2021

Variance Approximation Approaches For The Local Pivotal Method

Theodore Edward Owen

Variance Approximation Approaches For The Local Pivotal Method

By

Theodore Edward Owen

B.S., Idaho State University, Pocatello, ID, 2008
M.S., Idaho State University, Pocatello, ID, 2011

Dissertation

presented in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in Mathematics

The University of Montana
Missoula, MT

April 2021

Approved by:

Ashby Kinch, Associate Dean of Graduate School
Graduate School

Dr. David Patterson, Chair
Mathematical Sciences

Dr. David Affleck
Forest Management

Dr. Jonathan Graham
Mathematical Sciences

Dr. Brian Steele
Mathematical Sciences

Dr. Johnathan Bardsley
Mathematical Sciences

Owen, Theodore, PhD., April 2021 Mathematics

Variance Approximation Approaches For The Local Pivotal Method

Chairperson: Dr. David Patterson

The problem of estimating the variance of the Horvitz–Thompson estimator of the population total when selecting a sample with unequal inclusion probabilities using the local pivotal method is discussed and explored. Samples are selected using unequal inclusion probabilities so that the estimates using the Horvitz–Thompson estimator will have smaller variance than for simple random samples. The local pivotal method is one sampling method which can select samples with unequal inclusion probability without replacement. The local pivotal method also balances on other available auxiliary information so that the variability in estimates can be reduced further.

A promising variance estimator, bootstrap subsampling, which combines bootstrapping with rescaling to produce estimates of the variance is described and developed. This new variance estimator is compared to other estimators such as naive bootstrapping, the jackknife, the local neighborhood variance estimator of Stevens and Olsen, and the nearest neighbor estimator proposed by Grafström.

For five example populations, we compare the performance of the variance estimators. The local neighborhood variance estimator performs best where it is appropriate. The nearest neighbor estimator performs second best and is more widely applicable. The bootstrap subsample variance estimator tends to underestimate the variance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of this dissertation is to explore variance estimation for the Horvitz–Thompson estimator of the population total. The Horvitz–Thompson estimator uses an unequal inclusion probability design, and one sampling algorithm which can implement this design is the local pivotal method. In Chapter 1, we introduce sampling designs, estimators associated with those designs, and sampling algorithms. In Chapter 2, we introduce and give properties for the local pivotal method which is a sampling algorithm developed from the pivotal method, both of which are special cases of the splitting method. In Chapter 3, different variance approximation techniques are introduced and discussed. In Chapter 4, those variance approximation techniques are used on 5 example datasets to compare their performance.

1.1 Terminology

Sampling exists because in nearly all cases a census is impractical. Throughout this work, we assume that the population of interest is of size N where N is a

finite number. A sample of size n is to be selected from the population of N in some way. No assumption is made about the size of n relative to N. The different basic sampling methods which follow illustrate the common ways by which a sample is selected.

In **simple random sampling**, "...n distinct units are selected from the N units of the population in such a way that every possible combination of n units is equally likely to be the sample selected" (Thompson, 2012, p.11). Conceptually, a simple random sample can be chosen by assigning each individual in the population a card and selecting one card at a time from a well-mixed bag containing all of the cards. Once an individual's card is chosen, that card is not replaced in the bag. A simple random sample can be chosen by selecting individuals at random without replacement.

The main parameter of interest in this work is the population total, τ, which is

$$\tau = \sum_{i=1}^{N} y_i$$

where y_i is a response variable measured on individual i. When a sample is selected using simple random sampling, an unbiased estimator for the population total is

$$\hat{\tau}_{SRS} = \frac{N}{n} \sum_{i=1}^{n} y_i$$

with variance

$$\text{Var}(\hat{\tau}_{SRS}) = N(N-n)\frac{\sigma^2}{n}$$

where $\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu)^2$ is the population variance and μ is the popu-

lation mean. An unbiased estimator of the variance is

$$\widehat{\text{Var}}(\hat{\tau}_{SRS}) = N(N-n)\frac{s^2}{n}$$

where $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$ is the sample variance and $\bar{y}$ is the sample mean.

In **independent random sampling**, n units of the population are selected so that "each possible sequence of n units – distinguishing order of selection and possibly including repeat selections – has equal probability" of being selected (Thompson, 2012, p.19). Conceptually, an independent random sample can be chosen by assigning each individual in the population a card and selecting one card at a time from a well-mixed bag containing all of the cards. Once an individual's card is chosen, that card is replaced in the bag. An independent random sample can be chosen by selecting individuals with replacement, and each draw from the bag is independent of the previous draws.

An unbiased estimator for the population total under independent random sampling is

$$\hat{\tau}_{IRS} = \frac{N}{n}\sum_{i=1}^{n} y_i$$

with variance

$$\text{Var}(\hat{\tau}_{IRS}) = N(N-1)\frac{\sigma^2}{n}.$$

An unbiased estimator of the variance is

$$\widehat{\text{Var}}(\hat{\tau}_{IRS}) = N^2\frac{s^2}{n}.$$

Comparing the variances between simple random sampling and independent random sampling, for $n > 1$ the variance for simple random sampling is

smaller than the variance for independent random sampling. This is also the case for the estimators of the variance. Thus simple random sampling is generally preferred to independent random sampling. Simple random sampling is also preferred, conceptually, because the sample will always be n distinct object.

Simple random sampling and independent random sampling are examples of a sampling design combined with a sampling algorithm. A **sampling design** specifies a probability distribution on the set of possible samples. For a simple random sample, the sampling design is assigning to each possible combination of n distinct units the same probability. A **sampling algorithm** specifies the steps involved in choosing the sample. One common sampling algorithm for a simple random sample is to use a random number generator to select the sample from a numbered list of the population. A less common algorithm is selecting cards from a well–mixed bag.

The next two topics we discuss are sampling designs. They specify a probability distribution on the set of possible samples without specifying how to select the sample itself. For the first design, a common sampling algorithm exists. For the second design, there has been much work done to develop a sampling algorithm.

In an **unequal selection probability design** n units of the population are selected with replacement (as in independent random sampling) such that "on each draw the probability of selecting the ith unit of the population is p_i, for $i = 1, \ldots, N$" (Thompson, 2012, p.67), where p_i is the *selection probability* of individual i and $\sum_{i=1}^{N} p_i = 1$. One sampling algorithm for an unequal selection probability design is to select n cards from a well-mixed bag, with replacement, where the number of cards for each individual is proportional to

p_i.

An unbiased estimator of the total for an unequal selection probability design is the Hansen–Hurwitz estimator (Hansen, 1943)

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

with variance

$$\mathrm{Var}(\hat{\tau}_p) = \frac{1}{n}\sum_{i=1}^{N} p_i\left(\frac{y_i}{p_i} - \tau\right)^2.$$

An unbiased estimator for the variance is

$$\widehat{\mathrm{Var}}(\hat{\tau}_p) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(\frac{y_i}{p_i} - \hat{\tau}_p\right)^2.$$

In an **unequal inclusion probability design**, n units of the population are selected such that "[t]he probability that unit i is included... in the sample is π_i" (Thompson, 2012, p.22), where π_i is the *inclusion probability* for individual i and $\sum_{i=1}^{N}\pi_i = n$ is the desired sample size. For all of the designs so far discussed (simple random sampling design, independent random sampling design, and unequal selection probability design) inclusion probabilities can be calculated. An unequal inclusion probability design is the most general design we will discuss. Note the difference between selection probabilities and inclusion probabilities: selection probabilities specify a draw–by–draw probability and inclusion probabilities specify probabilities for samples of size n.

There is currently no standard sampling algorithm for an unequal inclusion probability design because different sampling algorithms can lead to the same inclusion probabilities. For example, simple random sampling and stratified sampling with sample sizes proportional to stratum sizes both lead to equal

inclusion probabilities for all individuals. We will discuss the development of unequal inclusion probability design sampling algorithms in the section on sampling history.

An unbiased estimator of the total for an unequal inclusion probability design is the Horvitz–Thompson estimator (Horvitz, 1952)

$$\hat{\tau}_\pi = \sum_{i=1}^{n} \frac{y_i}{\pi_i}$$

with variance

$$\mathrm{Var}(\hat{\tau}_\pi) = \sum_{i=1}^{N} \left(\frac{1-\pi_i}{\pi_i} \right) y_i^2 + \sum_{i=1}^{N} \sum_{j \neq i}^{N} \left(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_i \pi_j} \right) y_i y_j$$

where π_{ij} is the joint inclusion probability of individuals i and j. The joint inclusion probability is the probability that both individuals i and j are included in the sample. Provided that $\pi_{ij} > 0$ for all i and j, an estimator of the variance is (Horvitz, 1952)

$$\widehat{\mathrm{Var}}(\hat{\tau}_\pi) = \sum_{i=1}^{n} \left(\frac{1-\pi_i}{\pi_i^2} \right) y_i^2 + \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_i \pi_j} \right) \frac{y_i y_j}{\pi_{ij}}.$$

This estimator can sometimes be negative, so an alternative estimator is (Sen (1953) and Yates and Grundy (1953))

$$\widehat{\mathrm{Var}}_{SYG}(\hat{\tau}_\pi) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} \right) \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2.$$

Both of these estimators of the variance require that $\pi_{ij} > 0$, and both estimators are unbiased.

Unequal inclusion probabilities often arise as a result of how a sample is

chosen. For example, in line–intercept sampling the inclusion probabilities are proportional to the perpendicular widths of the objects and can be computed after the sample is collected. However, there are situations where it is advantageous to specify unequal inclusion probabilities before the sample is collected. This dissertation will address situations where we specify the inclusion probabilities before the sample is collected. For example, if we suspect that a positively–valued auxiliary variable is positively correlated with the variable of interest, then we can reduce the variance of the Horvitz–Thompson estimator of the population total by using inclusion probabilities that are proportional to that auxiliary variable. In the case where there is a perfect correlation between the auxiliary variable and the response variable, the variance is 0. Thus the stronger the correlation, the greater the reduction in variance.

Suppose that we have a positively–valued auxiliary variable, $\boldsymbol{x} = (x_1, \ldots, x_N)$, and we would like to select a sample of size n. We make no assumptions about the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$, but as noted above, the stronger the correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$ the greater the reduction in variance with the Horvitz–Thompson estimator. The standard way to construct inclusion probabilities based on $\boldsymbol{x}$ is (Deville, 1998)

$$\pi_i = n\frac{x_i}{\sum_{j=1}^{N} x_j} \tag{1.1}$$

for $i = 1, \ldots, N$. Then by construction the sum of the π_i is n, and, since none of the x_i is negative, the inclusion probabilities are all nonnegative. It is possible for π_i to be greater than 1 for some i (this occurs especially as the sample size, n, approaches the population size). In the case that a single $\pi_k \geq 1$ for some k, we let $\pi_k = 1$, then recompute equation 1.1 using $n-1$ and all x_i except x_k.

If there is more than one individual with inclusion probability greater than 1, we start with the individual with the largest inclusion probability, let that individual's inclusion probability be 1, then recompute equation 1.1 using the other x_i. Repeat this process until $\pi_i \leq 1$ for all i.

Example 1. Suppose that the auxiliary information we have collected for a population of size 8 is $\boldsymbol{x} = (1, 4, 3, 1, 6, 5, 2, 1)$, so $\sum_{i=1}^{8} x_i = 23$. Then for a sample of size 4, we compute from equation 1.1

$$\boldsymbol{\pi} = 4\frac{\boldsymbol{x}}{23} = \frac{4}{23} \cdot (1, 4, 3, 1, 6, 5, 2, 1) \approx$$

$$(0.17, 0.70, 0.52, 0.17, 1.04, 0.87, 0.35, 0.17).$$

Notice that the 5th individual has inclusion probability greater than 1, so we assign the inclusion probability of that individual to be 1, then compute the inclusion probabilities of the remaining individuals selecting a sample of size $4 - 1 = 3$. Then since $\sum_{i=1, i\neq 5}^{8} x_i = 17$,

$$\boldsymbol{\pi} = 3\frac{(1, 4, 3, 1, *, 5, 2, 1)}{17} \approx (0.18, 0.70, 0.53, 0.18, *, 0.88, 0.35, 0.18).$$

Now since all of the inclusion probabilities are less than or equal to 1, the final vector of inclusion probabilities is

$$\boldsymbol{\pi} \approx (0.18, 0.70, 0.53, 0.18, 1, 0.88, 0.35, 0.18).$$

Notice that the sum of the inclusion probabilities is 4, and every sample selected which respects these inclusion probabilities will select individual 5.

The above example demonstrates how to calculate the inclusion probabil-

ities if the auxiliary information is provided. Next we discuss an example of how to get such auxiliary information.

Example 2. This example is based on a study by Grafström (2013a). In the Remingstorp research site in the southwest of Sweden, 846 plots have been laid out on a 40 meter by 40 meter grid. These plots have been placed in forested areas, and one of the goals for the research site is to estimate the total volume of all trees at the site. Each plot consists of a circle of fixed radius of 10 meters, and every tree which is greater than 5 cm in diameter and which is inside of the circle of a plot is measured. From those measurements, we calculate the total volume of all trees at each plot, and those calculated values are used to estimate the total volume for all trees at the site.

We would like to select a sample of the 846 plots using an unequal inclusion probability design. An airborne laser scanning system capable of measuring the average height of vegetation with a resolution of 0.25 meters scanned the entire research site and collected the average vegetation height at the center of all 846 plots. The positively–valued auxiliary information, x_i, is then the average vegetation height for plot i where $i = 1, \ldots, 846$. With this information, we can now use equation 1.1 (page 7) to compute an inclusion probability vector as in example 1. We expect that the average vegetation height is positively associated with tree volume, so the Horvitz–Thompson estimate of the total volume will likely have smaller variance than an estimate of total volume using a simple random sample.

Next we consider the development of sampling and difficulties associated with developing a sampling algorithm for unequal inclusion probability designs.

1.2 History

Sampling theory's advent came from the realization that taking a census of the experimental units is often not practical. In 1895, A. N. Kiaer attempted to clarify what a "representative investigation" was and how to implement it (Seng, 1951). This was the precursor to a simple random sample. Simple random sampling is one possible sampling design. Other possible sampling designs incorporate unequal probability of selection, the first of which was described in 1926. At that same time the International Institute of Statistics advocated for two main methods of selecting a representative sample: random selection, and purposive selection. This second kind of selection is implemented when "[a] number of groups of units are selected which together yield nearly the same characteristics as the totality" (Seng, 1951, p. 223).

In 1934, Jerzy Neyman further refined the idea of purposive selection. In Neyman's notation (note the change in role of x and y from current notation), a population is stratified into M districts. An auxiliary variable, y_i, is collected on each district for $i = 1, \ldots, N$. The response variable is x, the sum of all x within district i is u_i, and the number of individuals in district i is v_i. Then

$$\bar{x_i} = \frac{u_i}{v_i}$$

is the mean response in district i. Neyman states that purposive selection should be used when "the basic hypothesis ... is that the numbers $\bar{x_i}$ are correlated with the control y_i and that the regression of $\bar{x_i}$ on y_i is linear" (Neyman, 1934, p. 571). Neyman also clarified what "the same characteristics

as the totality" means by stating the "weighted mean

$$Y' = \frac{\sum (vy)}{\sum (v)} \tag{1.2}$$

has the same value, or at least nearly the same as it is possible, as it has for the whole population, say Y" (Neyman, 1934, p. 571). The sums in equation 1.2 are over all observations within the sample. We will see this kind of criterion in later chapters when we consider "balanced" samples (see Equation 2.1 on page 40 for the definition of "balanced"). For Neyman, purposive sampling did incorporate random sampling. If two different districts contributed the same value to Y', one of those districts would be selected at random to be incorporated in the sample.

Since it seemed unlikely that many auxiliary variables would be found which have a linear relationship with a given response, Neyman concluded that using purposive selection "may give sometimes perfect results, but these will be due rather to the uncontrollable intuition of the investigator and good luck than to the method itself," and that "when using this method we are very much in the position of a gambler, betting at one time £100" (Neyman, 1934, p. 586). This critique of purposive selection helped drive statisticians toward random selection, and it would be nearly ten years before substantial progress was made in purposive selection theory.

In 1943, Morris Hansen and William Hurwitz advocated for sampling from substrata using unequal selection probabilities. They argued that a smaller error in the estimate can be found when using unequal selection probabilities and that times when this sampling strategy is useful "are frequently met in practice" (Hansen, 1943, p. 353). The Hansen–Hurwitz estimator later became

used for estimation with unequal selection probability designs.

Other statisticians began to see that situations where an unequal probability design was useful really were more frequently met than Neyman originally thought, and in 1952 Daniel Horvitz and Donovan Thompson developed the Horvitz–Thompson estimator for any sampling design where inclusion probabilities can be computed. One advantage of the Horvitz–Thompson estimator over the Hansen–Hurwitz estimator is that the Horvitz–Thompson estimator is computed for an unequal inclusion probability design which is more general (and can be applied to more sampling situations) than an unequal selection probability design. The Horvitz–Thompson estimator is sometimes called the Narain–Horvitz–Thompson estimator since R.D. Narain published similar results in 1951 (see Narain, 1951).

In estimating the variance of the Horvitz–Thompson estimator, we need π_{ij}, the joint inclusion probability of individuals i and j for all individuals in the sample. With a population of size N, there are $\binom{N}{2} = \frac{(N-1)N}{2}$ different joint inclusion probabilities. We estimate the variance by using only the inclusion probabilities of those individuals who are a part of the sample, but a sample of size 30 will have 435 different joint inclusion probabilities to compute. There is no general formula for calculating these joint inclusion probabilities. We will discuss other difficulties with estimating the variance of the Horvitz–Thompson estimator in Chapter 3.

Subsequent to the introduction of the Horvitz–Thompson estimator of the total, sampling algorithms for selecting a sample to achieve a given set of unequal inclusion probabilities abounded. According to Muhammad Hanif and K.R.W. Brewer, by 1980 "about 47 selection procedures have appeared in different research journals" (Hanif, 1980, p. 318). Finding an algorithm

for selecting a sample which satisfies the inclusion probabilities is a difficult task. In 2006, Yves Tillé claimed that of the 47 sampling algorithms presented by Hanif and Brewer "only 20 of them really work and many of the exact procedures are very slow to apply" (Tillé, 2006, p. 2). To say that a procedure "really works" is to say that the probability that the sample includes individual i is π_i.

One difficulty of the different algorithms is that, as Tillé points out, "[e]ach one of the unequal probability sampling methods... has particular joint inclusion probabilities," (Tillé, 2006, p. 137), so each algorithm has a different variance. This is a critical point: different algorithms for selecting an unequal inclusion probability sample will generally have different variances because different algorithms do not generally have the same π_{ij}. The choice of sampling algorithm does not depend only on satisfying a set of prescribed inclusion probabilities but also on other desired properties of the sample, such as "balance" on auxiliary variables (the specifics of "balance" will be discussed in the next chapter, see page 40). Since many of the different sampling algorithms have different variances, further sample properties, like "balance," can act to differentiate the different methods.

To summarize, an unequal inclusion probability design offers the most general design we have discussed while also potentially having small variance for the estimated total. The difficulty at this point in our exposition is to find a sampling algorithm which will respect a given unequal inclusion probability design and select "balanced" samples. In the next chapter, we will describe such an algorithm and discuss the properties which this algorithm should satisfy.

Chapter 2

The Pivotal and Local Pivotal Methods

2.1 Introduction

Our overarching goal is to estimate the variance of the population total, $\tau = \sum_{i=1}^{N} y_i$, using the Horvitz–Thompson estimator, $\hat{\tau}_\pi = \sum_{i=1}^{n} \frac{y_i}{\pi_i}$. This requires selecting a sample using an unequal inclusion probability design. The mechanics of actually selecting a sample when given a vector of inclusion probabilities is a difficult problem. In the previous chapter we noted that there were about 47 different algorithms developed by the 1980's, all of which claimed to be able to select a sample according to a given inclusion probability vector. New algorithms are still being developed, and for this work we are focused on the local pivotal method (Grafström, 2012) which is a special case of the pivotal method (Deville, 1998) which is itself a special case of the splitting method (Deville, 1998). To develop the properties of the local pivotal method, we first consider the splitting method and its properties.

2.2 The Splitting Method

For a population of size N, we consider the vector of inclusion probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$, where $0 \leq \pi_i \leq 1$ for $i = 1, \ldots, N$. We assume that the population is of finite size and the inclusion probability can be specified for each element of the population. The sample size is $n = \sum_{i=1}^{N} \pi_i$. A splitting method takes the original vector of probabilities and splits it into two or more different vectors of probabilities. The goal is for each split to reduce the complexity of the inclusion probability vectors. The following example shows a single split into three vectors.

Example 3. Suppose that an inclusion probability vector is $\boldsymbol{\pi} = (0.4, 0.2, 0.8, 0.6)$. Since $\sum_{i=1}^{4} \pi_i = 2$, we are trying to take a sample of size 2 from this population of 4 individuals. We split this probability vector into 3 vectors as in Figure 2.1. Two of the three resulting inclusion probability vectors are

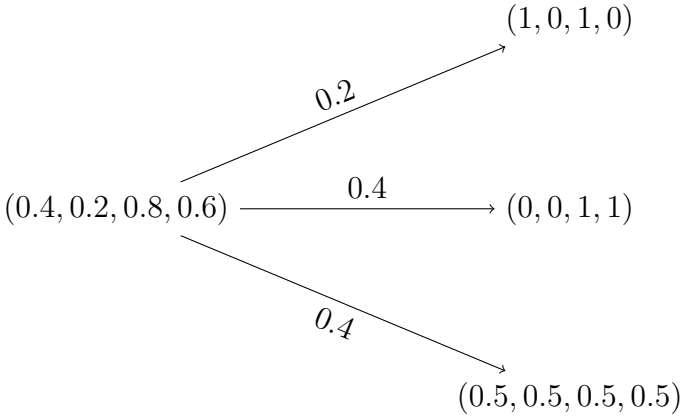

Figure 2.1: Splitting original inclusion probability vector into 3 vectors

simpler than the original probability vector because there are now individuals from the population who are either selected for the sample (those with an inclusion probability of 1) or excluded from the sample (those with an inclusion

probability of 0). Along each arrow in Figure 2.1, the value on the length of the arrow is the probability that we would choose that branch of the split.

With probability 0.2, we would choose the topmost branch. Since the topmost branch has inclusion probability vector $(1, 0, 1, 0)$, we would choose a sample which consists of individuals 1 and 3 if this branch were chosen. With probability 0.4, we would choose the middle branch, which would result in a sample of individuals 3 and 4 from the population.

With probability 0.4, we would choose the bottommost branch with resulting inclusion probability vector $(0.5, 0.5, 0.5, 0.5)$. This probability vector no longer requires us to select a sample using an unequal inclusion probability design. We could use an equal inclusion probability design to select the sample. One possible design for selecting this sample is a simple random sample design.

No matter the resulting branch which is chosen, the problem of selecting a sample is simpler. The overall goal of splitting methods is to perform many splits which reduce the complexity of the inclusion probability vectors finally resulting in vectors containing all 0's or 1's or with equal non–zero inclusion probabilities. In this example we only perform one split because we then have a way to select the sample no matter which branch is chosen.

We decided to use a value of 0.2 for the probability of the topmost branch because we wanted the topmost inclusion probability vector to select individuals 1 and 3 as the sample. We also decided to use a value of 0.4 for the probability of the middle branch because we wanted the middle inclusion probability vector to select individuals 3 and 4 as the sample. These decisions then dictated how the bottommost resulting inclusion probability vector would be formed. This is only one example of how a split into three vectors could oc-

cur. If we had wanted different topmost and middle resulting vectors, then we would form a different split.

Consider Figure 2.1 (page 15) as a diagram for a discrete random variable where there are three possible outcomes (the different branches which dictate the number of times an individual in the population is selected) with the probability of each outcome given on the corresponding arrow. Then the expected number of times that an individual in the population is chosen is the sum across each branch of the product of the values in the resulting inclusion probability vector with the probability of choosing that vector. We would expect to choose individual 1 an expected $0.2 \cdot 1 + 0.4 \cdot 0 + 0.4 \cdot 0.5 = 0.4$ times. Note that for all 4 individuals in the population, the expected number of times an individual is chosen is exactly the inclusion probability for that individual from the original inclusion probability vector. This property is how splitting methods preserve the original inclusion probability.

For the methods we will discuss in the rest of this work, we will only split into two vectors because we want the resulting unequal inclusion probability vectors to be different from the original vector for only 2 individuals. The following example demonstrates this splitting into two branches.

Example 4. Consider the same inclusion probability vector from Example 3 (page 15), $\boldsymbol{\pi} = (0.4, 0.2, 0.8, 0.6)$. Since $\sum_{i=1}^{4} \pi_i = 2$, we are still trying to take a sample of size 2 from a population of size 4. A splitting method which splits this vector into 2 branches could look like Figure 2.2 (page 18). With probability 0.6 we would choose the top branch of the first split. This would result in the inclusion of individual 4 in the final sample and the exclusion of individual 1 in the final sample. Unlike in Example 3, we now need to select

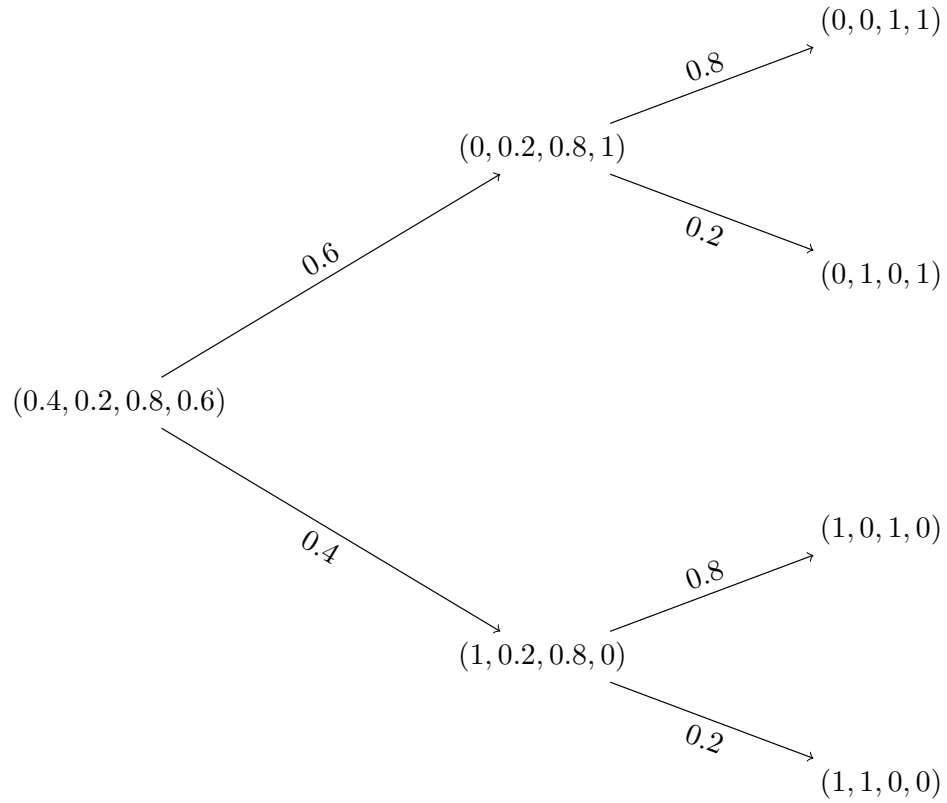

Figure 2.2: Final splitting diagram for splitting into 2 vectors

a sample using an unequal inclusion probability design for individuals 2 and 3. To do this we can perform another split, and with probability 0.8 we would choose the top branch again to arrive at a sample of individuals 3 and 4.

Note that each branch only changes the inclusion probability of two individuals from the previous inclusion probability vector. This property is what causes us to use two branches at each split.

Figure 2.2 is an example of a splitting method. It specifies how to take the original inclusion probability vector and select a sample by splitting. It is a very limited splitting method because it only indicates how to split this one inclusion probability vector. For larger populations and for arbitrary inclusion probability vectors, an algorithm specifies how each branch is formed. This will

be demonstrated immediately following Example 5. For smaller populations, we can use a diagram like Figure 2.2. Many different splitting methods exist based on the goal of the splitting. For this example, the goal was to modify the inclusion probability of two individuals at each step so that the resulting inclusion probabilities were 1 or 0.

With the final diagram of Figure 2.2 we can compute the joint inclusion probabilities, π_{ij}, for the individuals in this population. We have that $\pi_{34} = 0.6 \cdot 0.8 = 0.48$, since there is only one branch that ends with a sample of both individuals 3 and 4. Also, $\pi_{14} = 0$ since there are no branches which end with a sample of both individuals 1 and 4. This is one possible splitting method which satisfies the original inclusion probability vector. Another possible splitting method is presented in the next example.

Example 5. Using the same inclusion probability vector as in Examples 3 (page 15) and 4 (page 17), we could have the splitting method which still splits into two branches as shown in Figure 2.3 (page 20). This splitting method takes a sample of size 2 from the original population of size 4, and respects the original inclusion probability vector. Notice that this splitting method has different joint inclusion probabilities than the splitting method in Example 4. For this splitting method, we have $\pi_{34} = 0.\bar{6} \cdot 0.5 \cdot 0.8 + 0.\bar{3} \cdot 0.5 \cdot 0.8 = 0.4$ since there are now two different branches which select a sample of both individuals 3 and 4. Compare this to a joint inclusion probability of $\pi_{34} = 0.48$ for Example 4. Also, in this example we have $\pi_{14} = 0.\bar{6} \cdot 0.5 \cdot 0.2 + 0.\bar{6} \cdot 0.5 \cdot 0.2 = 0.1\bar{3}$. Compare this to a joint inclusion probability of $\pi_{14} = 0$ for Example 4.

This example and Example 4 show that two different splitting methods can satisfy the same original inclusion probability vector and have different joint

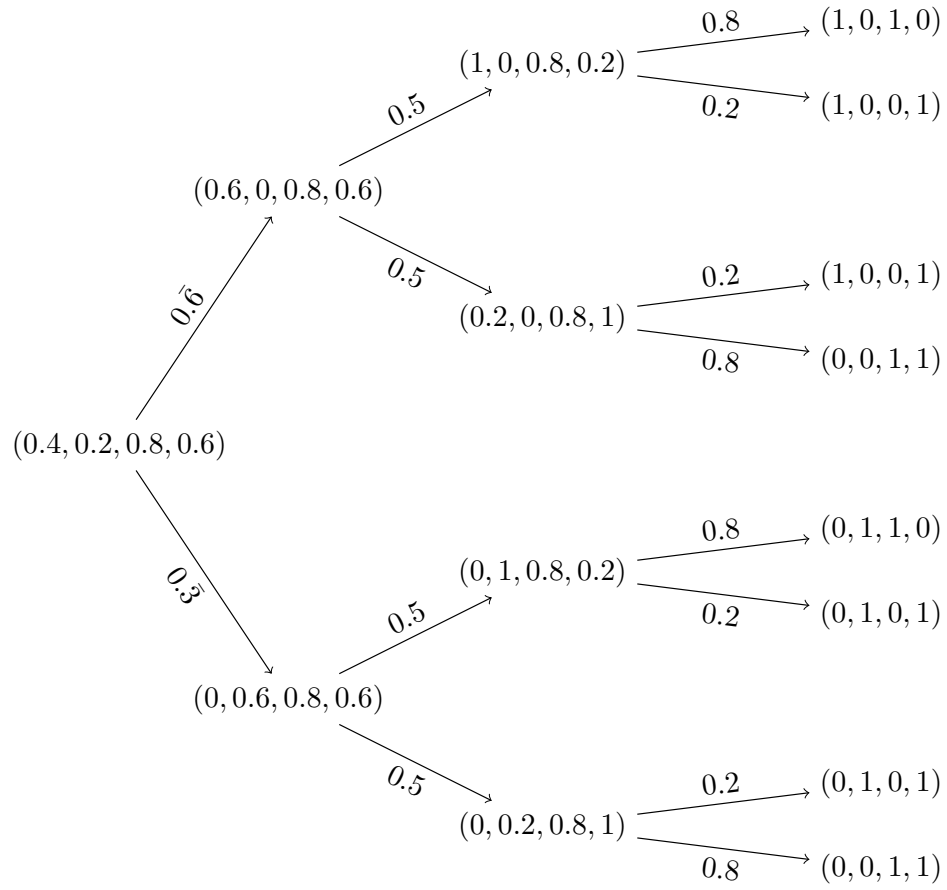

Figure 2.3: Final splitting diagram for splitting into 2 vectors with alternative splitting method

inclusion probabilities. This example also highlights the difficulty involved in computing the joint inclusion probabilities for any given splitting method: we need to have the complete splitting diagram in order to calculate the joint inclusion probabilities. This is manageable for a small population, such as a population of size 4, but quickly becomes unmanageable for larger populations.

For a general splitting method which splits into two branches we denote the two different resulting vectors as $\boldsymbol{\pi^a} = (\pi_1^a, \pi_2^a, \ldots, \pi_N^a)$ and $\boldsymbol{\pi^b} = (\pi_1^b, \pi_2^b, \ldots, \pi_N^b)$, chosen based on the goals of the particular method. For example, if our goal is to select a simple random sample with the $\boldsymbol{\pi^a}$ vector, then we would

require that $\pi_i^a = \pi_j^a$ for all i, j in $1, \ldots, N$.

For now we assume that the sum of the inclusion probabilities for each vector $\boldsymbol{\pi}$, $\boldsymbol{\pi^a}$, and $\boldsymbol{\pi^b}$ is an integer. We select $\boldsymbol{\pi^a}$ with probability α and $\boldsymbol{\pi^b}$ with probability $1 - \alpha$, where $0 < \alpha < 1$ and α is freely chosen. Further we require that $\pi_i = \alpha\pi_i^a + (1 - \alpha)\pi_i^b$, for $i = 1, \ldots, N$.

A schematic for this splitting procedure can be seen in Figure 2.4. For

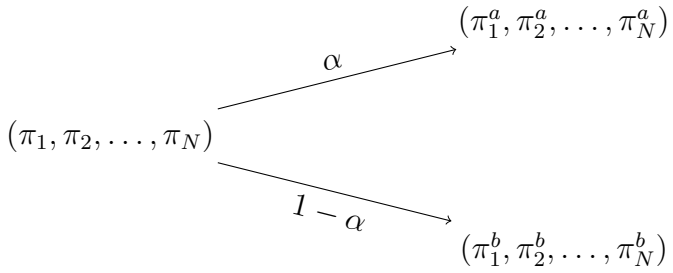

Figure 2.4: Splitting into two vectors for generic splitting method

a specific implementation of the splitting method, we specify the goal of the resulting vectors, then choose an α. The choice of α will be discussed further starting on the top of page 24.

Regardless of the specific splitting method involved, we generally want a splitting method to preserve the total of the inclusion probabilities of the initial inclusion probability vector. To do this we have two further constraints on the resulting vectors. We require that

$$(1) \quad \sum_{i=1}^{N} \pi_i^a = \sum_{i=1}^{N} \pi_i^b = \sum_{i=1}^{N} \pi_i = n$$

and

$$(2) \quad 0 \leq \pi_i^a \leq 1 \text{ and } 0 \leq \pi_i^b \leq 1, \text{for each } i = 1, \ldots, N.$$

Note that when π_i^a or π_i^b is 0, individual i will not be selected if $\boldsymbol{\pi}^a$ or $\boldsymbol{\pi}^b$,

respectively, is chosen (and vice versa for a resulting probability of 1). A tacit assumption of splitting methods is that once an individual has inclusion probability 1 or 0 the inclusion probability of that individual is no longer modified in subsequent splits. Thus, for some individual i, if $\pi_i = 0$, we have that $\pi_i^a = 0$ and $\pi_i^b = 0$ (and similarly if $\pi_i = 1$). This implies that, once an individual has been included or excluded at a given split, the individual will be included or excluded in all subsequent splits.

To illustrate the necessity (and independence) of constraints (1) and (2), consider the following example.

Example 6. Suppose that $\boldsymbol{\pi} = (0.2, 0.6, 0.4, 0.1, 0.7, 0.5, 0.5)$ (note that $\sum_{i=1}^{7} \pi_i = 3 = n$), then for $\alpha = 0.5$, we could have $\boldsymbol{\pi}^a = (0, 1, 0, 0, 1, 0, 0)$ and $\boldsymbol{\pi}^b = (0.4, 0.2, 0.8, 0.2, 0.4, 1, 1)$, which satisfies $\boldsymbol{\pi} = 0.5\boldsymbol{\pi}^a + 0.5\boldsymbol{\pi}^b$. To construct this example, we specified that $\boldsymbol{\pi}^{\boldsymbol{a}}$ would select individuals 2 and 5, then arbitrarily chose an α of 0.5. Then with probability 0.5 we would select the resulting vector which samples only individuals 2 and 5 from the population (and $\sum_{i=1}^{7} \pi_i^a = 2$), and with probability 0.5 we would select the resulting vector which guarantees to sample individuals 6 and 7 and samples the other individuals with unequal inclusion probabilities (and $\sum_{i=1}^{7} \pi_i^b = 4$). We have then satisfied constraint (2) and all of the properties of the splitting method, but have failed to satisfy constraint (1).

For the same $\alpha = 0.5$, we could have $\boldsymbol{\pi}^a = (0, 1, 1, 0, 1, 0, 0)$ and $\boldsymbol{\pi}^b = (0.4, 0.2, -0.2, 0.2, 0.4, 1, 1)$. Then $\boldsymbol{\pi} = 0.5\boldsymbol{\pi}^a + 0.5\boldsymbol{\pi}^b$, and we have now specified that individuals $2, 3$, and 5 are sampled by $\boldsymbol{\pi}^{\boldsymbol{a}}$. Constraint (1) is satisfied since $\sum_{i=1}^{7} \pi_i^a = \sum_{i=1}^{7} \pi_i^b = \sum_{i=1}^{7} \pi_i = 3$, but constraint (2) is now violated since $\pi_3^b = -0.2 \notin [0, 1]$. These two different splits show that the two addi-

tional constraints are independent and necessary.

If we split $\boldsymbol{\pi} = (0.2, 0.6, 0.4, 0.1, 0.7, 0.5, 0.5)$ into $\boldsymbol{\pi}^a = (0, 1, 0, 0, 1, 1, 0)$ and $\boldsymbol{\pi}^b = (0.4, 0.2, 0.8, 0.2, 0.4, 0, 1)$ with $\alpha = 0.5$, then we should split $\boldsymbol{\pi^b}$ so that we are not forced to take a sample with unequal inclusion probabilities. Since $\boldsymbol{\pi^a}$ identifies a sample, there is no reason to split it again. There is nothing about this splitting process which prevents splitting on the resulting probability vectors. We will follow the convention that t represents the split number, and the dependence of each probability vector and each α on t will be shown by a subscript. We start with $\boldsymbol{\pi} = \boldsymbol{\pi}_0$, and for each $t = 0, 1, 2, \ldots$, we have the notation given in Figure 2.5.

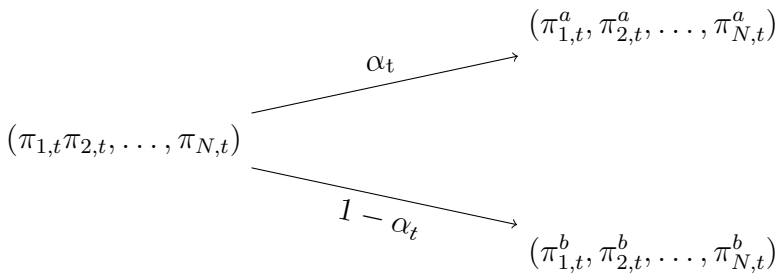

Figure 2.5: Splitting into two vectors at step t for generic splitting method

We then have $\boldsymbol{\pi}_{t+1} = \begin{cases} \boldsymbol{\pi}^a_t \text{ with probability } \alpha_t \\ \boldsymbol{\pi}^b_t \text{ with probability } 1 - \alpha_t. \end{cases}$

The process of splitting continues until a satisfactory pair of vectors is achieved. For example, we could stop when both the resulting vectors are such that all of the individuals with nonzero or non–one inclusion probability have the same probability, in which case we can take a simple random sample of those remaining individuals. We can also proceed with the splitting process until the resulting probability vectors have values which are all either 0 or 1, in which case the sample is determined for each vector.

The choice of these resulting vectors helps determine the choice of α. For example, in "splitting into simple random sampling" (Deville, 1998), we specify that $\boldsymbol{\pi}^{\boldsymbol{a}}$ should select a simple random sample. We choose $\alpha = \min\{\pi_{(1)}\frac{N}{n}, \frac{N}{N-n}(1-\pi_{(N)})\}$ where $\pi_{(1)}$ and $\pi_{(N)}$ are the smallest and largest values in $\boldsymbol{\pi}$, respectively. Then the result of the requirement that $\pi_i = \alpha\pi_i^a + (1-\alpha)\pi_i^b$ is that

$$\pi_i^a = \frac{n}{N} \text{ and } \pi_i^b = \frac{\pi_i - \alpha(\frac{n}{N})}{1-\alpha}$$

for $i = 1, \ldots, N$. This example and the pivotal and local pivotal methods which follow are examples of "[s]plitting methods based on the choice of $\boldsymbol{\pi}^{\boldsymbol{a}}$" (Tillé, 2006, p. 101). Another way to specify the splitting method is "[s]plitting method based on the choice of a direction" (Tillé, 2006, p. 102). We will not go into the details of this second method, but the general idea is to specify an increase or decrease in value from π_i to π_i^a. For example, if we specified that π_1^a is greater than π_1 and π_2^a is smaller than π_2, then that direction determines the splitting method.

Splitting methods can be used as the sampling algorithm for many unequal inclusion probability designs. Examples include the Generalized Sunter Method (Sunter, 1977), Brewer's Method (Brewer, 1963), Tillé's Elimination Method (Tillé, 1996), the Generalized Midzuno Method (Midzuno, 1950), and Chao's Method (Chao, 1982) (see Tillé, 2006, for ways to realize these as splitting methods).

2.2.1 Splitting Method Properties

Splitting into two vectors has the following properties:

Theorem 2.2.1. *(Tillé, 2006, p.101) For any $t \geq 1$, $E[\boldsymbol{\pi}_t|\boldsymbol{\pi}_{t-1}, \ldots, \boldsymbol{\pi}_0] = \boldsymbol{\pi}_{t-1}$.*

Proof. Since $\mathrm{E}[\boldsymbol{\pi}_t|\boldsymbol{\pi}_{t-1}, \ldots, \boldsymbol{\pi}_0]$ is the expected value of $\boldsymbol{\pi}_t$ after we have selected the probability vectors $\boldsymbol{\pi}_0$ up to $\boldsymbol{\pi}_{t-1}$, we treat $\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{t-1}$ as fixed. From $\boldsymbol{\pi}_{t-1}$ there are two vectors which occur: $\boldsymbol{\pi}^a_{t-1}$ with probability α_{t-1}, and $\boldsymbol{\pi}^b_{t-1}$ with probability $1 - \alpha_{t-1}$, so the conditional expectation is $\mathrm{E}[\boldsymbol{\pi}_t|\boldsymbol{\pi}_{t-1}, \ldots, \boldsymbol{\pi}_0] = (\alpha_{t-1})\boldsymbol{\pi}^a_{t-1} + (1-\alpha_{t-1})\boldsymbol{\pi}^b_{t-1}$. A defining property of splitting methods which split into two vectors is that $\pi_{i,t-1} = (\alpha_{t-1})\pi^a_{i,t-1} + (1 - \alpha_{t-1})\pi^b_{i,t-1}$ for $i = 1, \ldots, N$, so it follows that $(\alpha_{t-1})\boldsymbol{\pi}^a_{t-1} + (1 - \alpha_{t-1})\boldsymbol{\pi}^b_{t-1} = \boldsymbol{\pi}_{t-1}$.

Therefore $\mathrm{E}[\boldsymbol{\pi}_t|\boldsymbol{\pi}_{t-1}, \ldots, \boldsymbol{\pi}_0] = \boldsymbol{\pi}_{t-1}$. □

Corollary 2.2.1. *For any $t \geq 0$, $E[\boldsymbol{\pi}_t] = \boldsymbol{\pi}_0 = \boldsymbol{\pi}$.*

Proof. We know that for random variables X and Y, $\mathrm{E}[\mathrm{E}[Y|X]] = \mathrm{E}[Y]$ (DeGroot, 2012). Then by Theorem 2.2.1 above, $\mathrm{E}[\boldsymbol{\pi}_t|\boldsymbol{\pi}_{t-1}, \ldots, \boldsymbol{\pi}_0] = \boldsymbol{\pi}_{t-1}$, so taking expectation on both sides yields

$$\mathrm{E}[\boldsymbol{\pi}_t] = \mathrm{E}[\mathrm{E}[\boldsymbol{\pi}_t|\boldsymbol{\pi}_{t-1}, \ldots, \boldsymbol{\pi}_0]] = \mathrm{E}[\boldsymbol{\pi}_{t-1}], \qquad t = 1, 2, \ldots$$

Thus $\mathrm{E}[\boldsymbol{\pi}_t] = \mathrm{E}[\boldsymbol{\pi}_0]$ for $t \geq 1$. Note that $\boldsymbol{\pi}_0 = \boldsymbol{\pi}$ is fixed, so $\mathrm{E}[\boldsymbol{\pi}_0] = \mathrm{E}[\boldsymbol{\pi}] = \boldsymbol{\pi}$. □

Corollary 2.2.1 states that the splitting method respects the original probability vector in expectation at each step in the splitting process.

2.3 The Pivotal Method

The pivotal method (Deville, 1998) is a method to select a sample with specified inclusion probabilities which takes at most N steps, preserves the original inclusion probabilities in expectation at each step of the procedure, and only modifies two inclusion probabilities at each step. It proceeds by making a decision about inclusion or exclusion of at least one individual of the population at each step. To preserve the original inclusion probabilities at each step in expectation, we use a splitting method to select the sample, and, since we want to only modify two inclusion probabilities at each step, we split into two vectors. Finally, to ensure that the method is not biased in the way that the two individuals are selected, we will choose at random the two individuals whose inclusion probabilities will be modified.

The details of the pivotal method are given below; first note that since the steps in the pivotal method are the same at each iteration, we will describe how one step in the process occurs and suppress the subscript t notation to simplify the presentation. It is also not immediately clear that the method presented is a valid splitting method. This will be proven after the method is described.

Let $U = \{1, 2, \ldots, N\}$ and j and k be the indices of two randomly selected individuals (so that $j \in U$ and $k \in U$ with $j \neq k$). Then, depending on the value of $\pi_j + \pi_k$, we have two different cases for the split. The reason for these cases is that a splitting method needs to ensure that π_i^a and π_i^b are within $[0, 1]$ for all $i \in U$.

If $\pi_j + \pi_k \geq 1$, then $\alpha = \frac{(1-\pi_k)}{(2-\pi_j-\pi_k)}$, and

$$\pi_i^a = \begin{cases} \pi_i & \text{if } i \in U \setminus \{j,k\}, \\ 1 & \text{if } i = j, \\ \pi_j + \pi_k - 1 & \text{if } i = k, \end{cases} \qquad \pi_i^b = \begin{cases} \pi_i & \text{if } i \in U \setminus \{j,k\}, \\ \pi_j + \pi_k - 1 & \text{if } i = j, \\ 1 & \text{if } i = k. \end{cases}$$

If $\pi_j + \pi_k < 1$, then $\alpha = \frac{\pi_j}{(\pi_j+\pi_k)}$, and

$$\pi_i^a = \begin{cases} \pi_i & \text{if } i \in U \setminus \{j,k\}, \\ \pi_j + \pi_k & \text{if } i = j, \\ 0 & \text{if } i = k, \end{cases} \qquad \pi_i^b = \begin{cases} \pi_i & \text{if } i \in U \setminus \{j,k\}, \\ 0 & \text{if } i = j, \\ \pi_j + \pi_k & \text{if } i = k. \end{cases}$$

By construction in either case, we are guaranteed that π_i^a and π_i^b are within $[0,1]$ for all $i \in U$.

Lemma 2.3.1. *If $1 \leq \pi_j + \pi_k < 2$, then $\alpha \in [0,1]$ where $\alpha = \frac{(1-\pi_k)}{(2-\pi_j-\pi_k)}$. In the case that $\pi_j + \pi_k = 2$, no split is required.*

Proof. Observe that since $\alpha = \frac{(1-\pi_k)}{(1-\pi_k)+(1-\pi_j)}$ and $1 - \pi_j \geq 0$ and $1 - \pi_k \geq 0$ we get that $\alpha \in [0,1]$.

If $\pi_j + \pi_k = 2$, then it must be the case that $\pi_j = \pi_k = 1$. Since $\pi_j = \pi_k = 1$, both individuals j and k will be included in the sample generated by the original probability vector. Since the pivotal method makes a decision about inclusion or exclusion of the individuals chosen at each step (here the j and k), no decision needs to be made. Thus there is no reason to form a split. □

Lemma 2.3.2. *If $\pi_j + \pi_k < 1$, then $\alpha \in [0,1]$ where $\alpha = \frac{\pi_j}{(\pi_j+\pi_k)}$. In the case*

that $\pi_j + \pi_k = 0$, no split is required.

Proof. Since $0 \leq \pi_k,\ \pi_j \leq \pi_j + \pi_k$, so $\frac{\pi_j}{(\pi_j+\pi_k)} \leq 1$. Thus $\alpha \leq 1$. Since $\pi_j, \pi_k \geq 0$, $\frac{\pi_j}{(\pi_j+\pi_k)} \geq 0$, so $\alpha \geq 0$. Thus $\frac{\pi_j}{(\pi_j+\pi_k)} \in [0,1]$.

If $\pi_j + \pi_k = 0$, then argue as in the proof of Lemma 2.3.1 concluding that there is no reason to form a split because neither individuals j nor k will be included in the sample generated from the original probability vector. □

A consequence of Lemma 2.3.1 and Lemma 2.3.2 is that when the inclusion probability of an individual reaches 0 or 1 that individual should no longer be considered for selection in subsequent stages of the algorithm. Eliminating those individuals from consideration allows the method to select a sample in at most N steps. It also implies that the pivotal method selects a sample without replacement, because an individual with inclusion probability 1 will only be selected one time.

Lemma 2.3.3. *If $\pi_j + \pi_k < 1$ or $\pi_j + \pi_k \geq 1$, then $\pi_i = \alpha\pi_i^a + (1-\alpha)\pi_i^b$ for all $i \in U$, and $\sum_{i=1}^N \pi_i^a = \sum_{i=1}^N \pi_i^b = \sum_{i=1}^N \pi_i$.*

Proof. If $\pi_j + \pi_k < 1$, then $\alpha = \frac{\pi_j}{\pi_j+\pi_k}$ and there are three cases to consider for different i:

(1) if $i \in U \setminus \{j,k\}$ then

$$\alpha\pi_i^a + (1-\alpha)\pi_i^b = \alpha\pi_i + (1-\alpha)\pi_i = \pi_i,$$

(2) if $i = j$ then

$$\alpha\pi_i^a + (1-\alpha)\pi_i^b = \frac{\pi_j}{\pi_j+\pi_k}(\pi_j+\pi_k) + (1 - \frac{\pi_j}{\pi_j+\pi_k})(0) = \pi_j,$$

(3) if $i = k$ then

$$\alpha\pi_i^a + (1-\alpha)\pi_i^b = \frac{\pi_j}{\pi_j + \pi_k}(0) + (1 - \frac{\pi_j}{\pi_j + \pi_k})(\pi_j + \pi_k) = \pi_k.$$

If $\pi_j + \pi_k \geq 1$, then $\alpha = \frac{1-\pi_k}{2-\pi_j-\pi_k}$ and there are three cases to consider for different i:

(1) if $i \in U \setminus \{j, k\}$ then

$$\alpha\pi_i^a + (1-\alpha)\pi_i^b = \alpha\pi_i + (1-\alpha)\pi_i = \pi_i,$$

(2) if $i = j$ then

$$\alpha\pi_i^a + (1-\alpha)\pi_i^b = \frac{1-\pi_k}{2-\pi_j-\pi_k}(1) + \left(1 - \frac{1-\pi_k}{2-\pi_j-\pi_k}\right)(\pi_j + \pi_k - 1) =$$

$$\frac{1-\pi_k}{2-\pi_j-\pi_k} + \left(\frac{2-\pi_j-\pi_k-1+\pi_k}{2-\pi_j-\pi_k}\right)(\pi_j + \pi_k - 1) =$$

$$\frac{1-\pi_k}{2-\pi_j-\pi_k} + \left(\frac{1-\pi_j}{2-\pi_j-\pi_k}\right)(\pi_j + \pi_k - 1) =$$

$$\frac{1-\pi_k+\pi_j+\pi_k-1-\pi_j^2-\pi_j\pi_k+\pi_j}{2-\pi_j-\pi_k} =$$

$$\frac{(2-\pi_j-\pi_k)\pi_j}{2-\pi_j-\pi_k} = \pi_j,$$

(3) if $i = k$ then

$$\alpha\pi_i^a + (1-\alpha)\pi_i^b = \frac{1-\pi_k}{2-\pi_j-\pi_k}(\pi_j + \pi_k - 1) + \left(1 - \frac{1-\pi_k}{2-\pi_j-\pi_k}\right)(1) =$$

$$\frac{\pi_j+\pi_k-1-\pi_j\pi_k-\pi_k^2+\pi_k}{2-\pi_j-\pi_k} + \frac{2-\pi_j-\pi_k-1+\pi_k}{2-\pi_j-\pi_k} =$$

$$\frac{(2-\pi_j-\pi_k)\pi_k}{2-\pi_j-\pi_k} = \pi_k.$$

Thus $\pi_i = \alpha\pi_i^a + (1-\alpha)\pi_i^b$ for all $i \in U$.

Now if $\pi_j + \pi_k < 1$, then

$$\sum_{i=1}^{N} \pi_i^a = (\pi_j + \pi_k) + 0 + \sum_{i \in U \setminus \{j,k\}} \pi_i = \sum_{i=1}^{N} \pi_i, \text{ and}$$

$$\sum_{i=1}^{N} \pi_i^b = 0 + (\pi_j + \pi_k) + \sum_{i \in U \setminus \{j,k\}} \pi_i = \sum_{i=1}^{N} \pi_i.$$

If $\pi_j + \pi_k \geq 1$, then

$$\sum_{i=1}^{N} \pi_i^a = 1 + (\pi_j + \pi_k - 1) + \sum_{i \in U \setminus \{j,k\}} \pi_i = \sum_{i=1}^{N} \pi_i, \text{ and}$$

$$\sum_{i=1}^{N} \pi_i^b = (\pi_j + \pi_k - 1) + 1 + \sum_{i \in U \setminus \{j,k\}} \pi_i = \sum_{i=1}^{N} \pi_i.$$

Thus

$$\sum_{i=1}^{N} \pi_i^a = \sum_{i=1}^{N} \pi_i = \sum_{i=1}^{N} \pi_i^b.$$

□

Theorem 2.3.1. *The pivotal method is a valid splitting method, and thus preserves the original inclusion probabilities in expectation at each step of the process.*

Proof. By construction, the pivotal method splits the original probability vector into two resulting vectors, and, for each i, $0 \leq \pi_i^a \leq 1$ and $0 \leq \pi_i^b \leq 1$. From Lemmas 2.3.1 (page 27) and 2.3.2 (page 27), the value of α is guaranteed to be in $[0, 1]$. By Lemma 2.3.3 (page 28), the sum of the inclusion probabil-

ities for each of the resulting vectors is the same as the sum for the original probability vector, and π_i^a and π_i^b satisfy that $\pi_i = \alpha\pi_i^a + (1-\alpha)\pi_i^b$. Thus the pivotal method is a valid splitting method since it satisfies all of the properties of a splitting method. Then by Corollary 2.2.1 (page 25), the pivotal method preserves the original inclusion probabilities in expectation at each step in the process. □

We can think of the updated inclusion probabilities which are the values of the resulting probability vectors as the outcome of a competition between the two randomly chosen individuals in the population. Since those individuals which were not chosen are not competing, their inclusion probabilities do not change. For the chosen individuals, j and k, the winning competitor gains all of the combined probability of j and k up to a value of 1. Any leftover probability goes to the loser. In the case where $\pi_j + \pi_k > 1$ there is some leftover probability, $\pi_j + \pi_k - 1$, and in the case where $\pi_j + \pi_k \leq 1$ there is no leftover probability, so the losing competitor has updated inclusion probability 0 and will not be selected. With this competition analogy in mind, an alternative (and equivalent) representation of the pivotal method is given as follows (Grafström, 2017).

At any step, t, in the process let $\pi_W = \min(1, \pi_j + \pi_k)$ and $\pi_L = \pi_j + \pi_k - \pi_W$. Then for the two selected individuals, j and k, we update the inclusion probabilities according to

$$(\dots, \pi_j, \dots, \pi_k, \dots) = \begin{cases} (\dots, \pi_W, \dots, \pi_L, \dots) & \text{with probability } \frac{\pi_W - \pi_k}{\pi_W - \pi_L}, \\ (\dots, \pi_L, \dots, \pi_W, \dots) & \text{with probability } \frac{\pi_W - \pi_j}{\pi_W - \pi_L}. \end{cases}$$

The above condensed form is the same as the earlier form (page 27) which

needed two cases (one for when $\pi_j + \pi_k \geq 1$ and one for when $\pi_j + \pi_k < 1$). These two are equivalent forms of the pivotal method.

With the notation for the pivotal method in place, we now return to the inclusion probability vector from Example 4 (page 17) to demonstrate how the pivotal method selects a sample.

Example 7. Suppose that the original inclusion probability vector is $\boldsymbol{\pi} = (0.4, 0.2, 0.8, 0.6)$. Then the pivotal method starts with $\boldsymbol{\pi_0} = (0.4, 0.2, 0.8, 0.6)$. When we randomly choose two individuals from the population, suppose that we choose individuals 1 and 3. Then using the equation on page 27, we have that $\pi_1 + \pi_3 = 1.2 \geq 1$, so we use the first set of equations. This means that we calculate $\alpha_1 = \frac{1-\pi_3}{2-\pi_1-\pi_3} = \frac{1-0.8}{2-0.4-0.8} = 0.25$, $\boldsymbol{\pi}_1^a = (1, 0.2, 0.2, 0.6)$, and $\boldsymbol{\pi}_1^b = (0.2, 0.2, 1, 0.6)$. This first split using the pivotal method is shown in Figure 2.6.

Suppose that of the two branches in this first split, we choose the top branch. This occurs with probability 0.25. Then working with that inclusion probability vector, we repeat the pivotal method process again. Suppose individuals 2 and 4 are chosen. Note that individual 1 is no longer considered since the inclusion probability of that individual is now 1. Then $\pi_{2,1}^a + \pi_{4,1}^a = 0.2 + 0.6 = 0.8 < 1$, so we use the second set of equations on

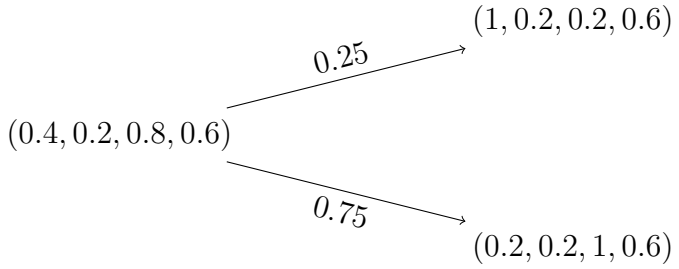

Figure 2.6: First split for pivotal method example

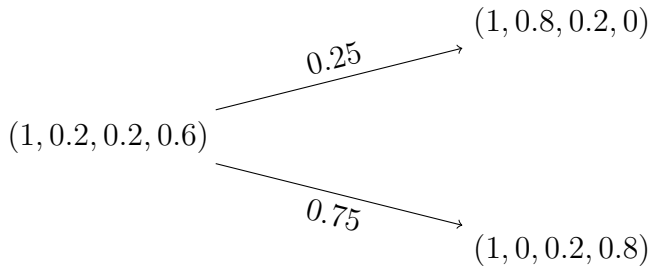

Figure 2.7: Second split for pivotal method example starting with top vector of first split

page 27. We calculate $\alpha_2 = \frac{\pi^a_{2,1}}{\pi^a_{2,1}+\pi^a_{4,1}} = \frac{0.2}{0.2+0.6} = 0.25$, $\boldsymbol{\pi}^a_2 = (1, 0.8, 0.2, 0)$, and $\boldsymbol{\pi}^b_2 = (1, 0, 0.2, 0.8)$. This second split is shown in Figure 2.7.

Suppose now that the bottom branch of this second split is chosen. We then repeat the pivotal method algorithm one last time to arrive at our sample. Since there are only two individuals left to choose from, we select individuals 3 and 4. Then since $\pi^b_{3,2} + \pi^b_{4,2} = 0.2 + 0.8 \geq 1$, we use the first set of equations from page 27. We calculate $\alpha_3 = \frac{1-0.8}{2-0.2-0.8} = 0.2$, $\boldsymbol{\pi}^a_3 = (1, 0, 1, 0)$, and $\boldsymbol{\pi}^b_3 = (1, 0, 0, 1)$. This third split is shown in Figure 2.8.

Suppose that of the two branches in this third split, we choose the bottom branch. Then the sample we would select from this simulation of the pivotal method would be individuals 1 and 4 from the population. A diagram of this entire process is shown in Figure 2.9. The choice of branch is shown in bold,

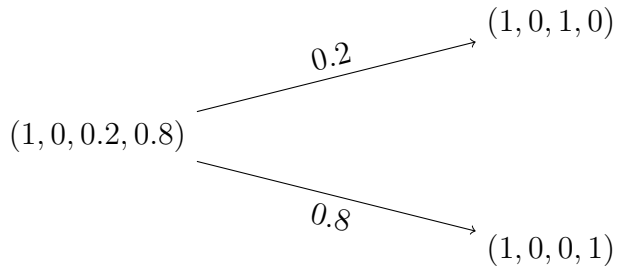

Figure 2.8: Third split for pivotal method example starting with bottom vector of second split

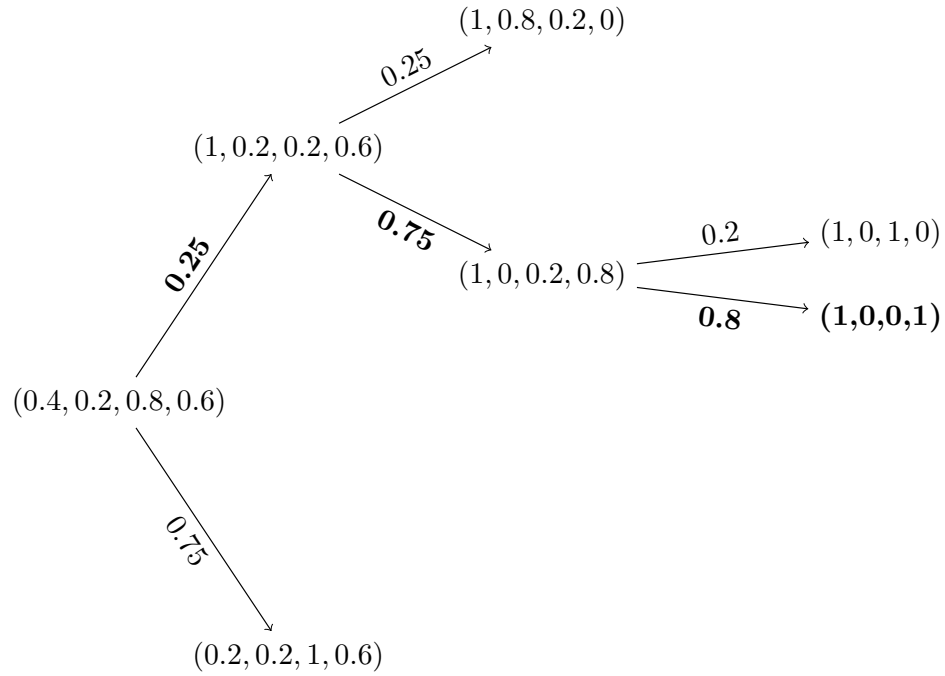

Figure 2.9: Final splitting diagram for one simulation of the pivotal method

and the final vector which selects the sample in this simulation is also shown in bold. If we were to simulate the pivotal method again with the same original inclusion probability vector, the branches we follow would likely be different because the random selection of individuals would be different. Instead of choosing individuals 1 and 3 for the first split, we may choose individuals 1 and 4 or individuals 2 and 3. This would change the diagram completely from the example shown here.

All of the arguments in this section and the previous section assumed that $n = \sum_{i=1}^{N} \pi_i$ is an integer. There is nothing in the description of the pivotal method or splitting methods in general that requires $\sum_{i=1}^{N} \pi_i$ to be an integer. There are considerations which occur if this condition is relaxed.

2.3.1 The Pivotal Method For Noninteger Sample Sizes

Suppose that $\sum_{i=1}^{N} \pi_i = \eta$, where $\eta \in (n, n+1)$ for some integer n. Then the sample size is a random variable where some samples will be selected with size n and others with size $n+1$. There is nothing in the description of splitting methods which requires that the sample size be fixed. The only constraint on the sample size is that at each step of the method $\sum_{i=1}^{N} \pi_i = \sum_{i=1}^{N} \pi_i^a = \sum_{i=1}^{N} \pi_i^b$. Thus any splitting method admits samples sizes which are not fixed (with possible modifications to how the splitting method ends depending on what kind of final sample is sought). For the pivotal method, an extra step is required to ensure that the original inclusion probabilities are preserved in expectation at that last step. See Grafström (2012a) for other approaches to noninteger sample size techniques. For the technique which follows see Grafström (2012a, p. 1479).

First, apply the pivotal method steps until there are n values in the resulting probability vector which are 1 and $N-n-1$ values which are 0.

Lemma 2.3.4. *When $\sum_{i=1}^{N} \pi_i = \eta$ where $\eta \in (n, n+1)$ for some integer n, the pivotal method can be applied until there are n individuals with inclusion probability 1 and $N-n-1$ individuals with inclusion probability 0.*

Proof. We know $\sum_{i=1}^{N} \pi_i > n$, so there is enough probability in the system to arrive at n individuals with inclusion probability 1, then for the other individuals, the only probability left in the system is $\eta - n \in (0, 1)$. The pivotal method will continue to select pairs of individuals with nonzero inclusion probabilities and shift the probability to the winner of the pair until there are no more pairs from which to choose. There are N total possible individuals, n of which will have inclusion probability 1, and only one of which will have nonzero inclusion

probability $\eta - n$, so there are $N - n - 1$ individuals with inclusion probability 0. □

We now split the next to last inclusion probability vector as in Figure 2.10, where $I_i \in \{0, 1\}$, and k is the step at which this splitting occurs. Then $\sum_{i=1}^{N} \pi_{i,k}^a = n + 1$ and $\sum_{i=1}^{N} \pi_{i,k}^b = n$.

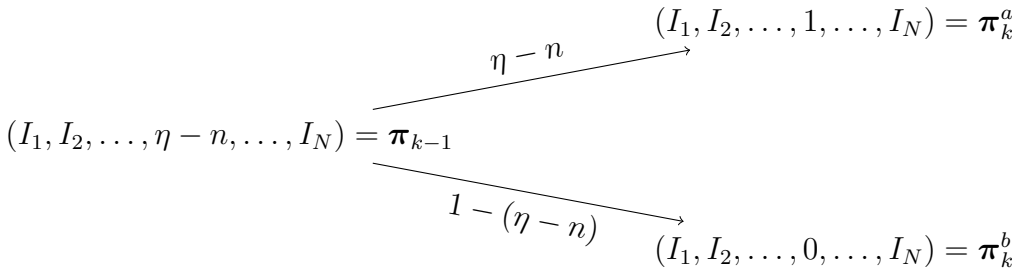

Figure 2.10: Final split into two vectors for last step with noninteger sample size

We know from Theorem 2.3.1 (page 30) that the pivotal method preserves inclusion probabilities in expectation for integer sample sizes, so we only need to prove that the last step shown in Figure 2.10 preserves the inclusion probabilities in the noninteger case.

Lemma 2.3.5. *For the last split (say split k) in the pivotal method with noninteger sample size, $E[\boldsymbol{\pi}_k | \boldsymbol{\pi}_{k-1}, \ldots, \boldsymbol{\pi}_0] = \boldsymbol{\pi}_{k-1}$.*

Proof. We compute the conditional expectation explicitly as

$$\mathrm{E}[\boldsymbol{\pi}_k | \boldsymbol{\pi}_{k-1}, \ldots, \boldsymbol{\pi}_0] = (\eta - n)\boldsymbol{\pi}_k^a + (1 - (\eta - n))\boldsymbol{\pi}_k^b.$$

Then for any individuals such that $I_i \in \{0, 1\}$ the expression above reduces to

$$(\eta - n)I_i + (1 - (\eta - n))I_i = I_i.$$

For the individual with inclusion probability $\eta - n$ the expression becomes

$$(\eta - n)(1) + (1 - (\eta - n))(0) = \eta - n.$$

Thus $(\eta - n)\boldsymbol{\pi}_k^a + (1 - (\eta - n))\boldsymbol{\pi}_k^b = \boldsymbol{\pi}_{k-1}$. □

It then follows from Corollary 2.2.1 (page 25) that the pivotal method with noninteger sample size preserves the original inclusion probabilities in expectation at each step in the process. Note that this technique violates the condition $\sum_{i=1}^N \pi_i = \sum_{i=1}^N \pi_i^a = \sum_{i=1}^N \pi_i^b$ because in the last step of the algorithm we have $\sum_{i=1}^N \pi_i = \eta \neq \sum_{i=1}^N \pi_i^a = n + 1 \neq \sum_{i=1}^N \pi_i^b = n$. This violation only occurs in the last step of the algorithm, and the original inclusion probabilities are still preserved in expectation.

For implementation of the pivotal method, the R package "BalancedSampling" is used (Grafström, 2018). The function *rpm* selects a pivotal method sample (the r in this case differentiates between random pivotal method, which is the method described in this section, and the local pivotal method, which is described in the next section). When the sum of the inclusion probabilities is not an integer, *rpm* uses the method given in this section to select the sample.

2.4 The Local Pivotal Method

2.4.1 Sampling Principles

The pivotal method presented above allows us to select a sample which respects the original inclusion probabilities. If the inclusion probabilities are at least approximately proportional to the response variable, then the resulting

estimator should have smaller variance than when using a simple random sample. We are very often able to collect more information about the population than can just be represented by inclusion probabilities, and individual inclusion probabilities do not generally ensure spatial balance in a sample over a landscape.

For example, suppose that we are interested in estimating the average annual income per household in a given city. One (of many possible) variables which could be used to construct inclusion probabilities is the physical size of the house at which each household resides. Let the inclusion probability be proportional to house size. The size of the house is positively related to the annual income of that household, so these inclusion probabilities should lead to a better estimate of the annual income of the city than a simple random sample would.

But suppose further that the geography of the city plays a role in the size of houses which can be built (in Missoula, MT, certain canyons which are desirable places for building expensive houses do not allow for large houses to be built in them), then the location of the house within the city should also play a role in the way the sample is selected. The size of each house as well as the location of each house can be easily collected on all of the houses within the city. We want to choose bigger houses because they will contribute more to the variability of the estimate than smaller houses. We also want to pick houses from a variety of locations to better match the distribution of houses in the population than a simple random sample would.

A natural question to ask is how can we incorporate the information from additional auxiliary variables in our sampling strategy. Tillé claims that "[t]hree principles can guide the choice of sample: the principle of random-

ization, the principle of overrepresentation, and the principle of restriction" (Tillé, 2017, p.179).

The principle of randomization states that any sampling strategy should employ as much randomization as possible in the selection of a sample. In the context of the pivotal method, randomization is employed in the selection of the two competing points at each stage of the process, and randomization plays a role in the selection of one of the resulting probability vectors, since we select one or the other with probability α_t and $1 - \alpha_t$, respectively.

The principle of overrepresentation states that we should select more often those members in the population who contribute more to the variance of the estimator. For example, suppose we know that the relationship between an explanatory variable x and a response variable y is $y = ax + b + \epsilon$ for constants a and b and error ϵ, and we are interested in estimating the slope, a. Then an optimal sampling strategy which employs overrepresentation would select those x which have the largest and smallest values, since the corresponding y contribute most to the variability in the estimate of a.

An unequal inclusion probability design employs overrepresentation in the use of the inclusion probabilities, π_i. By specifying the inclusion probabilities, we are giving preference to some members of the population by raising the likelihood of their being in the sample.

The principle of restriction states that those samples which produce poor estimates of the parameter of interest should be avoided. The pivotal method does not employ the principle of restriction because there is no mechanism besides the inclusion probabilities which dictates how a sample is chosen. In the splitting algorithm, the inclusion probabilities specify the α_t at each step depending on if $\pi_j + \pi_k \geq 1$ or $\pi_j + \pi_k < 1$, and the choice of which members of

the population compete is random up to those individuals who have already been included or excluded from the sample (since individuals j and k are chosen at random from those individuals whose inclusion probabilities are not 0 or 1).

One way to apply the principle of restriction is to choose samples which satisfy a "balancing" criterion. A sample is balanced if

$$\sum_{i=1}^{n} \frac{\boldsymbol{x}_i}{\pi_i} = \sum_{i=1}^{N} \boldsymbol{x}_i \tag{2.1}$$

where $\boldsymbol{x}_i$ is a vector of auxiliary variables for individual i in the population. The $\boldsymbol{x}_i$ on the left side of the equal sign in equation 2.1 use the index i for the sample individuals, and the $\boldsymbol{x}_i$ on the right side use the index i for the population. Individual i on the left side of the equation is not necessarily the same as individual i on the right side.

As an example of how equation 2.1 works, suppose we wanted to balance on geographic location, where the geographic location is given as (x, y)–coordinates, then to satisfy equation 2.1 we would want the sum of all x–coordinates for the population to be equal to the Horvitz–Thompson estimate of the x–coordinate total for the sample; simultaneously, we want the sum of all y–coordinates for the population to be equal to the Horvitz–Thompson estimate of the y–coordinate total for the sample.

This balancing criterion of equation 2.1 is nearly the same as the characteristic of purposive selection presented by Neyman in 1934 (see page 11, equation 1.2). The balancing criterion specifies that the Horvitz–Thompson estimate of the total of the *auxiliary variables* taken from the sample should be equal to, or nearly equal to, the population total of the auxiliary variables.

From the set of all possible samples which satisfy $\boldsymbol{\pi}$, we restrict to samples which also satisfy the balancing equation, 2.1.

Note that the auxiliary variables need not be only the location of individual i. To return to the example of estimating average annual income per household in a given city, we could "balance" based on the number of bathrooms or the year that the home was built. No assumption is made that the auxiliary space is only the location of the data points. The balance criterion in equation 2.1 only needs the values of the auxiliary variables to be numeric. No distance metric is needed to compare individuals.

Another way to apply the principle of restriction is to select samples which are "well spread." "Roughly speaking, a sample is well spread if the number of selected units is close to what is expected on average, in every part of the auxiliary space" (Grafström, 2013, p.36). A balanced sample is not necessarily well spread, but, as the following theorem states, all well spread samples are approximately balanced.

Theorem 2.4.1. *(Grafström, 2013) If a sample is well spread, then that sample is approximately balanced, i.e. the sample at least approximately satisfies equation 2.1 (page 40).*

Outline of Proof. The argument is that when we measure the spread of a sample using Voronoi polygons (Voronoï, 1908), then a sample which is well spread with respect to the Voronoi polygons is also approximately balanced with respect to equation 2.1 (see Grafström (2013) for more details of the proof than are provided below).

For a selected sample, a Voronoi polygon is formed around a particular sample point by including all population points which are closer to that par-

ticular sample point than to any other sample point. Different samples create different Voronoi polygons. To measure "closer" we use the following distance (Grafström, 2013, p.39): let $\boldsymbol{x}_i \in R^q$ be all available auxiliary variables for individual i, where $\{1, \ldots, p\}$ correspond to the quantitative variables and $\{p+1, \ldots, q\}$ to the qualitative variables. To measure the distance between units i and j in this q–dimensional space use

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{k=1}^{p}(x'_{ik} - x'_{jk})^2} + \sum_{k=p+1}^{q} 1_{x_{ik} \neq x_{jk}} \tag{2.2}$$

where x'_k is the standardized version of x_k. Also, $1_{x_{ik} \neq x_{jk}}$ is an indicator variable which takes the value of 1 when the values of x_{ik} and x_{jk} are not equal and takes the value of 0 otherwise. This is the distance measure we will use throughout this work.

Let P_i be the Voronoi polygon for sampled point i, so P_i is a list of the indices of individuals who are closer to point i than to any other point in the sample. Let $v_i = \sum_{j \in P_i} \pi_j$, the sum of the inclusion probabilities of all points within Voronoi polygon i. A well spread sample is defined to be a sample such that each v_i is equal or close to 1 for $i = 1, \ldots, n$. Essentially, well spread samples create Voronoi polygons from which we would expect, on average, to select one individual.

We can measure the degree of spread of a sample by considering how much the v_i vary from 1. This is computed by

$$B = \frac{1}{n}\sum_{i=1}^{n}(v_i - 1)^2. \tag{2.3}$$

A well spread sample will have a small B value: if a sample is well spread, then

the v_i will be equal or close to 1; B can be thought of as the mean squared error of the v_i from 1; thus a well spread sample should have a small B value. Grafström (2013) proves that a sample which is well spread with respect to the Voronoi polygons (having small B) is also approximately balanced with respect to equation 2.1 (page 40). See Stevens, 2004 for more details on the use of Voronoi polygons in spatial sampling. □

The terminology "spatially balanced" will be used in this work to refer to situations where the only auxiliary variables used are the spatial location of the data points. In this case the distance measure is Euclidean distance. Implicit in the use of Voronoi polygons with spatial auxiliary information is that the data are isotropic. This means that the spatial correlation pattern of the variable of interest, y, does not depend on the direction between points but only the distance between points. If any anisotropies exist, they need to be corrected before Voronoi polygons are constructed. Thus Theorem 2.4.1 (page 41) is still valid when anisotropies exist as long as those anisotropies are corrected. For all of the examples in this work where spatial auxiliary information is used the variables of interest are all isotropic.

2.4.2 The Local Pivotal Method

In order to incorporate the principle of restriction into the pivotal method so that we can achieve balance, we change the way in which the two competing individuals are selected. Since we know that well spread samples will be approximately balanced, we should select the two individuals so that the auxiliary information of the two individuals is well spread. One way to accomplish this is the local pivotal method (Grafström, 2012):

Step 1: Select an individual at random from the population whose inclusion probability is not 0 or 1; call the index of this individual j.

Step 2: Find a nearest neighbor to individual j using $d(\boldsymbol{x}_j, \boldsymbol{x}_k)$ as given in equation 2.2 on page 42. If more than one nearest neighbor exists, randomly select one of them, call the index of this individual k.

Step 3: Use the pivotal method on individuals j and k to update the inclusion probabilities.

Step 4: If all individuals in the population have inclusion probability 0 or 1, stop. Otherwise return to Step 1.

This algorithm assumes that $\sum_{i=1}^{N} \pi_i = n$ for some integer n. We extend the algorithm for the case where $\sum_{i=1}^{N} \pi_i = \eta \in (n, n+1)$ by adding a provision to Step 4 to read "If all individuals except one in the population have inclusion probability 0 or 1, proceed to Step 5. Otherwise return to Step 1" and adding "Step 5: For the last index with nonzero inclusion probability include that individual in the sample with probability $\eta - n$." Note that inclusion probability is the same as selection probability when the sample size is 1 like it is in this 5th step.

This sample selection technique is called the Local Pivotal Method, since individuals compete locally. Individuals which are closer together in the auxiliary space are more likely to compete in Step 3 of the algorithm above. The local pivotal method spreads the sample in auxiliary space by only selecting a few individuals from those which are clustered together (the number of individuals selected depends on the total inclusion probability of the cluster). To see how the local pivotal method proceeds to select a sample, we consider the

Individual	Inclusion Probability	x coordinate	y coordinate
1	0.2	9.5	4.1
2	0.5	3.1	4.2
3	0.8	3.2	9.5
4	0.4	1.2	3.2
5	0.1	8.1	4.5
6	0.3	6.8	1.7
7	0.7	1.0	2.4

Table 2.1: Inclusion probability and location for the population of 7 individuals of Example 8

following example.

Example 8. This example was generated in the open source statistical software R (see R Core Team (2019), and see appendix of this work, page 117, for commented code). Suppose that the population consists of 7 individuals. For each of these 7 individuals we have an inclusion probability as well as the location (as an (x, y)–coordinate). All of this auxiliary information is contained in Table 2.1.

Note that $\sum_{i=1}^{7} \pi_i = 3$, so we are interested in selecting a sample of size 3. We would also like to balance on the information given in the (x, y)–coordinates, so we will use the local pivotal method to select a sample.

In Figure 2.11 (page 46), the individuals are plotted at their (x, y)–coordinates with the inclusion probability of each individual shown next to the individual. The size and color of each point represents the inclusion probability, with smaller, more blue circles indicating inclusion probabilities of less than 0.5, and with larger, more red circles indicating inclusion probabilities of more

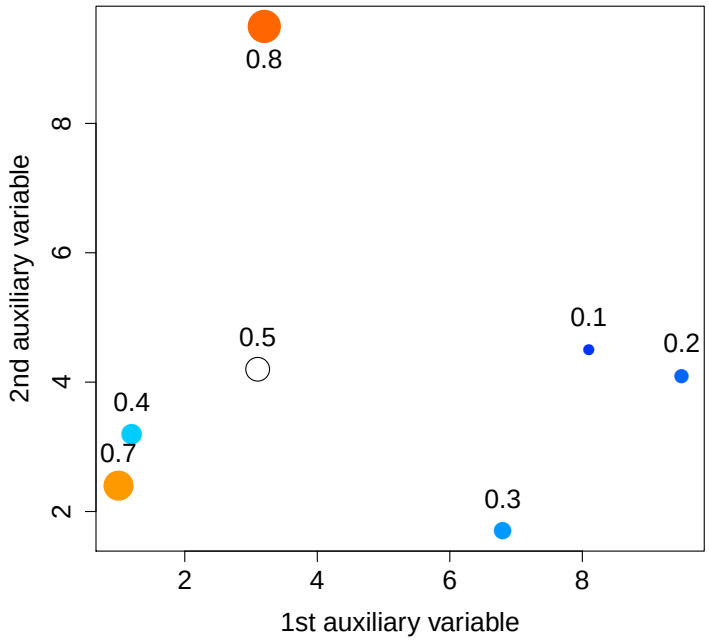

Figure 2.11: Plot of location and inclusion probabilities for the population of 7 individuals of Example 8

than 0.5. An individual with inclusion probability exactly 0.5 is indicated by a circle with white on the inside.

Following the steps in the algorithm given above for the local pivotal method (page 44), suppose that the first randomly selected individual is 3. Then the nearest neighbor to individual 2 is individual 3 (the distance to measure "nearest" here is the usual Euclidean distance in two dimensions). These two individuals are highlighted in Figure 2.12 (page 47). The randomly chosen individual is shown with a $\times$ through it, and the nearest neighbor is shown with a $+$ through it. We now use the pivotal method on these two individuals to update the inclusion probabilities. Recall that the pivotal method updates inclusion probabilities in a competition–like manner (see page 31 for details). Now that we have the two competitors chosen, we have the following values: $\pi_W = \min(1, \pi_3 + \pi_2) = \min(1, 0.8 + 0.5) = \min(1, 1.3) = 1$ and

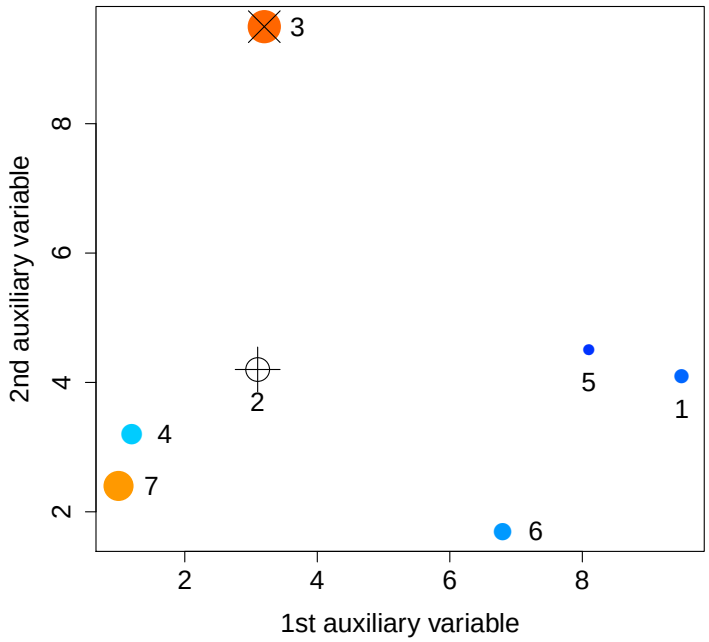

Figure 2.12: Plot of first randomly chosen individual (individual 3 marked with $\times$) and nearest neighbor (individual 2 marked with $+$) in Example 8

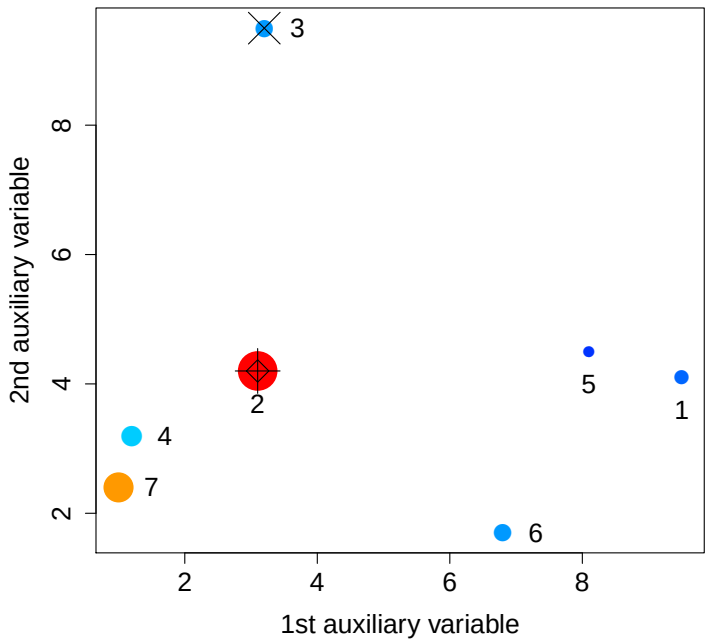

Figure 2.13: Plot of the first updated inclusion probabilities for individual 3 marked with $\times$ and nearest neighbor, individual 2 marked with $+$, in Example 8

$\pi_L = \pi_3 + \pi_2 - \pi_W = 0.8 + 0.5 - 1 = 0.3$. We then update the inclusion probabilities according to

$$(\dots, \pi_2, \pi_3, \dots) = \begin{cases} (\dots, \pi_W, \pi_L, \dots) & \text{with probability } \frac{\pi_W - \pi_3}{\pi_W - \pi_L} = \frac{2}{7}, \\ (\dots, \pi_L, \pi_W, \dots) & \text{with probability } \frac{\pi_W - \pi_2}{\pi_W - \pi_L} = \frac{5}{7}. \end{cases}$$

In this particular simulation, individual 2 wins the competition, and the resulting updated inclusion probabilities are shown in Figure 2.13 (page 47). Note that individual 2 now has inclusion probability 1, so the size and color of the circle have changed from Figure 2.12 (page 47). There is also a diamond inside of the circle for individual 2 indicating that individual 2 has inclusion probability 1. In the updated inclusion probabilities, individual 3 now has inclusion probability 0.3.

The local pivotal method algorithm now begins again by selecting a random individual (not individual 2 because it now has inclusion probability 1). In this simulation, the randomly chosen individual is individual 3, and the nearest neighbor to individual 3 which has non–one inclusion probability is individual 4. This is shown in Figure 2.14 (page 49).

The competition between individuals 3 and 4 has the following values: $\pi_W = \min(1, \pi_3 + \pi_4) = \min(1, 0.3 + 0.4) = \min(1, 0.7) = 0.7$ and $\pi_L = \pi_3 + \pi_4 - \pi_W = 0.3 + 0.4 - 0.7 = 0$. We then update the inclusion probabilities according to

$$(\dots, \pi_3, \pi_4, \dots) = \begin{cases} (\dots, \pi_W, \pi_L, \dots) & \text{with probability } \frac{\pi_W - \pi_4}{\pi_W - \pi_L} = \frac{3}{7}, \\ (\dots, \pi_L, \pi_W, \dots) & \text{with probability } \frac{\pi_W - \pi_3}{\pi_W - \pi_L} = \frac{4}{7}. \end{cases}$$

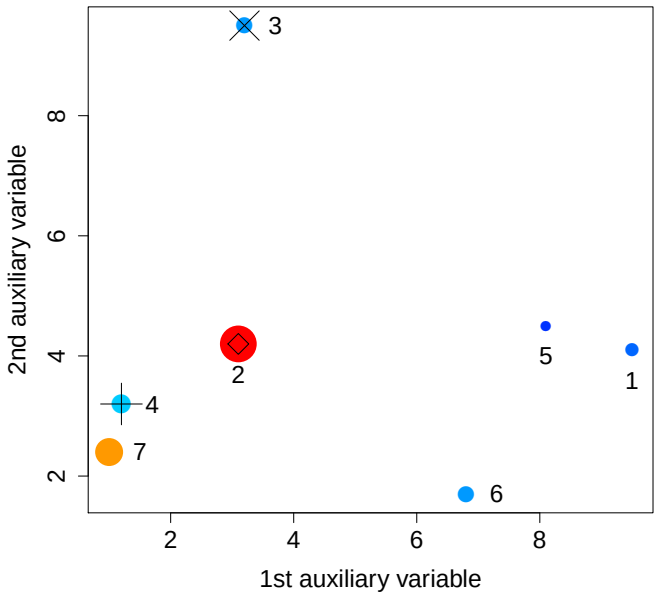

Figure 2.14: Plot of the second randomly chosen individual (individual 3 marked with $\times$) and nearest neighbor (individual 4 marked with $+$) in Example 8

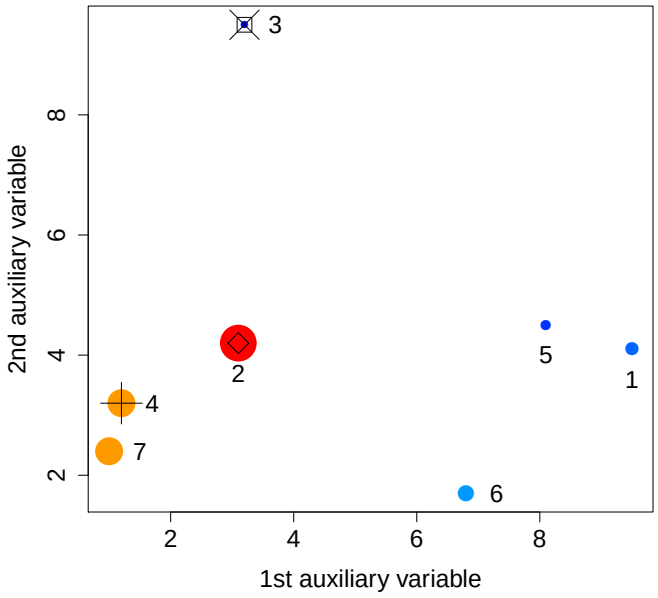

Figure 2.15: Plot of the second updated inclusion probabilities for individual 3 marked with $\times$ and nearest neighbor, individual 4 marked with $+$, in Example 8

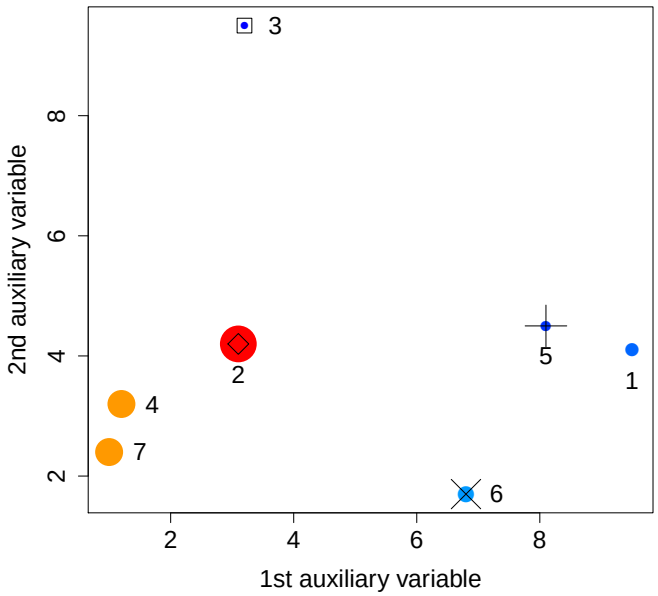

Figure 2.16: Plot of the third randomly chosen individual (individual 6 marked with $\times$) and nearest neighbor (individual 5 marked with $+$) in Example 8

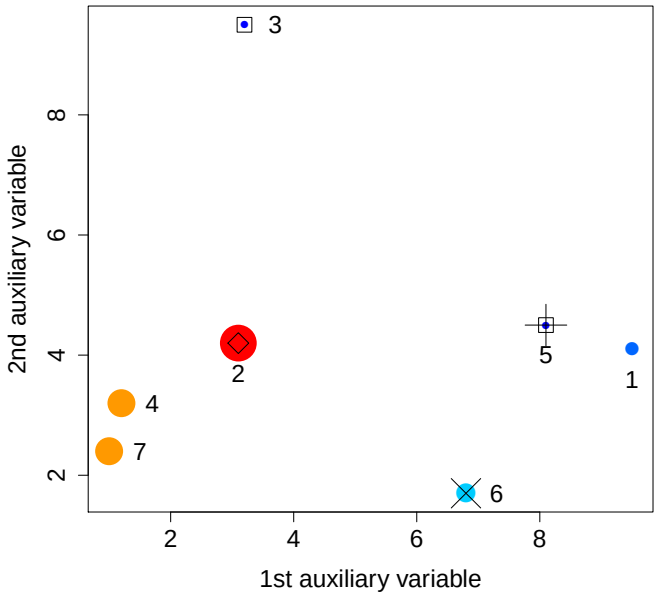

Figure 2.17: Plot of the third updated inclusion probabilities for individual 6 marked with $\times$ and nearest neighbor, individual 5 marked with $+$, in Example 8

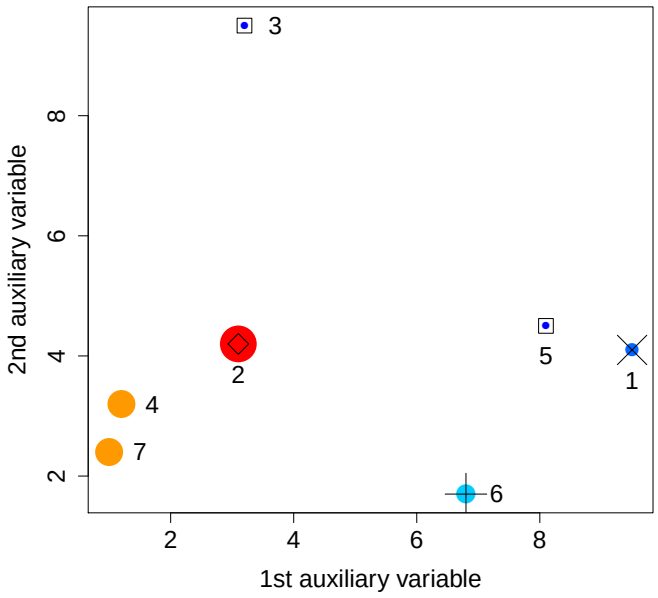

Figure 2.18: Plot of the fourth randomly chosen individual (individual 1 marked with $\times$) and nearest neighbor (individual 6 marked with $+$) in Example 8

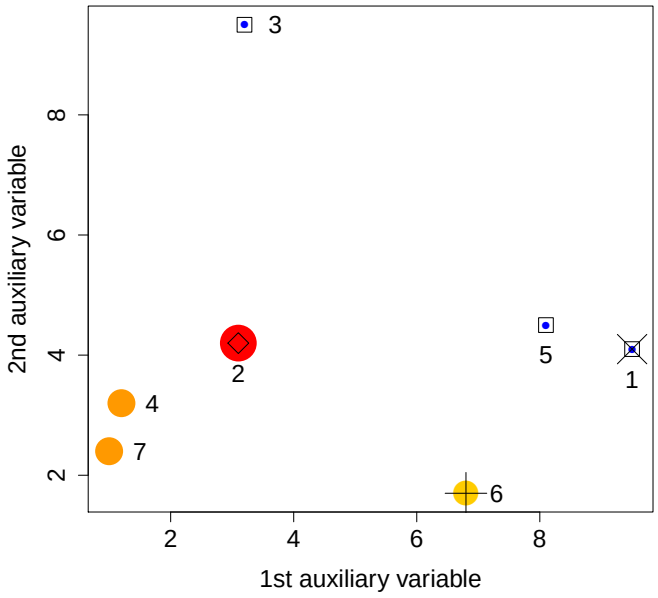

Figure 2.19: Plot of the fourth updated inclusion probabilities for individual 1 marked with $\times$ and nearest neighbor, individual 6 marked with $+$, in Example 8

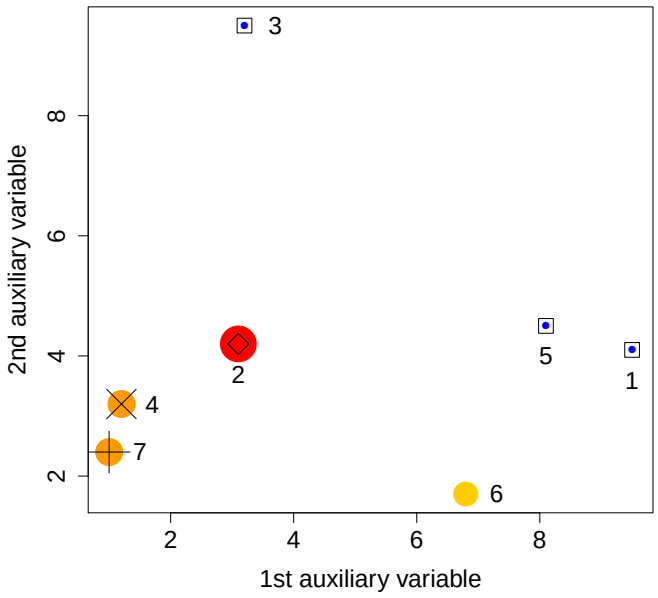

Figure 2.20: Plot of the fifth randomly chosen individual (individual 4 marked with $\times$) and nearest neighbor (individual 7 marked with $+$) in Example 8

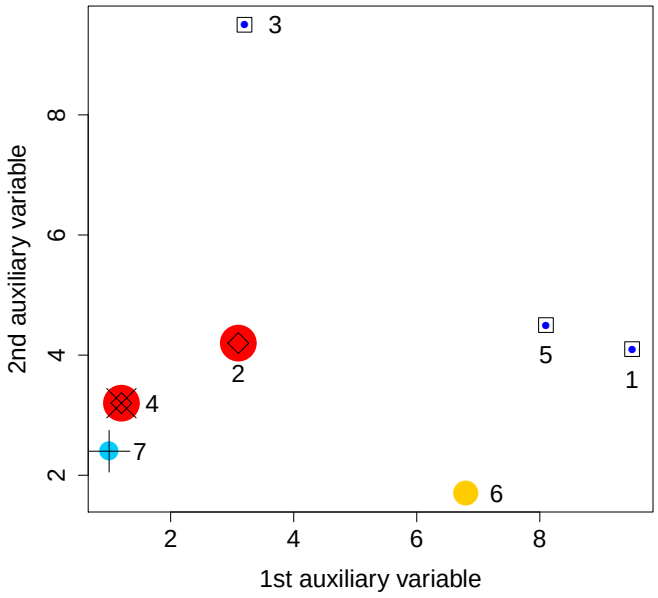

Figure 2.21: Plot of the fifth updated inclusion probabilities for individual 4 marked with $\times$ and nearest neighbor, individual 7 marked with $+$, in Example 8

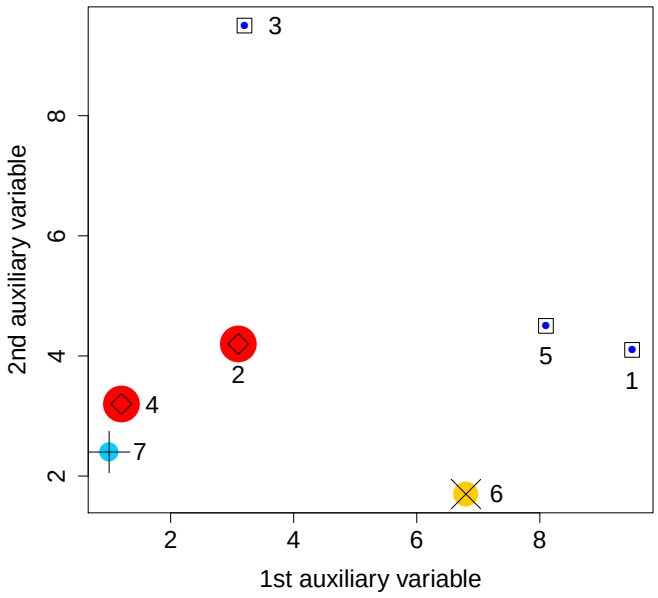

Figure 2.22: Plot of the sixth randomly chosen individual (individual 6 marked with $\times$) and nearest neighbor (individual 7 marked with $+$) in Example 8

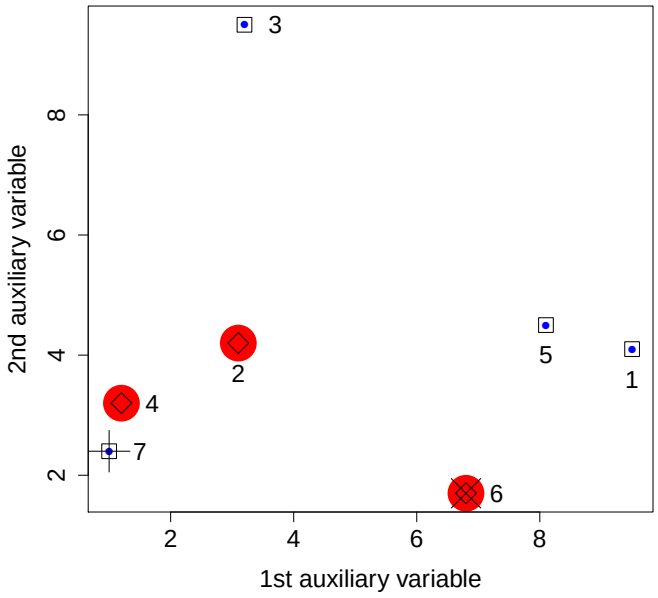

Figure 2.23: Plot of the sixth updated inclusion probabilities for individual 6 marked with $\times$ and nearest neighbor, individual 7 marked with $+$, in Example 8

Notice that we used the updated inclusion probability of individual 3 for these computations. In this particular simulation, individual 4 wins the competition, and the resulting updated inclusion probabilities are shown in Figure 2.15 (page 49). There is now one individual in the population with 0 inclusion probability (individual 3), and there is one individual in the population with inclusion probability 1, individual 2. Individual 4 now has inclusion probability 0.7. The local pivotal method algorithm begins again and continues until there are 3 individuals from the population chosen. Figure 2.16 (page 50) through Figure 2.23 (page 53) follow this process to the end.

Figure 2.23 (page 53) shows that the individuals which are included in the sample are individuals 2, 4, and 6. This is the resulting sample from one simulation of the local pivotal method. If we ran the simulation again, we would likely get a different sample.

In the example above, the final sample chosen appears to be well spread in the two dimensional auxiliary space. The following theorem confirms this observation.

Theorem 2.4.2. *(Grafström, 2014) The local pivotal method produces samples which are well spread, i.e. which have small* $B = \frac{1}{n}\sum_{i=1}^{n}(v_i - 1)^2$ *where* v_i *is the sum of the inclusion probabilities of points within Voronoi polygon* i*.*

The idea of the proof is to note that the algorithm of the local pivotal method will often move probability to and from individuals which are close in the auxiliary space. When we form Voronoi polygons in the auxiliary space, the probability of all of the individuals in each polygon will be concentrated at one point, and since the probability of the system does not move far in auxiliary space, the sum of the inclusion probabilities of all individuals within

each Voronoi polygon of the sample individuals will be nearly 1. This last attribute of each Voronoi polygon is what we defined as a well spread sample (see Grafström (2014) for details of the proof).

Corollary 2.4.1. *The local pivotal method is a splitting method that produces samples which are approximately balanced, i.e.* $\sum_{i=1}^{n} \frac{\boldsymbol{x}_i}{\pi_i} \approx \sum_{i=1}^{N} \boldsymbol{x}_i$.

Outline of Proof. Since the local pivotal method splits the original probability vector by the same process as the pivotal method, Theorem 2.3.1 (page 30) implies that the local pivotal method is a valid splitting method. By Theorem 2.4.2 (page 54), we know that the the local pivotal method produces samples which are well spread. Finally, by Theorem 2.4.1 (page 41), we know that samples which are well spread are approximately balanced. Thus local pivotal method samples are approximately balanced. □

Example 9. To illustrate the difference between the local pivotal method and the pivotal method, consider a population of size 8 with auxiliary information $\boldsymbol{x} = (1, 4, 3, 1, 7, 5, 3, 1)$ and inclusion probability vector $\boldsymbol{\pi} = (0.11, 0.32, 0.25, 0.13, 0.60, 0.32, 0.20, 0.07)$. We want to select a sample of size 2 from this population. The local pivotal method will select a sample which respects the inclusion probabilities as well as balancing on the $\boldsymbol{x}$ information. Since there is only one auxiliary variable, the balance is in one dimension, so the local pivotal method is trying to balance points selected along the number line between 1 and 7, the range of the x_i. If instead of a local pivotal method sample we choose a pivotal method sample, no balance on $\boldsymbol{x}$ is attempted.

A somewhat poorly balanced sample is individuals 3 and 7 since $x_3 = 3$, $x_7 = 3$, and $\sum_{i=1}^{8} x_i = 25$ while $\sum \frac{x_i}{\pi_i} = \frac{3}{0.25} + \frac{3}{0.20} = 27$. In a simulation of $1,000,000$ pivotal method samples, both individuals 3 and 7 were chosen

$23,824$ times. In $1,000,000$ local pivotal method samples, both individuals 3 and 7 were chosen 0 times.

The most unbalanced sample is individuals 1 and 4, with $\sum \frac{x_i}{\pi_i} = \frac{1}{0.11} + \frac{1}{0.13} = 16.78$. This sample of individuals 1 and 4 gives the worst estimate of the total of the $\boldsymbol{x}$ values out of all possible samples of two individuals. When using the pivotal method, this sample occurred $6,263$ times out of $1,000,000$ simulated samples, and the sample occurred 0 times out of $1,000,000$ when using the local pivotal method.

The most well balanced sample is individuals 1 and 6, with $\sum \frac{x_i}{\pi_i} = \frac{1}{0.11} + \frac{5}{0.32} = 24.72$. This sample occurred $17,034$ times in the simulation using the pivotal method and $29,804$ times using the local pivotal method.

In the above example, we see the change in frequency of samples between selecting a sample according to just the inclusion probability vector and selecting a sample according to the inclusion probability vector while also balancing on some auxiliary variable. In the following example, we consider how an inclusion probability vector and an auxiliary variable can arise in sampling applications.

Example 10. Returning to the forestry example from Chapter 1 (Example 2, page 9, see Grafström (2013a) for more details), recall that a population of 846 fixed radius plots at a research site in Sweden were to be sampled with the goal of estimating the total volume of all trees at the site. In Example 2 we collected auxiliary information on the average vegetation height at each plot, and used that information to construct an inclusion probability vector.

We would like to find a way to incorporate the spatial location of the plots into our sampling design, since the plots are placed on a 40 meter by 40

meter grid. This spatial information is important because forest attributes can exhibit spatial autocorrelation. This is the reason that the plots were placed according to a grid in the first place; forest attributes tend to be similar at plots which are close together.

With the pivotal method, we can select a sample which respects the inclusion probability vector that we computed from the average vegetation height, but the pivotal method does not incorporate the spatial information. With the local pivotal method, we can select a sample which respects the inclusion probabilities and also balances on the location of the plots. We focus on the local pivotal method in later analysis because of its ability to incorporate more auxiliary information than the pivotal method.

Next we return to the topic of noninteger sample sizes that we first encountered in section 2.3.1 (page 35).

2.4.3 Further Noninteger Sample Size Properties

Horvitz (1952) showed that the Horvitz–Thompson estimator is an unbiased estimator of the population total, τ, for fixed sample sizes. When the sample size varies as in the case of the pivotal and local pivotal methods for noninteger inclusion probability sums, it is not yet proven that the Horvitz–Thompson estimator is unbiased.

If $\sum_{i=1}^{N} \pi_i = \eta \in (n, n+1)$ for some integer n, then the pivotal and local pivotal methods select a sample of size n with probability $\eta - n$ and a sample of size $n+1$ with probability $1-(\eta-n)$. This is due to the last step in either method choosing to include or exclude the last individual with probability $\eta - n$. Note that the following theorem only applies to samples which have

been chosen using the pivotal or local pivotal method.

Theorem 2.4.3. *For variable sample sizes arising in the pivotal and local pivotal methods, the Horvitz–Thompson estimator,* $\widehat{\tau}_\pi = \sum_{i=1}^{n} \frac{y_i}{\pi_i}$*, is an unbiased estimator of the population total,* τ*.*

Proof. (This proof structure is similar to that presented by Cordy (1993)). Let $\sum_{i=1}^{N} \pi_i = \eta \in (n, n+1)$ for some integer, n. Then let z_i be an indicator variable which is 1 if individual i is in the sample and 0 if individual i is not in the sample. Then the Horvitz–Thompson estimator can be rewritten as

$$\widehat{\tau}_\pi = \sum_{i=1}^{n} \frac{y_i}{\pi_i} = \sum_{i=1}^{N} \frac{z_i y_i}{\pi_i}$$

where y_i is a constant for all i. Then due to the way in which the final split of the pivotal method is calculated, we have $\pi_i = (\eta - n)\pi_{n+1,i} + (1 - (\eta - n))\pi_{n,i}$, where $\pi_{n+1,i}$ is the inclusion probability of individual i for the split in which a sample of size $n+1$ is chosen, and $\pi_{n,i}$ is the inclusion probability of individual i for the split in which a sample of size n is chosen. Then $\mathrm{E}[z_i|n] = \pi_{n,i}$ and $\mathrm{E}[z_i|n+1] = \pi_{n+1,i}$ for each i.

We use the conditional expectation formula to get $\mathrm{E}[\widehat{\tau}_\pi] = \mathrm{E}[\mathrm{E}[\widehat{\tau}_\pi|M]]$, where $M \sim \text{Bernoulli}(\eta - n)$ and M takes the values of $n+1$ and n (with probability $\eta - n$ and $1 - (\eta - n)$, respectively). Then since there are only two possible outcomes for the sample size and the probability of each outcome is known, we get

$$\mathrm{E}[\mathrm{E}[\widehat{\tau}_\pi|M]] = (\eta - n)\mathrm{E}[\widehat{\tau}_\pi|n+1] + (1 - (\eta - n))\mathrm{E}[\widehat{\tau}_\pi|n] =$$

$$(\eta - n)(\sum_{i=1}^{N} \frac{\mathrm{E}[z_i|n+1]y_i}{\pi_i}) + (1 - (\eta - n))(\sum_{i=1}^{N} \frac{\mathrm{E}[z_i|n]y_i}{\pi_i}) =$$

$$\sum_{i=1}^{N} \frac{y_i}{\pi_i}\big((\eta - n)\mathrm{E}[z_i|n+1] + (1 - (\eta - n))\mathrm{E}[z_i|n]\big) =$$

$$\sum_{i=1}^{N} \frac{y_i}{\pi_i}\big((\eta - n)\pi_{n+1,i} + (1 - (\eta - n))\pi_{n,i}\big) =$$

$$\sum_{i=1}^{N} \frac{y_i}{\pi_i}\pi_i = \sum_{i=1}^{N} y_i = \tau.$$

Thus $\mathrm{E}[\widehat{\tau}_\pi] = \tau$, so $\widehat{\tau}_\pi$ is an unbiased estimator of τ. □

It can be shown in a similar way that the formula for the variance of the Horvitz–Thompson estimator is the same for fixed sample sizes as it is for variable sample sizes arising in the pivotal and local pivotal methods (see for a similar proof in Cordy (1993, p. 360-1)).

Example 11. To illustrate Theorem 2.4.3, return to the population from Example 1 (page 8), a population of size 8 with $\boldsymbol{x} = (1, 4, 3, 1, 6, 5, 2, 1)$. Additionally, suppose the response variable is $\boldsymbol{y} = (3, 20, 11, 2, 38, 24, 3, 3)$. In Example 1, a sample which was $\frac{1}{2}$ the size of the population or 4 was taken. Suppose now that the desired sample is $\frac{1}{3}$ the size of the population. Then using equation 1.1 (page 7) we have

$$\boldsymbol{\pi} = \frac{8}{3} \cdot \frac{\boldsymbol{x}}{23} \approx (0.12, 0.46, 0.35, 0.12, 0.70, 0.58, 0.23, 0.12).$$

The sum of the π_i is $2.68 \approx \frac{8}{3}$. Using the pivotal or local pivotal method we expect a sample of size 3 with probability 0.68 and a sample of size 2 with probability $1 - 0.68 = 0.32$. Suppose the first local pivotal sample selects individuals $2, 5$, and 7. Then the estimate of the population total for that

sample is

$$\widehat{\tau}_1 = \sum_{i=1}^{3} \frac{y_{[i]}}{\pi_{[i]}} = \frac{y_{[1]}}{\pi_{[1]}} + \frac{y_{[2]}}{\pi_{[2]}} + \frac{y_{[3]}}{\pi_{[3]}} = \frac{y_2}{\pi_2} + \frac{y_5}{\pi_5} + \frac{y_7}{\pi_7} =$$

$$\frac{20}{0.46} + \frac{38}{0.70} + \frac{3}{0.23} \approx 110.81,$$

where $y_{[i]}$ and $\pi_{[i]}$ are the values for the ith individual in the sample. The true population total is 104. Suppose the second local pivotal sample is individuals 6 and 8, then

$$\widehat{\tau}_2 = \frac{24}{0.58} + \frac{3}{0.12} \approx 66.38.$$

Taking 998 more samples and computing the estimated total for each sample produces a mean for all $1,000$ estimated totals of 103.84. By Theorem 2.4.3 this mean is expected to be close to the true total of 104.

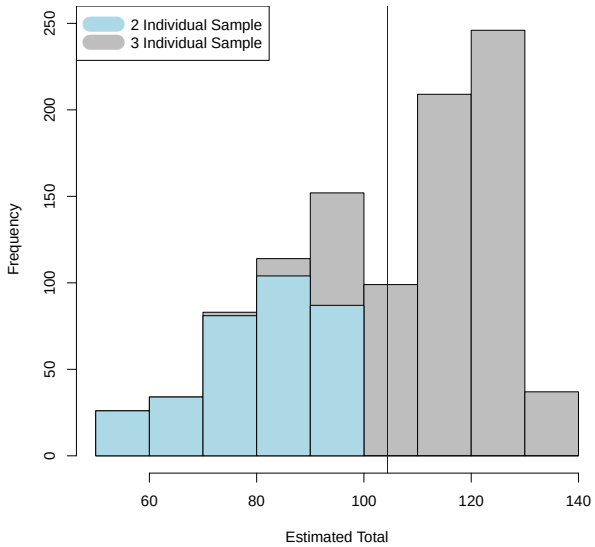

Figure 2.24: Histogram of $1,000$ LPM samples to illustrate Theorem 2.4.3. The blue vertical line indicates the mean of the $1,000$ estimated totals. The true population total is 104.

A histogram of the $1,000$ estimates is shown in Figure 2.24 (page 60) with a blue vertical line representing the population total. The distribution of estimates is somewhat left skewed. This is due to some samples only using two individuals to estimate the total which generally lead to underestimates of the total. Since those samples do not occur often, the frequency of such estimates is lower. There is also a peak at values higher than the true total due to samples with 3 individuals being selected which tend to overestimate the true total. The result of Theorem 2.4.3 (page 58) is that these estimates converge on average to the true total.

This chapter presented two methods for selecting a sample without replacement for an unequal inclusion probability design: the pivotal method and local pivotal method. Both are splitting methods, and the local pivotal method is a special case of the pivotal method. The local pivotal method attempts to select a sample which is balanced on some auxiliary variables. We now have a way to select the sample, so we can use this sample to estimate the population total. The problem now is to estimate the variability in that estimate of the total. The next chapter explores techniques for estimating the variability in estimates of the total.

Chapter 3

Variance Estimation

In Chapter 2, the pivotal and local pivotal methods were described and their properties developed. We then have two sampling algorithms available for selecting an unequal inclusion probability sample without replacement from a finite population. For any unequal inclusion probability sampling design, an unbiased estimator of the total is given by the Horvitz–Thompson estimator. This chapter focuses on different ways of estimating the variance of the Horvitz–Thompson estimator when the sample is selected using the local pivotal method. The estimators we will consider are: the simple random sample variance estimator; the local neighborhood variance estimator (Stevens, 2003); the nearest neighbor variance estimator (Wolter, 2007); the jackknife (Quenouille, 1949); the naive bootstrap (Efron, 1979); and bootstrap subsampling (Bickel, 1988). We will then, in Chapter 4, determine which of the estimators of the variance perform best when using the local pivotal method.

For estimating the population total for some response variable, y_i, the

variance of the Horvitz–Thompson estimator (Horvitz, 1952) is

$$\mathrm{Var}(\widehat{\tau}_\pi) = \sum_{i=1}^{N}\left(\frac{1-\pi_i}{\pi_i}\right)y_i^2 + \sum_{i=1}^{N}\sum_{j\neq i}^{N}\left(\frac{\pi_{ij}-\pi_i\pi_j}{\pi_i\pi_j}\right)y_iy_j$$

an unbiased estimator of this variance is

$$\widehat{\mathrm{Var}}(\widehat{\tau}_\pi) = \sum_{i=1}^{n}\left(\frac{1-\pi_i}{\pi_i^2}\right)y_i^2 + \sum_{i=1}^{n}\sum_{j\neq i}^{n}\left(\frac{\pi_{ij}-\pi_i\pi_j}{\pi_i\pi_j}\right)\frac{1}{\pi_{ij}}y_iy_j.$$

This estimator can sometimes take negative values. An alternative is the Sen–Yates–Grundy estimator (Sen, 1953 and Yates, 1953)

$$\widehat{\mathrm{Var}}_{SYG}(\widehat{\tau}_\pi) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\left(\frac{\pi_{ij}-\pi_i\pi_j}{\pi_{ij}}\right)\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2.$$

The Sen–Yates–Grundy estimator is unbiased and always gives non–negative values. A problem with these estimators of the variance is that they each depend on π_{ij}, the joint inclusion probability of elements i and j, which is the probability that elements i and j are both included in a sample. For neither the pivotal nor local pivotal method is there a known way to compute these joint inclusion probabilities, so neither variance estimator above can be used. These estimators are also sensitive to small joint inclusion probabilities, which will likely occur with the local pivotal method. Note that the pivotal and local pivotal methods likely have different variances since the local pivotal method restricts the selection of points which are close in the auxiliary space. This means that the joint inclusion probability for close points will be smaller with the local pivotal method than with the pivotal method. Since the pivotal method and local pivotal method likely have different variances, we will only

consider estimating the variance for the local pivotal method in this work.

To estimate the variance we classify the estimators mentioned at the beginning of this chapter into two categories: rescaling estimators and resampling estimators. We consider the simple random sample variance estimator, the local neighborhood variance estimator (which from this point on we will call the local variance estimator), and the nearest neighbor variance estimator as rescaling estimators. Of the resampling estimators, we examine the jackknife, the naive bootstrap, and bootstrap subsampling.

3.1 Rescaling Techniques

By rescaling here, we have two different ideas in mind. In the first case an estimator of the variance is rescaled by a value to improve the estimate. In the second case, instead of computing the variance using the mean from the entire sample, local means are used. In the usual summation for variance, we use $(y_i - \bar{y})^2$ with the same mean, $\bar{y}$, subtracted from each individual. A local means approach to estimating the variance would use $(y_i - \bar{y}_i)^2$, where $\bar{y}_i$ is the mean of individuals which are close to individual i. Thus each term in the sum will potentially have a different mean.

As an example of the first kind of rescaling, we consider the simple random sample variance estimator for a finite population. D'Orazio (2003) suggests taking the simple random sample variance estimator for a finite population and rescaling that value by a suitably chosen constant. The simple random sample variance estimator of the total for a finite population is

$$\widehat{\mathrm{Var}}_{SRS}(\hat{\tau}) = N\big(\frac{N-n}{n}\big)\frac{1}{n-1}\sum_{j=1}^{n}(y_i - \bar{y})^2.$$

D'Orazio is working with systematic sampling where the initial position is chosen randomly and subsequent points are systematically chosen from that initial position. One variance estimator D'Orazio mentions for linear systematic sampling is the simple random sampling estimator corrected for serial correlation. The estimator is

$$\widehat{\text{Var}}_w(\hat{\tau}) = w \cdot \widehat{\text{Var}}_{SRS}(\hat{\tau})$$

where $w = 1 + \frac{2}{\ln(r_k)} + \frac{2}{\frac{1}{r_k}-1}$, if $r_k > 0$ and w=1, if $r_k \leq 0$, and r_k is the serial correlation among sample units at lag k. This estimator can only incorporate one dimensional auxiliary information into the estimation of the variance. Because of this limitation, we will not use this estimator, but we mention it to demonstrate that rescaling an estimator by a constant has precedence in the variance estimation literature.

For the second kind of rescaling we consider the local variance estimator. Stevens and Olsen developed the local variance estimator for estimates from a generalized random tesselation stratified (GRTS) sampling design (see Stevens, 2004 for details on GRTS and Stevens, 2003 for the local variance estimator). The GRTS sampling design was developed to select a systematic sample which respects a prescribed set of inclusion probabilities and allows for the selection of additional individuals due to nonresponse. The sample is systematic only in two dimensions, and the GRTS sampling design can only be applied to two dimensional auxiliary information.

For example, suppose we are interested in estimating the water quality of rivers and streams in some region. In addition to knowing the geographic location of the rivers and streams we have the relative size of each river and

stream. We can use the GRTS sampling design to select a sample which is systematic with respect to the two dimensional geographic location and which respects inclusion probabilities specified by the relative size. We can also select more locations in the event that some sampled locations are inaccessible without violating the prescribed inclusion probabilities.

The local variance estimator estimates the variance of the total when samples are selected using the GRTS sampling design. These samples are well spread, so using Theorem 2.4.1 (page 41), these samples are also approximately balanced. Since the local pivotal method selects samples which are approximately balanced and which respect a prescribed set of inclusion probabilities, the local variance estimator may provide useful estimates of the variance of the total when samples are selected using the local pivotal method.

The local variance estimator is

$$\widehat{\mathrm{Var}}_{NBH}(\widehat{\tau}_\pi) = N^2 \sum_{i=1}^{n} \sum_{j \in D(i)} w_{ij} \left(\frac{y_j}{\pi_j} - \sum_{k \in D(i)} w_{ik} \frac{y_k}{\pi_k} \right)^2$$

which is also sometimes written as

$$\widehat{\mathrm{Var}}_{NBH}(\widehat{\tau}_\pi) = N^2 \sum_{i=1}^{n} \sum_{j \in D(i)} w_{ij} \left(\frac{y_j}{\pi_j} - \bar{y}_{D(i)} \right)^2$$

where $D(i)$ is a neighborhood of individual i which contains individual i and at least the three nearest neighbors of i, and w_{ij} are weights which are a function of π_j and the distance between individuals i and j. The weights are subject to the following two constraints:

1. The weight should decrease as π_j increases and decrease as the distance between individuals i and j increases,

2. $\sum_{i=1}^{n} w_{ij} = \sum_{j=1}^{n} w_{ij} = 1$.

Stevens (2003), p. 601, developed a procedure for calculating weights satisfying these two criteria which is implemented in the R package *spsurvey*. See the R package *spsurvey* and its documentation for more details of implementation (Kincaid, 2018). The package *spsurvey* is programmed to accept only two dimensional auxiliary information for use with the local variance estimator. Theoretically, the local variance estimator can be calculated on any dimension of auxiliary information. To accomplish this, we would need to rewrite major portions of the code from the *spsurvey* package. We will restrict our use of the local variance estimator to populations with two dimensional auxiliary information.

The last of the rescaling estimators we consider will be called the nearest neighbor estimator. Grafström and Schelin (2014) suggest this estimator for the variance of the total for samples selected using the local pivotal method, and it relies on fewer neighbors to compute the variance than Stevens and Olsen's estimator. This estimator was called v_{12} when introduced by Wolter (2007, p. 336); Grafström and Schelin do not name this estimator; we will call it the nearest neighbor estimator. The estimator is

$$\widehat{\mathrm{Var}}_{NN}(\widehat{\tau}_{\pi}) = \frac{1}{2}\sum_{i=1}^{n}\left(\frac{y_i}{\pi_i} - \frac{y_{j_i}}{\pi_{j_i}}\right)^2$$

where i indexes the sample individuals from 1 up to n, and j_i is the index of the nearest neighbor to i in the sample. The nearest neighbor variance estimator is widely used to estimate the variance of the total from local pivotal method samples (see Grafström and Schelin, 2014; Grafström and Matei, 2018a; Räty, et al., 2020).

The simple random sample variance estimator, the local variance estimator, and the nearest neighbor estimator will be compared to resampling estimators which are considered next.

3.2 Resampling Estimators

3.2.1 The Jackknife

The modern jackknife resampling technique (Efron, 1982) recomputes an estimator with a different individual from the sample missing each time. Then the recomputed values are used to estimate the variance of the estimator. For example if we have a sample $(y_1, y_2, \ldots, y_n)$, and we are interested in computing the variability of the median of the sample, we compute M_i as the median of $(y_1, y_2, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$ for $i = 1, 2, \ldots, n$. Then we have n estimates of the median of the sample, so we estimate the variability in our estimate of the median as the variability in the M_i.

A modern jackknife estimator of the variance of the total for use with unequal inclusion probability designs is (Berger, 2005)

$$\widehat{\mathrm{Var}}(\widehat{\tau}_\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \epsilon_{(i)}\epsilon_{(j)},$$

where

$$\epsilon_{(i)} = (1 - \frac{1}{\hat{N}\pi_i})(\widehat{\tau}_\pi - \widehat{\tau}_{\pi,-i}), \text{ for } i = 1, \ldots, n,$$

$\hat{N} = \sum_{i=1}^{n} \frac{1}{\pi_i}$, $\widehat{\tau}_\pi = \sum_{i=1}^{n} \frac{y_i}{\pi_i}$, and $\widehat{\tau}_{\pi,-i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{y_j}{\pi_j}$. One problem with this estimator is that it depends on the joint inclusion probabilities, π_{ij}. These aren't generally known for the pivotal or local pivotal method. This is the same

problem we encountered with the unbiased Horvitz–Thompson and Sen–Yates–Grundy estimators. Another problem with this estimator is that its formula involves dividing by π_{ij} which for some pairs of indices with the local pivotal method can be very close to 0. This will potentially give a vast overestimate of the variance, so this modern unequal inclusion probability design jackknife estimator will not be considered.

The classical jackknife resampling technique (Quenouille, 1949) computes "pseudovalues" which are constructed using values based on the modern jackknife approach. These "pseudovalues" are then used to estimate the variance of an estimator. The classical jackknife for estimating the variance of the total for an unequal inclusion probability without replacement sampling design for a finite population is as follows (see Wolter, 2007, p. 168).

First, as with the modern jackknife, we calculate estimates of the total with a different individual from the sample missing each time. These values are

$$\widehat{\tau}_{\pi,-i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{y_j}{\pi_j\left(\frac{n-1}{n}\right)}$$

for $i = 1, \ldots, n$. Note the factor of $\frac{n-1}{n}$ in the denominator of the summand which adjusts the inclusion probability value, π_j. This adjustment accounts for $\widehat{\tau}_{\pi,-i}$ estimating the total using $n-1$ observations instead of n observations.

Second, pseudovalues are computed from the $\widehat{\tau}_{\pi,-i}$. The pseudovalues are

$$\hat{\theta}_i = n\widehat{\tau}_{\pi} - (n-1)\widehat{\tau}_{\pi,-i},$$

for $i = 1, \ldots, n$, where $\widehat{\tau}_{\pi} = \sum_{i=1}^{n} \frac{y_i}{\pi_i}$. The pseudovalues are constructed so that $\hat{\bar{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}_i$ is an approximately unbiased estimate of the total, τ_{π} (Wolter,

2007, p. 152).

Finally, the estimated variance of the total is

$$\widehat{\mathrm{Var}}_{jac}(\widehat{\tau}_\pi) = \frac{1}{n(n-1)} \sum_{i=1}^{n} (\hat{\theta}_i - \hat{\bar{\theta}})^2.$$

The factor $\frac{1}{n(n-1)}$ results in the variance estimate being unbiased for a simple random sample with replacement design for a finite population (Wolter, 2007, p. 164). That factor is used above to attempt to generalize the results to unequal inclusion probability without replacement designs for finite populations.

We will test this classical jackknife resampling technique with local pivotal method samples. In all subsequent parts of this work, we will omit the adjective "classical" and call this technique "the jackknife." The next resampling technique we consider is bootstrapping.

3.2.2 Nonparametric Naive Bootstrap

The nonparametric bootstrap treats the sample as a new population (Efron, 1979). We then sample from this new population with replacement, compute the value of the estimator, and repeat this process many times after recording the value of the estimate. An estimate of the variance of the estimator is then found from the variance of the many computed estimates.

For example, if we have a sample $(y_1, y_2, \ldots, y_n)$, then we consider resampling with replacement on the set $\{y_1, y_2, \ldots, y_n\}$. We take such a sample of size n, writing that sample as $(y_1^*, y_2^*, \ldots, y_n^*)$. If we are interested in estimating the variance of the median of our original sample, then we compute the median of $(y_1^*, y_2^*, \ldots, y_n^*)$ and call that value M_1^*. We repeat this process many times, getting a collection of median bootstrap estimates $\{M_1^*, M_2^*, \ldots, M_b^*\}$, where b

is the number of times we have repeated this process. Then an estimate of the variance of the median of our original sample is the variance of these b bootstrap values. Bootstrapping has been widely used to estimate the variance of estimators for which no analytical expression for the variance exists (such as the median), and it has been shown to be a powerful tool in estimating these "difficult to find" variances.

Resampling using equal selection probabilities for each element of the sample, as in the median example above, is called naive bootstrapping (Barbiero, 2010). What makes it naive is that the way the original sample was selected is ignored and the resampling uses a random sample with replacement algorithm. The naive bootstrap is often used when the population is viewed as infinite.

The naive bootstrap will likely not give good estimates of the variance of the estimated population total when the sample is selected using the pivotal or local pivotal method because the pivotal and local pivotal methods can only be used when the population is finite. We will consider the naive bootstrap in subsequent analysis because no analysis of which we are aware demonstrates that naive bootstrapping gives poor estimates of the variance of the estimated total. Other bootstrap methods have been developed for the case where the population is finite. These are considered next.

3.2.3 Finite Population Bootstrap Methods

With a finite population, the naive bootstrap can lead to biased estimates of the variance. This occurs because using a random sample with replacement design in the bootstrap does not accurately reflect the nature of the population: it assumes, because of sampling with replacement, that the population

is infinite. Mashreghi classifies finite population bootstrap methods into three different types: pseudo–population bootstrapping, weighted bootstrapping, and direct bootstrapping (Mashreghi, 2016).

Pseudo–Population Bootstrapping

Pseudo–population bootstrapping (Gross, 1980) constructs a bootstrap population which attempts to mimic the overall population. Suppose that the original sample is $(y_1, y_2, \ldots, y_n)$ taken from a population of size N with a random sample without replacement algorithm. To form a pseudo–population, we form a set consisting of $y_1, y_2, \ldots, y_n$ repeated enough times so that the pseudo–population is of size N (or as close to N as is possible, different methods prescribe different ways of arriving at the pseudo–population size). The bootstrap procedure then selects a random sample without replacement from the pseudo–population. This pseudo–population bootstrap mimics the original sampling design in the resampling design.

In the case that the original sample was chosen with an unequal inclusion probability design, where the inclusion probabilities of the n individuals in the sample are $\pi_1, \pi_2, \ldots, \pi_n$, the pseudo–population is constructed from $\frac{1}{\pi_1}$ copies of y_1, $\frac{1}{\pi_2}$ copies of y_2, etc.. If $\frac{1}{\pi_i}$ is not an integer for some i in $1, \ldots, n$, Chauvet (2007) and Holmberg (1998) indicate to include $\lfloor \frac{1}{\pi_i} \rfloor$ copies of y_i for the first part of the pseudo–population. Then use Poisson sampling with inclusion probability $\frac{1}{\pi_i} - \lfloor \frac{1}{\pi_i} \rfloor$ to complete the pseudo–population. One sampling algorithm for Poisson sampling is to generate independent inclusion indicators I_i, $i = 1, 2, \ldots, n$, where $I_i \sim \text{Bin}(1, \frac{1}{\pi_i} - \lfloor \frac{1}{\pi_i} \rfloor)$ (Grafström, 2010, p. 86). Once the pseudo–population of size N is constructed, the same sampling design which produced the original sample is used on the pseudo–population

to select a bootstrap sample of size n.

The pseudo–population bootstrap could be useful for the pivotal method, but it will not produce good estimates for the local pivotal method as shown in the following. Suppose that $\pi_1 = 0.25$, then $\frac{1}{\pi_1} = 4$, so y_1 appears 4 times in the pseudo–population. Call those 4 appearances, $y_{1,1}, y_{1,2}, y_{1,3}$, and $y_{1,4}$ (with $\pi_{1,j} = 0.25$ for $j = 1, \ldots, 4$). If $y_{1,1}$ is chosen using the local pivotal method algorithm, then $y_{1,j}$ for $j = 2, 3$, or 4 will be chosen as the nearest neighbor, since all $y_{1,j}$ have identical auxiliary information to $y_{1,1}$. The winner of that competition will have updated inclusion probability 0.5. Continuing with the local pivotal method algorithm will likely result in one of the $y_{1,j}$ being selected (for $j = 1, \ldots, 4$) in the final bootstrap sample, since $\pi_{1,1}+\pi_{1,2}+\pi_{1,3}+\pi_{1,4} = 1$. The local pivotal method will likely select the same sample as the original sample. This would lead to a biased low estimate of the variance. For this reason, the pseudo–population bootstrap method will not be used in the later analysis.

Weighted Bootstrapping

Weighted bootstrapping (Rao, 1992) is associated with Bayesian Bootstrapping (Rubin, 1981) and generalized bootstrapping (Mason, 1992). For unequal inclusion probability designs, we rewrite the Horvitz–Thompson estimator as

$$\widehat{\tau}_\pi = \sum_{i=1}^{n} \frac{y_i}{\pi_i} = \sum_{i=1}^{n} \frac{1}{\pi_i} y_i = \sum_{i=1}^{n} w_i y_i$$

where $w_i = \frac{1}{\pi_i}$ are called the survey weights. Then $\widehat{\tau}_\pi$ is a weighted average of the sample observations. To compute a weighted bootstrap estimate of the

total, we compute

$$\widehat{\tau}_{WB} = \sum_{i=1}^{n} w_i^* y_i = \sum_{i=1}^{n} a_i^* w_i y_i$$

where $w_i^* = a_i^* w_i$ are the bootstrap weights which incorporate the survey weights. The survey weights, w_i, depend on the original sampling design. The a_i^* come from the bootstrap technique.

Bertail and Combris (1997) and Beaumont and Patak (2012) generate a_i^* from a distribution satisfying $\mathrm{E}^*(a_i^*) = 1$ and $\mathrm{E}^*((a_i^* - 1)(a_j^* - 1)) = \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}}$ where E^* is an expectation taken with respect to the a_i^*. These two properties of the a_i^* are chosen so that the weighted average, $\sum_{i=1}^{n} a_i^* w_i y_i$, is an unbiased estimator of the population total (from $\mathrm{E}^*(a_i^*) = 1$) and so that the variance of the weighted average is an unbiased estimator of the Horvitz–Thompson estimator variance (from $\mathrm{E}^*((a_i^* - 1)(a_j^* - 1)) = \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}}$).

To calculate a weighted bootstrap estimate of the variance of the total for an unequal inclusion probability design, we generate a_i^* for $i = 1, \ldots, n$ from a distribution with the properties given above, then we calculate $\widehat{\tau}_{WB} = \sum_{i=1}^{n} a_i^* w_i y_i = \sum_{i=1}^{n} a_i^* \frac{1}{\pi_i} y_i$. Each time we generate the a_i^*, we get another bootstrap estimate. We calculate b estimates of $\widehat{\tau}_{WB}$, and our estimate of the variance of the total is the variance of the b estimates. Weighted bootstrapping does not actually resample values from the original sample but instead generates weights which mimic how the resampling would have occurred.

Weighted bootstrap methods for unequal inclusion probability designs will likely produce large estimates of the variance of the total when a local pivotal method sample has been selected because the local pivotal method can have joint inclusion probabilities, π_{ij}, which are nearly zero. Then the $\mathrm{E}^*((a_i^* - 1)(a_j^* - 1)) = \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}}$ can be very large. This will likely lead to high variability

in the $\hat{\tau}_{WB}$ values, which will lead to a large estimate of the variance. For this reason weighted bootstrap methods will not be used in later analysis.

Direct Bootstrapping

Direct bootstrapping (Mashreghi, 2016) is any bootstrap method that samples from the original sample in some way without augmenting the sample. Direct bootstrap methods do not require the construction of a pseudo–population and actually resample from the sample itself to carry out estimation. If we were to use naive bootstrapping on a finite population, that would be considered a direct bootstrap method. Direct bootstrapping is different from weighted bootstrapping which does not resample but generates weights from a distribution which models the resampling process.

Mashreghi only discusses direct bootstrap methods for simple random sample without replacement designs. The methods discussed include Efron's original bootstrap (Efron, 1979) as well as methods by McCarthy and Snowden (1985), Rao and Wu (1988), and Sitter (1992). These methods (with the exception of Efron's) will not be considered here because they are only applicable to simple random sample without replacement designs on finite populations.

What we will do is consider a kind of direct bootstrap method, bootstrap subsampling, and demonstrate why it is promising for use with local pivotal method samples.

Direct Bootstrapping: Bootstrap Subsampling

"m out of n resampling" (Bickel, 1997) is a bootstrap method for simple random sample without replacement designs with infinite populations. Originally, m out of n resampling was developed to remedy naive bootstrap failures for

use with complicated estimators, like the 95th percentile. Especially where the estimator is not continuous, the naive bootstrap can produce poor estimates of the variance.

We will make m out of n resampling into a direct bootstrap method by applying it to a finite population, and we will use the local pivotal method in both the selection of the original sample of size n and in the selection of the subsample of size m. Part of what we will be testing is how the size of the subsample, m, affects the estimates of the variance. So that we do not have to always specify m, we will call our approach bootstrap subsampling instead of m out of n resampling. The following example will demonstrate how bootstrap subsampling can be useful with unequal inclusion probability designs.

Example 12. We generate a population of size $N = 1,024$. For each individual in the population, 3 values are generated: x_i, an auxiliary variable on which we want to balance; z_i, an auxiliary variable which will generate the inclusion probabilities; and w_i, the response variable. To generate each x_i, we randomly select an integer between 1 and 10 inclusive. Then $z_i = x_i + \epsilon_i$ where $\epsilon_i \sim \text{Unif}(0, 1)$, and $w_i = x_i + \eta_i$ where $\eta_i \sim \text{N}(0, 0.2)$ for $i = 1, \ldots, N$.

From this simulated population, we select one sample of size $n = 512$ using the local pivotal method. To generate the inclusion probability vector for the original sample, we use Equation 1.1 (page 7):

$$\pi_i = n \frac{z_i}{\sum_{j=1}^{N} z_j}$$

for $i = 1, \ldots, N$.

From this one sample of size $n = 512$, we select $2,000$ local pivotal method subsamples which are $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$ the size of the original sample. This

corresponds to subsamples of size $m = 256, 128, 64, 32$, and 16, respectively. To compute the inclusion probabilities for a given subsample of size m, we compute

$$\pi'_{[i]} = m\frac{z_{[i]}}{\sum_{j=1}^{n} z_{[j]}} \tag{3.1}$$

for $i = 1, \ldots, n$, where $z_{[i]}$ is an auxiliary variable value for individual i from the sample. The same procedure from Chapter 1 applies if $\pi'_{[i]} \geq 1$ for any i (see page 7). Note that z_i is the value from individual i from the population ($i = 1, \ldots, N$), and $z_{[i]}$ is the value from individual i from the sample ($i = 1, \ldots, n$). The updated inclusion probabilities, $\pi'_{[i]}$, and the auxiliary information, $x_{[i]}$, are used to select local pivotal method subsamples for a given subsample of size m.

Once a subsample is selected, we can estimate the total of the response variable, w, in two ways:

$$\widehat{\tau}_{\text{orig}} = \sum_{j=1}^{m} \frac{w_{[[j]]}}{\pi_{[[j]]}}$$

or

$$\widehat{\tau}_{\text{updat}} = \sum_{j=1}^{m} \frac{w_{[[j]]}}{\pi'_{[[j]]}}$$

where $w_{[[j]]}$ is the response value for individual j in the subsample for $j = 1, \ldots, m$ (and likewise for $\pi_{[[j]]}$ and $\pi'_{[[j]]}$).

First consider the estimated totals from estimator $\widehat{\tau}_{\text{updat}}$ shown on the right side of Figure 3.1. The total from the one original sample of $n = 512$ is $\sum_{i=1}^{n} w_{[i]} = 3,559.3$, so $\widehat{\tau}_{\text{updat}}$ is an unbiased estimator for the original sample total for all subsample sizes. The variability in the estimates also increases as the subsample size decreases. Both of these results are what we expect.

Now consider the estimated totals from estimator $\widehat{\tau}_{\text{orig}}$ shown on the left side

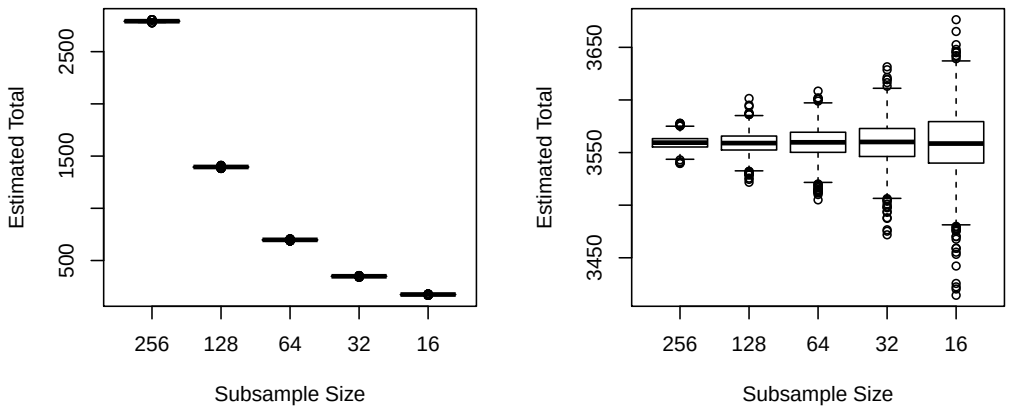

Figure 3.1: Boxplots of estimated totals from $2,000$ subsamples for indicated subsample sizes. The totals are estimated using $\widehat{\tau}_{\text{orig}}$ on the left and for $\widehat{\tau}_{\text{updat}}$ on the right.

of Figure 3.1. The estimated totals decrease as the subsample size decreases. This makes sense since we are estimating the total using inclusion probabilities for a sample of size 512 but using less than 512 individuals in each estimate. It is remarkable, and difficult to observe in the plot, that the variability in the estimates also decreases as the subsample size decreases. For subsamples of size $m = 256$, 128, 64, 32, and 16 the variance of these $2,000$ estimates are 21.3, 15.2, 8.4, 4.1, and 2.3, respectively. There is currently no known way to choose an optimal m. The five m values above were chosen so that the subsample size would be $m = \frac{512}{2^k}$ for $k = 1, 2, 3, 4, 5$. We want to rescale these variances to estimate the variance of the total.

This example demonstrated how bootstrap subsampling could work on a simulated population. Bootstrap subsampling is a kind of direct bootstrap method, and our goal in considering different bootstrap methods is to estimate the variance of the estimated total for samples selected using the local pivotal

method. From this example, we see that $\widehat{\tau}_{\text{orig}}$ produces underestimates of the population total (the true population total is $5,586.0$). This is not a concern because the Horvitz–Thompson estimator using the original sample is an unbiased estimator of the population total. We are interested in the *variability* of the $2,000$ $\widehat{\tau}_{\text{orig}}$ values.

We propose to rescale the variance of $\widehat{\tau}_{\text{orig}}$ in a similar way that D'Orazio rescales the simple random sample estimator (see page 65). The variance estimator we propose is

$$\widehat{\text{Var}}_{BooSub}(\widehat{\tau}_{\pi}) = \sqrt{\frac{n}{m}} \cdot \widehat{\text{Var}}\big(\widehat{\tau}_{\text{orig}}\big).$$

This estimator with the rescaling value will be called the bootstrap subsample variance estimator. The value $\sqrt{\frac{n}{m}}$ attempts to correct for the decreasing variance in $\widehat{\tau}_{\text{orig}}$ as the subsample size, m, decreases. We will evaluate this estimator on the example datasets in Chapter 4.

From Mashreghi's classification of finite population bootstrap methods (Mashreghi, 2016), we will not consider pseudo–population bootstrapping or weighted bootstrapping. We will consider one kind of direct bootstrapping, bootstrap subsampling (Bickel, 1988), which led to our proposed estimator, the bootstrap subsample variance estimator.

The bootstrap subsample variance estimator will be compared to the simple random sample variance estimator, the local variance estimator (Stevens, 2003), the nearest neighbor estimator (Wolter, 2007), the jackknife (Quenouille, 1949), and the naive bootstrap (Efron, 1979). We will compare these variance estimators in Chapter 4.

In Chapter 2, we developed the properties of the pivotal and local pivotal

methods which can be used to select a sample with an unequal inclusion probability design. With the local pivotal method, we now have a way to select a sample while balancing on given auxiliary information and get an estimate of the population total using the Horvitz–Thompson estimator. Different ways of calculating the variability of that estimate of the total has been the focus of this chapter. In the next chapter we evaluate those variance estimators.

Chapter 4

Performance of Estimators

In this chapter we will simulate sampling from several known populations which have different characteristics to compare the performance of the variance estimators from Chapter 3. Five populations will be used.

The first three populations are simulated ecological data where the auxiliary information on which we want to balance is spatial coordinates generated by a Matern cluster process. Very often unequal inclusion probability designs are used in Ecology (Nahorniak, 2015), so these datasets demonstrate how the local pivotal method can be utilized in an ecological setting.

The fourth population is simulated income data where the auxiliary information on which we want to balance is one dimensional. Because the distribution of the response variable for this population is right skewed and we sample a large proportion of the population, this population will be good for evaluating variance estimators (Antal, 2011).

The fifth population is real data from Baltimore, MD, USA using location and age of houses as the auxiliary information on which we want to balance, using house square footage to calculate inclusion probabilities, and using house

price as the response variable. This dataset demonstrates how the local pivotal method can balance on auxiliary information which is more than two dimensions and evaluates the variance estimators on real data.

Simulations are necessary to estimate the true value of the variance for the Horvitz–Thompson estimate of the total using local pivotal method samples since there is no way to compute the true variance. For each population, we will simulate $10,000$ local pivotal method samples. On each sample, we will compute the Horvitz–Thompson estimate of the total, $\widehat{\tau}_\pi$. The Monte Carlo variance of the estimates is

$$\widehat{\mathrm{Var}}_{est} = \frac{1}{10,000} \sum_{j=1}^{10,000} (\widehat{\tau}_{\pi,j} - \tau)^2$$

where $\widehat{\tau}_{\pi,j}$ is the Horvitz–Thompson estimate of the total using the jth sample, and τ is the true total for the population (which we will know for our populations). This variability in the estimates will be called the estimated true variance, and it will be treated as the true variance.

On each of the $10,000$ samples the variance will also be calculated using the simple random sample variance estimator, the local variance estimator, the nearest neighbor variance estimator, the jackknife estimator, the naive bootstrap estimator, and the bootstrap subsample variance estimator with subsamples of size $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{5}$, and $\frac{1}{8}$ of the original sample size. These fractions are arbitrarily chosen to represent a variety of possible subsample sizes. These 9 estimators of the variance will then be compared to the estimated true variance.

4.1 Performance Criteria

After simulation, we will have $10,000$ estimates of the variance for each of the 9 variance estimators. Numerically, these estimates are compared to the estimated true variance using relative bias percent, relative root mean square error percent, and confidence interval coverage percent. Visually, these estimates are compared to the estimated true variance using side–by–side boxplots.

Several authors use the relative bias percent of variance estimators as a measure of performance. We define relative bias percent as in Barbiero (2010) and Antal (2011):

$$\text{RBP } = 100 \times \frac{\frac{1}{10,000} \cdot \sum_{j=1}^{10,000} \widehat{\text{Var}}_{*,j}(\widehat{\tau}_\pi) - \widehat{\text{Var}}_{est}}{\widehat{\text{Var}}_{est}}$$

where $\widehat{\text{Var}}_{*,j}(\widehat{\tau}_\pi)$ is the estimated variance from simulation j for one of the 9 variance estimators. The $*$ in $\widehat{\text{Var}}_{*,j}(\widehat{\tau}_\pi)$ is a placeholder for one of the 9 different variance estimators. For example, $\widehat{\text{Var}}_{NN,j}(\widehat{\tau}_\pi)$ is the estimated variance using the Nearest Neighbor variance estimator.

Another measure of performance of the variance estimators is the relative root mean square error percent. We define relative root mean square error percent as in Antal (2011):

$$\text{RRMSEP} = 100 \times \frac{\sqrt{\frac{1}{10,000} \cdot \sum_{j=1}^{10,000} (\widehat{\text{Var}}_{*,j}(\widehat{\tau}_\pi) - \widehat{\text{Var}}_{est})^2}}{\widehat{\text{Var}}_{est}}.$$

For the final numeric measure of performance we calculate the percent of the time confidence intervals built with a variance estimate capture the true population total. This confidence interval coverage percent is used in D'Orazio

(2003), Antal (2011), and Stevens (2003). We compute 95% confidence interval coverage by the percent of the time, in $10,000$ simulations, that the following inequality is satisfied

$$\widehat{\tau}_{\pi,j} - t^* \cdot \sqrt{\widehat{\mathrm{Var}}_{*,j}(\widehat{\tau}_\pi)} \leq \tau \leq \widehat{\tau}_{\pi,j} + t^* \cdot \sqrt{\widehat{\mathrm{Var}}_{*,j}(\widehat{\tau}_\pi)}$$

where $\widehat{\tau}_{\pi,j}$ is the Horvitz–Thompson estimate of the total from the jth sample (for $j = 1, \ldots, 10,000$), and t^* is the critical value of the upper 2.5th percentile from a t distribution with $n - 1$ degrees of freedom.

A normal approximation is appropriate in the case that the sample size is greater than 50 (Thompson, 2012, p. 49), but to conform with the authors who utilize confidence interval coverage percent and because not all of our simulated sample sizes are greater than 50, only t–distribution intervals will be computed.

These three numeric performance criteria as well as side–by–side boxplots will be used to evaluate the 9 variance estimators in the following datasets.

4.2 Matern Generated Datasets

4.2.1 Description of Datasets

Since "[d]ata from ecological samples are often comprised of spatially correlated and/or clustered metrics" (Nahorniak, 2015, p. 7), we decided to generate the auxiliary information on which we want to balance using a spatially isotropic method where the clustering can be controlled. To accomplish this, we use a Matern cluster process (generated in R by *rMatClust* from the *spatstat* package (Baddeley, 2018); see Appendix B of this document, page 123,

for the R code which generates these populations).

The function *rMatClust* takes 3 arguments: κ, the intensity of the Poisson process which generates the cluster centers; *scale*, the radius of the different clusters around each center; and μ, the mean number of points per cluster, sometimes called "offspring". We fix $\kappa = 0.2$ and $\mu = 8$, then we generate 3 different sets of auxiliary information by setting *scale* to $1.0, 0.75$, or 0.5. Each set of auxiliary information is generated on a 25 by 25 grid.

For example, if $\kappa = 0.2$, *scale*$= 0.5$, and $\mu = 8$, then we expect 0.2 cluster centers per unit square with, on average, 8 points per cluster which are at distance less than 0.5 units from each cluster center. Then per unit square we expect $0.2 * 8 = 1.6$ points, so on the entire 25 by 25 grid we expect $1,000$ points (since $25 * 25 * 1.6 = 1,000$). Since this Matern cluster process incorporates randomness in the number of points per cluster and the number of cluster centers per unit square, the population size will not be known until the auxiliary information is generated. The expected population size is $N = 1,000$, but different realizations of the process will lead to slightly different population sizes. Changing the *scale* parameter will not change the expected population size; it will generate auxiliary information which is more (*scale* $= 0.5$) or less (*scale* $= 1.0$) clustered. The three generated populations of auxiliary information are shown in Figure 4.1 (page 86).

We use the $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates to generate the response variable $\boldsymbol{w}$. From $\boldsymbol{w}$ we generate another auxiliary information variable, $\boldsymbol{z}$, which will be used to create the inclusion probability vector. We estimate the total for $\boldsymbol{w}$ using local pivotal method samples which balance on the $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates while also respecting the inclusion probabilities generated by $\boldsymbol{z}$.

The variables $\boldsymbol{w}$ and $\boldsymbol{z}$ described below will be generated in the same way

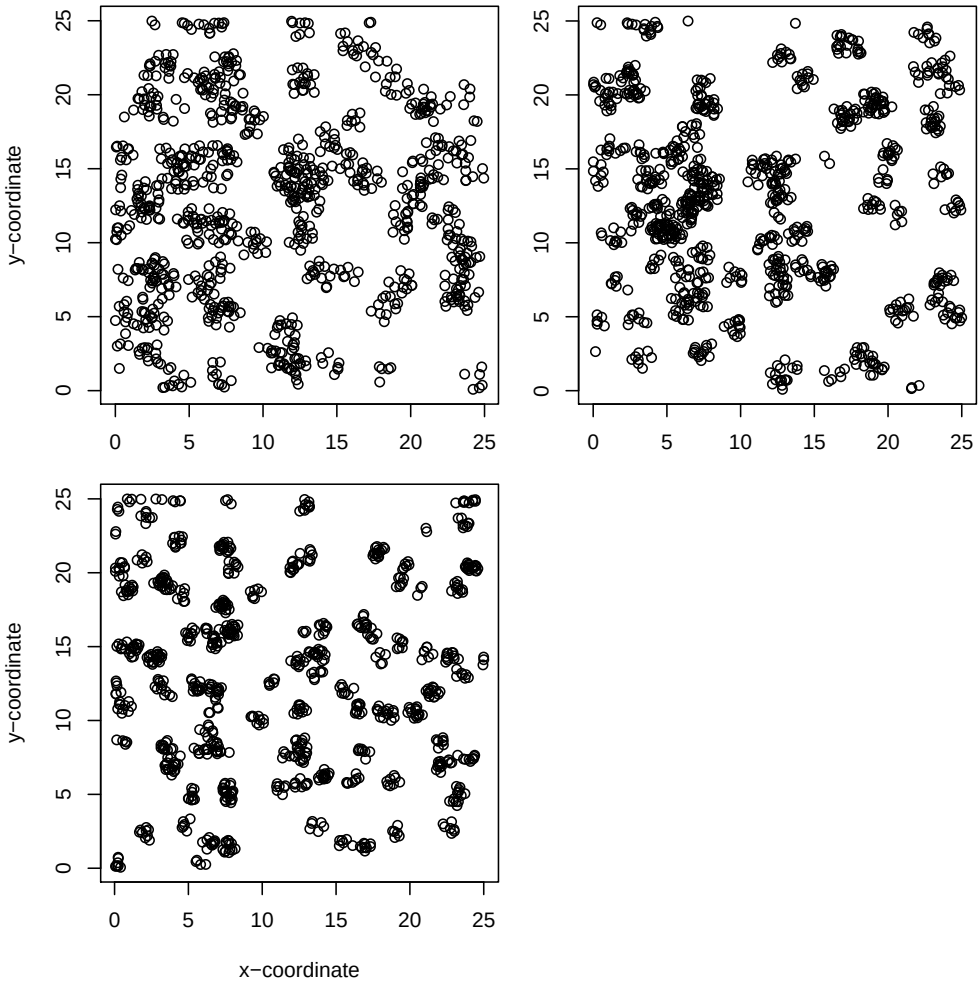

Figure 4.1: Two dimensional auxiliary information generated by a Matern cluster process using $scale = 1$ (top left plot, population size: $N = 918$), $scale = 0.75$ (top right plot, population size: $N = 912$), and $scale = 0.5$ (bottom left plot, population size: $N = 893$).

for all 3 sets of auxiliary information shown in Figure 4.1. To create the response variable, $\boldsymbol{w}$, we wanted a variable which could be thought of as the heights of trees and ranged between 0 and around 150. The response variable is

$$w_i = 200\frac{-(x_i-2)(x_i-8)^2+7,500}{12,205} + 200\frac{(y_i-10)(-y_i+25)+2,600}{31,324} + \epsilon_{i,1}$$

where $\epsilon_{i,1} \sim N(0,5)$, and $i = 1,\ldots,N$. If we plot the $\boldsymbol{w}$ values without the $\epsilon_{i,1}$, then the surface is shown in Figure 4.2 (page 88). Then

$$z_i = \sqrt{w_i} + \epsilon_{i,2}$$

where $\epsilon_{i,2} \sim N(0,0.1)$, and $i = 1,\ldots,N$. The equation for $\boldsymbol{w}$ is motivated by a desire for the response variable to take positive values which range from nearly 0 to 150. Then the auxiliary information, $\boldsymbol{z}$, is fairly strongly positively associated with the response. The inclusion probabilities will be constructed from $\boldsymbol{z}$ using equation 1.1 (page 7). For implementation of the data generation in R see Appendix B, page 123.

For each population, we select $10,000$ samples of size $n = 50$, $n = 100$, $n = 150$, and $n = 200$. For the naive bootstrap and bootstrap subsample variance estimators we select $2,000$ bootstrap samples and $2,000$ bootstrap subsamples for each of the $10,000$ original samples for each sample size. We compare the resulting variance estimates using side–by–side boxplots in the following section.

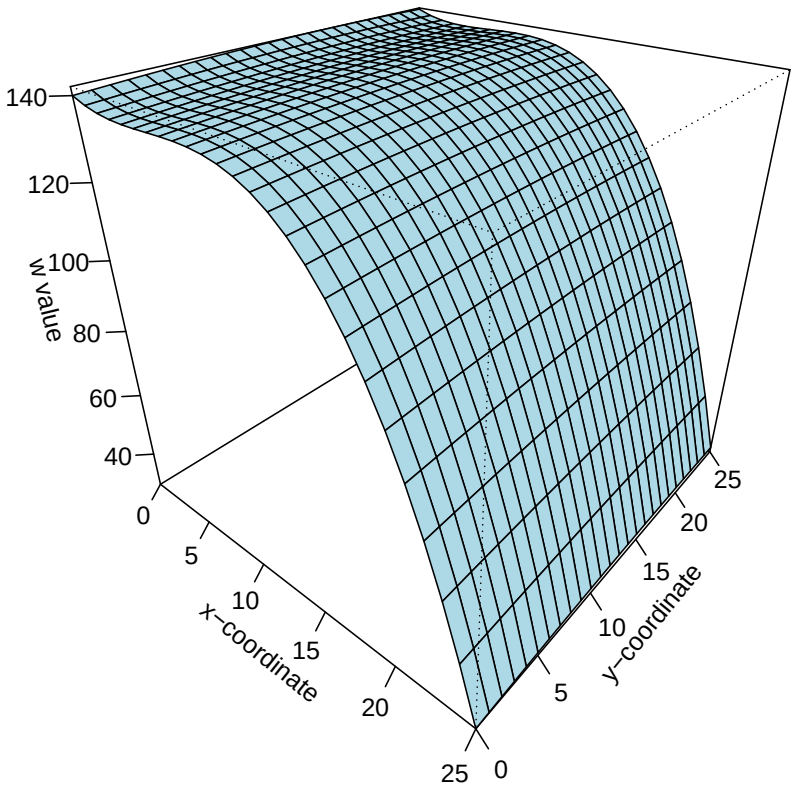

Figure 4.2: Surface of response variable $\boldsymbol{w}$ without $\epsilon_{i,1}$.

4.2.2 Visual Performance of Estimators

To make the relative size of the estimates easier to compare, we plot the standard deviations instead of the variances in all subsequent plots. Each set of boxplots will display the estimated true standard deviation for the sample size as a red horizontal line. The estimated true standard deviation is the square root of the estimated true variance defined on page 82.

In all subsequent plots the variance estimators are abbreviated in the following way: "NN" is the nearest neighbor estimator, "Local" is the local variance estimator, "Sub m" is the boostrap subsample estimator using subsamples of size m, "jk" is the jackknife variance estimator, "nb" is the naive bootstrap variance estimator, and "srs" is the simple random sample variance estimator.

The simple random sample, naive bootstrap, and jackknife variance estimators all tend to overestimate the estimated true standard deviation in the same way, regardless of the sample size, n, or the amount of clustering in the auxiliary information, *scale*. We display one example of this phenomenon in Figure 4.3 for $n = 200$ from the population with auxiliary information generated using $scale = 1.0$. It is difficult to compare the best performing estimators in Figure 4.3 because of the inclusion of the three worst performing estimators. In all subsequent plots, we will omit the simple random sample, naive bootstrap, and jackknife variance estimators.

For the population with auxiliary information generated using $scale = 1.0$, the nearest neighbor and local variance estimators perform best for all sample sizes (Figure 4.4, page 92). As the sample size increases, the nearest neighbor estimator tends to overestimate the estimated true standard deviation more

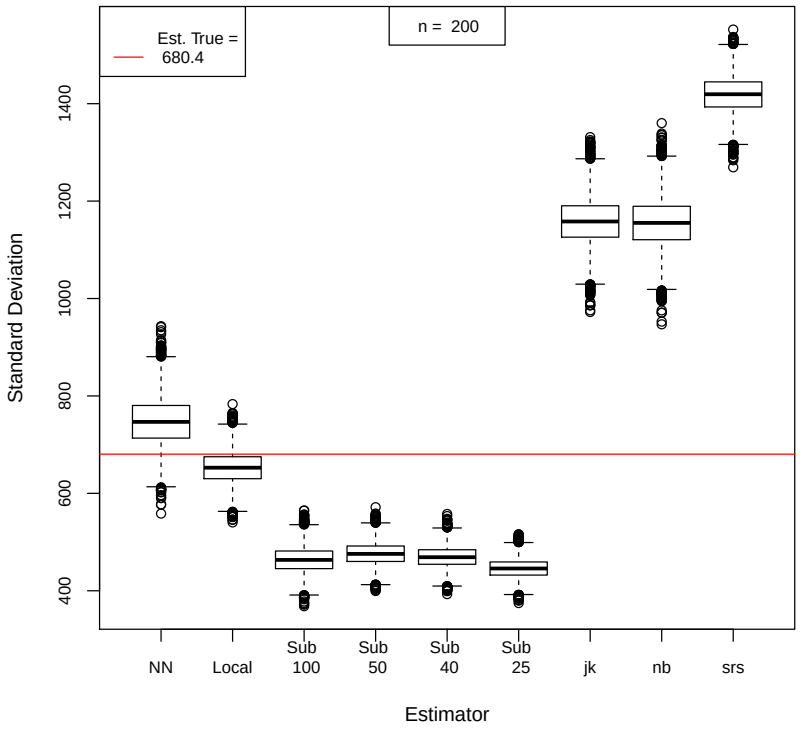

Figure 4.3: Boxplots of standard deviation estimates for all 9 estimators for samples of size $n = 200$. The population size is $N = 918$ with auxiliary information generated by a Matern cluster process with $scale = 1.0$.

and more (for more information on this phenomenon, see Appendix C (page 129)). The bootstrap subsample estimators generally underestimate the estimated true standard deviation; although, when the subsample size is not a whole number, the bootstrap subsample estimators perform nearly as well as the local variance estimator (see especially $n = 150$ in Figure 4.4).

For the population with auxiliary information generated using $scale = 0.75$, the nearest neighbor and local variance estimators perform best for all sample sizes (Figure 4.5, page 93). The same patterns noted in Figure 4.4 are also present in Figure 4.5.

For the population with auxiliary information generated using $scale = 0.5$, the nearest neighbor and local variance estimators perform best for all sample sizes (Figure 4.6, page 94). The same patterns noted in both Figures 4.4 and 4.5 are also present in Figure 4.6.

4.2.3 Numeric Performance of Estimators

The numeric performance criteria discussed here are those defined on page 83. We only consider the 6 best estimators from the boxplot analysis for numeric comparison.

For the population with auxiliary information with the least clustering ($scale = 1$), the nearest neighbor estimator has relative bias percent closest to 0 among the 6 estimators for small sample sizes (Table 4.1, page 95). For larger sample sizes, the local variance estimator has relative bias percent closest to 0. In populations with auxiliary information which is more clustered ($scale = 0.75$ or $scale = 0.5$), the local variance estimator almost always has relative bias percent closest to 0. The bootstrap subsample estimators using noninteger subsample sizes have relative bias percents closer to 0 than the nearest neighbor

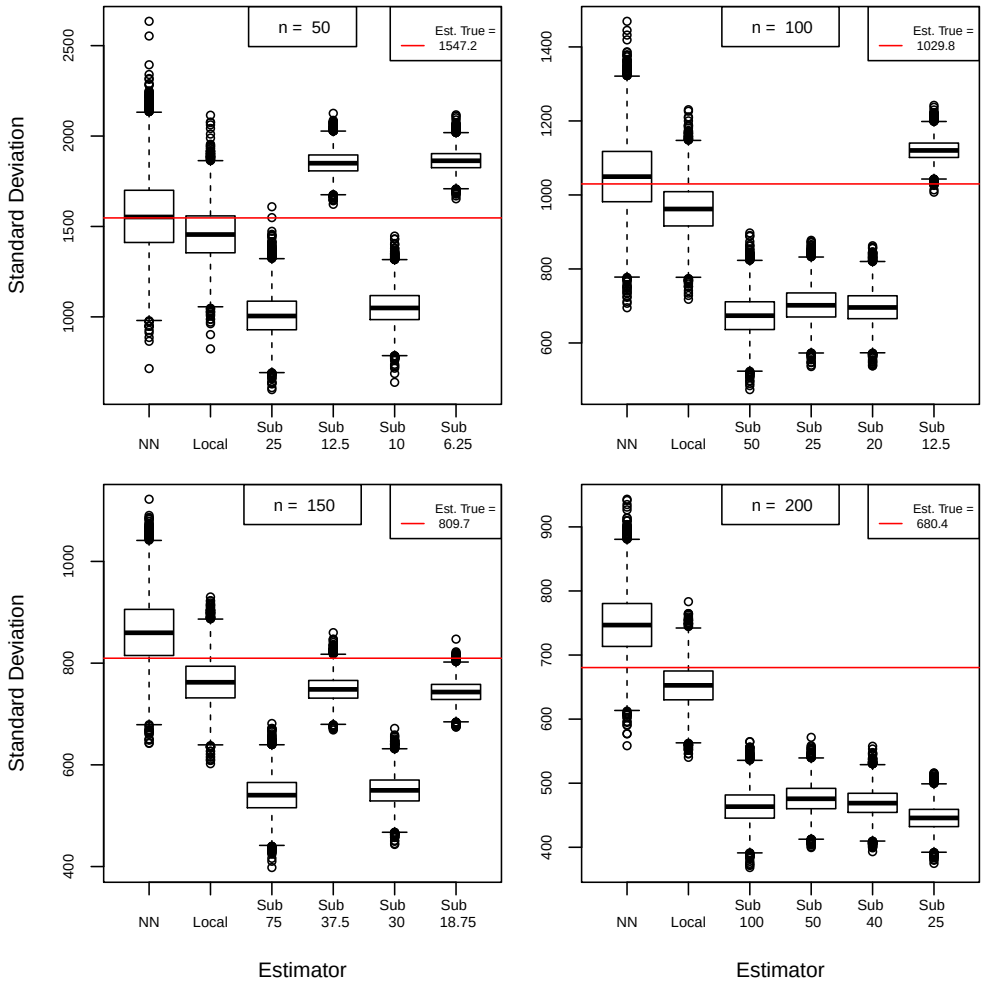

Figure 4.4: Boxplots of standard deviation estimates for the best 6 estimators. The population size is $N = 918$ with auxiliary information generated by a Matern cluster process with $scale = 1.0$.

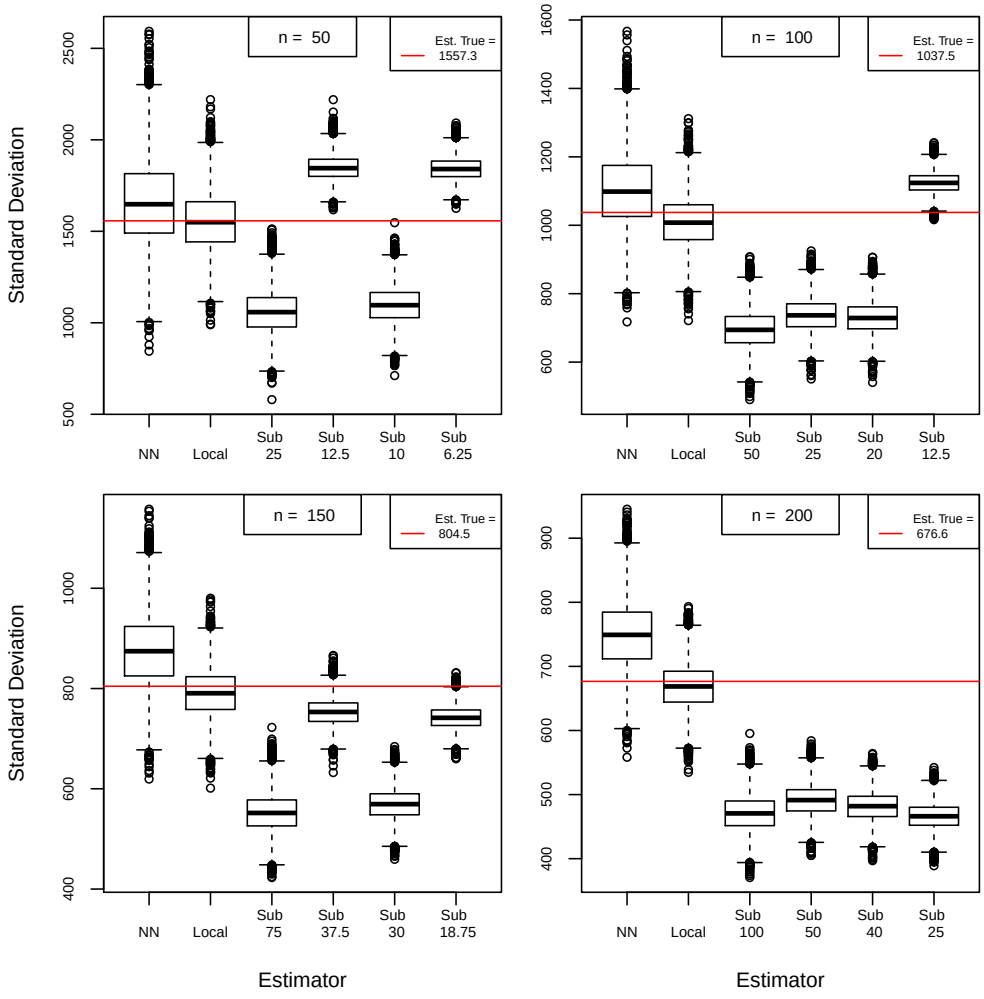

Figure 4.5: Boxplots of standard deviation estimates for the best 6 estimators. The population size is $N = 912$ with auxiliary information generated by a Matern cluster process with $scale = 0.75$.

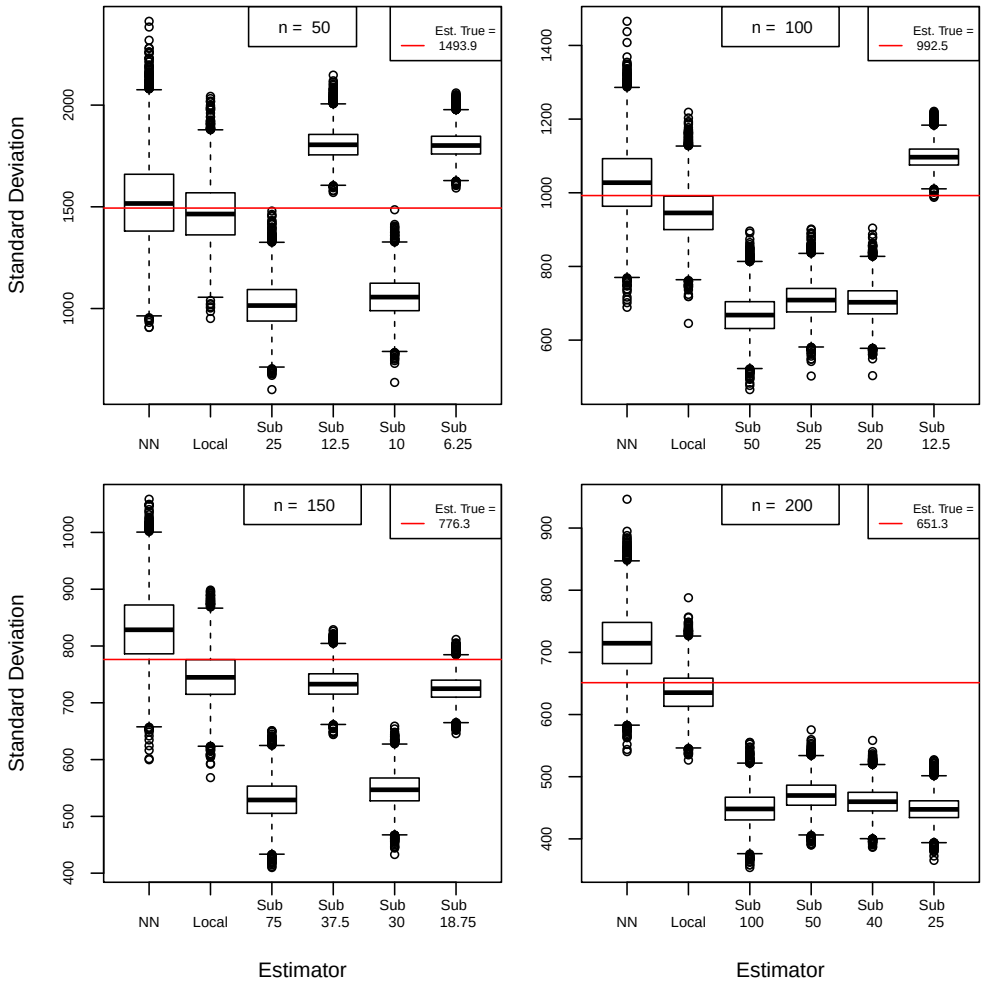

Figure 4.6: Boxplots of standard deviation estimates for the best 6 estimators. The population size is $N = 893$ with auxiliary information generated by a Matern cluster process with $scale = 0.5$.

scale (clustering)	Sample Size	Nearest Neighbor	Local Variance	Bootstrap Subsample Size: $\frac{n}{2}$	$\frac{n}{4}$	$\frac{n}{5}$	$\frac{n}{8}$
1 (least clustering)	$n = 50$	3.3	-10.2	-56.9	43.5	-53.3	45.4
	$n = 100$	5.0	-12.2	-56.8	-53.2	-54.0	18.6
	$n = 150$	13.7	-10.8	-55.2	-14.4	-53.7	-15.6
	$n = 200$	21.2	-7.7	-53.4	-51.0	-52.3	-57.0
0.75 (medium clustering)	$n = 50$	15.6	0.7	-53.0	41.1	-49.9	40.2
	$n = 100$	13.9	-4.8	-54.8	-49.3	-50.3	17.6
	$n = 150$	19.2	-3.0	-52.7	-12.2	-49.8	-14.9
	$n = 200$	23.3	-2.1	-51.4	-47.1	-49.2	-52.4
0.5 (most clustering)	$n = 50$	6.1	-2.5	-53.0	46.6	-49.4	46.1
	$n = 100$	8.4	-8.7	-54.4	-48.8	-49.6	22.4
	$n = 150$	14.8	-7.5	-53.3	-10.6	-50.1	-12.6
	$n = 200$	21.4	-4.4	-52.3	-47.7	-50.0	-52.6

Table 4.1: Matern generated dataset relative bias percent for the best 6 estimators for 3 different *scale* values and 4 different original sample sizes.

scale (clustering)	Sample Size	Nearest Neighbor	Local Variance	Bootstrap Subsample Size: $\frac{n}{2}$	$\frac{n}{4}$	$\frac{n}{5}$	$\frac{n}{8}$
1 (least clustering)	$n = 50$	28.5	21.2	57.8	44.7	54.1	46.3
	$n = 100$	20.7	17.5	57.3	53.6	54.4	19.6
	$n = 150$	22.6	15.2	55.5	15.5	54.0	16.3
	$n = 200$	26.6	12.1	53.7	51.2	52.5	57.1
0.75 (medium clustering)	$n = 50$	36.6	20.9	54.1	42.5	50.7	41.3
	$n = 100$	26.9	15.1	55.3	49.7	50.7	18.7
	$n = 150$	27.6	12.2	53.1	13.8	50.1	15.8
	$n = 200$	29.2	10.5	51.7	47.4	49.4	52.6
0.5 (most clustering)	$n = 50$	29.6	20.0	54.1	48.1	50.3	47.3
	$n = 100$	22.0	15.7	54.9	49.2	50.1	23.5
	$n = 150$	23.0	13.4	53.6	12.4	50.4	13.7
	$n = 200$	27.0	10.9	52.6	48.0	50.2	52.8

Table 4.2: Matern generated dataset relative root mean square error percent for the best 6 estimators for 3 different *scale* values and 4 different original sample sizes.

scale (clustering)	Sample Size	Nearest Neighbor	Local Variance	Bootstrap Subsample Size: $\frac{n}{2}$	$\frac{n}{4}$	$\frac{n}{5}$	$\frac{n}{8}$
1 (least clustering)	$n = 50$	95.0	94.2	80.3	98.4	82.1	98.5
	$n = 100$	95.0	93.4	80.3	82.3	82.0	96.9
	$n = 150$	96.1	93.6	80.8	93.3	81.7	93.0
	$n = 200$	96.7	94.0	81.9	83.1	82.5	80.2
0.75 (medium clustering)	$n = 50$	96.0	95.3	82.2	98.4	83.9	98.4
	$n = 100$	96.2	94.5	81.6	84.1	83.5	97.0
	$n = 150$	96.5	94.6	82.1	93.5	83.5	93.0
	$n = 200$	96.8	94.6	82.9	84.7	84.0	82.6
0.5 (most clustering)	$n = 50$	95.4	94.6	82.2	98.6	83.8	98.5
	$n = 100$	95.6	93.8	81.8	84.4	84.0	97.3
	$n = 150$	96.4	94.1	81.8	93.9	83.2	93.6
	$n = 200$	96.9	94.6	82.2	84.2	83.4	82.3

Table 4.3: Matern generated dataset 95% confidence interval coverage percent for the best 6 estimators for 3 different *scale* values and 4 different original sample sizes.

estimator on the more clustered datasets for larger sample sizes. For example, for $n = 150$ in both the medium clustering and most clustering datasets, the bootstrap subsample estimators using subsamples of size $\frac{n}{4} = 37.5$ and $\frac{n}{8} = 18.75$ have relative bias percents closer to 0 than the nearest neighbor estimator.

For all populations and nearly all sample sizes, the local variance estimator has the smallest relative root mean square error percent (Table 4.2, page 95). The one exception to this is the bootstrap subsample variance estimator using subsamples of size $\frac{n}{4} = 37.5$ which has a smaller relative root mean square error percent than the local variance estimator for the most clustered population ($scale = 0.5$) and for $n = 150$.

For the population with auxiliary information with the least clustering

(*scale* = 1), the nearest neighbor estimator has confidence interval coverage percent closest to 95 among the 6 estimators for small sample sizes (Table 4.3, page 96). In populations with auxiliary information which are more clustered (*scale* = 0.75 or *scale* = 0.5), the local variance estimator almost always has confidence interval coverage percent closest to 95.

4.2.4 Matern Generated Datasets Conclusions

For these datasets, the local variance estimator generally works best. The nearest neighbor estimator performs as well as, or better than, the local variance estimator when sample sizes are small ($n = 50$ or $n = 100$). The nearest neighbor estimator tends to perform worse as the sample size increases. The bootstap subsample estimators almost always perform worse than the local variance and nearest neighbor estimators. The bootstrap subsample estimators perform similarly across sample sizes and clustering (for integer subsample sizes), so a different scaling value than $\sqrt{\frac{n}{m}}$ (see page 79 for the form of the estimator) could produce estimates which are closer to the estimated true variance.

4.3 Simulated Income Dataset

4.3.1 Description of Dataset

The fourth population we consider mimics income distributions (see Appendix B, page 123, for R code), and the population is generated in the same way as in Antal (see 2011, p. 7 or 2014, p. 14).

We generate $\boldsymbol{x}$, the auxiliary information on which we want to balance,

from a Normal distribution. Using $\boldsymbol{x}$ we generate the response variable, $\boldsymbol{w}$, then we use $\boldsymbol{w}$ to generate $\boldsymbol{z}$, the auxiliary information which is used to create the inclusion probability vector. We estimate the total for $\boldsymbol{w}$ using local pivotal method samples which balance on $\boldsymbol{x}$ while also respecting the inclusion probabilities generated by $\boldsymbol{z}$.

The population size is $N = 150$ and the response, $\boldsymbol{w}$, are intended to have a right skewed distribution to simulate income distributions. A histogram of the distribution of the $\boldsymbol{w}$ is shown in Figure 4.7 (page 98). The $\boldsymbol{w}$ are generated

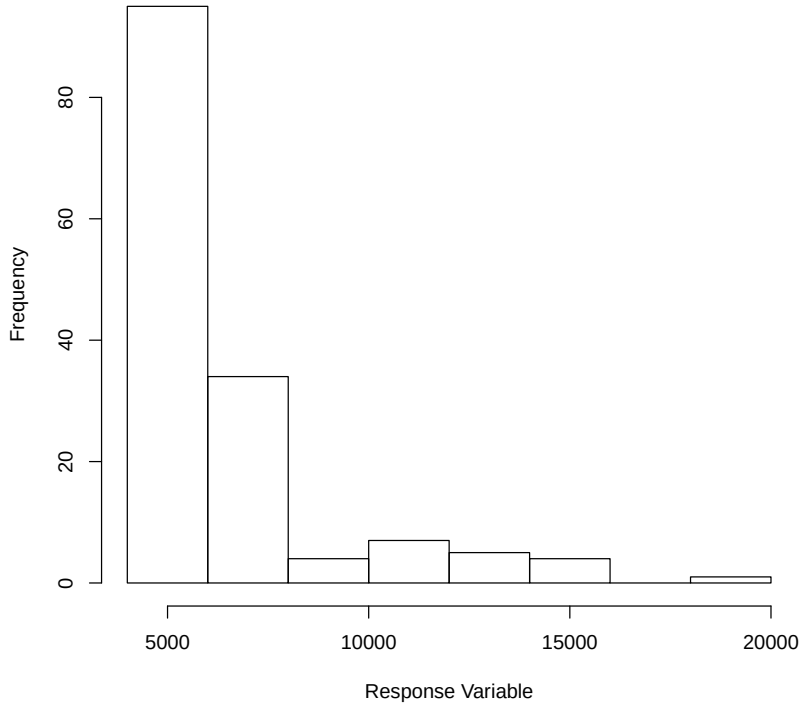

Figure 4.7: Histogram of the response variable, $\boldsymbol{w}$, for simulated income dataset.

by

$$w_i = (12.5 + 3x_i^{1.2} + 15\epsilon_i)^2 + 4,000$$

where $x_i \sim |\mathrm{N}(0,7)|$ and $\epsilon_i \sim \mathrm{N}(0,1)$ for $i = 1, \ldots, 150$. The x_i values come from the absolute value of realizations of a normal distribution. Then

$$z_i = w_i^{0.2} \cdot p_i$$

where $p_i \sim \mathrm{logN}(0, 0.25)$ for $i = 1, \ldots, 150$. We use $\boldsymbol{z}$ to calculate the unequal inclusion probability vector (using equation 1.1, page 7).

Once the population is generated, we select $10,000$ samples of size $n = 25$, $n = 50$, $n = 75$, and $n = 100$. On each of these samples, we compute the estimated variance using the simple random sample variance estimator, the nearest neighbor variance estimator, and the jackknife estimator. For the naive bootstrap and bootstrap subsample variance estimators, we select $2,000$ bootstrap samples and $2,000$ bootstrap subsamples for each of the $10,000$ original samples for each sample size. The local variance estimator cannot be calculated on this population because the auxiliary information on which we want to balance is not two dimensional.

The differences between this population and the Matern generated populations include: this population has a smaller and fixed population size ($N = 150$ versus an expected population size of $1,000$); the auxiliary information on which we want to balance is one dimensional instead of two dimensional; and the distribution of the response variable in this population is right skewed, whereas the distribution of the response variable in all three of the Matern generated populations is left skewed. We next compare the variance estimates from this population using side–by–side boxplots.

4.3.2 Visual Performance of Estimators

To make the relative size of the estimates easier to compare, we plot the standard deviations instead of the variances in the following plots. Each set of boxplots will display the estimated true standard deviation for the sample size as a red horizontal line. The estimated true standard deviation is the square root of the estimated true variance defined on page 82.

The simple random sample, naive bootstrap, and jackknife variance estimators all tend to overestimate the estimated true standard deviation in the same way, regardless of the original sample size, n. We display one example of this phenomenon in Figure 4.8 (page 101) for $n = 50$. It is difficult to compare the best performing estimators in Figure 4.8 because of the inclusion of the three worst performing estimators. In all subsequent plots, we will omit the simple random sample, naive bootstrap, and jackknife variance estimators.

For small sample sizes ($n = 25$), the nearest neighbor estimates are centered closer to the estimated true standard deviation than any of the bootstrap subsample estimates (Figure 4.9, page 102). As the sample size increases, the nearest neighbor estimator tends to overestimate the estimated true standard deviation more and more. The bootstrap subsample estimators which use fractional subsample sizes perform best for samples of size $n = 50$ and $n = 75$. For samples of size $n = 100$, the bootstrap subsample estimators which use integer subsample sizes performs best.

4.3.3 Numeric Performance of Estimators

The numeric performance criteria discussed here are those defined on page 83. We consider only the 5 best estimators from the boxplot analysis for numeric

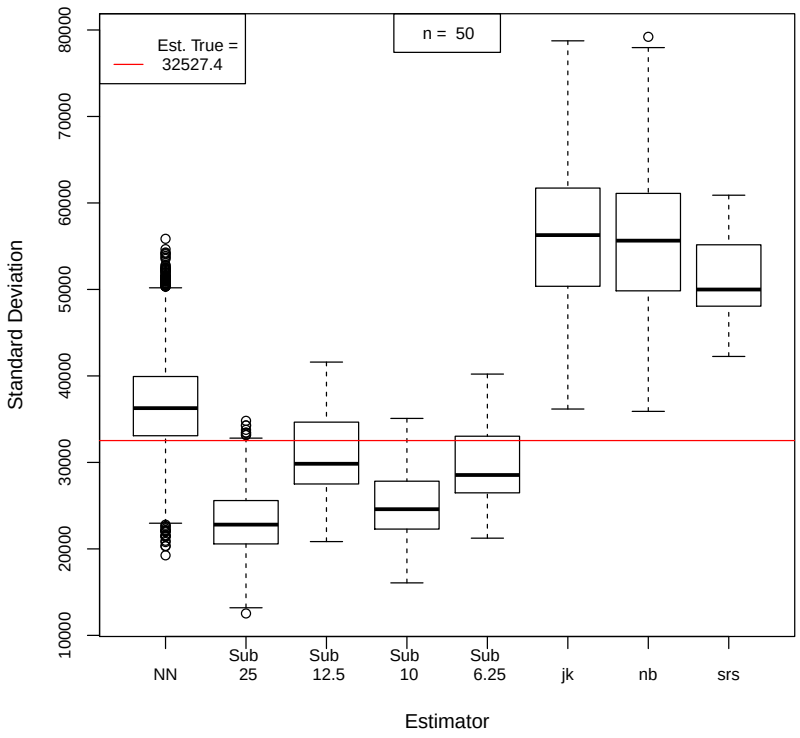

Figure 4.8: Boxplots of standard deviation estimates for all 8 estimators for the simulated income dataset for samples of size $n = 50$.

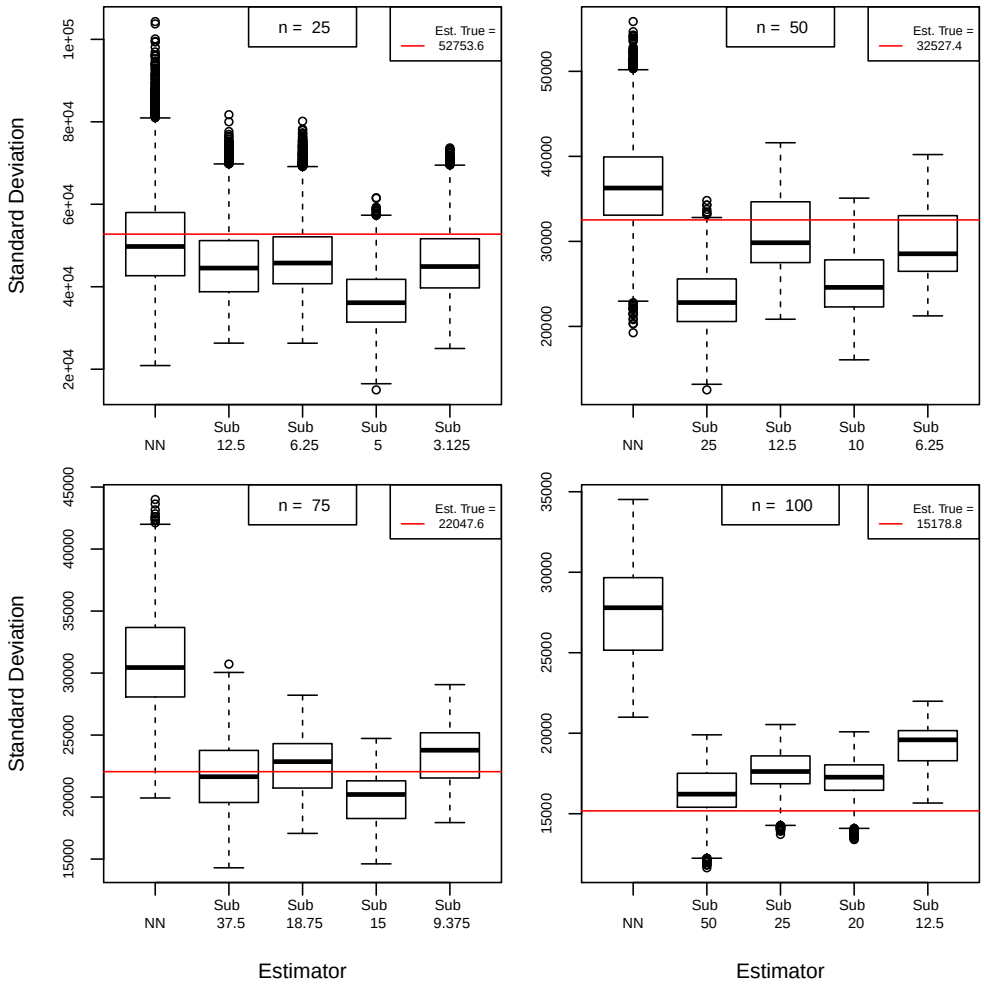

Figure 4.9: Boxplots of standard deviation estimates for the best 5 estimators for the simulated income dataset for 4 different sample sizes.

Performance Criterion	Sample Size	Nearest Neighbor	Bootstrap Subsample Size: $\frac{n}{2}$	$\frac{n}{4}$	$\frac{n}{5}$	$\frac{n}{8}$
Relative Bias Percent	$n = 25$	-0.9	-20.3	-15.9	-48.9	-20.9
	$n = 50$	29.2	-48.6	-8.9	-40.2	-16.4
	$n = 75$	100.0	-1.7	5.6	-18.2	13.9
	$n = 100$	232.4	15.7	34.7	28.3	61.6
Relative Root Mean Square Error Percent	$n = 25$	47.9	40.8	38.5	53.4	37.4
	$n = 50$	46.6	50.8	26.0	43.3	27.0
	$n = 75$	112.0	25.3	21.0	23.8	24.8
	$n = 100$	241.4	27.1	39.8	33.2	64.8
95% Confidence Interval Coverage Percent	$n = 25$	92.6	91.0	91.9	83.8	91.3
	$n = 50$	97.0	83.7	93.8	87.0	92.8
	$n = 75$	99.2	93.4	94.9	91.7	96.0
	$n = 100$	100.0	97.4	98.6	98.3	99.4

Table 4.4: Numeric results for simulated income dataset.

comparison.

For samples of size $n = 50$ and larger, the bootstrap subsample estimators have a relative bias percent closer to 0 than the nearest neighbor estimator (Table 4.4, page 103). At least one of the bootstrap subsample estimators has a smaller relative root mean square error percent than the nearest neighbor estimator for all sample sizes. The estimator with confidence interval coverage percent closest to 95% is the bootstrap subsample estimator using subsamples of size $\frac{n}{4} = 18.75$ when $n = 75$.

4.3.4 Simulated Income Dataset Conclusions

Generally, the bootstrap subsample estimators which use fractional subsample sizes perform best. The nearest neighbor estimator performs well for small

sample sizes, $n = 25$, and performs worse and worse as the sample size increases. The bootstrap subsample estimators using fractional subsample sizes perform similarly for each sample size, and the bootstrap subsample estimators using integer subsample sizes perform similarly for each sample size.

4.4 Baltimore Housing Dataset

4.4.1 Description of Dataset

The fifth and final population considered is housing prices and associated variables for Baltimore, MD from 1978. See Appendix B, page 123 for R code. These data are publicly available from GeoDa Center (Anselin, 2017). There are 211 houses in the dataset.

The auxiliary information on which we want to balance is the location of the house within the city and the age of each house. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be the the location variables where "[e]ach house is assigned coordinates by locating the address on the Maryland coordinate system. The units of this grid are thousands of feet, thus 5.28 grid units equals one mile" (Dubin, 1992, p. 441). Let $\boldsymbol{v}$ be the age of houses. Let $\boldsymbol{z}$ be the square footage of houses, the interior living space in hundreds of square feet. We compute inclusion probabilities from $\boldsymbol{z}$ using equation 1.1 (page 7). Let the response variable be $\boldsymbol{w}$, the sale price of the house in 1978 in thousands of dollars. We estimate the total for $\boldsymbol{w}$ using local pivotal method samples which balance on $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{v}$ while respecting the inclusion probabilities generated by $\boldsymbol{z}$.

We select $10,000$ samples of size $n = 25$, $n = 50$, $n = 75$, and $n = 100$. On each of these samples, we compute the estimated variance using the simple

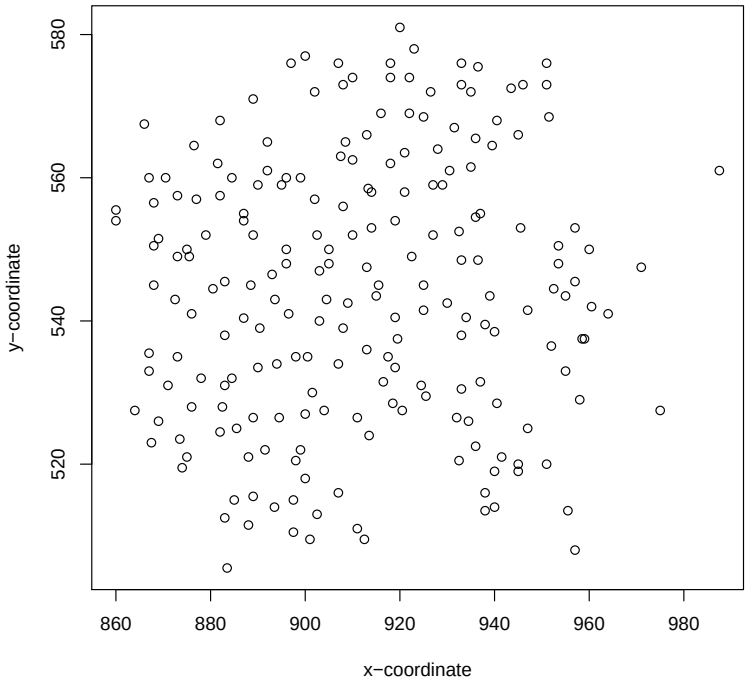

Figure 4.10: Scatterplot of 211 house locations in Baltimore, MD for 1978 housing price dataset

random sample variance estimator, the nearest neighbor variance estimator, and the jackknife estimator. For the naive bootstrap and bootstrap subsample variance estimators, we select $2,000$ bootstrap samples and $2,000$ bootstrap subsamples for each of the $10,000$ original samples for each sample size. The local variance estimator cannot be calculated on this population because the auxiliary information on which we want to balance is not two dimensional.

Differences between this population and the Matern generated populations include: the house locations in this population (Figure 4.10) are much less clustered than any of the two dimensional auxiliary information cases from the 3 Matern populations (Figure 4.1, page 86); and this population has a smaller and fixed population size ($N = 211$ versus an expected population size

of 1000).

The differences between this population and the previous four populations considered include: the auxiliary information on which we want to balance is three dimensional instead of one or two dimensional; and this population is comprised of real data whereas the previous populations are simulated. We next compare the variance estimates from this population using side–by–side boxplots.

4.4.2 Visual Distribution of Estimators

To make the relative size of the estimates easier to compare, we plot the standard deviations instead of the variances in the following plots. Each set of boxplots will display the estimated true standard deviation for the sample size as a red horizontal line. The estimated true standard deviation is the square root of the estimated true variance defined on page 82.

The simple random sample, naive bootstrap, and jackknife variance estimators all tend to overestimate the estimated true standard deviation in the same way, regardless of the original sample size, n. We display one example of this phenomenon in Figure 4.11 (page 107) for $n = 50$. It is difficult to compare the best performing estimators in Figure 4.11 because of the inclusion of the three worst performing estimators. In all subsequent plots, we will omit the simple random sample, naive bootstrap, and jackknife variance estimators.

For smaller sample sizes ($n = 25$ and $n = 50$), the nearest neighbor estimates are centered closer to the estimated true standard deviation than any of the bootstrap subsample estimates (Figure 4.12, page 108). As the sample size increases, the nearest neighbor estimator tends to overestimate the estimated

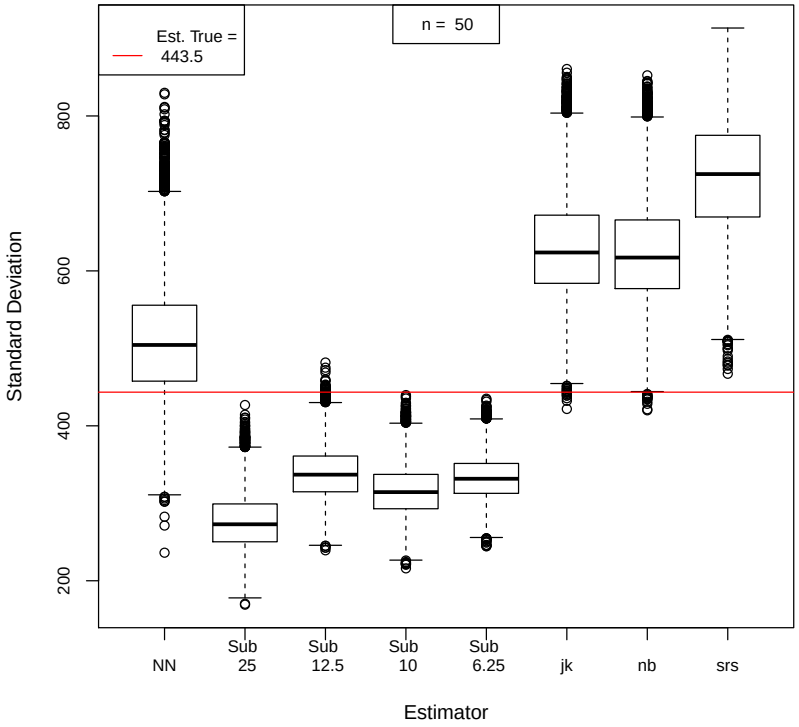

Figure 4.11: Boxplots of standard deviation estimates for all 8 estimators for the Baltimore housing dataset for samples of size $n = 50$.

true standard deviation more and more. The bootstrap subsample estimators all perform similarly regardless of the sample size.

4.4.3 Numeric Performance of Estimators

The numeric performance criteria discussed here are those defined on page 83. We consider only the 5 best estimators from the boxplot analysis for numeric comparison.

For samples of size $n = 75$ and larger, the bootstrap subsample estimators have a relative bias percent closer to 0 than the nearest neighbor estimator (Table 4.5, page 109). Nearly all of the bootstrap subsample estimators have

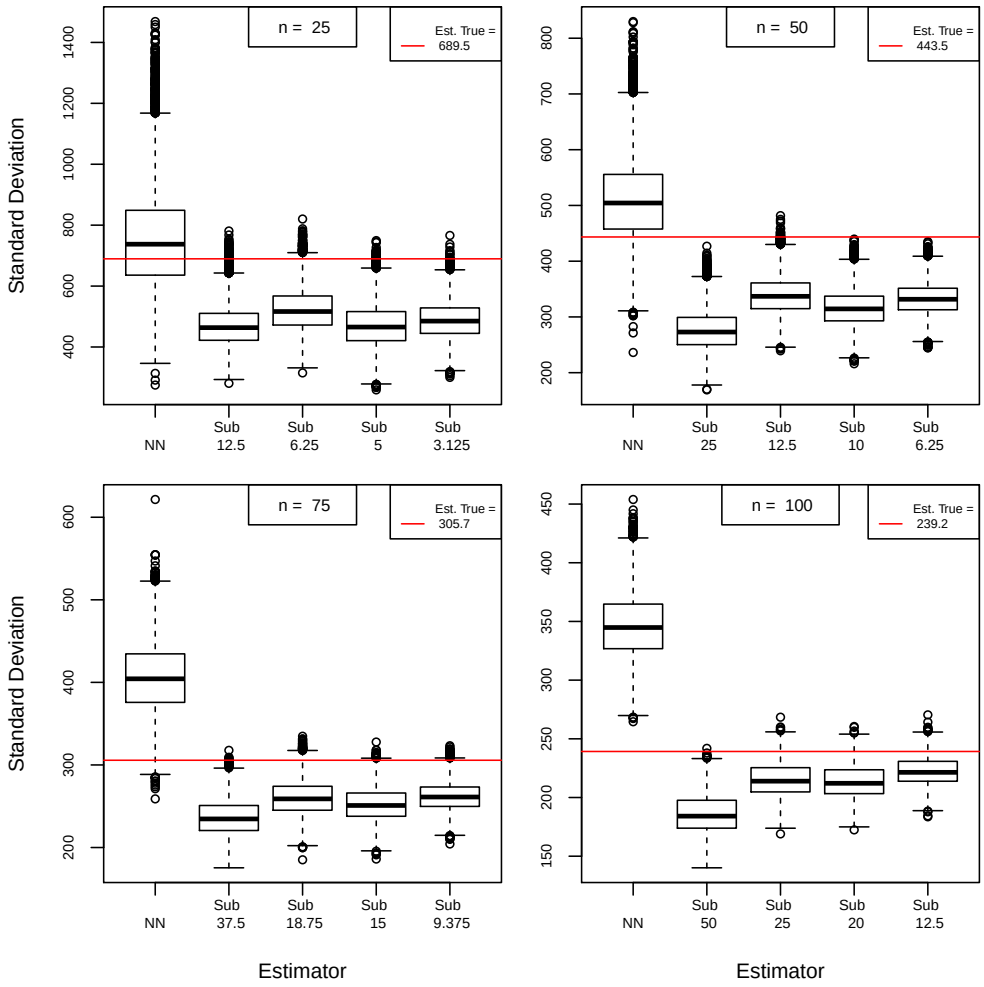

Figure 4.12: Boxplots of standard deviation estimates for the best 5 estimators for the Baltimore housing dataset for 4 different sample sizes.

a smaller relative root mean square error percent than the nearest neighbor estimator for all sample sizes. Only the bootstrap subsample estimator using subsamples of size $\frac{n}{2} = 25$ (for $n = 50$) has a larger relative root mean square error than the nearest neighbor estimator. The estimator with confidence interval coverage percent closest to 95% is the nearest neighbor estimator when $n = 25$.

Performance Criterion	Sample Size	Nearest Neighbor	Bootstrap Subsample Size: $\frac{n}{2}$	$\frac{n}{4}$	$\frac{n}{5}$	$\frac{n}{8}$
Relative Bias Percent	$n = 25$	24.1	-52.2	-41.4	-52.3	-49.0
	$n = 50$	35.0	-60.3	-40.8	-48.6	-43.3
	$n = 75$	78.8	-39.7	-27.1	-31.4	-26.3
	$n = 100$	111.0	-39.1	-18.7	-19.9	-13.3
Relative Root Mean Square Error Percent	$n = 25$	60.8	54.3	44.5	54.3	50.7
	$n = 50$	53.7	61.3	42.6	49.8	44.4
	$n = 75$	86.7	41.2	29.5	33.3	28.0
	$n = 100$	115.9	40.6	21.6	22.5	16.3
95% Confidence Interval Coverage Percent	$n = 25$	95.6	82.9	87.1	82.9	84.6
	$n = 50$	96.7	78.6	86.9	84.4	86.6
	$n = 75$	98.9	87.6	90.8	89.9	91.1
	$n = 100$	99.7	87.8	93.2	93.0	94.1

Table 4.5: Numeric results for Baltimore housing dataset.

4.4.4 Baltimore Housing Dataset Conclusions

Generally, the nearest neighbor estimator performs best for smaller sample sizes ($n = 25$ and $n = 50$). The nearest neighbor estimator performs worse and worse as the sample size increases. The bootstrap subsample estimators perform similarly for all sample sizes.

4.5 Summary Of Results

From the five populations examined we conclude that the simple random sample, naive bootstrap, and jackknife variance estimators almost always overestimate the estimated true variance. These three estimators should not be used to estimate the variance of the total of local pivotal method samples.

The local variance estimator provides the best estimate of the variance where the auxiliary information on which we want to balance is two dimensional. If the auxiliary information on which we want to balance is not two dimensional, then the nearest neighbor estimator provides the best estimate of the variance provided that the sample size is not more than about 25% of the population size. As the sample size increases, the estimates from the nearest neighbor estimator tend to get larger and larger.

The bootstrap subsample variance estimator that we proposed tends to underestimate the estimated true variance regardless of sample size. This consistent behavior indicates that a different scaling value may improve the performance of this estimator. We have not yet tried different scaling values to improve the performance.

Overall, for estimating the variance of the total for local pivotal method samples, the local variance estimator should be used, if possible. If the sample is small enough and the local variance estimator cannot be used, then the nearest neighbor estimator should be used.

Bibliography

[1] Anselin, Luc. (July 10, 2017). *Baltimore Home Sales 1970s*. UChicago Center for Spatial Data Science, https://geodacenter.github.io/data-and-lab//baltim/.

[2] Antal, E. and Y. Tillé. (2011). "A Direct Boostrap Method for Complex Sampling Designs from a Finite Population," *Journal of the American Statistical Association*, Vol. 106, No. 494, pp. 534–543.

[3] Antal, E. and Y. Tillé. (2014). "A New Resampling Method for Sampling Designs Without Replacement: the Doubled Half Bootstrap," *Computational Statistics*, Vol. 29, Issue 5, pp. 1345–1363.

[4] Baddeley, A. et. al. (November 4, 2018). "spatstat," *cran.r-project.org*. Version 1.57–1, https://cran.r-project.org/web/packages/spatstat/spatstat.pdf

[5] Barbiero, A. and F. Mecatti. (2010). "Bootstrap Algorithms for Variance Estimation in πPS sampling." *Complex Data Modeling and Computationally Intensive Statistical Methods*, edited by P. Mantovan and P. Secchi, Springer-Verlag Italia, pp. 57–69.

[6] Berger, Y.G. and C.J. Skinner. (2005). "A Jackknife Variance Estimator for Unequal Probability Sampling," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 67, No. 1, pp. 79–89.

[7] Beaumont, J–F. and Z. Patak. (2012). "On the Generalized Bootstrap for Sample Surveys with Special Attention to Poisson Sampling," *International Statistical Review*, Vol. 80, No. 1, pp. 127–148.

[8] Bertail, P. and P. Combris. (1997). "Bootstrap Généralisé D'un Sondage," *Annales D'économie et de Statistique*, Vol. 46, pp. 49–83.

[9] Bickel, P.J. and J.A. Yahav. (1988). "Richardson Extrapolation and the Bootstrap," *Journal of the American Statistical Association*, Vol. 83, No. 402, pp. 387–393.

[10] Bickel, P.J., F. Götze, and W.R. van Zwet. (1997). "Resampling fewer than n observations: Gains, Losses, and Remedies for Losses," *Statistica Sinica*, Vol. 7, pp. 1–31.

[11] Brewer, K.R.W. (1963). "A Model of Systematic Sampling with Unequal Probabilities," *Australian Journal of Statistics*, Vol. 5, pp. 5–13.

[12] Chao, M.T. (1982). "A General Purpose Unequal Probability Sampling Plan," *Biometrika*, Vol. 69, pp. 653–656.

[13] Chauvet, G. (2007). *Méthodes de Bootstrap en Population Finie*. Ph.D. thesis, Université de Rennes 2.

[14] Cordy, C. (1993). "An Extension of the Horvitz-Thompson Theorem to Point Sampling From a Continuous Universe," *Statistics and Probability Letters*, Vol. 18, pp. 353–362.

[15] D'Orazio, M. (2003). "Estimating the Variance of the Sample Mean in Two-Dimensional Systematic Sampling," *Journal of Agricultural, Biological, and Environmental Statistics*, Vol. 8, No. 3, pp. 280–295.

[16] DeGroot, M. and M. Schervish. (2012). *Probability and Statistics*. 4th ed., Pearson.

[17] Deville, J.C. and Y. Tillé. (1998). "Unequal probability sampling without replacement through a splitting method," *Biometrika*, Vol. 85, No. 1, pp. 89–101.

[18] Dubin, R.A. (1992). "Spatial Autocorrelation and Neighborhood Quality," *Regional Science and Urban Economics*, Vol. 22, pp. 433–452.

[19] Efron, B. (1979). "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, Vol. 7, No. 1, pp. 1–26.

[20] Efron, B. (1982). *The Jackknife, the Bootstrap, and Other Resampling Plans*. Society for Industrial and Applied Mathematics.

[21] Grafström, A., N.L.P. Lundström, and L. Schelin. (2012). "Spatially Balanced Sampling Through the Pivotal Method," *Biometrics*, Vol. 68, pp. 514–520.

[22] Grafström, A., L. Qualité, Y. Tillé, and A. Matei. (2012a). "Size Constrained Unequal Probability Sampling with Non–Integer Sum of Inclusion Probabilities," *Electronic Journal of Statistics*, Vol. 6, pp. 1477–1489.

[23] Grafström, A. and N.L.P. Lundström. (2013). "Why Well Spread Probability Samples Are Balanced," *Open Journal of Statistics*, Vol. 3, No. 1, pp. 36–41.

[24] Grafström, A. and A.H. Ringvall. (2013a). "Improving Forest Field Inventories by Using Remote Sensing Data in Novel Sampling Designs," *Canadian Journal of Forest Resources*, Vol. 43, pp. 1015–1022.

[25] Grafström, A. and L. Schelin. (2014). "How to Select Representative Samples," *Scandinavian Journal Of Statistics*, Vol. 41, pp. 277–290.

[26] Grafström, A., X. Zhao, M. Nylander, and H. Petersson. (2017). "A New Sampling Strategy For Forest Inventories Applied to the Temporary Clusters of the Swedish National Forest Inventory," *Canadian Journal of Forest Resources*, Vol. 47, pp. 1161–1167.

[27] Grafström, A. and J. Lisic. (August 22, 2018). "BalancedSampling," *cran.r-project.org*, Version 1.5.4, https://cran.r-project.org/web/packages/BalancedSampling/BalancedSampling.pdf

[28] Grafström A. and Alina Matei. (2018a). "Spatially Balanced Sampling of Continuous Populations," *Scandinavian Journal of Statistics*, Vol. 45, pp. 792–805.

[29] Gross, S.T. (1980). "Median Estimation in Sample Surveys," in *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pp. 181–184.

[30] Hanif, M. and K.R.W. Brewer. (1980). "Sampling with Unequal Probabilities without Replacement: A Review," *International Statistical Review*, Vol. 48, No. 3, pp. 317–335.

[31] Hansen, M.H. and W.N. Hurwitz. (1943). "On the Theory of Sampling from Finite Populations," *The Annals of Mathematical Statistics*, Vol. 14, No. 4, pp. 333–362.

[32] Holmberg, A. (1998). "A Bootstrap Approach to Probability Proportional to Size Sampling," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pp. 378–383.

[33] Horvitz, D.G. and D.J. Thompson. (1952). "A Generalization of Sampling Without Replacement From a Finite Universe," *Journal of the American Statistical Association*, Vol. 47, No. 260, pp. 663–685.

[34] Kincaid, R., A. Olsen, D. Stevens, C. Platt, D. White, and R. Remington. (June 12, 2018). "spsurvey," *cran.r-project.org*, Version 3.4, https://cran.r-project.org/web/packages/spsurvey/spsurvey.pdf

[35] Mashreghi, Z., D. Haziza, and C. Léger. (2016). "A Survey of Bootstrap Methods in Finite Population Sampling," *Statistics Surveys*, Vol. 10, pp. 1–52.

[36] Mason, D.M. and M.A. Newton. (1992). "A Rank Statistics Approach to the Consistency of a General Bootstrap," *Annals of Statistics*, Vol. 20, No. 3, pp. 1611–1624.

[37] McCarthy, P.J. and C.B. Snowden. (1985). "The Bootstrap and Finite Population Sampling," *Vital and Health Statistics*, Series 2, No. 95, pp. 1-23.

[38] Midzuno, H. (1950). "An Outline of the Theory of Sampling Systems," *Annals of the Institute of Statistical Mathematics*, Vol. 1, pp. 149–156.

[39] Nahorniak, M., D. P. Larsen, C. Volk, and C. E. Jordan. (2015). "Using Inverse Probability Bootstrap Sampling to Eliminate Sample Induced Bias in Model Based Analysis of Unequal Probability Samples," *PLoS ONE*, Vol. 10, No. 6, pp. 1–19.

[40] Narain, R.D. (1951). "On Sampling Without Replacement with Varying Probabilities," *Journal of the Indian Society of Agricultural Statistics*, Vol. 3, pp. 169–175.

[41] Neyman, J. (1934). "On Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Sampling," *Journal of the Royal Statistical Society*, Vol. 97, No. 4, pp. 558–625.

[42] Quenouille, M.H. (1949). "Approximate Tests of Correlation in Time–Series," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 11, No. 1, pp. 68–84.

[43] R Core Team (2019). "R: A Language and Environment for Statistical Computing," *R Foundation for Statistical Computing*. Vienna, Austria. https://www.R-project.org/.

[44] Rao, J.N.K, and C.F.J. Wu. (1988). "Resampling Inference with Complex Survey Data," *Journal of the American Statistical Association*, Vol. 83, No. 401, pp. 231–241.

[45] Rao, J.N.K., C.F.J. Wu, and K. Yue. (1992). "Some Recent Work on Resampling Methods for Complex Surveys," *Survey Methodology*, Vol. 18, No. 2, pp. 209–217.

[46] Räty, Minna, M. Kuronen, M. Myllymäki, A. Kangas, K. Mäkisara, and J. Heikkinen. (2020). "Comparison of the Local Pivotal Method and Systematic Sampling for National Forest Inventories," *Forest Ecosystems*, Vol. 7, No. 54, pp. 1-17.

[47] Rubin, D.B. (1981). "The Bayesian Bootstrap," *The Annals of Statistics*, Vol. 9, No. 1, pp. 130–134.

[48] Sen, A.R. (1953). "On the Estimate of the Variance in Sampling with Varying Probabilities," *Journal of the Indian Society of Agricultural Statistics*, Vol. 5, pp. 119–127.

[49] Seng, Y. P. (1951). "Historical Survey of the Development of Sampling Theories and Practice," *Journal of the Royal Statistical Society*, Vol. 114, No. 2, pp. 214–231.

[50] Sitter, R.R. (1992). "A Resampling Procedure for Complex Survey Data," *Journal of the American Statistical Association*, Vol. 87, No. 419, pp. 755–765.

[51] Stevens, D. L. and A. R. Olsen. (2003). "Variance Estimation for Spatially Balanced Samples of Environmental Resources," *Environmetrics*, Vol. 14, pp. 593–610.

[52] Stevens, D. L. and A. R. Olsen. (2004). "Spatially Balanced Sampling of Natural Resources," *Journal of the American Statistical Association*, Vol. 99, No. 465, pp. 262–278.

[53] Sunter, A. (1977). "List Sequential Sampling with Equal or Unequal Probabilities Without Replacement," *Applied Statistics*, Vol. 26, pp. 261–268.

[54] Tillé, Y. (1996). "An Elimination Procedure of Unequal Probability Sampling Without Replacement," *Biometrika*, Vol. 83, pp. 238–241.

[55] Tillé, Y. (2006). *Sampling Algorithms*. Springer.

[56] Tillé, Y. and M. Wilhelm. (2017). "Probability Sampling Designs: Principles for Choice of Design and Balancing," *Statistical Science*, Vol. 32, No. 2, pp. 176–189.

[57] Thompson, S.K. (2012). *Sampling*. 3rd ed., John Wiley & Sons.

[58] Voronoï, G. (1908). "Nouvelles Applications des Paramétres Continus á la Théorie des Formes Quadratiques. Premier Mémoire. Sur Quelques Propriétés des Formes Quadratiques Postives Parfaites," *Journal Für die Reine und Angewandte Mathematik*, Vol. 133, pp. 97–178.

[59] Wolter, K.M. (2007). *An Introduction to Variance Estimation*. 2nd ed., Springer Series in Statistics.

[60] Yates, F. and P.M. Grundy. (1953). "Selection Without Replacement from Within Strata with Probability Proportional to Size," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 15, No. 2, pp. 235–261.

Appendix A

Example 8 (page 45) R Code

```
#This is an attempt to plot visually what happens with LPM. The code
#is from the original implementation of the unoptimized LPM.

#The local pivotal method needs a set of original inclusion probabilities
#as well as an auxiliary variable matrix. Once these are input, the algorithm
#takes it from there.

################################ data input ################################
#first input the original inclusion probabilities:
prob<-c(0.2,0.5,0.8,0.4,0.1,0.3,0.7)
#Now auxiliary information:
x<-matrix(c(9.5,3.1,3.2,1.2,8.1,6.8,1,4.1,4.2,9.5,3.2,4.5,1.7,2.4),
nrow=7,ncol=2)
labelx<-c(9.5,3.1,3.2,1.2,8.1,6.8,1)
labely<-c(4.1,4.2,9.5,3.2,4.5,1.7,2.4)

n<-sum(prob)                    #target sample size

############################## initial plotting ############################
#For plot, get x and y coordinates of auxiliary variables:
xcoor<-x[,1]
ycoor<-x[,2]
```

```
#Initial 2-D plot with colors:
ce<-0.5+2*prob;ce1<-2
blue<-rep(0,length(prob))
green<-rep(0,length(prob))
red<-rep(0,length(prob))
red[prob>=0.5]<-1
green[prob<=0.5]<-2*prob[prob<=0.5]
green[prob>=0.5]<--2*prob[prob>=0.5]+2
blue[prob<=0.5]<-1
dev.new()
plot(xcoor,ycoor,xlab="1st auxiliary variable",ylab="2nd auxiliary variable",
pch=16,col=rgb(red,green,blue),cex=2*ce,cex.axis=1.5,cex.lab=1.5,
cex.main=1.5)
text(labelx[-3],labely[-3]+0.5,prob[-3],cex=1.5)
text(labelx[3],labely[3]-0.5,prob[3],cex=1.5)
points(xcoor[prob=="0.5"],ycoor[prob=="0.5"],cex=2*ce[prob=="0.5"])

par(ask=TRUE)              #stops the algorithm and asks for user input

################################ algorithm ###################################
N<-length(prob)            #determines the size of the population, N
col<-ncol(x)               #determines the number of aux. variables
index<-c(1:N)              #sets up an index set for the algorithm
p<-prob                    #sets up the vector of probabilities to be updated
r1<-runif(N,0,1)                  #for random selection of 1st element
r2<-runif(N,0,1)                  #for random selection of winner of comparison
for (i in 1:(N-1)){
        ri<-i+floor(r1[i]*(N-i))      #randomly selects element to start with
        mindi<-1e+200                 #sets a minimum distance for "nearest"
        for(j in i:N){
                if(j!=ri){            #avoids j and i being the same
                        d<-0.0        #starting distance from i to j
                        for(k in 1:col){
                                dp<-x[index[ri],k]-x[index[j],k]
                                d<-d+dp^2     #computes total Euclidean
                                              #distance i to j
                                } #end k loop
                        if(d<mindi){      #for each j, compute total distance
                                          #to i
```

```
                    rj<-j       #then select element closest to i
                    mindi<-d    #and update "nearest" sense
                    } #end if for mindi
                } #end if for j not equal ri
            } #end j loop.


    #The following plot shows what points are selected with the
    #randomization and nearest neighbor
    ce<-0.5+2*p
    blue<-rep(0,length(p))
    green<-rep(0,length(p))
    red<-rep(0,length(p))
    red[p>=0.5]<-1
    green[p<=0.5]<-2*p[p<=0.5]
    green[p>=0.5]<--2*p[p>=0.5]+2
    blue[p<=0.5]<-1
    plot(xcoor,ycoor,xlab="1st auxiliary variable",
    ylab="2nd auxiliary variable",
    pch=16,col=rgb(red,green,blue),cex=2*ce,cex.axis=1.5,cex.lab=1.5,
    cex.main=1.5)
    points(xcoor[index[ri]],ycoor[index[ri]],pch=4,
    cex=2*ce1) #random point x
    points(xcoor[index[rj]],ycoor[index[rj]],pch=3,
    cex=2*ce1) #near-neigh +
    points(xcoor[p=="0.5"],ycoor[p=="0.5"],cex=2*ce[p=="0.5"])
    points(xcoor[p=="1"],ycoor[p=="1"],cex=2,pch=5)
    points(xcoor[p=="0"],ycoor[p=="0"],cex=2,pch=0)

########## At this point we should have i's nearest neighbor ##########
########## that point is labeled rj. Now we let ri and rj compete #######
########## to see which point gets included in the sample (i.e. whose ###
########## inclusion probability is updated to 1 or 0. ##################

    a<-p[index[ri]]+p[index[rj]]  #computes sum of
    #original inclusion probs.
    if(a>1){                      #wp is winner's probability
            wp<-1                 #sets wp to max of 1 or sum
            } else wp<-a
```

```
lp<-p[index[ri]]+p[index[rj]]-wp     #computes loser's probability
if(r2[i] < (wp-p[index[rj]])/(wp-lp) ){ #selects winner
        p[index[ri]]<-wp
        p[index[rj]]<-lp
        } else {            #above and below assign the updated
        p[index[ri]]<-lp    #probabilities depending on which point
        p[index[rj]]<-wp    #wins
        } #end if and else

#The following plot shows the updated inclusion probabilities
ce<-0.5+2*p
blue<-rep(0,length(p))
green<-rep(0,length(p))
red<-rep(0,length(p))
red[p>=0.5]<-1
green[p<=0.5]<-2*p[p<=0.5]
green[p>=0.5]<--2*p[p>=0.5]+2
blue[p<=0.5]<-1
plot(xcoor,ycoor,xlab="1st auxiliary variable",
ylab="2nd auxiliary variable",
pch=16,col=rgb(red,green,blue),cex=2*ce,cex.axis=1.5,cex.lab=1.5,
cex.main=1.5)
points(xcoor[index[ri]],ycoor[index[ri]],pch=4,cex=2*ce1)
#random point is x
points(xcoor[index[rj]],ycoor[index[rj]],pch=3,cex=2*ce1)
#near-neigh is +
points(xcoor[p=="0.5"],ycoor[p=="0.5"],cex=2*ce[p=="0.5"])
points(xcoor[p=="1"],ycoor[p=="1"],cex=2,pch=5)
points(xcoor[p=="0"],ycoor[p=="0"],cex=2,pch=0)

########## At this point we have updated the inclusion probabilities #####
########## for i and j. Note that this algorithm doesn't know what to ####
########## do when we get to the end of the list (when i = N-1), so ######
########## the following makes a determination about how to assign #######
########## updated probabilities to those.

if(i==(N-1)){   #for next to last case, include or not
        if(runif(1,0,1) < p[index[ri]]){     #based on random chance
                p[index[ri]]<-1
```

```
                    } else {    #above and below decide to include or not
                    p[index[ri]]<-0 #for i = N-1
                    } #end if for ith element
                if(runif(1,0,1) < p[index[rj]]){
                    p[index[rj]]<-1
                    } else {    #above and below decide to includ or not
                    p[index[rj]]<-0 #for j = N
                    } #end if for jth element
                } #end if for i==N-1


########## Now all the inclusion probabilities have been updated (note #####
########## that we are still in the i loop, so at this point we have ######
########## only done it for a single i. The next thing to do is to ########
########## make sure that we don't repeat the indexes that have ###########
########## already had updated probabilities, so we rearrange the ##########
########## index set.

        m<-rj                   #assumes that rj will be moved
        if(p[index[ri]]==0 || p[index[ri]]==1){
            m<-ri               #decides if ri should be moved instead
            } #end if to start rearranging index values
        t<-index[i]
        index[i]<-index[m]      #These three lines exchange index elements
        index[m]<-t             #i and m
        } #end i loop
#Final plot of sample:
ce<-0.5+2*p
blue<-rep(0,length(p))
green<-rep(0,length(p))
red<-rep(0,length(p))
red[p>=0.5]<-1
green[p<=0.5]<-2*p[p<=0.5]
green[p>=0.5]<--2*p[p>=0.5]+2
blue[p<=0.5]<-1
plot(xcoor,ycoor,xlab="1st auxiliary variable",ylab="2nd auxiliary variable",
pch=16,col=rgb(red,green,blue),cex=2*ce,cex.axis=1.5,cex.lab=1.5,
cex.main=1.5)
points(xcoor[p=="1"],ycoor[p=="1"],cex=2,pch=5)
```

```
points(xcoor[p==”0”],ycoor[p==”0”],cex=2,pch=0)

########## We have finally ended the i loop. This means that we have #######
########## found the nearest neighbor to all points, computed updated #####
########## inclusion probabilities and rearranged the index set so that ####
########## no points have been compared twice. Now we can determine which ##
########## points to actually sample as follows:

n<-round(sum(p))                    #determine sample size
s<-rep(NA,n)                        #vector for which elements to select
c<-1                                #counting variable
for(i in 1:N){
        if(p[i]==1){                #if element i is included
                s[c]<-i             #put that value in s
                c<-c+1              #now increment the counter
                } #end if
        } #end for
                        #the output should be a vector which tells the
s                       #position of elements to select for the sample
par(ask=FALSE)          #returns plotting control to R
```

Appendix B

Chapter 4 R Code to Simulate Populations

```
############################## Necessary Packages ##############################
#install.packages("spatstat")
#library(spatstat)                #for Matern Cluster data generation
#install.packages("BalancedSampling")
#library(BalancedSampling)        #for lpm

#Need to run function LPM_PACK.r code first

#install.packages("sp")           #for functions related to local var
#library('sp')
#install.packages("spsurvey")     #for total.est in local var
#library('spsurvey')
#install.packages("sampling")
#library('sampling')
#install.packages("FNN")          #for get.knn
#library(FNN)

#Run one of the three following script sections to generate the populations
#used in the text.
########################  Matern Population Generation ####################
#Run this section of the script each time varying the second argument
```

```
#of rMatClust to scale = 1.0, 0.75, and 0.5. Also vary the sample size.
set.seed(1234)                    #for repeatability
a<-rMatClust(0.2,0.5,8,win=owin(c(0,25),c(0,25)))

#intensity is 0.2*8, then expand by 625 since domain is 25 x 25 (so expected
#number of points is 1000
N<-a$n                    #population size (varies due to Matern Cluster)

n<-100                    #sample size: use n=50, 100, 150, 200

aux<-data.frame(x=a$x,y=a$y)
aux$w<-((-(aux$x-2)*(aux$x-8)^2+7500)/12205)*200+
        ((-(-aux$y+10)*(-aux$y+25)+2600)/31324)*200+rnorm(N,0,5)
aux$z<-sqrt(aux$w)+rnorm(N,0,1) #variable to be estimated
#estimate w with z

#for inclusion probabilities based on z (based on inclusionprobabilities code
#from Sampling package):
prob<-n*aux$z/(sum(aux$z))


################ Simulated Financial Population Generation ################
#Run this section of the script to simulate the third population.
N<-150                            #population size
n<-50                             #sample sizes: use n=25, 50, 75, 100

set.seed(1234)                    #for repeatability
i<-rnorm(N,0,7)
x<-abs(i)
e<-rnorm(N,0,1)
y<-(12.5+3*x^(1.2)+15*e)^2 +4000          #response variable

p<-rlnorm(N,0,0.25)
z<-y^0.2*p
prob<-n*z/sum(z)

#This dataset only has 1 auxiliary variable and no spatial component.
aux<-data.frame(x=x,w=y,z=z)
```

```
#####################  Baltimore Housing Population ##########################
#1978 housing prices in Baltimore, MD, USA
d<-read.csv("C:\\Users\\math\\Desktop\\baltim.csv")
#read the data in however you have it. See example in the text for the
#location of the dataset.

N<-length(d$PRICE)
n<-50
aux<-data.frame(x=d$X, y=d$Y, v=d$AGE, z=d$SQFT, w=d$PRICE)
#estimate w balancing on x,y, and v. Build incl. prob. from z

#for inclusion probabilities based on z (based on inclusionprobabilities code
#from Sampling package):
prob<-n*aux$z/(sum(aux$z))



#####################  Simulated Sampling ###################################
#The following code is run once for each of the datasets above.
B1<-10000              #number of LPM samples to take
B<-2000                #number of bootstrap and bootstrap subsamples
                       #to do on each LPM sample

mat<-data.matrix(aux[,-c(3,4)])
col<-ncol(mat)
xstd<-matrix(data=NA,nrow=N,ncol=col)
for(i in 1:col){
        xstd[,i]<-(mat[,i]-mean(mat[,i]))/sd(mat[,i])
        }                          #This standardizes the auxiliary information
truew<-sum(aux$w)                  #true w value

htlpm<-rep(NA,B1)             #These lines set up storage for values.
srsvarlpm<-rep(NA,B1)         #The final variance estimates will be
jkvarlpm<-rep(NA,B1)          #stored in these vectors with each
localvarlpm<-rep(NA,B1)       #value representing one variance
nabootvarlpm<-rep(NA,B1)      #estimate.
Gvar<-rep(NA,B1)              #Gvar is the nearest neighbor variance
#The following lines set up storage for the Horvitz---Thompson estimates
#of the total for subsample sizes which are h=half, q=quarter, f=fifth,
```

```
# and e=eighth the size of the original sample.
htlpmh<-rep(NA,B)
htlpmq<-rep(NA,B)
htlpmf<-rep(NA,B)
htlpme<-rep(NA,B)
#The lines below store the variance estimates from B Horvitz---Thompson
#estimates of the total for the same subsample sizes as above.
subvarlpmh<-rep(NA,B1)
subvarlpmq<-rep(NA,B1)
subvarlpmf<-rep(NA,B1)
subvarlpme<-rep(NA,B1)
htlpmsamp<-rep(NA,B)
subvarlpmsamp<-rep(NA,B1)

for(i in 1:B1){
        s<-lpm1(prob,xstd)
        sam<-data.frame(w=aux$w[s],pi=prob[s])
        htlpm[i]<-sum(sam$w/sam$pi)
        srsvarlpm[i]<-(N*(N-n)/n)*var(sam$w)
        #jackknife estimator:
        htj<-rep(NA,n)
        theta<-rep(NA,n)
        for(j in 1:n){
                htj[j]<-sum(sam$w[-j]/(sam$pi[-j]*(n-1)/n))
                } #end excluded individual loop
        for(j in 1:n){
                theta[j]<-n*(sum(sam$w/sam$pi))-(n-1)*htj[j]
                } #end pseudovalue loop
        jkvarlpm[i]<-(1/(n*(n-1)))*sum((theta-mean(theta))^2)
        #for local variance estimator:
        u<-rep(0,N)
        u[s]<-1
        res = EST(aux$w,prob,xstd[,1],xstd[,2],u,vartype='Local');
        localvarlpm[i]<-res$SE^2;
        #Grafstrom variance (aka nearest neighbor variance):
        N1<-get.knn(xstd[s,],k=1)
        Gvar[i]<-0
        for (j in 1:length(s)){
                Gvar[i]<-Gvar[i]+(sam$w[j]/sam$pi[j]-
```

```
                sam$w[N1$nn.index[j]]/sam$pi[N1$nn.index[j]])^2
        } #end j loop
Gvar[i]<-(1/2)*Gvar[i]
newpop<-aux[s,]
pih<-(n/2)*newpop$z/(sum(newpop$z))
piq<-(n/4)*newpop$z/(sum(newpop$z))
pif<-(n/5)*newpop$z/(sum(newpop$z))
pie<-(n/8)*newpop$z/(sum(newpop$z))
mats<-data.matrix(newpop[,-c(3,4)])
col1<-ncol(mats)
xstds<-matrix(data=NA,nrow=length(s),ncol=col)
        for(j in 1:col1){
        xstds[,j]<-(mats[,j]-mean(mats[,j]))/sd(mats[,j])
        }
HT<-rep(NA,B)
for(j in 1:B){
        c<-sample(seq(1,n,1),n,replace=TRUE)
        HT[j]<-sum(sam$w[c]/sam$pi[c])      #for naive bootstrap
        ssamp<-lpm1(prob[s],xstds)
        htlpmsamp[j]<-sum(newpop$w[ssamp]/(prob[s])[ssamp])

        sh<-lpm1(pih,xstds)          #for boot subsample of size 1/2
        htlpmh[j]<-sum(newpop$w[sh]/(prob[s])[sh])

        sq<-lpm1(piq,xstds)          #for boot subsample of size 1/4
        htlpmq[j]<-sum(newpop$w[sq]/(prob[s])[sq])

        sf<-lpm1(pif,xstds)          #for boot subsample of size 1/5
        htlpmf[j]<-sum(newpop$w[sf]/(prob[s])[sf])

        se<-lpm1(pie,xstds)          #for boot subsample of size 1/8
        htlpme[j]<-sum(newpop$w[se]/(prob[s])[se])

        } #end j loop
nabootvarlpm[i]<-var(HT)
subvarlpmh[i]<-var(htlpmh)
subvarlpmq[i]<-var(htlpmq)
subvarlpmf[i]<-var(htlpmf)
subvarlpme[i]<-var(htlpme)
```

```
        subvarlpmsamp[i]<-var(htlpmsamp)
        } #end i loop

evar<-(1/B1)*sum((htlpm-truew)^2)        #estimated true variance
```

Appendix C

Nearest Neighbor Increasing Relative Bias with Increasing Sample Size

We want to demonstrate on a small population that the phenomenon of increasing relative bias in the nearest neighbor variance estimator with increasing sample size is not an artifact of coding. In a population of size $N = 4$, we want to select many local pivotal method samples of size $n = 2$ and size $n = 3$. Because the population is very small, the true variance of the Horvitz–Thompson estimate of the total, the true expected value of the nearest neighbor variance estimator, and the true relative bias will be computed.

The population, with auxiliary information given as x and y coordinates, with response value given as w, and with two sets of inclusion probabilities (one for selecting samples of size $n = 2$ and one for selecting samples of size $n = 3$) is given in Table C.1. We want to select samples which balance on the x and y coordinates while also respecting the given inclusion probabilities

Individ.	x coordinate	y coordinate	w value	π_i for $n = 2$	π_i for $n = 3$
1	1	2	1	0.4	0.6
2	2	1	6	0.6	0.9
3	5	4	3	0.4	0.6
4	4	5	5	0.6	0.9

Table C.1: Population of size 4 for testing nearest neighbor estimator.

(which are positively associated with the response, w).

Because the population is small, we can enumerate all possible local pivotal method samples. This allows us to exactly compute the joint inclusion probabilities for all pairs of individuals in the population, and we can use the joint inclusion probabilities to compute the true variance of the Horvitz–Thompson estimate of the total. Recall that the true variance of the Horvitz–Thompson estimate of the total is

$$\mathrm{Var}(\hat{\tau}_\pi) = \sum_{i=1}^{N} \left(\frac{1 - \pi_i}{\pi_i} \right) w_i^2 + \sum_{i=1}^{N} \sum_{j \neq i}^{N} \left(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_i \pi_j} \right) w_i w_j.$$

The true relative bias of the nearest neighbor variance estimator is

$$\mathrm{TRB} \ = \frac{\mathrm{E}(\widehat{\mathrm{Var}}_{NN}(\hat{\tau}_\pi)) - \mathrm{Var}(\hat{\tau}_\pi)}{\mathrm{Var}(\hat{\tau}_\pi)},$$

where $\mathrm{E}(\widehat{\mathrm{Var}}_{NN}(\hat{\tau}_\pi))$ is the expected value of the nearest neighbor variance estimator. We compute this expected value treating the nearest neighbor variance estimator as a discrete random variable where the probability of an estimate is the probability of one local pivotal method sample. Each distinct local pivotal method sample produces a different nearest neighbor variance estimate. We compute the probability of each local pivotal method sample by

Sample	Theor. Prob. (Exact)	Sim. Prob. (Rounded)	NN Var. Estimate (Exact)	Total Estimate (Exact)
		Samples of Size $n = 2$		
$(1, 0, 1, 0)$	0.16	0.1636	25	10
$(1, 0, 0, 1)$	0.24	0.2425	$34\frac{16}{576}$	$10.8\overline{3}$
$(0, 1, 1, 0)$	0.24	0.2345	$6\frac{1}{4}$	17.5
$(0, 1, 0, 1)$	0.36	0.3549	$2\frac{7}{9}$	$18.\overline{3}$
		Samples of Size $n = 3$		
$(1, 1, 1, 0)$	0.1	0.1040	$26\frac{7}{18}$	$13.\overline{3}$
$(1, 1, 0, 1)$	0.4	0.4005	$32\frac{91}{162}$	$13.\overline{8}$
$(1, 0, 1, 1)$	0.1	0.0950	$7\frac{47}{54}$	$12.\overline{2}$
$(0, 1, 1, 1)$	0.4	0.4005	$1\frac{113}{162}$	$17.\overline{2}$

Table C.2: All Possible Local Pivotal Method Samples of Size $n = 2$ and $n = 3$ with corresponding theoretical probability, simulated probability from $10,000$ simulations, Nearest Neighbor Variance estimate, and Total estimate

exhaustively listing all possible samples. This information is given in Table C.2.

For samples of size $n = 2$, the true variance is $\text{Var}(\hat{\tau}_\pi) = \frac{41}{3} = 13.\bar{6}$, the expected value of the nearest neighbor estimator is $\text{E}(\widehat{\text{Var}}_{NN}(\hat{\tau}_\pi)) = \frac{44}{3} = 14.\bar{6}$, so the true relative bias is $\text{TRB} = \frac{3}{41} \approx 0.073$.

For samples of size $n = 3$, the true variance is $\text{Var}(\hat{\tau}_\pi) = \frac{95}{27} = 3.\overline{518}$, the expected value of the nearest neighbor estimator is $\text{E}(\widehat{\text{Var}}_{NN}(\hat{\tau}_\pi)) = \frac{925}{54} = 17.1\overline{296}$, so the true relative bias is $\text{TRB} = \frac{147}{38} \approx 3.868$.

We can estimate the true relative bias using the relative bias

$$\text{RB} = \frac{\frac{1}{10,000} \cdot \sum_{j=1}^{10,000} \widehat{\text{Var}}_{NN,j}(\widehat{\tau}_\pi) - \widehat{\text{Var}}_{est}}{\widehat{\text{Var}}_{est}},$$

where $\widehat{\mathrm{Var}}_{NN,j}(\widehat{\tau}_\pi)$ is the nearest neighbor variance estimate from the jth sample out of $10,000$ local pivotal method samples, and $\widehat{\mathrm{Var}}_{est}$ is the variance of the Horvitz–Thompson estimates of the total from the same $10,000$ local pivotal method samples. The results of one simulation are shown in Figures C.1 and C.2, and the simulated probability of each sample is shown in the third column of Table C.2 (page 131). The relative biases are 0.076 and 3.926 for samples of size $n = 2$ and $n = 3$, respectively.

These results are consistent with the observation in the main text that the relative bias of the nearest neighbor estimator increases as the sample size increases. It is still unclear why this phenomenon occurs.

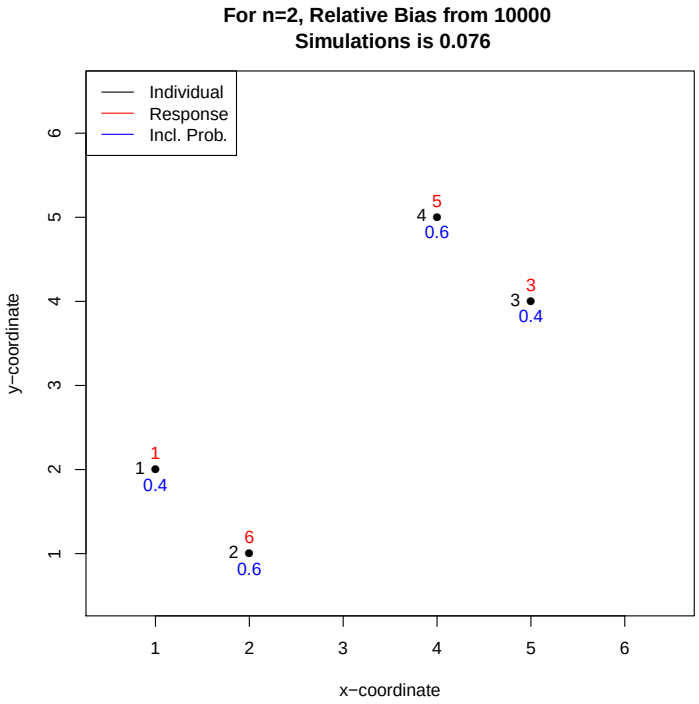

Figure C.1: Population information from population of size $N = 4$ with $n = 2$

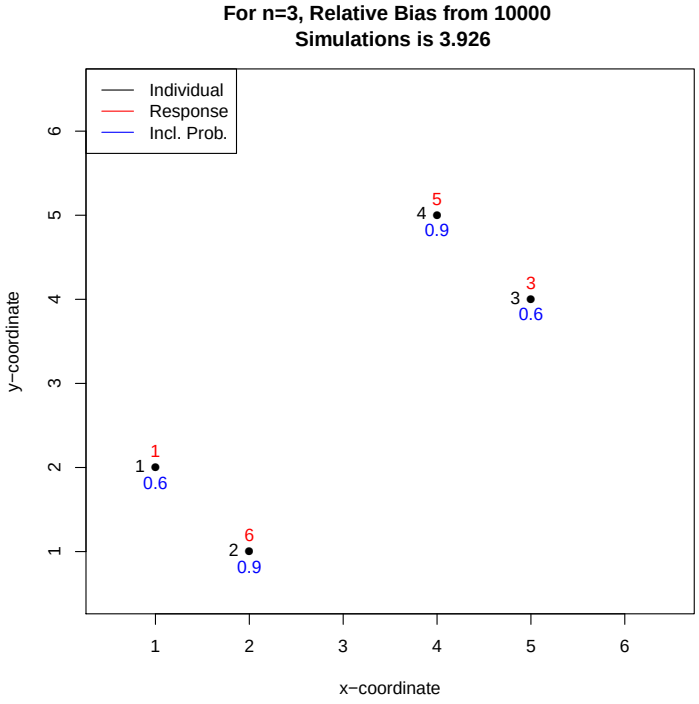

Figure C.2: Population information from population of size $N = 4$ with $n = 3$