LLNL-JRNL-728614-DRAFT

Social Network Analysis in R: A Software Review

B. Dabbs, S. Adhikari

April 5, 2017

Journal of Educational and Behavioral Statistics

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Social Network Analysis in R: A Software Review

Samrachana Adhikari[+], Beau Dabbs[*]
[+] Department of Health Care Policy, Harvard Medical School
[*]Lawrence Livermore National Laboratory

Abstract

1 Introduction

Relationships between entities in many complex natural and socio-economic systems can be represented by networks. Examples of such systems can be found in applications in biology (Davidson and Erwin, 2006), neuroscience (Henderson and Robinson, 2011), behavioral science (Haynie, 2001), education (Pitts and Spillane, 2009; Spillane et al., 2012), political science, and many other research fields and subfields (Fienberg, 2012). Networks are widely used in education research, for instance, to study how different units in educational settings interact amongst themselves or with other units, and to study the implications of these interactions on the overall learning behavior of the students (Pitts and Spillane, 2009; Penuel et al., 2009; Spillane et al., 2012; Spillane and Hopkins, 2013). In education research, teachers, students, administrators and more are examples of such units.

Analysis of network data spans a wide range of topics such as, understanding an underlying generative process of how and why a network is formed (Jacobs and Clauset, 2014); uncovering hidden communities in a network (Holland et al., 1983; Girvan and Newman, 2002); finding structures in network connections (Frank and Strauss, 1986; Robins et al., 2007); and understanding how the network connections influence behaviors of units in the network (Shalizi and Thomas, 2011). To tackle these aspects of network analysis, different types of network models have been developed by researchers across many disciplines (Goldenberg et al., 2010; Fienberg, 2012). Further, to facilitate implementation of these models various software packages have been developed.

In this paper, we review tools for analyzing social networks using packages available in a statistical programming language `R`. We focus primarily on a package **igraph** (Csardi and Nepusz, 2006) and a suite of packages known collectively as **statnet** (Handcock et al., 2008). These are the two most commonly used packages for analyzing network data in `R`, and provide a solid framework for understanding the means and methods of network analysis.

An overall goal of this paper is to both provide an introduction to network analysis for social science researchers, as well as to compare the two most popular packages for performing this analysis within R. To this end we will provide commentary on some of the useful conclusions that can be made based on the analysis of network data, and point towards methods of formalizing these conclusions. A code for installing software and performing all of the analyses covered in this paper is included in the supplementary material.

We begin in section 2 by introducing some preliminary notation for network data, as well as background on the two packages considered in this paper. Section 3 compares the capabilities of the two packages for visualizing and summarizing network data. We then show some examples of network modeling using the **ergm** and the **latentnet** packages within the **statnet** suite in section 4. Finally, we end the paper with a discussion in section 5 and a brief note on other resources available in **R** in section 6.

2 Background

2.1 Network Notation

Mathematically, a network can be represented as a graph $G = (\mathcal{V}, \mathcal{Y})$; defined in terms of a fixed set of vertices $\mathcal{V}$ containing n elements ($|\mathcal{V}| = n$), a set $\mathcal{Y}$ of edges between those vertices. For social network data, the vertices of the network typically represent people within the network, and the edges represent relationships, such as friendship, between those people.

The vertices, which are commonly referred to as nodes, can be represented using the names of the people they represent, but frequently are labeled using the integers 1 to n. Using this framework, the set of edges, $\mathcal{Y}$, consists of a set of pairs, (i, j), where i and j are both indices between 1 and n. Using the friendship example, $(i, j) \in \mathcal{Y}$ if person i and j are friends, and otherwise (i, j) is not in the edge set. One way to represent or store the information in a graph is as a node list of all of the vertices and a corresponding edge list containing the edges.

Another common way to represent network data is using an adjacency matrix. The adjacency matrix, Y, for a network G is the $n \times n$ matrix, where $Y_{ij} = 1$ if $(i, j) \in \mathcal{Y}$ and $Y_{ij} = 0$ otherwise. For many types of relationships it is typical to assume that the relationship is symmetric, i.e. Bob is friends with Alice if and only if Alice is friends with Bob. However for some relationships, such as advice seeking relationships, the connection may be one way, i.e. Bob asks Alice for advice, but Bob does not ask Alice for advice. If the relationship is assumed to be symmetric, we call the network *undirected*, otherwise we will call it *directed*.

In addition to the node and edge sets we will also frequently have covariate information about the nodes and/or the edges within a network. For example, we might have demographic information about the people in a social network or relative information such as the difference in age between each person in the network. We will represent this information using a set $\mathcal{X}$ of covariates that can be indexed by the node set, $i = 1, ..., n$, or the edge set of ordered pairs (i, j) with both i and j between 1 and n. For more complex network models we may also have a set of latent variables, $\mathcal{Z}$, that represent unobservable structure in the network. Typically these latent variables will be indexed by the nodes in the network. The models that use these latent structures estimate these latent variables, and then seek to learn something about the corresponding nodes in the network. In section 4.3 we will demonstrate how the **latentnet** package within the **statnet** suite uses positions in a latent Euclidean space to better understand a structure of a network.

2.2 Network Packages in R

The R programming language has been a staple for statistical analysts for decades, and many tools for network analysis are available. The two most commonly used tools in R are the **igraph** and **statnet** packages. These packages provide a diverse set of tools for a variety of analyses. However other tools are available for performing more specific tasks, and new tools are constantly under development. In section 6 we will discuss other packages currently available in **R**, as well as some software tools that appear to be missing currently. Here we give an overview of the two major network packages and the scope of their functionality.

The first package we consider, **igraph**, specializes in providing tools for exploratory network analysis. It contains functionalities for summarizing and visualizing networks, generating random graphs for simulations, and implementing graph algorithms in the context of large graphs (Csardi and Nepusz, 2006). In addition to providing many tools for analyzing graphs, **igraph** is able to load network data of many different formats, making it additionally useful as a tool for reading in networks of diverse storage forms.

The **statnet** suite is both an individual package, as well as the name for a collection of packages for performing various network analyses. The suite is developed and maintained by the **statet** development team based out of University of Washington. The base set of packages - **statnet**, **network**, and **ergm** - provide basic functionality for plotting networks and modeling networks within a Exponential Random Graph Modeling (ERGM) framework. We will cover this basic set of packages here, and also explore **sna** and **latentnet** packages that provide additional functionality. However we note that there are even more packages available in the **statnet** suite, and that new tools are still being produced today.

These required and optional components of the package **statnet** aim to provide a comprehensive framework for network modeling. The framework includes tools for descriptive techniques, for example network summarization and visualization, generative models for network model estimation and model evaluation, and permutation methods for model-based network simulation (see Handcock et al. (2008) for additional details). Our review focuses on tools for descriptive techniques and generative models.

2.3 Lazega Lawyers Dataset

As a working example in the paper, we used a network of lawyers from a network study of corporate law partnership that was carried out in a Northeastern US corporate law firm, referred to as SG&R, from 1988 to 1991 (Lazega, 2001). The dataset of lawyers includes (among others) measurements of different types of relationships among 71 attorneys (partners and associates) at the firm. Some examples of the relationships include, strong-coworker network, advice network, friendship network and indirect control networks. In addition to the edge related information, various lawyer specific attributes are a also part of the dataset. These attributes include information regarding seniority, formal status, the office location where they work, gender, the law school attended, their age and the number of years spent at the law firm.

In this paper, we focus on the friendship network of the lawyers within the law firm. A Binary tie in the friendship network is defined based on whether a lawyer socialized with other lawyers in the firm outside of the work setting. The friendship relationship in this context is considered to be directed, especially when the perception of a friendly relationship between the sender and the receiver of the friendship tie is different. So a lawyer who is nominated as a friend by another lawyer in the firm might not necessarily reciprocate the friendship tie and nominate the lawyer as their friend as well.

There are several questions we could ask about the friendship network of lawyers. For example, we may want to understand how the history of the lawyers, such as the number of years at the firm and the law school attended affect the occurrence of friendship ties in the network. We will begin this exploration by comparing visual summaries and by computing various summary statistics for the network. We discuss the exploratory and descriptive tools available in both **igraph** and **statnet** in section 3.

After performing some exploratory analyses we discuss many of the questions we would like to ask about these networks. Do lawyers cluster based on who their friends are in the firm? Are lawyers who went to the same school or who worked in the same firm more likely to nominate each other as friends? Is there an effect of seniority in defining who their friends are in the firm? Are the relationships different when considering the receiver specific attributes versus the sender specific attributes? An exploratory work can certainly suggest answers to these questions, but more explicit models of network behavior are required to find more definitive answers. In section 4, we review tools for modeling networks within the **statnet** suite, and obtain some answers to some of these basic questions.

3 Exploratory Analysis of Social Network Data

A common visual method to summarize a network is the sociogram plot (Moreno, 1935; Kolaczyk, 2009). Sociograms are used to visualize graphs by representing the nodes as points and the ties as lines connecting the points. The primary decision when making these plots is to determine where the nodes should be placed within the plot. Various methods have been developed for algorithmically placing the nodes such that tightly connected groups are closer together, and nodes with fewer connections are on the outer boundaries of the plot. Kolaczyk (2009) and Kolaczyk and Gabor (2014) have discussed popular techniques and conventions, along with software tools available for using the sociogram plot.

In addition to visual summaries, numerical summaries are also informative for exploring network data and capturing certain characteristics commonly observed in networks. Examples of such network characteristics include *centrality*, *reciprocity*, *transitivity*, *clustering*, etc, which can be described via common network statistics like *tie density*, *node degree*, *betweenness*, *closeness*, *triangles*, etc. We review some of the most commonly reviewed statistics here, but a more comprehensive list can be found in Kolaczyk (2009).

Some of the simplest summary statistics for network data involve the degrees of the nodes. For a directed network, we define the *out degree* for a node i to be the total number of edges from i to another node j. Mathematically we denote the out degree as $d_i^{out} = \sum_{j \neq i} Y_{ij}$. Similarly we define the *in degree* for a node i to be $d_i^{in} = \sum_{j \neq i} Y_{ji}$. We define the *degree* of a node to be $d_i = d_i^{in} + d_i^{out}$. If a network is undirected, then the in and out degrees are equal to one another, and we typically define the degree to simply be $d_i = d_i^{in} = d_i^{out}$.

The degree of a node is a common indicator of the overall connectedness of that node within the network, since higher degree nodes simply have more connections that other nodes. Network analysts are frequently interested in the average degree of the network, as well as the variability of the degree across nodes. It is also common to consider the overall *tie density* of the network, which is the ratio of total ties observed in the network to the total number of possible ties in the network. However you can also ask if most of the nodes in the network have a similar degree, or if instead there are a small number of central nodes with a large degree and a majority of nodes with relatively few connections.

Other statistics attempt to assess the *centrality* of particularly nodes in the network. One measure of node centrality is the *closeness* measure which measures the average number of steps it takes to get from node i to any node by taking the shortest path using only the observed edges in the network. Another measure of node centrality is the *betweenness* metric that determines how many shortest paths from node j to node k pass through node i. Thus nodes with high betweenness values are critical in guaranteeing information can flow quickly through the network, since removing them will cause the shortest paths between nodes in the network to increase.

There are also many global network statistics that are of interest. The tie density of a network is one example of a global network statistic, though many others are based on the overall connectivity of the network. Common measures include the diameter of the network, which is the longest shortest path in the network, and the network betweenness, which averages the betweenness of all of the nodes in the network.

The existence of and the search for 'communities' and analogous types of unspecified 'clusters' in a network can be addressed as a clustering problem (Kolaczyk, 2009). Notions of *clustering* in a graph are usually summarized by computing relative frequencies of edge connectivity. Transitivity of a graph is an example of such measure of relative frequency. It is computed as the ratio of the number of triangles in the network over the number of connected triples. The *connected triples* is represented by a subgraph of three vertices connected by two edges, also sometimes called a 2-star, whereas in a *triangle* all three vertices are connected by an edge.

3.1 Igraph

In this section we review the functionality of the **igraph** package for performing exploratory analysis on network data. The **igraph** package is primarily a tool for exploratory network analysis, and provides great flexibility for displaying network data. We will first review the methods for loading and examining network with the package, and then review the options for plotting networks and computing various summary statistics.

3.1.1 Loading and Accessing Network Data

Within the **igraph** package networks are stored as objects of type `igraph`. An object of type `igraph` can be thought of as a combination of a set of vertices, a set of edges between those vertices, and various attributes associated with both the vertices and the edges. The vertex attributes typically correspond to demographic information about the people in a social network, such as their age, gender, or location. Before any of the functions in the textbfigraph package can be used you must convert your network to an `igraph` object.

There are multiple methods for creating an `igraph` object. The `graph_from_edgelist` function allows you to convert an edgelist matrix to an `igraph` object and you can convert adjacency matrices to `igraph` objects using the function `graph_from_adjacency_matrix`.. If your network has weighed edges, you must use either the function *graph_from_data_frame()* or *graph_from_adjacency_matrix()* with the option *weighted* indicating the name for the weights.

However, one of the most useful features of igraph is the *read_graph* function which takes as arguments a filename and a format. The formats supported include gml, pajek, graphml and many others. With this function, igraph makes it easy to read in graphs that were originally generated from a multitude of sources. A companion function, *write_graph* allows similar flexibility in the storing of graphs. Thus it is possible to use igraph to translate networks from one file format to another using the corresponding read and write functions.

The *vertex_attr* function allows one to access the vertex attributes as a data.frame. You can also use the *vertex_attr* function to add or update the attributes for the vertices in the network. Edge attributes are also tracked within an `igraph` object, but these attributes are typically reserved for weighted networks, particularly since igraph will not allow attributes to be stored for pairs of nodes that are not connected in the network.

3.1.2 Visual Network Summaries

The igraph library really shines as a visualization tool for network data. We will focus on the basic plotting function, called by using `plot` on an igraph object, but note that there is also an rglplot and a tkplot method. The rglplot provides a representation of the network in three dimensional space, but currently doesn't actually take advantage of the third dimension of the plotting area. The tkplot method can be quite useful however, as it allows you to manipulate the position of the nodes of the graph by dragging and dropping with the mouse. We will focus on the various options for the layout and labeling of graphs using the default plot functionality.

As we mentioned at the beginning of this section, there are various algorithms that have been developed for automatically determining locations to place the nodes in a network. The igraph package provides access to many of these algorithms, though the default algorithm is the Fruchterman-Reingold method (Fruchterman and Reingold, 1991). The implementation of this method results in the exact location of the nodes varying a little bit each time the network is plotted, frequently resulting in some rotation of the overall layout of the nodes. However the igraph package provides functions which output the node locations for a given call to one of the algorithms which can then be stored to create multiple plots with the same layout.

Once we have a desirable set of positions for each of the nodes in the network, we can then begin to modify the appearance of the edges and the vertices in the network. All of the possible arguments to the plot function can be found in the help page for `igraph.plotting`. Things that can be modified include the shape, size, and color of the vertices and edges. You can even modify the exact size and width of the arrows in directed graphs. Figure 1 shows some examples of what can be done using these display options to examine the relationship between nodal covariates and the edges in the network.

3.1.3 Numerical Network Summaries

The **igraph** package also allows the user to calculate all of the summary statistics described in the beginning of this section, as well as many others not mentioned there. Table 1 provides a quick reference of functions for computing these statistics, and we discuss using these functions in the context of the Lazega lawyers dataset in section 3.3.

3.2 Statnet

The **statnet** suite of packages also offers functionality for displaying and summarizing network data. Most of the plotting functions can be found in the **network** package, which the **sna** package contains functions for computing network summary statistics. Overall the **statnet** suite can reproduce many of the capabilities of the **igraph** package, but tends to be less flexible in terms options for displaying networks and loading networks from different file types.

3.2.1 Loading and Accessing Network Data

In **statnet** a graph is represented by an object of class `network`. Similar to the `igraph` object, an object of class `network` is a collection of vertices, edges, and attributes which describe the network and the various covariate information about the network. To use the functions within the **statnet** package you must first convert your data to an object of type `network`.

An object of class network can be created using the function `network()`. The function takes inputs of types adjacency matrix, edgelist and incidence matrix. As a cautionary note, the **statnet** package assumes networks are directed by default, so you must specify **directed = FALSE** if you want an undirected network, even if the adjacency matrix is symmetric. It is also possible to read in networks stored in the *Pajek* format directly using the `read.paj` function, though many of the other file types supported by **igraph** are not readable using functions within **statnet**.

3.2.2 Visual Network Summaries

An object of class network can be visualized in **statnet** using the `plot` function. The **statnet** package has a few different functions for automatically determining node locations within a plot and uses the Fruchterman-Reingold method Fruchterman and Reingold (1991) by default. It is also possible to specify the coordinates of the nodes explicitly using the `coord` argument. It is also possible to manually manipulate the positions of the nodes using the mouse by passing the argument `interactive = TRUE`, however this interface is bulkier than the **igraph** interface.

Similar to plotting in **igraph**, most of the features of the plot can be modified, and we displayed some plots made using **statnet** in figure 2. We examine these plots in more detail and use them for exploratory analysis in section 3.3 shows some examples of plots generated using the **statnet** package.

Summary statistic	Function in **statnet**	Function in **igraph**
Node size	network.size()	vcount()
Edge count	network.edgecount()	ecount())
Tie density	network.density()	edge_density()
degree	degree()	degree()
In degree	degree(,cmode='indegree')	degree(,mode='in')
Out degree	degree(,cmode='outdegree')	degree(,mode='out')
Betweenness of a node	betweenness()	betweenness()
Edge betweenness		edge.betweenness()
Closeness of a node	closeness()	closeness()
Transitivity	gtrans()	transitivity()
Centrality measures	centralization()	centr_degree(), centr_betw(), ...

Table 1: Summary statistics and corresponding functions in **statnet** and **igraph**

3.2.3 Numerical Network Summaries

The package **sna** (Butts et al., 2008) in **statnet** provides a comprehensive list of functions to compute descriptive statistics at the graph or node level. Table 1 provides a list of some of functions for calculating some of the most common summary statistics for network data. In the next section we use some of these visual and numerical summaries to perform some preliminary analyses of the Lazega lawyers network.

3.3 Descriptive Statistics of the Lawyer's Network Using igraph and statnet

In this section we perform some exploratory analysis of the network of friendships in the Lazega lawyers dataset. We begin by examining sociogram plots of the networks. Using both **igraph** and **statnet** we fix an initial set of node locations using the force directed method described in Fruchterman and Reingold (1991). This algorithm causes groups of nodes that are more tightly connected to be closer together, with disconnected and low degree nodes being placed on the outskirts of the plot. These plots allow us to examine the relationship between the connections in the network and covariates associated with the edges. Figure 1 shows sociograms plotted using **igraph** and figure 2 shows sociograms plotted using **statnet**. We attempted to replicate the same style of plot using each package, so during the discussion we refer to each plot by order of appearance in the pair of figures.

The first plot we consider simply labels nodes based on the home office of each node. As you might expect, nodes in the Boston office are closer to one another in the sociogram, and the same is true of the lawyers in the Hartford offices. The second plot shows the law schools associated with each person at the firm, and shows that there is not a particularly clear structure, suggesting friendships may not be largely affected by the law school one attends.

The last two plots consider the number of years a lawyer has been with the firm, which larger nodes indicating more years at the firm. Most of the lawyers had been at the firm for less than 5

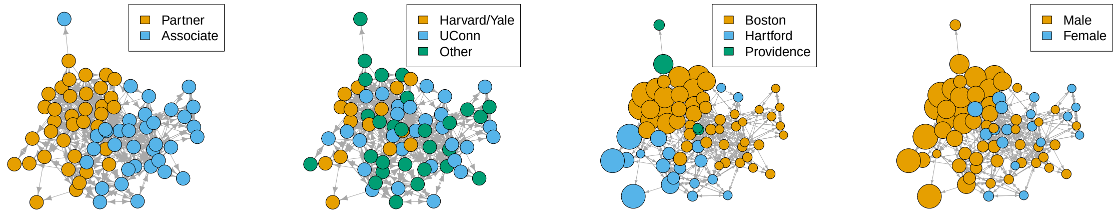

Figure 1: Socigrams created using **igraph** of the friendship network of Lazega laywers. In each plot the color indicates a particular categorical variable and in the last two plots the size of the nodes indicates represents the number of years at the law firm.

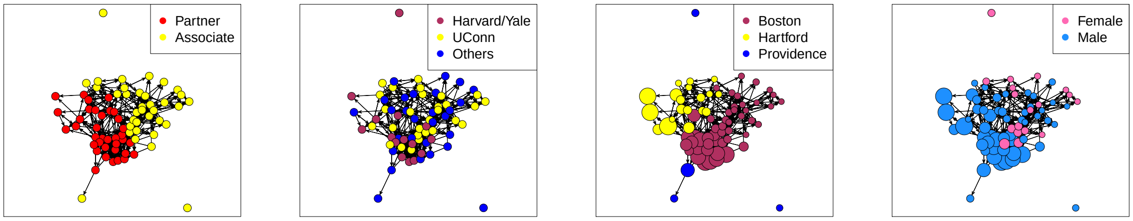

Figure 2: Socigrams created using **statnet** of the friendship network of Lazega laywers. In each plot the color indicates a particular categorical variable and in the last two plots the size of the nodes indicates represents the number of years at the law firm.

years, and the longest a lawyer had been with the firm was 32 years. The second to last plot shows the relationship between office and the number of years at the firm, showing that a tight cluster of long standing employees is present at the Boston office, where as the lawyers at the Hartford office are generally less connected.

The last plot shows that the number of years at the firm tends to be much smaller for the female lawyers at the firm. One result of this is that many of the female lawyers at the firm are not as well connected with the large block of male lawyers who have been at the firm for 25 or more years. In the next section we will use some statistical models to determine the significance of these effects, and attempt to model directly the relationship between these covariates and the propensity for lawyers to be friends within the firm.

We can also use numerical summary statistics to explore other properties of the network. The network level numerical summary statistics of the lawyer's friendship network are shown in Table 2, whereas the nodal level summary statistics are displayed as plots in Figure 3. The node level summaries include in-degree, out-degree, and betweenness.

From the numerical summaries, we see that the network is fairly small with 71 lawyers and somewhat sparse (edge density $\approx$ 12%). The visual summary of the nodal in-degrees and out-

Network size	71
Edge count	575
Network edge density	0.116
Transitivity	0.45

Table 2: Numerical network level summaries of the lawyers' friendship network data.

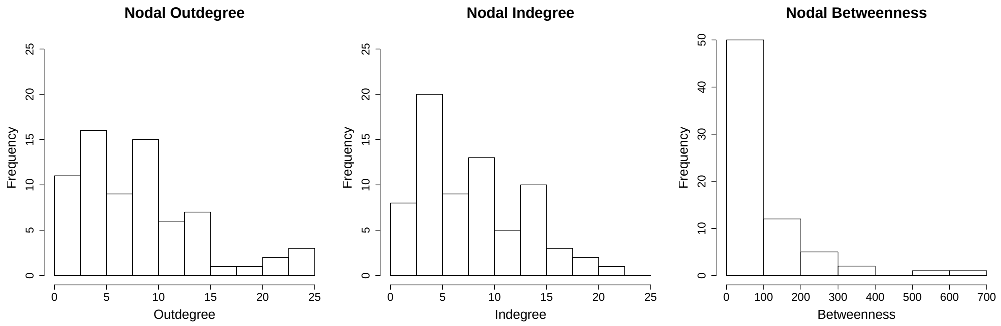

Figure 3: Node level summaries of the lawyers' friendship networks.

degrees shows that majority of the lawyers have small in and out-degrees with a median of about 5 in (out) ties. Finally, while most of the nodes have less than 100 betweeness measure, very few have relatively high betweenness indicating that these nodes with high betweenness measure have major role in connecting other nodes in the network.

3.4 Comparisons of these packages on summarizing and visualizing social network data

The plotting functions in packages **igraph** and **network** are both able to differentiate the attributes of nodes and edges by specifying size, color, etc. We found the supplementary document for **network** more straightforward compared to that of **igraph**. The **igraph** package has some extra features, such as the `tkplot` function, which allow the user to quickly explore the structure of a network in `R`. However, the **statnet** package feels more like a typical `R` package, making its use a bit easier for novice to intermediate users of `R`. Both of these packages provide all the tools necessary to perform exploratory analyses, and generate attractive plots displaying the connections and basic structure of a network.

4 Statistical Modeling of Social Network Data

4.1 Background: Statistical Models for Social Network Analysis

While descriptive statistics are useful ways to summarize features of an observed network, using these statistics for comparing networks, for making inference and for predicting future network is problematic. The concern arises because the descriptive statistics are inherently aggregate and thus do not give us any sense of variability or a notion of noise. We can overcome some of these issues by instead using a statistical model for network analysis.

Statistical modeling of network data is interesting and challenging because the "usual" statistical assumption of independence of observed units does not apply[1]. To that end, most of the models developed for statistical analysis of social network data aim to define probability distributions on graphs by accounting for their complex dependencies (Goldenberg et al., 2010; Fienberg, 2012). These models can be classified into different classes based on the assumptions about the dependence of ties in a network.

Existing statistical models for analyzing a single network can be broadly divided into two classes; Exponential random graph models and Conditionally independent dyad models. Goldenberg et al. (2010), Fienberg (2012) and Dabbs et al. (in prep.) provide a detailed review of statistical models for analyzing a network.

For simplicity, consider a discrete and undirected network Y such that

$$Y_{ij} = \begin{cases} \text{1 if there is a tie between i and j} \\ \text{0 if there is no tie} \end{cases}.$$

The exponential random graph models (ERGMs) (or alternatively also known as the $p*$ models) are defined by descriptive network statistics, such as in-degree, out-degree, number of triangles, etc.(Frank and Strauss, 1986; Wasserman and Pattinson, 1996). These network statistics are sufficient statistics of a probability distribution of a graph in ERGMs, which can be written as an exponential family distribution (Holland and Leinhardt, 1981; Wasserman and Pattinson, 1996; Robins et al., 2007; Goldenberg et al., 2010).

A general ERGM can be expressed as

$$P(Y|\theta) \quad = \quad \frac{1}{C(\theta)} \exp(\sum_{k=1}^{K} \theta_k^T s_k(Y))$$

where, $\{s_1(Y), \ldots, s_K(Y)\}$ is a set of sufficient statistics; $\{\theta_1, \ldots, \theta_K\}$ is the set of corresponding parameters that relate network statistics to a network distribution and need to be estimated. Finally, $C(\theta)$ is a normalizing constant.

Models based on ERGM representation aim to investigate certain known or assumed network properties, like centrality, reciprocity, and transitivity in the observed network. The assumed dependence structure among ties in a network influences the network statistics included in the

[1] The units of observation are ties or dyads in the case of network modeling.

ERGMs. The class of ERGMs includes an edge independence model, a dyadic independence model, Markov random graphs, and many other models with complex dependence structure that can be written in ERGM representation (Robins et al., 2007).

As the assumption on the tie dependence deviates from independence, the model gets more and more complicated. Many of these methods run into computational complications and convergence issues, for example convergence to degenerate solutions, when the model specification is complex, thus making them less desirable in real world applications (O'Malley, 2013; Handcock et al., 2003).

Alternatively, the conditionally independent dyad (CID) models are developed with an assumption that dyads in a network can be modeled as independent random variables, conditional on latent variables, observed nodal covariates, and the underlying network structures (Goldenberg et al., 2010; Jacobs and Clauset, 2014; Adhikari et al., 2015; Dabbs et al., in prep.). Some of the CID models make an additional assumption that the ties in a network are conditionally independent given covariates and latent variables, also called Conditionally Independent Tie (CIT) models.

Under the assumption of the CID models, conditional on the latent variables on nodes i and j and the observed covariate X_{ij}, the ties Y_{ij} are assumed to form independently with probabilities p_{ij} such that

$$p_{ij} = W(X_{ij}, f(z_i, z_j)). \tag{1}$$

Here, W is a link function that relates p_{ij} with X_{ij} and the node specific latent variables, z_i and z_j. If we assume Bernoulli distribution with probability p_{ij} on ties Y_{ij}, then W would be an inverse-logit function; $z_i \sim g$ are independent node-specific latent variables with density g, which are mapped to p_{ij} by a function f.

For example, a latent space network model (LSM) is a CID model such that the nodes in a network are assumed to lie in a low dimensional latent space and the probability of a tie p_{ij} is inversely related to the distance between the nodes in the latent space (Hoff et al., 2002). Then, conditional on the latent positions, the ties in a network are assumed to be independent. The LSM is written in notation as

$$\begin{aligned} Y_{ij} &\overset{i.i.d}{\sim} \text{Bernoulli}(p_{ij}) \\ \text{logit}(p_{ij}) &\sim \beta_0 - |Z_i - Z_j| \\ Z_i &\sim \text{Multivariate Normal}(0, \Sigma), \end{aligned} \tag{2}$$

where β_0 is an intercept term and Z_i represents the latent space position of a node i in a d-dimensional space with $|Z_i - Z_j|$ representing the latent Euclidean distance between node i and j. Edge level covariates can also be included in the LSM by adding them in the link function in Equation 2 (similar to regression).

Packages **ergm** and **latentnet** in the **Statnet** suite provide software support for implementing network analysis using the generative models discussed in this sectison.

4.2 ERGMs in Statnet

Exponential random graph models (ERGMs) are implemented in **R** using the package `ergm` in the `statnet` suite. Within the `ergm` library, we fit the ERGMs by specifying a *formula* similar to the one we would use when performing linear regression using the `lm` function in `R`. For example, if we wanted to fit an ERGM where the sufficient statistics are the number of edges in the network and the number of triangles in the network, we would call

```
ergm(Y ~ edges + triangle).
```

We covered some network statistics in section 3 while discussing numerical network summaries, but there are seemingly limitless different statistics that can be used when fitting ERGMs. In order to use a given sufficient statistic for the ERGM we must first know the specific keyword for that statistic. A comprehensive list of these keywords can be found in the help file for `ergm-terms`. The list is quite comprehensive and it's difficult to tell which statistics are more or less useful. However, if there is a statistic that a researcher wishes to use, it is quite easy to search for the correct term to estimate the corresponding ERGM.

4.2.1 ERGMs with Covariates

It is also possible to use the ERGM framework to model formation of edges as a result of both node and edge covariates. For instance, when estimating a model with the `edges` statistic and a single nodal covariate, for example `gender` in the Lazega Lawyers dataset, the result is the same as fitting a logistic regression as

$$logit(\mathbb{P}(Y_{ij} = 1|X, \beta_0, \beta_1) = \beta_0 + \beta_1(X_i + X_j). \tag{3}$$

Using the regression as shown in equation 3 we can examine the significance of the effects of the covariates in the Lazega Lawyers dataset, that were examined with exploratory methods in section 3.

As a first example, we examine the model using the covariates for the gender of an individual, the age and the number of years at the firm, as well as dummy variables indicating the law school of the individuals and the partnership status. Table 3 shows the results of fitting a regression using these covariates and the `ergm` function.

	Estimate	Std. Error	p-value
edges	-1.5874	0.3758	0.0000
nodecov.gender	0.0153	0.0845	0.8564
nodecov.harvard	0.1513	0.0985	0.1244
nodecov.yale	0.3709	0.0734	0.0000
nodecov.partner	0.5158	0.0978	0.0000
nodecov.years	0.0035	0.0068	0.6073
nodecov.age	-0.0174	0.0056	0.0017

Table 3: ERGM regression on nodal covariates.

Since we can think of this type of model as a logistic regression, the coefficients for the dummy variables represent an increase in the log odds of an edge occurring when that condition is met. Thus from the results displayed in table 3, we can determine that lawyers who are partners in the firm have more friends in the firm, and that lawyers who went to Yale have a significant increase in the log odds of having an edge and thus are more connected within the network in general. The other significant effect is of age, with increasing age suggesting a decrease in the overall number of ties, though this could be confounded somewhat with the slight positive effect of additional years working at the firm.

More generally, when we are presented with a set of nodal covariates, $X_1, ..., X_n$ for a network with n nodes we can use a logistic regression of the form

$$logit(\mathbb{P}(Y_{ij} = 1|X, \beta_0, \beta_1) = \beta_0 + \beta_1 f(X_i, X_j), \tag{4}$$

for any function f. The default function for the **ergm** package is the summing function, $f(X_i, X_j) = X_i + X_j$, which results in the model in equation 3. However, sometimes using a function that compares nodal covariates X_i and X_j makes more sense in a model. For instance, we might be primarily interested in whether or not attending the same law school increases the odds of a tie, or whether difference in age is a significant predictor of a tie probability. Using this idea, we fit a new model with the same dummy variables as before, but instead consider the absolute difference in age and years at the firm, thus using the function $f(X_i, X_j) = |X_i - X_j|$. We can accomplish this using the `edgecov` function within the `ergm` formula[2].

	Estimate	Std. Error	p-value
edges	-1.9322	0.1408	0.0000
nodecov.gender	-0.0711	0.0830	0.3919
nodecov.harvard	0.2384	0.0945	0.0117
nodecov.yale	0.4057	0.0756	0.0000
nodecov.partner	0.5749	0.0708	0.0000
edgecov.age	-0.0332	0.0075	0.0000
edgecov.years	-0.0971	0.0089	0.0000

Table 4: ERGM regression coefficients for model with difference in age and years at the firm as covariates.

Table 4 shows the results of the slightly more complicated logistic regression that uses the comparison function to compute the edge covariates. We now see that the difference in the number of years at the firm actually has a stronger negative effect on the probability of a tie than the difference in age.

Finally, we considered using the equality function with the gender covariate to determine if people of the same gender were more likely to be friends within the law firm. Table 5 shows the results of this final model. Unlike the previous two models, we see that gender does have a significant effect on the formation of ties, just not as a simple increase or decrease in degree based on the gender of an individual.

[2] You must first calculate the matrix M where entry m_{ij} is $f(X_i, X_j)$.

	Estimate	Std. Error	p-value
edges	-2.1919	0.1415	0.0000
edgecov.same.gender	0.3847	0.1041	0.0002
nodecov.harvard	0.2444	0.0942	0.0095
nodecov.yale	0.4151	0.0750	0.0000
nodecov.partner	0.5307	0.0672	0.0000
edgecov.age	-0.0340	0.0076	0.0000
edgecov.years	-0.0950	0.0089	0.0000

Table 5: ERGM regression coefficients for model with difference in age and years as well as an indicator of the gender of the sender and receiver of an edge being the same.

4.3 Latent Space Models in Statnet (Latentnet)

The package **latentnet** within the **statnet** suite implements the latent space model (LSM) of Hoff et al. (2002) in **R**. The package, however, has functionalities to implement a more general latent cluster model (LCM) of Handcock et al. (2007), which allows for mixture priors on the latent space positions. Here we will focus on using the **latentnet** to fit the LSM, which is a special case of the latent cluster model with cluster size equal to 1.

The LSM is fitted using a function `ergmm`, in the package **latentnet**. The first, and the only mandatory, argument to `ergmm` is a *formula*. We use the formula to specify a model that needs to be fitted along with a network object that is to be used as a response variable in the model.

A formula for the function `ergmm` in **latentnet** is specified using three different components; an intercept, a model specifications and dyadic level covariates. A model that is to be fitted is specified using a component `latent`. The component `latent` with arguments d for the dimension of the latent space is a required argument, whereas the argument G to specify the number of clusters is optional. The default value for G is zero, which corresponds to the LSM model of Hoff et al. (2002). Additional arguments can be provided to specify prior mean and variance components for the latent space positions. Finally, the component `latentcov` is used to add dyadic (edge level) covariate terms. Note, however, that the function `ergmm` does not admit terms that introduce conditional dyad-dependence. This is an important and rather special feature of the **latentnet** which allows it to fit models of class CID only and distinguishes **latentnet** model fits from the models fitted in the package **ergm**.

The function `ergmm` implements an MCMC sampling algorithm to get samples from the posterior distribution of the parameters in the model given data by finding reasonable initial values, running the MCMC sampler and performing a post-processing of the MCMC sample.The default options for the starting values and the tuning parameters in the function `ergmm` make running an MCMC to fit the model very convenient, especially for a user without any prior experience in running an MCMC algorithm or exploring diagnostics on the fitted MCMC chains. However, for someone with advanced experience in fitting MCMC, the chains can be manually controlled by specifying initialization, tuning parameters, burnin etc.

Model	Formula
I. LSM	`friend.net ~ euclidean(d = 2)`

II. LSM + One covariate	`friend.net ~ euclidean(d = 2)`
	`+ edgecov(same.status,"partner")`

III. LSM + Many covariates	`friend.net ~ euclidean(d = 2)`
	`+ edgecov(gender.same, "same.gender")`
	`+ edgecov(same.status, "partner")`
	`+ edgecov(age.diff, "age")`
	`+ edgecov(years.diff, "years")`
	`+ edgecov(school.same, "same.school")`

Table 6: Variations of the latent space models fitted on the lawyer's data using `ergmm` in the package **latentnet** and the corresponding formula to fit the models in the lawyers' friendship network.

4.3.1 LSM with Covarites fits on Lawyers' Network Data

To add dyad or edge level covariates in the LSM, observed nodal attributes need to be converted into edge attributes. The conversion usually depends on the goal of an analysis and needs to be specified by a researcher. These edge covariates could represent features like similarity in nodal characteristics, differences in nodal characteristics and sender or receiver specific characteristics. We discussed some of these covariates in subsection 4.2 while fitting the ERGMs with covariates on the lawyer's friendship network.

We fitted the LSM with 2-dimensional latent space on the Lawyer's network using the function `ergmm` in **latentnet** and investigated the posterior means of the latent positions. Next, we fitted the LSM along with the edge level covariate indicating whether two lawyers i and j share the same status in the firm. Finally, we fit an LSM with many edge level covariates. The formula used to fit the models along with the covariates are displayed in Table 6. (Codes that compute these edge covariates from the node level covariates are included in the supplementary document.)

The posterior mode of the latent positions from the fitted models are investigated in Figure 4. The nodes in the plots are colored based on whether a lawyer is a partner in the firm. We observe clustering of the lawyers' fitted latent positions based on their status in the firm when we fit the LSM without any covariates. However, when we include a dyadic covariate indicating whether two lawyers have a similar status in the model, the clustering of the latent positions based on the status is less clear. This observation is also made in the model with many edge covariates.

Summary of the `ergmm` fit can be called using the function `summary` on the object of class ergmm. The function returns the formula fitted along with a posterior mean of the intercepts and the coefficients (in model with covariates) and the 95% credible interval for the posterior estimates. The coefficient can be thought of as a unit increase in the log odds-ratio of forming an edge for a unit increase in the covariate, conditional on the latent structures. Further, it is deemed as 'significant' if the corresponding 95% credible interval does not include zero.

The summary of the corresponding covariate coefficients on the log-odds of forming a tie for the two models with covariates (and latent positions) are reported in Table 7. From the summary

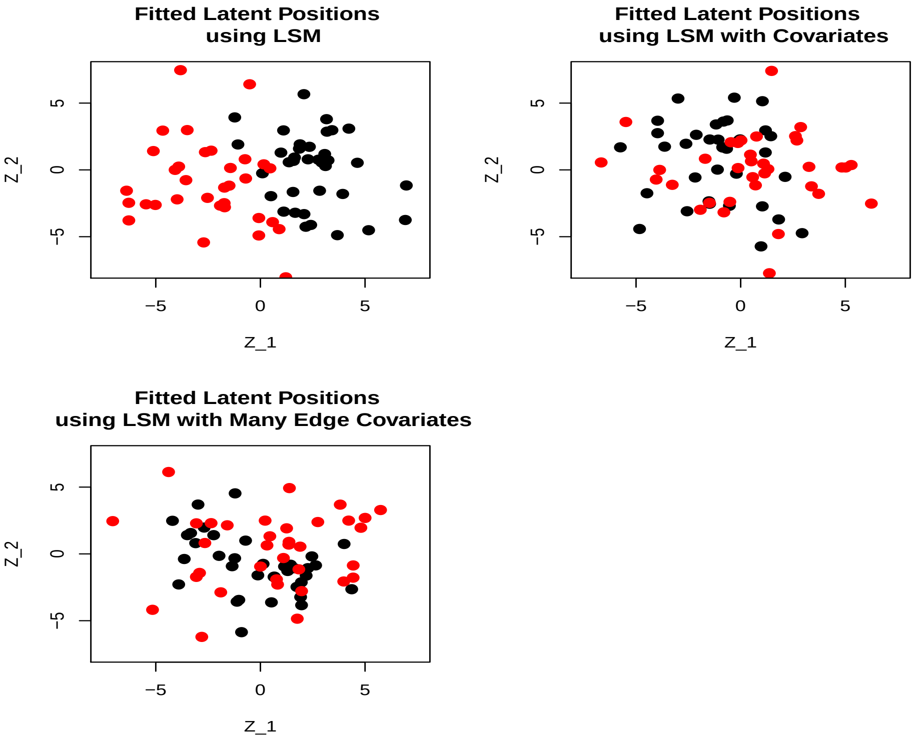

Figure 4: Posterior mode of the latent positions in the lawyer's network colored by the status of the lawyers in the firm for three different models.

in Table 7, we observe that the lawyers with the same status in the firm are more likely to consider each other as friends. Similar effect (with less magnitude) is also seen for two lawyers of same gender and who went to same law school. Further, the tie is less likely between lawyers of different age or who have spent different number of years in the firm.

5 Discussion

In this paper, we reviewed two commonly used R packages, **igraph** and **statnet**, in the context of social network analysis. More specifically, on one hand, we focussed on plotting and summarizing features of the package **igraph**. On the other hand, we reviewed tools available for both network summary and network modeling for the package suite **statnet**. Finally, we demonstrated the functionalities of these packages to answer relevant questions in the context of social network analysis using Lazega Lawyers' friendship netwok as an example.

We note that **igraph** has many attractive features for network plotting and summarizing compared to **statnet**. Of the many attractive features, **igraph**'s ability to deal with different data types and sizes efficiently is particularly notable, which we found lacking in **statnet**. However, the strength of **statnet** lies in its ability to tackle network analysis from both descriptive and modeling

Model II.	Covariate	Posterior Mean	95% Credible Interval
	edgecov.partner	1.66	(1.36,1.96)

Model III.	Covariate	Posterior Mean	95% Credible Interval
	edgecov.same.gender	0.52	(0.24, 0.80)
	edgecov.partner	1.23	(0.92, 1.55)
	edgecov.age	-0.04	(-0.06, -0.01)
	edgecov.years	-0.07	(-0.09, -0.045)
	edgecov.same.school	0.37	(0.13, 0.64)

Table 7: Difference in the log odds-ratio of forming a friendship tie in Lazega Lawyers' network conditional on the edge covariates.

perspectives by providing a complete set of tools at a researcher's disposal. The manuals for both of these packages are also very well maintained and continuously updated, which provide a great assistance for anyone who wishes to use these packages.

In this paper, we focused on analyzing unipartite network graphs, with ties between nodes of a similar class, which are observed at a fixed time point. However, many applied questions for social network analysis might focus on networks observed over many time points, on an ensemble of networks observed at a time point, and on relationships that extend beyond unipartite graphs. We have largely ignored software tools for analyzing these types of network data in this review and briefly mention some of the resources available in R for such analyses in Section 6.

6 Other Resources

There are packages available in R for social network analysis which are not discussed in the paper. For example, R package **ggnetwork** (Briatte, 2016) is a recently developed package for network visualization. The package **ggnetwork** provides a way to build network plots using a popular plotting R package **ggplot2**.

Another example is the R package **CIDnetworks** by Adhikari et al. (2015) that focuses on using generative network models, that fall under the class of conditionally independent dyads (CID) models, for social network analysis. The package combines different CID models, latent space model and covariates for example, additively for efficient computation and can handle ordinal and continuous network ties. Other packages extend network analyses to multiple network, such as the **HLSM** R package by Adhikari et al. (2016) which focuses on modeling network ensembles using the hierarchical latent space model introduced in Sweet et al. (2013).

Finally, **Rsiena** (Ripley et al., 2013) is another widely used R package for stochastic actor-oriented network models that is introduced in Snijders (1996). The package can be used to analyze cross-sectional network data, longitudinal network data, longitudinal data of networks and behavior, using the principles of stochastic actor-oriented network models. We note that the assumptions in the models considered in **Rsiena** are usually different from the ones considered in **statnet** and encourage readers to refer to the package's webpage for necessary references.

References

Adhikari, S., Dabbs, B., Junker, B., Sadinle, M., Sweet, T., and Thomas, A. (2015), *CIDnetworks: Generative Models for Complex Networks with Conditionally Independent Dyadic Structure*, r package version 0.8.1.

Adhikari, S., Junker, B., Sweet, T., Thomas, A. C., Adhikari, M. S., and ByteCompile, T. (2016), "Package ?HLSM?" .

Briatte, F. (2016), *ggnetwork: Geometries to Plot Networks with 'ggplot2'*, r package version 0.5.1.

Butts, C. T. et al. (2008), "Social network analysis with sna," *Journal of Statistical Software*, 24, 1–51.

Csardi, G. and Nepusz, T. (2006), "The Igraph Software Package For Complex Network Research," *InterJournal*, Complex Systems, 1695.

Dabbs, B., Adhikari, S., Sadinle, M., Junker, B. W., Sweet, T. M., and Thomas, A. C. (in prep.), "Conditionally Independent Network Models," .

Davidson, E. H. and Erwin, D. H. (2006), "Gene regulatory networks and the evolution of animal body plans," *Science*, 311, 796–800.

Fienberg, S. E. (2012), "A Brief History of Statistical Models for Network Analysis and Open Challenges," *Journal of Computational and Graphical Statistics*, 21, 825–839.

Frank, O. and Strauss, D. (1986), "Markov Graphs," *Journal of the American Statistical Association*, 81, 832–842.

Fruchterman, T. M. J. and Reingold, E. M. (1991), "Graph Drawing by Force-directed Placement," *Software- Practice and Experience*, 21, 1129–1164.

Girvan, M. and Newman, M. (2002), "Community structure in social and bilogical networks," *Proceedings of the National Academy of Sciences*, 99, 7821–7826.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010), "A Survey of Statistical Network Models," *Found. Trends Mach. Learn.*, 2, 129–233.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., and Morris, M. (2008), "statnet: Software tools for the representation, visualization, analysis and simulation of network data," *Journal of statistical software*, 24, 1548.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007), "Model-based clustering for social networks," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170, 301–354.

Handcock, M. S., Robins, G., Snijders, T. A., Moody, J., and Besag, J. (2003), "Assessing degeneracy in statistical models of social networks," Tech. rep., Citeseer.

Haynie, D. L. (2001), “Delinquent peers revisited: Does network structure matter?” *American journal of sociology*, 106, 1013–1057.

Henderson, J. A. and Robinson, P. A. (2011), “Geometric Effects on Complex Network Structure in the Cortex,” *Physical Review Letters*, 107, 018102.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), “Latent Space Approaches to Social Network Analysis,” *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, 97, 1090–1098.

Holland, P. W., Blackmond, K., and Leinhardt, S. (1983), “Stochastic Blockmodels: First steps,” *Social Networks*, 5, 109–137.

Holland, P. W. and Leinhardt, S. (1981), “An Exponential Family of Probability Distributions for Directed Graphs,” *Journal of the American Statistical Association*, 76, pp. 33–50.

Jacobs, A. Z. and Clauset, A. (2014), “A unified view of generative models for networks: models, methods, opportunities, and challenges,” *arXiv preprint arXiv:1411.4070*.

Kolaczyk, E. D. (2009), *Statistical Analysis of Network Data: Methods and Models*, Springer Publishing Company, Incorporated, 1st ed.

Kolaczyk, E. D. and Gabor, C. (2014), *Statistical Analysis of Network Data with R*, Springer New York.

Lazega, E. (2001), *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*, Oxford University Press on Demand.

Moreno, J. L. (1935), “Who Shall Survive? A New Approach to the Problem of Human Interrelations.” *The Journal of Social Psychology*, 6, 388–393.

O’Malley, J. A. (2013), “The analysis of social network data: as exciting frontier for statisticians,” *Statistics in Medicine*, 32, 539–555.

Penuel, W. R., Riel, M., Krause, A. E., and Frank, K. A. (2009), “Analyzing Teachers’ Professional Interactions in a School as Social Capital: A Social Network Approach,” *The Teachers College Record*, 111, 124–163.

Pitts, V. M. and Spillane, J. P. (2009), “Using social network methods to study school leadership,” *International Journal of Research & Method in Education*, 32, 185–207.

Ripley, R., Boitmanis, K., and Snijders, T. A. (2013), *RSiena: Siena - Simulation Investigation for Empirical Network Analysis*, r package version 1.1-232.

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. (2007), “An introduction to exponential random graph (p*) models for social networks,” *Social Networks*, 29, 173–191.

Shalizi, C. R. and Thomas, A. C. (2011), "Homophily and contagion are generically confounded in observational social network studies," *Sociological methods & research*, 40, 211–239.

Snijders, T. A. (1996), "Stochastic actor-oriented models for network change," *Journal of mathematical sociology*, 21, 149–172.

Spillane, J. P. and Hopkins, M. (2013), "Organizing for instruction in education systems and school organizations: How the subject matters," *Journal of Curriculum Studies*, 45, 721–747.

Spillane, J. P., Kim, C. M., and Frank, K. A. (2012), "Instructional advice and information providing and receiving behavior in elementary schools exploring tie formation as a building block in social capital development," *American Educational Research Journal*.

Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013), "Hierarchical Network Models for Education Research: Hierrarchical Latent Space Models," *Journal of Educational and Behavioral Statistics*, 38, 295–318.

Wasserman, S. and Pattinson, P. (1996), "Logit Models and Logistic Regressions for Social Networks: I. An Introduction to Markov Graphs and $p*$," *Psychometrika*, 61, 401–425.