# Handling Missing Data in Exponential Random Graph Models:

# A Comparison of Approaches

Nathan T. Abe

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor in Philosophy

University of Washington

2021

Reading Committee:

Elizabeth Sanders, Chair

Min Li

Jessica Thompson

Program Authorized to Offer Degree:

College of Education

University of Washington

## ABSTRACT

Handling Missing Data in Exponential Random Graph Models: A Comparison of Approaches

Nathan T. Abe

Chair of the Supervisory Committee:
Associate Professor Elizabeth A. Sanders
College of Education

This dissertation represents a series of studies focused on comparing imputation approaches for single-mode networks, also known as graphs, that are missing tie information due to a variety of potential causes unrelated to network properties, such as illness or technology failure. Additionally, in social network measurement designs that ask subjects to nominate other people in the network based on free recall (rather than forced choice), missingness can arise when people outside the surveyed network are sometimes nominated. The aim of this dissertation is to understand best approaches for handling this type of missingness in terms of coefficient estimation accuracy and precision. Specifically, Study 1 compared imputation approaches for graphs with binary valued ties measured at a single time point; Study 2 investigated approaches for handling missingness in graphs with binary valued ties measured at two time points; and Study 3 focused on approaches for handling missingness in graphs with integer valued ties (e.g., counts and ratings) measured at a single time point. Monte Carlo simulations were conducted in R using `statnet`, with varied network sizes, densities/mean values, and missingness levels. With a focus on use of the exponential random graph family of models (ERGMs), approaches to handling missing tie data included `statnet` default approaches (e.g., node-wise deletion or assuming all missingness represents no tie) as well as approaches in better alignment with modern missingness handling (e.g., stochastic imputation based on mean sample values). Findings consistently showed that stochastic imputation was best for minimizing bias and maximizing precision. Last, I describe a new *R* package, `netImp`, developed to implement imputation approaches so that researchers can obtain complete data matrices prior to analysis. Limitations, policy implications, and future research directions are discussed.

*Keywords*: social networks, SNA, exponential random graph model, missing data, netImp

## ACKNOWLEDGEMENTS

I would like to thank all the faculty and students in Measurement & Statistics as well as my committee for their support throughout my graduate studies at the University of Washington, College of Education. It was an amazing adventure and I would never have been able to complete this program without all of their support.

I am forever in debt to my committee members for their guidance and wisdom throughout this process. I am deeply grateful to Dr. Liz Sanders, my advisor, for her tireless effort to encourage, teach, and support me in my pursuit of learning, and ultimately, my doctoral degree. I could never have done this without you. Special thanks as well to Dr. Min Li who was always kind and supportive, but always held me to rigorous standards, and to Dr. Jessica Thompson for her generosity in sharing the data from the AST and PASTL projects I used as inspiration for this dissertation – she also provided key feedback that ensured my methodological work would be relevant to practitioners. I also thank Dr. Adrian Dobra for his expertise in statistics as well as his dry sense of humor and straightforwardness; I tremendously enjoyed taking his classes. Last but not least, I am very grateful to Dr. David Knight, who was willing to join my committee late in the process. Thank you!

I would also like to thank my family and friends as well. Their constant support helped to keep me going, especially during the pandemic. They relentlessly served as my sounding board for ideas, project presentations, and job talks – even if they did not completely understand what I was talking about! I am profoundly appreciative of their patience and understanding, and their unfailing belief in me.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1.

## Network Analysis in Education Research

### Dissertation Overview

This dissertation aims to determine the best approach for handling missing data in social networks that are analyzed using exponential random graph models. I begin by providing an overview of social networks and related terminology, as well as basic types of analyses for network data. In Chapter 2, I review missing data theory, including traditional and modern approaches to missingness handling. In both Chapters 1-2, I review studies published in top-tier education journals to shed light on how social network analysis is being taken up in the field, including how applied researchers have been handling missingness. This leads to Chapters 3-5, which compare missingness handling approaches for three types of network data: binary, single time point; binary, multiple time point; and weighted, single time point. Chapter 6 describes a new *R* package, `netImp`, designed to impute data for binary and valued networks with data assumed missing at random. Last but not least, Chapter 7 synthesizes the findings and describes limitations and future research directions.

### What are Social Network Analyses?

In the social and behavioral sciences, social network analysis (SNA) is a set of methods used to describe and test characteristics of human interactions within a network of individuals and their influence (Knoke & Yang, 2011). For example, SNA has been used to investigate the spread of disease and drug use, physical and emotional health outcomes, political party affiliations, trade patterns, kinship, and social media relations and uses (e.g., Granovetter, 1973; Valente, Gallaher, & Mouttapa, 2004; White, 2014). In education, SNA may be used to understand connections among teachers within and across schools, or among principals within

and across districts, particularly during or after professional development in the hopes of increasing information sharing amongst participants, thereby improving instruction or leadership (Thompson et al., 2019; Windschitl, Thompson, Braaten, & Stroupe, 2012). Irrespective of discipline, a fundamental goal of a social network analysis is to gain insight into the structure of the interactions among individuals from a specific population.

**Network Terminology**

Before discussing specific SNA methodologies or their prevalence in education research, it is instructive to highlight relevant terminology. First, **networks** are *relationship patterns* among entities; in statistical theory they are known as *graphs*. The size of a network, *N*, refers to the number of entities comprising the network. These entities are called **nodes** (also known as actors, vertices, and subjects); nodes can be humans or any other type of individual, including animals, plants, objects, events, and ideas; the term "social network" refers to networks with nodes that are human. Networks can have one type of node (called single-mode, just people, for example) or more than one type (e.g., two-mode, with people and events, for example). The studies in this dissertation pertain to on single-mode (one type of node) networks only.

The fundamental unit of analysis in networks is the **dyad**: dyads are *pairs of nodes*. If there are *N* nodes in a single-mode network, then there are *N**(*N*-1) dyads possible in the network. **Ties** (also referred to as links, edges, vertex endpoints, and arcs) are the *connections* among dyads (Robins & Lusher, 2013). Ties among dyads are often valued as binary (presence = 1 vs. absence = 0), or "weighted" using an integer to reflect the strength of the connection (e.g., rating scales and counts).

A network can be characterized as undirected or directed. An **undirected** network is one that captures associative connections that co-occur without temporal or perceived order. An

example of an undirected network would include measuring teacher connections via a shared curriculum, practice, or project. In this type of network, teachers A, B, and C may have all worked jointly together and therefore there would be ties among all possible dyads (pairs of nodes): AB, AC, BC, as well as BA, CA, and CB. Alternatively, networks with connections that are directional are called **directed**. A directed network, for example, would include teachers nominating other teachers as advice-givers; in such networks, teacher A may nominate teachers B and C, but teachers B and C may not nominate teacher A in return. Further, teachers B and C may not nominate each other either. In this example, there would only be ties for two of the six dyads among the three nodes: AB and AC, but not BC, BA, CA, or CB. Terminology that is often used with regard to directed networks includes "senders" and "receivers" which are synonyms (in this context) to "nominators" and "nominees." The sender/nominator initiates the connection to the receiver/nominee.

Instead of a subject X variable data matrix traditionally used in individual-level statistical analyses, network analyses typically make use of a node X node **adjacency matrix**, as well as an edgelist and a nodelist. In an undirected network, the matrix would be symmetric because every sender would also be a receiver. In a directed network, the rows represent actions or communications initiated by the nodes (as senders/nominators), and the columns represent the actions or communications received by the nodes (receivers/nominees). Figure 1.1 displays an adjacency matrix for a directed network with $N = 7$ nodes; we may imagine this to be a network of teachers nominating peers from whom they seek pedagogical advice. Looking at the rows, we see that Jennifer nominated Sarah, Jim nominated Jennifer and John, Sarah nominated no one, and so forth. Examination of the columns indicates, on the other hand, that Jennifer, Jim, Amy,

and Justin all received one nomination whereas Sarah and John received two, and Beth received none.

**Figure 1.1**

*Example of an Adjacency Matrix for a Directed Teacher Advice Network*

| | Jennifer | Jim | Sarah | Amy | John | Justin | Beth |
|---|---|---|---|---|---|---|---|
| Jennifer | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Jim | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Sarah | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Amy | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| John | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| Justin | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Beth | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

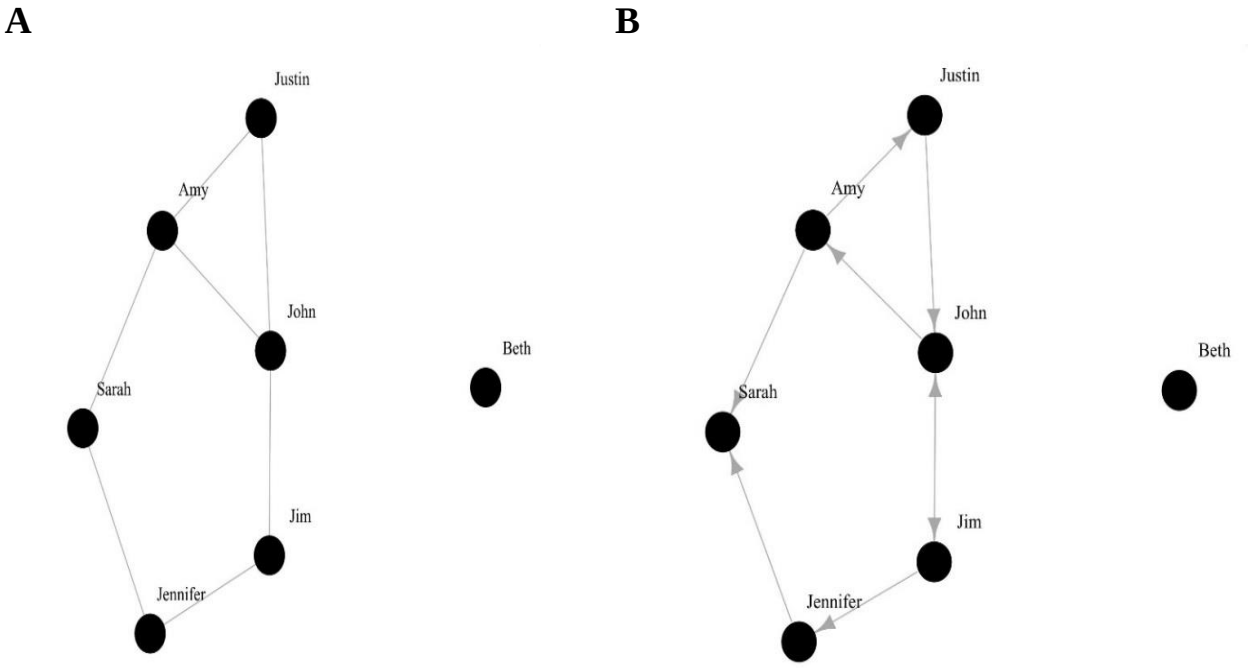
As mentioned above, the adjacency matrix is typically accompanied by an edgelist and nodelist. An **edgelist** is a simple ASCII-type file derived from the adjacency matrix, which is a list of all dyads observed with a tie; in this case, the list would have Jennifer-Sarah, Jim-Jennifer, Jim-John, etc. The **nodelist**, on the other hand, would be a list of all the nodes (seven in this case) along with corresponding covariate information if available, such as whether or not the node is a teacher or principal, or years of experience. Despite the use of a nodelist, it is important to emphasize that most SNA methodologies will focus on dyads as the unit of analysis, not individuals; in other words, node characteristics will be used to predict connections, not individual behavior.

A staple of all social network analyses includes the **sociogram**, which is a two-dimensional *data visualization* that pictorially represents the network nodes and ties. Often the nodes are assigned sizes that correspond to network descriptive statistics (more about network descriptives forthcoming) as well as colors and shapes that correspond to node characteristics (based on any measurement scale) (e.g., teachers and principals as nodes with different shapes, at

different schools represented by different colors). The ties will be represented with lines and arrows if the network is directed, or just lines if the network is undirected. The lines may also have thicknesses and colors that represent network characteristics. Figure 1.2 provides examples of a sociogram corresponding to the adjacency matrix shown earlier (Panel A = undirected, Panel B = directed); the nodes are represented as black circles and ties among dyads displayed as lines.

**Figure 1.2**

*Example Sociograms for Undirected (Panel A) and Directed (Panel B) Teacher Networks*




If a node does not nominate anyone and was also not nominated by any other nodes, the node is referred to as an **isolate**. For example, in Figures 1.1 and 1.2B, Beth was not nominated by other teachers and she herself also did not nominate anyone (no ties to or from Beth). Identifying isolates may be of practical importance for decision makers; for example, isolates may represent teachers who feel left out (perceived or otherwise) and therefore are at risk for leaving their job.

Not only can networks be directed or undirected, but they can also be **unweighted** (ties that are valued as binary) or **weighted** (ties that are valued as a magnitude). Unweighted networks simply have a dichotomous relationship of existing or not existing (tie or no tie), like the examples given in Figures 1.1 and 1.2. This said, most person-person relationships can be argued as being *not* strictly dichotomous (Krivitsky & Handcock, 2014). For example, friendship networks, advice seeking networks, and international relations all have varying degrees of strength or frequency of interactions. As an example, if we imagine that the teacher-teacher directed network shown in Figures 1.1 and 1.2 were reconsidered as the number of times teachers sought advice from each other, our adjacency matrix and corresponding sociogram may look like those shown in Figure 1.3, Panels A and B, respectively.

**Network Descriptive Statistics, in Brief**

There are a number of excellent texts that define network statistics (Carolan, 2014; Harris, 2014; Yang et al., 2017). Below I provide a few of the most common types of descriptive tabulations for networks.

**Density.** Density is perhaps the most important *overall network* measure: it is the mean connectedness of the network. For binary valued networks, density can range from 0 to 1; it is calculated as the proportion of observed ties out of the number of dyads (possible ties). While there are no rules of thumb for density levels like there are for effect sizes or other metrics, densities closer to 0 are termed “sparse” and densities closer to 1 may be considered “saturated.”

For an undirected network, the number of ties possible is equal to the number of unique combinations of dyads = $N*(N-1)/2$; for directed networks, the number of ties possible is equal to the number of all possible combinations of dyads = $N*(N-1)$. For valued networks, the “density” becomes the mean value either unique combinations of dyads (undirected) or all possible

combinations of dyads (directed); the type of mean used depends on the distribution assumed of the values. To gain a sense of this concept, Figure 1.4 illustrates the difference between two binary directed networks that are the same size ($N$ = 20) but have different density levels.

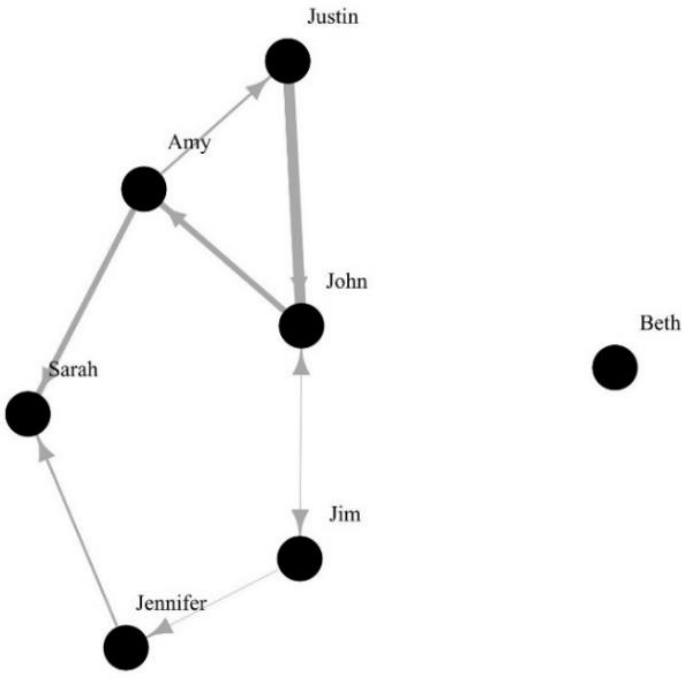
**Figure 1.3**

*Example of a Weighted, Directed Network Adjacency Matrix and Corresponding Sociogram*

**A**

| | Jennifer | Jim | Sarah | Amy | John | Justin | Beth |
|---|---|---|---|---|---|---|---|
| Jennifer | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| Jim | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Sarah | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Amy | 0 | 0 | 3 | 0 | 0 | 2 | 0 |
| John | 0 | 1 | 0 | 3 | 0 | 0 | 0 |
| Justin | 0 | 0 | 0 | 0 | 4 | 0 | 0 |
| Beth | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**B**

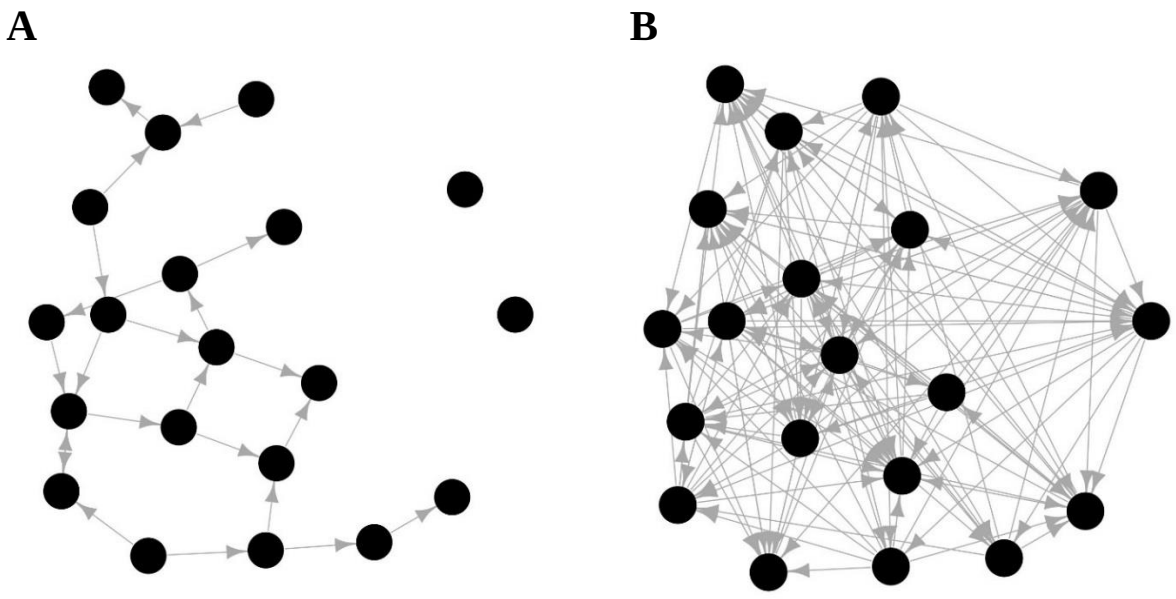


**Tie properties**. The simplest is perhaps **reciprocity**, which is also referred to as *mutuality*; reciprocity is the percentage of *ties* in the network for which dyad members both send and receive nominations to and from each other, relative to the number of ties observed. Examination of the directed network in Figures 1.1 and 1.2B shows that John and Justin have a reciprocal tie, represented with a double arrow. This is the only mutual tie out of all the ties, which indicates that mutuality in this network is quite low.

Similar to reciprocity, **homophily** is a term used to describe *dyads* that have the same attribute, such as students who share the same school, are in the same class, or have the same race/ethnicity (Lusher & Robins, 2013). Theoretically, homophilous dyads have a higher probability of a tie existing between them than non-homophilous dyads (Goodreau et al., 2009). For example, teachers who teach in the same school or teach the same subject will be more likely to seek advice from each other than those from different schools or who teach other subjects.

**Figure 1.4**

*Examples of Networks of Size 20 with Low (Panel A, 6%) and High (Panel B, 36%) Densities*



**Measures of Centrality and Centralization**. Some of the most popular network descriptive statistics are called measures of centrality or centralization, all of which are ways of describing connections in the network. Importantly, the term centrality refers to individual node properties, whereas the term centralization refers to the overall network as a whole. As an aside, in order to understand some of these statistics, it is fundamental to understand the idea of a "**path**." A path from one node to another can be direct (number of steps = 1) or indirect through other nodes (number of steps = 2 or more). The more connected nodes are in general, the shorter the path distances.

**Degree centrality** provides a count of the number of ties a node has. For a directed network, this is divided into in-degree and out-degree centrality. In-degree centrality (sometimes referred to as popularity) is a count of the number of ties sent into the node (receiving/nominee ties); out-degree centrality is the number of ties being sent out from the node (sending/nominating ties) (e.g., Lusher & Robins, 2013). In Figure 1.1, Amy has an in-degree centrality = 1 because Amy was nominated only once, by John, whereas Amy's out-degree centrality = 2 because Amy nominated two people, Justin and Sarah. **Degree centralization**, on the other hand, is a calculable ratio ranging from 0 to 1 that indicates the extent to which ties in the network (overall, as well as in- and out-degree) are concentrated on a single node or group of nodes (Carolan, 2014).

Another type of centrality is called **betweenness centrality**. Instead of counting ties, this statistic counts the number of unique shortest paths (known as geodesics) that a node may be in the middle of, serving as a bridge between other nodes in the network (Runspan et al., 2014). Looking at Figure 1.2B, John has the highest betweenness centrality = 8 because there are eight paths from one node to another linked by John. A similar **betweenness centralization** measure

can also be computed for the entire network ranging from 0 to 1. Networks with high betweenness centralization values indicate that there are many nodes serving as bridges among other nodes (Carolan, 2014).

**Closeness centrality** is yet another popular network descriptive statistic, which is a value that indicates the extent to which individual nodes are able to spread information efficiently through a network (Golbeck, 2013). The value is the mean of the node's inverse distance to all other nodes; in other words, high values indicate that the node is fairly close to all the other nodes in the network. Again, in Figure 1.2B, John has the highest closeness value because John is the most connected to others in the network. **Closeness centralization** on the other hand is the degree to which nodes are, on average, close to each other; again the ratio ranges from 0 to 1, with higher values indicating greater network efficiency.

The last measure of centrality is the **eigenvector centrality** which indicates a node's relative importance compared to its neighbors (as its name implies, the adjacency matrix can be mathematically decomposed into eigenvalues and eigenvectors). A node that has high eigenvector centrality is thought to be well connected even if it has very few observed in- or out-degree ties, given the pattern of its indirect associations. When looking at Figure 1.2B, John likely has the highest eigenvector centrality because he is the most connected. Like the other measures of centrality, a network-level centralization statistic can also be computed.

Last but not least, two other important measures of centrality exist, known as distance and diameter. **Distance** is a node- and network-level measure of the average length of a path from one node to another, regardless of directionality of ties. For example, Amy's (nodal) distance from Jennifer is two. However, the average distance of all ties in the network is 1.90, indicating the network does not have many ties across all nodes. **Diameter** on the other hand is a

measure of the longest **geodesic** path in the network taking into account the directionality of ties. (A geodesic path is the path with the minimum number of steps (ties) taken to get to other nodes in the network; for this reason, it is often called the "longest-shortest path.") If paths tend to be short (i.e., few steps between nodes), information can hypothetically travel more efficiently through the network compared to paths that are longer. Like other measures of centrality, distance and diameter can be measured for each node, and then summarized as centralization statistics for the entire network.

In addition to these measures of centrality, it is often the case that researchers will examine the number of types of "triangles" among dyads with and without ties, known as **transitivity**. These structures often act as "structures" or clusters of nodes in the network; excellent descriptions can be found in Harris (2014), among others.

**Types of Network Analyses**

While the term 'social network' generally refers to a collection of individuals who are connected to each other in a specific relational context (e.g., social media, families, classrooms, and workplaces), the term **social network analysis** (SNA) involves data analysis that permits researchers to draw conclusions about those relations. Freeman (2004) outlined four characteristics of SNA that may be used differentiate it from traditional individual-level analyses, including: (a) a focus on relationships among individuals, rather than individual behaviors or attributes; (b) analysis based on relationship patterns; (c) use of graphic imagery to illustrate patterns of relationships; and (d) use of mathematical tools to understand the nature of those patterns of relationships.

Broadly, SNAs can be classified into three types of methodologies: qualitative, quantitative descriptive, and quantitative inferential. An example of a qualitative approach to

SNA might be to conduct an ethnographic study that collects and analyzes discourse, interview, and observational data about the systems of support for students within a particular school, including teachers, counselors, administrators, and students. In contrast, quantitative SNA involves surveying or observing individuals to determine who (or what) they interact with. Those data can then be tabulated to compute descriptive information, like the statistics described earlier as well as associated sociograms. Principals, for example, can use network descriptives to better understand patterns of connections among their teachers to potentially create pathways for facilitating greater teacher interactions, thereby improving practice and reducing burnout.

The third type of SNA – the one this dissertation focuses on – quantitative inferential, uses **statistical models** to predict connections in a network using observed network properties as well as individual- and dyad-level covariates, much like a regression model uses observed individual data to predict individual behaviors. These results can then be used to draw inferences (conclusions) about the network probabilistically, rather than trying to make sense of descriptive statistics and sociogram patterns that may in fact be due to chance fluctuations. Three major kinds of statistical models are in use today for single-mode networks[1] (i.e., networks that examine connections among one type of node), including: exponential random graph models (ERGMs), stochastic actor-based models (SABMs) and latent space models (LSMs).

**SNA Methodologies in Educational Research**

Although a variety of disciplines have used SNA for decades, the adoption of SNA as an analytical tool for education research appears to be slower. To obtain an estimate of the nature and prevalence of SNAs in recent educational research, I conducted a literature review of articles

[1] Two-mode or multi-mode networks, on the other hand, include measuring connections among more than one type of unit of analysis, such as people connecting to events, rather than people; these networks result in what is known as “bipartite” graphs.

published from 2015 to 2020 in well-respected (and relatively high-impact) U.S. education research journals, as well as articles published in an interdisciplinary journal dedicated to network analysis, *Social Networks* (Impact Factor: 2.376), as a point of comparison. The education research journals selected included: *American Education Research Journal* (AERJ; Impact Factor: 5.013), *Educational Evaluation and Policy Analysis* (EEPA; Impact Factor: 4.000), *Journal of Educational Behavioral Statistics* (JEBS; Impact Factor: 2.042), *Journal of Educational Psychology* (JEP; Impact Factor: 5.178), *Journal of Learning Sciences* (JLS; Impact Factor: 3.588), and *Teachers College Record* (TCR; Impact Factor: 0.970). Within each journal's website, I specified the year range and used the following global (i.e., appearing anywhere in the article) search terms: "social network analysis," "network analysis," and "social networks."

The search identified a total of 444 unique articles out of 2524 published. For each of the identified articles, I reviewed the type of network data collected (binary or weighted/valued) as well as the type of SNA employed. Results (see Table 1.1) revealed that the prevalence of any SNAs between 2015 and 2020 in the selected education journals was relatively low, with 54 articles, representing 3% of the articles published. Of studies that used SNA, most collected binary network data (i.e., any connection vs. no connection: 87% of education journals, and 70% of the articles in the interdisciplinary journal dedicated to social networks), and most used qualitative methodology or methods that were secondary analyses of individual-level network data (e.g., using network status tabulations to predict individual behavior). Reporting descriptive statistics was the second most popular SNA approach in education journals (30%), and was also a relatively well-used approach in the journal dedicated to social networks (9%). Among the statistical model-based SNAs, ERGM-based models were the most used (11% and 18% of SNAs in the educational journals and the social network journal, respectively).

**Table 1.1**

*Literature Review Results for SNA Methodologies in Education Research, 2015-2020*

| Journal | Total Articles | SNA Used | | Type of Network Data (if Quantified) | | | | Type of SNA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Binary | | Weighted | | Qualitative or Other | | Quantitative Descriptive | | Quantitative Model-Based | | | |
| | | | | | | | | | | | | ERGM | | LSM | | SABM | |
| | | *N* | *(%)* | *N* | *(%)* | *N* | *(%)* | *N* | *(%)* | *N* | *(%)* | *N* | *(%)* | *N* | *(%)* | *N* | *(%)* |
| *Education* | | | | | | | | | | | | | | | | | |
| AERJ | 321 | 17 | (5%) | 14 | (82%) | 3 | (18%) | 6 | (35%) | 5 | (29%) | 3 | (18%) | 3 | (18%) | 0 | (0%) |
| EEPA | 177 | 3 | (2%) | 2 | (67%) | 1 | (33%) | 1 | (33%) | 1 | (33%) | 0 | (0%) | 1 | (33%) | 0 | (0%) |
| JEBS | 132 | 6 | (5%) | 6 | (100%) | 0 | (0%) | 3 | (50%) | 0 | (0%) | 2 | (33%) | 1 | (17%) | 0 | (0%) |
| JEP | 571 | 6 | (1%) | 4 | (67%) | 2 | (33%) | 2 | (33%) | 2 | (33%) | 0 | (0%) | 0 | (0%) | 2 | (33%) |
| JLS | 147 | 2 | (1%) | 2 | (100%) | 0 | (0%) | 1 | (50%) | 1 | (50%) | 0 | (0%) | 0 | (0%) | 0 | (0%) |
| TCR | 786 | 20 | (3%) | 19 | (95%) | 1 | (5%) | 12 | (60%) | 7 | (35%) | 1 | (5%) | 0 | (0%) | 0 | (0%) |
| Total | 2134 | 54 | (3%) | 47 | (87%) | 7 | (13%) | 25 | (46%) | 16 | (30%) | 6 | (11%) | 5 | (9%) | 2 | (4%) |
| *Interdisciplinary* | | | | | | | | | | | | | | | | | |
| Social Networks | 390 | 390 | (100%) | 272 | (70%) | 118 | (30%) | 257 | (66%) | 34 | (9%) | 69 | (18%) | 5 | (1%) | 27 | (7%) |

**Exponential Random Graph Models (ERGMs)**

As was observed in the literature review, the most popular statistical model found in the selected education and interdisciplinary journals was the family of exponential random graph models, known as ERGMs (Leifeld & Cranmer, 2016). It is likely that the popularity of this class of models is that they are similar to the more well-known logistic regression model and therefore have (fairly) straightforward interpretations, and that they are able to handle binary and weighted directed and undirected network data. Further, multi-time point ERGMs as well as Bayesian ERGMs have been developed. As such, although other classes of network models have also been developed (i.e., LSMs and SABMs), ERGMs will be the focus of this dissertation.

Single time point (cross-sectional) ERGMs for binary data estimate the probability of a tie forming in the network, conditional on the network's characteristics. For directed networks, ERGM can also estimate the likelihood of a mutual tie occurring for any new tie formed. Further, ERGM can also be used with ties that are valued as integers (i.e., "weighted") to estimate the mean magnitude of connections in the network, conditional on the network characteristics. Importantly, covariates are often used in these models to better understand what might correlate with the formation of a new tie (e.g., one may use node-level demographic information, such as school membership, to predict the likelihood of a new tie forming among teachers in a professional development network); further, network structures such as transitivity can also be incorporated if specified. All model parameters are estimated in log-odds (logits), much like logistic regression, and because a likelihood-based function is used to estimate parameters, nested models can be compared using likelihood ratio tests and fit statistics (e.g., the Bayesian Information Criterion). This said, the interpretation of covariate effects is not as easy as in a regular regression model – the coefficients represent a change statistic that is associated with the

*sum* of two nodes with the same characteristic (see Harris, 2014). Finally, ERGMs are fairly easily conducted using `statnet` suite of packages in *R* (Handcock et al., 2008; Handcock et al., 2016); data visualizations are most effectively generated using the *R* `Igraph` package (Csardi & Nepusz, 2006).

**The Exponential Random Graph Model**

Statistical models for network data have a rich history, beginning with simple graph models in the 1950s that evolved into more sophisticated dyadic independence models (Erdos & Renyi, 1959; Frank & Strauss, 1986; Karonski, 1982), also known as $p_1$ models (Holland & Leinhardt, 1981), and later, dyadic dependence models, also known as p* models (Wasserman & Pattison, 1996) which are the basis of the modern family of exponential random graph models (ERGMs). Essentially, there was a shift over time from predicting the likelihood of an entire network to predicting the likelihood of a tie conditional on the network properties; additionally, there was an explicit awareness that dyads with similar connections were not independent of one another (e.g., the ideas behind homophily and transitivity), and that dyads with similar individual node or relationship characteristics are also more likely to be connected (i.e., there was a need for being able to incorporate covariates) (Harris, 2014).

The basic ERGM probability mass function for binary valued networks measured at a single time point is as follows (Hunter & Handcock, 2006):

$$Pr_{\theta:\eta,g}(Y = y|x) = \frac{\exp(\eta(\theta) * g(y;x)}{K_{\eta,g}(\theta;x)} \quad , \qquad (1.1)$$

where the probability observing a tie in network *y*, conditional on the network properties, *x*, is an exponential function of product of the vector of *k* model parameters to be estimated, $\theta$, and a vector of change statistics associated with the parameters and network properties modeled, divided by a normalizing constant, κ (denoted "c" sometimes) to ensure the model is estimable.

The model is estimated in log-odds (logit) form using a maximum likelihood estimation algorithm. The most notable differences between a logistic regression and an ERGM coefficient interpretation is that, in ERGMs, the predictor slope values are the expected change in the likelihood of a tie forming for the *sum* of the characteristics for each node in the dyad, and that all coefficient values are *conditional on the network properties* modeled.

Researchers, based on theoretical understandings of their nodes and the network (and potentially also the observed cluster structures among nodes), must decide which characteristics of nodes and network structures are important for inclusion in the model. Indeed, the ERGM allows for both node and structural effects to be considered simultaneously as a representation of the social processes. Unfortunately, this simultaneous modeling can also cause "degeneracy" issues (cf., Harris, 2014, p. 25), which results in model non-convergence. The most common occurrences of this happen with either sparse networks with very few ties, or saturated networks with nearly all possible ties (Handcock, 2003b; Snijders, 2002). Yet, despite the degeneracy issues and the somewhat difficult interpretation of the model slope coefficients, benefits from using ERGM may outweigh the risks – it is similar to logistic regression in many regards and quite flexible at handling both network structures and node- or dyad-level covariates. Indeed, as the literature review revealed, ERGM is the more popular model used for inferential SNA.

**Summary**

This chapter focused on a broad overview of social network analysis (SNA), including network data terminology, the prevalence of different SNA data and approaches in educational research, and a brief review of the most popular modeling family – ERGMs. We have yet to describe the issues involved in handling missingness in social network data analysis, however. The next chapter reviews missing data theory and how it relates to SNA.

# CHAPTER 2.

## Missing Data: Patterns, Mechanisms, and Approaches

This dissertation focuses on missing data handling approaches for social network data. While the previous chapter described social network analyses in broad terms, it did not delve into the issues involved in missing data handling. As such, the present chapter delves into missing data theory in general, and then describes the issues for social network analyses – ultimately leading to the research question motivating this dissertation.

Missing data can occur in any research but especially that involving human subjects, both as a matter of seemingly random circumstances as well as due to research processes themselves. In online survey research for example, missingness may occur when people never receive the survey (i.e., it is in their junk or spam email folder) or when they experience technology failures during the survey. These may be thought of as "random" causes of missingness.

Other missingness is not as random in nature. For example, participants in a longitudinal study may be more likely to move from the research location as time goes on, and therefore missingness may be expected the longer the time period a study spans (i.e., missingness related to time). A more problematic example of missingness is when study conditions themselves cause missingness: for example, control group participants in a randomized study who become worse on the outcome(s) may drop out of the study because they no longer can (or want to) participate; in this case, higher levels of missingness will be related to both the independent variable (study condition) and the dependent variable(s).

The simplest approach to handling missingness is to drop cases that are missing data from analyses, also known as **listwise deletion** (Bodner, 2006, as cited in Enders, 2010); in fact, this is the default procedure for handling missingness in models estimated with ordinary least squares

in any major statistical software package. Nevertheless, unless the missingness is unrelated to the outcome – directly, or indirectly – dropping cases can result in biased parameter estimates of the independent variable(s) effects. In other words, dropping cases could result in inflated Type I error or decreased power, depending on specific circumstances. Further, even if the missingness is unrelated to the outcome, dropping cases will always result in a smaller sample size, which in turn yields larger standard errors and decreased power.

**Types of Missingness**

Before probing further into approaches to handling missingness, it is instructive to review missing data theory, a now well-developed scholarly area of quantitative methods. Schafer and Graham (2002) discuss two basic types of missing data: item nonresponse and wave nonresponse. **Item nonresponse** occurs when a respondent completes a part of a survey or questionnaire (some of the items), but fails to complete other parts. This type of nonresponse can happen when a respondent accidentally skips a question or when the participant purposefully skips a question and either cannot respond to it (for example, when an item is poorly constructed or inappropriate), forgets to return to it (for example, may have wanted time to think about their answer), or has no intention of responding (for example, concerned about privacy). Yet other reasons can include technology or equipment failures or when the research design itself uses planned missingness to reduce survey burden (known as missing by design).

The second type of missing data is **wave nonresponse**. This applies to both cross-sectional and longitudinal research, where a "wave" represents a data collection time point. Wave nonresponse happens when a respondent does not complete an entire survey or set of measures at a given time point. In longitudinal research, this often occurs when a respondent completes the first data collection wave but then drops out of the study at a later date (also

known as attrition); however, sometimes wave nonresponse occurs when respondents enter a study late and are missing one or more earlier data collection time points.

Enders (2010) takes a more nuanced approach to classifying missingness patterns, as distinct from missing data mechanisms (causes), by using a model-based lens that keeps in mind that most studies utilize multiple measures of independent (exogenous) variables, or Xs, and/or multiple measures of dependent (endogenous) variables, or Ys. (Note that the term 'variables' includes items on a survey questionnaire or assessment, not just scale scores.) Specifically, he divides missingness on dependent variables into **six missingness patterns**: univariate, unit nonresponse, monotone, general, planned, and latent variable (pp. 2-5).

A univariate pattern is one in which only one variable is missing data, a relatively rare situation but yet one that methodologists have often studied for its simplicity. Unit nonresponse, on the other hand, is most common: this is when missingness occurs more globally for a set of participants (the "units"), including the examples for item nonresponse described above. A monotone pattern is one in which there is a steady increase in missingness, which can occur often in longitudinal designs. The fourth type, the general pattern, involves missingness that appears to be randomly dispersed. Planned missingness, the fifth pattern, is common in large-scale surveys and assessments to reduce respondent burden; in this design, typically everyone has data on an "anchor" variable but the other variables are missing equal proportions of non-overlapping data. Last but not least, the latent variable pattern is unique to structural equation models: the latent variable is missing all information and data from the observed variables are used, in conjunction with maximum likelihood, to estimate the "missing" information.

**Missingness Mechanisms**

Missing data patterns are not the same as missing data mechanisms. Missing data mechanisms constitute the 'causes' of missingness via hypothesized relationships between observed and missing data (Enders, 2010, p. 3). Rubin (1976) introduced three missingness mechanisms: Missing at Random, Missing Completely at Random, and Missing Not at Random. It is the mechanism of missingness that informs appropriate missing data handling.

**Missing at random (MAR).** Data are said to me missing at random (MAR) when the *probability of missing data* on the dependent variable, Y, is not related to the *values* of Y, even if it is related to other measured variables (Xs), so long as those X variables are included in the model (Allison, 2009; Enders, 2010). In other words, subjects with lower scores on Y should be equally likely to be missing data on Y as subjects with average or higher scores on Y. Formally, the definition of MAR is:

$$\Pr\left(Y_{missing} \middle| Y_{observed}, X\right) = \Pr(Y_{missing}|X), \quad (2.1)$$

which states that the probability of missing data on Y, conditional on the observed X and Y values, is equal to the probability of missing data on Y controlling for X alone (Allison, 2009).

Another useful way to express this definition is as a probability distribution:

$$p(R|Y_{observed}, \phi), \quad (2.2)$$

where *p* is the probability distribution of *R*, *R* is a binary variable that takes on a 1 if Y is observed and 0 if missing, $Y_{observed}$ is the observed outcome, and $\phi$ is a parameter or set of parameters that describe the relationship between *R* and the data (Enders, 2010).

As an example, MAR would be satisfied if a researcher predicting school belongingness and finds that Asian students have higher rates of missing data on this variable than White students. As long as (a) there is no relationship between the outcome, school belongingness (Y),

and race categories (X) (e.g., there is no difference between race categories on school belongingness levels), and (b) the researcher plans to include race as a predictor in the model, then MAR can be assumed because there is no residual relationship between the propensity for missing data and the dependent variable, after controlling for race.

**Missing completely at random (MCAR).** The Missing Completely at Random mechanism (MCAR) requires that the probability of missing data on the dependent variable Y is unrelated to other measured variables (Xs) as well as values of Y (Enders, 2010). Another way of stating this is the observed data is a simple random sample of the scores that would have been analyzed had the data been complete. Since MCAR requires missingness to be completely unrelated to any of the data, both $Y_{observed}$ and $Y_{missing}$ are unrelated to *R*. The probability distribution can be expressed as:

$$p(R|\phi), \tag{2.3}$$

where *p* is the probability distribution of *R*, *R* is a binary variable that takes on a 1 if Y is observed and 0 if missing, and ϕ is a parameter or set of parameters that describe the relationship between *R* and the data (Enders, 2010).

As an example, MCAR would be satisfied if a researcher predicting school belongingness finds that none of the measured variables (Xs) were related to the probability of missingness on school belongingness. This would be a situation in which the "holes" in the data were distributed in a haphazard fashion, such as when students are randomly absent or some technology failure occurs during data collection.

**Missing not at random (MNAR).** In both MAR and MCAR, it is assumed that the likelihood of *missingness* on Y was unrelated to the *values* of Y. In the examples given above, this assumption would not be satisfied if students missing data are those that also have lower

school belongingness scores. Data is considered Missing Not at Random (MNAR) when the probability of missing data is related to the values of Y, even after controlling for other variables (Xs) in the model. Again using Enders (2010) notation, the probability distribution for *R* assuming MNAR is written as:

$$p(R|Y_{observed}, Y_{missinn}, \phi), \quad (2.4)$$

where *p* is the probability distribution of *R*, and *R* is a binary variable equal to 1 if Y is observed and 0 if missing, $Y_{observed}$ is the observed outcome, and $Y_{missing}$ is the missing part of the data; $\phi$ is a parameter or set of parameters that describe the relationship between *R* and the data.

**Missing Data Handling Approaches**

On a practical level, researchers rarely can know with certainty why data are missing, making it impossible to fully describe the missingness mechanism. Nevertheless, missingness mechanisms and analytic techniques play a crucial role in accurate parameter estimation. Rubin (1976) formally showed that models employing likelihood-based methods, including full information maximum likelihood (FIML) estimation and multiple imputation, do not require any knowledge about $\phi$ if the missingness can be considered MCAR or MAR. In other words, data that are MCAR or MAR have "ignorable" missingness, so long as data with MAR include covariates correlated with the probability of missingness (Allison, 2009). In contrast, only data for which MCAR can be assumed can rely solely on estimating parameters using a sampling distribution based on the observed data; in other words, non-likelihood based models that employ listwise deletion (e.g., ordinary least squares regression on observed data) should only be used when MCAR can be assumed (Enders, 2010). Last but not least, MNAR data must be handled with great care and cannot rely on likelihood-based methods alone for parameter estimation.

**Traditional missingness handling methods.** Despite what is known about missing data amongst methodologists, traditional methods such as listwise deletion, pairwise deletion, and single imputation, remain prevalent in applied research – especially listwise deletion (e.g., Peugh & Enders, 2004). **Listwise deletion** is completed by deleting from the sample data any cases that have missing data on any variables used in analyses. This method is also what is used by default in many statistical programs. There are two main advantages to using this method: it can be used for any kind of statistical analysis, from a basic 2-group *t*-test to a complex structural equation model (Allison, 2009), and it will result in unbiased parameter estimates if the data are MCAR, since reducing the sample to those with complete data can be considered a random subsample of the original sample (albeit lower power due to reduced *N* and therefore larger standard errors).

**Pairwise deletion** is a missing data handling method that focuses on analyzing data using the variance-covariance matrix of variables, rather than the original people-variable data matrix. In this method, each element (variable) of the matrix is estimated from all data available; in other words, the variances of the variables will depend on those who responded to those respective variables only, and the covariances will depend on the set of individuals who responded to each of the pairs of variables. In theory, this method seems like it would work well because it makes use of all available data. However, because the variances and covariances of the variables are based on different combinations of subsamples, parameter estimates will be biased unless MCAR holds (Allison, 2009; Schafer & Graham, 2002). Indeed, if the data are MCAR, pairwise deletion is preferable over listwise deletion given that it naturally includes more information (Enders, 2010). Nevertheless, even if the data are MCAR, pairwise deletion also suffers from lower power compared to likelihood-based model estimates (Schafer & Graham, 2002).

Finally, **single imputation** methods involve replacing missing values with real numeric values, creating a complete dataset for use in analyses (this differs from multiple imputation, which fills in missing values with random draws from a distribution, and then repeats the process to create multiple complete datasets that are then analyzed and pooled a particular way). These methods include mean substitution, regression imputation, stochastic regression imputations, hot-deck imputation, and last observation carried forward. Although convenient, except for the stochastic regression imputation, single imputation methods do not take into account imputation error. As such, singly imputed data are treated as if they are real data, causing parameter standard errors to be underestimated, in turn leading to Type I error inflation (Enders, 2010).

**Likelihood-based missingness handling approaches**. Modern approaches to handling missingness include maximum likelihood estimation and multiple imputation methods. The goal of **maximum likelihood estimation** is to identify the population parameter values that have the greatest probability of producing the observed sample of the data (Schafer & Graham, 2002). When the missing data mechanism is MCAR or MAR, researchers can obtain the likelihood by summing the observed likelihood over all possible values of the missing data, which allows for unbiased parameter estimates and their standard errors.

In order to use maximum likelihood estimation with missing data, an expectation-maximization (EM) algorithm is required. The EM algorithm is a two-step process to obtain missing data values, where E is the expectation step and M is the maximization step (Allison, 2009). This iterative process begins with an initial estimate of the mean vector and the covariance matrix. First, using the available observed data, the maximum likelihood estimates of the means, variances, and covariances need to be calculated with ordinary least squares equations, like those given below (computational versions for a 1-predictor model shown).

$$\hat{\mu}_Y = \frac{1}{N}\Sigma Y \quad (2.5)$$

$$\hat{\sigma}_Y^2 = \frac{1}{N}\left(\Sigma Y^2 - \frac{(\Sigma Y)^2}{N}\right) \quad (2.6)$$

$$\hat{\sigma}_{X,Y} = \frac{1}{N}\left(\Sigma XY - \frac{\Sigma X \Sigma Y}{N}\right) \quad (2.7)$$

The E step fills in the missing values, so the M step can use the above equations to generate parameter estimates. The E step uses the elements of the mean vector and covariance matrix to build regression equations to predict values for the missing data from observed variables. As an example, the necessary equations for a 1-predictor model assuming the dependent variable contains missing data are as follows.

$$\hat{\beta}_1 = \frac{\hat{\sigma}_{X,Y}}{\hat{\sigma}_X^2} \quad (2.8)$$

$$\hat{\beta}_0 = \hat{\mu}_Y - \hat{\beta}_1\hat{\mu}_X \quad (2.9)$$

$$\hat{\sigma}_{Y|X}^2 = \hat{\sigma}_Y^2 - \hat{\beta}_1^2\hat{\sigma}_X^2 \quad (2.10)$$

$$\hat{Y}_1 = \hat{\beta}_0 + \hat{\beta}_1 X_1 \quad (2.11)$$

where $\hat{\beta}_0$ is the intercept, $\hat{\beta}_1$ is the slope, $\hat{\sigma}_{Y|X}^2$ is the residual variance from the regression of Y on X, and $\hat{Y}_1$ is the predicted Y score for a given value of X (Enders, 2010).

The M step applies complete data formulas to the filled-in data to generate updated estimates of the mean vector and the covariance matrix. The algorithm then updates the parameter estimates and continues to iterate the E step again, generating new regression equations to predict the missing values. The process continues until the algorithm converges on the maximum likelihood estimates. Importantly, the EM algorithm does not magically replace or impute missing data, but instead uses all available data to estimate the parameters and their variances and covariances (Enders, 2010).

Although this method will produce more accurate estimates compared to traditional methods, it is not perfect and will produce biased estimates when the data are MNAR. Further, the method is limited to models that employ likelihood-based estimation (not all do), and situations in which there is only missingness on the dependent variables, not predictor variables.

**Multiple imputation** is the other modern approach to handling missingness, developed in part to overcome the limitations with maximum likelihood estimation. As its name implies, holes in a dataset can be filled in using draws from a specified multivariate distribution, and then this process can be repeated for obtaining a desired number of "complete" datasets (the number of datasets corresponds monotonically to the amount of missingness in the original data to assure unbiased in parameters as well as efficient standard errors so that results can be consistently replicated). Those complete datasets are each analyzed separately, and thereafter, results are pooled to ensure they incorporate both the within- and between-imputation error (Allison, 2009).

Multiple imputation is based on the Bayesian theoretical perspective (Enders, 2010; Schafer & Graham, 2002), which permits treating parameters as random, rather than fixed – an essential part of the multiple imputation process (Schafer & Graham, 2002). Similar to maximum likelihood estimation, the multiple imputation process uses an iterative algorithm to cycle between an imputation step and posterior step (Enders, 2010). The imputation step utilizes stochastic regression to impute the missing values (draws from a multivariate distribution constructed from the observed data estimates), while the posterior step uses the filled-in data to generate new estimates of the mean vector and covariance matrix until it reaches a given criterion (Schafer & Graham, 2002).

Although multiple imputation allows researchers to use just about any model on the "complete" datasets, obtaining the imputed values requires specialized software and the pooling

of the results estimates can be cumbersome, particularly for complex models (Allison, 2009). It is likely these for these reasons that multiple imputation is not used more often.

**Missingness Handling in Social Network Models**

Finally, we turn to missing data issues in social networks specifically. As described in Chapter 1, surveys are often used to measure social networks. In network surveys, respondents are typically asked to make a list of individuals with whom they connect (for a variety of reasons, such as professional advice, collaborations, or friendships), and they may also be asked to rate the frequency with which they interact with their nominations. Like any survey, participants may be absent or experience technology failures that prevent them from responding. They may also simply be unmotivated to respond (often we will not know). However, unlike regular surveys, another manner that missingness can arise in network measurement is when respondents nominate people who were not on the original survey list. Further, in longitudinal studies, respondents may come into the network late or leave the network early, in addition to the other ways that missingness might arise. How might missingness be appropriately handled when one wishes to model the network as a whole?

Turning back to the literature review described in Chapter 1, I examined the subset of articles that employed model-based SNA to classify each article by how it handled missingness. Of the 54 educational studies that employed SNA, 19 (35%) used a model-based approach; of these 19 studies, 15 (79%) used listwise deletion (dropping nodes with missingness); the remainder used some form of single imputation (random or inserting a particular value, 10.5%), or used likelihood-based estimation (10.5%). The rates were similar in the interdisciplinary journal as well: of the 123 studies that used model-based SNA, 70% used listwise deletion and

20% used some form of imputation; 10% used a likelihood-based approach. Overall, there was a strong tendency for researchers to use listwise deletion.

It may come as no surprise that listwise deletion is the major default setting for estimating binary ERGMs in R `statnet`. The other available missing data handling approaches include three types of single imputation methods: insert all 0s (assumes the node has no ties with any other node in the network), insert all 1s (assumes the node has a tie with all other nodes in the network), and for multiple time point models, one can insert values from a previous or subsequent time point (termed "borrowing auxiliary information."). None of these four approaches (listwise deletion or single imputation without a stochastic process) is appealing, given the findings from missing data theory.

To date, four studies have investigated missingness handling in social networks, all of which have focused on use of the ERGM family. The first was by Robins, Pattison, & Woolcock (2004), who conducted an exploratory study of four approaches to handling missingness using two existing datasets: listwise node deletion (assumes MCAR), constraining nonrespondent nodes' ties to be "homogeneous" (MAR), and then two alternative ERGMs that directly incorporate missingness as a node-level covariate (e.g., to handle MAR and MNAR). They randomly and non-randomly deleted nodes then examined whether the different approaches reproduced certain structures in the complete data (e.g., triads). On the basis of their findings, they argue that incorporating information about nodes and the structure of the network should be used to estimate ties for the entire network, including nonrespondents. Similar to Robins et al. (2004), Handcock & Gile (2010) developed a framework for taking into account sampling error (i.e., nonresponse) and then applied their framework to real data, ultimately making the case that social network surveys could employ planned missingness designs so that MCAR can be

assumed. Importantly, both of these studies made use of limited existing data to demonstrate their ideas, and as such, it is unclear how their results generalize to a population of networks; further, neither provided practical solutions for applied researchers using ERGMs.

The last two studies focused on investigating stochastic imputation methods. Ouzienko & Obradovic (2014) demonstrated that imputations that incorporate all available information across time points could be used for longitudinal network research surveys during the modeling process, rather than single imputation or by using a "missingness" covariate like Robins et al. (2004). However, their methodology is restricted to longitudinal data and again is missing practical solutions for applied researchers. Last but not least, Smith et al. (2017) conducted an extensive study of bias resulting from MNAR situations (i.e., when very central nodes are missing vs. non-central nodes). The problem of course is that researchers cannot know when a highly central person is missing from their network, unless they have covariate information serving as a proxy. Although Smith et al. go on to develop an app to estimate bias from missingness, they do not offer practical solutions for dealing with MCAR/MAR data.

In summary, while there has been a handful of studies on specific issues that can occur due to missingness in network data, no one has systematically studied the bias resulting from default missingness handling methods in `statnet`, which is the free, popular package for estimating a wide range of ERGMs. Moreover, no one to date has offered researchers a practical solution involving modern missingness handling methods for reducing bias in coefficient accuracy and precision.

**Overarching Research Question**

Taking into account the prior research, this dissertation contributes to the literature by systematically investigating default missingness approaches for ERGM estimation in the popular

*R* package `statnet` as compared with three stochastic single-imputation methods. Specifically, the overarching research question was: *Under what conditions are different missing data handling procedures best for social networks modeled using ERGM?* Although each of the three forthcoming experiments assumes MCAR, this work uses simulation, not observed datasets, for evaluating parameter recovery and standard error bias, and has practical solutions for end users. The first two studies specifically examine situations in which the data collected are binary (the presence or absence of a tie among nodes); the third investigates situations in which the data collected are valued (e.g., some type of magnitude in the interactions among nodes). We expected, and found, that a stochastic process would work best. Thereafter, we illustrate a package for *R* called `netImp` that applied researchers can use for their missing data.

# CHAPTER 3.

## Study 1: Missing Data Handling for Single Time Point Networks with Binary Ties

As reviewed at the end of the prior chapter, modeling exponential random graph models (ERGMs) in `statnet` have two built-in missing data imputation methods: either node listwise deletion, or non-stochastic single-imputation (e.g., insert all 0s or all 1s). The present study undertakes this comparison, and further, is not limited to the two approaches that `statnet` uses. Instead, this study compares all of the (reasonable) potential missing data imputation methods that could be utilized in practice, including node deletion (common in practice), the two used in ERGM and three using probability-based methods (inserting a random mix of 0s and 1s based on equal- or population density- or sample density-based probabilities).

Specifically, the over-arching research question was: *Under what conditions are certain missing data handling approaches best for estimating binary network data with ERGM?*

## Method

### Data Generation

Simulated directed binary adjacency matrices were generated in *R* using our own code (see Appendix A) to reflect network characteristics based loosely on data from two science teacher professional development network projects: Ambitious Science Teaching (AST; Thompson et al., 2019) and Partnership for Ambitious Science Teacher Leaders (PASTL; Feldman et al., 2018), both funded by the National Science Foundation. In these projects, teachers nominated peer teachers, teacher leaders, school principals, and university researchers for pedagogical advice seeking and collaboration. For the current study, we manipulated network size (2 levels, a between-subjects factor), network density (3 levels, a between-subjects factor), missingness level (2 levels, a within-subjects factor), and missingness handling approach (6

levels, a within subjects factor). Specifically, there were 1,000 directed, binary matrices generated for each of the six combinations of network sizes and densities. For each of the 6,000 matrices, missingness levels and missingness handling approaches applied. Thereafter, an intercept-only ERGM was conducted on each of the 6000 x 2 x 6 = 72,000 datasets. The details for each of the conditions are as follows.

**Network size.** In the aforementioned applied projects (AST and PASTL), network sizes ranged from 20 to 60, depending on year of the study. As such, we set our simulated network sizes to either 20 or 50 nodes. These sizes are also consistent with many publicly available network datasets that are often used for demonstration purposes, including Sampson's monk data which had 18 nodes (Sampson, 1969), Lazega's law firm data which had 71 nodes (Lazega, 2001), and Florentine Family dataset which had 17 nodes (Breiger & Pattison, 1986).

**Network density**. The observed densities in the AST and PASTL applied projects were rather sparse, ranging from 2 to 6%. However, within a school, the densities were often greater because teachers tended to have more collaborations within their own building. Moreover, for the purpose of the present project, we wished to avoid degeneracy problems which can occur with very sparse networks. As such, for each of the two network sizes, we set the densities at modest levels, including 20, 30, and 40%. Further, for brevity, reciprocity (i.e., percent of ties formed that mutually nominate each other) was held constant at 50%, which was typical in the aforementioned applied projects.

**Missingness**. Across the simulated networks, there were two missingness levels induced (at random) *at the node level*: 10% and 20%, again in keeping with what was observed in the two applied teacher network projects.

**Missingness handling**. For every dataset, there were six missingness handling approaches employed, as follows:

1. node deletion (dropping cases with incomplete data; most common in practice);
2. replacing all missing values with 0s (a default option);
3. replacing all missing values with 1s (a default option);
4. replacing all missing values with a 50% probability of 0 or 1 (much like an uninformative/uniform prior in Bayesian statistics);
5. replacing all missing values with a probability of 0 or 1 based on the *true* density of a given particular network (i.e., prior to induced missingness); and
6. replacing all missing values with a probability of 0 or 1 that was based on the *sample* density of the network (i.e., what researchers would have available to them in reality).

Again, in total, there were 2 network sizes x 3 density levels x 2 missingness levels x 6 missingness approaches = 72 conditions applied to 1,000 simulated datasets. An intercept-only ERGM was conducted using the `statnet` package in *R* on each of these conditions. Additionally, an ERGM was also completed for each of the original networks with no missingness (1,000 networks x 2 sizes x 3 densities = 6,000 of these).

The ERGM-based intercept estimate and its standard error are interpreted in logits and can therefore be translated into predicted probabilities (of a tie forming in the network), as follows:

$$\Pr(tie) = \frac{1}{1+e^{((-1)*(intercept))}}. \quad (3.1)$$

Relative bias for both the intercept estimate and its standard error were then computed using results from the corresponding complete dataset as the true values; specifically, relative

bias was computed as the deviation between the estimated and true value, divided by the true value. We summarized these data descriptively and additionally evaluated the extent of which the study design factors contributed to differences in relative bias, especially in terms of missingness handling approaches, using mixed-model analyses of variance (ANOVAs) on the coefficient and standard error relative bias data. We also used the rule of thumb that less than 5% bias is typically considered acceptable for coefficients, and less than 10% is considered acceptable for standard errors (Hoogland & Boomsma, 1998).

## Results

### Data with No Missingness

Recall that the ERGM estimates for the complete networks were used as the "true" values for comparison with the estimates from missingness-handled datasets in order to compute relative bias. Table 3.1 reports the ERGM results for the intercept coefficient estimates and their standard errors for each of the network size-density combinations when data were complete.

**Table 3.1**

*Mean Intercept Coefficient and Standard Error Estimates for Networks with Complete Data*

| Condition | *Coeff* | *SE* |
|---|---|---|
| Smaller Network (20 Nodes) | | |
| 20% Density | -1.39 | 0.13 |
| 30% Density | -0.86 | 0.11 |
| 40% Density | -0.41 | 0.10 |
| Larger Network (50 Nodes) | | |
| 20% Density | -1.39 | 0.05 |
| 30% Density | -0.85 | 0.04 |
| 40% Density | -0.41 | 0.04 |

*Note*. $N$ = 1,000 adjacency matrices for each of the six conditions; all values in logits.

As one might expect, there was little variation in the coefficient estimates (predicted density, in logits) for the network sizes, but the standard errors were affected as one would

expect (smaller networks had standard errors over twice the size of that of the larger networks). More importantly, the coefficient estimates (in logits) showed good estimation of the desired data generation process: using the formula shown in Table 3.1, the predicted probability of a tie in the network for the 20% density condition across network sizes (and networks) averaged 19.94%; for the 30% and 40% density conditions, the results translated to predicted probabilities of 29.73% and 39.89%, respectively.

**Data with Missingness**

**Overall results**. Table 3.2 summarizes intercept coefficient and standard error relative bias for networks with missingness induced, by missingness handling approach.

**Table 3.2**

*Mean Relative Bias by Missingness Handling Approaches across Conditions*

| Missingness Handling Approach | *Coeff* | *SE* |
|---|---|---|
| Node Deletion | <1% | 11% |
| Inserting all 0s | 35% | 5% |
| Inserting all 1s | -63% | -7% |
| 50% Density-Based Random Imp. | -17% | -3% |
| True Density-Based Random Imp. | <1% | <1% |
| Sample Density-Based Random Imp. | <1% | <1% |

*Note.* $N$ = 12,000 estimates per missingness handling approach, collapsed across network size, density, and missingness levels.

As can be seen, node deletion produced close to zero bias for the coefficient, but had the largest standard error inflation of all the approaches. This method, therefore, would lead to increased Type II error and decreased power. Although the other approaches, on average, did not result in substantial standard error bias (i.e., less than 10% rule of thumb; Hoogland & Boomsma, 1998), there were notable patterns in the coefficient bias. Inserting all 0s over-estimated the coefficient (recall this is in logits, so there would be an artificially low density level estimated by 35%), and inserting all 1s resulted in the opposite direction of bias

(overestimating the network connections by 63%). Using a random imputation based on 50% also biased the coefficient to slightly to overestimate the network connections (which makes sense because the true densities in the conditions ranged from 20 to 40%, which is lower than 50%). Most importantly, imputing data using a density based on the original complete network (true) or listwise deleted network (sample) had the least bias in both the coefficient and standard error; these latter two methods also did not differ significantly from each other (*t*-test *p*s > 0.05).

**Design effects on coefficient relative bias**. Mixed model ANOVA results for relative bias on the coefficient estimate showed large main effects for missingness level ($F(1, 5994) = 2324.56$, $p < .001$, $\eta^2 = .28$) and missingness approach ($F(5, 29970) = 187058.41$, $p < .001$, $\eta^2 = .97$), and a very small main effect for density ($F(2, 5994) = 34.73$, $p < .001$, $\eta^2 = .01$). Inspection of the marginal means revealed that conditions with greater missingness (20% missingness compared to 10%) had greater overestimation; conditions with greater density generally had more underestimation compared to those with lower density had slightly more overestimation.

More importantly, there were two significant 2-way interactions between approach and missingness level ($\eta^2 = .80$, $p < .001$) and approach and density ($\eta^2 = .83$, $p < .001$) on bias, as well as a significant 3-way interaction among these factors ($\eta^2 = .42$, $p < .001$). To understand the nature of these interactions, we plotted coefficient relative bias for each missingness handling approach by missingness level (Figure 3.1) and density (Figure 3.2). As can be seen in Figure 3.1, relative bias became more pronounced with increased missingness for the two non-stochastic single imputation approaches (inserting all 0s or all 1s) and the stochastic (uninformative) 50% imputation approach. (Recall that all three of these approaches had unacceptably high relative bias, irrespective of the level of missingness.) As can be seen in Figure 3.2, increased density exacerbated the coefficient bias observed in the two non-stochastic single imputation approaches

and decreased bias in the 50% stochastic imputation approach. Although not illustrated, the 3-way interaction among these factors revealed that the bias present in the problematic approaches was amplified when both higher missingness and higher density conditions were present.

**Figure 3.1**

*Relative Bias by Missingness Handling Approach and Missingness Level for Tie Coefficient*

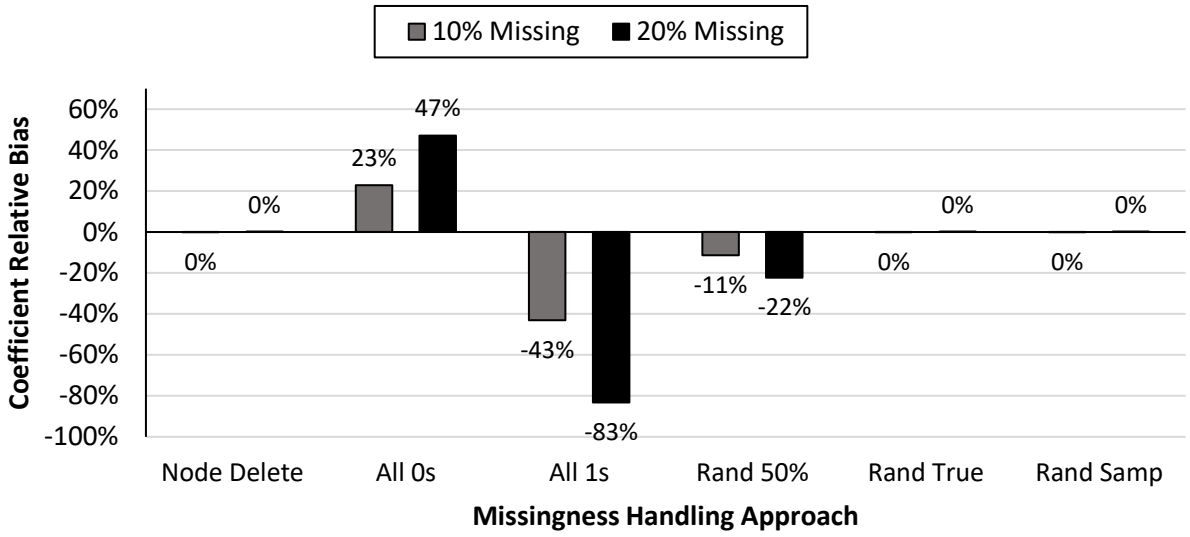
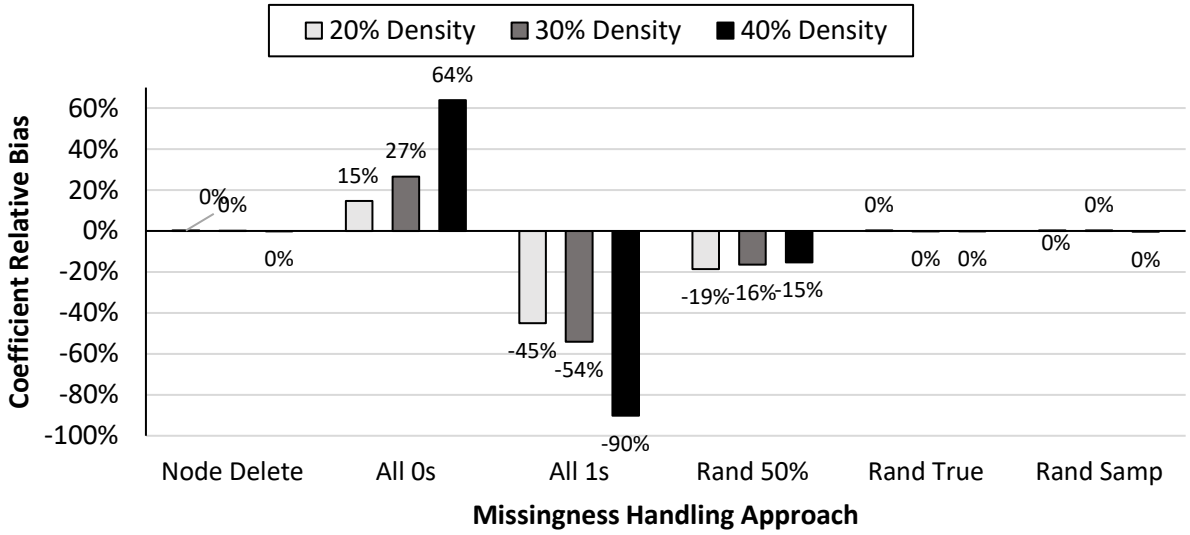



**Figure 3.2**

*Relative Bias by Missingness Handling Approach and Density Level for Tie Coefficient*



**Design effects on coefficient standard error bias**. To evaluate the extent to which the study design factors contributed to differences in the standard error estimate relative bias, we again conducted a mixed-model analysis of variance on the standard error relative bias data. The ANOVA results for relative bias on the coefficient standard error estimate showed very small main effects for missingness level ($F(1, 5994) = 25.96$, $p < .001$, $\eta^2 = .01$) and network size ($F(1, 5994) = 34.65$, $p < .001$, $\eta^2 = <.01$), and large main effects for missingness approach ($F(5, 29970) = 323635.05$, $p < .001$, $\eta^2 = .98$) and density ($F(2, 5994) = 7860.55$, $p < .001$, $\eta^2 = .72$). Inspection of the marginal means revealed that conditions with greater missingness (20% missingness compared to 10%) and a larger network size (50 nodes compared to 20) had slightly less standard error bias (however, this is across all handling approaches and there are important interaction effects noted next). More substantively, there were pronounced differences in standard error bias among missingness handling approaches (see again Table 3.1; only node deletion, the most popular approach to missingness handling, had unacceptably high bias), and density levels (greater density conditions had less bias).

There were a number of interactions found to be significant; however, most of them were quite small ($\eta^2 < .01$) and likely reflective of the extremely large sample size. As such, we focus our attention on the three 2-way interactions with effect sizes of $\eta^2 \geq .01$, including approach by missingness level ($\eta^2 = .70$, $p < .001$), approach by density ($\eta^2 = .87$, $p < .001$), and missingness level by density ($\eta^2 = .17$, $p < .001$). Again, to understand the nature of these interactions, we plotted mean coefficient standard error relative bias for each missingness handling approach by missingness level (Figure 3.3) and density (Figure 3.4), as well as a plot for missingness level by density collapsed across approaches (Figure 3.5). As can be seen in Figure 3.3, relative bias became more pronounced with increased missingness for all approaches except the two

stochastic density-based imputation methods. (Recall that only node deletion had unacceptably high relative bias greater than 10%.)

**Figure 3.3**

*Relative Bias by Missingness Handling Approach and Missingness Level for Tie Standard Error*

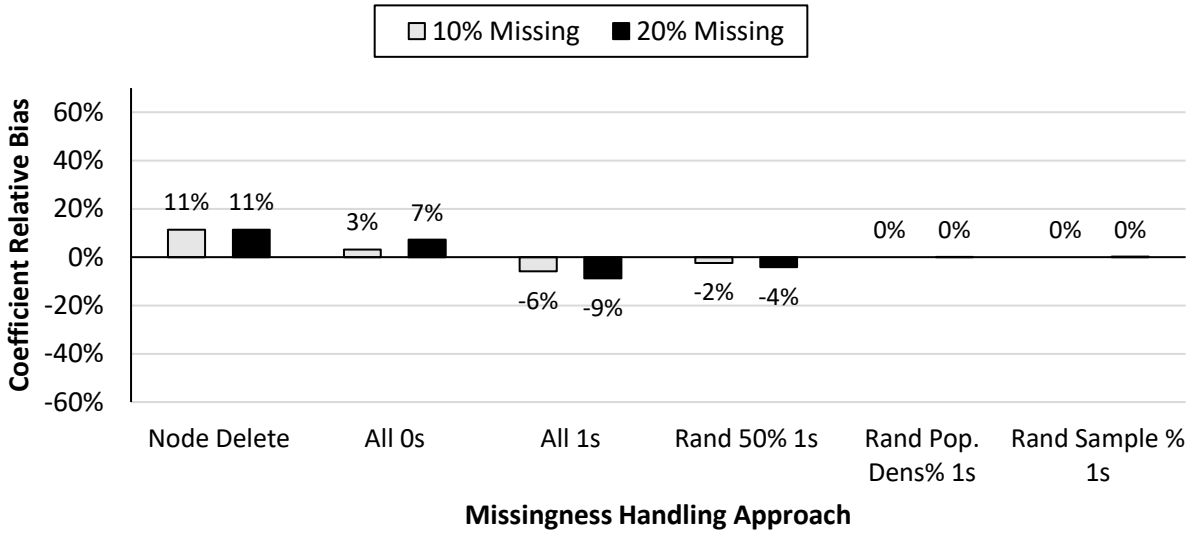
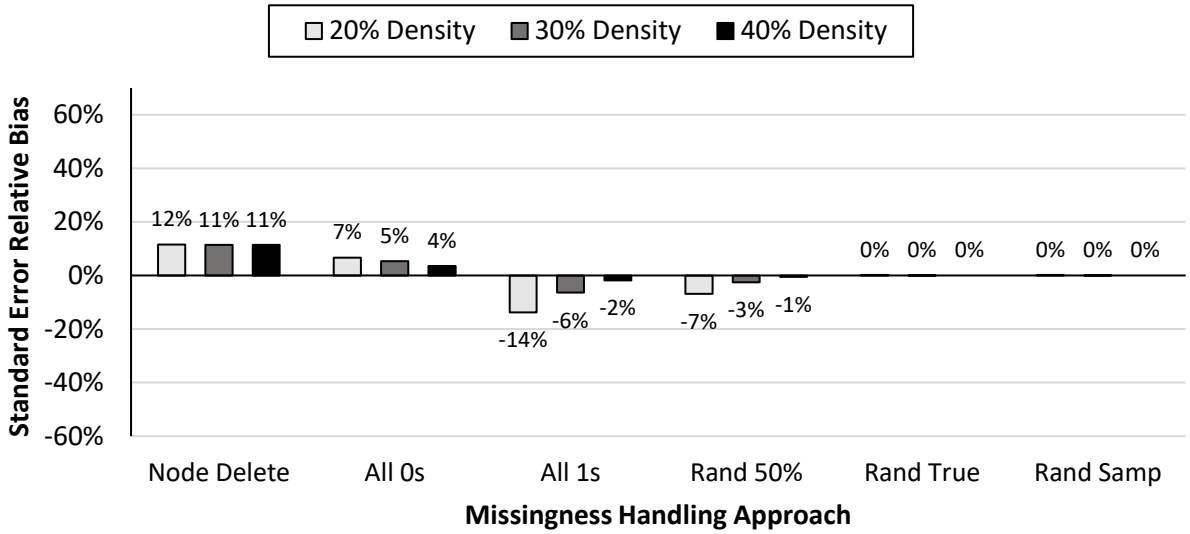



**Figure 3.4**

*Relative Bias by Missingness Handling Approach and Density Level for Tie Standard Error*



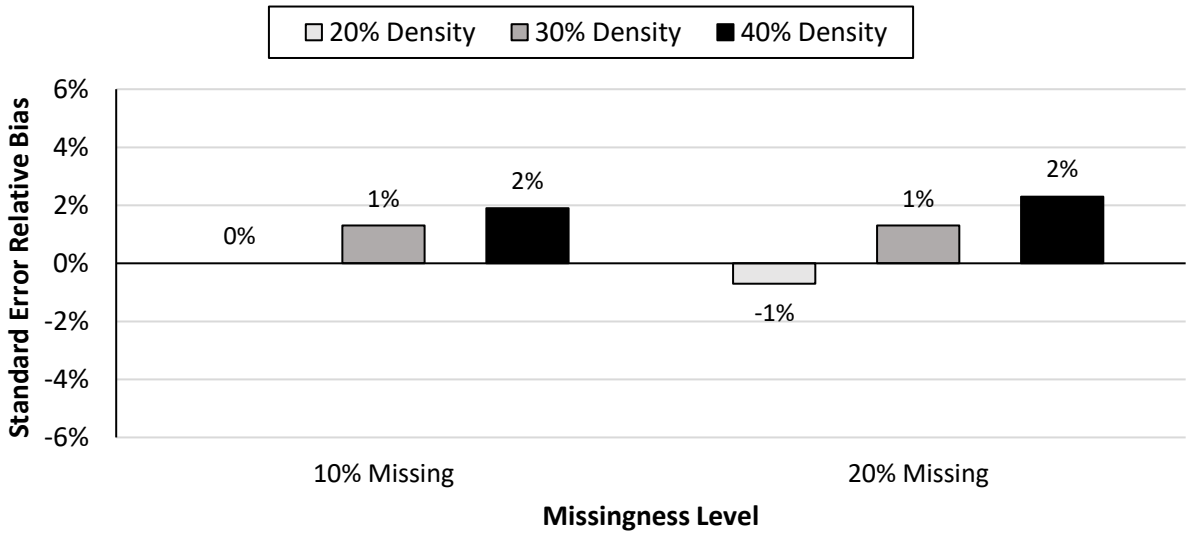

As can be seen in Figure 3.4, networks with greater density exhibited less bias for the four

approaches that had any bias (i.e., there was no bias and no effect of density on bias for the two stochastic density-based imputation approaches).

Last but not least, Figure 3.5 below displays the 2-way interaction between missingness and density on standard error bias, irrespective of approach to handling missingness. As can be seen, there is mostly just two main effects being displayed: increased missingness and increased density both leading to (slightly) greater standard error overestimation. Indeed, the interaction appears to be driven by a slight underestimation of the standard error for conditions in which there is more missingness and lower density. Again, this finding is irrespective of missingness handling approach, and not the focus of this project.

**Figure 3.5**

*Relative Bias by Missingness and Density Levels for Tie Standard Error*




**Next Steps**

Study 1 results showed the promise of using a modern stochastic single-imputation approach to handling missing data for binary, directed networks measured at a single time point. Although node deletion has been a popular approach to handling missingness in social network

research, it revealed unacceptably high inflated standard errors – 11% overestimated, on average. Nevertheless, node deletion is a better option than any of the non-stochastic single-imputation approaches like inserting all 0s or all 1s, as those created quite extreme bias in the coefficient estimates even if the standard errors were relatively unharmed. The next chapter extends the study of imputation approaches by examining similar conditions for a two time-point network scenario in which both tie formation (density) and tie dissolution (persistence) are estimated.

# CHAPTER 4.

## Study 2: Missing Data Handling for Two Time Point Networks with Binary Ties

Fairly recently, statisticians have extended SNA to incorporate **temporal** social networks (i.e., multiple time points that are used to estimate overall network parameters – not to be confused with modeling network change over time, per se). First, the ERGM was extended as a "temporal" ERGM (TERGM) estimated using the `btergm` package within the `statnet` suite in *R* (Leifeld, Cranmer, & Desmarais, 2018). This model estimates both tie *formation* (likelihood of a new tie forming in the network, conditional on network characteristics) and tie *dissolution* (likelihood that a new tie formed will persist) after taking into account the network characteristics at prior time point(s), much a pretest covariate in a posttest analysis of covariance. For example, if a principal collaboration program aimed at improving teacher retention is implemented at the beginning of the school year, this model could be used with data collected at two time points to predict whether future implementations of the program have a high likelihood of forming new ties, and whether the new ties formed might persist over the course of a year. Even if the probability of new tie formation turns out to be lower than hoped for, if persistence likelihood is high, such a finding would indicate that the ties that are formed are strong.

Despite its advancement, TERGM suffers from estimation and interpretation problems because the two focal parameters, tie formation and dissolution, are conflated. Krivitsky and colleagues developed a model that was able to overcome these problems using a "separable" TERGM (known as STERGM) using the `tergm` package within the `statnet` suite in *R* (Carnegie, Krivitsky, Hunter, & Goodreau, 2015; Krivitsky & Handcock, 2014, 2021). The model is really *two* models in which the first estimates the probability of observing a *new* tie forming (from a dyad that did not have a tie previously) in the most recent network (time *t*),

conditional on the previous network ($t - 1$). Separately, the second models the probability that an already formed tie from the previous network ($t - 1$) persists in the most recent network (time $t$). Specifically, the tie *formation* model is as follows.

$$\boldsymbol{Pr}\left(\boldsymbol{Y^t = y^t \middle| Y^{t-1} = y^{t-1};\ \theta}\right) = \frac{\boldsymbol{exp}(\boldsymbol{\eta(\theta)} * \boldsymbol{g}(\boldsymbol{y^t}, \boldsymbol{y^{t-1}}))}{\boldsymbol{\kappa_{\eta^+.g^+}(\theta^+, y^{t-1})\kappa_{\eta^-.g^-}(\theta^-, y^{t-1})}} \qquad (4.1)$$

In equation 4.1, $\eta(\theta)$ is a vector of $k$ model parameters to be estimated; $g(y^t, y^{t-1})$ is a vector of $k$ change statistics (associated with $\theta$) based on the tie forming at $t$ time point, conditional on tie forming at the prior time point ($t - 1$). The denominator is a normalizing constant meant to keep the model estimable. The tie persistence/dissolution model is similarly specified (Koskinen, Caimo & Lomi, 2015).

It is assumed that tie formation and persistence are constant over time – in other words, that tie formation and persistence do not interact with time, if there are more than two time points. This is because both models look at the observed differences between time points and not the changes within the network between time points (Krivitsky & Handcock, 2014). To be clear, STERGM does not quantify how a network at one given time point *differs* from that same network at another time point. Last, like any binary ERGM, the results of this model will be in log-odds units.

**Missing Data in Multiple Time Point Network Analyses**

In order to estimate a STERGM (or TERGM for that matter), researchers must choose how to handle missing data a priori. As is well known, missing data is more likely to occur in longitudinal studies, especially studies of people. Some individuals may enter a study late (thereby missing data at the start of the study) while others may drop out, typically when people change locations (thereby missing data at the end of the study). As has already been discussed in Chapter 2, dropping all individuals who are missing any data (i.e., listwise deletion) is a popular

approach in practice; however, as observed in Chapters 2-3, it is ill advised to do so because, even under the assumption of missingness completely at random (MCAR), the standard errors of the parameters will naturally be inflated due to the necessarily smaller sample size that accompanies deleting cases with missing data. Additionally, Study 1 findings indicated that other approaches that are default methods in `statnet` (inserting all 0s or all 1s) are also problematic in terms of biased coefficient estimates. Importantly, the STERGM package also has another default option, which we term "using auxiliary information." This approach to handling missingness takes information from a previous or subsequent time point to insert into the missing values where they occur. For example, if we are missing someone's tie information from the second wave of data collection, we could use their tie information from the first wave to insert into that person's missing information in wave 2.

### Research Question

The present study undertakes a comparison of the accuracy of STERGM estimates for different missingness handling approaches for networks measured across two time points, similar to Study 1. In addition to the approaches used in Study 1, Study 2 also incorporates another missingness handling approach: the use of auxiliary information from prior or subsequent time points. Specifically, our over-arching research question was: *Under what conditions are certain missing data handling approaches best for estimating two time point binary social network data with STERGM?*

## Method

### Data Generation

Similar to Study 1, simulated directed binary adjacency matrices were generated in *R* using our own code (see Appendix B) to reflect network characteristics based loosely on data

from two science teacher professional development network projects: Ambitious Science Teaching (AST; Thompson et al., 2019) and Partnership for Ambitious Science Teacher Leaders (PASTL; Feldman et al., 2018). Recall that for these projects, teachers nominated peer teachers, teacher leaders, school principals, and university researchers for pedagogical advice seeking and collaboration. In both studies, teachers were measured across multiple years in order to capture network conditions over time, with handfuls of missing data for teachers who either came into the study later or who left the study before it ended.

Similar to Study 1, we again manipulated factors related to missingness level, network size, network density, and approaches to missingness. However, in this study we reduced the number of density levels to two instead of three levels, and assumed networks were measured at two time points instead of one and therefore included four additional missingness level patterns (combinations among time 1 and time 2). Further, we also included an additional approach to handling missingness (use of auxiliary information). We generated $N$ = 1,000 directed, binary matrices for each of the four combinations of network sizes and time 1 densities. For each of the 4,000 matrices, 6 missingness levels and 7 missingness handling approaches applied. Thereafter, an intercept-only STERGM was conducted on each of the 4000 x 6 x 7 = 168,000 datasets. The details for each of the conditions are as follows.

**Network size.** The same network sizes were used as those in Study 1: either 20 or 50 nodes (see Study 1 for rationale).

**Network density**. We used the same network densities with 50% mutuality as those in Study 1, except that we dropped the last level for brevity given the added conditions for the present study; specifically, for each of the two network sizes, we set the initial densities at 20% and 30%. We assumed a constant 10% increase in density from the first time point (T1) to the

second (T2), which is in alignment with the assumption of STERGM that time does not interact with network properties. Importantly, this 'growth' in density was generated at random (two different networks generated at random with the same number of nodes but different density levels); in other words, an individual who is nominated at T1 was not necessarily continued to be nominated at T2. Given that the networks used as a pair were actually independently drawn, and given that ties are binary values, tie formation and tie persistence should be approximately equal to the average of the pair of network's densities.

**Missingness.** Across the simulated networks, there were six missingness levels induced (at random) *at the node level*: 10% at T1 or T2, 20% at T1 or T2, 10% missing at both T1 and T2, and 20% missing at both T1 and T2. These conditions reflect the real-world combinations of missingness one might observe in a longitudinal study in which individuals may be missing at any given time point. Note that, for the purpose of this study, we constrained the missingness to ensure that no individual node was ever missing data at both T1 and T2.

**Missingness handling.** For each of the generated datasets, we applied seven missingness handling approaches, as follows:

1. node deletion (dropping cases with incomplete data; most common in practice);
2. replacing all missing values at T1 and T2 with 0s (a default option);
3. replacing all missing values at T1 and T2 with 1s (a default option);
4. **replacing missing values at T1 with data from T2, and missing values at T2 with T1 (another default option, only for STERGM)**
5. replacing all missing values at T1 and T2 with a 50% probability of 0 or 1 (much like an uninformative/uniform prior in Bayesian statistics);

6. replacing all missing values with a probability of 0 or 1 based on the *true* density of a given particular network (i.e., prior to induced missingness); and
7. replacing all missing values with a probability of 0 or 1 that was based on the *sample* density of the network (i.e., what researchers would have available to them in reality).

Again, in total, there were 2 network sizes x 2 density levels x 6 missingness levels x 7 missingness approaches = 168 conditions applied to 1,000 simulated datasets. For each, an intercept-only STERGM was conducted using *R*'s `statnet` suite of packages. Additionally, a STERGM was also completed for each of the original networks with no missingness (1,000 networks x 2 sizes x 2 densities = 4,000 of these). As described in Study 1 (Chapter 3), the model intercepts (for tie formation and dissolution) are interpreted in logits. Like Study 1, relative bias was computed using estimates from each network's corresponding complete dataset as the true values. We summarized these data descriptively and additionally evaluated the extent of which the study design factors contributed to differences in relative bias, especially in terms of missingness handling approaches, using mixed-model analyses of variance (ANOVAs) on the tie formation and tie persistence coefficient and standard error relative bias data.

## Results

### No Missingness

Table 4.1 reports the average STERGM coefficient and standard error estimates across the complete datasets (no missingness) for tie formation and dissolution, the latter of which we are calling "persistence" for ease of understanding.

**Table 4.1**

*Mean Intercept Coefficient and Standard Error Estimates for Networks with Complete Data*

| Condition | STERGM | | | |
|---|---|---|---|---|
| | Tie Formation | | Tie Persistence | |
| | *Coeff* | *SE* | *Coeff* | *SE* |
| Smaller Network (20 Nodes) | | | | |
| 20% T1 Density (T2 30%) | -1.21 | 0.18 | -1.01 | 0.30 |
| 30% T1 Density (T2 40%) | -0.63 | 0.21 | -0.49 | 0.23 |
| Larger Network (50 Nodes) | | | | |
| 20% T1 Density (T2 30%) | -1.20 | 0.07 | -1.00 | 0.12 |
| 30% T1 Density (T2 40%) | -0.63 | 0.08 | -0.49 | 0.09 |

*Note*. $N$ = 1,000 adjacency matrices per cell analyzed except for $N$ = 20 with 20% T1 density ($N$ = 999) due to nonconvergence; all values in logits.

**Tie formation**. The log-odds of the probability of a new tie being added (*formed*) at T2 for T1 dyads that were not connected was -1.21 and -1.20 logits for smaller and larger networks, respectively. This translates to a 23% predicted probability of a new tie forming (slightly underestimating the average of the two time point densities of 25%, which could simply be sampling error). For the more dense network condition (T1 30% and T2 40%), the average tie formation coefficient was -0.63, which translates to a 35% predicted probability of a new tie forming in the network, which is the average of the two time points' densities.

**Tie persistence**. The persistence of a dyad staying linked from T1 to T2 for the lower density condition was estimated on average at -1.01 and -1.00 logits for smaller and larger networks, respectively, which translates to a predicted probability of 27%; for the higher density condition, the estimates were both -0.49 logits, which translates to a predicted probability of 38%. These are slight overestimations of the expected probability equal to the average of the densities across time points; however, this may simply reflect sampling error.

**Data with Missingness**

**Overall results.** Table 4.2 summarizes coefficient and standard error relative bias for tie formation and persistence in networks with missingness, by missingness handling approach. Across conditions, all non-stochastic missingness methods produced unacceptably high formation and persistence coefficient estimation bias, including node deletion (i.e., more than 5%). Tie formation in particular was not well estimated by any method except for the 50% imputation method (e.g., like using an uninformative prior). The worst method by far was use of auxiliary information, a default method available in `statnet` specific only to STERGMs.

**Table 4.2**

*Mean Relative Bias by Missingness Handling Approaches across Conditions*

| Missingness Handling Approach | Tie Formation | | Tie Persistence | |
|---|---|---|---|---|
| | *Coeff* | *SE* | *Coeff* | *SE* |
| Node Deletion | 11% | 26% | 15% | 33% |
| Inserting all 0s | 26% | -4% | 40% | 10% |
| Inserting all 1s | -36% | 15% | -27% | -10% |
| Inserting Auxiliary Information | 46% | 0% | -98% | -1% |
| 50% Density-Based Random Imp. | -5% | 2% | 0% | -4% |
| True Density-Based Random Imp. | 4% | -1% | 9% | 2% |
| Sample Density-Based Random Imp. | 4% | -1% | 9% | 2% |

*Note*. $N$ = 23,999 estimates per missingness handling approach (one model for the lower density, smaller network size condition did not converge); values are collapsed across network size, density, and missingness levels.

Standard errors for tie formation were found to be unacceptably biased for node deletion and the "inserting all 1s" condition, both of which inflated the standard errors and would result in increased Type II error and decreased power. For tie persistence, the standard errors were inflated for node deletion and the "inserting all 0s" condition; the "inserting all 1s" condition resulted in underestimates of the standard error thereby leading to increased Type I error. On

balance, the methods producing the least bias in both coefficient and standard error estimation include the three stochastic imputation methods.

**Design effects on coefficient relative bias**. Mixed model ANOVA results for relative bias on the **tie formation coefficient** showed that all main effects were significant. Conditions with greater missingness exhibited more bias ($F(5, 19470) = 1494.06$, $p < .001$, $\eta^2 = .28$), and approaches that used non-stochastic single imputation methods for handling missingness had greater bias ($F(6, 23364) = 63155.55$, $p < .001$, $\eta^2 = .94$). Networks with lower density were more likely to be overestimated and vice versa ($F(1, 3894) = 600.57$, $p < .001$, $\eta^2 = .13$), and similarly, smaller sized networks were also more likely to be overestimated and vice versa ($F(1, 3894) = 763.78$, $p < .001$, $\eta^2 = .16$).

All 2- and 3-way interactions were statistically significant ($F$-test $p$s < .001). However, given our relatively large sample size of simulations and our motivation to understand which missingness handling approaches are best at reducing bias, we limited our attention to results for significant 2-way interactions with approach type that explained at least 1% of the variation in the relative bias data. (Additionally, we already have information about the results for other types of interaction from Study 1.) Those significant interactions included approach by missingness level ($\eta^2 = .74$, $p < .001$), approach by density ($\eta^2 = .33$, $p < .001$), and approach by network size ($\eta^2 = .12$, $p < .001$). To understand the nature of these interactions, we plotted tie formation coefficient relative bias for each missingness handling approach by missingness level (Figure 4.1), density (Figure 4.2), and network size (Figure 4.3).

As can be seen in Figure 4.1, relative bias became more pronounced with increasing missingness, but really for the three non-stochastic single imputation approaches (inserting all 0s, all 1s, or auxiliary information).

**Figure 4.1**

*Relative Bias Estimates by Missingness Approach and Missingness Level for Formation Coeff*

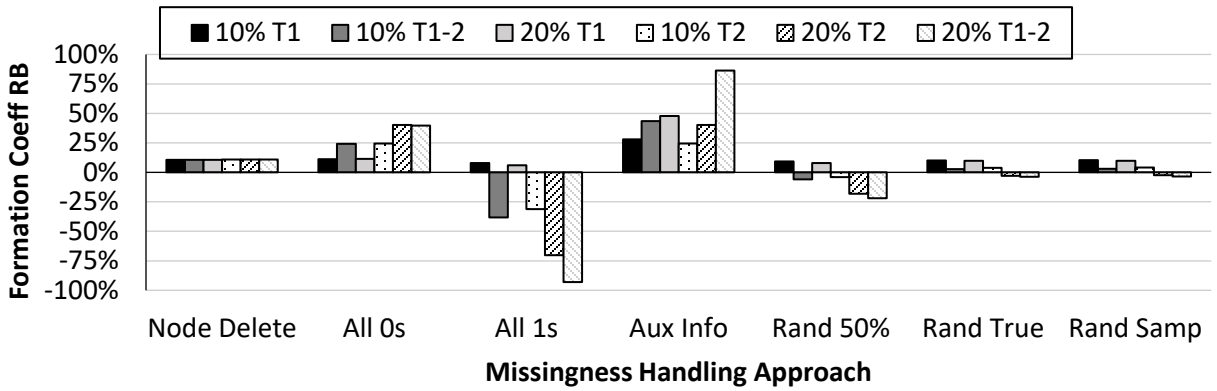



**Figure 4.2**

*Relative Bias Estimates by Missingness Approach and Density Level for Formation Coeff*

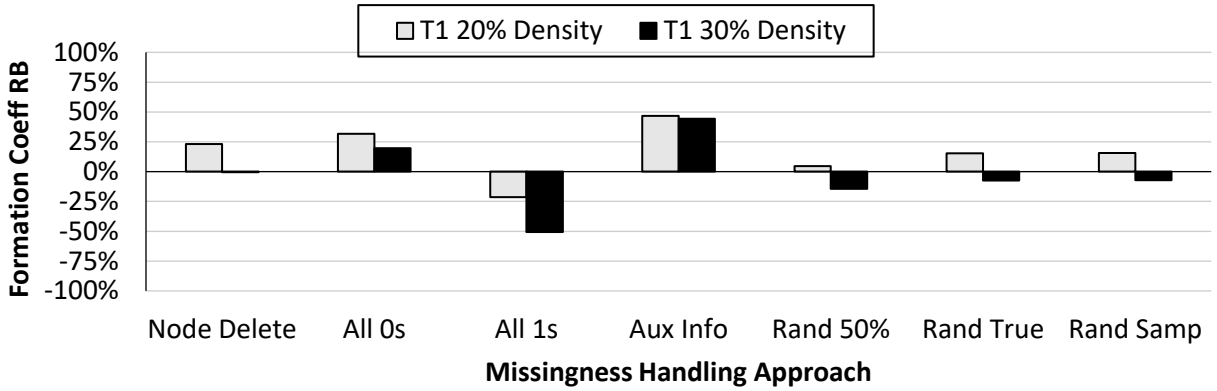
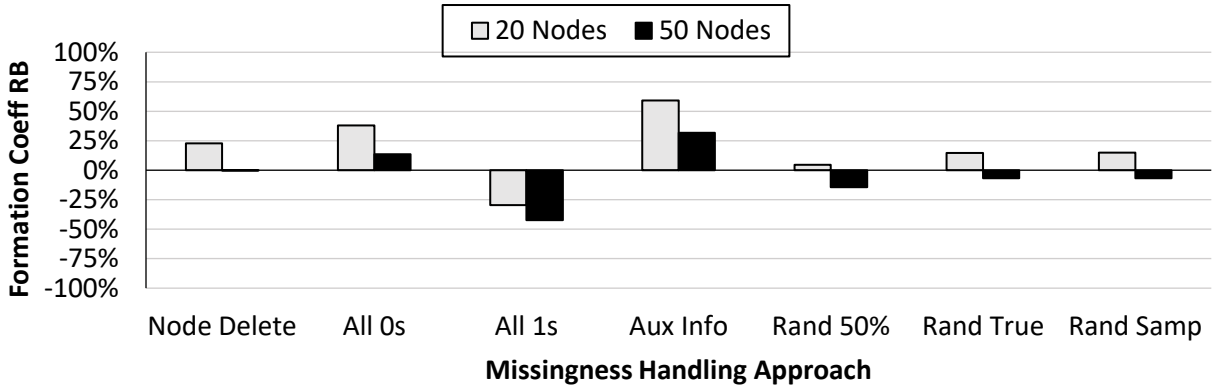



**Figure 4.3**

*Relative Bias Estimates by Missingness Approach and Network Size for Formation Coeff*



In Figure 4.2, we observe that the direction of coefficient bias depended on density level (but for only those approaches exhibiting any bias): inserting all 1s or imputing 50% 1s into a less dense distribution will overestimate density and vice versa. Last, in Figure 4.3, we see that, in general, larger networks exhibited less bias but this was only pronounced for approaches that exhibited bias in the first place.

ANOVA results for relative bias on **tie persistence coefficient** estimate showed nearly identical findings to those for tie formation. All main effects and interactions were significant. The main effect for missingness level showed that greater missingness resulted in more bias, ($F(5, 19470) = 1251.39$, $p < .001$, $\eta^2 = .24$). The missingness approach main effect showed that approaches markedly varied in their estimation accuracy, with use of auxiliary information being the worst performer ($F(5, 23364) = 4634.08$, $p < .001$, $\eta^2 = .93$). The density main effect revealed that more dense networks tended to have more negative bias ($F(1, 3894) = 663.26$, $p < .001$, $\eta^2 = .15$), and the network size main effect showed that larger sized networks also tended to have more negative bias ($F(1, 3894) = 459.89$, $p < .001$, $\eta^2 = .11$). All 2- and 3-way interactions were statistically significant ($F$-test $p$s $< .001$). However, again, given our relatively large sample size of simulations and our motivation to understand which missingness handling approaches are best at reducing bias, we limited our attention to results for significant 2-way interactions with approach type that explained at least 1% of the variation in the relative bias data. Just like interactions detected for tie formation, interactions detected for tie persistence included approach by missingness level ($\eta^2 = .67$, $p < .001$), approach by density ($\eta^2 = .18$, $p < .001$), and approach by network size ($\eta^2 = .12$, $p < .001$). To understand the nature of these interactions, we plotted tie persistence coefficient relative bias for each missingness handling approach by missingness level (Figure 4.4), density (Figure 4.5), and network size (Figure 4.6).

**Figure 4.4**

*Relative Bias Estimates by Missingness Approach and Missingness Level for Persistence Coeff*

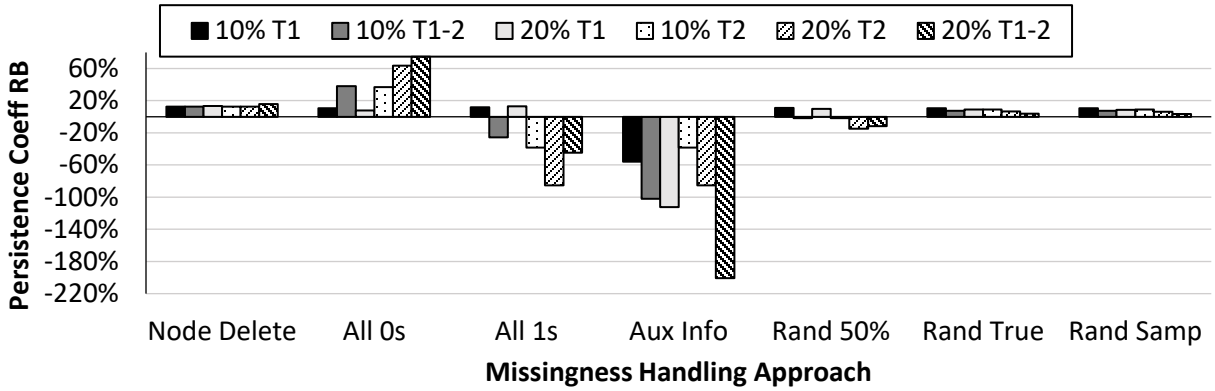



**Figure 4.5**

*Relative Bias Estimates by Missingness Approach and T1 Density Level for Persistence Coeff*

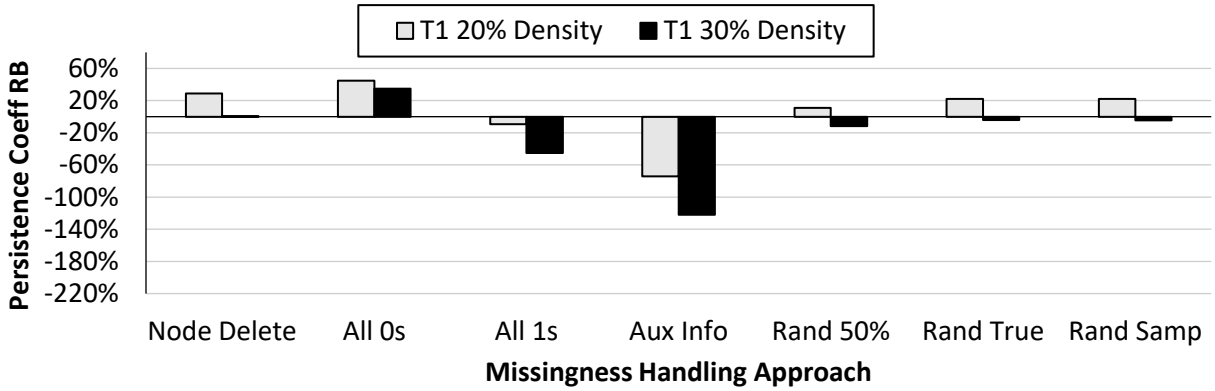
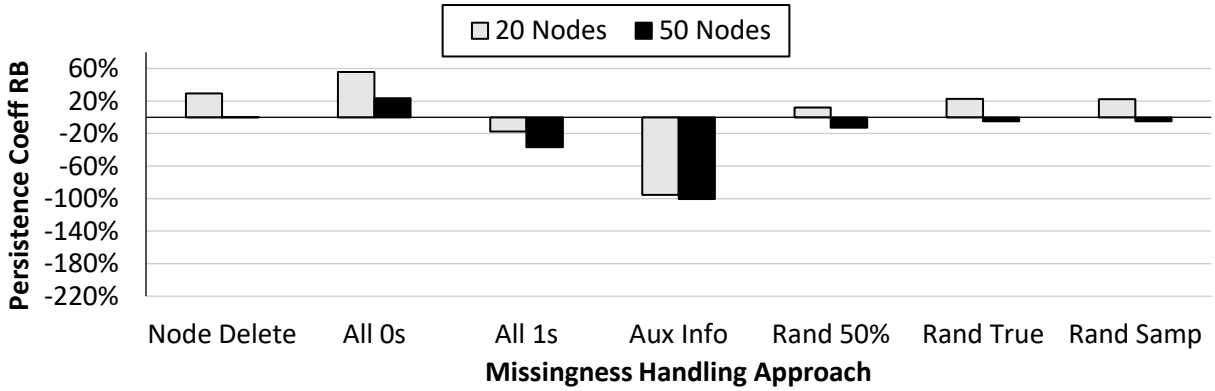



**Figure 4.6**

*Relative Bias Estimates by Missingness Approach and Network Size for Persistence Coeff*



As can be seen in these figures, relative bias was more pronounced for higher levels of missingness, density, and network size for the three non-stochastic imputation approaches, especially for the auxiliary information approach.

**Design effects on standard error relative bias**. Mixed model ANOVA results for relative bias on the **tie formation standard error** showed that all main effects were significant. Conditions with greater missingness exhibited more bias ($F(5, 19470) = 68639.96$, $p < .001$, $\eta^2 = .95$), and approaches that used non-stochastic single imputation methods for handling missingness had greater bias ($F(6, 23364) = 413240.59$, $p < .001$, $\eta^2 = .99$). Networks with lower density were more likely to be overestimated and vice versa ($F(1, 3894) = 1349.64$, $p < .001$, $\eta^2 = .26$), and similarly, smaller sized networks were also more likely to be overestimated and vice versa ($F(1, 3894) = 717.26$, $p < .001$, $\eta^2 = .16$).

Given our relatively large sample size of simulations and our motivation to understand which missingness handling approaches are best at reducing bias, we limited our attention to results for significant 2-way interactions with approach type that explained at least 1% of the variation in the relative bias data. Those significant interactions included approach by missingness level ($\eta^2 = .94$, $p < .001$) and approach by density ($\eta^2 = .72$, $p < .001$). As can be seen in Figure 4.7, relative bias was generally more pronounced with increasing missingness, but mostly for the node deletion method. In Figure 4.8, we observe that the direction of coefficient bias depended on density level (but only for node deletion or inserting all 0s or all 1s methods.

For bias in the **tie persistence standard error**, again the ANOVA results showed all main effects and interactions were significant ($p$s $< .001$). Focusing on the two-way interactions with missingness handling approach, with approach by missingness level $\eta^2 = .87$, approach by density $\eta^2 = .52$, and approach by network size $\eta^2 = .12$, we found that design effects on relative

**Figure 4.7**

*Relative Bias by Missingness Approach and Missingness Level for Formation Standard Error*

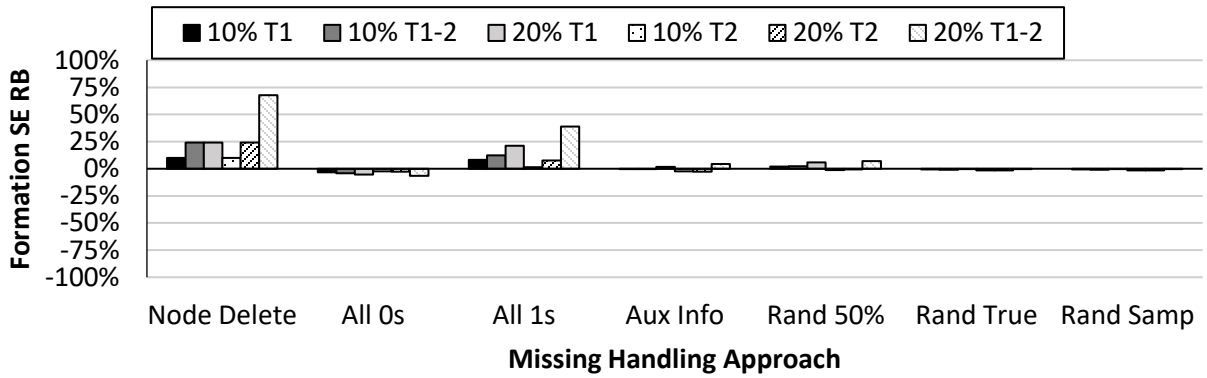



**Figure 4.8**

*Relative Bias by Missingness Approach and T1 Density for Formation Standard Error*

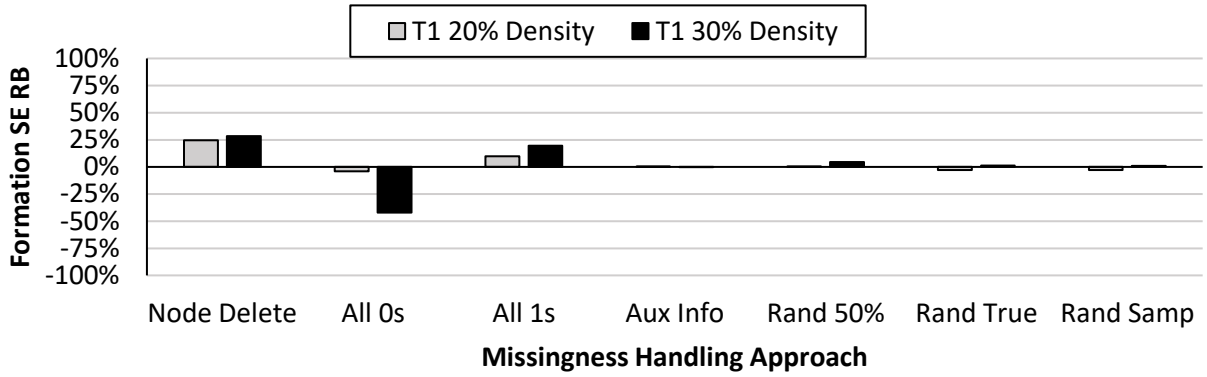
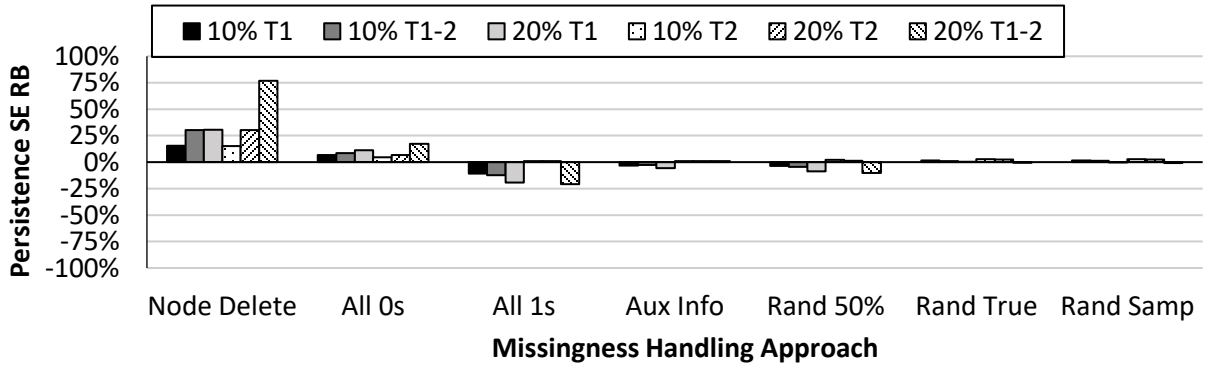



**Figure 4.9**

*Relative Bias by Missingness Approach and Missingness Level for Persistence Standard Error*



bias were mostly isolated to the node deletion method. Specifically, the node deletion method resulted in greater bias for conditions with higher missingness (Figure 4.9), and the lower network size and density conditions (see Figures 4.10 and 4.11, respectively).

**Figure 4.10**

*Relative Bias by Missingness Approach and T1 Density Level for Persistence Standard Error*

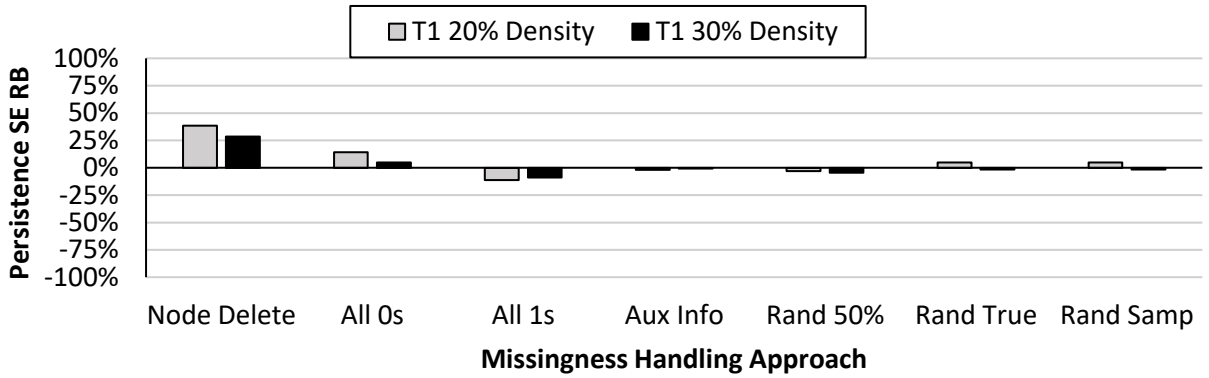
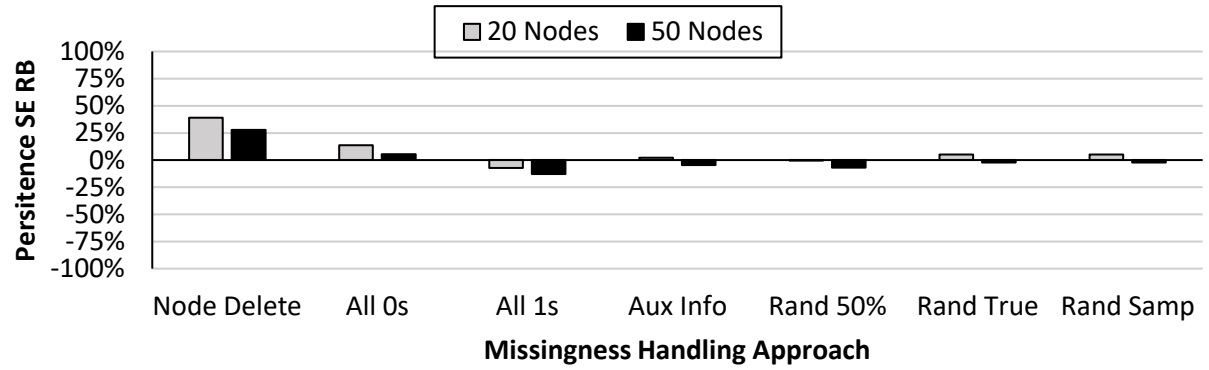



**Figure 4.11**

*Relative Bias by Missingness Approach and Network Size for Persistence Standard Error*




**Next Steps**

Study 2 results extended Study 1 results to multiple time point models. Results were consistent with Study 1 in that the stochastic true and sample density-based imputation approaches were best maximizing parameter estimation accuracy (i.e., least biased coefficients) and precision (i.e., least biased standard errors). Use of the "auxiliary information" approach, whereby one borrows information from previous or subsequent time points to "fill in" missing

data yielded the most bias in the coefficients, and node deletion resulted in the most bias standard error estimates. Of course, like Study 1, the Study 2 findings are specific to binary valued networks in particular. The next chapter extends this work to the case of handling missing data in weighted social network analyses.

# CHAPTER 5.

## Study 3: Missing Data Handling for Single Time Point Networks with Weighted Ties

In addition to extending the basic ERGM to multiple time points, statisticians have also extended ERGMs to incorporate **weighted** or "**valued**" ties, such as ratings and counts that quantify the magnitude of relationships[2] (Krivitsky, 2014). Whereas binary valued network model results are interpreted as the probability of a tie forming in the network conditional on network properties (Scott, 2015), weighted or "valued" network model results are interpreted as the log-odds of the *expected value* (mean) of a tie in the network, given the network properties. In other words, the intercept coefficient provides an estimate of the conditional average strength of relationships in the network. For example, in a teacher network, research questions could be focused on how often teachers collaborate on teaching practices, or how many times they collaborated over the past year, instead of simply whether or not they ever collaborate.

This shift in focus from estimating the likelihood of a tie to estimating the strength of ties invokes the need for a **reference distribution**: the assumed shape of the tie values. For frequency data, it may be appropriate to assume a normal distribution; however, for count data, it may be more appropriate to assume a Poisson. The sample space of binary ERGM defines the sample space as a subset of a power set, whereas the sample space in a valued ERGM is defined as $\mathcal{Y} \subseteq \mathbb{N}_0^{\mathbb{Y}}$, a set of mappings that assigns each dyad($i$, $j$) $\in \mathbb{Y}$ a count. The network model is interpreted as the probability of observing a tie *count* conditional on $x$ network properties, as follows (Krivitsky, 2014):

$$\Pr_{\theta;h,\eta,g}(\mathbf{Y} = \mathbf{y}|\mathbf{x}) = \frac{h(\mathbf{y}) * \exp\left(\boldsymbol{\eta}(\boldsymbol{\theta}) * \mathbf{g}(\mathbf{y};\mathbf{x})\right)}{\kappa_{h,\eta,g}(\boldsymbol{\theta};\mathbf{x})}, \quad (5.1)$$

[2] Although binary valued networks can be analyzed across multiple time points, models for multiple time point weighted networks are still under development (personal communication, 2019).

with the normalizing constant as:

$$\kappa_{h,\eta,g}(\theta) = \sum_{y \in Y} h(y) \exp\big(\eta(\theta) \cdot g(y)\big). \quad (5.2)$$

For model 5.1, *h(y)* represents the reference distribution, *η(θ)* is a vector of *k* model parameters to be estimated, *g(y;x)* is a vector of change statistics (associated with *θ*), each representing a change in the network statistics *when a tie increases by 1 unit* in the network (conditional on modeled network properties, x; units depend on *h*), and:

$$\Theta \subseteq \Theta_N = \{\theta' \in \mathbb{R}^q : k_{h,\eta,g}(\theta') < \infty\}. \quad (5.3)$$

With $\Theta_N$ being the natural parameter space if the ERGM is linear. Equation 5.3 is trivial for binary ERGMs, but for counts it is much more complex (Krivitsky, 2014).

The reference measure is the distribution relative to which the exponential form is specified. For binary ERGMs, *h* is usually not explicitly specified due to the terms with fixed parameters implicitly being absorbed into *h*. For valued networks, and in particular count data, the specification of *h* is required (Krivitsky, 2014). The reference distributions available in `statnet`'s regular `ergm` package include Discrete Uniform, Continuous Uniform, and Standard Normal; Complete Order for ranks is available in the `ergm.rank` package, and Poisson, Geometric, and Binomial reference distributions are available in the `ergm.count` package (Hunter, et al., 2008b; Krivistky & Butts, 2017; Krivitsky, 2014).

**Missingness in Weighted Social Networks**

Our prior results in Chapters 3-4 showed the optimal approach for handling missing data is to base the probabilities of a tie on the sample's network density level (although node deletion is still reasonable in terms of bias, it has far less precision, particularly in small sized networks; see Chapters 3-4). In the present study, we examined three potential approaches for missingness in networks with integer valued ties that were generated as normally distributed; specifically,

these included node deletion, stochastic single imputation using the discrete uniform distribution, and stochastic single imputation using the multivariate distribution (with values rounded to create discrete integer values, as would be assumed in real data).

### Research Question

As with all the `statnet` suite of packages, there is currently no built-in missing data stochastic imputation methods for analyzing weighted networks in *R*. Thus, the overarching research question for this paper is: *Which missing data handling approach is best for analyzing normally distributed social network data using weighted ERGM?*

## Method

### Data Generation

Similar to studies 1 and 2, simulated directed weighted directed adjacency matrices were generated in *R* using our own code (see Appendix C) to reflect network characteristics loosely based on data from two science teacher professional development network projects: Ambitious Science Teaching (AST; Thompson et al., 2019) and Partnership for Ambitious Science Teacher Leaders (PASTL; Feldman et al., 2018). Recall that for these projects, teachers nominated peer teachers, teacher leaders, school principals, and university researchers for pedagogical advice seeking and collaboration. Teachers also reported the frequency with which they interacted with their peers about their science teaching practices. Modeling this type of frequency data requires use of a weighted ERGM, rather than a binary valued ERGM.

**Mean tie strength**. Tie values were generated as random draws from a multivariate normal distribution with a mean of 1.50, variance of 1.67, and covariance of 0.325. These parameters were inspired by observed teacher network data to reflect tie strengths ranging from approximately 0 to 3 (as a count or rating scale), with a small correlation among nodes (e.g., the

idea that a teacher who reports a greater magnitude of interaction with one teacher is also slightly more likely to have a greater magnitude with other teachers). Generated data were rounded to whole numbers to create discrete values.

**Network size**. For this study, we used four levels of network sizes to incorporate a broader range of realistic situations, including 10, 20, 50, or 100 nodes (recall that Studies 1 and 2 used sizes of 20 and 50 nodes only).

**Missingness levels.** We also incorporated more levels of missingness at the node level for a slightly broader range of missingness: 10%, 20%, and 30% (Studies 1 and 2 both used 10% and 20% only).

**Missingness handling approaches.** Three missingness handling approaches were used, including: node deletion, stochastic single imputation using the discrete uniform distribution ranging from 0 to 3 (akin to an "uninformative" prior within the bounds of the range assumed in the data), and stochastic single imputation using the multivariate normal distribution.

Overall, there were 1,000 directed, weighted matrices generated for each of the four network sizes, and for each of these, 3 missingness levels and 3 missing handling approaches were applied. An intercept-only weighted ERGM was then conducted using `statnet` (using the regular `ergm` package with the standard normal reference distribution) on each of the 4000 x 3 x 3 = 36,000 datasets. For these models, the intercept results (mean tie strength) are in the scale they were generated in (for example, values in a rating scale measure in points). Similar to Studies 1 and 2, relative bias was computed using estimates from each network's corresponding complete dataset as the true values. We again summarized data descriptively and evaluated the extent to which study design factors contributed to relative bias using mixed-model analyses of variance (ANOVAs).

## Results

### No Missingness

Table 5.1 reports the average weighted ERGM intercept coefficient and standard error estimates across the complete datasets (no missingness). As can be seen, the strength of a tie in the network averaged 1.57, which is close to the multivariate normal mean of 1.50 used to generate the data initially (the slight overestimation may be due to the rounding used after initial data generation). It is also evident that coefficient estimation was largely invariant to network size, and as expected, greater network size provided better precision (smaller standard errors).

**Table 5.1**

*Mean Intercept Coefficient and Standard Error Estimates for Networks with Complete Data*

| Network Size | *Coeff* | *SE* |
|---|---|---|
| Smallest Network (10 Nodes) | 1.57 | 0.20 |
| Smaller Network (20 Nodes) | 1.57 | 0.10 |
| Larger Network (50 Nodes) | 1.57 | 0.04 |
| Largest Network (100 Nodes) | 1.58 | 0.02 |

*Note. N* = 1,000 adjacency matrices for each of the four conditions.

### Data with Missingness

**Overall results.** Table 5.2 summarizes intercept coefficient and standard error relative bias for networks with missingness induced, by missingness handling approaches.

**Table 5.2**

*Mean Relative Bias by Missingness Handling Approaches across Conditions*

| Missingness Handling Approach | *Coeff* | *(SE)* |
|---|---|---|
| Node Deletion | <1% | 26% |
| Uniform Distribution Random Imp. | <1% | <1% |
| Multivariate Normal Distribution Random Imp. | <1% | <1% |

*Note: N* = 12,000 estimates collapsed across network sizes and missingness levels.

**Design effects on coefficient relative bias**. The ANOVA results for the coefficient estimates showed a small main effect for approach ($F(2, 7992) = 134.90$, $p < .001$, $\eta^2 = .03$); inspection of the marginal means showed that the use of the random uniform distribution imputation approach resulted in slightly more coefficient bias (0.90%, which is <1%) than the other two approaches (bias for node deletion averaged 0.20% and bias for random normal imputation averaged 0.10%, both of which are <1%). Although there was one significant interaction between approach and missingness level, the interaction explained <1% of the variation in coefficient bias and simply reflected the fact that the node deletion approach had more bias when there was more missingness in the data. Last, we found no main effect for network size on bias ($F$-test $p > .05$).

**Design effects on coefficient standard error bias**. All main effects and interactions were significant in the ANOVA results for relative bias on the coefficient standard error. Greater missingness led to more inflated standard errors ($F(2, 7992) = 3558.05$, $p < .001$, $\eta^2 = .47$), and node deletion had significantly more standard error overestimation than the other missingness handling approaches ($F(2, 7992) = 33313.84$, $p < .001$, $\eta^2 = .89$). Larger network sizes, on the other hand, produced less bias overall ($F(3, 3996) = 65.21$, $p < .001$, $\eta^2 = .05$).

Focusing on 2-way interactions with missingness handling approaches with effect sizes of $\eta^2 \geq .01$, we found that missingness level and network size only mattered for the node deletion approach. Specifically, for the node deletion condition, greater missingness related to greater standard error inflation ($\eta^2 = .51$, $p < .001$; see Figure 5.1), and larger network sizes related to slightly less standard error inflation ($\eta^2 = .08$, $p < .001$; see Figure 5.2). Both single imputation approaches (discrete uniform and multivariate normal), on the other hand, were largely

unaffected by design conditions and did not differ from each other. Indeed, only node deletion showed unacceptable levels of standard error bias (greater than 10%).

**Figure 5.1**

*Relative Bias by Missingness Handling Approach and Missingness Level for Tie Standard Error*

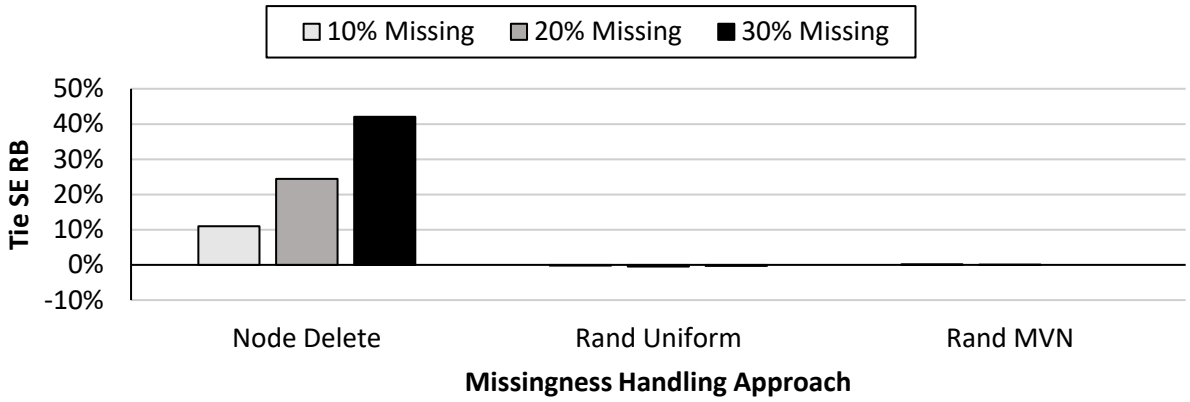
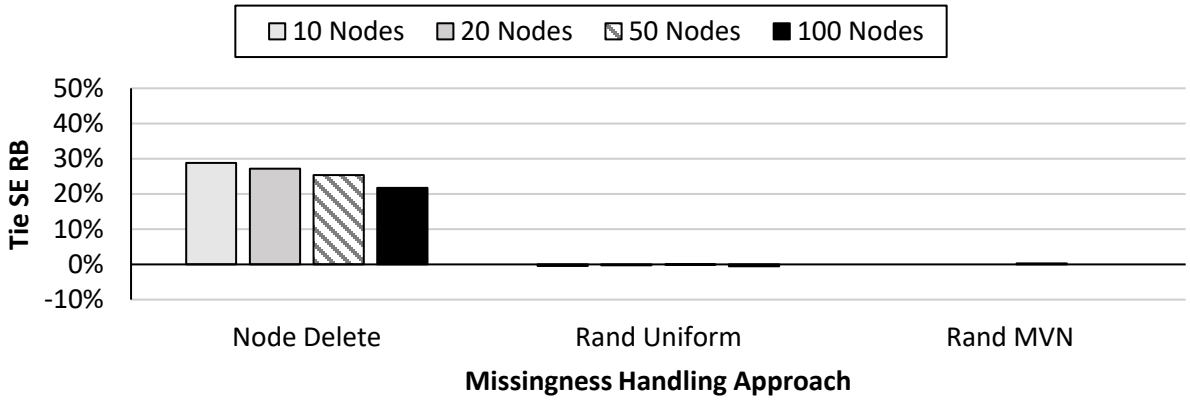



**Figure 5.2**

*Relative Bias by Missingness Handling Approach and Network Size for Tie Standard Error*




**Next Steps**

The present study compared missing data handling approaches for weighted valued networks. Consistent with our findings for binary networks, these results showed that, while node deletion (the default option for many researchers) can result in fairly unbiased coefficient

estimates, the method lacks precision (i.e., the standard errors are inflated). Instead, using the multivariate normal random imputation method, or the uniform random imputation method, allowed for relatively unbiased coefficient estimates and better precision. Nevertheless, stochastic imputation approaches for handling missingness are yet not available in any of the packages in *R*'s `statnet` suite. The following chapter describes a novel package developed for *R* that can impute data prior to analyses, called `netImp`.

# CHAPTER 6.

## R package for Imputing Missing Network Data (`netImp`)

The diverse nature of data that can be represented by random graphs spans many disciplines, such as life sciences, health sciences, and social sciences. The `netImp` package for R is a package designed to impute missing data for statistical network analysis. In particular, the package allows users to fill in missing data information for undirected and directed network matrices, and binary and weighted/valued networks, which can subsequently be analyzed through any of the SNA inferential methods, but the focus of this paper will be on exponential random graph based models (ERGMs). This article describes the technical background of the `netImp` package.

### Installing `netImp`

The package `netImp` is currently under review by CRAN for publication. All methods included in this package are from research discussed in previous papers. Please email the author for distribution until the package has been approved.

**Figure 6.1**

*Installing `netImp`*

```
Install.packages("netImp")
```

### Data Format

When read into *R*, network data usually takes on one of two forms: either as an adjacency matrix or edgelist. This package requires data to be in adjacency matrix format, with missing data formatted as "NA". Missing data must also be at the node level (i.e., missing data across the rows of the matrix).

**Figure 6.2**

*Generating Missing data in* `netImp`

```
net.m = missing(net, .2)
net.m
      [,1] [,2] [,3] [,4] [,5]
[,1]     0    0    1    0    0
[,2]     1    0    1    1    0
[,3]     1    0    0    1    1
[,4]     0    0    0    0    0
[,5]    NA   NA   NA   NA   NA
```

**Node Deletion for Binary or Weighted Networks**

The node deletion option removes nodes with missing data. This option assumes the node is no longer part of the network. In doing so, the estimates tend to have a rather small bias, but the standard errors, or the precision, of the model tend to be vastly inflate. This option seems to work well if there is little concern about the standard errors. This option works both binary and valued ERGM and STERGM, and for both directed and undirected networks. It is also worth noting deletion, all zeros and all ones are the options given in the ERGM package.

**Figure 6.3**

*Code and Output for Node Deletion Approach*

```
net.d = del(net.m)
net.d
      [,1] [,2] [,3] [,4]
[,1]     0    0    1    0
[,2]     1    0    1    1
[,3]     1    0    0    1
[,4]     0    0    0    0
```

**Inserting all Zeros for Binary or Weighted Networks (Not Recommended)**

This option imputes all zeros in the network for missing node data. This assumes the node would have no ties to any other node in the network. This tends to increase bias downward

in the network due to the assumption of no ties. Using this option would be best when the network is very sparse/have a very low density. This option works with binary ERGM and STERGM, and for both directed and undirected networks.

**Figure 6.4**

*Code and Output for Inserting 0s*

```
net.z = zeros(net.m, directed = TRUE)
net.z
     [,1] [,2] [,3] [,4] [,5]
[,1]    0    0    1    0    0
[,2]    1    0    1    1    0
[,3]    1    0    0    1    1
[,4]    0    0    0    0    0
[,5]    0    0    0    0    0
```

**Inserting all Ones for Binary Networks (Not Recommended)**

This option imputes all ones for missing node data. This option assumes the nodes would have ties to every other node in the network. This tends to bias the network upwards due to having ties to all other nodes. This option is best used when the density is very high or as close to a density as 1. This option works with binary ERGM and STERGM networks, and for both directed and undirected networks.

**Figure 6.5**

*Code and Output for Inserting all 1s*

```
net.o = ones(net.m, directed = TRUE)
net.o
     [,1] [,2] [,3] [,4] [,5]
[,1]    0    0    1    0    0
[,2]    1    0    1    1    0
[,3]    1    0    0    1    1
[,4]    0    0    0    0    0
[,5]    1    1    1    1    1
```

**Imputing 50% Random Probability of a Tie for Binary Networks**

This option allows for a 50% probability to either have a tie (1) or not have a tie (0) for each node. This option tends to have a higher bias than deletion, but lower than all ones or all zeros. This method works well if the density is close to 50%. The option works with binary ERGM and STERGM, and both directed and undirected networks. This function also has an option to limit the total number of ties in the imputed values. The function parameter nomlim = 3 is to limit the total number of ties that could exist in the network. This is useful if the method of recording ties in the network is limited to any number below the network size. In this case the network is limited only three nominations per node, then the imputation method will only generate a maximum of three ties in the network for the missing node data based on the equal probability of a tie.

**Figure 6.6**

*Code and Output for 50% Random Probability*

```
net.crd = comrd(net.m, directed = TRUE, nomlim = 3)
net.crd
     [,1] [,2] [,3] [,4] [,5]
[,1]    0    0    1    0    0
[,2]    1    0    1    1    0
[,3]    1    0    0    1    1
[,4]    0    0    0    0    0
[,5]    0    1    1    0    0
```

**Random Density (unweighted)**

The random density option uses the density of the network if the nodes were removed (listwise deletion) to impute the probability of a tie or no tie to occur for the missing values in each node. The function parameter nomlim = 3 is to limit the total number of ties that could exist in the network. This is useful if the method of recording ties in the network is limited to any

number below the network size. In this case the network is limited only three nominations per node, then the imputation method will only generate a maximum of three ties in the network for the missing node data based on the density probability. The nomination limitation will break down however, if the number of nominations is too small and the overall density probability is too high (rare and unrealistic combinations). This method had a rather smaller bias and low bias in standard errors. This method is the one most recommended by research previously presented. This option works with binary ERGM and STERGM, and both directed and undirected networks.

**Figure 6.7**

*Code and Output for Random Density Observed as the Probability for a Tie*

```
net.r = rd(net.m, directed = TRUE, nomlim = 3)
net.r
     [,1] [,2] [,3] [,4] [,5]
[,1]    0    0    1    0    0
[,2]    1    0    1    1    0
[,3]    1    0    0    1    1
[,4]    0    0    0    0    0
[,5]    0    1    0    1    0
```

**Imputation from Previous or Subsequent Time Point (unweighted/weighted)**

This method uses data from two time points of networks and pulls data that is not missing and imputes them into the other network with missing data. This method assumes across any two time points, nodes with missing data does not change between measurements. This has the potential to bias the formation and persistence of ties either upwards or downwards depending the network. This option works with both directed and undirected multiple time point networks. As the code for auxiliary imputation creates a list of matrices, the unlist function separates the

list by identifying the object to be unlisted, then matrix number in the list, then the size of the network.

**Figure 6.8**

*Code and Output for Auxiliary Information Imputation*

```
net1
      [,1] [,2] [,3] [,4] [,5]
[,1]     0    0    1    0    0
[,2]     1    0    1    1    0
[,3]     1    0    0    1    1
[,4]     0    0    0    0    0
[,5]    NA   NA   NA   NA   NA


net2
      [,1] [,2] [,3] [,4] [,5]
[,1]     0    0    0    1    0
[,2]    NA   NA   NA   NA   NA
[,3]     1    1    0    0    1
[,4]     0    0    1    0    1
[,5]     1    0    0    1    0


net.i = imp(net1, net2, directed = TRUE)
net.i1 = imp.unlist(net.i, listnum = 1, n = 5)
net.i1
      [,1] [,2] [,3] [,4] [,5]
[,1]     0    0    1    0    0
[,2]     1    0    1    1    0
[,3]     1    0    0    1    1
[,4]     0    0    0    0    0
[,5]     1    0    0    1    0


net.i2 = imp.unlist(net.i, listnum = 2, n = 5)
net.i2
      [,1] [,2] [,3] [,4] [,5]
[,1]     0    0    0    1    0
[,2]     1    0    1    1    0
[,3]     1    1    0    0    1
[,4]     0    0    1    0    1
[,5]     1    0    0    1    0
```

**Uniform Distribution (Weighted)**

The uniform distribution allows for an equal probability of each potential value of a tie in the network. This means, in a valued network, each value seen in the network has an equal probability of occurring. This can bias the estimates since the values in a network are seldom uniformly distributed. This option works with both directed and undirected valued networks.

**Figure 6.9**

*Code and Output for Uniform Distribution Imputation*

```
netw.m
      [,1] [,2] [,3] [,4] [,5]
[,1]    0    3    2    0    1
[,2]    3    0    3    2    0
[,3]   NA   NA   NA   NA   NA
[,4]    1    2    3    0    3
[,5]    3    1    0    0    0


netw.uni = unirdw(netw.m, values = c(0,1,2,3), directed = TRUE)
netw.uni
      [,1] [,2] [,3] [,4] [,5]
[,1]    0    3    2    0    1
[,2]    3    0    3    2    0
[,3]    0    3    0    1    1
[,4]    1    2    3    0    3
[,5]    3    1    0    0    0
```

**Multivariate Normal Distribution (weighted)**

The multivariate normal distribution imputes missing data from a multivariate normal distribution spanning the range of values seen in the network. This produces the least amount of bias assuming the data in the network has a multivariate distribution. If a multivariate normal distribution is not assumed, the distribution assumed needs to be changed to an appropriate distribution. This option works with both directed and undirected valued networks.

**Figure 6.10**

*Code and Output for Multivariate Normal Distribution Imputation*

```
n = 5
cov = .325
var = 1.67
sigma = matrix(cov, n, n)
diag(sigma) = var
sigma
      [,1]  [,2]  [,3]  [,4]  [,5]
[,1] 1.670 0.325 0.325 0.325 0.325
[,2] 0.325 1.670 0.325 0.325 0.325
[,3] 0.325 0.325 1.670 0.325 0.325
[,4] 0.325 0.325 0.325 1.670 0.325
[,5] 0.325 0.325 0.325 0.325 1.670


netw.mvr = mlrdw(netw.m, lower = 0, upper = 3, mean = 1.5, sigma = sigma, directed =
TRUE
netw.mvr
     [,1] [,2] [,3] [,4] [,5]
[,1]    0    3    2    0    1
[,2]    3    0    3    2    0
[,3]    2    1    0    0    1
[,4]    1    2    3    0    3
[,5]    3    1    0    0    0
```

**Network Generation**

Lastly, `netImp` also includes the option to generate adjacency matrices as either directed or undirected and binary or valued. The networks are generated based on a specified network size, density and reciprocity when applicable. Users can also create various amounts of missing data to explore the different methods of social network imputation included in this package.

**Figure 6.11**

*Adjacency Matrix Generation of Undirected Binary Network*

```
net.undirected = r.ntwk.undirected(n = 5, density = .4)
net.undirected
      [,1] [,2] [,3] [,4] [,5]
[,1]    0    0    1    1    0
[,2]    0    0    0    0    1
[,3]    1    0    0    1    0
[,4]    1    0    1    0    0
[,5]    0    1    0    0    0
```

**Figure 6.12**

*Adjacency Matrix Generation of Directed Binary Network*

```
net.directed = r.ntwk.directed(n = 5, density = .4, recip = .2)
net.directed
      [,1] [,2] [,3] [,4] [,5]
[,1]    0    0    1    0    0
[,2]    1    0    1    1    1
[,3]    0    0    0    0    0
[,4]    0    0    0    0    1
[,5]    0    1    1    0    0
```

**Figure 6.13**

*Adjacency Matrix Generation of Undirected Weighted Network*

```
net.weighted.undirected = wght_mat.undirected(n = 5, mean = 1.5, cov = .325, var = 1.67)
net.weighted.undirected
      [,1] [,2] [,3] [,4] [,5]
[,1]    0    3    2    1    2
[,2]    3    0    2    2    0
[,3]    2    2    0    3    0
[,4]    1    2    3    0    2
[,5]    2    0    0    2    0
```

**Figure 6.14**

*Adjacency Matrix Generation of Directed Weighted Network*

```
net.weighted.directed = wght_mat.directed(n = 5, mean = 1.5, cov = .325, var = 1.67
net.weighted.directed
      [,1] [,2] [,3] [,4] [,5]
[,1]    0    2    0    1    2
[,2]    2    0    3    3    1
[,3]    1    0    0    0    0
[,4]    3    2    0    0    3
[,5]    2    1    1    3    0
```

**Next Steps**

This package is the first step in improving estimation procedures for social network data. The authors acknowledge this is a rather simplistic method of imputation, which can be thought of as single imputation (as compared to multiple imputation methods). When unavailable, which tends to occur in sparse networks, multiple imputation methods break down due to small levels of variance in the distribution of values. This package is designed to aid in imputing missing data to improve estimation of ERGMs and networks in general. This package also accounts for network designs with limited nominations. For example, some network studies only allow for up to 5-10 nominations, limiting the number of total ties and the overall density of the network. For more in-depth information on the techniques outlined in this paper, please refer to the previous sections using simulations to identify the least bias methods for imputing missing data for networks (ERGM, STERGM, and Weighted ERGM).

Research and updates to the `netImp` package is far from complete. For example, another of the limitations is the inability to incorporate covariates into the model. Covariates can potentially alter the probability of ties in the network. Structural zeros could exist due to various network constraints, such as distance not allowing individuals to interact or individuals who

work in difference departments or companies. The next steps for this package are to include covariates into the model in order to handle missingness at random (MAR) situations (see Chapter 2), as well as to provide insights for creating multiply imputed datasets and computations for pooling ERGM estimates.

# CHAPTER 7.

## Summary of Findings and Future Research Directions

*"Studies have shown that accurate numbers aren't any more useful than the ones you make up."*

*– Dilbert*

This dissertation systematically investigated different missingness handling approaches for social network analyses involving the family of exponential random graph models (ERGMs), and presents a new package in *R* for imputing missing values for networks that are assumed to have missingness completely at random (MCAR). Study 1 (Chapter 3) examined binary valued networks at a single time point, Study 2 (Chapter 4) focused on binary networks measured at two time points, and Study 3 (Chapter 5) looked at weighted valued networks measured at a single time point. Across all three studies, results showed that the optimal approach to handling missingness for network data, in terms of reducing coefficient bias and maximizing precision, was the stochastic single imputation method based on observed or true network density; the worst missing data handling methods were the non-stochastic imputation approaches that are included as default options in the popular `statnet` suite of packages for ERGMs.

Nevertheless, missing data is a fact of life for almost every researcher. Like any research involving surveys, in network measurement there are very plausible situations where MCAR or MAR can be assumed and utilized so that all available data can be preserved, rather than discarding any node with missingness. But, without an easy-to-implement means of imputing network data, it seems quite likely that researchers will continue to use node deletion, or one of the problematic non-stochastic methods, as a method of choice for their network analyses. As such, this dissertation also offers a new, user-friendly package for imputing complete datasets based on the various network imputation approaches, called `netimp`. While the package is

simple at this point in time, it can be useful in the present and of course, can also be readily improved upon in the years to come.

**Limitations and Future Research Suggestions**

The results presented in this dissertation, like those from any Monte Carlo simulation, are specific to the conditions examined and should not be overgeneralized. We do not know, for example, whether missingness approaches will behave differently for much larger sized networks, more sparse or dense networks, or networks with more than two time points (as in Study 2); nevertheless, it seems likely that the general pattern of results would not change because we varied these conditions across studies and consistently found the same results. Moreover, although we did not examine the reproduction of data structures (such as types of triads, see for example Smith et al., 2017), there should be no apparent difference in the results for coefficient bias compared to accurately reproduced structures like triad counts given that we assumed MCAR in the data generation process. Other considerations for future research may also include different distributions for the weighted ERGM evaluated in Study 3; this is a relatively new model and it may be the case that optimal imputation methods are not invariant to different data generation mechanisms than those used in the current study.

A more important limitation, compared to those outlined above, for future research to undertake is the use of covariates (predictors) or network structures (e.g., triads) in the data generation, model, or missingness handling processes in order to evaluate approaches under missingness at random (MAR; when missingness is related to a covariate that was measured and can be included in the modeling process). For example, generating networks with dependencies such as school membership, job role, and age. Realistically, networks have dependencies that impact the probability of ties between nodes.

Another important limitation is our use of stochastic single, instead of multiple, imputation. Multiple imputation methods are superior to single imputation methods because any single imputation is one potential plausible and so results from analyses using a singly imputed dataset may be sample-specific and lack replicability. Indeed, the number of imputations one should use in more traditional analyses has been the subject of much discussion with different rules of thumb, but the basic gist is that the greater the missing data amount, the greater the number of imputations that are needed to get replicable standard errors (e.g., von Hippel, 2020). We did not undertake multiple imputation in these studies due to the complexity involved in the pooling of each imputation analyses' results given the already quite complicated and lengthy modeling process involved with ERGMs. Nevertheless, multiple imputation is the next logical step in improving missingness handling for network data, in addition to adding covariates to the missingness imputation process so that MAR can be assumed.

Last but not least, this set of studies relied on one class of models, the ERGM family, and one software package, `statnet`. We selected this class of models because it was the most popular in both education and international research journals (see Chapter 1); we used `statnet` because the authors are the same as those who developed the ERGMs. Nevertheless, latent space models (LSMs) are a viable alternative modeling approach; this class of models are implemented both in `statnet` (`latentnet`) as well as other packages, such as `amen` (Hoff, 2015). LSMs have an advantage over ERGMs in that they do not suffer from degeneracy problems because they take a different approach to estimating network connections, in part by directly modeling distances among nodes in an unobserved, or "latent" space, conditional on network properties (these distances can be defined in a number of ways). This said, LSM results are not as easily

interpreted like ERGMs, and different software make different assumptions and yield different results. Indeed, of the four packages that exist, only `amen` can handle missing data.

As a check on the comparability of `amen` and `statnet` missingness handling, I conducted a single set of simulations for direct, binary valued single time point networks to compare the coefficient estimates from `amen`'s likelihood-based handling of missing data (without imputation) with `amen`'s LSM and `statnet`'s ERGM results with sample density-based stochastic single imputation missingness handling. As shown in Table 7.1, the results are nearly identical across models (note that `amen` uses a probit model and ERGM uses a logit model (probit results = 1.7*logit results). Despite these positive results, it is important to emphasize that ERGM is still the more popular approach for network modeling, potentially because predictor effects in LSM are generally more difficult to interpret.

**Research and Policy Implications**

There are a few key takeaways for applied researchers who want to use network data as part of their research agenda. **First, researchers should determine what they really want to know about the network they are studying before embarking on a network research design**, with a solid understanding of: (a) who constitutes the network, (b) the *nature of the interactions* among nodes, (c) the anticipated *covariates that might predict* those interactions, and (d) how understanding these interactions *maps on to policy implications*. Like any research, defining the population of interest is key for network measurement. For example, should the network comprise only one type of node role, like teachers, or should it include anyone in the building (e.g., teachers, principals, and other specialists)? Knowing this information allows researchers to properly measure all network members instead of accidentally leaving some out that leads to missingness that is likely not missing at random.

**Table 7.1**

*Intercept Coefficient Estimates and Predicted Probabilities from* `amen` *LSMs and* `statnet` *ERGMs with Imputation*

| | amen (Probit) | | amen (Logit Translation) | | statnet (Logit) | Predicted Probabilities | | |
|---|---|---|---|---|---|---|---|---|
| Condition | LSM without imputation | LSM netImp | LSM without imputation | LSM netImp | ERGM netimp | amen LSM without imputation | amen LSM netImp | statnet ERGM netimp |
| Smaller Networks (20 Nodes) | | | | | | | | |
| 20% Density | | | | | | | | |
| 10% Missing | -0.89 | -0.86 | -1.52 | -1.46 | -1.31 | 18% | 19% | 21% |
| 20% Missing | -0.81 | -0.85 | -1.38 | -1.44 | -1.31 | 20% | 19% | 21% |
| 30% Density | | | | | | | | |
| 10% Missing | -0.57 | -0.54 | -0.96 | -0.91 | -0.83 | 28% | 29% | 30% |
| 20% Missing | -0.58 | -0.58 | -0.98 | -0.98 | -0.90 | 27% | 27% | 29% |
| 40% Density | | | | | | | | |
| 10% Missing | -0.26 | -0.29 | -0.45 | -0.49 | -0.45 | 39% | 38% | 39% |
| 20% Missing | -0.29 | -0.36 | -0.49 | -0.62 | -0.56 | 38% | 35% | 36% |
| Larger Networks (50 Nodes) | | | | | | | | |
| 20% Density | | | | | | | | |
| 10% Missing | -0.85 | -0.86 | -1.45 | -1.46 | -1.37 | 19% | 19% | 20% |
| 20% Missing | -0.87 | -0.86 | -1.48 | -1.46 | -1.38 | 19% | 19% | 20% |
| 30% Density | | | | | | | | |
| 10% Missing | -0.55 | -0.54 | -0.93 | -0.91 | -0.86 | 28% | 29% | 30% |
| 20% Missing | -0.52 | -0.51 | -0.89 | -0.86 | -0.80 | 29% | 30% | 31% |
| 40% Density | | | | | | | | |
| 10% Missing | -0.26 | -0.26 | -0.44 | -0.44 | -0.41 | 39% | 39% | 40% |
| 20% Missing | -0.26 | -0.25 | -0.43 | -0.42 | -0.39 | 39% | 40% | 40% |

The nature of the interactions of interest is the next piece of the puzzle – it could be the mere presence or absence of a linkage, or it could be the magnitude associated with the interaction (e.g., do the teachers work together – a binary outcome, or do the teachers work a lot together – a weighted outcome). This understanding informs the type of response options researchers would provide to participants, as well as the type of analysis that would be performed on the data. Importantly, statistical models for weighted data are still in their early stages and are more limited than models for binary data. This said, weighted data can be turned into binary data whereas the reverse is not true.

Next, defining and measuring which covariates (theoretically) will predict the nature of the focal interactions will determine the "other" data collected on participants, besides network interaction data (e.g., years of experience, subject matter taught, etc.); these data will need to be collected from existing records, observations, or self-reported survey data.

Of utmost importance is deeply understanding what the resulting network results will provide in terms of policy implications. As one example, it is possible that analysis results will allow policy makers to understand where an existing network is "working" and where it is not (Are connections more likely for some network members but not others? Maybe teachers who teach younger grade levels are less cohesive than those teaching older grades; the policy implication might be to set aside an hour on Fridays specific for those teaching younger grades to interact). As another example, the research aim may be to improve the network over time so that claims can be made about network growth and cohesion (e.g., professional development used to improve teacher interactions)? If so, a longitudinal data collection process (and analysis) will need to be undertaken, with some careful thinking about whether the network will include new

members at each time point, or the same members throughout the period of the study (recall that incorporating new members at each wave would induce "missingness" at earlier time points).

Finally, researchers can be pro-active in **trying to avoid missing data in the first place**, as well as being proactive **in collecting data from members that might be more likely to be missing in the future** so that missingness assumptions may be checked. Assuming that most networks are measured with online surveys (rather than direct observation), below are some key ideas to consider during the research design phase.

1. Avoid open-ended nomination choices; instead use a drop-down list of restricted potential nominees (or a smart-list that searches a database for potential nominees). This disallows participants from nominating people outside the network bounds.
2. As a follow-up to #1: if there is a fear that someone in the network was missed for the big network list, add one "Other" category for people to nominate other people not on the drop-down or smart list.
3. Pre-determine the number of nominations expected per nominator. Is it two? Five? Seven? One can gather this information with some pilot studying or by looking at prior research.
4. Using information from #3, ask nominators generate a list of nominees before asking them the specifics about the nature of the interactions.
5. Try to use a matrix-type design to ask nominators about the types of interactions they had with their nominees – preferably prepopulated with nominee names crossed with interaction characteristics. This reduces non-response (missingness) because it reduces survey length and thereby reduces participant fatigue.

6. When asking for the magnitude of interactions, it is advisable to use a scale that may be considered interval scale, such as 1-5 or counts, as long as counts are bounded and limited (i.e., for the subsequent analysis as well as participant ease of response).
7. Keep the number of questions about the interactions with nominees to a minimum, again to reduce missingness (or invalid responses) due to participant fatigue.
8. Like any survey, send well-timed reminders and provide incentives for participants to complete their network measurement survey. It is well known that online survey response rates often range from 10 to 30% (e.g., Dillman et al., 2014).
9. Last but not least, *anticipate* and *collect data on predictors of missingness*. This is different from collecting predictors of network connectivity – this is intentionally considering why some participants may not respond to the survey at any given time point. For example, if principals or certain staff within a district are less likely to be responsive, it may be worthwhile to incentivize those network participants differently. As another example, if teachers with more years of experience are less likely to participate for some reason, such as burnout, it may be worthwhile to survey participants about such characteristics at the onset of the study, rather than during network measurement. That way, comparisons can be made about whether responders differ from non-responders on certain characteristics: these results can then be used to determine whether missingness at random is a tenable assumption for missingness imputation purposes.

**Conclusion**

Network analysis is largely a newcomer to educational research, representing just 3% of peer-reviewed research articles in well-respected educational research journals in the past five

years (see Chapter 1). However, studies focused on improving teaching and learning may increasingly shift toward a more sociological, network-type orientation: a lens that assumes the transmission of information occurs as a social, rather than individual, phenomenon – such as teacher advice networks for improving teaching practice (e.g., Thompson et al., 2019). For these types of studies, handling missing data appropriately is important for obtaining accurate and precise network information.

In this spirit, the results of this dissertation demonstrate the need for a modern missingness approach compared with current default methods like node deletion. Moreover, this dissertation also provides a practical solution that is better than node deletion; specifically, a new *R* package was developed that provides researchers with a stochastic single imputation, density-based method they can use to obtain complete datasets, prior to computing descriptive statistics or conducting inferential analyses. It is hoped that future research can build directly on this work, especially by incorporating covariates into the modeling process, and including use of multiple rather than single imputation.

# REFERENCES

Allison, P. D. (2009). *Missing Data*. Thousand Oaks, CA: Sage.

Bodner, T. E. (2006). Missing data: Prevalence and reporting practices. *Psychological Reports, 99*(3), 675-680. https://doi.org/10.2466/PR0.99.7.675-680

Breiger R., & Pattison, P. (1986). Cumulated social roles: The duality of persons and their algebras. *Social Networks, 8*, 215-256. https://doi.org/10.1016/0378-8733(86)90006-7

Carnegie, N., Krivitsky, P., Hunter, D., & Goodreau, S. (2015). An approximation method for improving dynamic network model fitting. *Journal of Computational and Graphical Statistics, 24*(2), 502-519. doi: 10.1080/10618600.2014.903087

Carolan, B. (2014). *Social Network Analysis and Education: Theory, Methods and Applications*. Thousand Oaks, CA: SAGE.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods, 6*(4), 330-351. https://doi.org/10.1037/1082-989X.6.4.330

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*. 1695. https://igraph.org

Dillman, D. A., Smyth, J. D., & Christian, L. M. (2014). *Internet, Phone, Mail, and Mixed-Mode Surveys: The Tailored Response Method, 4th Ed.* Hoboken, NJ: John Wiley & Sons, Inc.

Eidsaa, M., & Almaas, E. (2013). S-core network decomposition: A generalization of k-core analysis to weighted networks. *Physical Review E, 88*(6), 062819. doi:10.1103/physreve.88.062819

Enders, C. K. (2010). *Applied Missing Data Analysis*. New York, NY: Guilford Press.

Feldman, S., Abe., N., & Sanders, E. A. (2018). *Report: Partnership for ambitious science teacher leadership: Exploring in-service networks for professional learning in the context of new science teaching and learning standards*. Seattle, WA

Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social Networks*, *1*(3), 215-239.

Gile, K. J., & Handcock, M. S. (2016). Analysis of networks with missing data with application to the National Longitudinal Study of Adolescent Health. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 66*(3), 501-519. doi:10.1111/rssc.12184

Golbeck, J. (2013). *Analyzing the social web*. Elsevier.

Goodreau, S., Kitts, J. A., & Morris, M. (2009). Birds of a feather, or friend of a friend? Using exponential random graph models to investiage adolescent social networks. *Demography, 46*(1), 103-125. doi: 10.1353/dem.0.0045

Graham, J. W. (2012). *Missing Data Analysis and Design*. New York, NY: Springer.

Granovetter, M. S. (1973). The strength of weak ties. *American Journal of Sociology, 78*(6), 1360-1380.

Grunspan, D. Z., Wiggins, B. L., & Goodreau, S. M. (2014). Understanding classrooms through social network analysis: A primer for social network analysis in education research. *CBE Life Sci Educ*, *13*(2), 167-179. doi:10.1187/cbe.13-08-0162

Handcock, M. (2003). *Assessing Degeneracy in Statistical Models of Social Networks*. CSSS working paper 39. University of Washington. Seattle, WA. Retrieved from https://www.csss.washington.edu/research/working-papers/39

Handcock, M. S., & Gile, K. J. (2010). Modeling social networks from sampled data. *The Annals of Applied Statistics*, *4*(1), 5-25. doi: 10.1214/08-AOAS221

Handcock, M., Hunter, D., Butts, C., Goodreau, S., Krivitsky, P., Bender-deMoll, S., & Morris, M. (2016). *statnet: Software Tools for the Statistical Analysis of Network Data* (Version R package version 2016.9). Retrieved from CRAN.R-project.org/package=statnet

Handcock, M., Hunter, D., Butts, C., Goodreau, S., & Morris, M. (2008). statnet: Software tools for the representation, visualization, analysis and simulation of network Data. *Journal of Statistical Software, 24*, 1-11. doi: 10.18637/jss.v024.i01

Harris, J. (2014) *An Introduction to Exponential Random Graph Modeling*. Thousand Oaks, CA: SAGE.

Hoff, P. D. (2015). *Dyadic Data Analysis with amen*. arXiv:1506.08237.

Hoogland, J. J., & Boomsma, A. (1998). Robustness studies in covariance structure modeling: An overview and a meta-analysis. *Sociological Methods & Research*, *26*(3), 329-367. https://doi.org/10.1177/0049124198026003003

Huisman, M., & Steglich, C. (2008). Treatment of non-response in longitudinal network studies. *Social Networks*, *30*(4), 297-308. https://doi.org/10.1016/j.socnet.2008.04.004

Huisman, M. (2014). Imputation of missing network data: Some simple procedures. *Encyclopedia of Social Network Analysis and Mining*, 707–715. https://doi.org/10.1007/978-1-4614-6170-8_394

Hunter D.R., Handcock, M.S., Butts C.T., Goodreau S.M., Morris, M. (2008b). Ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3), doi: 10.18637/jss.v024.i03

Knoke, D. & Yang, S. (2011). *Social Network Analysis*. Thousand Oaks, CA: SAGE.

Koskinen, J. H., Robins, G. L., Wang, P., & Pattison, P. E. (2013). Bayesian analysis for partially observed network data, missing ties, attributes and actors. *Social Networks, 35*(4), 514-527. doi:10.1016/j.socnet.2013.07.003

Krause, R. W., Huisman, M., & Snijders, T. A. B. (2018). Multiple imputation for longitudinal network data. *Italian Journal of Applied Statistics, 30*(1), 1-22. doi:10.26398/IJAS.0030-002

Krivitsky, P. N. (2014). Exponential-family random graph models for valued networks. *Electronic Journal of Statistics, 6*, 1100-1128. doi:10.1214/12-ejs696.

Krivitsky, P., Butts, C.T. (2017). Exponential-family random graph models for rank-order relational data. *Sociological Methodology*, 47(1), doi: https://doi.org/10.1177/0081175017692623

Krivitsky, P., & Handcock, M. (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society, Series B, 76*(1), 29-46. https://doi.org/10.1111/rssb.12014

Krivitsky, P,, & Handcock, M. (2021). tergm: Fit, simulate and diagnose models for network evolution based on exponential-family random graph models. The Statnet Project. R package version 4.0.2, https://CRAN.R-project.org/package=tergm

Lazega, E. (2001). The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership. *Oxford University Press*. doi: 10.1093/acprof:oso/9780199242726.001.0001

Leifeld, P., & Cranmer, S. J. (2019). A theoretical and empirical comparison of the temporal exponential random graph model and the stochastic actor-oriented model. *Network Science*, 7(1). doi: 10.1017/nws.2018.26

Leifeld, P., Cranmer , S. J., & Desmarais, B. A. (2018). Temporal exponential random graph models with btergm estimation and bootstrap confidence intervals. *Journal of Statistical Software, 83*(6). doi: 10.18637/jss.v083.i06

Lusher, D., & Robins, G. (2013). *Formation of Social Network Structure*. New York: Cambridge University Press.

Ouzienko, V., & Obradovic, Z. (2014). Imputation of missing links and attributes in longitudinal social surveys. *Machine Learning, 95*, 329-356. https://doi.org/10.1007/s10994-013-5420-1

Pilny, A., & Atouba, Y. (2017). Modeling valued organizational communication networks using exponential random graph models. *Management Communication Quarterly, 32*(2), 250-264. doi:10.1177/0893318917737179

Peugh, J. L., & Enders, C. K. (2004). Missing data in educational research: A review of reporting practices and suggestions for improvement. *Review of Educational Research, 74*(4), 525-556. https://doi.org/10.3102/00346543074004525

Robins, G., & Lusher, D. (2013). What are exponential random graph models? In D. Lusher, J. Koskinen, & G. Robins (Eds.), *Exponential Random Graph Models for Social Networks* (pp. 9-15). New York, NY: Cambridge.

Robins, G., Pattison, P., & Woolcock, J. (2004). Missing data in networks: Exponential random graph (p*) models for networks with nonrespondents, *Social Networks, 26*, 257-283. doi:10.1016/j.socnet.2004.05.001

Rubin, D. B. (1976). Inference and missing data. *Biometrika, 63*(3), 581-592. https://doi.org/10.2307/2335739

Sampson, S. (1969). Crisis in a cloister. *Unpublished doctoral dissertation, Cornell University*.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods, 7*(2), 147-177. https://doi.org/10.1037/1082-989X.7.2.147

Scott, T. A. (2015). Analyzing policy networks using valued exponential random graph models: Do government-sponsored collaborative groups enhance organizational networks? *Policy Studies Journal, 44*(2), 215-244. doi:10.1111/psj.12118

Smith, J. A., & Moody, J. (2013). Structural effects of network sampling coverage I: Nodes missing at random. *Social Networks, 35*(4), 652-668. https://doi.org/10.1016/j.socnet.2013.09.003

Smith, J. A., Moody, J., & Morgan, J. H. (2017). Network sampling coverage II: The effect of non-random missing data on network measurement. *Social Networks, 48*, 78-99. https://doi.org/10.1016/j.socnet.2016.04.005

Snijders, T. A. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure, 3*, 1-40.

Snijders, T. A., Koskinen, J., & Schweinberger, M. (2010). Maximum likelihood estimation for social network dynamics. *Annals of Applied Statistics, 4*(2), 567-588. doi:10.1214/09-AOAS313

Thompson, J., Richards, J., Shim, S-Y., Lohwasser, K., Chew, C., Sjoberg, B.& Morris, A. (2019). Networked PLCs: Footholds into creating and improving knowledge of ambitious and equitable teaching practices in RPP. *AERA Open*. https://doi.org/10.1177/2332858419875718

Valente, T. W., Gallaher, P., & Mouttapa, M. (2004). Using social networks to understand and prevent substance use: A transdisciplinary perspective. *Substance Use and Misuse, 39*, 1685-1712. doi:10.1081/JA-200033210

von Hippel, P. T. (2020). How many Imputations do you need? A two-stage calculation using a quadratic rule. *Sociological Methods & Research, 49*(3), 699-718. doi:10.1177/0049124117747303

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*. New York, NY: Cambridge University Press.

White, D. R. (2014). Kinship, class, and community. In J. Scott & P. J. Carrington (Eds.), *The SAGE Handbook of Social Network Analysis* (pp. 129-147). London: SAGE.

Windschitl, M., Thompson, J., Braaten, M., & Stroupe, D. (2012). The beginner's repetoire: Proposing a core set of instructional practices and tools for teachers of science. *Science Education, 96*, 878-903. doi: 10.1002/sce.21027

Yang, S., Keller, F., Zheng, L. (2017) *Social Network Analysis: Methods and Examples*. Thousand Oaks, CA: SAGE.

Young, J. (2011). How do they 'end up together'? A social network analysis of self-control, homophily, and adolescent relationships. *Journal of Quantitative Criminology 27*(3), 251-273. doi: 10.1007/s10940-010-9105-7

## Appendix A

Sample *R* Code for Study 1

```
########################
###NETWORK GENERATION###
########################
r.ntwk.diag<-function(m) {
  n<-ncol(m)
  #density
  d<-sum(m, na.rm = TRUE)/(n^2-n)
  #reciprocity. short but harder to read formula. multiply preserves 1*1
entries.
  r<-sum(m*t(m), na.rm = TRUE)/sum(m, na.rm = TRUE)
  return(c(sum(m, na.rm = TRUE), sum(m*t(m), na.rm = TRUE),d,r))
}

r.ntwk<-function(n, density, recip) {
  possible<-n^2-n
  r.upper.ties<-possible*density*recip/2
  nr.ties<-possible*density-r.upper.ties*2

  #create matrix to store draws
  draws<-matrix(0, nrow = (n^2-n)/2, ncol = 3)
  count=1
  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      draws[count,1]<-i
      draws[count,2]<-j
      count=count+1
    }
  }

  #draw recipricol and non-reciprocal ties
  ties.sample<-sample.int(n=(n^2-n)/2, size=r.upper.ties+nr.ties,
replace=FALSE)
  #save in draws matrix as 1
  for (x in 1:length(ties.sample)) {
    draws[ties.sample[x], 3]<-1
  }
  #draw out the non-reciprocal
  nr.ties.sample<-sample(ties.sample, size=nr.ties, replace=FALSE)
  #recode nr in draws matrix as 2, nr flips as 3
  for (x in 1:length(nr.ties.sample)) {
    flip<-rbinom(1,1,.5)
    draws[ties.sample[x], 3]<-2+flip
  }

  #construct the network matrix
  ntwk<-matrix(0, nrow = n, ncol = n)
  for (x in 1:nrow(draws)) {
    i<-draws[x,1]
    j<-draws[x,2]
    if (draws[x,3]==1) {
      ntwk[i,j]<-1
      ntwk[j,i]<-1
```

```
    }
    if (draws[x,3]==2) {
      ntwk[i,j]<-1
    }
    if (draws[x,3]==3) {
      ntwk[j,i]<-1
    }
  }
  return(ntwk)
}

########################################
###ERGM WRAPPER FUNCTION TO SAVE RESUTS###
########################################
erg = function(ntwk) {
  net = network(ntwk, directed = T)
  net.fit <- ergm(
    net~edges + mutual)

  result<-matrix(as.numeric(NA), nrow=1, ncol=10)
  #colnames(result)<-c( )
  if (is.na(net.fit)){
    return(result)
  }
  result[1,1]<-summary(net.fit)$coefs[1,1]
  result[1,2]<-summary(net.fit)$coefs[1,2]
  result[1,3]<-summary(net.fit)$coefs[1,3]
  result[1,4]<-summary(net.fit)$coefs[1,4]
  result[1,5]<-summary(net.fit)$coefs[2,1]
  result[1,6]<-summary(net.fit)$coefs[2,2]
  result[1,7]<-summary(net.fit)$coefs[2,3]
  result[1,8]<-summary(net.fit)$coefs[2,4]
  result[1,9]<-summary(net.fit)$aic
  result[1,10]<-summary(net.fit)$bic
  return(result)
}

######################
###GENERATE MISSING###
######################
no.miss1 = mapply(erg, netlist, SIMPLIFY = F)
no.miss1 = do.call(rbind.data.frame, no.miss1)

#####################################
###FUNCTION THAT CREATES MISSING DATA###
#####################################
missing.data = function(ntwk1, per.miss1){
  n.miss1<-round(per.miss1*nrow(ntwk1), 0)
  miss.rows1<-sample.int(nrow(ntwk1), size = n.miss1, replace=FALSE)
  return(list(miss.rows1))
}

###IMPUTATION METHODS###

all.zeros<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-0}
  return(ntwk)
```

```
}

all.ones<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-1}
  return(ntwk)
}

random<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z)}
  return(ntwk)
}

random.density<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-d1, d1))}
  return(ntwk)
}

random.density.2<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-d2, d2))}
  return(ntwk)
}

random.density.3<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-d3, d3))}
  return(ntwk)
}

deletion<-function(ntwk, miss.rows1, miss.rows2) {
  ntwk=ntwk[-c(miss.rows1,miss.rows2),-c(miss.rows1,miss.rows2)]
  return(ntwk)
}

random.dens3<-function(ntwk, miss.rows) {
  ntwk.miss=ntwk[-miss.rows,-miss.rows]
  probs=sum(ntwk.miss)/(nrow(ntwk.miss)*(nrow(ntwk.miss)-1))
  ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-probs,
probs))
  return(ntwk)
}

#####################################################################
###GENERATES NETWORKS AND CREATES LISTS FOR ALL OF THE NETWORKS###
#####################################################################
set.seed(1988)
nsims=1000
z=20
d1 = .2
d2 = .3
d3 = .4

netlist2 = list(nsims)
netlist3 = list(nsims)

netlist = list(nsims)
for(i in 1:nsims) {
  netlist[[i]] = r.ntwk(z, d1, .5)
  netlist2[[i]] = r.ntwk(z, d2, .5)
```

```
  netlist3[[i]] = r.ntwk(z, d3, .5)
}

###################################
###NO MISSING DATA ERGM ANALYSIS###
###################################
no.miss1 = mapply(erg, netlist, SIMPLIFY = F)
no.miss1 = do.call(rbind.data.frame, no.miss1)

no.miss2 = mapply(erg, netlist2, SIMPLIFY = F)
no.miss2 = do.call(rbind.data.frame, no.miss2)

no.miss3 = mapply(erg, netlist3, SIMPLIFY = F)
no.miss3 = do.call(rbind.data.frame, no.miss3)


# ###ADDING COLUMN NAMES AND OTHER PERTINENT INFORMATION###
full_d20<-as.data.frame(no.miss1)
colnames(full_d20)=c("Tie", "Tie_std", "Tie_Z", "Tie_p", "Mut", "Mut_std",
"Mut_z", "Mut_p", "AIC", "BIC")
full_d20$NodeSz = 20
full_d20$dens = .2
full_d20$Sim = 1:nsims

full_d30<-as.data.frame(no.miss2)
colnames(full_d30)=c("Tie", "Tie_std", "Tie_Z", "Tie_p", "Mut", "Mut_std",
"Mut_z", "Mut_p", "AIC", "BIC")
full_d30$NodeSz = 20
full_d30$dens = .3
full_d30$Sim = 1:nsims

full_d40<-as.data.frame(no.miss3)
colnames(full_d40)=c("Tie", "Tie_std", "Tie_Z", "Tie_p", "Mut", "Mut_std",
"Mut_z", "Mut_p", "AIC", "BIC")
full_d40$NodeSz = 20
full_d40$dens = .4
full_d40$Sim = 1:nsims

FULL_20node = rbind(full_d20, full_d30, full_d40)

###################################
###MISSING DATA for 20% Density###
###################################
miss1 = mapply(missing.data, netlist, .1, SIMPLIFY = F)
miss2 = mapply(missing.data, netlist, .2, SIMPLIFY = F)

#########20% DENSITY; 10% MISSING#########
all.zero_2010 = mapply(all.zeros, netlist, unlist(miss1, recursive=F),
SIMPLIFY = F)
all.zeros_2010 = mapply(erg, all.zero_2010)
#all.zeros_2010 = do.call(rbind.data.frame, all.zeros_2010)
all.zeros_2010 = as.data.frame(t(all.zeros_2010))

all.one_2010 = mapply(all.ones, netlist, unlist(miss1, recursive=F), SIMPLIFY
= F)
all.ones_2010 = mapply(erg, all.one_2010)
#all.one_2010 = do.call(rbind.data.frame, all.one_2010)
```

```
all.ones_2010 = as.data.frame(t(all.ones_2010))

eq_rand_2010 = mapply(random, netlist, unlist(miss1, recursive=F), SIMPLIFY =
F)
eq_rands_2010 = mapply(erg, eq_rand_2010)
#eq_rand_2010 = do.call(rbind.data.frame, eq_rand_2010)
eq_rands_2010 = as.data.frame(t(eq_rands_2010))

kn_den_2010 = mapply(random.density, netlist, unlist(miss1, recursive=F),
SIMPLIFY = F)

kn_dens_2010 = mapply(erg, kn_den_2010)
#kn_den_2010 = do.call(rbind.data.frame, kn_den_2010)
kn_dens_2010 = as.data.frame(t(kn_dens_2010))

unkn_den_2010 = mapply(random.dens3, netlist, unlist(miss1, recursive=F),
SIMPLIFY = F)
unkn_dens_2010 = mapply(erg, unkn_den_2010)
#unkn_den_2010 = do.call(rbind.data.frame, unkn_den_2010)
unkn_dens_2010 = as.data.frame(t(unkn_dens_2010))

colnames(all.zeros_2010)=c("Tie_A1_M1", "Tie_std_A1_M1", "Tie_z_A1_M1",
"Tie_p_A1_M1", "Mut_e_A1_M1", "Mut_std_A1_M1", "Mut_z_A1_M1", "Mut_p_A1_M1",
"AIC_A1_M1", "BIC_A1_M1")
colnames(all.ones_2010)=c("Tie_A2_M1", "Tie_std_A2_M1", "Tie_z_A2_M1",
"Tie_p_A2_M1", "Mut_e_A2_M1", "Mut_std_A2_M1", "Mut_z_A2_M1", "Mut_p_A2_M1",
"AIC_A2_M1", "BIC_A2_M1")
colnames(eq_rands_2010)=c("Tie_A3_M1", "Tie_std_A3_M1", "Tie_z_A3_M1",
"Tie_p_A3_M1", "Mut_e_A3_M1", "Mut_std_A3_M1", "Mut_z_A3_M1", "Mut_p_A3_M1",
"AIC_A3_M1", "BIC_A3_M1")
colnames(kn_dens_2010)=c("Tie_A4_M1", "Tie_std_A4_M1", "Tie_z_A4_M1",
"Tie_p_A4_M1", "Mut_e_A4_M1", "Mut_std_A4_M1", "Mut_z_A4_M1", "Mut_p_A4_M1",
"AIC_A4_M1", "BIC_A4_M1")
colnames(unkn_dens_2010)=c("Tie_A5_M1", "Tie_std_A5_M1", "Tie_z_A5_M1",
"Tie_p_A5_M1", "Mut_e_A5_M1", "Mut_std_A5_M1", "Mut_z_A5_M1", "Mut_p_A5_M1",
"AIC_A5_M1", "BIC_A5_M1")

all_2010 = cbind(all.zeros_2010, all.ones_2010, eq_rands_2010, kn_dens_2010,
unkn_dens_2010)
```

## Appendix B

Sample *R* Code for Study 2

```
########################
###NETWORK GENERATION###
########################
r.ntwk.diag<-function(m) {
  n<-ncol(m)
  #density
  d<-sum(m, na.rm = TRUE)/(n^2-n)
  #reciprocity. short but harder to read formula. multiply preserves 1*1
entries.
  r<-sum(m*t(m), na.rm = TRUE)/sum(m, na.rm = TRUE)
  return(c(sum(m, na.rm = TRUE), sum(m*t(m), na.rm = TRUE),d,r))
}

r.ntwk<-function(n, density, recip) {
  possible<-n^2-n
  r.upper.ties<-possible*density*recip/2
  nr.ties<-possible*density-r.upper.ties*2

  #create matrix to store draws
  draws<-matrix(0, nrow = (n^2-n)/2, ncol = 3)
  count=1
  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      draws[count,1]<-i
      draws[count,2]<-j
      count=count+1
    }
  }

  #draw recipricol and non-reciprocal ties
  ties.sample<-sample.int(n=(n^2-n)/2, size=r.upper.ties+nr.ties,
replace=FALSE)
  #save in draws matrix as 1
  for (x in 1:length(ties.sample)) {
    draws[ties.sample[x], 3]<-1
  }
  #draw out the non-reciprocal
  nr.ties.sample<-sample(ties.sample, size=nr.ties, replace=FALSE)
  #recode nr in draws matrix as 2, nr flips as 3
  for (x in 1:length(nr.ties.sample)) {
    flip<-rbinom(1,1,.5)
    draws[ties.sample[x], 3]<-2+flip
  }

  #construct the network matrix
  ntwk<-matrix(0, nrow = n, ncol = n)
  for (x in 1:nrow(draws)) {
    i<-draws[x,1]
    j<-draws[x,2]
    if (draws[x,3]==1) {
      ntwk[i,j]<-1
      ntwk[j,i]<-1
```

```
    }
    if (draws[x,3]==2) {
      ntwk[i,j]<-1
    }
    if (draws[x,3]==3) {
      ntwk[j,i]<-1
    }
  }
  return(ntwk)
}

#####################################################
###STERGM WRAPPER TO SAVE RESULTS OF EACH ANALYSIS###
#####################################################
strgm<- function(ntwk1, ntwk2) {
  net1 <- network(ntwk1, directed=T)
  net2 <- network(ntwk2, directed=T)
  net <- list()
  net[[1]] <- net1
  net[[2]] <- net2
  net.fit <- tryCatch(stergm(
    net,
    formation = ~ edges+mutual,
    dissolution = ~ edges+mutual,
    estimate = "CMLE",
    control = control.stergm(
      init.form = NULL,
      init.diss = NULL,
      init.method = 0,
      force.main = FALSE,
      SA.restart.on.err = FALSE
    ),
    times = 1:2
  ), error=function(e){cat("ERROR :",conditionMessage(e), "\n");return(NA)})

  result<-matrix(as.numeric(NA), nrow=1, ncol=16)
  colnames(result)<-c( )
  if (is.na(net.fit)){
    return(result)
  }
  result[1,1]<-summary(net.fit)$formation$coef[1,1]
  result[1,2]<-summary(net.fit)$formation$coef[1,2]
  result[1,3]<-summary(net.fit)$formation$coef[1,4]
  result[1,4]<-summary(net.fit)$formation$coef[2,1]
  result[1,5]<-summary(net.fit)$formation$coef[2,2]
  result[1,6]<-summary(net.fit)$formation$coef[2,4]
  result[1,7]<-summary(net.fit)$formation$aic
  result[1,8]<-summary(net.fit)$formation$bic
  result[1,9]<-summary(net.fit)$dissolution$coef[1,1]
  result[1,10]<-summary(net.fit)$dissolution$coef[1,2]
  result[1,11]<-summary(net.fit)$dissolution$coef[1,4]
  result[1,12]<-summary(net.fit)$dissolution$coef[2,1]
  result[1,13]<-summary(net.fit)$dissolution$coef[2,2]
  result[1,14]<-summary(net.fit)$dissolution$coef[2,4]
  result[1,15]<-summary(net.fit)$dissolution$aic
  result[1,16]<-summary(net.fit)$dissolution$bic
```

```
  return(result)
}

#############################################################################
###FUNCTION THAT CREATES MISSING DATA WHERE NO NODE HAS MISSING DATA IN BOTH
TIMEPOINTS###
#############################################################################
missing.data = function(ntwk1, ntwk2,per.miss1, per.miss2, overlap=F){
  n.miss1<-round(per.miss1*nrow(ntwk1), 0)
  n.miss2<-round(per.miss2*nrow(ntwk2), 0)

  if (overlap==TRUE) {
    miss.rows1<-sample.int(nrow(ntwk1), size = n.miss1, replace=FALSE)
    miss.rows2<-sample.int(nrow(ntwk2), size = n.miss2, replace=FALSE)
  }
  if (overlap==FALSE) {
    miss.rows1<-sample.int(nrow(ntwk1), size = n.miss1, replace=FALSE)
    miss2.possible<-setdiff(1:nrow(ntwk2), miss.rows1)
    miss.rows2<-sample(miss2.possible, size = n.miss2, replace=FALSE)
  }
  return(list(miss1=miss.rows1, miss2=miss.rows2))
}

########################
###IMPUTATION METHODS###
########################

all.zeros<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-0}
  return(ntwk)
}

all.ones<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-1}
  return(ntwk)
}

random<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z)}
  return(ntwk)
}

random.density<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-d1, d1))}
  return(ntwk)
}

random.density.2<-function(ntwk, miss.rows) {
  {ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-d2, d2))}
  return(ntwk)
}

deletion<-function(ntwk, miss.rows1, miss.rows2) {
  ntwk=ntwk[-c(miss.rows1,miss.rows2),-c(miss.rows1,miss.rows2)]
  return(ntwk)
}
```

```
impute<-function(ntwk1, ntwk2, miss.rows1, miss.rows2) {
  ntwk1[miss.rows1,]=ntwk2[miss.rows1,]
  ntwk2[miss.rows2,]=ntwk1[miss.rows2,]
  return(list(impute.data1=ntwk1, impute.data2=ntwk2))
}

random.dens3<-function(ntwk, miss.rows) {
  ntwk.miss=ntwk[-miss.rows,-miss.rows]
  probs=sum(ntwk.miss)/(nrow(ntwk.miss)*(nrow(ntwk.miss)-1))
  ntwk[miss.rows, ]<-sample(c(0,1), replace=TRUE, size=z, prob=c(1-probs,
probs))
  return(ntwk)
}

record<- function(ntwk, miss.rows) {
  ntwk.miss=ntwk[-miss.rows,-miss.rows]
  probs=sum(ntwk.miss)/(nrow(ntwk.miss)*(nrow(ntwk.miss)-1))
  return(probs)
}

##########################################################
###GENERATES AND CREATES LISTS FOR ALL OF THE NETWORKS###
##########################################################
set.seed(0)
nsims=1000
z=20
d1 = .3
d2 = .4

netlist2 = list(nsims)

netlist = list(nsims)
for(i in 1:nsims) {
  netlist[[i]] = r.ntwk(z, d1, .5)
  netlist2[[i]] = r.ntwk(z, d2, .5)

}

#####################################
###NO MISSING DATA STERGM ANALYSIS###
#####################################
no.miss = mapply(strgm, netlist, netlist2, SIMPLIFY = F)
no.miss = do.call(rbind.data.frame, no.miss)


# ###ADDING COLUMN NAMES AND OTHER PERTINENT INFORMATION###
full<-as.data.frame(no.miss)
colnames(full)[colnames(full)=="V1"] <- "F_Tie"
colnames(full)[colnames(full)=="V2"] <- "F_Tie_SE"
colnames(full)[colnames(full)=="V3"] <- "F_Tie_P-value"

colnames(full)[colnames(full)=="V4"] <- "F_Mutual"
colnames(full)[colnames(full)=="V5"] <- "F_Mutual_SE"
colnames(full)[colnames(full)=="V6"] <- "F_Mutual_P-value"
colnames(full)[colnames(full)=="V7"] <- "F_AIC"
colnames(full)[colnames(full)=="V8"] <- "F_BIC"
```

```
colnames(full)[colnames(full)=="V9"] <- "D_Tie"
colnames(full)[colnames(full)=="V10"] <- "D_Tie_SE"
colnames(full)[colnames(full)=="V11"] <- "D_Tie_P-value"

colnames(full)[colnames(full)=="V12"] <- "D_Mutual"
colnames(full)[colnames(full)=="V13"] <- "D_Mutual_SE"
colnames(full)[colnames(full)=="V14"] <- "D_Mutual_P-value"
colnames(full)[colnames(full)=="V15"] <- "D_AIC"
colnames(full)[colnames(full)=="V16"] <- "D_BIC"

full$Density1 <- d1
full$Density2 <- d2
full$Recip1 <- .5
full$Recip2 <- .5
full$Miss_Type <- "None"
full$Miss <- 0
full$Sim <- 1:nsims


###################################
### 10% missing STERGM ANALYSIS ###
###################################
set.seed(1)

miss1 = mapply(missing.data, netlist, netlist2, .1, 0, SIMPLIFY = F)
miss2 = mapply(missing.data, netlist, netlist2, 0, .1, SIMPLIFY = F)
miss3 = mapply(missing.data, netlist, netlist2, .1, .1, SIMPLIFY = F)

############################
###10% missing at t1 only###
############################
all.zero1 = mapply(all.zeros, netlist, lapply(miss1, function(x)x$miss1),
SIMPLIFY = F)
zero.results10T1 = mapply(strgm, all.zero1, netlist2, SIMPLIFY = F)
zero.results10T1 = do.call(rbind.data.frame, zero.results10T1)

all.ones1 = mapply(all.ones, netlist, lapply(miss1, function(x)x$miss1),
SIMPLIFY = F)
one.results10T1 = mapply(strgm, all.ones1, netlist2, SIMPLIFY = F)
one.results10T1 = do.call(rbind.data.frame, one.results10T1)

random1 = mapply(random, netlist, lapply(miss1, function(x)x$miss1), SIMPLIFY
= F)
random.results10T1 = mapply(strgm, random1, netlist2, SIMPLIFY = F)
random.results10T1 = do.call(rbind.data.frame, random.results10T1)

random.density1 = mapply(random.density, netlist, lapply(miss1,
function(x)x$miss1), SIMPLIFY = F)
random.density.results10T1 = mapply(strgm, random.density1, netlist2,
SIMPLIFY = F)
random.density.results10T1 = do.call(rbind.data.frame,
random.density.results10T1)

deletion1 = mapply(deletion, netlist,
lapply(miss1,function(x)x$miss1),lapply(miss1,function(x)x$miss2), SIMPLIFY =
F)
```

```
deletion2 = mapply(deletion, netlist2,
lapply(miss1,function(x)x$miss1),lapply(miss1,function(x)x$miss2), SIMPLIFY =
F)
deletion.results10T1 = mapply(strgm, deletion1, deletion2, SIMPLIFY = F)
deletion.results10T1 = do.call(rbind.data.frame, deletion.results10T1)

impute1 = mapply(impute, netlist, netlist2, lapply(miss1,function(x)x$miss1),
lapply(miss1,function(x)x$miss2), SIMPLIFY = F)
impute.results10T1 = mapply(strgm, lapply(impute1,function(x)x$impute.data1),
lapply(impute1,function(x)x$impute.data2), SIMPLIFY = F)
impute.results10T1 = do.call(rbind.data.frame, impute.results10T1)

rand.dens1 = mapply(random.dens3, netlist, lapply(miss1, function(x)x$miss1),
SIMPLIFY = F)
rand.dens.results10T1 = mapply(strgm, rand.dens1, netlist2, SIMPLIFY = F)
rand.dens.results10T1 = do.call(rbind.data.frame, rand.dens.results10T1)
```

## Appendix C

Sample *R* Code for Study 3

```
library(statnet)
library(MASS)
library(purrr)

#10 simulations to start with:
#Node Sizes:       10, 20, 50, 100
#Missing amounts:  10% (M1), 20% (M2), 30% (M3)
#Missing approach:   (1) deletion, (2) discrete uniform, (3)random
multivariate normal

########################
###NETWORK GENERATION###
########################
wght_mat = function(n, mean , cov, var) {
  sigma = matrix(cov, n, n)
  diag(sigma) = var
  mu = rep(mean, n)
  test=round(mvrnorm(n, mu, sigma))
  diag(test)=0
  test[test<0]=0
  test[test>3]=3
  return(test)
}

#################################
###ERGM WRAPPER TO SAVE RESULTS###
#################################
ergms = function(ntwk) {
  net = as.network(ntwk, directed = T, matrix.type = "a", ignore.eval = F,
names.eval = "nominations")
  net.fit <- ergm(
    net~sum + mutual,
    response = "nominations",
    reference = ~StdNormal)

  result<-matrix(as.numeric(NA), nrow=1, ncol=10)
  #colnames(result)<-c( )
  if (is.na(net.fit)){
    return(result)
  }
  result[1,1]<-summary(net.fit)$coefs[1,1]
  result[1,2]<-summary(net.fit)$coefs[1,2]
  result[1,3]<-summary(net.fit)$coefs[1,3]
  result[1,4]<-summary(net.fit)$coefs[1,4]
  result[1,5]<-summary(net.fit)$coefs[2,1]
  result[1,6]<-summary(net.fit)$coefs[2,2]
  result[1,7]<-summary(net.fit)$coefs[2,3]
  result[1,8]<-summary(net.fit)$coefs[2,4]
  result[1,9]<-summary(net.fit)$aic
  result[1,10]<-summary(net.fit)$bic
  return(result)
}
```

```
#############################################
###Single timepoint missing data generation###
#############################################
missing.data = function(ntwk1, per.miss1){
  n.miss1<-round(per.miss1*nrow(ntwk1), 0)
  miss.rows1<-sample.int(nrow(ntwk1), size = n.miss1, replace=FALSE)
  return(list(miss.rows1))
}

################################
###NETWORK GENERATION PROCESS###
################################
n=10
mu = 1.5
cov = .325
var = 1.67
sigma = matrix(cov,n,n)
diag(sigma) = var
mean = rep(1.5, n)
set.seed(2931)
nsims=5
netlist = list(nsims)
for(i in 1:nsims) {
  netlist[[i]] = wght_mat(n, mu, cov, var)
}

############################
###MISSING AMOUNT LEVELS###
############################
miss1 = mapply(missing.data, netlist, .1, SIMPLIFY = F)
miss2 = mapply(missing.data, netlist, .2, SIMPLIFY = F)
miss3 = mapply(missing.data, netlist, .3, SIMPLIFY = F)

#####################################
###HOW MISSING DATA IS ACCOUNTED FOR###
#####################################
unif.random<-function(ntwk1, miss.rows1) {
  {ntwk1[miss.rows1, ]<-sample(c(0,1,2,3), size = n, replace = T, prob =
c(.25,.25,.25,.25))}
  return(ntwk1)
}


deletion<-function(ntwk, miss.rows1) {
  ntwk=ntwk[-miss.rows1,-miss.rows1]
  return(ntwk)
}


multi.random<-function(ntwk1, miss.rows1, per.miss1) {
  {ntwk1[miss.rows1, ]<-round(mvrnorm(round(per.miss1*nrow(ntwk1)), mean,
sigma), 0)
  diag(ntwk1) = 0
  ntwk1[miss.rows1, ][ntwk1[miss.rows1, ]<0] = 0
  ntwk1[miss.rows1, ][ntwk1[miss.rows1, ]>3] = 3
  }
  return(ntwk1)
```

```
}

##################10 NODES#######################
################################################
############### NO MISSING ####################

no.miss = mapply(ergms, netlist)
t1 = t(no.miss)

full_n10<-as.data.frame(t1)
colnames(full_n10)=c("Full_sum_e", "Full_sum_std", "Full_sum_z",
"Full_sum_p", "Full_mut_e", "Full_mut_std", "Full_mut_z", "Full_mut_p",
"Full_AIC", "Full_BIC")
full_n10$NodeSz = 10
full_n10$Sim = 1:nsims

############# 10 NODES, 10% MISSING#################
UR1010 = mapply(unif.random, netlist, unlist(miss1, recursive = F), SIMPLIFY
= F)
UR_10_10 = mapply(ergms, UR1010)
uR_1010 = as.data.frame(t(UR1010))
colnames(UR_1010)=c("sum_e_A2_M1", "sum_std_A2_M1", "sum_z_A2_M1",
"sum_p_A2_M1", "mut_e_A2_M1", "mut_std_A2_M1", "mut_z_A2_M1", "mut_p_A2_M1",
"AIC_A2_M1", "BIC_A2_M1")

del1010 = mapply(deletion, netlist, unlist(miss1, recursive=F), SIMPLIFY = F)
DEL_10_10 = mapply(ergms, del1010)
del1010 = t(DEL_10_10)
delete_10_10 = as.data.frame(del1010)
colnames(delete_10_10)=c("sum_e_A1_M1", "sum_std_A1_M1", "sum_z_A1_M1",
"sum_p_A1_M1", "mut_e_A1_M1", "mut_std_A1_M1", "mut_z_A1_M1", "mut_p_A1_M1",
"AIC_A1_M1", "BIC_A1_M1")

mr1010 = mapply(multi.random, netlist, unlist(miss1, recursive = F), .1,
SIMPLIFY = F)
MR_10_10 = mapply(ergms, mr1010)
mr1010 = t(MR_10_10)
multi.ran_10_10 = as.data.frame(mr1010)
colnames(multi.ran_10_10)=c("sum_e_A3_M1", "sum_std_A3_M1", "sum_z_A3_M1",
"sum_p_A3_M1", "mut_e_A3_M1", "mut_std_A3_M1", "mut_z_A3_M1", "mut_p_A3_M1",
"AIC_A3_M1", "BIC_A3_M1")
```