# A Simulation Study in using Random Graph Models to fit Social Networks

---

A Comprehensive Evaluation Report
Presented to
The Statistics Faculty
Amherst College

---

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts
in
Statistics

---

Levi Lee

April 03, 2017

# Acknowledgements

This project would not have been possible without the support of so many people. First and foremost, I would like to thank everyone currently from the Mathematics and Statistics departments along with some previous faculty as well. In particular, Professor Xiaofei (Susan) Wang and Professor Eunice Kim were some of the first people that really got me heavily involved with R and R-Studio through the course work they provided. I would never imagine that one day, I would be doing something advanced with such software. I would also like to my fellow stats majors for always providing such amazing help when I get stuck. My gratitude goes to Melody Owen '17, Jordan Browning '17, and Muling Si '17 for being such wonderful partners in Advanced Data Analysis and for always bearing with me. After seeing how wonderfully they all code really inspired me to improve my own skills as well. Many thanks to Stephany Flores-Ramos '17 and Azka Javaid '17 as well for the constant support and company that they offered. They taught me that homework is often best handled with discussion and collaboration. Special thanks to Ningyue (Christina) Wang '16 for all the senior advice when I was a junior. To all the students and friends who knew I was working on this project, thank you for the constant moral support. I wish I can name all of you; it's the little things that I appreciate the most. To those who also had a hand in helping with this project, even in the slightest, you are truly a cut above. Special thanks to Nguyen (Johnson) Tran, a childhood friend from Florida who was kind enough to gift me a wonderful textbook on statistics, which proved to be very useful in my research. I want to dedicate this next paragraph to my family. Without their love and support, I would not be at Amherst today. Since the day I started school, my parents have done everything in their power to make sure the only thing I ever really need to focus on was my education. Nothing could ever amount to the sacrifices they have made. Being able to focus solely on academics, I now realize, was a privilage that not everyone had. My penultimate thanks goes to Professor Nicholas Horton, whom I have had the pleasure of having a professor for two courses in a row. He is a professor, mentor, and coach who really has a sense of the direction statistics is taking, and is really preparing us all to "think with data." Finally, I want to give my final thanks to Professor Amy Wagaman, who has been my "three-times advisor" for Mathematics, Statistics, and the Mathematics Honor Thesis. She has been an amazing advior in all regards, and I cannot thank her enough for her patience, understanding, and dedication to my learning. Working on such a project has really been challenging, but was definitely the highlight of my final year in college. It is thanks to her that I was able to explore the facinating world of networks and pursue graduate study in this field. I hope that this project has made her proud.

# Table of Contents

**Introduction** . . . . . **1**
0.1 Basic Terminology and Definitions . . . . . 1
0.2 Example 1: Zachary's Karate Club . . . . . 3
0.3 Example 2: Lazega's Law Firm . . . . . 3
0.4 Modeling Real-World Networks . . . . . 4

**Chapter 1: Descriptive Network Statistics** . . . . . **7**
1.1 Transitivity . . . . . 7
1.2 Average Path Length . . . . . 8
1.3 Diameter . . . . . 8
1.4 Centrality . . . . . 8

**Chapter 2: Graph Models** . . . . . **11**
2.1 Erdős-Rényi . . . . . 11
2.2 Watts-Strogatz . . . . . 13
2.3 Exponential Random Graph Models (ERGMs) . . . . . 14
2.3.1 Deriving Erdős-Rényi-Gilbert from ERGMs . . . . . 15

**Chapter 3: Simulation Study** . . . . . **17**
3.1 Description of Our Dataset . . . . . 17
3.2 Generating Random Graphs . . . . . 19
3.2.1 Using the Erdős-Rényi Model . . . . . 19
3.2.2 Using the Watts-Strogatz Model . . . . . 20
3.2.3 Using ERGMs . . . . . 22

**Conclusion** . . . . . **25**

**Appendix A: R/R-Markdown Code** . . . . . **27**

**References** . . . . . **35**

# List of Tables

1.1 Average centrality measures for the karate and lawyer networks . . . 9

3.1 Comparisons of Network Statistics in among graphs simulated from classical graph models and the observed network . . . . . 23

# List of Figures

1 Zachary's Karate Club ($N_V = 34$, $N_E = 78$) . . . . . 4
2 Lazega's Law Firm ($N_V = 36$, $N_E = 115$) . . . . . 5

2.1 Examples of graphs generated from the Erdős-Rényi model. Left and Right: $N_V = 10$, $p = 0.25$. . . . . 12
2.2 Examples of graphs generated from the Watts-Strogatz model. Both: $N_V = 10\$$, $r = 2$. Left: $p = 0$. Right: $p = 0.25$. . . . . 14

3.1 Overview of our component of the Facebook network . . . . . 18

# Abstract

Networks see usage in across many disciplines. We will focus on social networks, a category of which that tend to exhibit some characteristics unique to themselves. Is it possible to model social networks despite this uniqueness? To answer this, we begin with an overview of networks, network statistics, and graph models. We decide to look at Erdős-Rényi and Watts-Strogatz models as well as exponential random graph models (ERGMs). We conduct a simulation study where we apply various graph models onto an observed network; here, we choose to analyze a component of the Facebook network. We compare the network statistics of this Facebook network with those calculated from graphs of parable magnitude simulated from their respective graph models. We conclude that models such as the Erdős-Rényi and Watts-Strogatz models do not seem to accurately fit the Facebook network for these network statistics. Work on applying ERGMs and fitting them to the Facebook network is still a current work in progress.

# Introduction

Generally speaking, a *network* is a collection of inter-related objects. This broad definition allows networks to be used in a variety of disciplines. Here, we focus on *social networks*, in which these objects are either individuals or groups (though they need not be human), called *actors* and the relationships among them are called *ties*. These ties can be friendships, collaborations, email exchanges, and so on. Social networks have been a growing field of study since the 1930s. Even before the computer age, people have observed unique relationships among individuals (Kolaczyk 2009 and Newman 2010). One famous example is Milgram's letter experiments. Travers & Milgram (1967) considered 296 individuals in Nebraska and Massachusetts who were instructed to construct chains of correspondence via letter until the correspondence reached a target person in Boston. Out of the 64 letter chains that succeeded, they discovered that, overall, these chains did not require that many people. Surprisingly, the median number of people the letters had to go through was approximately six, giving rise to the popular term "six degrees of separation." This phenomenon is not true for all networks; in fact, many social networks exhibit characteristics not seen in other types of networks. Would it be possible then, for us to model social networks depite phenomena such as the "six degrees of seperation?" Models for networks certainly exisit, but how accurate are they? It turns out that through a simulation study, we can hope to find an answer (or at least be pointed to the right direction). However, before heading straight to a simulation study, we must be careful in our approach. Exactly how do we mathematically define a network? What does a model for a network consist of? How would we measure accuracy for such models? We begin with some mathematical background before describing the components of the simulation study and we will see many examples of social networks in subsequent sections.

## 0.1 Basic Terminology and Definitions[1]

More formally, a network is collection of vertices and edges. We borrow much of the mathematical terminology for networks and network statistics from graph theory. A network is a *graph*, which is an object denoted $G = (V, E)$ such that $E$ is the set of edges and $V$ is the set of vertices. Each edge connects two vertices, and so for any two

[1]The terms and definitions listed here is certainly not an exhaustive list. This section follows largely from Kolaczyk's *Statistical Analysis of Network Data*. Please see Kolaczyk (2009) for more details.

vertices $i, j \in V$, if $i$ and $j$ have a connection then the ordered pair $\{i, j\} = e \in E$. In this case, we say $i$ is *adjacent* to $j$ and that $i$ and $j$ are *neighbors*. A vertex is *incident* on an edge if that vertex is the endpoint of that edge. For example, a vertex $i$ would be incident on $e = \{i, j\}$ if j does not connect with any other vertex in V. Here, we make no distinction in the ordering of vertices. That is, $\{i, j\}$ is the same as $\{j, i\}$. Such a graph is called *undirected* (or *non-directed*).

The *order* of a graph G is defined to be the number of vertices in $G$, denoted $N_V = |V|$. Similarly, the *size* of $G$ is the number of edges in $G$, denoted $N_E = |E|$.

Graphs that have ordering to their edges, that is, where $\{i, j\} \in E$ is different from $\{j, i\} \in E$, are called *directed graphs* (or *digraphs*), and the edges in a directed graph are called *directed edges* (or *arcs*).

We can also represent a network with just the connections that are present (or not present) between vertices by using an adjacency matrix. Given a graph $G = (V, E)$, the adjacency matrix $\mathbf{A}$ is an $N_V$ x $N_V$ matrix in which:

$$A_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

Matrix $\mathbf{A}$ is symmetric if $G$ is a undirected graph, but this may not necessarily be the case if $G$ is a directed graph. This is because $\{i, j\} \in E$ does not necessarily guarantee that $\{j, i\} \in E$ for directed graphs.

While the adjacency matrix $\mathbf{A}$ is a data structure that fully characterizes a graph, for large $N_V$ size and reading time on the part of the computer can become an issue very quickly, especially in the context of working with data structures and particular algorithms. For a graph with $N_V$ vertices, the corresponding adjacency matrix will have $N^2$ elements. Thus, we can also consider an alternative: an *edge list* is a two-column list of all the edges and their corresponding vertices present in a graph. More often than not, a graph will not have an edge between every pair of vertices, so it is more often the case that using an edge list will mean we are dealing with strictly less than $N^2$ elements. Larger networks, as we will soon see, are more often than not represented with an edge list.

To describe how to move about in a network, we define a *walk* in graph $G$ to be a sequence that begins and ends with two vertices and alternates between vertices and edges. The sequence $(a_0, e_1, a_1, e_2, ..., v_{l-1}, e_l, v_l)$ denotes one path from vertex $a_0$ to vertex $a_l$, where for all $e_i \in E$, $e_i$ connects points $a_{i-1}$ and $a_i \in V$. Using this notation, we define the *length* of the walk to be the value $l$. More refined than a walk, a *trail* is defined to be a walk in which the sequence does not traverse through the same edge more than once. And even more refined still is a *path* in which a walk does not travel through the same vertex twice.

A *circuit* is a trail in which the starting and ending vertices is the same. A *cycle* is a walk with length of at least three that starts and ends at the same vertex but all other vertices are unique. An acyclic graph contains no cycles. The length of the shortest path(s) between any two vertices is defined to be the *distance* (or *geodesic distance*) between vertices. This distance is infinite if no path between two particular vertices exist. The *diameter* is the longest (geodesic) distance achieved in a graph. In

a digraph, a *directed walk* is analogous to a walk, but uses arcs to get from one vertex to another.

A vertex $j$ is *reachable* from another vertex $i$ is there is a walk from $i$ to $j$. G is *connected* if all vertices in $G$ are is reachable by some other vertex in $G$.

A *component* is a subgraph that is maximally connected. A *clique* is a graph or subgraph that is complete. A graph is *regular* if each vertex has the same *degree*. That is, each vertex has the same number of edges connecting to it. Degree is usually denoted $k$, and a particular vertex with a degree $k$ is called a *$k$-star*. A graph is *complete* if every vertex has an edge with all other vertices in the graph. An undirected graph is *simple* if it has no *loops* and *multi-edges*, meaning vertices do not have an edge that points to itself and any pair of vertices do not have more than one edge between them. Edges on a simple graph are called *proper edges*.

Connectedness and simpleness are actually very important characteristics of networks as many of the descriptive statistics that we will see can only be applied graphs with such properties. In fact, our focus here will be limited to simple and connected graphs.

A graph can contain other information or data, called *attributes*, beyond just the collection of vertices and edges. Incorporating these attributes, otherwise known as *decorating* a graph, can occur in both the vertices, edges, and even in the entire graph itself. For example, it is possible to assign *edge weights* to each edge. That is, we assign an edge weight $w_e$ to $e = \{i, j\} \in E$. A *weighted graph* is a graph whose edges include weights that are nonzero and between 0 an 1. Both edge and vertex attributes can be discrete or continuous. An example of a discrete attribute is a binary case of positive or negative relationship between pairs of vertices. An example of a continuous attribute would is the strength (or weight) of the relationship between pairs.

## 0.2 Example 1: Zachary's Karate Club

One famous (or infamous) social network, called "the karate club of Zachary," is observed by anthropologist Zachary (1977). The vertices indicate the 34 individual members of the karate club while the edges indicate the 78 social ties among pairs of the members. Figure 1 provides what is called a "visual summary" of the karate network data. As shown by the colors of the vertices, the club, at this particular moment in time, is being divided into two clusters centering around two vertices as labeled by letters (the master and primary disciple of the dojo).

It is possible to display more than just the actors and their relationships. As the network shows, we can provide more information by assigning other visual cues–such as vertex shapes to determine gender, vertex size to indicate belt rank, and edge weights to determine the strength of the social tie–to each member.

## 0.3 Example 2: Lazega's Law Firm

Consider a network of lawyers that reflects the relational data collected by Lazega & Pattison (1999) from a New England law firm consisting of over 70 lawyers to study

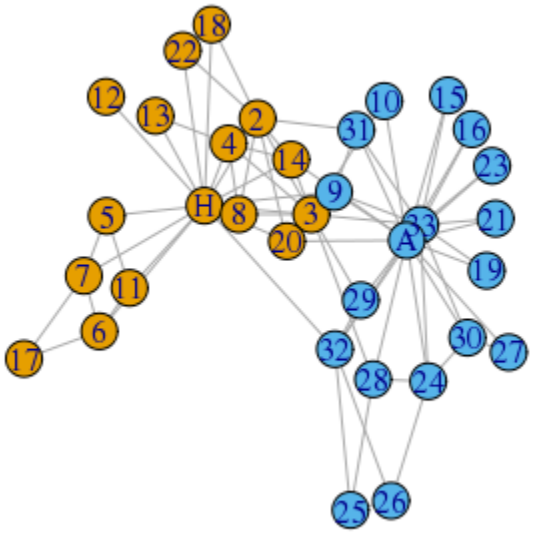


Figure 1: Zachary's Karate Club ($N_V = 34$, $N_E = 78$)

cooperation in organizations. This network has 36 lawyers and 115 collaborations. In Figure 2, each lawyer is individually labeled with smaller numbers indicating seniority. Shapes for each lawyer indicate the type of practice: litigation (circles) and corporate (squares). The shapes' sizes indicate the relative number of years of having been in with the law firm. Color is used to indicate office location (red, blue, and yellow). Edges indicate collaboration between partners. The graph below provides a visual summary of the lawyer network data.

We can also continue adding layers of data to this visual as well. For example, if the appropriate data was recorded, it is possible quantify and/or qualify the collaborations between the lawyers. We can apply weights to give an idea of how frequently any pair of lawyers interacted with each other in this resource exchange; we can even apply line types (solid, dashed, double-dashed, etc.) to indicate how this collaboration took place: was it done in person, through the phone, through email, or some combination of the three?

## 0.4 Modeling Real-World Networks

As hinted earlier, one topic of study with practical applications is using models to "model" observed networks. They are more formally known as *graph models* and for now, we can think of them ways to simulate graphs. Which models can generate graphs that are visually and structurally similar to the social networks that we observed? In reality, there are many graph models to choose from. With a network chosen, one approach is through a simulation study. We can generate graphs from different graph models and compare *descriptive network statistics* between the simulated graphs and

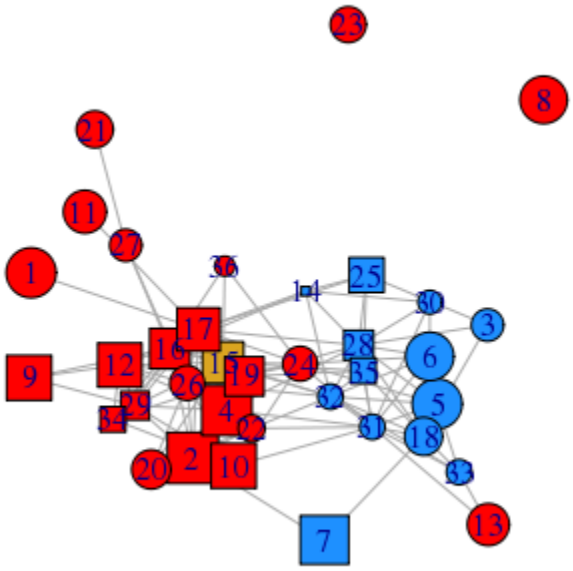


Figure 2: Lazega's Law Firm ($N_V = 36$, $N_E = 115$)

the observed graphs to see how accurate these graph models are.

Being able to model observed networks would benefit many fields. By accurately predicting how networks form, it may be possible to understand how friendships among people form or even how pieces of information spread across people. More generally, we can improve traffic flow, provide better product suggestions to online shoppers, and help prevent the spread of diseases.

In this work, we will provide a general overview about network statistics and graph models, followed by a simulation study that tests different graph models and see how well they "fit" a target social network of our choosing. In Chapter 1, we will focus on various network statistics. These statistics have direct (but sometimes non-trivial) analogs to the measures of spread and central tendency seen in introductory Statistics. For example, we will see that *centrality*, a measure of importance of vertices in graphs, is simular to the arithmitic mean for a set. In Chapter 2, we focus on describing various graph models, such as the Erdős-Rényi and Watts-Strogatz models and several different exponential random graph models (ERGMs). These graph models will be used to fit a social network of our choosing. Here, we will be using an undirected, simple, and connected component of the Facebook network consisting of 4039 vertices and 88234 edges. Chapter 3 describes our simulation study and results. We compare the network statistics of our simulated graphs to those of the observed data set. We close with our Conclusion (Chapter 4), which describes some of the limitations in this study and potential future work.

# Chapter 1

# Descriptive Network Statistics

Descriptive statistics for networks are somewhat analogous to those seen in elementary statistics. In fact, we have seen a few already in the section of basic terminology in our Introduction. We describe a few below and explain them in the context of Zachary's karate club and Lazega's lawy firm. Much of what is explained below follows from Kolaczyk (2009), Newman (2010) and Csárdi & Nepusz (2006). Please refer to these texts for more information.

## 1.1 Transitivity

*Transitivity*, otherwise known as the *clustering coefficient*, is the probability that adjacent vertices of a particular vertex are connected. In other words, this value provides a sense of how well-connected the graph is. A higher transitivity implies a graph having many nodes close to each other with links connecting all of them. There are many different ways to calculate transitivity, which largely depends on whether one wishes to talk about the network globally or locally.

Globally speaking, this value is the ratio of connected triples to all possible triangles in the graph. We define a *triangle* (or *triad*) as subgraphs with three nodes; if these nodes are connected by two edges, it is a *connected triple*. Formulaically, the clustering coefficient, denoted with uppercase $C$, is

$$C = \frac{(\text{number of triangles}) \times 3}{\text{number of connected triples}}$$

The factor of 3 represents the fact that a triangle is counted three times when we consider the number of triples in the network.

The local transitivity is calculated for each vertex in the graph and is the ratio of triangles connected to a particular vertex to the connected triples centered on the vertex. For a particular vertex $i$, the local transitivity is denoted as

$$C_i = \frac{(\text{number of pairs of neighbors of } i \text{ that are connected})}{(\text{number of pairs of neighbors of } i)}$$

For our purposes, we will use the local average transitivity, which is the average of the local transitivity values for each node in the graph.

In Zachary's Karate Club and Lazega's Law Firm, the local average transitivity values are: 0.5879 and 0.4867, respectively, and the global transitivity values are 0.2557 and 0.3888, respectively. This means that while the lawyer network has a greater amount of clustering than the karate network overall, locally speaking, there are more nodes in the karate network that form many triangles and connected triples compared to the lawyer network. In fact, as we will see, social networks in general tend to have higher-than-average transitivity values.

## 1.2 Average Path Length

As mentioned earlier, *average path length* (or *average geodesic distance*) is the average of the shortest paths of all distinct pairs of nodes in the network. This is a also commonly used statistic among social networks since average path lengths tend to be much less than what we would expect if the edges were placed between nodes randomly.

The karate network has an average path length of 2.4082 while the lawyer network has an average path length of 2.1444. This means that in both cases, any two people in both networks are ever separated by an average of about two people.

## 1.3 Diameter

As mentioned earlier as well, the *diameter* of a graph is the longest of all the shortest paths between distinct pairs of nodes. This value gives our graph a notion of distance. Unlike average path length, however, diameter can be unpredictable at times since this value can be an outlier compared to other path lengths in the network.

The diameters of the karate and lawyer networks are 13 and 5, respectively. In a law firm of 36, people are only as far as five people away from each other, but for the karate network, even in a group of 34, there exists two club members that do not have a direct relationship with each other and are separated by 12 other members.

## 1.4 Centrality

*Centrality* provides a measure of the "importance" to each node in the network. It is similar to the measures of central tendency seen in elementary statistics. Many different types of centrality have been proposed, some of which are discussed below. Here, we assume our graph $G$ is undirected.

*Vertex degree* is perhaps the most common type of centrality. It is not surprising that vertices with higher vertex degrees are considered to be more central to the network than those with lower vertex degrees. One of the properties of networks is that the degree of the nodes of a graph tend to follow a *power law distribution*. This property indicates that quite a few nodes with particularly high degrees occur very frequently (compared to if it followed a Poisson distribution, for example). The mean

of the vertex degrees of all the vertices of the graph yields the *mean* (or *average*) *degree*, often denoted with lowercase $c$.

*Closeness centrality* measures how close a vertex is to other vertices by based on the inverse of the total distance of the vertex from all others.

$$c_{Cl}(i) = \frac{1}{\sum_{i \in V} d(i,j)}$$

where $dist(i,j)$ is the geodesic distance between the vertices $i, j \in V$. To compare across graphs and with other centrality measures, this measure is often normalized to lie in the interval $[0,1]$ by multiplying by a factor of $N_V - 1$.

*Betweenness centrality* measures are aimed at summarizing the extent to which a vertex is located between other pairs of vertices. The most commonly used version of betweenness centrality is

$$c_B(i) = \sum_{g \neq h \neq i \in V} \frac{\sigma(g,h|i)}{\sigma(g,h)}$$

,

where $\sigma(g,h|i)$ is the total number of shortest paths between $g$ and $h$ that pass through $i$, and $\sigma(g,h) = \sum_V \sigma(g,h|i)$. If the shortest paths are unique, $c_B(i)$ just counts the number of shortest paths going through $i$. This normalized by dividing by a factor of $\frac{(N_V-1)(N_V-2)}{2}$

*Eigenvector centrality* is based on the idea of "status," "prestige," or "rank;" the more central the neighbors of a vertex are, the more central that vertex itself is. One definition of such a centrality measure is:

$$c_{Ei}(i) = \alpha \sum_{\{i,j\} \in E} c_{Ei}(u)$$

The vector $c_{Ei} = (c_{Ei}(1), ..., c_{Ei}(N_V))^T$ is the solution to the eigenvalue problem $\mathbf{A}\mathbf{c}_{Ei} = \alpha^{-1}\mathbf{c}_{Ei}$, where $\mathbf{A}$ is the adjacency matrix for network graph $G$.

Centrality measures, like the local average transitivity, provide a value for each vertex in the graph. Because of this, we can obtain an overall centrality measure for the graph as a whole if we average the values for each vertex.

Table 1.1: Average centrality measures for the karate and lawyer networks

| Network Statistic | Zachary's Karate Club | Lazega's Law Firm |
|---|---|---|
| Average Degree | 4.588 | 6.389 |
| Average Betweenness Centrality | 26.194 | 17.833 |
| Average Closeness Centrality | 0.005 | 0.007 |
| Average Eigenvector Centrality | 0.377 | 0.458 |

Table 1.1 above shows the average centrality measures for both the karate and

lawyer networks. Because each type of centrality is measuring importance of nodes differently, it comes as no surprise that all of the average centrality values within one network will be different from each other. These average values, however, do not give any indication of which nodes in the network are important, or how many. Additionally, the use of an average means that the value can be subjected to some very drastic outliers that could potentially skew its value.

We will use these seven network statistics listed above in our simulation study. By calculating these values from our observed network, it is possible for us to compare our network with graphs simulated from our graph models.

# Chapter 2

# Graph Models

A *graph model* takes in fixed parameters that generate a graph that can vary in structure with each iteration. Equivalently, it is also possible to consider a model for a graph as a collection, or *ensemble*,

$$\{\mathbb{P}_\theta(G), G \in \mathcal{G} : \theta \in \Theta\}$$

in which $G$ is a collection or ensemble of possible graphs, $P_\theta$ is a probability distribution on $G$, and $\theta$ is a vector of parameters, ranging over possible parameters in $\Theta$.

Many of the explanations and derivations for the Erdős-Rényi and Watts-Strogatz networks here follow from Newman (2010). The section about exponential random graph models (ERGMs) follow largely from C. Butts et al. (2015).

## 2.1 Erdős-Rényi

The *Erdős-Rényi model* (also known as the *Erdős-Rényi-Gilbert model*) is one of the most studied graph models.[1] It is also one of the simplest, as it takes only two parameters. The model $G(N_V, N_E)$, first suggested by Gilbert (1959), takes in $N_V$, the number of nodes, and $N_E$ the number of edges. Erdős and Rényi (1959, 1960, 1964) considered the model of the form $G(N_V, p)$, where instead of using the number of edges, the probability of an edge forming between any pairs of nodes is fixed. Our focus is on the latter model.

In the $G(N_V, p)$ model, graphs constructed according to this model could potentially look every different from each other. This goes back to the idea that a graph model can be thought of as a probability distribution over an ensemble of networks, and all networks in this particular ensemble have equal probability of being chosen. We

[1] Some of our network statistics that we calculated in R did not match those seen in the websites. For example, we calculated the diameter and transitivity of this Facebook component to be 17 and 0.617, respectively. However, the SNAP website reports the diameter and "average clustering coefficient" to be 8 and 0.6055, respectively. However, other values, such as node and edge count, as well as the number of triangles, are equal. While SNAP's algorithm used to calculate their network statistics is unclear, for the purposes of this simulation study, we will be using the values that we calculated from the Facebook component to compare with those generated from our graph models.


Figure 2.1: Examples of graphs generated from the Erdős-Rényi model. Left and Right: $N_V = 10$, $p = 0.25$.

immediately see a drawback in graphs generated from this model: it places no significance on structures that we may see in our observed graph–such as dense clusters or cliques.

In spite of this, the Erdős-Rényi model has very nice properties, some of which we discuss here. Even with only two parameters $N_V$ and $p$ on hand, we can still create formulas that allow us to calculate various network statistics, such as average degree and clustering coefficient, and describe the model's degree distribution and giant component.

As we have mentioned earlier, $G(N_V, p)$ model is the collection of simple graphs with exactly $n$ vertices, meaning that any simple graph $G$ with exactly $N_V$ vertices has probability

$$P(G) = p^{N_E}(1-p)^{\binom{N_V}{2}-N_E)}$$

of being picked. For precisely $N_E$ edges, there are $\binom{\binom{n}{2}}{m}$ ways to arange the $N_E$ edges among the $\binom{N_V}{2}$ possible edges. Thus, total probability of a random graph $G$ with $N_E$ edges and $N_V$ vertices is

$$P(N_E) = \binom{\binom{n}{2}}{m} p^{N_E}(1-p)^{\binom{N_V}{2}-N_E)}$$

This, however, is simply a binomial distribution, where we have some probability of success ($p$), two possible outcomes (edge formation or no edge formation), a finite number of trials ($\binom{N_V}{2}$ distinct edges), and $\binom{\binom{N_V}{2}}{m}$ different ways in which the outcomes

can be arranged. Using this, the mean value of $N_E$ for the model is then a weighted average. It is the sum of the products of every possible number of edges $N_E$ and the total probability that a graph with $N_V$ vertices and N_{E} edges appears, $P(N_E)$. However, because we know the probability of an edge forming, the mean number of edges would eqaul to the product of the total number of possible vertices $\binom{N_V}{2}$ and the probability $p$. This makes sense as we can expect that $100p$ percent of the possible edges in the graph to actually have edges. Thus,

$$\langle N_E \rangle = \sum_{N_E=0}^{\binom{n}{2}} N_E P(N_E) = \binom{n}{2} p$$

For a graph with exactly $N_E$ edges, it is easy to see that the mean degree is $\frac{2N_E}{N_V}$. The factor of 2 allows an edge to be counted as part of the degree for each of the pair vertices that it connects. By taking a similar weighted average, where we sum the products of the average degree of a graph with $N_E$ edges with the total probability of a graph with $N_E$ edges being chosen, we can get the average degree, denoted $c$, for this model. Using the formula for the mean number of edges above, we can simplify this weighted average

$$c = \langle N_V \rangle = \sum_{N_E=0}^{\binom{n}{2}} \frac{2N_E}{N_V} P(N_E) = \frac{2}{N_V} \binom{N_V}{2} p = (N_V - 1)p$$

The right-hand side of this equation makes sense; for any vertex on the random graph, we would expect $100p$ percent of the other $N_V - 1$ vertices to be connected to it.

We can also describe the distribution of the $G(N_V, p)$. We show it is a binomial distribution, and that it actually converges to the Poisson distribution as the number of vertices {N_V} increases. To see this, observe that the probability $p_k$ of

To use this model in our simulation study, we simply take the number of nodes and the probability of a link forming equal to the number of observed edges divided by the number of possible edges (the number of nodes choose 2). That is, $\frac{N_E}{\binom{N_V}{2}}$.

## 2.2 Watts-Strogatz

Watts & Strogatz (1998) noticed that many networks have high levels of clustering and only require short average path lengths between mode nodes. In the Erdős-Rényi model, simulated graphs of parable magnitude tend to have smaller-than-expected clustering coefficients. In the Watts-Strogatz model, we start with $N_V$ vertices arranged in a circular fashion, $r$ number of beginning neighbors for each node, and the probability $p$ of an edge being moved to another pair of vertices.

In a regular lattice with 10 nodes, each having two neighbors, the clustering coefficient is relatively high at exactly 0.5. Diameter and average path length are nontrivial, and can be rather high as well (at 3 and 1.667, respectively).

By adding a probability of rewiring, the clustering coefficient increases, while diameter and average path length changes, but not necessarily in predictable ways.


Figure 2.2: Examples of graphs generated from the Watts-Strogatz model. Both: $N_V = 10\$, r = 2$. Left: $p = 0$. Right: $p = 0.25$.

This makes sense because this rewiring essentially creates random shortcuts in the network framework. In the right-hand graph of Figure 3.2, which is the left-hand graph after a rewiring probability of 0.25, we see that the transitivity, diameter, and average path length are now 0.507, 4, and 1.733, respectively.

While it is entirely possible that a network could potentially have such a kind of rewiring process, it is less likely that this is true in the context of social networks. In other words, ties between people do not usually disappear and randomly reappear between others. However, we can still make use of the fact that it starts with a connected lattice structure with some number of neighbors and randomly add edges until we get a graph of similar order and size. In the case where we start with a lattice of $r = 1$ neighbor, however, this is almost no different from the Erdős-Rényi model. This is because a lattice structure with only one neighbor is very similar to a random assignment of edges (but with the restiction that each node only has a degree of 1). Additionally, depending on the number of edges that need to be added to the network, the starting lattice may not have a large effect on the overall structure of the graph generated.

## 2.3 Exponential Random Graph Models (ERGMs)

Exponential random graph models (ERGMs) are a class of models that can also be used to generate probability distributions for a set of random graphs. ERGMs have enormous flexibility in construction since it allows any combinition of parameters from a given data set to be used in constructing the model. We are not only able to simulate

additional graphs from the underlying probability distribution like in the Erdős-Rényi and Watts-Strogatz models, but we are also able to obtain maximum-likelihood estimates of the model for the data set, conduct goodness-of-fit tests for

The general form for an ERGM is as follows:

$$P(Y = y) = \frac{exp(\theta' g(y))}{k(\theta)}$$

where $Y$ is the random variable for the state of the network and realization y, $g(y)$ is the vector of model statistics for the network $y$, $\theta$ is the vector of coefficients for those statistics, and $k(\theta)$ is the quantity in the numerator summed over all possible networks.

At the time of this writing, we are still working on the explanation of properties of ERGMs. This will be included as part of the Mathematics thesis.

### 2.3.1 Deriving Erdős-Rényi-Gilbert from ERGMs

The Erdős-Rényi-Gilbert and Watts-Strogatz models are actually special cases of ERGMs. Here, we will show that the Erdős-Rényi model can be derived from an ERGM that only takes into acount the number of edges in a graph.

# Chapter 3

# Simulation Study

## 3.1 Description of Our Dataset

Here we will use a subset of the Facebook network to fit graph models to the observed model and assess each one's accuracy. This Facebook network was compiled by Stanford University and accessed through its Stanford Large Network Data Collection (SNAP). For more information, please see Jure Leskovec & Krevl (2014). Our subset of the Facebook network was presented as an edge list. The social network is one giant component composed of 4039 nodes and 88234 edges, which represent 4039 anonymous users and the 88234 connections between them, respectively.[1] This network is simple and undirected, meaning the network consists of established, mutual friendships. The first column of values in Table 3.1 below describes some characteristics of this network. We use the `igraph`, `sna`, `network`, `ergm`, `statnet` packages in our study. Please see, Csárdi & Nepusz (2006), Carter T Butts & others (2008), Carter T. Butts (2015), Carter T. Butts (2008), Handcock et al. (2017), Hunter et al. (2008), Handcock et al. (2016), Handcock, Hunter, Butts, Goodreau, & Morris (2008), and Bojanowski (2015) for more details.

```
setwd("~/STAT495-Lee/LeeThesis/data")

#edge list
facebookcombined <- read.table(gzfile("facebook_combined.txt.gz"), header=F)

#igraph object
fbel <- graph.data.frame(facebookcombined)
```

[1] Some of our network statistics that we calculated in R did not match those seen in the websites. For example, we calculated the diameter and transitivity of this Facebook component to be 17 and 0.617, respectively. However, the SNAP website reports the diameter and "average clustering coefficient" to be 8 and 0.6055, respectively. However, other values, such as node and edge count, as well as the number of triangles, are equal. While SNAP's algorithm used to calculate their network statistics is unclear, for the purposes of this simulation study, we will be using the values that we calculated from the Facebook component to compare with those generated from our graph models.

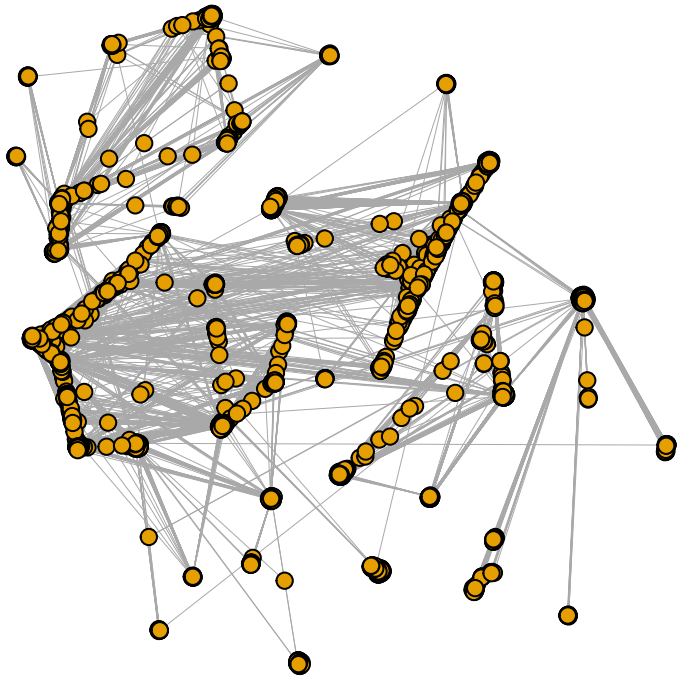

Figure 3.1: Overview of our component of the Facebook network

## 3.2 Generating Random Graphs

For each model, we created graphs of parable magnitude using some of the information calculated from the our Facebook network. We compared selected statistical network statistics of the Facebook network with that of the graphs we generate to see if the observed network and the simulated graphs have similar characteristics. We explain the process of generating one random graph for each model in the sections below. We simulated 1000 random graphs for each model and recorded the statistics such as transitivity, diameter, and average centrality values. Because we had 1000 values (one for each random graph for each model), we then recorded the average of these values. This means that for statistics such as average degree, we were interested in recording an average of the averages. The first column of Table 3.1 shows the network statisitcs we were interested in and the average statistics recorded under each model as well as the standard deviation.

### 3.2.1 Using the Erdős-Rényi Model

The Erdős-Rényi model only takes in two parameters, which are the number of nodes and the fixed probability of edge formation. The number of edges is already given as 4039, but to find the probability, we estimate this by taking the number of obsevred edges and divide it by the number of possible edges. For a graph $G$, this estimated probability is

$$\hat{p} = \frac{N_E}{\binom{N_V}{2}}$$

where $N_E$ is the number of edges in $G$, $V_G$ is the number of nodes in $G$.

Thus, in order to simulate one graph from this model, we must look at each possible edge among the 4039 nodes and determine if an edge will form based on the given probability.

```
set.seed(499)

numsim <- 1000

g.er.fbel.ecount <- rep(NA, numsim)
g.er.fbel.coef <- rep(NA, numsim)
g.er.fbel.apl <- rep(NA, numsim)
g.er.fbel.dia <- rep(NA, numsim)
g.er.fbel.avgdeg <- rep(NA, numsim)
g.er.fbel.avgbtwcen <- rep(NA, numsim)
g.er.fbel.avgclocen <- rep(NA, numsim)
g.er.fbel.avgeigveccen <- rep(NA, numsim)

#probability used in for-loop below
```

```
p = ecount(fbel)/choose(vcount(fbel), 2)

for (i in 1:numsim) {
  # n = number of vertices, p = probability of a link
  #p is caluclated as number of edges over number of possible edges
  #p is then (4039 choose 2) as calculated above
  #igraph object
  g.er.fbel <- erdos.renyi.game(n = vcount(fbel), p)

  #observe the network statistics for this simulated graph
  g.er.fbel.ecount[i] <- ecount(g.er.fbel)
  g.er.fbel.coef[i] <- transitivity(g.er.fbel, type="localaverage")
  g.er.fbel.apl[i] <- average.path.length(g.er.fbel)
  g.er.fbel.dia[i] <- diameter(g.er.fbel)
  g.er.fbel.avgdeg[i] <- mean(igraph::degree(g.er.fbel))
  g.er.fbel.avgbtwcen[i] <- mean(igraph::betweenness(g.er.fbel))
  g.er.fbel.avgclocen[i] <- mean(igraph::closeness(g.er.fbel))
  g.er.fbel.avgeigveccen[i] <- mean(igraph::eigen_centrality(g.er.fbel)$vector)
}

#save values as dataframe
g.er.fbel <- as.data.frame(cbind(g.er.fbel.ecount, g.er.fbel.coef,
                                 g.er.fbel.apl, g.er.fbel.dia,
                                 g.er.fbel.avgdeg, g.er.fbel.avgbtwcen,
                                 g.er.fbel.avgclocen, g.er.fbel.avgeigveccen))
```

### 3.2.2 Using the Watts-Strogatz Model

The Watts-Strogatz model takes in the following: the number of nodes $N_V$, the number of starting neighbors $r$, and the probability of an edge to be rewired $p$. However, instead of rewiring our edges with a fixed probability, we will instead randomly add edges until we have approximately the same number as our observed network. To do this, we start with a lattice with 4039 nodes and assign a number of edges to the nodes equal to the smallest degree seen in our Facebook network (which we find to be 1). This is our starting number of neighbors for each node. In this situation, $p = 0$. We then randomly add edges equal to the difference between the our starting lattice and the observed network. That is, we add $88234 - 4039 = 84195$ edges to our lattice. Finally, we simplify our simulated graph to eliminate multi-edges and loops. Since there are $\binom{4039}{2}$ edges to choose from, intuitively, we can see that having many multi-edges or loops will be infrequent. This makes up one simulated graph from the model.

```
set.seed(499)
```

```
numsim <- 1000

g.ws.fbel.ecount <- rep(NA,numsim)
g.ws.fbel.coef <- rep(NA, numsim)
g.ws.fbel.apl <- rep(NA, numsim)
g.ws.fbel.dia <- rep(NA, numsim)
g.ws.fbel.avgdeg <- rep(NA, numsim)
g.ws.fbel.avgbtwcen <- rep(NA, numsim)
g.ws.fbel.avgclocen <- rep(NA, numsim)
g.ws.fbel.avgeigveccen <- rep(NA, numsim)

for (i in 1:numsim) {
  #create a lattice with the same number of vertices as 'fbel'
  #let the starting number of neighbors be
   #equal to the minimum vertex degree of 'fbel'
  g.ws.fbel <- watts.strogatz.game(dim = 1, size = vcount(fbel),
                                   nei = min(degree(fbg)), p = 0)

  #generate list of random vertex values to add edges to lattice
  #each pair in the list represents one random edge
  randomedgepairs <- sample(1:vcount(fbel), 2*(ecount(fbel)-vcount(fbel)),
                            replace=TRUE)
  g.ws.fbel.pre <- add_edges(g.ws.fbel, randomedgepairs)


  #igraph object
  g.ws.fbel <- simplify(g.ws.fbel.pre)

  #observe the network statistics for this simulated graph
  g.ws.fbel.ecount[i] <- ecount(g.ws.fbel)
  g.ws.fbel.coef[i] <- transitivity(g.ws.fbel, type="localaverage")
  g.ws.fbel.apl[i] <- average.path.length(g.ws.fbel)
  g.ws.fbel.dia[i] <- diameter(g.ws.fbel)
  g.ws.fbel.avgdeg[i] <- mean(igraph::degree(g.ws.fbel))
  g.ws.fbel.avgbtwcen[i] <- mean(igraph::betweenness(g.ws.fbel))
  g.ws.fbel.avgclocen[i] <- mean(igraph::closeness(g.ws.fbel))
  g.ws.fbel.avgeigveccen[i] <- mean(igraph::eigen_centrality(g.ws.fbel)$vector)
}

#save values as dataframe
g.ws.fbel <- as.data.frame(cbind(g.ws.fbel.ecount, g.ws.fbel.coef,
                                 g.ws.fbel.apl, g.ws.fbel.dia,
                                 g.ws.fbel.avgdeg, g.ws.fbel.avgbtwcen,
                                 g.ws.fbel.avgclocen, g.ws.fbel.avgeigveccen))
```

### 3.2.3 Using ERGMs

At the time of this writing, I am still currently working on the code for this section of the simulation study. A description of random graph construction and results, along with an associated code chuck will be included in the thesis submission of this project.

```
#needs an object of class network
#this should be the same as Erdős-Rényi
g.ergm1.fbg <- ergm(fbg ~ edges)

set.seed(499)
g.ergm1.fbg.sim <- simulate(g.ergm1.fbg, numsim=1000)



g.ergm2.fbg <- ergm(fbg ~ edges+triangle)

set.seed(499)
g.ergm2.fbg.sim <- simulate(g.ergm2.fbg, numsim=1000)
```

Table 3.1: Comparisons of Network Statistics in among graphs simulated from classical graph models and the observed network

| Network Statistic | Observed | Erdős-Rényi | Watts-Strogatz |
|---|---|---|---|
| Transitivity | 0.617 | 0.0108 ± 0.0001 | 0.0107 ± 9.135e-05 |
| Average Path Length | 4.338 | 2.606 ± 0.002 | 2.6093 ± 0.0002 |
| Diameter | 17 | 3.96 ± 0.21 | 3.95 ± 0.22 |
| Average Degree | 43.691 | 43.69 ± 0.14 | 43.45 ± 0.01 |
| Average Betweenness Centrality | 2072.642 | 3242 ± 4 | 3249.2 ± 0.4 |
| Average Closeness Centrality | 8.881e-08 | 9.507e-05 ± 7.230e-08 | 9.494e-05 ± 7.319e-09 |
| Average Eigenvector Centrality | 0.040 | 0.6202 ± 0.0224 | 0.6235 ± 0.0227 |

Based on the results above, we see that the estimated average statistics for the Erdős-Rényi and Watts-Strogatz models are very similar to each other but are drastically different from many of the observed values of our Facebook network. This means that while the graphs generated from each model look very similar to each other, they look drastically different from our observed Facebook network. This is no surprise since the methods used to construct graphs under both models are very likely to generate almost the same number of verticles. We even see this in our simulation study; in the fourth row of Table 3.1, the average degree estimate for both models was within one degree from each other. Additionally, in both constructions, we are adding the edges randomly, so they are almost always spread evenly throughout the 4039 nodes. This means that additional features of the observed network, such as large clusters or degrees with high centralities, are likely to be missed by graphs generated from these two models. This is why the transitivity, average path length, and diameter for both models are smaller than what was actually observed. Similarly, because of how dispersed the edges are throughout the nodes, it is likely that there are many more nodes with higher centralities than in the observed network, making the average centrality measures for the graph models larger than those seen in the observed model. The only statistic whose value was actually close to, if not equal to, the observed value was the average degree. However, this makes sense since we fixed the probability $p$ of a link forming in the Erdős-Rényi model (which of is directly proportional to the average degree) and because we ensured that every graph generated from the Watts-Strogatz model has about the same number of edges as the observed network. Thus, we have shown through this simulation study that the Erdős-Rényi and Watts-Strogatz models are not very accurate in modeling our Facebook network for these particular network statistics.

# Conclusion

In this work, we have provided a brief overview of networks, network statistics, and graph models. We also applied certain graph models and fit them to our component of the Facebook network. By letting the graph models take in parameters that match those of our Facebook network, we generated graphs of parable magnitude. We compare the network statistics between these generated graphs and our target network to measure each model's accuracy. We showed through a simulation study that both the Erdős-Rényi and Watts-Strogatz models do not do a good job of modeling the Facebook network for particular network statistics.

Over the next month, we will be creating ERGM models to fit the Facebook network and simulating from these models to see if the network statistics of these simulated graphs are closer to that of the observed network. As described in Chapter 3, once main advantage of ERGMs is that they allow for flexibility in model construction. This means we will be able to look at structures commonly seen in models and use them as parameters in ERGMs. This can include things such as k-stars (nodes with degree $k$), triangles, and distance, for example.

This simulation study only scratches the surface of what network science has to offer. Firstly, we can bring in other social networks into this study. Could underlying undirected graphs of directed networks be better fit for the models here? Could network size be a factor as well? It is also possible for us to test other network statistics. Without providng the definitions, we can consider values such as *reciprocity*, *assortivity*, or even other centrality measures. See Newman (2010) for more details. It is equally possible for us to choose other models to test our current Facebook network on. In fact, many other graphs models exist beyond what was presented here. Because social networks exhibit certain characteristics different from other kinds of networks, it comes as no surprise that there have been other models that are specifically tailored to its study. See Toivonen et al. (2009) for more a study of *network evolution models* and *nodal attribute models*, two specific types of social network models. Also, see Jurij Leskovec, Chakrabarti, Kleinberg, & Faloutsos (2005) and Jure Leskovec, Chakrabarti, Kleinberg, Faloutsos, & Ghahramani (2010) for an overview of networks in general and also recent developments in network science. They introduce *Kronecker graphs* that have the benefit of only requiring a few parameters to fit to a target network in linear time. A successful simulation study has many implications that can benefit not only just network science, but all disciplines.

# Appendix A

# R/R-Markdown Code

This first appendix includes all of the R chunks of code used throughout the document (using the `include = FALSE` chunk tag) to help with readability and/or setup. Some code chunks were shown in the main body of the chapters for clarity and illustration.

**In the main Rmd file:**

This is the setup file for the template.

```
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
}
library(acstats)
```

The following are the packages used throughout this work. In particular, the `igraph`, `network`, `statnet`, and `ergm` packages were heavily used throughout all the chapters.

```
library(sand)
library(igraph)
library(network)
library(sna)
library(statnet)
library(ergm)
library(xtable)
```

```
options(xtable.comment = FALSE)
options(digits = 4)
```

### In Introduction

We use the `karate` and `lawyer` data sets as examples to illustrate our discussion of networks.

```
data(karate)
data(lazega)
```

The following code chunk generates a visual representing Zachary's karate club network.

```
plot(karate)
```

The following code chunk generates a visual representing Lazega's law firm network.

```
# Office location indicated by color.
colbar <- c("red", "dodgerblue", "goldenrod")
v.colors <- colbar[V(lazega)$Office]
# Type of practice indicated by vertex shape.
v.shapes <- c("circle", "square")[V(lazega)$Practice]
# Vertex size proportional to years with firm.
v.size <- 3.5*sqrt(V(lazega)$Years)
# Label vertices according to seniority.
v.label <- V(lazega)$Seniority
# Reproducible layout.
set.seed(42)
l <- layout.fruchterman.reingold(lazega)
plot(lazega, layout=l, vertex.color=v.colors,
     vertex.shape=v.shapes, vertex.size=v.size,
     vertex.label=v.label)
```

### In Chapter 1

```
mean(igraph::degree(karate))
#average degree centrality: 4.588235
mean(igraph::betweenness(karate))
#average betweenness centrality: 26.19363
mean(igraph::closeness(karate))
#average closeness centrality: 0.005450958
mean(igraph::eigen_centrality(karate)$vector)
```

```
#average eigenvector centrality 0.3772814

mean(igraph::degree(lazega))
#average degree centrality: 6.388889
mean(igraph::betweenness(lazega))
#average betweenness centrality: 17.83333
mean(igraph::closeness(lazega))
#average closeness centrality: 0.00670515
mean(igraph::eigen_centrality(lazega)$vector)
#average eigenvector centrality 0.4578719
```

### In Chapter 2

This generates two igraph objects from the Erdős-Rényi model with specified parameters–the number of nodes being 10 and probability of edge formating being 0.25.

```
set.seed(499)

g1.er <- erdos.renyi.game(n = 10, p = 0.25)
g2.er <- erdos.renyi.game(n = 10, p = 0.25)
```

The following code plots the two graphs simulated from the Erdős-Rényi model.

```
par(mfrow=c(1,2))
plot(g1.er, vertex.size=20, vertex.label.cex = 0.75)
plot(g2.er, vertex.size=20, vertex.label.cex = 0.75)
```

This generates two igraph objects from the Watts-Strogatz model with specified parameters–the number of nodes being 10, number of neighbors being 2, and probabilities 0 (left) and 0.25 (right).

```
set.seed(499)

g1.ws <- watts.strogatz.game(dim = 1, size = 10, nei = 2, p = 0)
g2.ws <- watts.strogatz.game(dim = 1, size = 10, nei = 2, p = 0.25)
```

The following code plots the two graphs simulated from the Watts-Strogatz model.

```
par(mfrow=c(1,2))
plot(g1.ws, vertex.size=20, vertex.label.cex = 0.75)
plot(g2.ws, vertex.size=20, vertex.label.cex = 0.75)
```

### In Chapter 3

We load in our Facebook data set, that is, our observed network. It is a simple, undiected, connected component of the network in the form of an edge list. We convert to an `igraph` object in order to calculate certain network statistics.

```
setwd("~/STAT495-Lee/LeeThesis/data")

#edge list
facebookcombined <- read.table(gzfile("facebook_combined.txt.gz"), header=F)

#igraph object
fbel <- graph.data.frame(facebookcombined)
```

Below are some numbers (and the associated code for it) that characterize the observed Facebook network as well as some associated descriptive statistics.

```
length(unique(c(facebookcombined$V2, facebookcombined$V1))); vcount(fbel)
#number of vertices: 4039
nrow(facebookcombined); ecount(fbel)
#number of edges: 88234
sum(count_triangles(fbel))/3
#number of unique triangles (up to ordering): 1612010

transitivity(fbel, type="localaverage") #0.6170038
diameter(fbel) #17
average.path.length(fbel) #4.337744

mean(igraph::degree(fbel))
#average degree centrality: 43.69101
mean(igraph::betweenness(fbel))
#average betweenness centrality: 2072.642
mean(igraph::closeness(fbel))
#average closeness centrality: 8.881448e-08
mean(igraph::eigen_centrality(fbel)$vector)
#average eigenvector centrality 0.04047316
```

The code below plots the entire component of the Facebook that SNAP has provided.

```
plot(fbel,
     edge.arrow.size = 0,
     edge.width = 0.05,
     vertex.label = NA,
     vertex.size = 5)
```

The following code describes the data simulation process using the Erdős-Rényi model. We create seven empty vectors and fill the $i^{th}$ term of each vector by running through a for loop that calculates seven network statisitcs for the $i^{th}$ random graph generated from the model.

```
set.seed(499)

numsim <- 1000

g.er.fbel.ecount <- rep(NA, numsim)
g.er.fbel.coef <- rep(NA, numsim)
g.er.fbel.apl <- rep(NA, numsim)
g.er.fbel.dia <- rep(NA, numsim)
g.er.fbel.avgdeg <- rep(NA, numsim)
g.er.fbel.avgbtwcen <- rep(NA, numsim)
g.er.fbel.avgclocen <- rep(NA, numsim)
g.er.fbel.avgeigveccen <- rep(NA, numsim)

#probability used in for-loop below
p = ecount(fbel)/choose(vcount(fbel), 2)

for (i in 1:numsim) {
  # n = number of vertices, p = probability of a link
  #p is caluclated as number of edges over number of possible edges
  #p is then (4039 choose 2) as calculated above
  #igraph object
  g.er.fbel <- erdos.renyi.game(n = vcount(fbel), p)

  #observe the network statistics for this simulated graph
  g.er.fbel.ecount[i] <- ecount(g.er.fbel)
  g.er.fbel.coef[i] <- transitivity(g.er.fbel, type="localaverage")
  g.er.fbel.apl[i] <- average.path.length(g.er.fbel)
  g.er.fbel.dia[i] <- diameter(g.er.fbel)
  g.er.fbel.avgdeg[i] <- mean(igraph::degree(g.er.fbel))
  g.er.fbel.avgbtwcen[i] <- mean(igraph::betweenness(g.er.fbel))
  g.er.fbel.avgclocen[i] <- mean(igraph::closeness(g.er.fbel))
  g.er.fbel.avgeigveccen[i] <- mean(igraph::eigen_centrality(g.er.fbel)$vector)
}

#save values as dataframe
g.er.fbel <- as.data.frame(cbind(g.er.fbel.ecount, g.er.fbel.coef,
                                 g.er.fbel.apl, g.er.fbel.dia,
                                 g.er.fbel.avgdeg, g.er.fbel.avgbtwcen,
                                 g.er.fbel.avgclocen, g.er.fbel.avgeigveccen))
```

The following code describes the data simulation process using the Watts-Strogatz model. This process is analogous to the process described above.

```
set.seed(499)

numsim <- 1000

g.ws.fbel.ecount <- rep(NA,numsim)
g.ws.fbel.coef <- rep(NA, numsim)
g.ws.fbel.apl <- rep(NA, numsim)
g.ws.fbel.dia <- rep(NA, numsim)
g.ws.fbel.avgdeg <- rep(NA, numsim)
g.ws.fbel.avgbtwcen <- rep(NA, numsim)
g.ws.fbel.avgclocen <- rep(NA, numsim)
g.ws.fbel.avgeigveccen <- rep(NA, numsim)

for (i in 1:numsim) {
  #create a lattice with the same number of vertices as 'fbel'
  #let the starting number of neighbors be
   #equal to the minimum vertex degree of 'fbel'
  g.ws.fbel <- watts.strogatz.game(dim = 1, size = vcount(fbel),
                                   nei = min(degree(fbg)), p = 0)

  #generate list of random vertex values to add edges to lattice
  #each pair in the list represents one random edge
  randomedgepairs <- sample(1:vcount(fbel), 2*(ecount(fbel)-vcount(fbel)),
                            replace=TRUE)
  g.ws.fbel.pre <- add_edges(g.ws.fbel, randomedgepairs)


  #igraph object
  g.ws.fbel <- simplify(g.ws.fbel.pre)

  #observe the network statistics for this simulated graph
  g.ws.fbel.ecount[i] <- ecount(g.ws.fbel)
  g.ws.fbel.coef[i] <- transitivity(g.ws.fbel, type="localaverage")
  g.ws.fbel.apl[i] <- average.path.length(g.ws.fbel)
  g.ws.fbel.dia[i] <- diameter(g.ws.fbel)
  g.ws.fbel.avgdeg[i] <- mean(igraph::degree(g.ws.fbel))
  g.ws.fbel.avgbtwcen[i] <- mean(igraph::betweenness(g.ws.fbel))
  g.ws.fbel.avgclocen[i] <- mean(igraph::closeness(g.ws.fbel))
  g.ws.fbel.avgeigveccen[i] <- mean(igraph::eigen_centrality(g.ws.fbel)$vector)
}

#save values as dataframe
```

```
g.ws.fbel <- as.data.frame(cbind(g.ws.fbel.ecount, g.ws.fbel.coef,
                                 g.ws.fbel.apl, g.ws.fbel.dia,
                                 g.ws.fbel.avgdeg, g.ws.fbel.avgbtwcen,
                                 g.ws.fbel.avgclocen, g.ws.fbel.avgeigveccen))
```

The following code descibes how to fit graphs using ERGMs and simulating from the models that have been constructed.

```
#needs an object of class network
#this should be the same as Erdős-Rényi
g.ergm1.fbg <- ergm(fbg ~ edges)

set.seed(499)
g.ergm1.fbg.sim <- simulate(g.ergm1.fbg, numsim=1000)



g.ergm2.fbg <- ergm(fbg ~ edges+triangle)

set.seed(499)
g.ergm2.fbg.sim <- simulate(g.ergm2.fbg, numsim=1000)
```

### In Conclusion

None

# References


Baumer, B., Kaplan, D., & Horton, N. (2017). *Modern Data Science With R: With Digital Download*. Taylor & Francis. Retrieved from `https://books.google.com/books?id=Gv1nvgAACAAJ`

Bojanowski, M. (2015). *Intergraph: Coercion routines for network data objects*. Retrieved from `http://mbojan.github.io/intergraph`

Butts, C. T. (2008). Network: A package for managing relational data in r. *Journal of Statistical Software*, *24*(2). Retrieved from `http://www.jstatsoft.org/v24/i02/paper`

Butts, C. T. (2015). *Network: Classes for relational data*. The Statnet Project (`http://statnet.org`). Retrieved from `http://CRAN.R-project.org/package=network`

Butts, C. T., & others. (2008). Social network analysis with sna. *Journal of Statistical Software*, *24*(6), 1–51.

Butts, C., Hunter, D., Handcock, M. S., Morris, M., Krivtisky, P. N., Almquist, Z., ... Bender de-Moll, S. (2015, June). Introduction to Exponential-family Random Graph (ERG or p*) modeling with ergm. Retrieved from `https://statnet.org/trac/raw-attachment/wiki/Sunbelt2015/ergm_tutorial.pdf`

Csárdi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, *Complex Systems*, 1695. Retrieved from `http://igraph.org`

Erdős, P., & Rényi, A. (1959). On random graphs, i. *Publicationes Mathematicae (Debrecen)*, *6*, 290–297.

Erdős, P., & Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, *5*(1), 17–60.

Erdős, P., & Rényi, A. (1964). On the strength of connectedness of a random graph. *Acta Mathematica Academiae Scientiarum Hungarica*, *12*(1-2), 261–267.

Gilbert, E. N. (1959). Random graphs. *The Annals of Mathematical Statistics*, *30*(4), 1141–1144.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., & Morris, M.

(2008). Statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*, *24*(1), 1–11. Retrieved from http://www.jstatsoft.org/v24/i01

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., & Morris, M. (2017). *Ergm: Fit, simulate and diagnose exponential-family models for networks*. The Statnet Project (http://www.statnet.org). Retrieved from https://CRAN.R-project.org/package=ergm

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., Bender-deMoll, S., & Morris, M. (2016). *Statnet: Software tools for the statistical analysis of network data*. The Statnet Project (http://www.statnet.org). Retrieved from CRAN.R-project.org/package=statnet

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., Morris, & Martina. (2008). Ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, *24*(3), 1–29.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. Springer Science & Business Media.

Lazega, E., & Pattison, P. E. (1999). Multiplexity, generalized exchange and cooperation in organizations: A case study. *Social Networks*, *21*(1), 67–90.

Leskovec, J., & Krevl, A. (2014, June). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Leskovec, J., Chakrabarti, D., Kleinberg, J., & Faloutsos, C. (2005). Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication. In *Knowledge Discovery in Databases: PKDD 2005* (pp. 133–145). Springer, Berlin, Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/11564126_17

Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., & Ghahramani, Z. (2010). Kronecker Graphs: An Approach to Modeling Networks. *Journal of Machine Learning Research*, *11*(Feb), 985–1042. Retrieved from http://www.jmlr.org/papers/v11/leskovec10a.html

Newman, M. (2010). *Networks: An Introduction*. OUP Oxford.

Toivonen, R., Kovanen, L., Kivelä, M., Onnela, J.-P., Saramäki, J., & Kaski, K. (2009). A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks*, *31*(4), 240–254. http://doi.org/10.1016/j.socnet.2009.06.004

Travers, J., & Milgram, S. (1967). The small world problem. *Phychology Today*, *1*, 61–67.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of "small-world" networks.

*Nature*, *393*(6684), 440–442. http://doi.org/10.1038/30918

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, *33*(4), 452–473. http://doi.org/10.1086/jar.33.4.3629752