlation between the Department attributes and the clique formation. Outside of a few non-department connections, the clique groupings match the departments. It can be seen that there are inter-department liaisons that transfer work, specifically between Marketing and Sales, and between Marketing and Manufacturing. These relationships are shown by the cliques 6, 7, and parts of 8.

Due to the lack of interconnectedness of the friendship network, no clique analysis was performed.

III. K-Core Analysis

In network analysis, the k-core is a variation that addresses the rarity of cliques in observed social networks. A k-core is a maximal subgraph where each vertex is connected to at least k other vertices within the subgraph. Unlike cliques, k-cores offer several advantages: they are nested (each member of a higher k-core is also a member of a lower k-core), they do not overlap, and they are easy to identify. To understand the k-core structure in the network, the graph's density is calculated, representing the ratio of the actual number of edges to the largest possible number of edges in the graph, assuming no multiedges are present. The density value provides an indication of how interconnected the network is. The graph.coreness function is then utilized to identify the k-core structure in the network. It returns a vector that lists the highest core each vertex belongs to in the network.

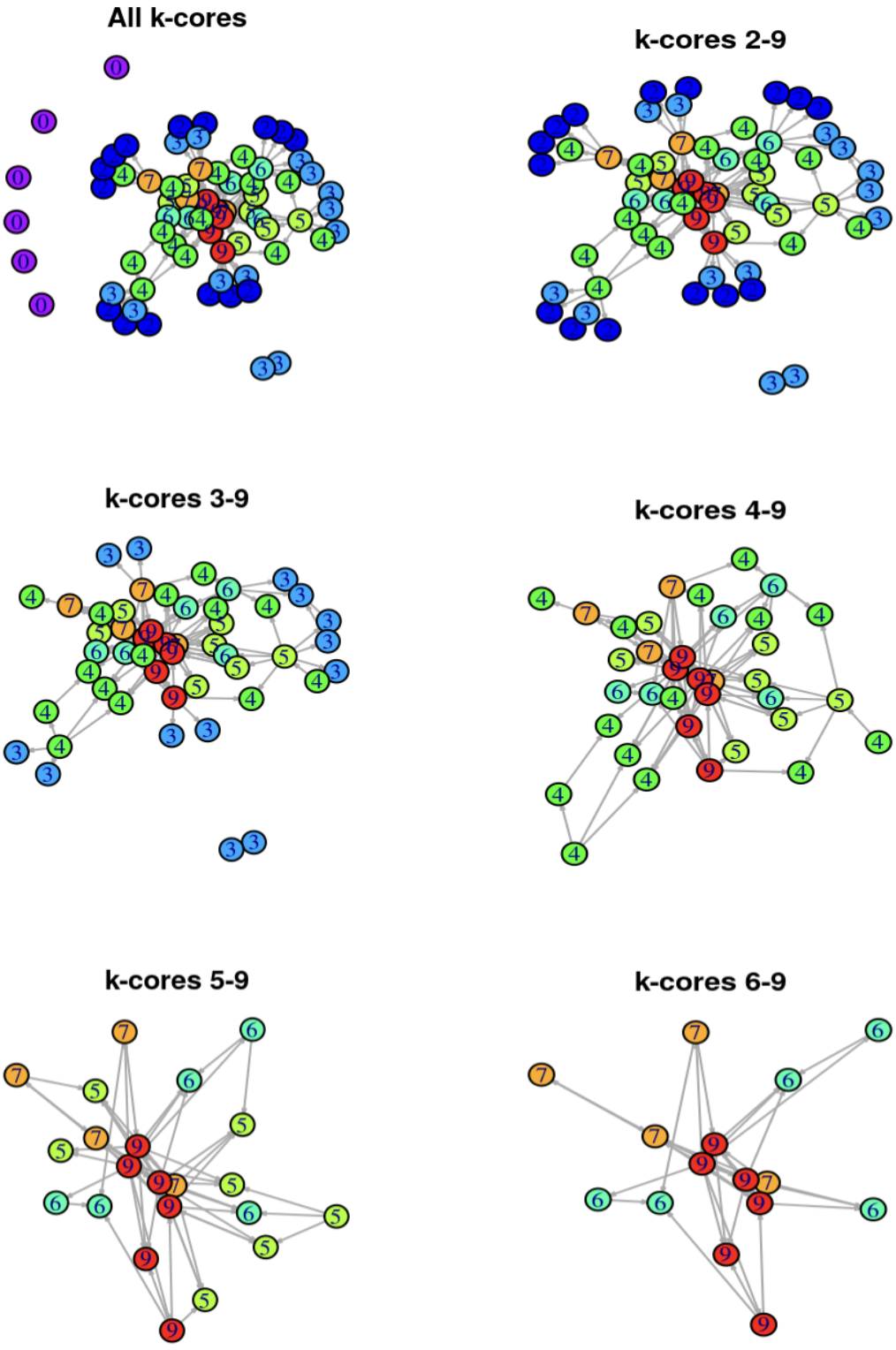
Advice Network



The graph on the left shows the Advice network, classified by the structure of the network, and the graph on the right shows the network, color-coded by coreness, where the k-core is labeled

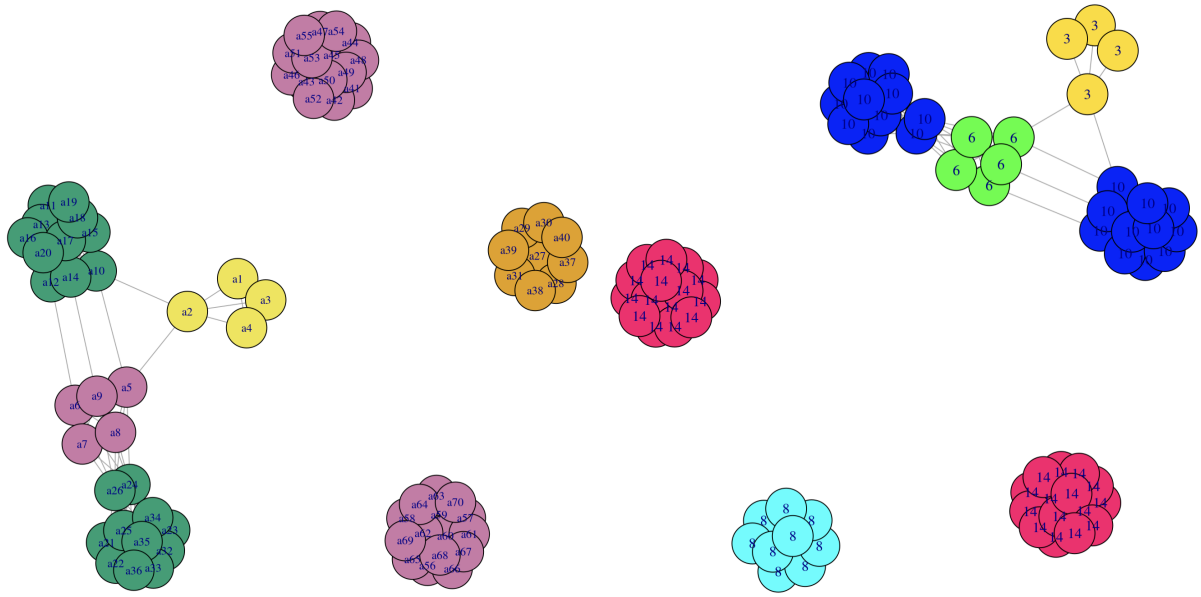
and shows how the center of the graph is more highly connected, compared to the edges. The density of this network is only 0.051, with 9 cores identified.

Advice Network k-core Decomposition



The graphs above show the elimination process of the Advice network cores. The graphs display only the removal of the first 6 cores, as the rest of the decomposition shows no new information about the structure of the network. The breakdown of the network strips off the outer nodes of the network revealing the highly connected nodes at the center.
The nodes with the highest level of connectivity are the mentors of the company and show how the advice flows from the center-most nodes out to the entry level employees.

Working Network



The Working network has clusters of highly connected subgroups that have a density of 0.160 and 14 cores. The subgroups are so tightly packed that analyzing their core decomposition would not yield any specific knowledge about the working subgroups of the company. This makes logical sense for the analysis of the working network because, unlike advice which spans over several departments, the working network is more contained to isolated subgroups.

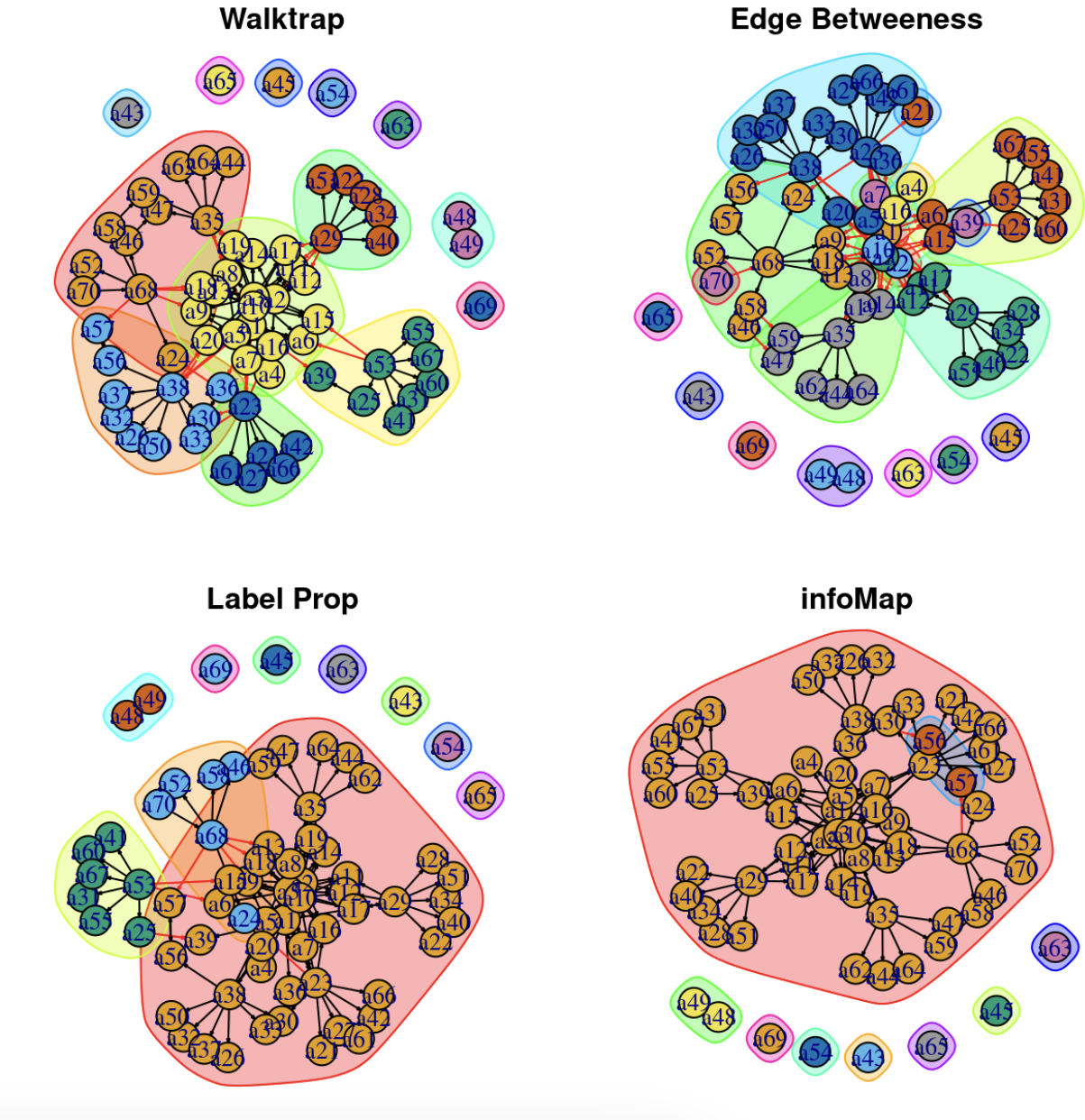
IV. Modularity And Community Detection

Modularity is an essential characteristic of networks that plays a significant role in many community detection algorithms. It measures the structural properties of a network, specifically focusing on the degree of clustering observed within groups of nodes and the density of connections between these groups. Modularity provides a chance-corrected statistic that helps quantify the extent to which nodes form clusters that are denser internally and sparser externally.

Advice Network: Seniority

When analyzing the Advice network's modularity, four different algorithms were used: Cluster Walktrap, Edge Betweenness, Label Prop, and Infomap. Each algorithm produced a different

modularity score: 0.02144107, 0.4510999, 0.3336058, and 0.1776159. Due to the low value (not close to 1) it can be concluded that across multiple algorithms this network, with respect to the seniority attribute exhibits low clustering. This means the interconnectedness of the network obscures any clustering that would be explainable by the attribute of seniority. This is counter to the logical assessment of the network where levels of seniority would depict network substructures due to how the mentorship program was run.



These graphs show clustering performed by the algorithms on the network in regards to the seniority attribute. The Edge Betweenness graph had the highest modularity score, but still does not capture the majority of the seniority subgroupings. Both the Walktrap and the Edge Betweenness algorithms pick up on the branching behavior of the network, with entire branches

of the Advice network being clustered together. This would make sense as it defines how the advice flows from the center of the graph, out to a sub-layer, and then finally being distributed to the outer edge of employees. The Label Prop and infoMap algorithms seem to only capture the isolates on the edge, and the center-most cluster of nodes.

Overall, when looking at the Advice, Working, and Friendship networks, there are clear characteristics of the structure of the networks, with a strong story being told by the advice network in relation to the seniority attribute. While there is not enough numeric evidence to support clear subgroups, the clique analysis showed how seniority was a strong predictor of clique formation, with cliques being formed around senior-level employees. The k-core analysis added more weight to the number of edges of the network, showing how the center of the network held the most connections, radiating out to spread advice to the outer nodes. The Working network had a clear and transparent set of subgroups, explained almost entirely by the department that the employees worked in. The interconnectedness of these groups was shown by the cutpoints and bridges, where liaisons in the executive department worked with marketing and sales. Lastly, the weakest analysis was drawn from the friendship network, which lacked the interconnectivity to draw numerical conclusions about the subgroups of the network.

The subgroups of this company are strongest within the working network, and weakest within the friendship network. In order to strengthen the understanding of the communities within the company, further data on their attributes should be collected in order to understand how their work, advice, and friendship can be classified into groups. There is a clear distinction between department work, and a lack of friendship groups. A cross-team sport, or project may help to strengthen these subgroups and push deeper interactions within the company. These kinds of interactions are important to strengthening the resilience of the company by creating stronger sub-teams that may be better at things like problem solving, creative ideation, and building rapport among co-workers.