University of California
Los Angeles

A Comparison of Estimators for Respondent-Driven Sampling

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Margaret Meek Lange

2014

Abstract of the Thesis

A Comparison of Estimators for Respondent-Driven Sampling

by

Margaret Meek Lange

Master of Science in Statistics

University of California, Los Angeles, 2014

Professor Mark Handcock, Chair

Respondent-driven sampling, or RDS, is used to draw samples from hard-to-reach or marginalized populations and to make inferences about the populations based on the samples. Such sampling begins with an initial, or "seed," sample from the population of interest. It then exploits the networked structure of those populations, relying on the members themselves to recruit further members of the population for the sample. A number of estimators for RDS have already been developed. Each estimator is motivated by a model that makes a number of assumptions about seed selection, respondent behavior, and certain properties of the underlying social network itself. In a series of articles, Gile and Handcock have used the `statnet` package in R to simulate respondent-driven sampling under a variety of conditions. They then use these simulations to examine the bias and variance of different estimators when assumptions are not perfectly or at all fulfilled.

The goal of this project is twofold. First, the original R code to simulate respondent-driven sampling is quite slow. In the `statnet` package RDSdevelopment, we have written C code to duplicate and extend the functionality of the R code. The new code is considerably faster than the old R code. Second, we examine the sensitivity of current estimators to a previously unstudied aspect of respondent behavior: how accurately respondents report their number of contacts in the network, also known as their "degree." The two networks we consider are a previously simulated network, fauxmadrona, and the Project 90 network. The latter is constructed from an actual population of heterosexuals at high-risk for HIV living in Colorado Springs, CO, in the late 1980s and early 1990s. We carry out simulations on both networks using `statnet` in which we simulate imperfect recall of degree. Under conditions of imperfect recall, estimators tend to increase in variance as recall erodes. There is also change in bias, though the direction of change varies from estimator to estimator and is not monotonic with the increase in recall error.

Finally, we introduce a variation on the current estimators by replacing reported degree by reported rank of degree in each estimator formula. Rank of degree is calculated by a carrying out a rank transformation on the reported degree distribution of each sample of individuals from the networked population. A rank transformation of a set of numbers maps each member of a set onto its rank with respect to the other members of the set. A number of rank transformations are possible. The rank transformation known as "standard competition" degrades estimator performance on the fauxmadrona and the Project 90 network. The dense rank transformation leaves estimator performance mostly unchanged, at least on the networks treated by the thesis.

The thesis of Margaret Meek Lange is approved.

Qing Zhou

Jan de Leeuw

Mark Handcock, Committee Chair

University of California, Los Angeles

2014

To my parents

Table of Contents

1 Introduction **1**

2 Review of Current Literature on RDS **4**

2.1 Introduction to Social Networks 4

2.2 Existing Estimators for Respondent-driven Sampling 6

2.3 Recall Error and Rank Estimators 10

3 The Old and New Code for RDS **13**

3.1 The Original `statnet` Package 13

3.2 Revisions to the Code 14

4 Results **17**

4.1 Timing 17

4.2 Simulation Results 21

5 Analysis of Project 90 . . . . **27**

5.1 Project 90: An Overview 27

5.2 Results 28

6 Conclusion **47**

7 Appendix: Recall Error and Rank Transformations with Probability Sampling **49**

8 A Final Note on the Code **59**

References **60**

List of Figures

4.1 18

4.2 N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486 20

4.3 N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486 22

4.4 N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486 23

4.5 N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486; Dense transformation in the middle and SC transformation on the right. 25

5.1 29

5.2 N (population) = 3011; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.13318; population relative activity = 1.152845 31

5.3 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 2.641745; population relative activity = 0.9742704 32

5.4 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.2042409; population relative activity = 0.8379733 33

5.5 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.253184; population relative activity = 1.114024 34

5.6 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.255978; population relative activity = 1.176488 35

5.7 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.215924; population relative activity = 1.074868 36

5.8 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.023402; population relative activity = 1.096503 37

5.9 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.107955; population relative activity = 1.212837 38

5.10 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.032841; population relative activity = 0.9633545 39

5.11 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.068735; population relative activity = 0.9702125 40

5.12 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.024186; population relative activity = 1.115212 41

5.13 N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.033291; population relative activity = 1.252161 42

7.1 Degree Distribution of first sample at different levels of recall error. No rank transformation on degrees. Samples drawn with replacement. 52

7.2 Relative Degree Distributions comparing Disease = 1 and Disease = 0 in first sample at different levels of recall error. No rank transformation on degrees. Samples drawn with replacement. 53

7.3 Distribution of μ for different levels of recall error. No rank transformation on degrees. Samples drawn with replacement. 54

7.4 Degree Distribution of first sample at different levels of recall error. Standard competition rank transformation on degrees. Samples drawn with replacement. 55

7.5 Distribution of μ for different levels of recall error. Standard competition rank transformation on degrees. Samples drawn with replacement. 56

7.6 Degree Distribution of first sample at different levels of recall error. Dense rank transformation on degrees. Samples drawn with replacement. 57

7.7 Distribution of μ for different levels of recall error. Dense rank transformation on degrees. Samples drawn with replacement. . . . 58

List of Tables

2.1 SC = Standard Competition, MC = Modified Competition, D = Dense, O = Ordinal, F = Fractional 11

4.1 Comparison of R and C Code 19

4.2 Scaled MSE for Fauxmadrona Disease, Perfect Recall 21

4.3 Scaled MSE for V-H estimator on Fauxmadrona 21

4.4 Scaled MSE for SS estimator on Fauxmadrona 24

4.5 Scaled MSE for Estimator SS on Fauxmadrona 26

5.1 Scaled MSE for Mean, S-H, V-H, and SS Estimators on Project 90 43

5.2 Scaled MSE for SS Estimator on Project 90 44

5.3 Scaled MSE for S-H Estimator on Project 90 44

5.4 Scaled MSE for Rank SS Estimator on Project 90 45

5.5 Scaled MSE for Rank S-H Estimator on Project 90 45

5.6 Scaled MSE for V-H, SS, S-H, and Rank (S-S, dense) at 50 percent recall 46

5.7 Scaled MSE for V-H, SS, S-H, and Rank (S-H, dense) at 50 percent recall 46

7.1 Scaled Bias, Variance, and Mean-squared Error of Estimator on Samples drawn with replacement 51

Acknowledgments

Thanks to my committee chair, Mark Handcock, for helping me out with this project from start to finish. He was patient with me as I worked to finish the thesis while employed and living in Australia. His contributions included everything from helping me debug my C code to clarifying complicated statistical concepts for me. Thanks also to my other committee members, Jan de Leeuw and Qing Zhou, for reading my thesis and commenting on it. Thanks to the department administrator, Glenda Jones, for helping me file my thesis after being gone from the department for two years. Finally, thanks to my parents. My father, Kenneth Lange, introduced me to the UCLA Statistics department and engaged me in discussions about statistics throughout the duration of the masters program. My mother, Eugenia Lange, edited the final draft of the manuscript and helped me to improve it.

CHAPTER 1

Introduction

Sampling consists of gathering information about a subset of a population and making inferences about the population based on that information. Probability sampling is the most common variety of sampling [9]. It consists of choosing random individuals from the population of interest from an enumeration of its members, or sampling frame [14]. The exact way the researcher goes about this task is known as the sampling method. The probability that each member will be included in the sample is known to the researcher. Each member of the sample can be regarded as the realization of a random variable. Statistics associated with the sample, such as the mean or total, can likewise be regarded as random variables. Features of the distribution of key sampling statistics are well established. The sample mean is an unbiased estimate of the population mean, while the variance of the sample mean can be estimated using information from the sample as well [9].

However, when no sampling frame is available, and the probability of inclusion of each population member is unknown, the situation is different and more difficult. The most common sampling methods may not work and the choice of estimator is not an obvious one. In this paper, we focus on the case of stigmatized or vulnerable populations, which raise just these issues. For instance, one might want to take a sample to estimate the proportion of sex workers in Phnom Penh, Cambodia, who are HIV-positive. Such stigmatized populations make it difficult, both empirically and ethically, to establish a sampling frame. A directory of the population is unlikely to exist and compiling directories for the purpose of research may risk revealing sensitive information to the public.

Respondent-driven sampling (RDS) is a sampling method designed for such situations. It exploits the fact that members of stigmatized populations are often linked together in a social network. In RDS, an initial, or “seed,” sample from the desired population is recruited. After answering the questions of interest to the researcher, each individual in the seed sample is given one or more coupons. The point then is for each respondent to recruit as many additional members of the population as coupons. These new individuals in turn report to the researchers, answer the questions of interest, and are themselves given coupons. This cycle is repeated until the researchers have a sample of the desired size [3]. This allows researchers to gather a sample in the absence of a sampling frame. When one member of the stigmatized population recruits another for research, the researcher only meets the recruit if he or she chooses to participate in the research project. In this way, confidentiality issues are also mitigated.

Although the method of RDS solves the problem of how to collect a sample, it does not immediately tell us what estimator to use. The standard results, mentioned above, do not hold in this case, because the probability of inclusion of each individual in the sample is not known. Proposed estimators have built on models of RDS that make a number of simplifying assumptions. The literature on RDS has then employed computer simulations to examine the properties of such proposed estimators, especially when those assumption do not exactly hold [3, 2, 4]. This paper will take up this project.

Our paper has two aims. The first is to describe the code that we have written to speed up the simulation of RDS. Speeding up the code was useful to our own research. Moreover, the code is incorporated in the open-source R package RDS-development and is thus accessible to future researchers. Our second goal is to use the new code to examine departures from one particular simplifying assumption that has yet to be investigated. This assumption is that individuals accurately remember their “degree,” which is the number of total acquaintances they have that also belong to the population in question. We would like to examine the performance of existing and modified estimators in the presence of recall error.

The next chapter discusses the concept of a social network in more detail. It then builds on that theoretical foundation to describe existing estimators for RDS and their properties. Chapter 3 describes the new code, while chapter 4 discusses the effect of recall error on existing and modified estimators. Chapter 5 tries out the simulations on a real-world social network, and chapter 6 summarizes the paper’s findings. Finally, an appendix to the thesis discusses the problem of recall error in the simplified setting of probability sampling.

CHAPTER 2

Review of Current Literature on RDS

2.1 Introduction to Social Networks

The estimators for RDS take advantage of the fact that many vulnerable populations are examples of social networks. To understand these estimators, it is first necessary to understand the social network concept and its terminology. A social network is a group of related actors. Two actors who stand in a relationship to one another are called a dyad or alters. A network can be undirected or directed. If it is undirected, all relationships are reciprocated; if it is directed, actor A can be related to actor B without actor B being related to actor A. The number of alters possessed by a given actor is known as its degree. Variables can be defined at the level of relationships and at the level of individual actors [16]. In a friendship network of schoolchildren, the duration of a particular friendship could be a dyad-level variable. In contrast, the gender of an individual child would be an actor attribute variable.

A visual depiction of a social network represents actors as nodes and relations as links between the nodes. Mathematically, a social network can be represented as a graph or as a sociomatrix [3]. In this paper we will use sociomatrix rather than graph notation. The network on N actors is represented by a $N \times N$ matrix y such that $y_{ij} = 1$ if j and i are linked, and $y_{ij} = 0$ otherwise. We also use z to refer to an actor attribute variable of interest, with z_i the value of that variable for the i^{th} actor. In the cases this paper considers, z is dichotomous and takes on the value of 0 or 1. In that case, the population mean for z, μ, is called the prevalence:

$$\mu \;=\; \frac{1}{N}\sum_{i}^{N} z_i.$$

Following Gile and Handcock [3], we will refer to the generic $z_i = 1$ as "infected" and $z_i = 0$ as "uninfected." We will use d_i to refer to the nodal degree of the i^{th} actor. We can also represent the nodal degree distribution visually as a histogram.

Some network statistics reflect only patterns of relationships, or only relationships and dyad-level variables. However, in this paper, we will consider statistics that take into account relationships and actor attribute variables. In particular, we will need to refer to homophily and relative activity. A network is said to display homophily when nodes with the same level of a key variable are more likely to be joined than nodes with different levels of that variable. For instance, a network of injecting drug users would display homophily if two drug users who are HIV positive are more likely to know each other than an injecting drug user who is HIV positive and one who is HIV negative. In this case, homophily might be explained by the fact that injecting drug users who know each other are more likely to have shared needles and to have infected each other with the HIV virus.

Gile and Handcock [3, 4] operationalize homophily, R, as the "relative probability of an edge between two infected nodes, and an edge between an infected and an uninfected node." More formally,

$$R \;=\; \frac{\frac{2}{N^1(N^1-1)}\sum_{ij} z_i z_j y_{ij}}{\frac{1}{N^1 N^0}\sum_{ij} z_i(1-z_j)y_{ij}}.$$

Here $N^0 = N(1-\mu)$ is the number of uninfected nodes, and $N^1 = N\mu$ is the number of infected nodes. As stated earlier, z_i represents the infection status of the i^{th} actor. Finally, y_{ij} is 1 if i is connected to j and 0 otherwise.

In turn, the concept of relative activity, ω, captures the fact that nodes with one level of a key variable may have a higher degree than nodes with another level [3]. For instance, in a network of inhabitants of Los Angeles, people who are employed may have more friends, family, and acquaintances than people who are unemployed. The formula [4] for ω is

$$\omega = \frac{\sum_{i=1}^{N} d_i z_i}{\sum_{i=1}^{N} z_i} \frac{\sum_{i=1}^{N}(1-z_i)}{\sum_{i=1}^{N} d_i(1-z_i)}.$$

Here z_i refers again to the infection status of the i^{th} node and d_i refers to that node's degree.

2.2 Existing Estimators for Respondent-driven Sampling

As mentioned in Chapter 1, respondent-driven sampling, or RDS, exploits the networked structure of certain hard-to-reach populations to gather samples from them. The literature on RDS has generated a number of estimators. Generally speaking, the method is to use the sample gathered to arrive at estimates of network characteristics, and then to use those estimates in turn to estimate the population characteristic under study. The first estimator, the Salganik-Heckathorn (S-H) estimator, is based on a model developed in 2004 [12] that employs this method.

Consider a dichotomous variable of interest, such as the generic "infection." The networked population can be divided into infected nodes (group A) and uninfected nodes (group B). Now, let D_A and D_B refer to the average degrees of members of groups A and B respectively. Next, let $C_{A,B}$ refer to the probability that following a random link beginning with a person in group A will end in group B. Likewise, let $C_{B,A}$ refer to the probability that following a random link beginning with a person in group B will end in group A. Finally, use $PP_A = \mu_A$ to refer to proportion of the population that is in group A, or infected. Assuming that that network ties are reciprocated, algebraic manipulation leads to the following result:

$$\mu_A \quad = \quad \frac{D_B \cdot C_{B,A}}{D_A \cdot C_{A,B} + D_B \cdot C_{B,A}}.$$

Salganik and Heckathorn devise separate estimators for each component of the right-hand side. These estimators use information from the sample. Then, given estimators for each component of μ_A, Salganik and Heckathorn simply assume

$$\hat{\mu}_A \quad = \quad \frac{\widehat{D}_B \cdot \widehat{C}_{B,A}}{\widehat{D}_A \cdot \widehat{C}_{A,B} + \widehat{D}_B \cdot \widehat{C}_{B,A}}.$$

A newer estimator, the Volz-Heckathorn (V-H) estimator [15], builds on a model of RDS as a Markov chain to estimate the mean trait of interest. A Markov chain is defined on a finite number of possible states. The chain moves to a new state at each epoch or generation [7]. Transitions between the states are described by a matrix $P = (p_{ij})$. The transition matrix gives the probability of moving from state i to state j, that is, $\Pr(Z_n = j \mid Z_{n-1} = 1)$, where n is an epoch. The matrix is the same for every transition. Moreover, the process is "memoryless":

$$\Pr(Z_n = i_n \mid Z_{n-1} = i_{n-1}, ..., Z_0 = i_0) \quad = \quad \Pr(Z_n = i_n \mid Z_{n-1} = i_{n-1}).$$

The probability $p_{ij}^{(n)} = \Pr(Z_n = j \mid Z_0 = i)$ is equal to the ith row and jth column of the matrix P^n [7]. Markov chains with the property of ergodicity converge. Thus, $\lim_{n\to\infty} P^n = P^\infty$, where P^∞ is a matrix whose rows are identical. In other words, after enough transitions between states have occurred, the probability of the next state no longer depends on the starting point [7].

Modelling RDS as a convergent Markov chain on a social network requires a number of simplifying assumptions [3]. First, the model assumes sampling is with replacement, while in reality it is without replacement. That is, in practice, once an individual is chosen for the sample, he or she cannot be chosen a second time.

Second, the model assumes that gathering the sample requires a large number of waves, while in reality a few waves may be sufficient to collect the target number of individuals. Third, the model assumes that an individual chooses randomly among alters when giving out a coupon or coupons. The alternative to random referral is biased referral, in which individuals are more likely to hand out coupons to alters either with or without the property of interest. Finally, the model assumes that reported degree is a good measure of actual degree. Based on these assumptions, an estimator of the mean trait value is the following:

$$\hat{\mu}_{\text{V-H}} = \frac{\sum_{i=1}^{N} S_i \frac{z_i}{\tilde{d}_i}}{\sum_{i=1}^{N} S_i \frac{1}{\tilde{d}_i}}.$$

Here S_i is 1 if unit i is sampled and is 0 otherwise; z_i is the trait of interest measured on the sample individual i; and $\tilde{d}_i$ is the self-reported nodal degree used to approximate the true nodal degree d_i.

Gile and Handcock observe that $\hat{\mu}_{\text{V-H}}$ can be regarded as the ratio of two Hansen-Hurwitz estimators. In general, the Hansen-Hurwitz estimator assumes that sampling is with replacement and that on each draw the probability of selecting the i^{th} unit of a population with N members is p_i. Then, for a sample of size n, the Hansen-Hurwitz estimator of the total of a variable y is

$$\hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} \frac{y_i}{p_i}.$$

Note that the Hansen-Hurwitz estimator assumes that sampling is with replacement and thus relies on one of the simplifying assumptions mentioned above.

Gile and Handcock use simulations and boxplots to examine the sensitivity of the V-H estimator to the above assumptions. First, they examine the effect of biased seed selection on the performance of the V-H estimator. They find that when seed selection is biased, estimator bias increases as the number of waves of sampling decreases. Under conditions of biased seed selection, high homophily (R) also results in a biased estimator with a greater variance. Finally, even when seed selection is random, high homophily in the networked population results in higher estimator variance [3].

To test the sensitivity of the estimator to the assumption that the sample is with replacement, Gile and Handcock vary the sampling fraction, namely the fraction of the total population represented by the sample. For instance, if the sample has 50 individuals and the total population is size 500, then the sampling fraction is 0.10. The larger the sampling fraction, the less accurate the assumption that sampling is with replacement. They find that when relative activity (ω) is exactly 1, bias does not increase with the sampling fraction. However, when relative activity moves away from 1, the bias of the V-H estimator increases with the sampling fraction.

Finally, Gile and Handcock use simulations to establish that the V-H estimator outperforms the S-H estimator under a variety of levels of population homophily, population relative activity, sampling fractions, and numbers of waves of sampling [3]. Performance is measured by comparing the mean-squared error of the two estimators, which is the variance of the estimator plus the bias squared. Based on these findings, they conclude the V-H estimator is better than the S-H estimator overall.

A later estimator, the successive sampling or SS estimator, seeks to address some of the problems with the V-H estimator identified by the 2009 article. The SS estimator drops the assumption that RDS is without replacement. Its formula is the following:

$$\hat{\mu}_{SS} = \frac{\sum_{j=1}^{N} \frac{\mathbf{S}_j \mathbf{z}_j}{\hat{\pi}(\mathbf{d}_j)}}{\sum_{j=1}^{N} \frac{\mathbf{S}_j}{\hat{\pi}(\mathbf{d}_j)}}.$$

The function π takes a nodal degree and maps it onto the inclusion probability of the node in the sample. The function is estimated iteratively using an algorithm based on a model of RDS as a “self-avoiding walk,” and therefore a without-replacement process [2]. Gile uses simulations to show that the SS estimator substantively out-performs the V-H estimator under conditions of a large sampling fraction combined with relative activity not equal to 1. Moreover, the SS estimator is not substantively worse than the V-H estimator under conditions of high homophily and few sampling waves [2].

2.3 Recall Error and Rank Estimators

We have now reviewed treatments of the sensitivity of the V-H estimator to a number of its underlying assumptions. However, the assumption that reported degree is a good measure of actual degree has not yet been considered. This thesis will examine the performance of various estimators when that assumption no longer perfectly holds. We will use a modified version of the binomial distribution [9] to model recall error. The binomial distribution, as is well known, describes the probable number of successes over a finite number of independent Bernoulli trials, each of which has a fixed probability of success or failure. In this case, a trial consists of recalling a particular acquaintance, and the distribution describes the total number of acquaintances remembered. The distribution is modified so that a total of 0 successes is impossible; both 1 and 0 from a conventional binomial distribution are coded as 1. This modification is motivated by the fact that an individual would be weighted 0 if he or she reported no acquaintances.

$$\Pr(x = k) \quad = \quad \begin{cases} (1-p)^n + \binom{n}{k} p^k (1-p)^{n-k} & \text{for } k = 1 \\ \binom{n}{k} p^k (1-p)^{n-k} & \text{for } 2 \leq k \leq n, \end{cases}$$

where n is the nodal degree and p is the probability of recall of any one acquaintance. Note that this representation of recall error means that the reported degree is always less than or equal to actual degree. We conjecture that the performance of the estimators will degrade with decreasing p. We will examine this conjecture using simulations.

Our second investigation will begin by introducing a rank transformation on the recalled degrees. Consider the distribution of reported degrees associated with the individuals in the sample. A rank transformation replaces each degree with its rank with respect to all other members of the sample distribution of reported degrees. The transformed sample degree distribution is then used to compute one of the existing estimators. The hope is that, under conditions of poor recall, the rank estimators will result in a reduced mean-squared error.

A number of options exist for ranking sets of numbers. These include two kinds of competition-style ranking, dense ranking, ordinal ranking, and fractional ranking [17]. Table 2.1 shows how each method of ranking transforms four scores, the middle terms of which are equal. The definitions for each form of ranking are as follows. (Here a "higher" rank refers to a lower number, so that a rank of 1 is the highest possible.) Standard competition ranking gives each score a rank of 1 plus the number of items ranked higher. Equal scores get equal ranks. Modified competition ranking gives each score a rank equal to the number of items equal to or above it. This scheme also gives equal scores an equal rank. Dense ranking gives each score a rank of 1 plus the number of distinct scores above it. Again, equal scores receive an equal rank. Ordinal ranking gives each score a distinct

Table 2.1: SC = Standard Competition, MC = Modified Competition, D = Dense, O = Ordinal, F = Fractional

Degree	7	8	8	9
Rank (SC)	1	2	2	4
Rank (MC)	1	3	3	4
Rank (D)	1	2	2	3
Rank (O)	1	2	3	4
Rank (F)	1	2.5	2.5	4

ordinal number. The scheme does not give equal scores an equal rank. Ranks can be assigned to equal scores randomly or according to some rule that depends on an aspect of the score other than its value. Fractional ranking gives each score a rank of 1 plus the number of items above it plus half the number of items equal to it. This scheme, again, gives equal scores equal values.

A rank estimator is a modified version of one of the other estimators after a rank transformation has been performed. For example, $\tilde{r}_i = \text{Rank}(\tilde{d}_i)$ is the rank of a self-reported nodal degree and is used to approximate the rank of the true nodal degree d_i. A rank version of the V-H estimator is

$$\hat{\mu}_{\text{Rank}} = \frac{\sum_{i=1}^{N} S_i \frac{z_i}{\tilde{r}_i}}{\sum_{i=1}^{N} S_i \frac{1}{\tilde{r}_i}}.$$

In every case in which a rank estimator is introduced, both the original RDS estimator and the rank transformation will be specified.

CHAPTER 3

The Old and New Code for RDS

3.1 The Original `statnet` Package

Code to simulate RDS and to calculate relevant estimators already existed at the outset of this project. It is worth describing before moving to a discussion of our modifications of that code. This original code is located in the R package RDSdevelopment, which is part of the `statnet` project [6]. RDSdevelopment requires and builds on earlier R packages from the same project, namely ergm [5] and network [1]. The original RDSdevelopment algorithm for RDS is coded entirely in R.

The main functions for this algorithm are included in the file SimulateRDS.R. They are create.rds.sampler and do.simulation. The function create.rds.sampler defines a number of subsidiary functions relevant to RDS and static variables local to the R environment of these functions. Then create.rds.sampler returns the subsidiary functions in a list. From an object-oriented programming perspective, create.rds.sampler acts like an object constructor for a class of specialized RDS simulation objects. Like a constructor object, create.rds.sampler takes a number of important parameters that together define the simulation instance. These parameters are: (1) the network from which to sample; (2) the number of individuals in the seed sample; (3) the size of the sample to gather; (4) the number of coupons each individual will have; (5) the distribution from which to draw the seeds; (6) a boolean variable indicating whether sampling will be done without replacement;

(7) a variable indicating the probability with which any coupon is returned; (8) a boolean variable indicating whether more seeds can be recruited if the available connections have been exhausted; and (9) three variables coding for the amount of referral bias.

The function do.simulation takes two arguments: a list object returned by create.rds.sampler and the number of samples to be simulated. The result returned by this R function is a RDSDataFrame object, which can serve as an argument to the constructor functions for estimator objects in RDSdevelopment. The S-H, V-H, and SS estimators are all coded as object classes. The rank estimators are calculated by a function named Rank.estimates in a R file of the same name. Rank.estimates is currently a wrapper function that takes parameters to indicate which of the rank transformations and which of the three other estimators (V-H, S-H, or SS) to use.

3.2 Revisions to the Code

In writing R packages, it is common practice to use the C language to program the most computationally demanding portions of the code. The R language includes a number of special functions that can act as an interface between the C code and the parts of a R package written in R [13]. The ergm and network packages already take advantage of this functionality; a large percentage of their code is written in C [5, 1]. This section of my thesis outlines how the R code to simulate RDS was converted into C code that duplicates and extends the original R code.

The C code is included in a new version of the package RDSdevelopment, in the files RDSSample.c and RDSSample.h. The highest level function is called CRDSSample. A R wrapper function, rdssampleC, calls CRDSSample and returns an R object of type RDSdataframe to be analyzed. This wrapper function is included in the file rdssamplecodeC.R. The three files containing these functions have been added to appropriate sub-folders of the RDSdevelopment package and the package is ready to install.

The new code allows the user to specify the following parameters: (1) the network from which to sample; (2) the size of seed sample; (3) the target sample size; (4) the number of coupons per individual; and (5) the distribution from which to draw the seeds. It is assumed that sampling will be without replacement and that new seeds will be recruited if connections are exhausted. The new code also allows for a number of samples to be drawn at a time. The new code does not, at this point, allow for modeling biased referral, though presumably that functionality could be added.

Unlike the pre-existing R code, the new C code does not include a single variable to indicate the probability a coupon will be returned. Instead, the C code includes a more sophisticated model of the time dimension of the RDS process. The most important function from the perspective of the simulation is TicketEvent, which is called from CRDSSample. There are two kinds of ticket events, each of which is represented by a separate function called from TicketEvent. One event occurs when an individual A in the population recruits a second individual B by passing a coupon. This event is represented by the function RecruitOne. The second event occurs when an individual B who has been recruited returns to the center, fills out a survey, and is given coupons in turn. The event is represented by the function CompleteSurvey.

Each event includes the scheduling of the next associated event or events. For instance, CompleteSurvey searches for all of the neighbors of the individual and assigns each individual-alter dyad a time in the future when the individual will attempt to recruit the alter. The time is assigned by adding the present time and a number drawn from the exponential distribution

$$f(x \mid \lambda) \;=\; \begin{cases} \lambda e^{-\lambda x} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0. \end{cases}$$

Draws from the exponential distribution are made using the C function exp_rand. This function is provided by the Rmath.c library, which calls the R function dexp and sets $\lambda = 1$. In RecruitOne, a similar procedure is followed. In this case, again using the present time and a number drawn from the exponential distribution, a time is assigned when the recruited individual will arrive at the center to take the survey. This innovation, as we shall see, also complicates the R code's assumption that the order of sampling will follow the waves of respondents exactly. This method allows for the possibility that an individual from wave 2 will arrive at the center before an individual from wave 1 has given away his coupons. This model is truer to life than the original R model.

A final R function, called checkSample, was written to check that the RDS samples generated by the C code met certain basic requirements. These included making sure no one was recruited more than once, no recruiter used more than the allocated number of coupons, and that everyone was recruited by someone recruited earlier in the sample or by a member of a seed sample.

In the next section of the thesis, we show the time gained by replacing the original R code with the C code add-in. We then describe the simulations used to test the sensitivity of different estimators to less than perfect recall of degree.

CHAPTER 4

Results

4.1 Timing

First, we compared the original R code with the new C implementation of respondent-driven sampling. To do so, we performed the following test on two separate computers, both Apple Macintoches. For each test, we performed 1000 RDS simulations using the old code and 1000 identical simulations using the new code. Each sample consisted of 500 individuals. The seed sample consisted of 10 individuals drawn proportional to degree. The number of coupons was 2. The samples were drawn from the artifically-generated undirected network fauxmadrona, an example provided by the R package RDSdevelopment.

The fauxmadrona network consists of 1000 individuals. The actor attribute variable of interest is called **disease**. Twenty percent of individuals are infected; the rest are not. The population homophily for the **disease** variable is 1.52931 and the population relative activity is 1.759489. The degree distribution of the fauxmadrona network is depicted by a histogram in Figure 4.1. The color-coding of the histogram according to disease level gives a sense of relative activity. The mean degree is 7.172 and the standard deviation is 3.209. Table 4.1 shows the results of the two tests on this network. By replacing the R code with C code, we achieved a 73% decrease in simulation time on the first computer and a 62% decrease on the second.

The second round of simulated samples was used to compare estimators. Fig-

Figure 4.1:

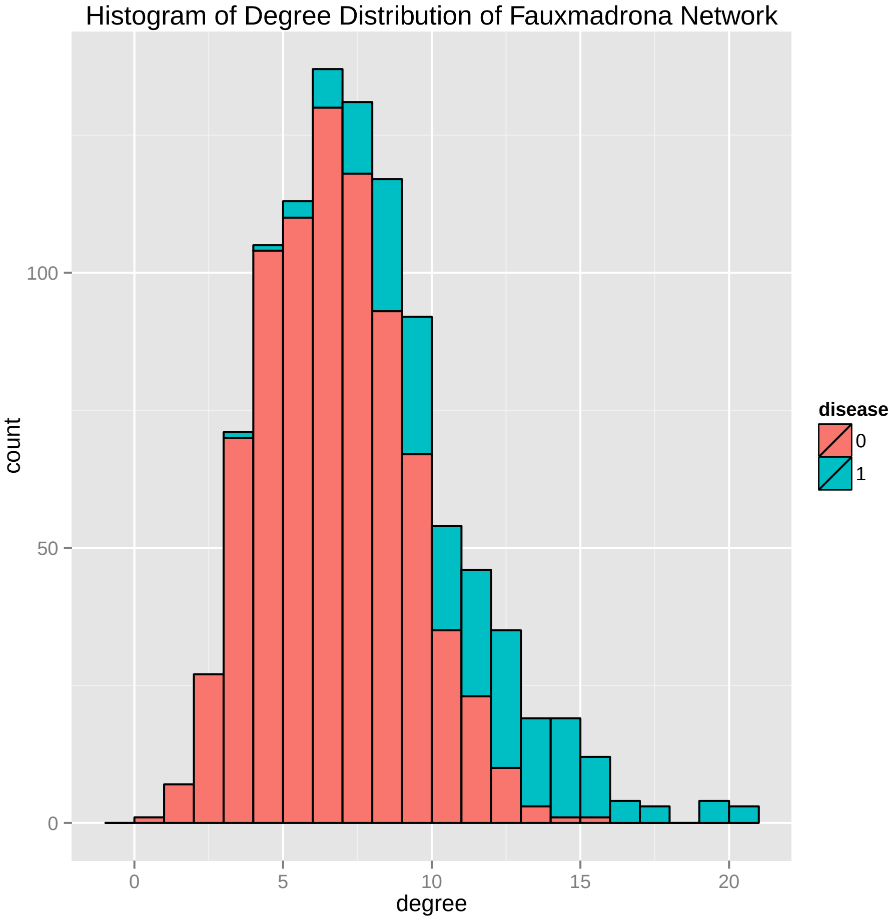

Histogram of Degree Distribution of Fauxmadrona Network
count
100
50
0
0
5
10
15
20
degree
disease
0
1

Table 4.1: Comparison of R and C Code

	Mean Time per sample (1)	Mean Time per sample (2)
R Code	0.2573 sec	0.2815
C Code	0.0699 sec	0.1081

ure 4.2 shows boxplots of the results. The variable estimated is **disease**. The labels Mean, S-H, V-H, and SS indicate which estimator was used. The number following the label indicates the percentage recall, in this case always 100. The letter “R” indicates that the old code was used to simulate RDS, and the letter “C” means that the new code was used. The true **disease** mean of 0.20 is indicated in red.

First of all, note that the R and C code produce the same ranking of estimators. In each case, the sample mean is the most biased, followed by the S-H mean, the V-H mean, and the SS mean. Since the sampling fraction is large (0.50) and the relative activity of **disease** is considerably greater than 1, it is not surprising that the SS estimator is the best performer, and the fact that the R and C results are similar is encouraging. Table 4.2 presents an evaluation of the estimators using samples from the C code only. The assessment is made using the root mean-squared error multiplied by 100. The formula for the mean-squared error of the V-H estimator is

$$\text{MSE} = \frac{\sum_{i=1}^{1000} (\overline{\mu}_{\text{V-H},i} - \mu)^2}{1000},$$

where $\overline{\mu}_{\text{V-H},i}$ is the V-H estimator applied to the ith sample and μ is the true population mean. These results are scaled by taking the square root and multiplying it by 100.

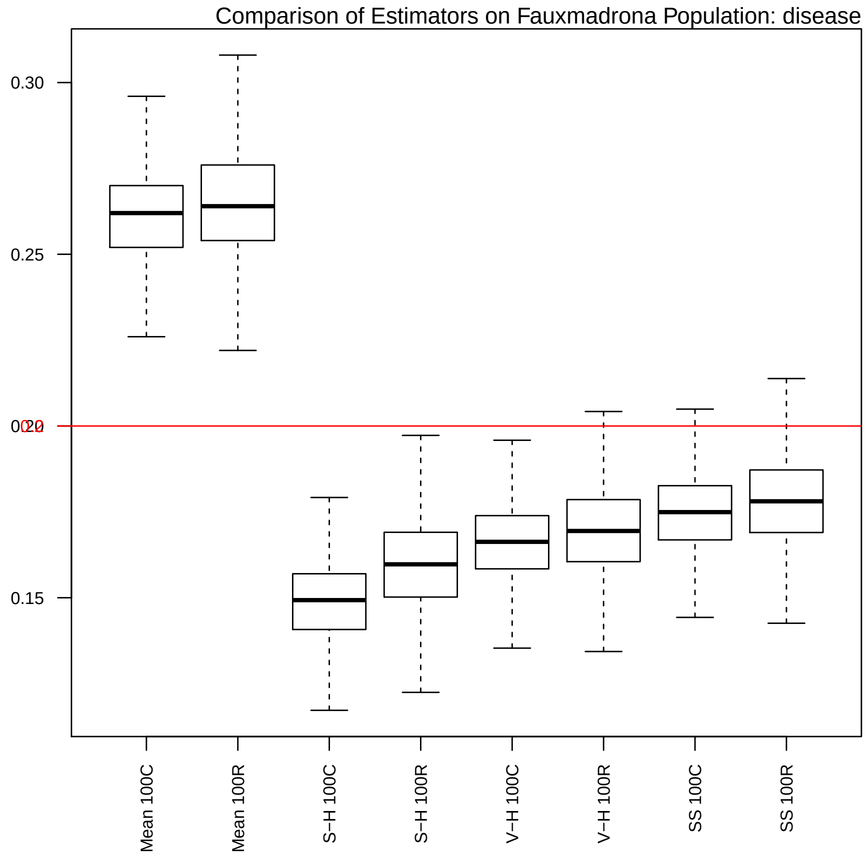


Figure 4.2: N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486

Table 4.2: Scaled MSE for Fauxmadrona Disease, Perfect Recall

	Mean 100	V-H 100	S-H 100	SS 100
disease	6.28	3.57	5.24	2.79

4.2 Simulation Results

Next, we used the new simulation code and the fauxmadrona network to examine the performance of the V-H and SS estimators under conditions of imperfect recall of degree. To compare the various estimators on the fauxmadrona network, we simulated multiple sets of 1000 samples each. The first set of samples, with 100 percent recall, is the same set as before. The second set of samples have 90 percent recall; the third 80 percent recall; the fourth 70 percent; the fifth 60 percent recall; and the sixth 50 percent recall. Recall error is modeled using the modified binomial distribution discussed in Chapter 2. All samples are of size 500. Each member of the sample was drawn from the fauxmadrona population. For each sample, 10 seeds were chosen, drawn proportional to the true degree. Again, each individual received two coupons.

Figure 4.3 shows boxplots for the SS estimators and Table 4.3 shows the corresponding MSE values. Figure 4.4 shows boxplots for the V-H estimators and Table 4.4 shows the corresponding MSE values. In both cases, the boxplots show that the bias of the estimator increases as recall error moves from 100 to 60 percent, and then begins to decrease again at 50 percent. The variance appears to be increasing slightly monotonically. These trends are replicated under a probability sampling scenario discussed in the appendix to chapter four, and thus are not specific to RDS.

Table 4.3: Scaled MSE for V-H estimator on Fauxmadrona

	V-H 100	V-H 90	V-H 80	V-H 70	V-H 60	V-H 50
disease	3.57	3.76	3.97	4.07	4.23	3.91

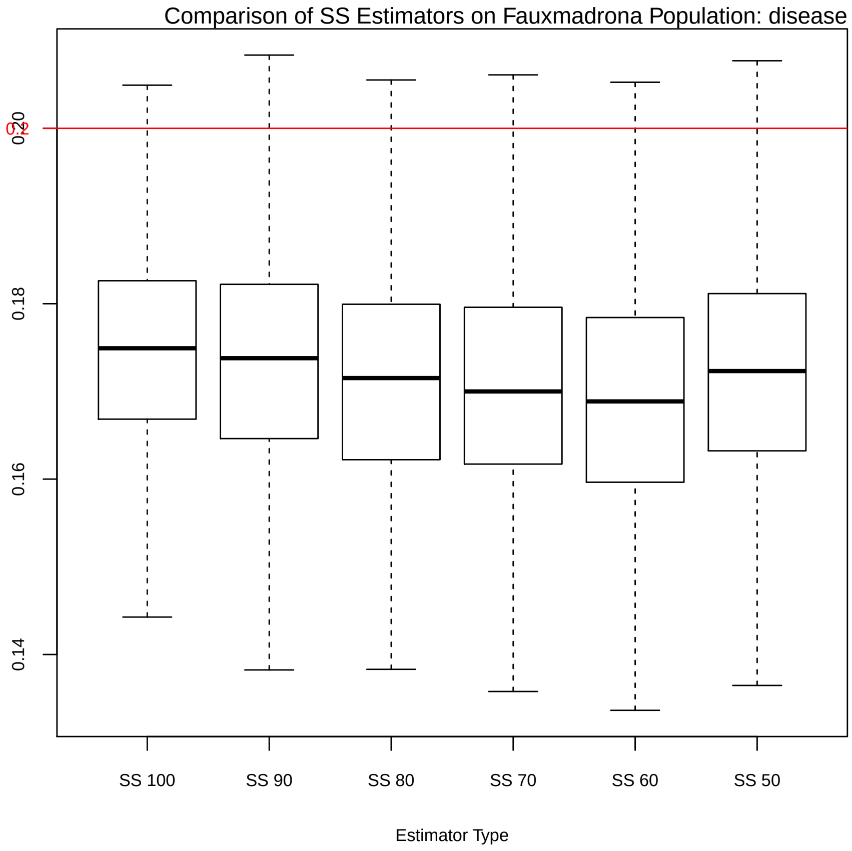


Figure 4.3: N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486

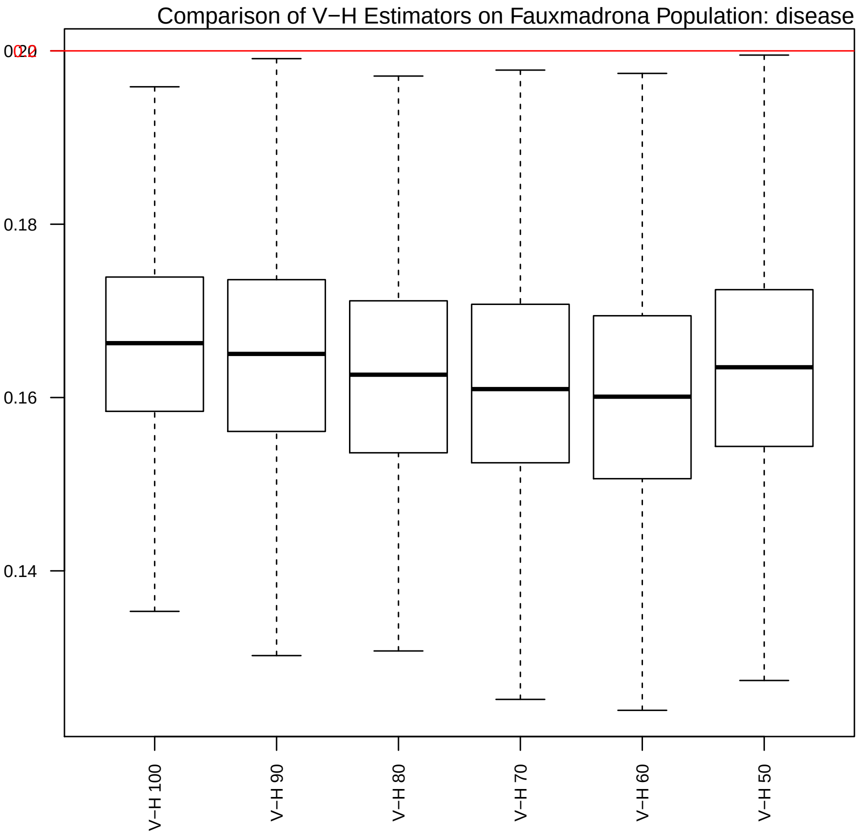


Figure 4.4: N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486

Table 4.4: Scaled MSE for SS estimator on Fauxmadrona

	SS 100	SS 90	SS 80	SS 70	SS 60	SS 50
disease	2.79	2.97	3.15	3.24	3.38	3.11

Finally, we examined the strategy of replacing reported degree with rank of reported degree as an estimate of degree. Since the SS estimator was the best performer under standard conditions (no recall error), we used two rank transformations with this estimator. We hoped that, when recall is less than perfect, the estimators using rank of reported degree would perform better. First we tried the dense ranking transformation. Boxplots are shown in Figure 4.5. MSE results are shown in Table 4.5. According to the table, the resulting rank estimator beats the untransformed estimator when recall is 50 percent. It is worse at 100 percent and the same at 90 percent. These trends are slightly different from those in the appendix under probability sampling conditions.

Next we tried the standard competition rank transformation. Boxplot results are shown in Figure 4.5 and MSE results are shown in Table 4.5. It is clear that the standard competition transformation does not improve the performance of the SS estimator. Although the variance of the Rank estimator is less than that of the regular SS estimator under conditions of 90 percent recall, the Rank estimator also shows a much greater bias. By 50 percent recall, the variance of the Rank estimator is much increased and the estimator bias remains substantial. The regular SS estimator beats the standard competition Rank version of the estimator according to the MSE at both 50 and 90 percent recall. These trends are replicated in the appendix to the thesis under probability sampling conditions.

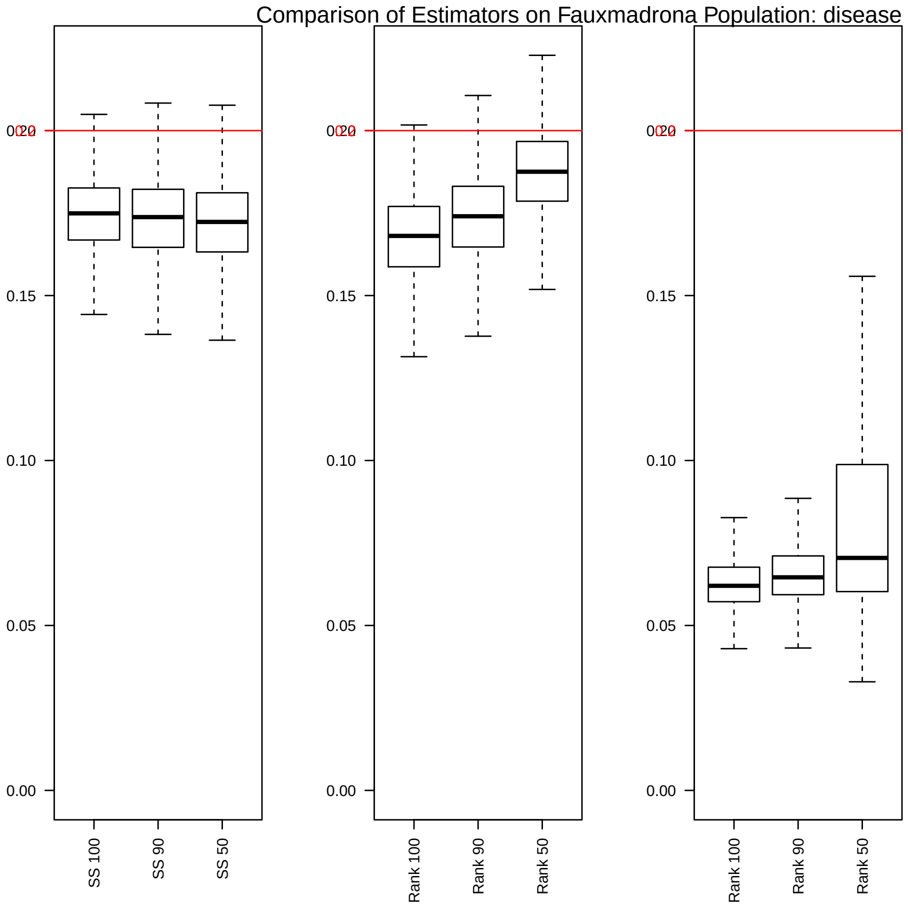


Figure 4.5: N (population) = 1000; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.52931, relative activity = 1.759486; Dense transformation in the middle and SC transformation on the right.

Table 4.5: Scaled MSE for Estimator SS on Fauxmadrona

Recall	SS	Rank (SS, dense)	Rank (SS, SC)
100	2.79	3.47	13.73
90	2.97	2.97	13.40
50	3.11	1.87	12.24

CHAPTER 5

Analysis of Project 90

5.1 Project 90: An Overview

Project 90 refers to a public health project undertaken by the American Center for Disease Control between 1987 and 1992 [10]. The goal of the project was to investigate the incidence and spread of the HIV virus in a heterosexual population thought to be at high risk for the disease. The CDC wished to investigate not just persons at risk but the relations among them. Previous investigations suggested that global structure, as well as individual behavior, had a role to play in the spread of HIV. Project 90 concentrated on sex workers, injecting drug users, and their partners living in Colorado Springs, CO [8].

The CDC partnered with the local public health department to reach out to prospective participants. The public health workers already knew most of Colorado Springs' female sex workers, many of whom agreed to participate in the study. They recruited injecting drug users from the local methadone clinic and through outreach efforts. Participants were interviewed about personal details and agreed to take blood tests. They were asked in detail about those with whom they were in close personal contact. "Cross contacts," those individuals mentioned by more than one interviewee, were contacted and invited to join the study.

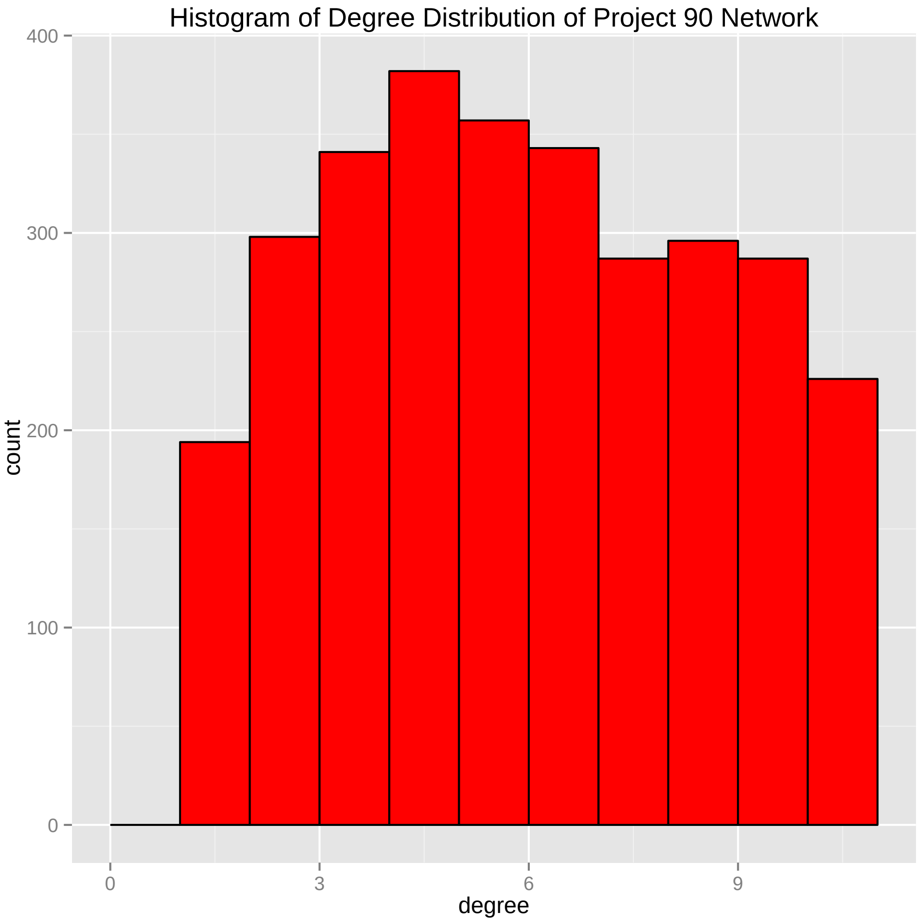
During the four years of the study (1988-1991), 595 individuals were interviewed altogether [10, 11, 8, 18]. Together, those participants and their contacts who were not themselves recruited comprised a social network of 5,162 distinct individuals [18]. This network contained a “giant” connected component of 3,016 individuals [11].

It would be possible to treat the Project 90 social network as a sample from a larger population. In fact, some of the recruiting even involved link-tracing, when cross-contacts were invited to join the study. However, for the purposes of this paper, we will treat the Project 90 network as a population. The network we will work with, which has been trimmed of unconnected members, has 3,011 individuals. Its degree distribution is depicted in a histogram in Figure 5.1. The mean degree is 5.448 and the standard deviation is 2.662. While the initial interest was in HIV prevalence and spread, we will focus on 12 dichotomous actor attribute variables describing demographic characteristics. These variables are called **disabled**, **nonwhite**, **sex.work.client**, **drug.dealer**, **sex.worker**, **thief**, **housewife**, **unemployed**, **retired**, **homeless**, **drug.cook**, and **pimp**. Each is coded 1 if the individual falls into the described category, and 0 otherwise. The variables range in relative activity from 0.8379733 (**sex.work.client**) to 1.252161 (**pimp**). They range in homophily from 1.023402 (**housewife**) to 2.641745 (**nonwhite**). The last statistic is especially noteworthy, as it indicates a certain level of racial segregation in the Project 90 population.

5.2 Results

To compare the various estimators, we again used C code to simulate separate sets of 1000 samples each. In all cases, samples have 500 individual members drawn from a total population of 3011 individuals, for a sample fraction of 0.166. For each simulation, ten seeds were chosen, drawn proportional to the true degree. RDS was simulated using 2 coupons per individual. We again used the modified binomial distribution to model recall error. As in the fauxmadrona simulations, each time the percent recall error was varied a new set of 1000 samples was drawn.

Figure 5.1:


Histogram of Degree Distribution of Project 90 Network
count
400
300
200
100
0
0
3
6
9
degree

Holding recall error constant, the same samples were used to estimate the true mean of the twelve different variables under consideration.

Figures 5.2 through 5.13 show boxplots for the results of simulations with and without recall error. Each figure is devoted to a single variable. The population homophily and relative activity of the variable in question is included in the caption. Each figure contains three different graphs. In each graph, the variable's true mean is indicated by a labeled horizontal line. The first graph uses boxplots to compare the sample mean, the S-H estimator, the V-H estimator, and the SS estimator. The labels S-H, V-H, SS, and Mean indicate which estimator was used. The number following the label indicates the percentage recall, in this case always 100. The MSE results for these figures are summarized in Table 5.1. The best results for each variable is bolded.

The next graph in each figure examines the effects of recall error. To do so, we first chose the best of the SS, S-H, and V-H estimators under conditions of no recall error. These were the SS estimator for the variables **disabled**, **sex.work.client**, **housewife**, **retired**, **homeless**, and **drug.cook**, and the S-H estimator for the variables **nonwhite**, **drug.dealer**, **sex.worker**, **thief**, **unemployed**, and **pimp**. The boxplots are again labelled with the estimator names and the percent recall. The MSE results for the SS variables are shown in Table 5.2 and the MSE results for the S-H variables are shown in Table 5.3.

The final graph in each figure shows the results of a dense rank transformation. The estimators used were SS for the SS group and S-H for the S-H group. The MSE results for the rank transformations are summarized in Table 5.4 (the SS group) and Table 5.5 (the S-H group). A standard competition transformation was attempted additionally, but the resulting estimator uniformly performed poorly.

The results for the Project 90 simulation are less easy to interpret than the fauxmadrona results. The first question is whether the estimator results under conditions of 100 percent recall are in accordance with Gile and Handcock's results.

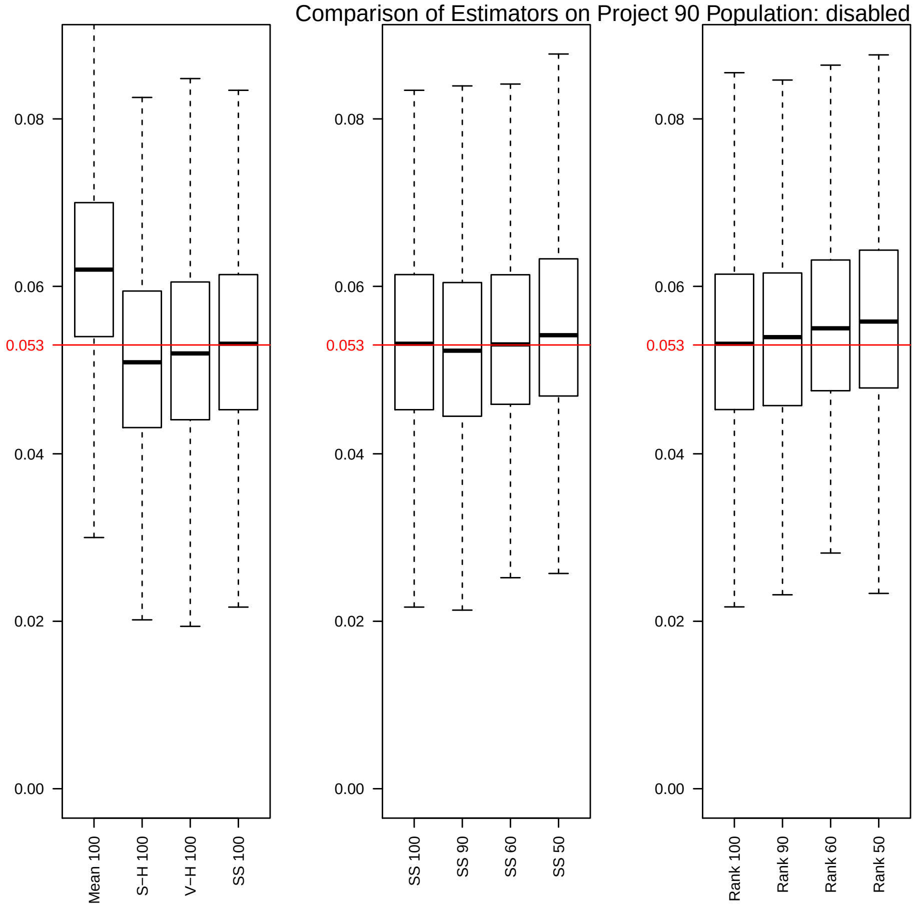


Figure 5.2: N (population) = 3011; n (sample size) = 500; seeds = 10, drawn proportional to degree; coupons per individual= 2; total samples = 1000; population homophily = 1.13318; population relative activity = 1.152845

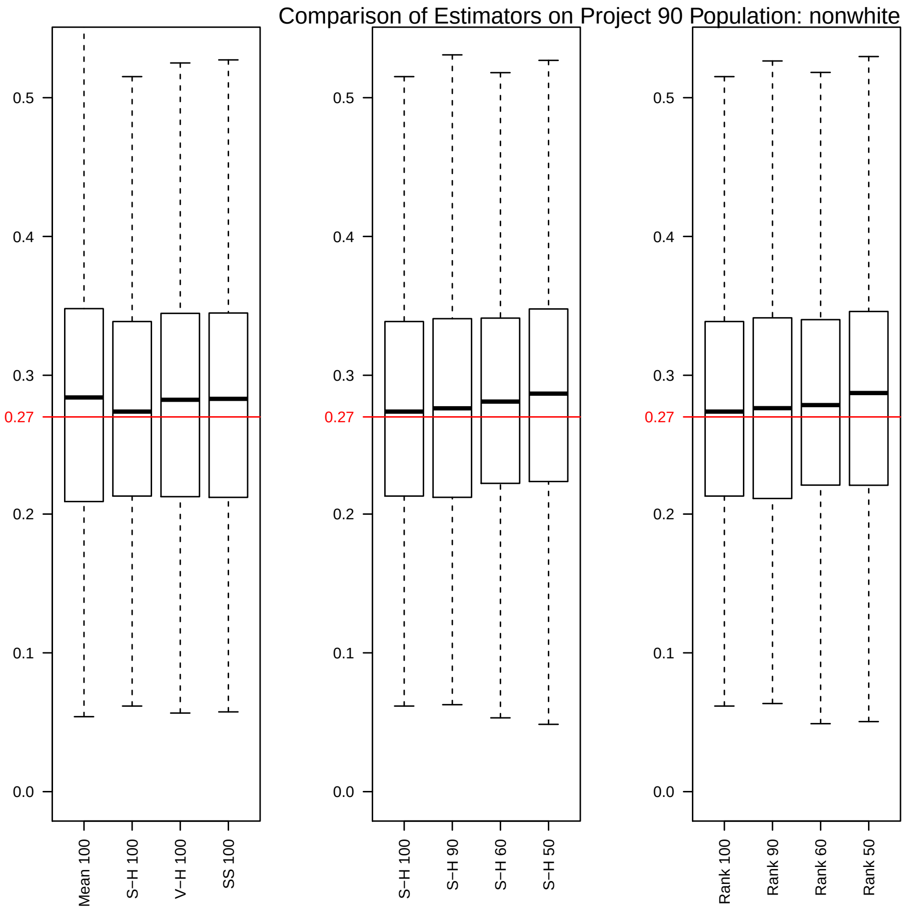


Figure 5.3: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 2.641745; population relative activity = 0.9742704

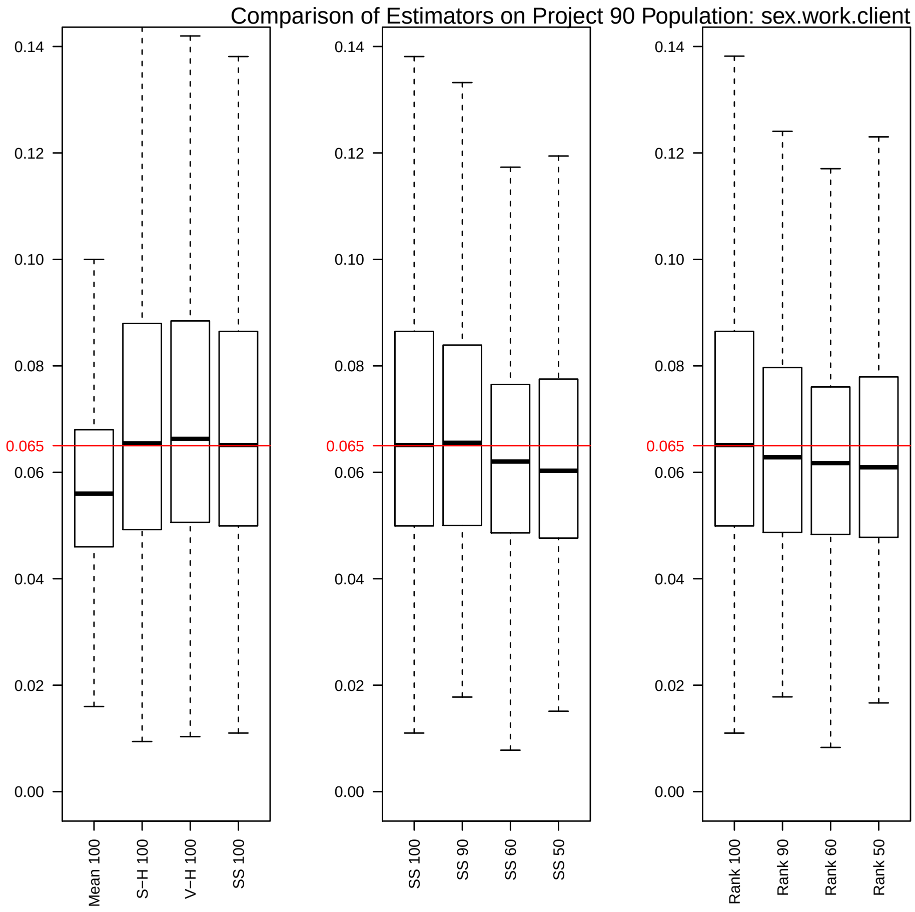


Figure 5.4: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.2042409; population relative activity = 0.8379733

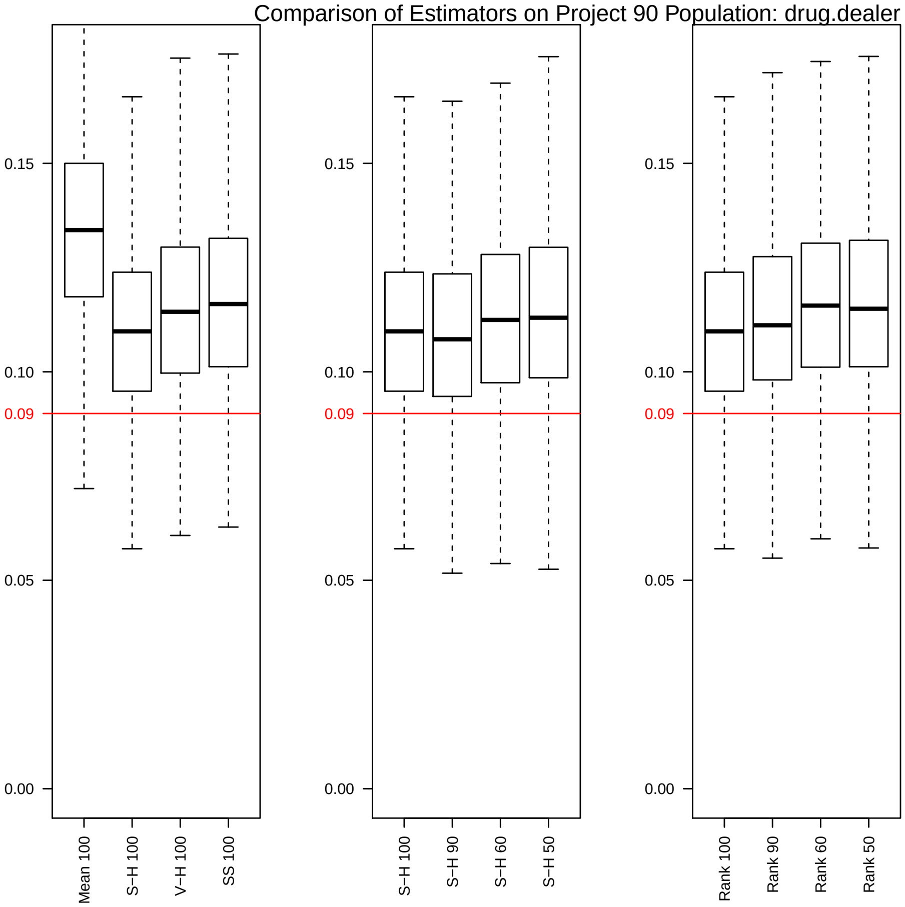


Figure 5.5: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.253184; population relative activity = 1.114024

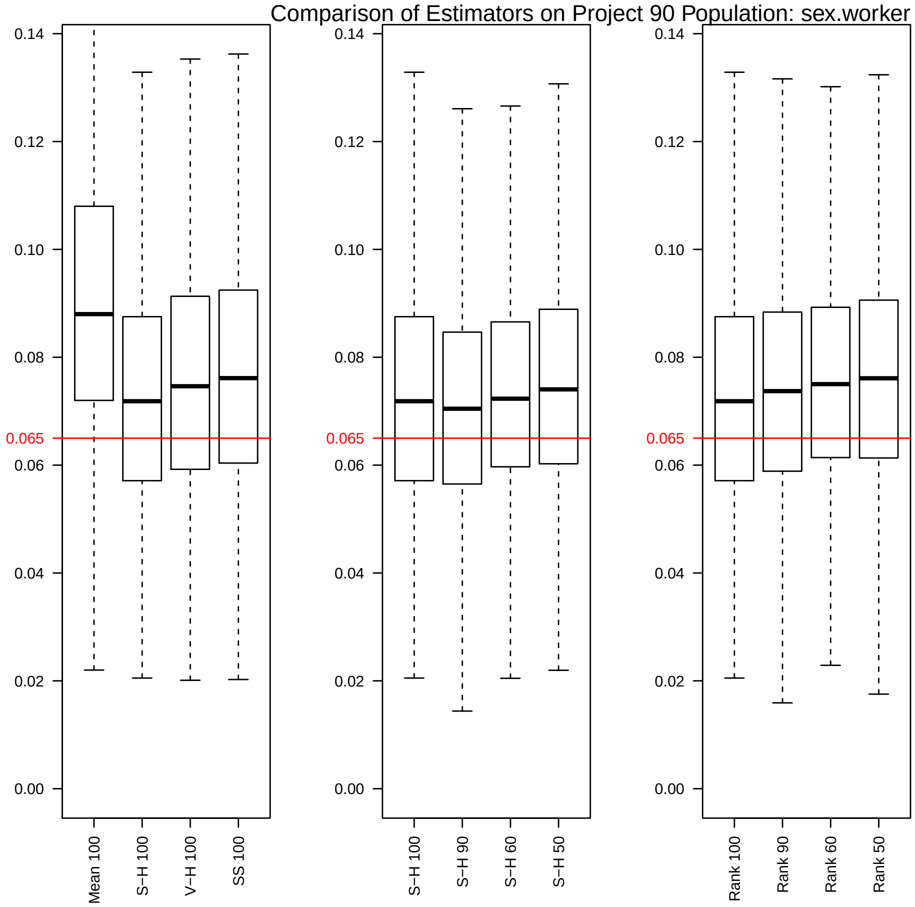


Figure 5.6: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.255978; population relative activity = 1.176488

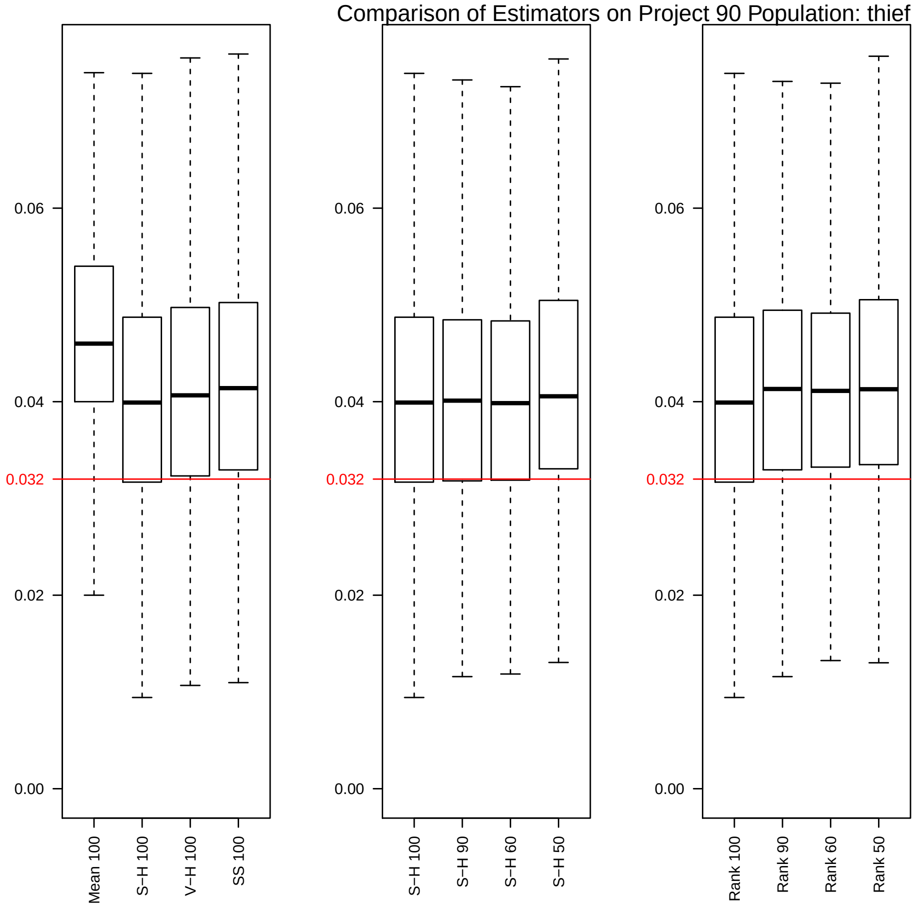


Figure 5.7: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.215924; population relative activity = 1.074868

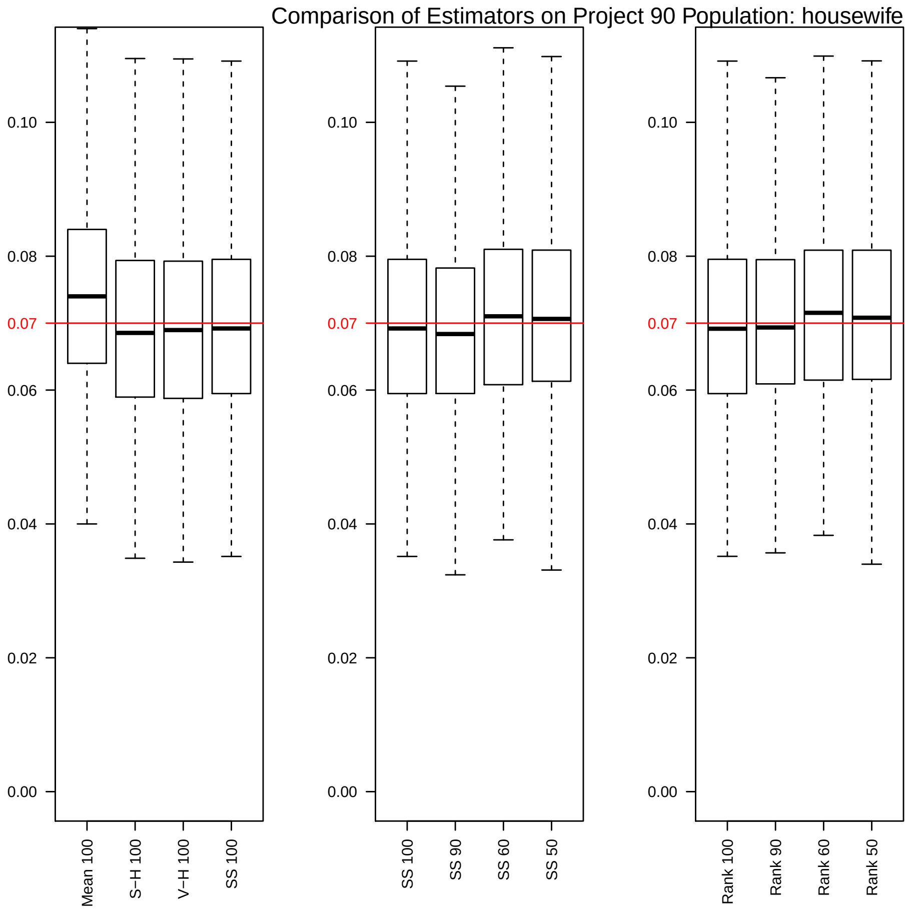


Figure 5.8: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.023402; population relative activity = 1.096503

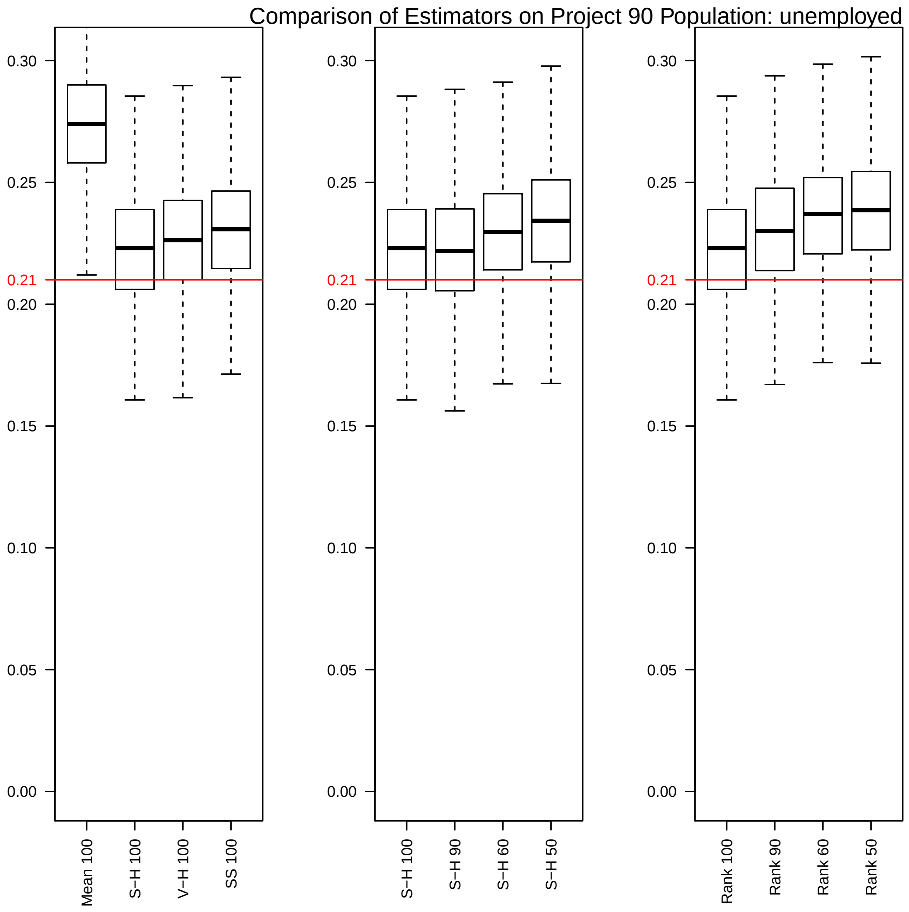


Figure 5.9: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.107955; population relative activity = 1.212837

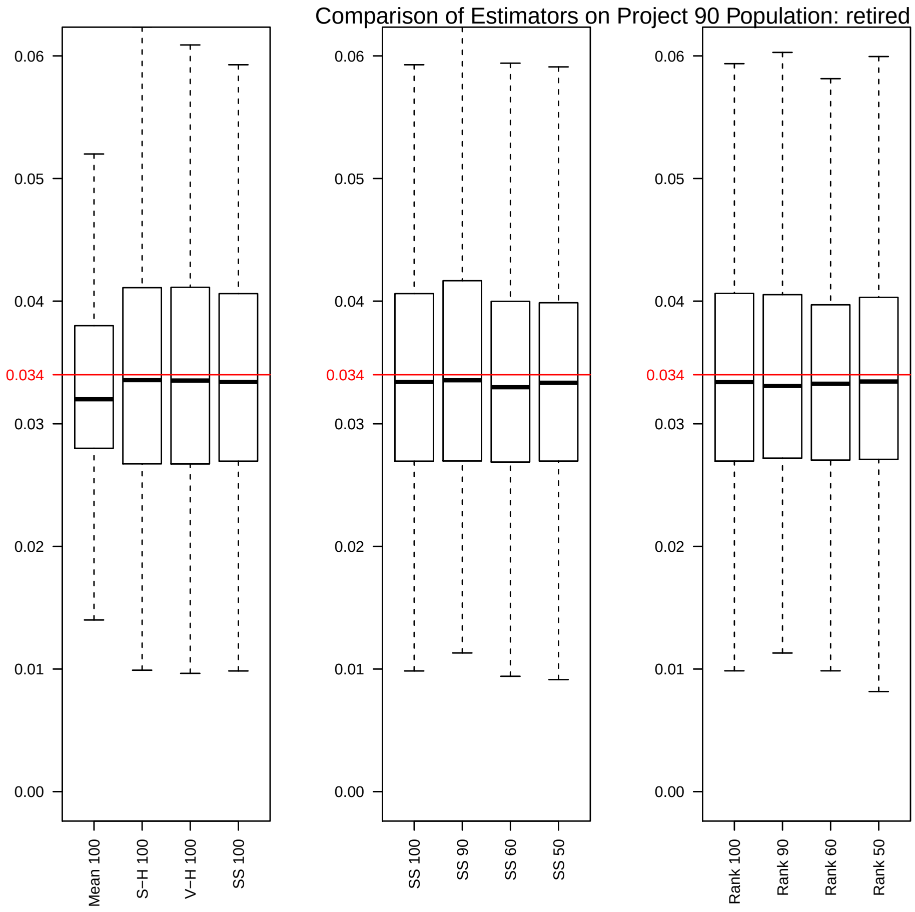


Figure 5.10: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.032841; population relative activity = 0.9633545

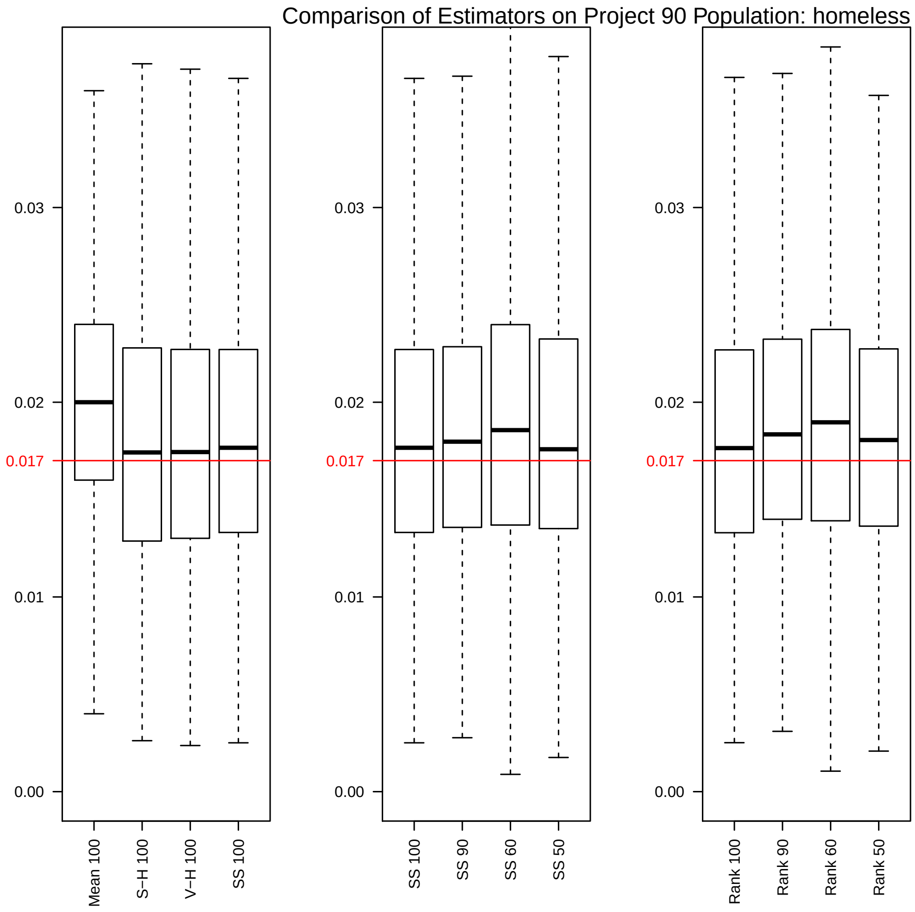


Figure 5.11: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.068735; population relative activity = 0.9702125

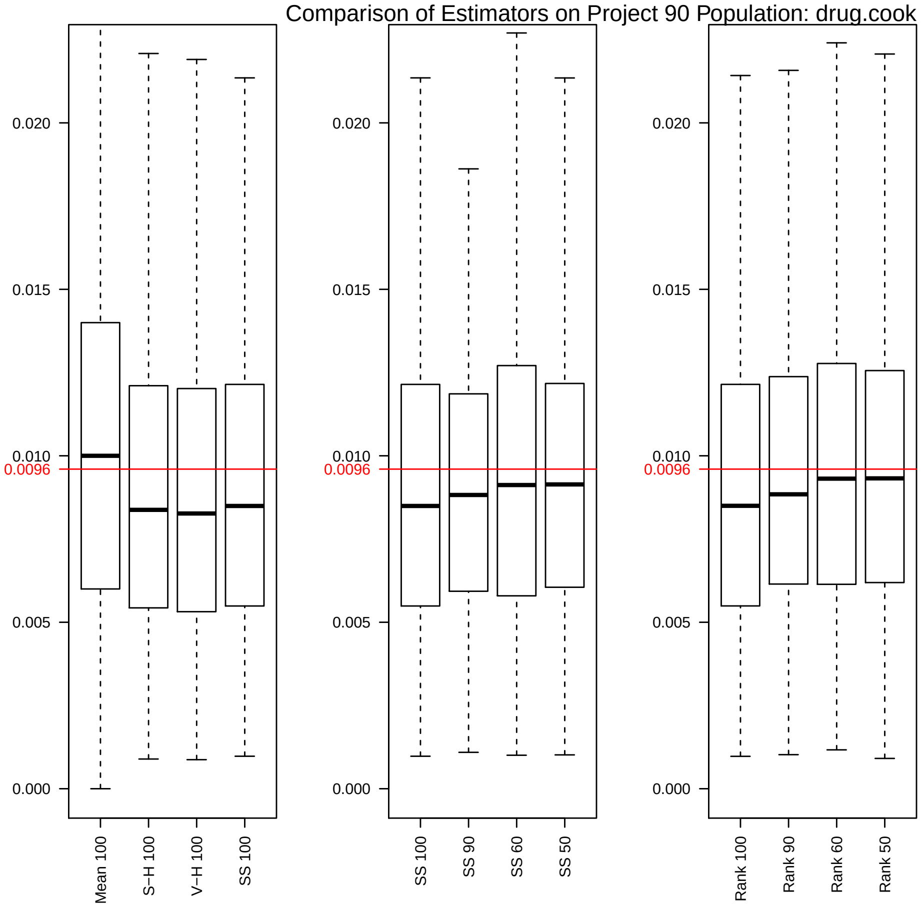


Figure 5.12: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.024186; population relative activity = 1.115212

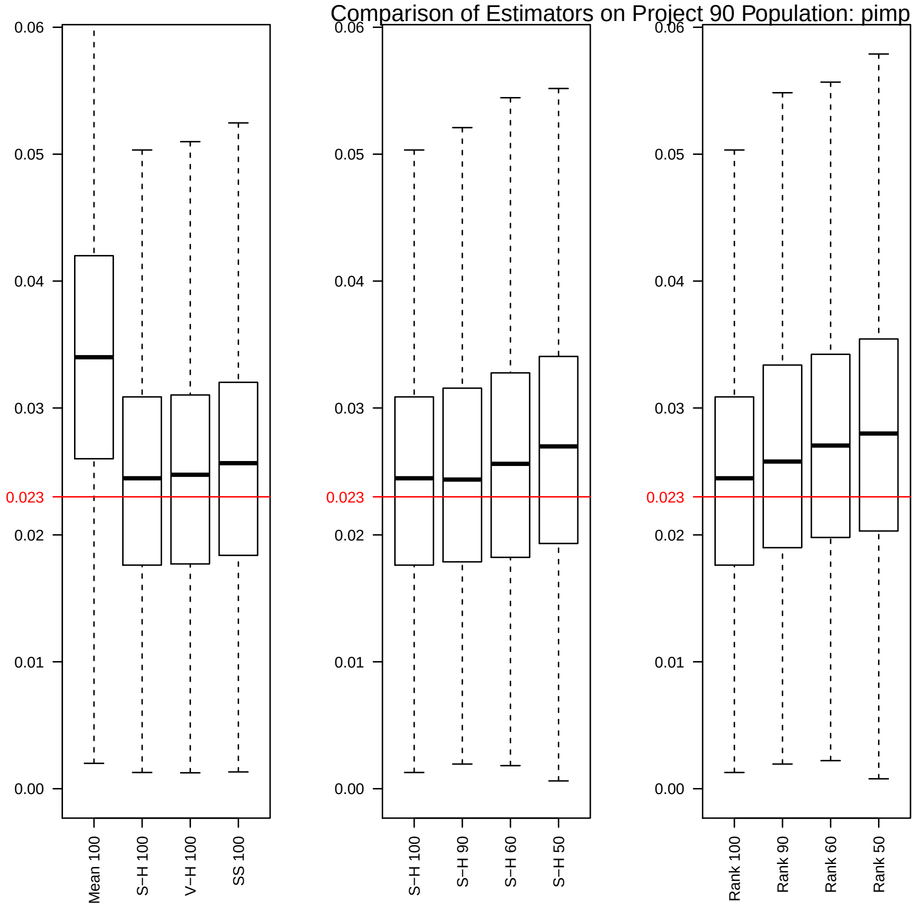


Figure 5.13: N = 3011; n = 500; seeds = 10, drawn proportional to degree; coupons = 2; total samples = 1000; population homophily = 1.033291; population relative activity = 1.252161

Table 5.1: Scaled MSE for Mean, S-H, V-H, and SS Estimators on Project 90

	Mean 100	S-H 100	V-H 100	SS 100
disabled	1.47	1.16	1.17	**1.15**
nonwhite	9.69	**8.90**	9.44	9.45
sex.work.client	**1.86**	2.74	2.73	2.60
drug.dealer	4.94	**2.90**	3.31	3.43
sex.worker	3.56	**2.19**	2.39	2.46
thief	1.97	**1.53**	1.62	1.64
housewife	**1.45**	1.50	1.48	1.46
unemployed	7.05	**2.86**	3.06	3.32
retired	**0.71**	1.10	1.07	1.03
homeless	**0.65**	0.75	0.74	0.72
drug.cook	**0.46**	3.17	0.51	0.49
pimp	1.66	**0.97**	0.97	1.01

Since the seeds are drawn proportional to degree, there are no results of seed bias to consider. Next, Gile and Handcock show that the SS estimator is a better performer under conditions of large sampling fraction and relative activity that is less than or greater than 1. It turns that, for the Project 90 results, the mean relative activity for the SS group is 1.023, while the mean relative activity for the S-H group is 1.134. The difference is not particularly striking, and the mean for the SS group is actually less than the mean for the other estimators. The fact that the SS estimator is not a better performer under conditions of high relative activity may be due to the fact that the sampling fraction is relatively small. Finally, the mean homophily for the SS group is 1.081, while the mean homophily for the S-H group is 1.418. This difference does seem substantial. This suggests that under conditions of high homophily, the S-H estimator might actually outperform the SS estimator. The two estimators were not directed compared by Gile [2], but the

Table 5.2: Scaled MSE for SS Estimator on Project 90

	SS 100	SS 90	SS 60	SS 50
disabled	1.15	1.15	1.18	1.22
sex.work.client	2.60	2.42	2.09	2.18
housewife	1.46	1.41	1.52	1.47
retired	1.03	1.09	0.95	0.95
homeless	0.72	0.72	0.79	0.71
drug.cook	0.49	0.21	0.50	0.20

Table 5.3: Scaled MSE for S-H Estimator on Project 90

	S-H 100	S-H 90	S-H 60	S-H 50
nonwhite	8.90	8.88	8.72	9.23
drug.dealer	2.90	2.85	3.18	3.30
sex.worker	2.19	2.19	2.09	2.30
thief	1.53	1.52	1.53	1.62
unemployed	2.86	2.77	3.30	3.61
pimp	0.97	0.98	1.06	1.15

result does contradict the hope that the SS estimator is basically as good as or better than other estimators under non-standard conditions. Most strikingly, by the standard of the MSE, the arithmetic mean does better than any of the special estimators in 5 out of 12 cases.

Changes in recall tend to affect the bias of the estimators, but not in any one direction, and not always monotonically. Nor does the variance uniformly increase with increasing recall value. The standard-competition rank transformation (not shown), generally speaking increases the MSE of the estimators. This transformation dramatically increases the bias of the estimators, mostly pushing values downward. It sometimes decreases the variance of estimators for modest recall error, but not uniformly. On the other hand, the dense ranking transformation

Table 5.4: Scaled MSE for Rank SS Estimator on Project 90

	Rank 100	Rank 90	Rank 60	Rank 50
disabled	3.42	3.07	3.45	3.42
sex.work.client	11.28	8.21	6.57	5.99
housewife	3.99	3.47	4.21	3.91
retired	3.99	3.65	3.50	3.28
homeless	2.03	1.91	2.16	2.09
drug.cook	1.62	1.55	1.63	1.57

Table 5.5: Scaled MSE for Rank S-H Estimator on Project 90

	Rank 100	Rank 90	Rank 60	Rank 50
nonwhite	10.92	10.32	11.58	11.34
drug.dealer	5.30	4.60	5.22	4.81
sex.worker	4.92	4.18	4.16	3.92
thief	3.17	2.87	3.10	3.18
unemployed	10.01	8.40	7.76	7.45
pimp	1.93	1.89	2.23	2.05

does not make much of a difference to the MSE. This is because even at low levels of recall, most degrees are represented in samples from Project 90. Thus the dense transformation results in the same degree distribution in most cases, and in similar MSE values.

Table 5.6: Scaled MSE for V-H, SS, S-H, and Rank (S-S, dense) at 50 percent recall

	V-H 50	S-H 50	SS 50	Rank 50
disabled	1.23	1.19	1.22	1.24
sex.work.client	2.23	2.27	2.18	2.19
housewife	1.49	1.50	1.47	1.44
retired	0.98	0.99	0.95	0.95
homeless	0.73	0.74	0.71	0.70
drug.cook	0.20	0.47	0.20	0.48

Table 5.7: Scaled MSE for V-H, SS, S-H, and Rank (S-H, dense) at 50 percent recall

	V-H 50	SS 50	S-H 50	Rank 50
nonwhite	9.90	9.90	9.23	9.20
drug.dealer	3.72	3.81	3.30	3.42
sex.worker	2.54	2.61	2.30	2.41
thief	1.71	1.72	1.62	1.64
unemployed	3.95	4.18	3.61	3.90
pimp	1.16	1.19	1.15	1.18

CHAPTER 6

Conclusion

Respondent-driven sampling (RDS) is used to collect samples when standard probability sampling is difficult or impossible. Since analytic properties of estimators developed for RDS are difficult to determine, simulations have been used to test the sensitivity of estimators to a variety of assumptions. This thesis presents C code written to speed up the RDS simulation functionality offered by the R package RDSdevelopment. It shows that the C code performs more quickly than the earlier code, written purely in R, and produces samples similar but not identical to the earlier code.

Many of the RDS estimators use an individual's reported number of connections to weight information about that individual in calculations. We use simulations to examine the sensitivity of the estimators to error in survey participants' recall of degrees. We find that recall error, modeled by a modified binomial distribution, is associated with changing bias in estimators, but that the direction of the bias is different across estimators and not monotonic as error increases.

We also look at the effects of performing a rank transformation on the sample degree distribution before computing estimators. We consider two different rank transformations, the standard competition transformation and the dense transformation. A standard competition rank transformation of reported degrees sometimes decreases the variance of estimators but also tends to introduce large amounts of bias. For the networks we consider, the dense transformation does not change the sample degree distribution very much. Thus the resulting estimators are similar as well.

In the appendix to the thesis, we examine the effects of recall error and rank transformations in the context of straightforward probability sampling from the “fauxmadrona” population. If individuals are chosen with replacement for a sample proportional to their degree, the H-H estimator is a good estimate for the sample mean. But if weights are replaced in the H-H estimator by “recalled weights,” and error is introduced, estimator performance changes. We show that as recall error increases, the H-H estimator bias first increases negatively and then positively. Moreover, rank transformations do not help in the cases we consider in this context either. As in the RDS context, the standard competition transformation results in increased MSE and the dense transformation does not make much of a difference.

Thus, even in the absence of the complications of RDS and network-specific features like homophily, recall error changes estimators. The next step would be to try to model these changes analytically in the probability sampling case, if possible. We could also consider different degree distributions, for which the dense rank transformation might make a difference. Or we could look at other ways of adjusting the reported recall under conditions of recall error. Finally, once we had explored the probability sampling context more thoroughly, we could see how network features like homophily might complicate matters further.

CHAPTER 7

Appendix: Recall Error and Rank Transformations with Probability Sampling

The respondent-driven sampling context is, without a doubt, complex. In order to explore the effect of recall error and rank transformations in a simpler context, we also tried probability sampling from the fauxmadrona population described in Chapter 4. Here we used R's "sample" function to draw samples of 500, with replacement, treating the nodal degrees as the sampling weights. We then used the V-H estimator to estimate the sample mean. In this case, given the way we gathered the sample, the V-H estimator can be regarded simply as the ratio of two Hansen-Hurwitz estimators. (Hereafter, in this context, we will refer to the V-H estimator as the "H-H estimator.") We modeled recall error from 90 percent all the way to 0 percent, using the same 1000 samples and the modified binomial distribution presented in Chapter 2. We then tested a dense and standard competition rank version of the H-H estimator as well. Because the time needed to carry out estimator calculations was negligible, we were able to look at a wide variety of recall errors and their effect on estimator performance. In order better to analyze the situation, we also examined the degree distribution of the first sample drawn under a variety of conditions.

Figure 7.1 shows histograms of the degree distribution of the first sample under a variety of different levels of recall error. The different colors indicate the proportion of each degree for which the disease equals 1 or 0. Then Figure 7.2 presents the same information using a relative distribution plot. Under high levels of recall, the relative activity in the population is reflected in that of the sample. However, as recall erodes, more and more reported degrees are set equal to 1, regardless of the disease level of the individual. Thus the sample degree distributions of the infected and uninfected nodes converge. The relative distribution plots reflect this trend.

Boxplots for the results without rank transformation (hereafter referred to as the untransformed H-H estimator) are presented in Figure 7.3. We can see that, under conditions of increasing recall error, the untransformed H-H estimator mean first decreases, then increases until it is back to the original level, and finally increases until it reaches the level of the arithmetic mean. We conjecture that the U-shape is due to two competing tendencies. The first trend occurs as recall error erodes and more and more weights are set to 1, and the estimator approaches the mean. We conjecture that the second trend, which at first pushes the bias down, has to do with the range of the degree distribution. The greater the arithmetic difference between the high and low degrees, the more powerful this trend is. Scaled estimator bias, variance, and MSE are given in Table 7.1. Table 7.1 reflects the U-shaped trend just described and also reveals that the variance increases modestly and monotonically as recall erodes. Thus, the V-H and SS estimators under RDS conditions and the untransformed H-H estimator under probability sampling conditions proportional to degree exhibit the same very general trends.

The degree distribution of the first sample, after a standard competition transformation, is shown in Figure 7.4. Boxplots for the standard competition rank results are presented in Figure 7.5. Then, the degree distribution of the first sample, after a dense transformation, is shown in Figure 7.6. Boxplots for the dense rank results are presented in Figure 7.7. Table 7.1 summarizes this information as well.

Figure 7.5 show that, between 90 and 50 percent recall, the standard competition rank H-H variance is smaller than that of the untransformed H-H estimator. However, the estimator bias is so great that the rank H-H MSE results do not beat the untransformed H-H MSE results. As recall continues to decline, the rank H-H estimator bias decreases but the variance increases. Table 7.1 shows that, judged by the standard of the MSE, the untransformed H-H estimator is better than the standard competition rank H-H estimator at most levels of recall.

Figure 7.7. and Table 7.1 suggest that a dense transformation does not make a big difference to the MSE. This again reflects the degree distribution of the fauxmadrona network.

Table 7.1: Scaled Bias, Variance, and Mean-squared Error of Estimator on Samples drawn with replacement

	100	90	80	70	60	50	25	10
H Bias	0.03	-0.19	-0.42	-0.63	-0.69	-0.43	3.12	8.08
Rank (SC) H Bias	-15.00	-14.63	-14.42	-14.82	-14.70	-13.43	-3.96	5.90
Rank (D) H Bias	-1.11	-0.55	-0.45	-0.63	-0.69	-0.43	3.12	8.08
H Variance	1.71	1.71	1.76	1.77	1.80	1.90	2.02	2.05
Rank (SC) H Variance	1.22	1.75	2.18	2.54	3.21	3.39	2.69	2.18
Rank (D) H Variance	1.81	1.80	1.75	1.77	1.80	1.90	2.02	2.05
H MSE	1.71	1.73	1.81	1.88	1.92	1.95	3.72	8.33
Rank (SC) H MSE	15.05	14.73	14.59	15.04	15.05	13.85	4.78	6.29
Rank (D) H MSE	2.12	1.88	1.80	1.87	1.92	1.95	3.72	8.33

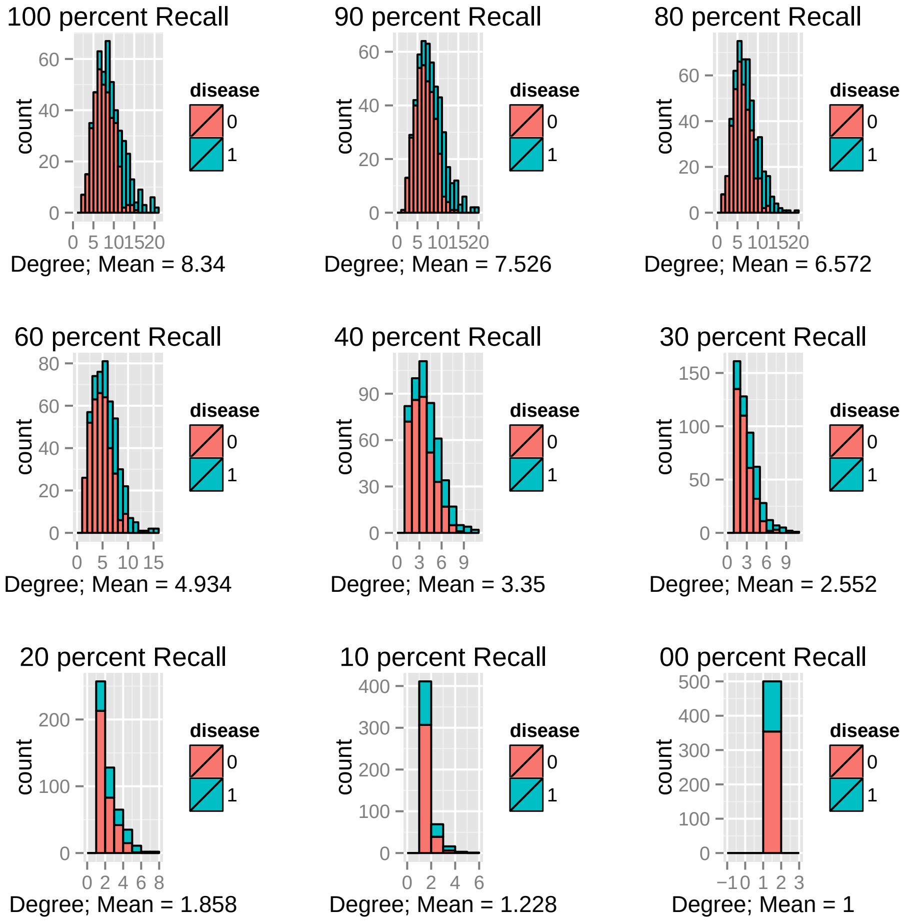


Figure 7.1: Degree Distribution of first sample at different levels of recall error. No rank transformation on degrees. Samples drawn with replacement.

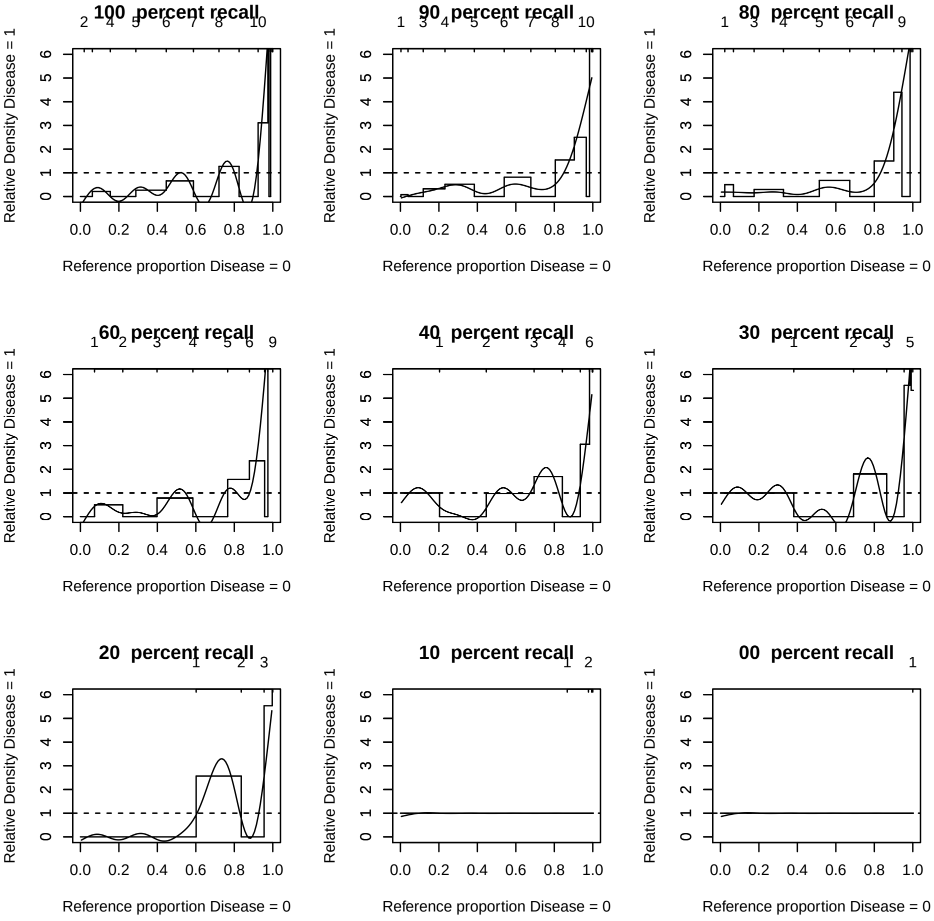


Figure 7.2: Relative Degree Distributions comparing Disease = 1 and Disease = 0 in first sample at different levels of recall error. No rank transformation on degrees. Samples drawn with replacement.

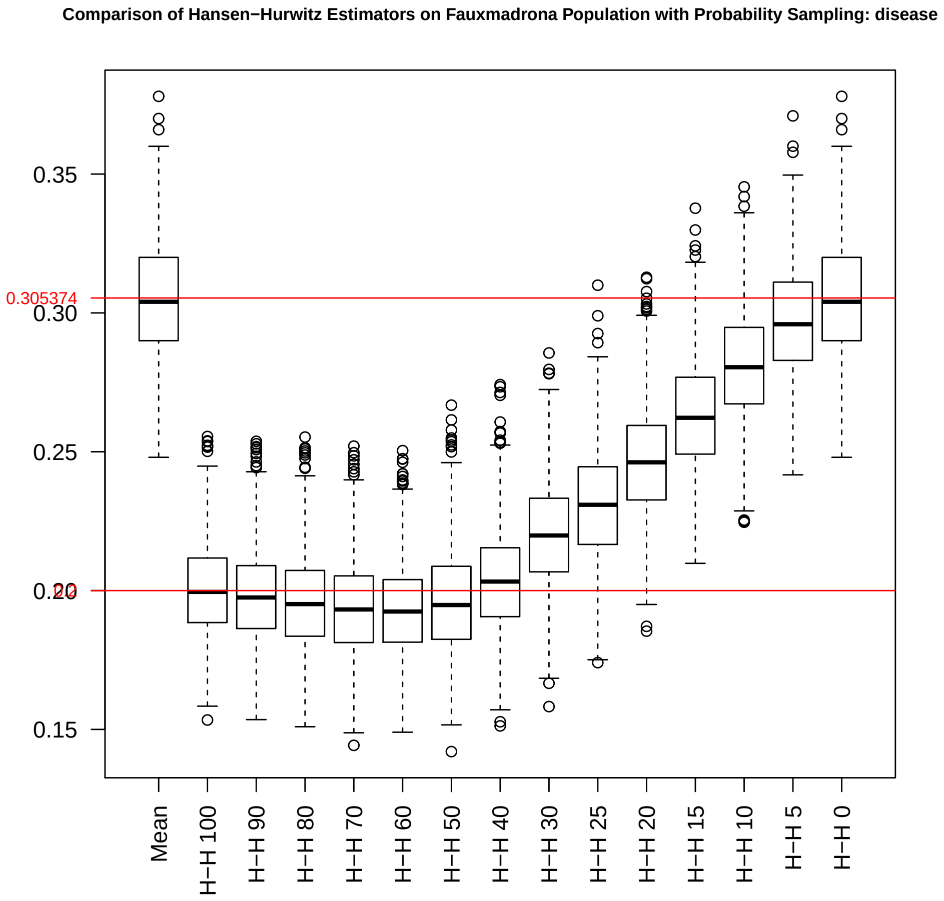


Figure 7.3: Distribution of μ for different levels of recall error. No rank transformation on degrees. Samples drawn with replacement.

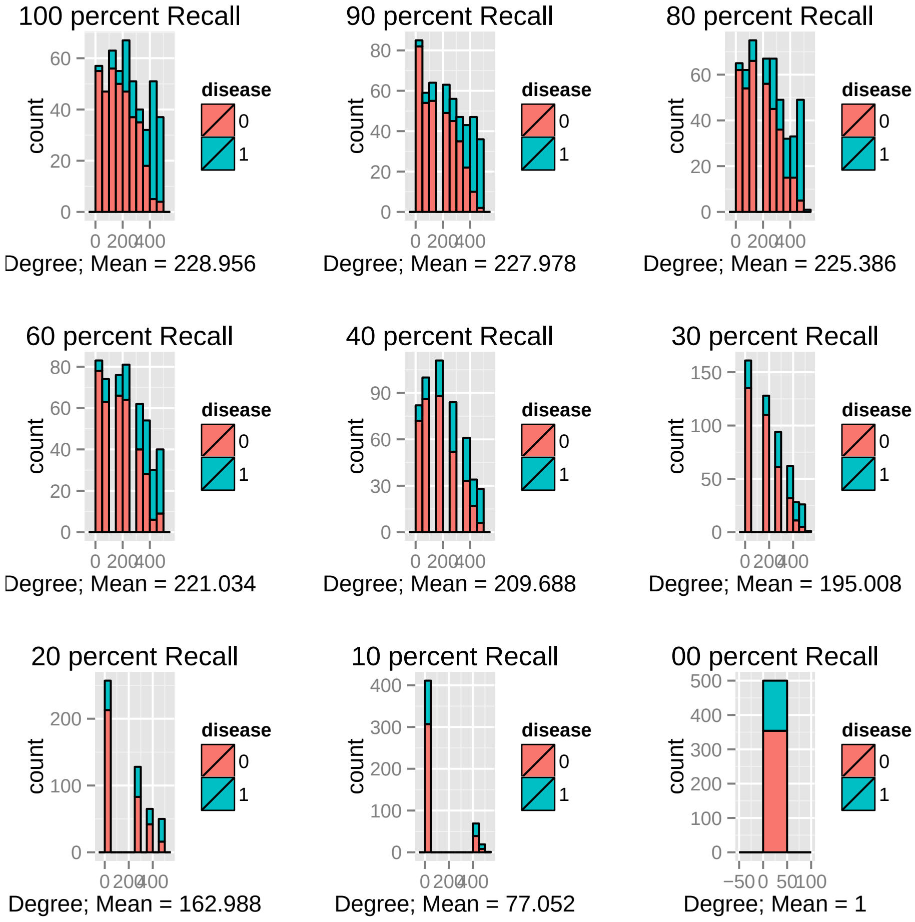


Figure 7.4: Degree Distribution of first sample at different levels of recall error. Standard competition rank transformation on degrees. Samples drawn with replacement.

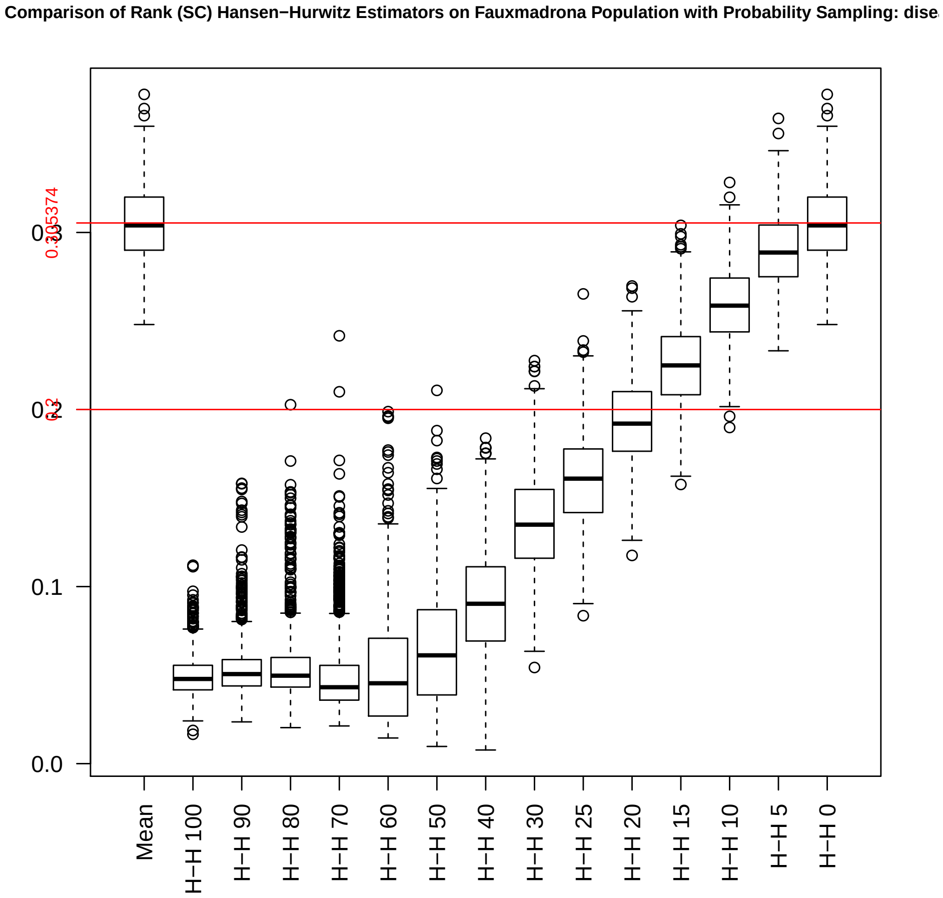


Figure 7.5: Distribution of μ for different levels of recall error. Standard competition rank transformation on degrees. Samples drawn with replacement.

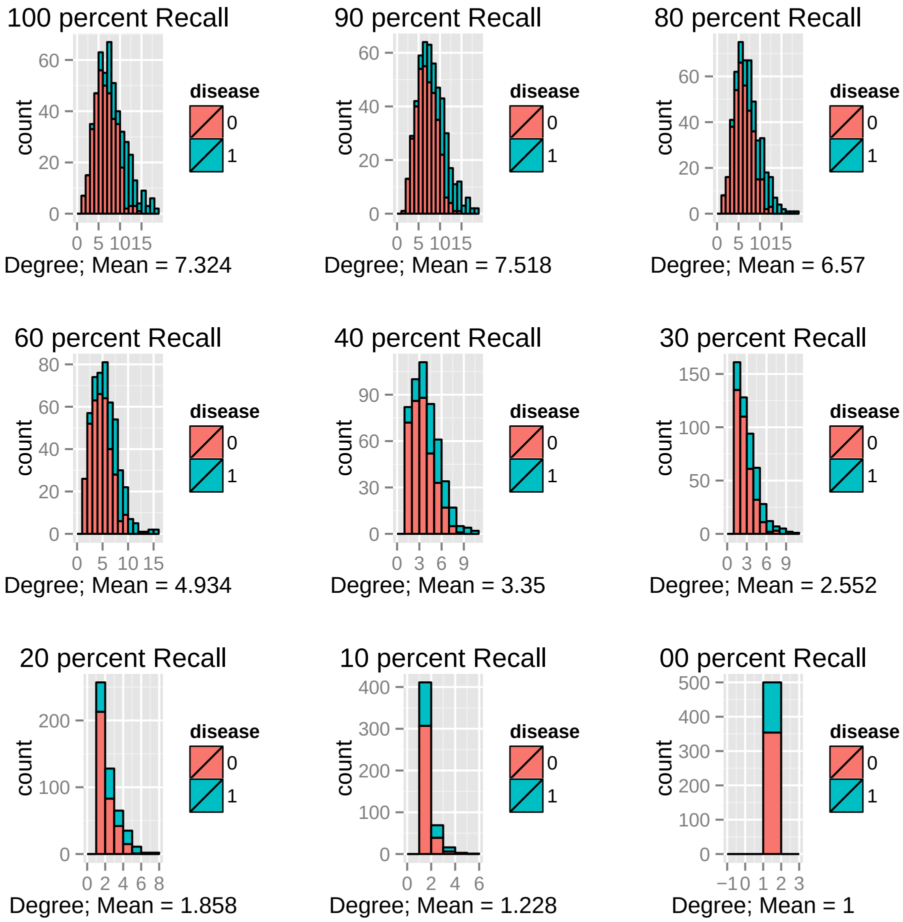


Figure 7.6: Degree Distribution of first sample at different levels of recall error. Dense rank transformation on degrees. Samples drawn with replacement.

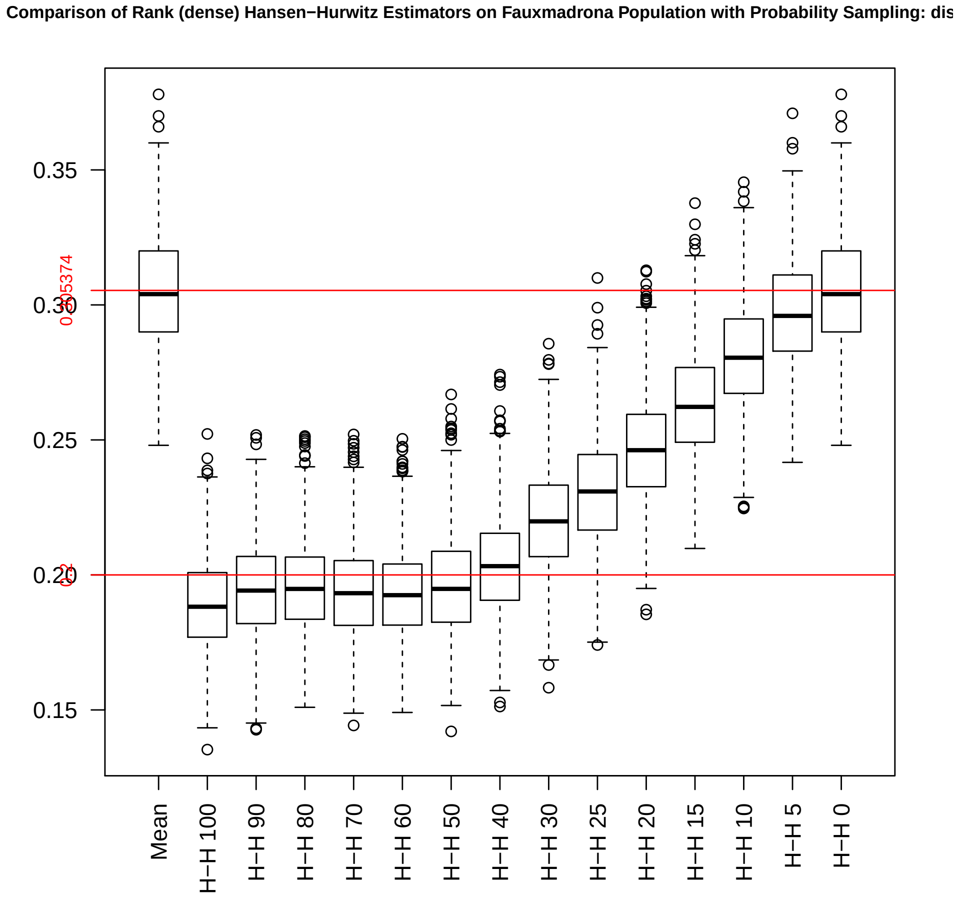


Figure 7.7: Distribution of μ for different levels of recall error. Dense rank transformation on degrees. Samples drawn with replacement.

CHAPTER 8

A Final Note on the Code

Code for this project includes the new C and R Code that simulates respondent-driven sampling, already described in Chapter 3 of the thesis. This code can be accessed through the latest version of the RDSdevelopment project, which is now available on Mark Handcock's UCLA website.

The simulations and estimator results for the project also relied on a considerable amount of R Code. The code is available by request from the author, who can be reached by email at margaret.meek@gmail.com. The data has also been saved in a number of RData objects and is available upon request as well.

References

[1] Carter T. Butts, Mark S. Handcock, and David R. Hunter. *network: Classes for Relational Data.* Irvine, CA, 2013. R package version 1.8.1.

[2] K.J. Gile. Improved inference for respondent-driven sampling data with application to hiv prevalence estimation. *Journal of the American Statistical Association*, 106(493):135–146, 2011.

[3] K.J. Gile and M.S. Handcock. Respondent-driven sampling: An assessment of current methodology. *Arxiv preprint arXiv:0904.1855*, 2009.

[4] Krista J Gile and Mark S Handcock. Network model-assisted inference from respondent-driven sampling data. *arXiv preprint arXiv:1108.0298*, 2011.

[5] Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, Pavel N. Krivitsky, and Martina Morris. *ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks.* Seattle, WA, 2012. Version 3.0-3. Project home page at urlstatnet.org.

[6] Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris. *statnet: Software tools for the Statistical Modeling of Network Data.* Seattle, WA, 2003.

[7] Kenneth Lange. *Numerical Analysis for Statisticians: Second Edition.* Springer, 2010.

[8] JJ Potterat, DE Woodhouse, SQ Muth, R Rothenberg, WW Darrow, AS Klovdahl, and JB Muth. Network dynamism: History and lessons of the colorado springs study. *Network epidemiology: a handbook for survey design and data collection, ed. M. Morris, New York: Oxford University Press*, pages 87–114, 2004.

[9] John A. Rice. *Mathematical Statistics and Data Analysis: Third Edition.* Brooks/Cole, 2007.

[10] Richard B Rothenberg, John J Potterat, Donald E Woodhouse, Stephen Q Muth, William W Darrow, and Alden S Klovdahl. Social network dynamics and hiv transmission. *Aids*, 12(12):1529–1536, 1998.

[11] Richard B Rothenberg, Donald E Woodhouse, John J Potterat, Stephen Q Muth, William W Darrow, and Alden S Klovdahl. Social networks in disease transmission: the colorado springs study. *NIDA research monograph*, 151:3–19, 1995.

[12] Matthew J Salganik and Douglas D Heckathorn. Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological methodology*, 34(1):193–240, 2004.

[13] R Core Team. Writing r extensions. *R Foundation for Statistical Computing*, 1999.

[14] Steven K. Thompson. *Sampling: Second Edition*. Wiley, 2002.

[15] Erik Volz and Douglas D Heckathorn. Probability based estimation theory for respondent driven sampling. *Journal of Official Statistics*, 24(1):79, 2008.

[16] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and Applications*. Cambridge University Press, 1994.

[17] Wikipedia. Ranking, 2014. [Online; accessed 27-January-2014].

[18] Donald E Woodhouse, Richard B Rothenberg, John J Potterat, William W Darrow, Stephen Q Muth, Alden S Klovdahl, Helen P Zimmerman, Helen L Rogers, Tammy S Maldonado, John B Muth, et al. Mapping a social network of heterosexuals at high risk for hiv infection. *Aids*, 8(9):1331–1336, 1994.