Formatting Data for One and Two Mode Undirected Social Network Analysis

Practical Assessment, Research & Evaluation

A peer-reviewed electronic journal.

Volume 26 Number 24, November 2021 ISSN 1531-7714

Formatting Data for One and Two Mode Undirected Social Network Analysis

Austin T. Boyd, *University of Tennessee, Knoxville*
Louis M. Rocconi, *University of Tennessee, Knoxville*

Social Network Analysis (SNA) is a statistical method used to analyze the social structure and interactions among individuals within a network. SNA is used extensively in a number of disciplines such as sociology, geography, and communications research. However, the use of SNA by practitioners and researchers in assessment and evaluation is much lower than their counterparts in other social science disciplines. One of the primary barriers to utilizing SNA in social science research is correctly formatting the data for use. The focus of this article is to provide researchers with a tool for restructuring long form data so that it can be used to conduct social network analyses and generate undirected sociograms.

Introduction

Social network analysis has become an increasingly popular technique in the social and behavioral sciences. Social network analysis (SNA), also known as organizational network analysis or simply network analysis, is a method of observing social structure in terms of relationships between social objects through visual representations (Tichy & Fombrun, 1979). SNA has been used to study criminal behavior (McGloin, 2005), the spread of sexually transmitted diseases (Youm & Laumann, 2002), online collaborative learning (Saqr et al., 2018), and COVID-19 transmissions (Nagarajan et al., 2020). SNA is also a valuable tool for assessment and evaluation professionals. SNA can help evaluators understand the network embedded within a program, identify gatekeepers and cliques within a network, and identify who has access to whom and to what resources. Educational researchers and assessment professionals can use SNA to understand how relationships form in schools and classrooms, examine attendance at professional training opportunities, or explore how interactions within a network shape educational outcomes. Despite SNA's wide use, much of the data collected for assessment and evaluation research do not fit the format required by most software programs that conduct social network analysis, and reformatting the data to fit this form can be a daunting task.

Restructuring data is frequently required when performing repeated-measures analyses and multivariate analysis of variance or when switching between different software programs (Long & Teetor, 2019). For example, to analyze repeated-measures data using a linear mixed model, data must be pivoted from wide form to long form before the analysis can begin. Wide form data, which is also referred to as unstacked data, is presented as a table with separate columns for each measured variable and separate rows for each unique individual or case. Survey data obtained through online survey platforms such as Qualtrics, QuestionPro, or Survey Monkey are collected and stored in this format. Long form data, also referred to

as narrow, stacked, or tall data, contains all of the same information as wide form, however instead of using separate columns for each measured variable, one column is used to list the measured values while another is used to indicate the variable being measured. This is done for each case in the data, resulting in a data table with a length equal to the number of cases times the number of measured variables. Long form data are common in SQL-based database systems (van der Loo, 2021); however, most data collected for assessment and evaluation purposes are usually collected in wide format and will need to be transformed into long format. For SNA in assessment and evaluation research, the issue that arises is how to transform long form data into an adjacency matrix, which is the typical data structure for SNA software programs (i.e., programs that conduct SNA analyses).

This article addresses the issue of formatting data for one and two mode undirected SNAs. We focus on undirected SNAs for two reasons. First, data that are capable of being used for directed SNAs by nature already exist in a format accepted by most SNA software programs. This is because the data would indicate source and target nodes, such as in an edge table or adjacency list. Second, undirected SNAs are often based on data in which a common aspect is shared between individuals. In these situations, directionality cannot be determined, and the data are usually in a format more akin to survey data (i.e., wide format). This article addresses this issue by providing researchers with a tool and instructions to reformat long form data into a format accepted by most SNA software programs. In order to reformat the data, we created a function in R which we call snafu, which takes long form data and reformats it into adjacency and affiliation matrices. In addition to reformatting the data, the function exports the newly created matrices to new .csv files, preserving the original data. The new files can be imported into any software that can be used for SNA, such as Gephi (Bastian, Heymann, & Jacomy, 2009), R (e.g., igraph, sna), or UCINET (Borgatti et al., 2002), and used to conduct the analysis.

Social Network Analysis

As a brief primer on SNA, we will discuss a few key aspects of SNA that are important to understand in order to prepare the data properly for analysis. For a more detailed explanation on conducting and interpreting an SNA, see Yang, Keller, and Zheng (2017) or Aggarwal (2011). The primary output of an SNA is the visual diagram that represents the network, known as a sociogram. Within each sociogram are two main parts, the nodes and edges. The nodes are discrete individuals or collective social units within the network (e.g., groups, events, places) (Carolan, 2014). These nodes, usually represented by circles in a sociogram, are what researchers are interested in understanding the relationships between. The edges are what connect one node to another. These are the lines drawn between the nodes which indicate a connection based on some common factor. Common factors can be any characteristic shared by the nodes that are being used to group them, such as a class they are enrolled in or an event that they attended.

Sociograms can be directed, undirected, or a mixture of the two. Directed sociograms are used to indicate the direction of the relationships that occur between pairs of nodes, or dyads, within the network. These relationships can be unidirectional, where one node acts as the sender, or source, while the other is the target, or receiver, or bidirectional in which the flow of information travels in both directions. An example of a unidirectional interaction in a network would be the path that a post is shared on social media or the transmission of a virus from one person to the next. A bidirectional interaction would be reciprocated friendships between any two individuals. Undirected sociograms are used when there is no way to determine the direction of the relationship, there is no direction of the relationship, or when all relationships in the network occur in both directions, such as friendships on Facebook or classes shared by students at a given school. In these undirected sociograms, neither node in a dyad would be considered the source or the target.

In addition to showing directionality, sociograms can also show the network using one mode or two mode matrices. One mode matrices, also known as adjacency matrices, are square matrices that only contain the nodes and the weight of the edges, in the form of frequency counts, that connect them. Using a one mode matrix creates a sociogram showing the direct relationship between any two nodes based on the number of shared events between them. One mode SNAs can answer research questions such as how people interact within a program or to identify factions within a group. Unlike one mode matrices, two mode matrices, also known as affiliation matrices, are not

square matrices. These matrices list both the nodes and the edges in one matrix, with all frequencies between them being one or zero. Using a two mode matrix creates a sociogram where the nodes are both the original nodes and also the edges which were connecting them in the one mode matrix sociogram. This allows the relationships between both the nodes and the events to be seen. In these sociograms, all edges have a weight of one and the connections will only occur between a node and an event, unlike in the one mode matrix where nodes are directly connected to each other. Two mode SNAs can answer research questions such as what programs people are most involved in and which sources people are getting information from.

Data Format

Many SNA software programs, such as Gephi, import data in four different formats: as a node and an edge table, an adjacency list, an adjacency matrix, or an affiliation matrix (Bastian, Heymann, & Jacomy, 2009). The node and edge table format (Figure 1a) uses two separate files, one containing a table of all the nodes in the network and another containing all of the edges between the nodes by showing the source and target nodes. In the adjacency list (Figure 1b), each row indicates a pathway of nodes through the network. The adjacency matrix form (Figure 1c) is a square matrix of the nodes, much like a correlation matrix of variables, that contains counts of the number of interactions between nodes. Finally, the affiliation matrix (Figure 1d) is a matrix of the nodes and the edges, where the nodes are the capital letters and the edges are the lowercase letters, that contains either a one or a zero indicating if there is a link between the node and the edge. See Figure 1 for examples of the four formats.

Figure 1. Data formats for SNA

a.

	A
1	Nodes
2	A
3	B
4	C
5	D

	A	B
1	Source	Target
2	A	B
3	A	B
4	A	C
5	B	C
6	B	D
7	C	D

b.

	A	B	C
1	A	B	C
2	A	B	
3	B	C	D
4	C	D	
5	C	D	
6	C	D	

c.

	A	B	C	D	E
1		A	B	C	D
2	A	0	2	1	0
3	B	2	0	1	1
4	C	1	1	0	3
5	D	0	1	3	0

d.

	A	B	C	D	E	F	G
1		a	b	c	d	e	f
2	A	1	1	0	0	0	0
3	B	1	1	1	0	0	0
4	C	1	0	0	1	1	1
5	D	0	0	1	1	1	1

Note: Figure 1 (a.) displays a node and an edge list, (b.) displays the same data as an adjacency list, (c.) as an adjacency matrix, and (d.) as an affiliation matrix.

Unfortunately, data are not usually collected in these formats. This is especially true when dealing with survey data, which is the most common way evaluation and assessment researchers collect data (Janus, 2016). Survey data is most commonly stored in wide form, where each case represents an individual participant and their responses for each of the survey items. To use the function presented in this article, the user should first transform the data from wide form data into long form data. Reformatting from wide to long data can easily be done in any software program, including R, SPSS, Stata, and SAS. The following line of R code uses the pivot_longer function included in the tidyverse package (Wickham et al., 2019) to transform data from wide to long form:

```
long <- wide_data %>% pivot_longer
(column1:column2, names_to =
"new_column_name", values_to =
"new_column_name")
```

For a more detailed explanation on transforming data from wide form to long form and vice versa, see Long & Teetor (2019), Wickham & Henry (2020), and Wickham (2007). Once the data have been transformed into long form, the snafu function can be used to transform the data into both an adjacency matrix and affiliation matrix as seen in Figure 1c. and 1d. above.

The `snafu` Function

The R function `snafu` takes long form data and reformats it into adjacency and affiliation matrix form. Not only does it reformat the data, but it also creates three separate matrices for the nodes and the edges, which allows the SNA to be conducted three ways: (1) to view how the individuals are linked by the events, (2) to view how the events are linked by the individuals, and (3) to view how the individuals and the events are linked to each other. In the first matrix, which we refer to as the "nodes" matrix, the row and column names indicate the individuals, and the cells indicate the number of events linking them. This matrix produces a sociogram where the individuals are the nodes and the events are the edges. The second matrix, which we refer to as the "edges" matrix, is the reverse, where the row and column names indicate the events, and the cells indicate the number of individuals linking them. This matrix produces a sociogram where the events are the nodes and the individuals are the edges. The third matrix, which we refer to as the "two mode" matrix, contains the original nodes in the first column and the edges along the first row, and the cells indicate whether or not there is a connection between them. This matrix produces a sociogram where both the events and individuals are the nodes. These matrices are exported as three separate .csv files.

The `snafu` function contains three arguments, two of which are required for the function to run. The first necessary argument, x, must be a vector (or column from a data frame) containing the cases from the long form dataset that will act as the nodes. The second necessary argument, y, is another vector (or column from a data frame) containing the events from the long form data that will provide the edges between the nodes. If either of these arguments is not specified, then the function will return an error indicating that one of the arguments is missing. The third argument, diagonal, is optional and determines whether to set the diagonal of the matrix to zero. This argument can be either TRUE keeping the diagonal as is, or FALSE, which sets the diagonal equal to zero. If left out of the function, the argument's default is FALSE, and the diagonal will be set to zero.

The body of the function contains ten statements, beginning with

```
M <- as.matrix(table(x, y))
```

which creates a matrix of length x by width y using the vectors provided. This will be saved as the two mode affiliation matrix. Once the x by y matrix has been created, the following two lines calculate the cross-product matrices of the nodes and edges, respectively. This results in two square matrices, one with dimensions x by x and the other y by y.

```
Nodes<-tcrossprod(M)
Edges<-crossprod(M)
```

Next, the function replaces the values in the lower triangle of both matrices with zero since the two matrices are symmetric about the diagonal and including both halves of the matrix would inflate the relationships in the analysis by a factor of two. While the relationship strengths would remain proportional to each other without removing the lower triangle of the matrix, their magnitude compared to non-existing relationships, or relationships with values equal to zero, would be doubled. At the same time, the function also uses the third argument to determine whether to replace the diagonals with zero as well. Values along the diagonal of the matrix indicate the node has a relationship with itself, creating self-links, also known as loops or self-loops. By replacing these values with zero, the function prevents any self-links from occurring when conducting the SNA.

```
if (diagonal == FALSE){
      Nodes[lower.tri(Nodes,diag =
      TRUE)]=0
      Edges[lower.tri(Edges,diag =
      TRUE)]=0
    } else {
      Nodes[lower.tri(Nodes,diag =
      FALSE)]=0
      Edges[lower.tri(Edges,diag =
      FALSE)]=0
    }
```

From here the matrices are ready to be saved to the global environment and exported to new .csv files. The function saves and exports the three matrices with the names Nodes, Edges, and TwoMode to the global environment and the user's current working directory.

```
      Nodes<<-Nodes
      Edges<<-Edges
      TwoMode<<-M
write.table(Nodes, file = "Nodes.csv",
na="", col.names=NA, sep=",")
write.table(Edges, file = "Edges.csv",
na="", col.names=NA, sep=",")
write.table(M, file = "TwoMode.csv",
na="", col.names=NA, sep=",")
```

Finally, the function ends by printing the statement "The matrices have been exported as .csv files named Nodes, Edges, and TwoMode. These can be found in your working directory." followed by the user's current working directory where the files have been saved using the last two statements.

```
r <- c("The matrices have been exported as
.csv files named Nodes, Edges, and TwoMode.
These can be found in your working
directory." ,getwd())
   print(r)
```

The new .csv files can be used to conduct an SNA. The complete `snafu` function is available in the Appendix.

One Mode Undirected Example

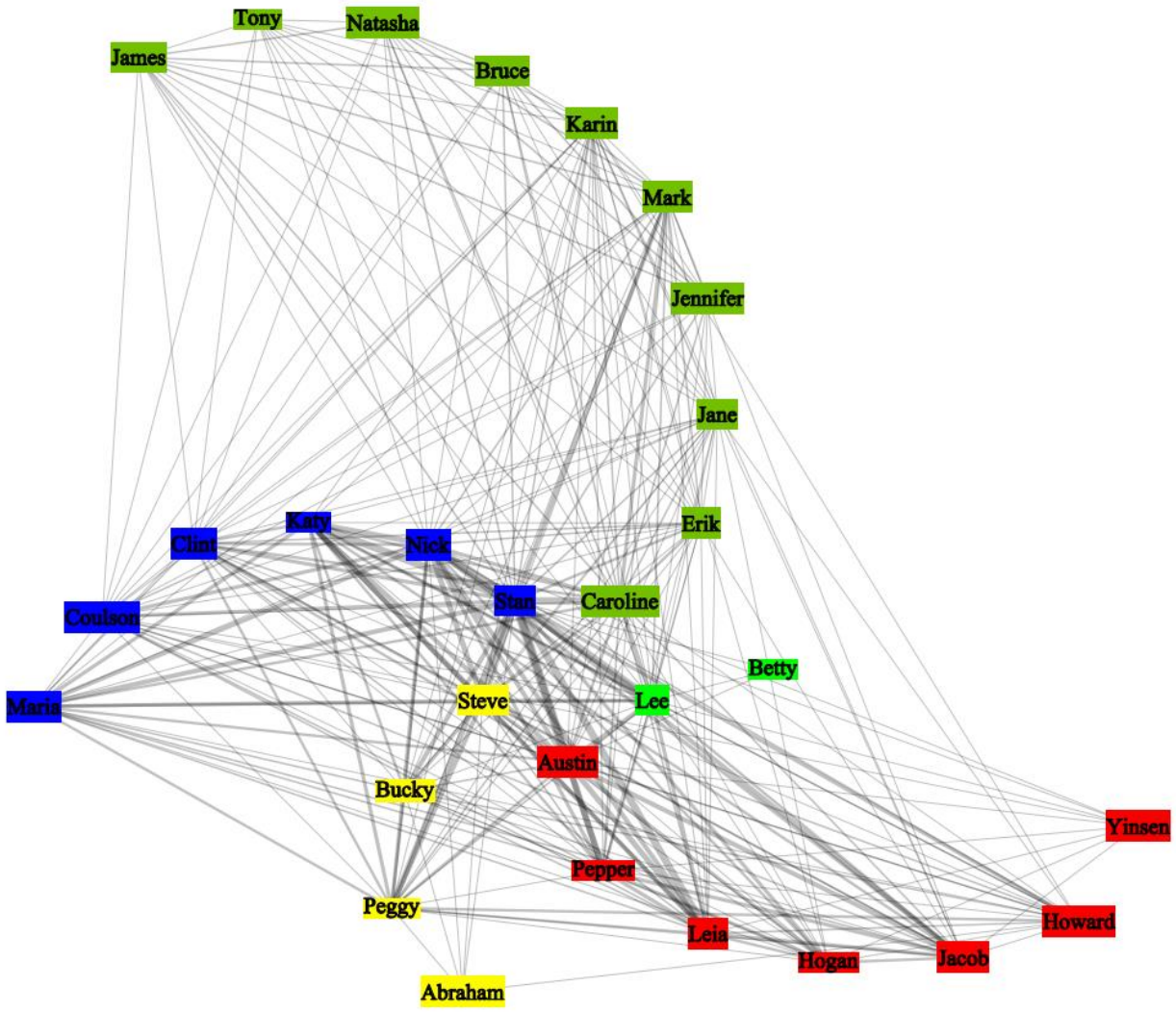
To demonstrate the utility of the `snafu` function, we created a toy data set representing elementary school students attending after school programs offered in their district. The data for this analysis was created and stored in a .csv file. The data set consists of two columns, the first containing the names of 29 fictitious elementary school students while the second contains the after-school programs which they attended. The data were recorded in long form, so each student is repeated as many times as necessary to cover programs which they attended.

The created data set was then run through the `snafu` function using the following line of code:

```
snafu(Data$Student, Data$Program)
```

Once completed, the function produced three matrices. The file named Nodes.csv contained the adjacency matrix of the number of shared programs between each of the students, the one named Edges.csv contained the adjacency matrix of the number of shared students between each of the programs, and the final named TwoMode.csv contained the affiliation matrix of the students and programs. For this example we used the Nodes.csv. This .csv file was then imported into Gephi (Bastian, Heymann, & Jacomy, 2009) to create the sociogram for the one mode networks, see Figure 2. The sociogram was modified in the program to increase legibility and ease of understanding.

Figure 2. One Mode Sociogram



Note: The node colors indicate the grade level of the student (e.g., Red = 1st Grade, Blue = 2nd Grade, Green = 3rd Grade, and Yellow = 4th Grade).

Two Mode Undirected Example

To demonstrate the utility of the `snafu` function for two mode undirected research, an SNA was conducted examining the connections among high schools and professional development trainings. The data for this analysis came from a large, suburban school district in the southeastern United States. These data contain information on the district's high school teachers' attendance at professional development trainings held by the county. The .csv file consists of three columns, the first containing the teacher ID numbers, followed by the school the teacher was from, and finally the ID number of the training attended. This data set contained 3920 teachers across 12 high schools and 85 different trainings that were attended. The data were first recorded in long form, so each teacher and school is repeated as many times as necessary to cover all training sessions which were attended.

The created data set was then run through the `snafu` function using the following line of code:

```
snafu(SchoolData$School,
SchoolData$Training)
```

Once completed, the function produced three matrices. The file named Nodes.csv contained the adjacency matrix of the number of shared trainings between the schools, the one named Edges.csv contained the adjacency matrix of the number of schools attending each training, and the final named TwoMode.csv contained the affiliation matrix of the schools and trainings. For this example we used the Nodes.csv. adjacency matrix and the TwoMode.csv matrix to examine the connections among the high schools based on the teacher attendance to the professional development training and the relationships between the high schools and the trainings. These two .csv files were then imported into Gephi to create the sociograms for the two networks, see Figures 3 and 4. Both sociograms were modified in the program to increase legibility and ease of understanding. The nodes representing schools in both figures are labeled H1 through H12, while the trainings in Figure 4 are labeled 1 through 85.

Conclusion

The goal of this paper was to present a simple and effective method for reformatting data for use in SNA. It is our hope that by mitigating the data format issue, researchers and practitioners will increase their use of SNA. The function simply requires two columns from a long form data set that represent the nodes and edges one is interested in exploring. Any researcher, regardless of their field of study, can utilize the function to find the social networks that exist within their data.

While our hope is this function can help further the use of SNA in assessment and evaluation research, it is limited in that it can only be used for one and two mode undirected sociograms. Due to the format of the data needed to run the function, there is no way to separate the interactions between two nodes based on any potential direction of the relationships. It only sums the total relationships between the two nodes, producing matrices that can only be used for undirected sociograms. However, we believe our function is still applicable and valuable to assessment and evaluation research, and any attempt to discuss the

Figure 3. One Mode Schools Sociogram

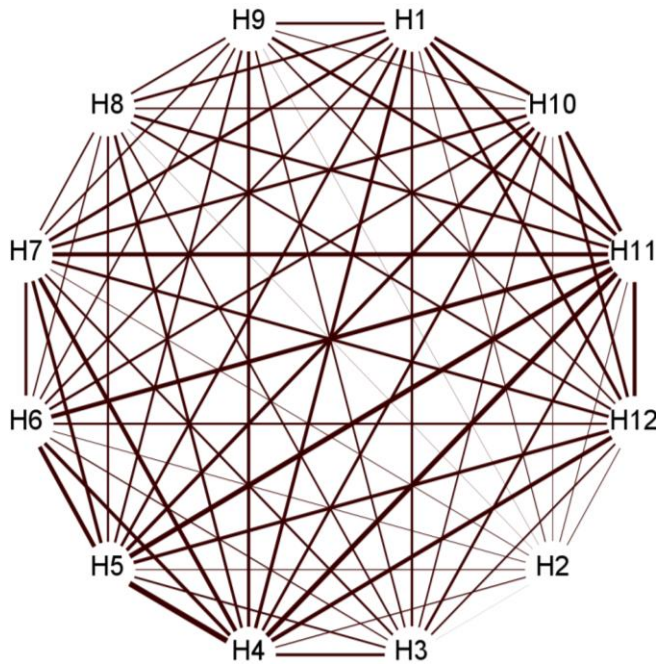
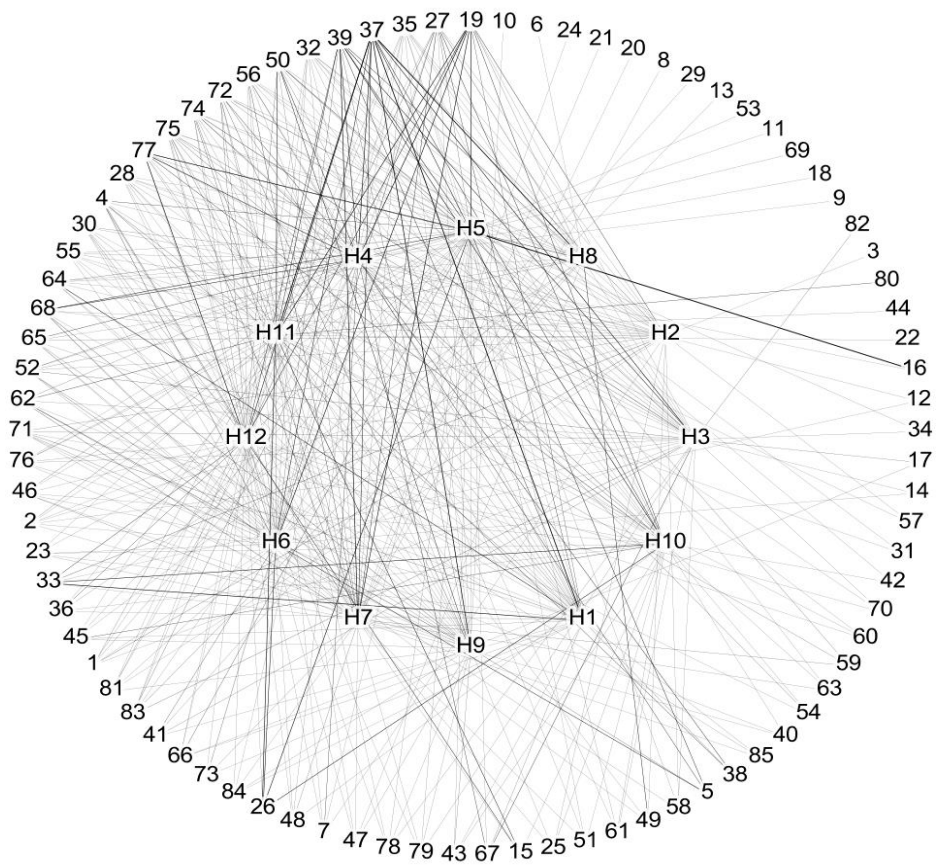


Figure 4. Two Mode Schools and Trainings Sociogram



directionality of the relationships in the sociograms should be justified with supporting evidence. Moreover, data typical for directed SNA are usually collected in the form needed for most SNA programs, so no reformatting is usually necessary.

Since our reformatting solution only works in R, we have created an R Studio Cloud project to give non-R users the ability to utilize our function via a web browser. R Studio Cloud is a web-based platform that allows people to share R projects. The following link provides access to the `snafu` cloud project: https://rstudio.cloud/project/2933894. Users will first need to create a free R Studio Cloud account, then click on the "To_Run_snafu.R" file in the lower right pane for instructions on utilizing R Studio Cloud to run the `snafu` function.

Any researcher can utilize the `snafu` function to conduct an SNA and create undirected sociograms if the data are in long form. This paper has described the function to reformat long form data into adjacency matrices for use in SNA. The `snafu` function provides researchers a simple and quick way to format their data for use in creating sociograms of the underlying social network present. Simply supply the function with vectors containing the nodes and edges and the original data will be restructured.

References

Aggarwal, C. C. (2011). An Introduction to Social Network Data Analytics. *Social Network Data Analytics*. Springer, Boston, MA

Bastian M., Heymann S., & Jacomy M. (2009). Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media. From AAAI.

Borgatti, S.P., Everett, M.G. and Freeman, L.C. 2002. Ucinet for Windows: Software for Social Network Analysis. Harvard, MA: Analytic Technologies.

Carolan, B. V. (2014). *Social network analysis and education: theory, methods & applications*. Sage.

Janus, S. (2016). *Becoming a knowledge-sharing organization: A handbook for scaling up solutions through knowledge capturing and sharing*. World Bank Group. https://openknowledge.worldbank.org/handle/10986/25320

Long, J. D., & Teetor, P. (2019). *R Cookbook: proven recipes for data analysis, statistics, and graphics* (2nd ed.). Beijing: O'Reilly.

McGloin, J. M. (2005). Policy and intervention considerations of a network analysis of street gangs*. *Criminology Public Policy*, *4*(3), 607–635. https://doi.org/10.1111/j.1745-9133.2005.00306.x

Nagarajan, K., Muniyandi, M., Palani, B., & Sellappan, S. (2020). Social network analysis methods for exploring SARS-COV-2 contact tracing data. *BMC Medical Research Methodology*, *20*(1). https://doi.org/10.1186/s12874-020-01119-3

Saqr, M., Fors, U., Tedre, M., & Nouri, J. (2018). How social network analysis can be used to monitor online collaborative learning and guide an informed intervention. *PLOS ONE*, *13*(3). https://doi.org/10.1371/journal.pone.0194777

Tichy, N., & Fombrun, C. (1979). Network Analysis in Organizational Settings. *Human Relations, 32*(11), 923–965. https://doi.org/10.1177/001872677903201103

van der Loo, M. P. J. (2021) *The Data Validation Cookbook version 1.0.4*. https://data-cleaning.github.io/validate.

Wickham, H. (2007). "Reshaping Data with the reshape Package." *Journal of Statistical Software*, *21*(12), 1–20. http://www.jstatsoft.org/v21/i12/.

Wickham et al., (2019). Welcome to the tidyverse. *Journal of Open Source Software, 4*(43), 1686, https://doi.org/10.21105/joss.01686

Wickham, H. & Henry, L. (2020). tidyr: Tidy Messy Data. R package version 1.1.0. https://CRAN.R-project.org/package=tidyr

Yang, S., Keller, F., & Zheng, L. (2017). *Social network analysis: methods and examples*. Los Angeles: Sage.

Youm, Y., & Laumann, E. O. (2002). Social network effects on the transmission of sexually transmitted diseases. *Sexually Transmitted Diseases*, *29*(11), 689–697. https://doi.org/10.1097/00007435-200211000-00012

Citation:

Boyd, A. T., & Rocconi, L. M. (2021). Formatting data for one and two mode undirected social network analysis. *Practical Assessment, Research & Evaluation*, 26(24). Available online: https://scholarworks.umass.edu/pare/vol26/iss1/24/

Corresponding Author

Austin T. Boyd
Department of Educational Psychology and Counseling (Evaluation, Statistics, and Methodology)
The University of Tennessee, Knoxville
Knoxville, Tennessee, USA

Email: aboyd26@vols.utk.edu

Appendix A

```
# SNAFU

snafu <- function(x, y, diagonal=FALSE) {

  # Create a 2 mode x by y affiliation matrix using vectors x and y
  M <- as.matrix(table(x, y))


  #Transform into an x by x and a y by y matrix
  Nodes <- tcrossprod(M)
  Edges <- crossprod(M)

  #Remove lower triangle of each matrix and
  #determine whether to keep the diagonal
  #Default is FALSE = Remove the diagonal
  #TRUE = Keep the diagonal
  if (diagonal == FALSE){
            Nodes[lower.tri(Nodes,diag = TRUE)]=0
            Edges[lower.tri(Edges,diag = TRUE)]=0
       } else {
            Nodes[lower.tri(Nodes,diag = FALSE)]=0
            Edges[lower.tri(Edges,diag = FALSE)]=0
       }


  #Save the matrices to the global environment
  Nodes<<-Nodes
  Edges<<-Edges
  TwoMode<<-M

  #Export the reformatted matrices as a new
  #.csv files to the user's working directory
  write.table(Nodes, file = "Nodes.csv", na="", col.names=NA, sep=",")
  write.table(Edges, file = "Edges.csv", na="", col.names=NA, sep=",")
  write.table(M, file = "TwoMode.csv", na="", col.names=NA, sep=",")

  #Print output statement and user's working directory
  r <- c("The matrices have been exported as .csv files named Nodes,
Edges, and TwoMode. These can be found in your working directory.", getwd())
  print(r)

  }
```