Environmental monitoring poses two challenges to statistical analysis: complex data and complex survey designs. Monitoring for system health involves measuring physical, chemical, and biological properties that have complex relations. Exploring these relations is an integral part of understanding how systems are changing under stress. How does one explore high dimensional data? Many of our current methods rely on “black-box” mathematical methods. Visualization techniques on the other hand are either restricted to low dimensions or hopelessly out of context. The first topic explored in this dissertation suggests a direct search method for use in projection pursuit guided tours.

In Chapter 2 a direct search method for index optimization, the multidirectional pattern search, was explored for use in projection pursuit guided tours. The benefit of this method is that it does not require the projection pursuit index to be continuously differentiable; in contrast to existing methods that require differentiability. Computational comparisons with test data revealed the feasibility
and promise of the method. It successfully found hidden structure in 4 of 6 test data sets. The study demonstrates that the direct search method lends itself well to use in guided tours and allows for non-differentiable indices.

Evaluating estimators of the population variance is covered in Chapter 3. Good estimates of the population variance are useful when designing a survey. These estimates may come from a pilot project or survey. Often in environmental sampling simple random sampling is not possible; instead complex designs are used. In this case there is no clear estimator for the population variance. We propose an estimator that is (1) based on a methods of moments approach and (2) extendible to more complex variance component models. Several other estimators have been proposed in the literature. This study compared our method of moment estimator to other variance estimators. Unfortunately our estimator did not do as well as some of the other estimators that have been suggested implying that these estimators do not perform similarly as the literature suggests they do. Two estimators, the sample variance and a ratio estimator based on the Horvitz-Thompson Theorem and a consistency argument, proved to be favorable.
Tools for Environmental Statistics: Creative Visualization and Estimating Variance from Complex Surveys

by

Jean-Yves Pip Courbois

A THESIS

submitted to

Oregon State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Presented January 7, 2000
Commencement June 2000

APPROVED:

Redacted for privacy

Chair of Department of Statistics

Redacted for privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University Libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Jean-Yves Pip Courbois
Acknowledgment

I wish to thank my major professor, Dr. Scott Urquhart, for all his help, guidance, and friendship. I wish to also thank my committee, Dr. Maggie Neiss, from the Department of Math and Science Education, Dr. Cliff Pereira, Dr. Fred Ramsey, and Dr. Jeff Arthur, from Statistics. Dr. Arthur especially took time to help with the first manuscript before becoming a member of my committee. Some people not on my committee also need to be acknowledged. Dr. Tony Olsen at the Environmental Protection Agency was invaluable to my success. He provided support for four years of study without return. Moreover he supplied many interesting conversations and research ideas. Also, I would like to thank Dr. Dan Carr from George Mason University for his ideas and collaboration. Dr. Di Cook from Iowa State University for her support during the research and writing of the first manuscript. Dr. Virginia Lesser was a member of my committee from the beginning but unfortunately could not be there at the end. She spent extra time with me during initial research of the second manuscript. All the funding for this research was from the United States Environmental Protection Agency NNEMS grant number U914916-01-0. Finally I owe a big debt to my partner Hilary Geisheker who suffered through this process from the beginning to end.
Contribution of Authors

Dr. Scott Urquhart proposed the initial question that led to the second manuscript. Dr. Urquhart was also involved in the writing of the second manuscript.
# Table of Contents

1. INTRODUCTION ................................................................. 1

2. A DIRECT SEARCH METHOD FOR PROJECTION PURSUIT GUIDED TOUR OPTIMIZATION ........................................... 6
   2.1 Abstract ........................................................................ 7
   2.2 Introduction ................................................................... 7
   2.3 Projection-Pursuit Guided Tours ........................................ 9
      2.3.1 The Grand Tour ....................................................... 10
      2.3.2 Exploratory Projection Pursuit ................................. 11
      2.3.3 Projection-Pursuit Guided Tours ............................... 12
   2.4 Pattern Search Optimization ............................................ 14
   2.5 Computational Comparisons .......................................... 18
   2.6 Conclusions .................................................................. 20
   2.7 Special Acknowledgments .............................................. 23
   2.8 References .................................................................... 23

3. HORVITZ-THOMPSON BASED ESTIMATORS FOR FINITE POPULATION VARIANCE COMPONENTS. PART I: THE POPULATION VARIANCE .................................................. 25
   3.1 Abstract ........................................................................ 26
   3.2 Introduction ................................................................... 27
   3.3 Notation and Preliminary Results ..................................... 32
### Table of Contents (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 The Estimators for the Finite Population Variance</td>
<td>36</td>
</tr>
<tr>
<td>3.4.1 Introduction to Variance Estimation and Estimation Criteria</td>
<td>36</td>
</tr>
<tr>
<td>3.4.2 The Estimators</td>
<td>39</td>
</tr>
<tr>
<td>3.4.3 Ratio Forms of the Derived Estimators</td>
<td>42</td>
</tr>
<tr>
<td>3.4.4 Analytical Comparisons</td>
<td>45</td>
</tr>
<tr>
<td>3.4.5 Inclusion Probability Approximations</td>
<td>46</td>
</tr>
<tr>
<td>3.5 Methods: A Simulation Experiment For Assessing the Estimators</td>
<td>47</td>
</tr>
<tr>
<td>3.5.1 The GRS Sample Plan</td>
<td>47</td>
</tr>
<tr>
<td>3.5.2 Population Families</td>
<td>49</td>
</tr>
<tr>
<td>3.5.3 Treatment Structure</td>
<td>51</td>
</tr>
<tr>
<td>3.5.4 Real Populations</td>
<td>54</td>
</tr>
<tr>
<td>3.5.5 Experimental Design Structure</td>
<td>57</td>
</tr>
<tr>
<td>3.5.6 Response Structure</td>
<td>59</td>
</tr>
<tr>
<td>3.5.7 Simulation Algorithm</td>
<td>59</td>
</tr>
<tr>
<td>3.6 Results</td>
<td>61</td>
</tr>
<tr>
<td>3.6.1 Comparison of the Estimators with Artificial Populations</td>
<td>64</td>
</tr>
<tr>
<td>3.6.2 Real Populations and the Effect of Sample Size</td>
<td>69</td>
</tr>
<tr>
<td>3.7 Conclusions</td>
<td>72</td>
</tr>
<tr>
<td>3.8 Discussion</td>
<td>76</td>
</tr>
<tr>
<td>3.9 References</td>
<td>79</td>
</tr>
<tr>
<td>4 SUMMARY</td>
<td>82</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>85</td>
</tr>
<tr>
<td>APPENDIX: Listing of Computer Code</td>
<td>89</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Flow diagram for projection pursuit guided tours</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>For optimization in 2 dimensions the simplex has 3 sides</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>Six test data sets</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Best projections from the grand tour</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>Best projections after steepest ascent optimization</td>
<td>21</td>
</tr>
<tr>
<td>2.6</td>
<td>Best projections after pattern search optimization</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Families of population structure for the auxiliary variable, ( x )</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>Lake 1 population</td>
<td>55</td>
</tr>
<tr>
<td>3.3</td>
<td>Lake 2 population</td>
<td>57</td>
</tr>
<tr>
<td>3.4</td>
<td>English population</td>
<td>58</td>
</tr>
<tr>
<td>3.5</td>
<td>Contour plots of estimator mean square error</td>
<td>62</td>
</tr>
<tr>
<td>3.6</td>
<td>Best estimators for the uniform family</td>
<td>68</td>
</tr>
<tr>
<td>3.7</td>
<td>Best estimators' bias and mean square error for the English population</td>
<td>71</td>
</tr>
<tr>
<td>3.8</td>
<td>Mean square error difference between naive estimator and the HT-R estimator</td>
<td>75</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>43</td>
</tr>
</tbody>
</table>

3×3+1 structure, symbols, and abbreviations for the 10 estimators

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>44</td>
</tr>
</tbody>
</table>

Descriptive properties of the estimators

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>52</td>
</tr>
</tbody>
</table>

Description of families, treatments, and treatment levels in the simulation experiments

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>66</td>
</tr>
</tbody>
</table>

Minimum, average, and maximum bias and mean square error for simulations with the gamma family

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>70</td>
</tr>
</tbody>
</table>

Bias and mean square error for estimators from the real population simulations

Chapter 1

Introduction

Monitoring environmental populations presents two challenges for statistical analysis: complex data and complex designs. This thesis presents practical solutions for these complexities: (1) a method for visualizing non-linear and high-dimensional data (complex data); and (2) a method for estimating the population variance from a sample selected with a complex design. The original motivation to work on these topics stems from unanswered questions concerning the analysis and design of large-scale environmental monitoring surveys, components of ongoing research at the United States Environmental Protection Agency’s (US-EPA) Environmental Monitoring and Assessment Program (EMAP) (Messer, Linthurst, and Overton, 1991). Large-scale environmental monitoring studies supply a special breed of statistical questions not completely addressed by the statistical community. The two in particular addressed in this thesis are the almost sure multivariate and non-linear interdependence of environmental measurements, complex data, and the complex designs induced by inherent frame errors and the objectives of environmental monitoring surveys. We supply practical tools that begin to address these questions. Neither tool is a complete solution as is,
but join the collection of literature on these subjects. Each topic in this thesis is worthwhile studying by itself and so that they might be applied separately I present each in stand-alone chapters.

The first topic investigates the complex nature of environmental data. Complexities result from environmental data being multivariate and interdependent. Monitoring environmental condition or health involves measuring physical, chemical, and biological processes (Hunsaker, Carpenter, and Messer, 1990) but also “pertinent social, economic, and institutional processes” (Hunsaker et. al. 1990). To monitor these processes requires measuring many different variables. In natural settings these processes are interrelated (not independent) and relationships are non-linear. This is complicated by the fact that we do not always know what to measure. What we want to know is often easy to define, such as lake health, but difficult to measure. How do we measure and quantify lake health? Often what we know how to measure in the field is a proxy or an index for what we wish to know (Messer et. al. 1991 and Hunsaker et. al. 1990). Producing and evaluating indexes is implicitly a multivariate problem. In other cases scientists may collect environmental monitoring data for the purpose of data exploration; that is, to elicit hypotheses for further exploration. To answer questions such as “What is the full impact of a perturbation?” we are interested in data exploration of new variables and their relationships. For these reasons environmental monitoring data is complex.
Chapter 2 proposes a tool for exploring multivariate environmental data. The method adds to the literature on projection-pursuit guided tours, a high-dimensional visualization technique, and as such does not stand-alone. Projection-pursuit guided tours are methods for exploring the non-linear structure of high-dimensional data (complex data). These methods have few inferential features; rather they are data exploration techniques recently known as data mining techniques. In chapter 2, I propose a method of optimizing projection-pursuit indexes that does not require the existence of derivatives, in contrast to existing methods. This research furthers the knowledge in the field by allowing for data exploration by broadening the class of usable indexes. These non-differentiable indexes have been previously unavailable for projection-pursuit guided tours.

The second challenge environmental statistics faces and addressed in this thesis is the use of complex designs. The designs that interest us result from incomplete or erroneous sampling frames and complex survey objectives. Quite often a frame does not exist. This is the case in many of the large-scale surveys conducted by the US-EPA. For example, to select a sample of lakes for a study to monitor lakes across the entire United States requires a list of the population of lakes. A complete list of lakes does not exist for the United States. A general workaround EMAP uses is to associate every point on a map with a lake and select a sample of map points (Larsen and Christie, 1991). This solution inevitably leads to unequal inclusion probabilities for lakes and thus a complex design. Survey objectives or political matters may also influence the design. Continuing with the
same example, it may be of interest to be certain the sample includes representatives from a specific sub-population of lakes, such as large lakes. This situation might arise because large lakes are where much of our nation’s water resources are tied up; that is, there may be more water in one large lake than there is in many small lakes. Or this situation might arise because we want to be able to make specific inference to the sub-population with known precision. But selecting a simple random sample of lakes may leave large lakes not included in the sample due to the distribution of lake size being skewed toward small lakes. Solutions are to stratify, which is impossible without a complete frame, or to give large lakes a higher inclusion probability, a solution that leads to complex designs. For these reasons complex designs may be unavoidable in large scale monitoring programs.

Chapter 3 of this thesis deals with the practical problem of population variance estimation under complex designs. Several estimators have been suggested in the literature but up until now it has been unknown which of these should be used under what population and design conditions. The study in Chapter 3 is an empirical study (also known as a Monte-Carlo or simulation study) of variance estimators designed to supply practical advice for researchers estimating a variance with a complex design.

Chapter 3 takes the design-based viewpoint of statistical inference. Analysis of environmental surveys in today’s environmentally intense time requires care and non-partisan methods. Model-based methods are conditional on assumed models and hence can be contested relative to the adequacy of the model. Most
often a correct model cannot be known. Design-based methods, on the other hand, assume nothing about the population and hence are robust to such contention.

Because politically, environmental monitoring will always be a contested topic as our natural desire to grow and prosper competes with a limited availability of natural resources, design-based inferences are desirable.

Environmental statistics poses new problems for scientists. Many of the problems stem from the complex nature of environmental data and the reliance on complex sample designs. This thesis supplies two tools that help address these complexities. Chapter 2 discusses a new method for optimizing projection pursuit indexes, Chapter 3 introduces methods for estimating the variance from a complex design, and Chapter 4 contains our conclusions.
Chapter 2

A DIRECT-SEARCH METHOD FOR PROJECTION PURSUIT GUIDED TOUR OPTIMIZATION.

Jean-Yves Pip Courbois; Oregon State University
Department of Statistics, Kidder 44, Corvallis, OR 97331-4606

American Statistical Association 1999 Proceedings
of the Section on Statistical Graphics.
American Statistical Association,
1429 Duke Street, Alexandria, VA. 22314
August 1999, 18 pages, in press.
2.1 Abstract

A direct-search method for index optimization, the multidirectional pattern search, was explored for use in projection pursuit guided tours. The benefit of this method is that it does not require the projection pursuit index to be continuously differentiable. Computational comparisons with test data revealed the feasibility and promise of the method. It successfully found hidden structure in 4 of 6 test data sets. The direct-search method lends itself well to use in guided tours and allows for non-differentiable indices.

key words: direct-search optimization, multivariate data analysis, projection pursuit, projection pursuit guided tours, visualization.

2.2 Introduction

Projection-pursuit guided tour is an exploratory data analysis technique for finding low dimensional structure in high dimensional data. This technique, described in Cook et. al. (1995), combines the data visualization of grand tours (Asimov, 1985) with the non-linear structure discovery mechanisms of exploratory projection pursuit (Friedman and Tukey, 1974). The latter two methods help users discover nonlinear structure in high dimensional data by exploring the low dimensional projections of the data. Guided tours smoothly interpolate through data projections in real time while the user watches, eventually visiting all low dimensional data projections; implementations include Prime9 (Asimov, 1985),
Xgobi (Swayne, Cook, and Buja, 1998) and XPloreN (Carr, 1996). Exploratory projection pursuit, on the other hand, involves optimizing an objective function that measures the “interestingness” of data projections. The user only sees the local maxima. Implementations are described in a large collection of papers including Friedman and Tukey (1974), Nason (1995), Posse (1995), and Cook et. al. (1995).

Projection-pursuit guided tours combine these methods, using the real time rotations from grand tours and the optimization from projection pursuit. As of this writing the only known implementation of projection-pursuit guided tours is Xgobi (Swayne et. al., 1998). This paper examines optimization in projection-pursuit guided tours. Attention is restricted to projections into two dimensions but generalization is straightforward. The purpose of this paper is to suggest an alternative optimization method to steepest ascent that is used in the Xgobi system. This new optimization method is suitable for exploratory projection pursuit as well as guided tours. The advantage this technique has over the steepest ascent is the ability to handle non-differentiable indices.

The next section describes the grand tour and exploratory projection pursuit in detail and shows how these two combine to form projection-pursuit guided tours. Section 3 describes the multidirectional search algorithm. This method belongs to a class of methods known generally as direct-search methods that replace the use of the objective-function derivative with extra objective function evaluations. Section 4 describes some preliminary testing to demonstrate the feasibility and promise of the multidirectional search algorithm for use in projection pursuit guided tours.
2.3 Projection-pursuit guided tours

Consider exploratory analysis of a high dimensional data set $Z$ that consists of $n$ independent and identically distributed observations on $p$ variables. As an initial search for structure, a scientist may explore the first two moments of the data with summary statistics and principle component analysis. These methods reveal linear structure. After this initial look at the linear structure, the scientist may want to explore nonlinear structure. It makes sense at this time to restrict the nonlinear search to the residuals from the linear analysis which is what the transformation sphering does. Sphering is a transformation that results in the data being of the form $\bar{Z} = 0_p$ and $ZZ'/n = I_p$. For a good debate of the merits and dangers of sphering see the discussions following Jones and Sibson's paper on projection pursuit (Jones and Sibson, 1987). Assume the data set $Z$ is sphered for the remainder of this paper.

Projecting the $p$ dimensional data into a 2 dimensional subspace and looking at the projected data may reveal interesting nonlinear structure. The drawback in this strategy is that only dimension 2 or lower structure will be revealed (see Furnas and Buja, 1994 and Cook et. al. 1995, p. 167) for discussion. Of course, unless the scientist is extremely lucky he or she will not discover interesting structure by viewing any one projection. To know the data may require looking at all 2-dimensional projections, potentially a huge task. Projection-pursuit guided tours help make this task manageable. To understand projection-
pursuit guided tours, we must first understand exploratory projection pursuit and the grand tour.

This paper uses the following notation. Define $n$ and $p$ as above by the number of observations and the dimension of each observation. The set of orthonormal matrices $\Gamma$, where $\Gamma = [\alpha, \beta]$, $\alpha, \beta \in \mathbb{R}^p$ such that $\alpha'\alpha = \beta'\beta = 1$ and $\alpha'\beta = 0$, parameterizes the set of all 2-dimensional projections. Each observation orthogonally projected onto the plane $\Gamma$ is $X = \Gamma Z$ where $X$ lies in $\mathbb{R}^2$.

2.3.1 The grand tour

The grand tour, first described by Asimov (1985), is an interactive method of observing sequences of data projections. Originally, Asimov's idea was to visit all $d$-dimensional projections (with $d \leq p$) of the data by constructing and following a space filling curve through the space of all $d$ dimensional projections (Asimov, 1985). Building the curve proved intractable, so the algorithm compromises by randomly selecting planes from the set of all planes and then interpolating between these selected target planes. The algorithm interpolates between target planes by discretely following a smooth curve. The discrete steps are close enough and the algorithm fast enough so that the viewer sees a continuous real time "movie". For a detailed discussion of this interpolation see Buja et. al. (1998). Watching this type of grand tour for an extended period of time approximates visiting all projections (Cook et. al., 1995). Unfortunately, such a
period of time tends to be extremely long for large $p$ and there is no guarantee that any particularly interesting projection will be visited. Different implementations of the grand tour depend on different values for $d$; implementations exist where $d = 1, 2, 3, \text{and } p$. The current most sophisticated implementation, XGobi, includes manual data rotation (Cook and Buja, 1997), the capacity to handle missing data (Swayne and Buja, 1997), links to popular GIS software (Cook, et. al., 1996), and projections into 1, 2, and 3 dimensions (Cook et. al., 1998).

2.3.2 Exploratory projection pursuit

Exploratory projection pursuit, a static multivariate data exploration technique, is also based on projections of the original data. Projection pursuit, instead of trying to examine all the possible projections, uses an index that measures the “interestingness” of each projection. The index is maximized over the set of all projections. Although there may be a global maximum, non-linear structure may be present resulting in many local maxima, so exploratory projection pursuit usually results in a sequence of structure revealing projections. Interestingness can be defined in a variety of ways depending on the non-linear structure of interest. The most commonly used definition actually defines what is not interesting: normally distributed data. These indices look for non-normal projections. The Chi-squared ($\chi^2$) and Natural Hermite indices used in this paper are this type. The problem with projection pursuit is that locally maximizing
projections are mainly viewed out of context and interpretation can be a problem. Also it has been shown that the success of projection pursuit (or ability to find known structure in a data set) relies heavily on its optimization algorithm (Posse, 1995).

2.3.3 Projection-pursuit guided tours

Projection-pursuit guided tours overcome the weaknesses in each of the two methods above by combining their strengths into one algorithm. It uses the grand tour algorithm to rotate through data projections and the projection pursuit optimization criteria to guide the rotations toward interesting structure (FIGURE 2.1). The dynamic nature of the grand tour resolves the problems of context in projection pursuit while optimization from projection pursuit resolves the curse of dimensionality that plagues the grand tour.

An optimization algorithm for the guided tour needs to be quickly evaluated, have an adjustable step size, and be local. It needs to return a target plane quickly because projection pursuit guided tour is a real time interactive method. It needs to be iterative and follow a path in order to identify target planes that are not too distant from each other for interpolation on the way to a maxima. Finally it needs to be local so that the method does not optimize to a target plane far from what the user has seen that sparked the invocation of optimization. These three criteria eliminate some of the successful global search methods used in exploratory projection pursuit such as the random search method of Posse (1990)
Input data Z

Sphere Z such that $E(Z) = 0$ and $COV(Z) = I$

Define an initial projection, either 1. Two data coordinates, 2. A random projection, 3. first two principal coordinates, or 4. the results of a coarse search.

Grand tour: Rotate between randomly chosen target planes.

Interaction: Rotate by hand, include/exclude variables. If a projection suggests interesting structure, engage projection pursuit optimization.

Optimize

Optimization selects new projection plane (target plane) a pre-specified distance from the current projection plane.

Rotate between target planes found by steps in the optimization.

Is stopping criteria met? = local optimum.

Interpretation: Hand rotate and whip spin for better interpretability or back transform to original coordinates. Isolate clusters, eliminate outliers, Gaussianize current projection

Look for more structure

Continue grand tour

FIGURE 2.1
Flow diagram for projection-pursuit guided tours.


The Xgobi system uses steepest ascent for optimization. In iteration $k$ of steepest ascent the new projection plane is described by the matrix $\Gamma^{k+1}$ that is the
orthonormalization of

\[ \Gamma^* + \delta \frac{\partial I(X)}{\partial X} \frac{\partial X}{\partial \Gamma} \]

where \( I(X) \) is the projection-pursuit index and \( \delta \) a step size. Steepest ascent requires that the derivative of the index function \( \frac{\partial I(X)}{\partial X} \) exists.

2.4 Pattern search optimization

The requirement that the index function be continuously differentiable limits the type of indices we can use in guided tours. For this reason this paper suggests using the multidirectional search algorithm (Torczon, 1991) which does not require derivative information. Methods that replace derivative information with extra function evaluations are known as direct-search methods. The multidirectional search algorithm is an improvement of the popular Nelder-Mead Simplex method (Nelder and Mead, 1960).

The multidirectional search algorithm is easily illustrated when optimizing in two dimensions (FIGURE 2.2). The following text describes the algorithm for projection pursuit guided tours.
For optimization in 2 dimensions the simplex has 3 sides. Each point in the figure represents an index evaluation for an iteration; the size of the point is proportional to its index value. The figure displays arrangements of index values on the original simplex, the "line searches", and the resulting simplex for the next iteration.
The initialization step constructs a regular simplex in the parameter space.

A regular simplex is a polyhedron containing \( q + 1 \) equidistant points in \( q \) dimensions. For projection pursuit \( q = 2p \). All that is needed is a starting matrix \( \Gamma^0 \) and then the elements in the 2\( p + 1 \) sided regular simplex \( S^0 = \{ \Gamma^0_0, \Gamma^0_1, \ldots, \Gamma^0_{2p} \} \) are

\[
\gamma^0_i(j) = \begin{cases} 
\gamma^0_0(j) + \delta_1 & j = i \\
\gamma^0_1(j) + \delta_2 & j \neq i 
\end{cases}
\]

where \( \gamma^0_i(j) \) is the \( j^{th} \) element of the \( i^{th} \) vertex, on the simplex of the 0\( ^{th} \) iteration (initialization),

\[
\delta_1 = \left[ \frac{\sqrt{p+1} + p - 1}{p \sqrt{2}} \right] \delta, \\
\delta_2 = \left[ \frac{\sqrt{p+1} - 1}{p \sqrt{2}} \right] \delta,
\]

and \( \delta \) is the simplex size parameter.

After construction of \( S^0 \), the algorithm is an iterative algorithm consisting of exploratory moves around the current iterate \( S^j \) in the form of a series of discrete and bounded line searches (FIGURE 2.2). The line searches result in a new simplex. At iteration \( k \) the algorithm finds the vertex of the simplex with the best index value \( \Gamma^k_{(0)} \) and reflects the entire simplex through it. The new simplex is \( \Gamma^k_0 = \Gamma^k_{(0)} \) and

\[
\Gamma^{k+1}_j = \Gamma^{k}_{(0)} + \theta \left( \Gamma^{k}_{(0)} - \Gamma^{k}_{(j)} \right)
\]
for \( j = 1, \ldots, 2p \) where \( \theta \) describes the line search exploratory moves of the algorithm. When \( \theta = 1 \) the reflected simplex is the same size as the original simplex but rotated around the point \( \Gamma^k_{(0)} \), this is the rotation step. The other possible moves are expansion, \( \theta > 1 \), and contraction, \( 0 < \theta < 1 \) (FIGURE 2.2). Torczon (1991) recommends that \( \theta \) take the values 2 for expansion and 0.5 for contraction. Notice that the new simplex has changed orientation and has possibly changed size (FIGURE 2.2). Also note that each iteration consists of \( 2 \times 2p \) evaluations of the index. The algorithm’s simplex can be thought of as an inchworm inching through the variable space looking for the highest point.

In projection-pursuit guided tours, each new vertex of a simplex is orthonormalized in order to project the data and calculate the index value. The effect of this post-orthonormalization on the algorithm is unknown. Many of the benefits of the multidirectional search algorithm result from its uniform stepping through the parameter space (Torczon 1991). As constructed, this is true for \( \mathbb{R}^{2p} \) but probably not for the space of all pairs of orthogonal unit length vectors in \( \mathbb{R}^p \).

Optimization is stopped when the standard deviation of the function values at the simplex vertices is smaller than a specified stopping criteria. For a guided tour, we want to visualize not only the final local maximizing projected data but also the optimization process in real time. This means we have to define a step size between which we interpolate. There is an equivalence between one iteration of a steepest ascent maximization and \( p \) steps in a \( p \) dimensional pattern search.
FIGURE 2.3
Six test data sets. These structures are hidden in 8 dimensions: A. four clusters, B. donut, C. L-shape, D. cross, E. two groups, and F. spiral. The small plus sign in each plot marks the origin.

maximization. The equivalence is based on the finite difference derivative approximation. Xgobi interpolates between projections every $\delta$ steps in the steepest ascent so it seems reasonable to interpolate between our best iterates every $\delta \times p$ iterations.

2.5 Computational comparisons

This section demonstrates the feasibility and promise of the multidirectional search algorithm for projection-pursuit guided tours. The objective function in projection pursuit depends on the multivariate data set, so analytic results about the strengths and weaknesses of an optimization algorithm are unavailable. We rely on
FIGURE 2.4
Best projections from the grand tour: A. four clusters, B. donut, C. L-shape, D. cross, E. two groups, and F. spiral. The small plus sign in each plot marks the origin.

empirical evidence using test data to make our case.

To determine the promise of the multidirectional search, following Posse (1995), I have created six data sets with hidden nonlinear structure. Each data set has 400 observations in 8 dimensions. The structure is rotated into the 8 variables. FIGURE 2.3 shows the hidden structure. The objective is to find this structure with a projection-pursuit guided tour. Posse demonstrated the power of projection pursuit on these data. I use the grand tour, rotating though the projections, until a promising view is found. I then record the projection for this promising view and in turn engage each optimization, steepest ascent first and then the multidirectional pattern search. Steepest ascent was implemented using Xgobi and the Natural
Hermite index (Cook et. al. 1993). The multidirectional search was implemented in the MatLab© language along with the $\chi^2$ index (Posse, 1995). The comparisons differ in both index and optimization.

The grand tour alone was only marginally successful in identifying the hidden structure in the four clusters and the donut data sets (FIGURE 2.4). Most projections appeared bivariate normal. Eight dimensions is large. As a result, except for the four clusters and donut data, optimization had to be turned on haphazardly. Optimization, though, revealed the structure in most of the data sets (FIGURES 2.5 and 2.6). The multidirectional search algorithm out performed the steepest ascent by finding the structure in the first four data sets. Steepest ascent identified the first two and the structure in the two-groups data. Also the steepest ascent optimizing plane in the L-shape data set was close enough to the structure that it could be easily hand rotated into view.

The multidirectional search algorithm proved to be slow. Each iteration required up to 32 (4 times the dimension of the data) index evaluations. Even with this many evaluations, finding an optimum projection took under 2 minutes in each case.

2.6 Conclusions

The guided tour offers two benefits over the grand tour and projection pursuit analyses. First, projections are not visualized out of context, and second,
FIGURE 2.5
Best projections after steepest ascent optimization: A. four clusters, B. donut, C. L-shape, D. cross, E. two groups, and F. spiral.

FIGURE 2.6
Best projections after pattern search optimization: A. four clusters, B. donut, C. L-shape, D. cross, E. two groups, F. spiral.
optimization allows for discovery of concealed structure. The multidirectional
search proved to be a successful alternative to steepest ascent optimization. It also
imports the benefit of being able to optimize non-differentiable indices such as the
$\chi^2$ index. Although the multidirectional search algorithm found more of the
structure, the steepest ascent algorithm found one of the more difficult structures,
the two groups. These two methods should not be considered in competition but
rather as two allied tools. In that light with these tools we found the structure of all
the data except the spiral.

Steepest ascent optimized much faster than the multidirectional search.
Speed may be important when the data sets are large. An alternative direct-search
algorithm is the Nelder-Mead optimization algorithm. The Nelder-Mead algorithm
does not share the convergence criteria of the multidirectional search algorithm but
requires only a few index evaluations per iteration.

Neither strategy performed as well as a projection pursuit analysis has
performed in this context (Posse, 1995). The multidirectional search algorithm is
also suitable for use in exploratory projection pursuit by setting the simplex size
and step size parameters large enough to start with a global search. Further testing
needs to be done in this area.

These data sets were artificial in a number of ways. Most importantly, they
each contained non-linear structure only in one direction. Real data would be
expected to have many different interacting structures. Real data then is expected
to make the grand tour more valuable, but the optimization more difficult.
Planned further research includes exploring optimizing on a sphere instead of using the post orthogonalization, examination of the Nelder-Mead algorithm, and further exploration of the step size and simplex size parameters.

2.7 Special Acknowledgments

Support for this research from the United States Environmental Protection Agency NNEMS grant number U914916-01-0 is gratefully acknowledged. The author would also like to thank Tony Olsen, Scott Urquhart, Di Cook, and Jeff Arthur for their contributions and guidance.

2.8 References


Chapter 3

HORVITZ-THOMPSON BASED ESTIMATORS FOR FINITE POPULATION VARIANCE COMPONENTS.
PART I: THE POPULATION VARIANCE.

Jean-Yves Pip Courbois and N. Scott Urquhart

To be submitted to Communications in Statistics – Simulation and Computation.
Marcel Dekker, Inc., NY.
3.1 Abstract

The population variance is a useful parameter for designing a survey or in its own right. Under simple random sampling the sample variance is unbiased for the population variance and it is considered the estimator of choice. However, constraints and objectives in environmental surveys often result in complex survey designs -- designs that have unequal element inclusion probabilities. How do we estimate a population variance when sampling with a complex design? This question was explored from a model-assisted design-based perspective. A method-of-moments (MOM) estimator for the population variance was derived using an analysis-of-variance approach. Several alternative estimators from the survey literature were also considered. Because the population variance is like an average, ratio estimators and generalized-ratio estimators also were derived. A Monte-Carlo experiment with a factorial structure was used to compare the 10 resulting estimators. The experiment used a population space approach. Comparisons were made with respect to the estimators' simulated mean square error (MSE). The results suggest that the design correlation, the correlation between the inclusion probabilities and the response, was the most important factor in differentiating the estimators. Under conditions of low design correlation (approximately <$0.4) the sample variance outperformed the other estimators. Elsewhere in the population space, two ratio estimators, one based on consistency and the Horvitz-Thompson (HT) theorem and the other based on a corollary to the HT theorem, behaved similarly and outperformed the other estimators. Unfortunately, our MOM
estimator performed poorly. When design correlation is low the results of this study demonstrate that we can ignore the design and estimate the variance with the variance of the sample. Under conditions of high design correlation, on the other hand, the design should not be ignored and HT-based ratio estimators prove best. Although intuition supports these findings, the resilience of the naïve estimator and the observation that not all the HT-based estimators perform equally well are surprises.

3.2 Introduction

Objectives of a sample survey sometimes include estimating a finite population variance. For instance Thompson (1992, Chapter 4) describes a method for sample size determination which ultimately requires an estimate of the population variance. The variance estimate ordinarily comes from a preliminary survey. A survey that has this character is the US Environmental Protection Agency’s Environmental Monitoring and Assessment Program-Surface Waters Northeast Lakes Pilot study (EMAP lakes) (Larsen and Christie, 1993). A key objective of this pilot study is to gain population insight toward planning future sampling strategies. Toward this goal, in addition to others, the pilot project aims to estimate the relative magnitudes of three variance components including a year component, a lake component, and a measurement component (Larsen, et. al., 1995a, and Urquhart, Paulsen, and Larsen, 1998). US-EPA researchers have several reasons for estimating these variance components. By comparing the
variances of the different indicators of lake health (lake trophic condition) researchers can determine which will be most sensitive for trend detection in a national survey anticipated in the future (Larsen, Urquhart, and Kugler. 1995b).

Relative magnitudes of the three components of variance suggest where sampling effort should be concentrated (Larsen, et. al., 1991). For example, if the year variance component is large relative to the lake variance component, designs should concentrate on sampling a larger number of years and a fewer number of lakes given fixed resources. In other situations the population variance may be a parameter of interest in its own right. For example, in ecological applications the population variance may be an indicator of population health; a population with a low variance being in some sense healthier than one with a high variance. Thus sound estimators of the population variance will prove useful, as will estimators of its components.

Thompson’s (1992) example uses the sample variance $s^2$ as an estimator of the population variance. His preliminary survey uses simple random sampling. The sample variance is regarded as the estimator of choice in this case (Cochran, 1977, and Särndal et. al., 1992). But the EMAP lakes pilot project mentioned above uses a complex sample design to select a sample of lakes (for details about the design see Larsen, Thorton et. al., 1995). A complex sample design is a design that does not assign equal inclusion probabilities to population elements, as does simple random sampling.
Sampling with unequal inclusion probabilities is common in multipurpose environmental surveys. The US Environmental Protection Agency's National Stream Survey – NSS (Stehman and Overton, 1994a), the multipurpose monitoring plan for the California Bight (Stevens, 1997), and the EMAP lakes pilot project all use unequal inclusion probabilities. The complex designs for all of these surveys arise due to survey objectives and practical constraints. With unequal inclusion probabilities the sample variance might produce misleading estimates, because it is not design unbiased. How should we estimate a finite population variance when sampling with a complex design?

Särndal, Swensson, and Wretman (1992, section 5.9) briefly discuss the problem of estimating the population variance under complex sample designs. They suggest three estimators but do little to compare these estimators' relative merits or properties; they state simply that the estimators should not behave differently. But the first of the three estimators is "weighted" (a ratio estimator) while the other two are not; this alone is likely to cause differences. Stehman and Overton (1994b) give a consistent and non-negative estimator that is the same as Särndal, et. al.'s ratio suggestion. Other than consistency they provide no properties of their suggested estimator. Liu and Thompson (1983) prove that a "generalized" Horvitz-Thompson estimator of the population variance (the second suggested by Särndal, et. al.) is inadmissible in the set of unbiased estimators. They also show that an estimator based on the Yates-Grundy form (the third suggested by Särndal et. al.) is admissible among unbiased estimators under mild conditions.
Taken together these contradictory results provide only unsubstantiated suggestions, not a solution to the problem of which estimator should be used. Our sampling literature lacks a Monte Carlo study of this estimation problem such as Royall and Cumberland (1981) did for the ratio estimator and its variance and Stehman and Overton (1994a) did for the variance of the Horvitz-Thompson estimator. Studies such as these provide pragmatic answers for situations where more than one estimator is possible but an analytical comparison of the estimators proves difficult, if not intractable.

This study examines estimators of the population variance when sampling a finite population with a complex sample design. Lack of analytical theorems that pinpoint a best estimator for an estimation problem, a variety of estimator classes, and the complexity of the probabilistic mechanisms in design-based inference demand the simulation or Monte Carlo approach (Stehman and Overton, 1994a, and Särndal et. al., 1992, page 57). We examine the estimators’ analytical properties as far as possible but rely on a simulation study to provide pragmatic suggestions as to which estimators are appropriate under what conditions.

We suggest an estimator based on a method of moments procedure. This method of moments (MOM) estimator is the same as the second estimator suggested by Särndal et. al. (1992) and one suggested by Lui and Thompson (1983). We also show how this estimator can be weighted by Horvitz-Thompson estimators of the population total, producing two possible ratio estimators.
The estimators suggested by other researchers mentioned above are also examined. In particular we examine the unweighted sample variance $s^2$. The sample variance interests us because of its simplicity and how commonly it is used, often inappropriately (Urquhart et. al., 1998). Our study asks: "When is $s^2$ applicable?", "What can go wrong if $s^2$ is naively calculated from a sample collected with a complex design?", and "Which estimators perform better than the others and under what population and sample design conditions?"

Our comparisons are restricted to general random sampling (GRS) designs (McDonald, 1996, and Stehman and Overton, 1994b). General random sample designs are of interest as a result of their recent common use in large-scale environmental surveys and because of their simple method of sample selection. Although thus restricted, our results should provide guidance in many other situations.

Our simulations use two categories of populations: artificial and real. The artificial populations, created from known theoretical probability distributions, provide control over simulation parameters of interest. These simulations with artificial populations use a population space approach, varying simulation parameters in a factorial structure in order to make inference to an encompassed set of possible parameters (Stehman and Overton, 1994a). We vary four population parameters: 1. the shape or distribution of the auxiliary variable, 2. the correlation between response and auxiliary variable, 3. the variance of the inclusion probabilities, and 4. the mean of the response variable, and one sample design
parameter: the sample size. Real populations on the other hand are from real data. They supply population structure we are unable to develop artificially and add a sense of realism to results. Real data also provide a check for results from the artificial populations. The simulations in this paper are designed to encompass a wide range of populations but do not span the range of all possible. If a practitioner can place their population and sample design into one of our categories and choose an estimator based on our results then we have succeeded.

The layout of this paper is as follows. Section 3.3 introduces the notation and some preliminary results; section 3.4 derives the estimators and their characteristics; section 3.5 describes the empirical experiment we use to compare the estimators; section 3.6 contains the results of the experiment; and sections 3.7 & 3.8 discuss these results.

3.3 Notation and Preliminary Results

Let $U$ be our universe of interest consisting of $N$ elements, $1 < N < \infty$. We distinguish between the universe $U$ which is the set of elements and the population $y_i, i \in U$ which is the set of $N$ responses. A sample design $p(s)$ selects a sample $s$ such that the first-order inclusion probability for element $i \in U$ is $\pi_i$ and the second-order inclusion probability for elements $i, j \in U \times U$ is $\pi_{ij}$.

Let $H_i$ denote the sample-membership indicator for element $i \in U$, thus $H_i = 1$
when \( i \in s \) and \( H_i = 0 \) when \( i \notin s \) where \( s \) is the random sample. The \( H_i \) are random (indicator) variables whose sampling distribution depend upon \( p(s) \).

Vector-based notation allows for convenience in calculations and succinct notation (Dol, Steerneman, and Wansbeek, 1996). Throughout the rest of this paper we use it as much as possible; let

- \( y \) denote the \( N \times 1 \) vector of \( y_i \)'s,
- \( \pi \) denote the \( N \times 1 \) vector of \( \pi_i \)'s,
- \( \Pi \) denote the \( N \times N \) matrix of \( \pi_i \)'s, with \( \pi_i \)'s on the diagonal, and
- \( H \) denote the \( N \times 1 \) vector of \( H_i \)'s.

Define \( \circ \) and \( \otimes \) as elementwise multiplication and division respectively (the Hadamard product). The \textit{augmented random sample} in vector notation is \( y \circ H \). We use the dot/dash notation from Cornfield and Tukey (1956) to denote sums of sample values and population values respectively. For example,

\[
\pi_+ = \sum_{i=1}^N \pi_i = 1_N' (\pi \circ H) \quad \text{and} \quad \pi_- = \sum_{i=1}^N \pi_i = 1N \pi \]

denote the sum of the sample inclusion probabilities and the sum of the population inclusion probabilities respectively. Recall that \( \pi_- = n \), the sample size. We also borrow the accent bréve from Särndal et. al. (1992) to indicate a sample weighted scalar or vector, that is

\[
\bar{H}_i = H_i / \pi_i \quad \text{and} \quad \bar{H} = H \circ \pi .
\]

To demonstrate the vector notation, some population parameters are written in vector notation as follows. The mean is \( \overline{y}_- = \frac{1}{N} 1_N' y \); the deviations from this
mean or element effects are \( e = (I - \frac{1}{N} J)y \) (note, for the rest of the paper the length of vectors and size of matrices such as \( 1 \) and \( I \) are \( N \), the population size); and the variance is \( \sigma^2 = \frac{1}{(N-1)} y' (I - \frac{1}{N} J) y = \frac{1}{(N-1)} e' e \). The population in a linear model fashion is decomposed into the mean plus the deviation vector by \( y = 1\overline{y} + e \). The variance of the inclusion probabilities is \( \sigma^2 = \frac{1}{(N-1)} \pi' (I - \frac{1}{N} J) \pi \).

Upper-case letters denote random variables and lower-case, fixed values. The design-based approach takes the population as fixed and the sample-membership indicators as random, hence these are \( y \) and \( H \) respectively. The only exception to this rule is fixed matrices will be written as upper-case letters, but in bold, \( A \).

All expectations in this paper are with respect to the probability measure induced by the sample design. Some useful results are given by Särndal et. al. (1992) §2.6:

a) \( E(H) = \pi \)

b) \( E(HH') = \Pi \), and

c) \( \text{var}(H) = \Pi - \pi \pi' \).

Because we use corollaries to the Horvitz-Thompson (HT) Theorem we review the HT theorem here and present two corollaries:
**Horvitz-Thompson Theorem:** (Horvitz and Thompson, 1952) If unit $i \in U$ is selected with probability $\pi_i$, where $\pi_i > 0$ for all $i \in U$, then for any fixed $(N \times 1)$ vector $\lambda$, $\lambda' (y \circ \tilde{H})$ is unbiased for $\lambda' y$.

Proof: See Horvitz and Thompson, 1952.

**Corollary 1.** Let the $(N \times 1)$-vector $z$ be a second population on $U$. Denote

$\tilde{H} = HH' \circ \Pi$. If $\pi_{ij} > 0$ for all $i, j \in U \times U$, then for any fixed $(N \times N)$ matrix $A$,

$$ E \left( y' \left( A \circ \tilde{H} \right) z \right) = y' Az $$

Proof: $H$ is the only random element in the expectation and $E[HH'] = \Pi$ from b above. Quadratics result by taking $z = y$. This Corollary is adapted from a lemma stated in a series of memos by Dave Birkes, Trent McDonald, Tom Kincaid, and Scott Urquhart, 1995, Oregon State University.

**Corollary 2.** Let $z_{ij}$ be a population defined on $U \times U$. If $\pi_{ij} > 0$ for all $i, j \in U \times U$ then $\tilde{z} = \sum_{i,j \in U} z_{ij} / \pi_{ij}$ is unbiased for $z = \sum_{i,j \in U} z_{ij}$.

Proof: See Särndal et. al. (1992), page 48.
Corollary 2 is the basis for much of Lui and Thompson's (1983) results. Consider the universe \( V = U \times U \) then the inclusion probabilities for "elements" will be the \( \pi_{ij} \) and estimator suggested by the Horvitz Thompson Theorem is the estimator presented in the corollary. This approach is known as the batch approach.

3.4 Estimators for the Finite Population Variance

In this section we develop ten variance estimators having foundations in the Horvitz-Thompson (HT) Theorem and its two corollaries (Corollaries 1 and 2). We take the approach that the Horvitz-Thompson Theorem provides a foundation for finite-population estimation (Overton and Stehman, 1995). Unfortunately, the HT theorem and its corollaries suggest more than one unique estimator and do not supply a method for choosing the best from these estimators. Sampling theory in general lacks criteria that provide a best estimator. This section demonstrates how different expressions of the population variance along with the corollaries from Section 3.3 arrive at 10 competing estimators.

3.4.1 Introduction to Variance Estimation and Estimation Criteria

Three expressions for the population variance are useful for the development of estimators. The most common expression for the variance is the quadratic form \( \sigma^2 = (N - 1)^{-1} \sum_{i=1}^{N} (y_i - \bar{y})^2 = (N - 1)^{-1} y'(I - \frac{1}{N} J) y \). A second
expression, the bi-linear form, demonstrates how the variance is the sum of two population "averages":

$$\sigma^2 = \frac{1}{(N-1)} \sum_{i=1}^{N} y_i^2 - \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j$$

$$= \frac{1}{(N-1)} y'y - \frac{1}{N(N-1)} y'11'y$$

(3.1)

And a third expression, the Yates-Grundy form, provides a second quadratic form

$$\sigma^2 = \frac{1}{2N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} (y_i - y_j)^2.$$  (3.2)

The name of this quadratic form (3.2) is due to Yates and Grundy (1953) who derived a similar form for the variance of the Horvitz-Thompson estimator of the population total. We see that these three forms lead to different estimators, but first we examine some estimation criteria we would like our estimators to have: consistency, invariance, and non-negativity.

We first consider consistency. Consistency and unbiasedness are both criteria that attempt to ensure that one's estimator is estimating the parameter one aims to estimate (Fisher, 1959). Recall that the standard definition is a consistent estimator converges to its estimand in probability as the sample size goes to infinity. This definition has limited use in the finite population sampling because our sample size cannot exceed our population size hence cannot be let to go to infinity. As a result two alternative definitions of consistency are regularly applied in the sampling context. An estimator is finite consistent if when the sample size is
equal to the population size, that is the sample is a census, the estimator equals the estimand (Cochran, 1977, Section 2.4). Because a census implies $\pi_i = \pi_j = 1$, and $\pi_{ij} = \pi_{kl} = 1$, finite consistency is easy to evaluate but by the same token is not very useful. Moreover, an estimator is *Fisher consistent* if the estimator is a function of the observed frequencies of the population outcomes and assumes the parametric value when the expectations of these frequencies are substituted. Fisher consistency is defined completely in Overton and Stehman (1995) and Fisher (1956, page 143). It is common practice to demonstrate Fisher consistency by assuming functions of consistent estimators are consistent (Overton and Stehman, 1995 and Fisher, 1956). Thus any estimator that is a function of a HT estimator is Fisher consistent because HT estimators are Fisher consistent. We will see that most of our estimators are not unbiased but most are Fisher consistent. One estimator, the naïve estimator, is neither and hence should be watched carefully.

A second useful criterion is invariance. The population variance is *y*-location invariant, that is for any $c \in \mathbb{R}$ we have that $\sigma^2(y + c) = \sigma^2(y)$. It is also *y*-scale invariant, that is for any $b$ we have that $\sigma^2(by) = b^2 \sigma^2(y)$. Since the parameter of interest possesses these qualities it makes sense to require these qualities of our estimators. As an example, when measuring a temperature response in Centigrade, we expect the variance estimate to be proportional to the estimate when measuring the response in Fahrenheit. We will see that some of our estimators are not *y*-location invariant but only so in expected value; that is, for an
estimator \( \hat{S}^2(y) \) and constant \( c \), we may not have \( \hat{S}^2(y+c) = \hat{S}^2(y) \) for all \( y \) but instead \( E\hat{S}^2(y+c) = E\hat{S}^2(y) \). These estimators remain in consideration but also will be scrutinized.

As a final criterion we expect our estimators to be non-negative since the population variance is so. As with the other criteria we shall see that not all the estimators have this quality. Estimators that assume negative estimates need special consideration.

3.4.2 The Estimators

Corollary 1 suggests unbiased estimators can be constructed using a method of moments (MOM) approach. Beginning with the quadratic form of the population variance, \( \frac{1}{N-1} y'yB \) where \( B = (I - \frac{1}{N} J) \), then Corollary 1 suggests the unbiased method-of-moments estimator \( \frac{1}{N-1} y'(B \circ \tilde{H}_*) y \). Denote this estimator, the MOM estimator, with the symbol \( S_n^2 \). This MOM estimator is found in equation 5.9.14 in Särndal et. al. (1992) and shown to be inadmissible among unbiased estimators in Lui and Thompson (1983). Under simple random sampling it is equal to the sample variance. \( S_n^2 \) is not \( y \)-location invariant but is so in expected value, because for any constant \( c \)
The two additional terms beyond invariance compare traditional Horvitz-Thompson (HT) based estimation with estimation based on Corollary 2. Both additional terms have expectation zero.

This MOM estimator, as was hinted in the introduction, interests us particularly because of its flexibility. We ultimately are interested in estimating population variance components. Given an orthogonal ANOVA type decomposition of the population variance $y'y = y'\mathbf{A}y + y'\mathbf{B}y + \cdots$, we can estimate its components with method-of-moments estimators suggested by Corollary 1, $y'(\mathbf{A} \odot \mathbf{H}_*)y, y'(\mathbf{B} \odot \mathbf{H}_*)y, \ldots$. For this reason we are specifically interested in how the MOM estimator performs under the present simplified conditions.

A second approach to estimating the variance stems from the bi-linear form (3.1) being linear in two means. This form suggests we can use the HT estimator for the total of the squares $\sum_{n,s} \frac{y^2_{n,s}}{\pi_i} = y'(y \odot \mathbf{H})$ to estimate the total sum of squares $y'y$ as well as using the square of the Horvitz-Thompson estimator for the total, $(1'(y \odot \mathbf{H}))^2$, to estimate the square of the total $y_2$. Denote this estimator, the HT estimator, with the symbol $S^2$. It can be written in vector form as
This HT estimator is only approximately unbiased even under simple random sampling. Its bias is

\[ E(S^2 - \sigma^2) = \frac{1}{N(N-1)} y' (J - \Pi \otimes \pi \pi') y \]

\[ = \frac{1}{N(N-1)} \left( \sum_{i=1}^{N} y_i^2 \left( 1 - \frac{1}{\pi_i} \right) - \sum_{i=1}^{N} \sum_{j \neq i} y_i y_j \left( 1 - \frac{\pi_{ij}}{\pi_i \pi_j} \right) \right). \]

Another approach to estimation uses the Yates-Grundy version of the population variance (3.2). From this form Corollary 2 suggests the unbiased estimator

\[ S^2 = \frac{1}{2N(N-1)} \sum_{i \neq j} \left( \frac{(y_i - y_j)^2}{\pi_{ij}} \right). \]

Denote this estimator the YG estimator. It is the third estimator suggested by Särndal et. al. (1992, equation 5.9.17). Lui and Thompson also suggest this estimator and show that it is admissible over the set of unbiased estimators.

As stated previously, we are interested in the consequence of using the unweighted sample variance under a complex design. We call this the naïve estimator because it ignores the sampling probabilities (design); it has a simple vector form:
\[ s^2 = (n-1)^{-1} \sum_{i=1}^{N} (Y_i - \bar{Y})^2 = (n-1)^{-1} (y \odot H)' (I - \frac{1}{n} J) (y \odot H) \]

The naive estimator has many good properties but suffers primarily due to being neither Fisher consistent nor unbiased, although it is finite consistent. It is only unbiased in the case of simple random sampling (where \( \pi = \frac{n}{N} \cdot 1 \)) but for complex sampling designs the bias is

\[
E(s^2 - \sigma^2) = \frac{1}{n} \sum_{i=1}^{N} y_i^2 \left( \pi_i - \frac{n}{N} \right) - \frac{1}{n(n-1)} \sum_{i=1}^{N} \sum_{j \neq i} y_i y_j \left( \pi_{ij} - \frac{n(n-1)}{N(N-1)} \right) (3.3)
\]

We see that this bias may be small under two design considerations. First, the bias shrinks as the variance of the \( \pi \) shrinks (we will see the reason for this in section 3.5.3). Second, the terms \( \pi_i - \frac{n}{N} \) and \( \pi_{ij} - \frac{n(n-1)}{N(N-1)} \) each sum to zero implying this bias also is likely to be small if the \( Y \) and \( \pi \) are unrelated.

### 3.4.3 Ratio Forms of the Derived Estimators

It is commonly suggested in survey sampling that when estimating the mean of a population, instead of using the actual population size in the denominator, use the HT estimator of the population size in the denominator (Särndal et al., 1992, section 5.7; Thompson, 1992, page 70; and Stehman and Overton, 1994, equation 2.17). A variety of reasons, outlined in Särndal et al. (1992), support this approach even when the population size is known. Following this advice, it may be advantageous to estimate \( (N-1) \) or \( N(N-1) \) instead of using the true values. For
$N$ one would undoubtedly use the HT estimator $\hat{N} = l' \tilde{H}$ . For $N^2$, on the other hand, there are two possibilities both based on HT estimation and consistency, the square of the HT estimator $\hat{N}^2$ or an estimator suggested by Corollary 2, $\tilde{N} = \sum_{i,j} \frac{1}{\pi_{ij}}$. So $N(N-1) = N^2 - N$ can be estimated by either $\hat{N}(\hat{N} - 1)$ or $N^2 - \hat{N}$. The derived estimators that use $\hat{N}(\hat{N} - 1)$ in the denominator are denoted with a hat such as $\hat{S}_{\Pi}^2$ and abbreviated with a “-R” that indicates a “ratio” estimator. Those that use $\tilde{N}^2 - \hat{N}$ in the denominator are denoted with tilde such as $\tilde{S}_{\Pi}^2$ and abbreviated with a “-GR” that indicates a “generalized ratio” estimator. The complete set of estimators have a $3 \times 3 + 1$ structure (TABLE 3.1).

It is worth noting that estimating the size of the population is often necessary in environmental sampling because the true population size is often unknown.

The properties of all the estimators are given in TABLE 3.2.

<table>
<thead>
<tr>
<th>Estimator abbreviation</th>
<th>Population size divisor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N$</td>
</tr>
<tr>
<td>Na&quot;ive</td>
<td>$s^2$</td>
</tr>
<tr>
<td>MOME</td>
<td>$S_\pi^2$</td>
</tr>
<tr>
<td>HTE</td>
<td>$S_\pi^2$</td>
</tr>
<tr>
<td>YGE</td>
<td>$S_\pi^2$</td>
</tr>
</tbody>
</table>

TABLE 3.1
3×3+1 structure, symbols, and abbreviations for the 10 estimators.
<table>
<thead>
<tr>
<th>Estimator</th>
<th>Abbreviation</th>
<th>Suggested by</th>
<th>Consistency</th>
<th>Un-biased</th>
<th>Non-negative</th>
<th>y-Invariance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s^2$</td>
<td>NAIVE</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2_{fi}$</td>
<td>MOME</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>$S^2_{fi}$</td>
<td>MOME-R</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>?</td>
<td>No</td>
</tr>
<tr>
<td>$S^2_{fi}$</td>
<td>MOME-GR</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2_x$</td>
<td>HT</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2_x$</td>
<td>HT-R</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2_x$</td>
<td>HT-GR</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2$</td>
<td>YG</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2$</td>
<td>YG-R</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$S^2$</td>
<td>YG-GR</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**TABLE 3.2**

Descriptive properties of the estimators. The column “Suggested by” denotes where in the literature the estimators are found; SS&W refers to Särndal et. al. (1992), L&T refers to Liu and Thompson (1983), and S&O refers to Stehman and Overton (1994b).
3.4.4 Analytical Comparisons

A useful strategy to analytically compare the estimators is to explore how they estimate the two parts of the variance in its bi-linear form (3.1). Ignoring the multiplier \((N - 1)^{-1}\) the three derived estimators share estimators for the first part \((\sum y_i^2)\) and second part \((\frac{1}{n} \sum \sum y_i y_j)\) of the bi-linear form:

\[
S^2_x = \sum_{i=1}^{N} y_i^2 \frac{H_i}{\pi_i} - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \frac{H_i H_j}{\pi_i \pi_j}
\]

\[
S^2_n = \sum_{i=1}^{N} y_i^2 \frac{H_i}{\pi_i} - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \frac{H_i H_j}{\pi_{ij}}
\]

\[
S^2 = \frac{1}{N} \sum_{i=1}^{N} y_i^2 \sum_{j=1}^{N} \frac{H_i H_j}{\pi_{ij}} - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \frac{H_i H_j}{\pi_{ij}}
\]

(3.4)

Each estimator uses either the HT estimator or the estimator suggested by corollary 2 for each part of the variance. The HT estimator uses only the HT theorem, the YG estimator, only Corollary 2 and the MOM estimator, a mixture of the HT theorem and Corollary 2.

Examination of (3.4) reveals that if we consider the HT Theorem as a basis for finite population estimation, the MOM estimator is the most "natural" of the three. It estimates the first part of the population variance with the HT estimator and likewise it estimates the second part with the batch estimator (see the proof of Corollary 2).
3.4.5 Inclusion Probability Approximations

Some of the estimators suggested above require second-order inclusion probabilities. The exact computation of second-order inclusion probabilities is cumbersome for general random sample (GRS) designs (Hidiriglou and Gray, 1980), so we rely on two approximations. The approximations come from Hartley and Rao (1962) and Stehman and Overton (1989). The Hartley-Rao approximation is

$$\pi_{ij}^{hr} = \frac{(n-1)}{n} \pi_i \pi_j + \frac{(n-1)}{n^3} \left( \pi_i^2 \pi_j + \pi_i \pi_j^2 \right) - \frac{(n-1)}{n} \pi_i \pi_j \sum_{j=1}^{N} \pi_j + \frac{2(n-1)}{n^3} \left( \pi_i^3 \pi_j + \pi_i \pi_j^3 + \pi_i^2 \pi_j^2 \right) - \frac{3(n-1)}{n^4} \left( \pi_i^2 \pi_j + \pi_i \pi_j^2 \right) \sum_{j=1}^{N} \pi_j^2,$$

$$+ \frac{3(n-1)}{n^5} \pi_i \pi_j \left( \sum_{j=1}^{N} \pi_j^2 \right)^2 - \frac{2(n-1)}{n^4} \pi_i \pi_j \sum_{j=1}^{N} \pi_j^3$$

and the Stehman-Overton approximation is

$$\pi_{ij}^o = \frac{(n-1) \pi_i \pi_j}{2n - \pi_i - \pi_j}.$$

For $\pi_ii$ both approximations use $\pi_i$ since neither has $\pi_i^* = \pi_i$. For a comparison between these approximations we refer the interested reader to explore Stehman and Overton (1989). The Stehman-Overton approximation is especially desirable because it requires knowledge of inclusion probabilities only for the units in the sample.
3.5 Methods: A Simulation Experiment for Assessing the Estimators

This section describes the simulation experiments we use to assess the properties of the variance estimators. The experiments address two specific questions toward this objective: Under what conditions does the sample variance perform well? and Under what conditions do particular estimators outperform the others?

The experiments use artificial and real populations. The artificial populations follow (are realizations of) known probability distributions and allow for precise control of simulation parameters. The real populations, on the other hand, come from two environmental surveys and provide unforeseen and non-parametric conditions. We open this section with a description of the GRS sample plan, then we describe the simulation experiments for the artificial populations followed by the real populations, and we conclude with a detailed description of the simulation algorithm.

3.5.1 The GRS Sample Plan

In our simulations we consider only general random sample (GRS) designs. The designs in the GRS family are fixed-sample-size probability-proportional-to-an-auxiliary-variable (\( \pi px \)) designs that share the same sampling plan or method of sample selection. GRS are a subset of the more general, but rather unspecific, variable probability systematic (VPS) designs, which are also \( \pi px \) designs (Brewer
and Hanif, 1983, and Sunter, 1986). A VPS sample plan selects a sample by constructing a line whereon each universe element is represented by a line segment whose length equals the element’s inclusion probability. After a random start, a systematic sample then is selected by taking equal sized steps along the line and including in the sample every element on whose line segment a step lands (Stevens, 1997, Brewer and Hanif, 1983, and Sunter, 1986). GRS plans restrict the VPS plan by requiring that the population of first order inclusion probabilities sum to a pre-specified \( n \), producing a fixed sample size, and requiring that the elements be arranged on the line in random order.

We are primarily interested in GRS plans because of their common use in environmental surveys (Stehman and Overton, 1994b). Many environmental surveys conducted by the EPA, the National Stream Survey (Stehman and Overton, 1994a) and EMAP lake pilot in particular, use GRS plans. But we are also interested in GRS plans because they are considered the easiest \( \pi px \) plan to implement where the second order inclusion probabilities can be calculated (Hidiriglou and Gray, 1980) or approximated (Hartley and Rao, 1962, and Stehman and Overton, 1989).

Although we restrict our inference to GRS plans, within these plans we vary the nature of the auxiliary variable, hence the inclusion probabilities, and the sample proportion.
3.5.2 Population Families

The GRS family of designs defines a method of selecting a $\pi px$ design; the particular designs though, are distinguished by the distribution of their inclusion probabilities. The inclusion probabilities in turn depend on the auxiliary variable. This auxiliary variable, labeled $x$ above and subsequently, should be strictly positive and not be so variable as to produce degenerate designs (see Section 3.5.3). Toward our goal of generality, we aim to describe common auxiliary-variable structure found in environmental surveys by considering several distributions for $x$. 

FIGURE 3.1
Families of population structure for the auxiliary variable, denoted $x$. 

UNIFORM

GAMMA

BETA

MIXTURE

GROUPED
(FIGURE 3.1): one that is uniform (UNIFORM), one that consists of many small values but is positively skewed (GAMMA), one that consists of a few rare elements with small values and many large valued elements (BETA), one that consists of two clusters (MIXTURE), and one that consists of a finite number of groups (GROUPS).

The GROUPS family best describes the EMAP lake design (Larsen and Christie, 1993). We created the GROUPS family to have group sizes of auxiliary variable values proportional to those in the estimated distribution of US lake sizes in Larsen and Christie (1993, TABLE 3-1).

Notice that the GROUPS population resembles a grouped (or discrete) version of the GAMMA family. Taking this perspective we could think of grouping any of the other population families. The number of groups or bins then could be another parameter in the simulation experiment with the original population families having $N$ groups (see McDonald, 1996). This perspective may be useful but is not explored in this paper.

The responses for the families are generated using a linear regression model. After the random vector $x$ is generated following one of the families above, the response vector $y$ is generated as $y = \mathbf{1}\mu + \rho x + u$, where $\mu \in \mathbb{R}$, $-1 \leq \rho \leq 1$, and $u$ is a vector of independent and identically distributed normal random variables with 0 mean and variance $1 - \rho^2$. Both the response and the auxiliary variable are then standardized to have unit variance. Hence in all the artificial populations, we are estimating a variance equal to 1.
Because family comparisons are not of interest, we run a separate simulation experiment for each of these families. Populations created in the above manner are the artificial populations. The parameters $x$, $\mu$, and $\rho$ give us control over the structure in the population.

3.5.3 Treatment Structure

For the simulation experiments using artificial populations we consider three simulation parameters or factors that describe population structure (besides the population family): population correlation, inclusion probability variance, and response mean (TABLE 3.3) and a final factor that describes the sample design, the sample size. Our aim is for the set of population factors to be general enough so that researchers can associate real populations with them and at the same time to describe the main population structures that affect the performance of the estimators. The levels of these factors reflect what is seen in the EMAP lake survey data.

The bias of the naïve estimator (3.3) suggests two factors that affect estimator performance. The formula (3.3) suggests the estimator will be biased when the response and the inclusion probabilities are associated some way or when the design is "far" from simple random sampling. For the association between the inclusion probabilities and the responses we build populations with a linear relationships between the response and the auxiliary variable (Section 3.5.2).
<table>
<thead>
<tr>
<th>Family</th>
<th>$N$</th>
<th>Distribution of $x$</th>
<th>Distribution of $y$</th>
<th>Correlation</th>
<th>$\sigma_x^2$ ($n = 50$)</th>
<th>$\bar{y}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIFORM</td>
<td>1,000</td>
<td>Uniform $(0,1)$</td>
<td>Conditional normal</td>
<td>-0.95,...,0.95</td>
<td>0.018</td>
<td>0.6</td>
</tr>
<tr>
<td>GAMMA</td>
<td>1,000</td>
<td>Gamma $(\alpha)$</td>
<td>Conditional normal</td>
<td>-0.95,...,0.95</td>
<td>0.05,...,0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>BETA</td>
<td>1,000</td>
<td>Beta $(a,b)$</td>
<td>Conditional normal</td>
<td>-0.95,...,0.95</td>
<td>0.001</td>
<td>0.6</td>
</tr>
<tr>
<td>MIXTURE</td>
<td>1,000</td>
<td>Mixture of 2 normals</td>
<td>Conditional normal</td>
<td>-0.95,...,0.95</td>
<td>0.05,...,0.43</td>
<td>0</td>
</tr>
<tr>
<td>GROUPS</td>
<td>1,000</td>
<td>Discrete</td>
<td>Conditional normal</td>
<td>-0.95,...,0.95</td>
<td>0.08,0.22</td>
<td>0</td>
</tr>
<tr>
<td>LAKE 1</td>
<td>344</td>
<td>lake area &amp; (lake area)½</td>
<td>log total nitrogen</td>
<td>-0.11 and -0.21</td>
<td>0.21 and 0.58</td>
<td>5.92</td>
</tr>
<tr>
<td>LAKE 2</td>
<td>302</td>
<td>lake area &amp; (lake area)½</td>
<td>log secchi depth</td>
<td>0.13 and 0.23</td>
<td>0.22 and 0.59</td>
<td>0.934</td>
</tr>
<tr>
<td>ENGLISH</td>
<td>924</td>
<td>Tow length in hours</td>
<td>Log pounds English Sole caught in tow</td>
<td>-0.06</td>
<td>0.003</td>
<td>4.7</td>
</tr>
</tbody>
</table>

**TABLE 3.3**  
Description of Families, treatments, and treatment levels in the simulation experiments. The variance of the inclusion probabilities, $\sigma_x^2$, is given as the fraction from simple random sampling.
At the same time, we consider population correlations (the parameter denoted with \( \rho \)) that span the range of possible correlation (\( \rho = -1, \ldots, 0, \ldots, 1 \)) (TABLE 3.3).

No doubt other relationships occur in nature but the relationships between the chemical responses and lake area (the principal auxiliary variable) in the EMAP lakes data all appear linear with modest correlation, ranging from -0.30 to 0.27. Out of curiosity we are interested in correlations higher than those found in the EMAP lakes data including extreme correlations close to unity, although we admit these may rarely occur in environmental surveys.

The second factor suggested by the bias of the naïve estimator (3.3) is how far the design is from simple random sampling. The variance of the inclusion probabilities \( \sigma^2 \) measures the distance from simple random sampling (McDonald, 1996). This variance for GRS designs is constrained by the inequality

\[
0 \leq \sigma^2 \leq \frac{n(N-n)}{N(N-1)}.
\]

When \( \sigma^2 = 0 \) the design is equivalent to simple random sampling and \( \pi_i = \frac{1}{N} \) for all elements. When \( \sigma^2 = \frac{n(N-n)}{N(N-1)} \), on the other hand, the design degenerates to one that assigns probability 1 to the \( n \) units with the largest auxiliary variable values and 0 to all other units. So the fraction \( \sigma^2 / \frac{n(N-n)}{N(N-1)} \) measures how far the particular design is from simple random sampling. Our simulations use fractions ranging from about 0 to 0.5 (TABLE 3.3); values closer to 1 describe designs which rarely would be used in practice.
Since some of the estimators are not location invariant with respect to the response, a final population factor in the experiment is the average of the population response. We consider populations with means of 0 and 6 (TABLE 3.3). These levels also stem from the EMAP lake data where the means for the chemical variables on the logarithmic scale range from -0.2 to 6.

Finally we vary the sample size. We consider samples of 50 and 100 elements; except in the case of the English family where we consider more sample sizes, see Section 3.5.4. For a universe of 1,000 elements these translate to sample proportions of 0.05 and 0.10. These sample proportions are reasonable when it comes to environmental surveys. An estimate of the EMAP lakes survey sample proportion is 0.092. A limit on the size of the population due to computing power (1,000 elements), invokes a trade-off between low sample size (poor estimates) and high sample proportions (unrealistic conditions). We keep a relatively large sample size sacrificing a small sample proportion.

3.5.4 Real Populations

The real populations come from two environmental surveys. The first is the sampled data from the EMAP lakes pilot survey mentioned previously (Larsen and Christie, 1993) from which we consider two responses. The first response, Lake 1, is the natural log of total nitrogen (FIGURE 3.2). The universe consists of 344 lakes (elements). We consider two auxiliary variables: lake area and square root of lake area. The population correlation between log total nitrogen and lake area is -
FIGURE 3.2
Lake 1 population: Represented are 344 lakes from the EMAP surface waters lake pilot survey. The response, log total nitrogen, is plotted against the auxiliary variable and inclusion probabilities from lake area (A & B) and square-root lake area (C & D).

0.11 and between log total nitrogen and square-root lake area is -0.21. The variance of the inclusion probabilities calculated from lake area is 0.072 resulting in

\[ \frac{\sigma^2}{\frac{n(N-n)}{N(N-1)}} = 0.58 \] when the sample size is 50; the variance of the inclusion probabilities calculated from square-root lake area is 0.026 resulting in

\[ \frac{\sigma^2}{\frac{n(N-n)}{N(N-1)}} = 0.21 \] when the sample size is 50 (TABLE 3.3). The variance of log total nitrogen is 0.221.
The second population from the EMAP lakes pilot survey, Lake 2, has the natural log of mean secchi depth (FIGURE 3.3) as its response. The universe consists of 302 lakes. We also use lake area and square root lake area as auxiliary variables. The population correlation between log secchi depth and lake area is 0.13 and between log secchi depth and square-root lake area is 0.23. The variances of the inclusion probabilities are 0.0812 for lake area resulting in

\[ \frac{\sigma^2}{\frac{n(N-n)}{N(N-1)}} = 0.59 \] and 0.031 for square-root lake area resulting in

\[ \frac{\sigma^2}{\frac{n(N-n)}{N(N-1)}} = 0.22 \] when the sample size is 50 (TABLE 3.3). The variance of log secchi depth is 0.458.

The third real population, the English family, consists of log book entries from all the fishing-trawler tows originating from Newport, Oregon in 1996. The response is natural log of pounds of English Sole caught in the tow. Each tow is considered a sample unit; there are 924 tows in the universe. The auxiliary variable is the duration of the tow. Tows under one hour or over three hours in duration were eliminated from the list prior to sampling. For this population we consider sample sizes from 10 to 150. The population correlation between the tow length and log pounds English is -0.06. The variance of the inclusion probabilities for sample size 50 is \(1.3 \times 10^{-4}\) resulting in

\[ \frac{\sigma^2}{\frac{n(N-n)}{N(N-1)}} = 0.003 \] and for sample size
FIGURE 3.3

The lake 2 population consists of 302 lakes from the EMAP surface waters lake pilot survey. The response, log mean secchi depth, is plotted against the auxiliary variables and inclusion probabilities, lake area (A & B) and square-root lake area (C & D).

150 is 0.0018 resulting in $\sigma^2 \sqrt{n(N-n)}/N(N-1) = 0.013$ (TABLE 3.3, FIGURE 3.4). The variance of log pounds of English is 1.58.

3.5.5 Experimental Design Structure

Because we have no interest in comparison of families as explained earlier, the simulation experiment is actually a series of separate, independent experiments, one for each family. Within each family a factorial treatment design assesses the
FIGURE 3.4
ENGLISH population. Pictured are all 924 one to three hours tows originating in Newport, OR, in 1996. The response, log pounds of English, is plotted against the auxiliary variable, length of tow, and the inclusion probabilities when the samples size is 50 and 150.

effects of the four factors: population correlation, variance of inclusion probabilities, response mean, and sample size. Stehman and Overton (1994a) use a factorial structure described as a population space approach. Our approach is similar.

Simulation experiments may have two possible experimental units depending on their perspective. Typically, a design-based simulation selects a large number of samples from a fixed population on which the estimators are judged. Several populations of varying structure might be considered (Royall and Cumberland 1981, Stehman and Overton, 1994a, and McDonald, 1996). In such an experiment the samples are the experimental unit and inference is conditional on
the realized population (Stehman and Overton, 1994a). A second experimental unit
results when repeat population realizations are considered. For now we take the
conditional approach in this research so our results are conditional on our realized
populations.

3.5.6 Response Structure

The responses from a simulation should reflect the quality of the estimators
under consideration. The principal performance criteria we evaluate from
simulations are the estimators' empirical bias and mean square error (MSE). For an
arbitrary estimator $S^2$ the bias, conditional on the family realization $f$, is defined
as $E[(S^2 - \sigma^2)|f]$ where the expectation is approximated by the average of the
estimates from 10,000 simulated GRS samples. Likewise, the empirical MSE is
defined as $E[(S^2 - \sigma^2)^2|f] = \text{bias}^2 + \text{var}(S^2|f)$ where $\text{var}(S^2|f)$ is
approximated by the variance of the estimates from 10,000 simulated GRS samples.
Another criterion we use to judge the estimators is the proportion of negative
estimates.

3.5.7 Simulation Algorithm

A list of simulation parameters, along with their labels, is:
• Population family.
• Population correlation: $\rho$ -- linear correlation between response and auxiliary variable.
• Variance of inclusion probabilities: $\sigma^2_x$ -- may be scaled as the measure from simple random sampling, that is $0 \leq \sigma^2_x \leq 1$.
• Mean of population: $\bar{y}_x$.
• Sample size: $n$.

The levels for each of these parameters vary according to the population family (TABLE 3.3).

A simulation proceeds as follows:

I. Construct $\pi$ given family $x$ and $n$. Estimate the off-diagonal elements of $\Pi$.

II. Construct the response $y$ given $x$ and $\rho$.

III. Repeat the following 10,000 times:
   A. Draw a size-$n$ GRS sample from $U$ using $\pi$.
   B. Calculate the estimates from the $y$ in the sample.

IV. Approximate the estimators' sampling distributions using the 10,000 estimates.

The simulations are performed with SAS-IML version 6.1 and 7.0 on personal computers. The simulation code is included in the Appendix.
3.6 Results

We summarize our results in graph form. No attempt is made to statistically compare the estimators. Because there is a good deal of output from this set of experiments we concentrate on results that address the questions of interest and present only a subset of these. Recall that these results are conditional on population realizations, as described in Section 3.5.6. The current section is divided into two sub-sections. The first presents the results from the artificial populations describing the effects of population correlation, inclusion probability variance, and the population average. The second subsection presents the results from the real populations emphasizing the effect of sample size.

The Stehman-Overton and the Hartley-Rao second-order inclusion probability approximations behaved similarly. We made no attempt to compare these to or to calculate the true second-order inclusion probabilities. For the resolution of this study the approximations resulted in similar variance estimates. As a result, in what follows we present only estimates using the Stehman-Overton approximation.

Surprisingly, the effects of the simulation parameters (population correlation, inclusion probability variance, population average, and sample size) were similar across the auxiliary-variable families. Hence we only present the results for the GAMMA family (TABLE 3.4 and FIGURE 3.5, A and B), the UNIFORM family (FIGURE 3.6), and the three real populations (TABLE 3.5 and FIGURE 3.7).
FIGURE 3.5 A.
Contour plots of estimator MSEs. The contour lines are along the 10% quantiles for the combined MSEs (all estimators). Results are from one realization of the GAMMA family with $\bar{Y}_e = 0$ and the sample size 50. The horizontal axes are population correlation and the vertical axes are the measure from simple random sampling.
Contour plots of estimator MSEs. The contour lines are along the 10% quantiles of the combined MSEs. Results are from one realization of the GAMMA family. With \( \bar{y}_z = 6 \) and the sample size 50. The horizontal axes are population correlation and the vertical axes are the measure from simple random sampling.
3.6.1 Comparison of the Estimators with the Artificial Populations

In this section we present the estimators in order of complexity from the simplest (the naïve estimator) to the most complex (the generalized ratio estimators). One can look at FIGURE 3.5 and see the order we have chosen corresponds to examining the results column by column starting with the naïve estimator in the upper left corner. To compare the estimators over the entire population space we use locally weighted regression (loess) to interpolate between simulation evaluations. The simulation evaluations are at regular locations within the parameter space as described in the design (Section 3.5.3). This procedure is completed separately for the cases \( \bar{y}_1 = 0 \) and \( \bar{y}_1 = 6 \).

As expected the naïve estimator performed best when there was little population correlation (equation (3.3) and FIGURES 3.5 and 3.6). The plot of the naïve estimator bias against population correlation holding other parameters constant had an upside-down “U” shape and the MSE, a “U” shape (best seen in FIGURE 3.6, A and C but also FIGURE 3.5); large bias and MSE were associated with large magnitude of population correlation. The bias seemed to be responsible for the shape of the MSE plot as the naïve estimator’s variance decreased with the magnitude of the population correlation (FIGURE 3.6 B). The U-shape induced by population correlation thinned as the level of inclusion-probability variance increased. The entire surface had a half-cone shape (FIGURE 3.5). For low inclusion-probability variance the trough of the MSE “U” roughly spanned the population correlations of -0.5 to 0.5 while for high inclusion-probability variance,
on the other hand, this trough roughly spanned the population correlations of only -0.2 to 0.2 (FIGURE 3.5). Also as we expected, holding population correlation constant the MSE of the naïve estimator increases with increasing inclusion-probability variance.

The three estimators that use the true population size as divisors; the MOM estimator, the HT estimator, and the YG estimator; convincingly performed poorly in terms of their MSEs throughout the population space. Almost entirely their range of MSE is above the 50 percentile (FIGURE 3.5). Despite being unbiased the MOM estimator showed an average MSE of 22 attributable to its high variance (TABLE 3.4). Furthermore over almost the entire population space its MSE is over the 60 percentile (FIGURE 3.5). The HT estimator had a negative average bias of -13.11 and an average MSE of over 3 million (TABLE 3.4). These estimators’ MSEs improved with decreasing inclusion-probability variance, but other than this effect, neither demonstrated any consistent response pattern over the population space (FIGURE 3.5). Both of these estimators also assumed negative values; an average of 9% of the MOM estimates and 17% of the HT estimates were negative. The YG estimator, on the other hand, is unbiased, y-location invariant, and non-negative, but still suffered because of a larger average MSE, 36.4, than the naïve estimator and its ratio counterparts (TABLE 3.4).
Of the ratio estimators the Horvitz-Thompson ratio estimator and the Yates-Grundy ratio estimator behaved similarly. Their empirical MSEs were, relative to the naïve estimator's, quite flat throughout the population space and remained almost entirely between the 10 and 50 percentiles (FIGURE 3.5). Although biased, their biases were on average smaller than other estimators' biases, including the unbiased estimators' biases! The average bias for the HT-R estimator was 0.07 and for the YG-R estimator 0.09, compared with −0.11 for the MOM estimator and −0.2 for the YG estimator (TABLE 3.4). Like the naïve estimator the shape of their bias curves was an upside-down "U" with high absolute bias associated with high population correlation (FIGURE 3.6A) but this shape was less pronounced as in the
naïve estimator’s bias curve. Only when both absolute correlation and inclusion-probability variance have large values did these estimators have MSE values that exceeded the 50 percentile, values of around 2 (FIGURE 3.5 A). Their average MSE are virtually identical, 0.254 for HT-R and 0.262 for YG-R (TABLE 3.4). They are both also non-negative and y-location invariant.

The MOM-R estimator performed similarly to the two other HT-based ratio estimators when $\bar{y}_- = 0$, where its average bias was 0.17 and average MSE was 0.341 but when $\bar{y}_- = 6$ its average bias increased by a factor of 28 and average MSE increased by a factor of 167 (TABLE 3.4). When $\bar{y}_- = 6$ there is an overriding dependence on the inclusion probability variance that dominates the MSE (FIGURE 3.5, compare A to B). The MOM-R estimator, although it is difficult to prove it is non-negative analytically, assumed no negative values throughout the entire simulations.

Of the generalized-ratio estimators only the YG-GR estimator is y-location invariant and non-negative. The YG-GR estimator behaved similarly to the YG-R estimator but demonstrated a larger average bias and a slightly larger average MSE (TABLE 3.4). The MOM-GR estimator also behaved similarly to the MOM-R estimator when $\bar{y}_- = 0$. The MOM-GR is not y-location invariant but the simulation suggested that it is closer to being y-location invariant in expectation than the MOM or the MOM-R estimators; increasing $\bar{y}_-$ from 0 to 6 induced a trivial increase in the bias and only a 7 fold increase in the MSE (TABLE 3.4).
FIGURE 3.6
The best estimators for the UNIFORM family. The "U" shaped pattern of the naïve estimator exists in all the families and shrinks with the inclusion probability variance. Estimator traces not on the plots are left off the scale.
The patterns of the MSE over the population space were similar as well (FIGURE 3.5, compare A to B). The HT-GR estimator was variable with a MSE that ranged between 36.5 and 621,200 at high inclusion-probability variances. Both the MOM-GR and the HT-GR assume negative values. An average of 9% of the MOM-GR and 79% of the HT-GR estimates were negative.

3.6.2 Real Populations and the Effect of Sample Size

Results from the real populations supply little more information than what is presented above. TABLE 3.5 shows the estimators' average empirical bias and MSE from the simulations using the real populations. The results are what we expect from populations with near zero population correlation. The Lake 1 and Lake 2 populations have large values for inclusion probability variance (>0.5) which have little effect due to the lack of population correlation.

For the English population we ran simulations for several sample sizes. As we expect, increasing the sample size decreased estimators' bias and MSE (FIGURE 3.7). Perhaps unexpected, though, the decrease in bias was restricted to the ratio estimators, the naïve estimator's bias did not improve as the sample size increased (FIGURE 3.7 A). The decrease in MSE, on the other hand, is similar for the ratio and the naïve estimators (FIGURE 3.7 B).
<table>
<thead>
<tr>
<th>Population</th>
<th>Lake 1</th>
<th>Lake 2</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>log(total nitrogen)</td>
<td>log(secchi depth)</td>
<td>log(English caught)</td>
</tr>
<tr>
<td>Auxiliary variable</td>
<td>lake area</td>
<td>(lake area)²</td>
<td>lake area</td>
</tr>
<tr>
<td>estimator</td>
<td>bias</td>
<td>MSE</td>
<td>bias</td>
</tr>
<tr>
<td>NAIVE</td>
<td>-0.07</td>
<td>0.006</td>
<td>-0.05</td>
</tr>
<tr>
<td>MOM</td>
<td>0.07</td>
<td>382.349</td>
<td>0.19</td>
</tr>
<tr>
<td>MOM-R</td>
<td>4.60</td>
<td>31.734</td>
<td>0.78</td>
</tr>
<tr>
<td>MOM-GR</td>
<td>0.21</td>
<td>0.445</td>
<td>0.04</td>
</tr>
<tr>
<td>HT</td>
<td>-8.26</td>
<td>1301.097</td>
<td>-0.62</td>
</tr>
<tr>
<td>HT-R</td>
<td>-0.05</td>
<td>0.023</td>
<td>-0.01</td>
</tr>
<tr>
<td>HT-GR</td>
<td>-5.63</td>
<td>65.537</td>
<td>-0.76</td>
</tr>
<tr>
<td>YG</td>
<td>0.00</td>
<td>0.105</td>
<td>0.00</td>
</tr>
<tr>
<td>YG-R</td>
<td>-0.04</td>
<td>0.023</td>
<td>-0.00</td>
</tr>
<tr>
<td>YG-GR</td>
<td>-0.02</td>
<td>0.030</td>
<td>-0.00</td>
</tr>
</tbody>
</table>

**TABLE 3.5**
Bias and mean square error for estimators from the real population simulations. All results are from sample size 50.
FIGURE 3.7
Best estimators' bias and mean square error for the English population.
3.7 Conclusions

This section answers the questions presented in Section 3.5. First we consider the naïve estimator and then the best from the derived estimators. Perhaps the most striking observation is that the results were similar across the auxiliary-variable families. This implies that the simulation factors we chose catch most of the variation in the MSE. Both population correlation and inclusion probability variance are important factors.

Our results clearly suggest that the naïve estimator is the superior estimator when the response and auxiliary variables are close to uncorrelated, regardless of the other population parameters. However the exact cut-off value of population correlation for use of the naïve estimator depends on the level of inclusion-probability variance (FIGURE 3.8). The simulations confirm that the naïve estimator is biased but demonstrate that it has a smaller variance than the other unbiased estimators.

Under high population correlation the naïve estimator may be misleading for an important reason not completely demonstrated in the above plots and tables. The bias of the naïve estimator is a concern because the naïve estimator is not Fisher consistent. The reason the empirical MSE of the naïve estimator may be comparable to the MSE of the ratio estimators is because of its extremely small variance. But in some cases, because it is biased, this estimator may not be estimating what we believe it to be estimating. For this reason in cases of
intermediate to high population correlation, where the naïve estimator is biased, scientists should probably use a HT-based estimator.

The simulations suggest using either the YG-R or the HT-R estimator when the naïve estimator may be questionable (FIGURE 3.8). These estimators performed similarly throughout the parameter space (FIGURE 3.5). The HT-R has the added bonus of not requiring the second-order inclusion probabilities.

Other estimators proved unworthy. The MOM-R estimator did not perform as well as the other two ratio estimators. It may have suffered because it is composed of a mixture of the HT theorem based estimators and the corollary-based (batch) estimators and thus is difficult to weight properly (3.4). Using second-order inclusion probabilities to estimate the population size also proved to be unfavorable. Although the MOM-GR may be a better estimator than the MOM-R because it is approximately y-location invariant, it is not a clear favorite because of its slightly larger MSE when $\bar{y}_x = 0$ and may give negative values. The HT-GR, not surprisingly, performed poorly. Using first-order inclusion probabilities in the numerator and second-order inclusion probabilities in the denominator proves to be destabilizing rather than stabilizing. The YG-GR estimator, on the other hand, only performed slightly worse than the YG-R estimator, but does not warrant the extra complications of using second-order inclusion probabilities in the denominator.

The estimators that use the true population size in the denominator prove to be unreliable as well. Although some of these are unbiased, their variances are much higher than their ratio counterparts and they can give negative estimates.
Because our results were similar across the different families classifying real populations should not be difficult. To determine the best estimator to use researchers will need an estimate of the population correlation and the variance of the inclusion probabilities (or equivalently the coefficient of variation for the auxiliary variable – See Stehman and Overton, 1989). If population structures cannot be determined, the YG-R and the HT-R estimators give reliable results throughout the “realistic” population space \(|\rho| \leq 0.5\). But the simulations suggest that none of the estimators may give good estimates in the regions of the population space with both high population correlation and high inclusion probability variance (FIGURE 3.5).

The results from the real populations support the artificial population findings. The naïve estimator proved to be superior to the other estimators in simulations involving the three real populations (TABLE 3.5). The simulations using artificial populations predict this result as all three real populations had low population correlation (FIGURES 3.2, 3.3, and 3.4). The Lake 1 and Lake 2 populations are best modeled by the GAMMA family while the English population is perhaps best modeled by the UNIFORM family.

Even under the ideal conditions of low population correlation and inclusion probability variance given by the English population, the sample size proves to be important when estimating variances. Sample sizes of less than 50 may be suspect (FIGURE 3.7). But the simulations also suggest there is a point of diminishing
FIGURE 3.8
Mean square error difference between the naïve estimator and the HT-R estimator. The red portion implies naïve estimator’s MSE is less than the HT-R estimator’s MSE and the blue, the converse.
returns. For the English population this point may be around sample sizes of 75 to 100 units or sample proportions of about 0.075 to 0.1.

Finally the simulation provides an unofficial critique of the second-order inclusion probability approximations. The MOM and YG estimators are both unbiased but the simulation suggested they have a modest negative bias that may be due to these approximations (TABLE 3.4). If this is true, the generalized-ratio estimators may have performed poorly strictly due to this effect. Although it may be interesting to determine how the estimators perform using the true second-order inclusion probabilities, we are primarily interested in how these estimators perform under real conditions. Given the complexity of calculating the true inclusion probabilities, researchers are unlikely to make the calculations and hence likely to use the approximations. The Overton approximation is especially useful because one only needs to know inclusion probability values for the units in the sample and not in the entire population. Because real situations interest us, this study considers the approximations part of the estimation process.

3.8 Discussion

Our ultimate goal for this research is to derive estimators for variance components in complex finite population surveys. Toward this goal we derived Corollary 1 that provides method of moments estimators that account for the sampling design and apply to real populations. What we liked about this corollary is it applies to any quadratic population function and hence to the various sums of
squares that interest us. The population variance is the logical place to begin our investigation. Corollary 1 suggests the MOM estimator. Estimating the population variance is also a good starting place because there are other estimators by which we judge our MOM estimator. Analytical examination of the estimators (3.4) provided no evidence that our estimator will not perform as well as other suggested estimators. Nor did the literature (Särndal et. al., 1992).

The results of the simulation though are not as we wished. The YG-R and the HT-R estimators convincingly outperformed all the MOM based estimators (TABLE 3.4). It is not clear whether the Yates-Grundy and Horvitz-Thompson estimators have straightforward extensions to more complex population sums of squares. This will need to be explored.

This study suggests that for complex multipurpose surveys where variance estimation is an objective, designs that minimize the correlation between the response and the auxiliary variables are beneficial, one can then use the simple sample variance as an estimator. Perhaps this result will extend to more complex variance component models.

These results both support and refute the claims by Lui and Thompson (1983), Särndal et. al. (1992), and Stehman and Overton (1994b). They endorse the Yates-Grundy estimator being admissible among unbiased estimators, however the class of unbiased estimators includes the MOM and the YG estimators but none of the other estimators considered in this study (Lui and Thompson, 1983). Because of their high variance neither of these estimators performed as well as some of the
others. This suggests admissibility theorems, when restricted to unbiased estimators such as Lui and Thompson's, are of limited use in the design-based arena. Our results disagree with Särndal et. al.'s (1992) claim that the HT-R estimator, the MOM estimator, and the YG estimator, perform similarly; unless they mean for much larger sample sizes, for which we have no comment. On the other hand, we suggest, Särndal et. al. (1992) should state that two of the three ratio versions of these estimators, HT-R and YG-R, behave similarly, a conclusion which our evidence supports even for moderate sample sizes (FIGURE 3.7). Finally, these results support Stehman and Overton's (1994a) conclusion that using the HT ratio estimator rather than the HT estimator can improve variance estimation.

As discussed in Section 3.5.5 simulation experiments such as this one have two possible experimental units depending on their perspective. Typically, as has been done here, a design-based simulation selects large number of samples from a fixed population on which the estimators are judged. The samples are the experimental unit and inference is conditional on the realized population. A model-assisted approach, on the other hand, suggests that a simulation should also include replicate populations. In this sense a population realization is an experimental unit and the samples are sub-units within populations. This adds a second source of variation for evaluation. This second source of variation describes the consistency of the estimator over populations of the same structure. Inference then is conditional on the population structure rather than the realized population. Performing such a model-assisted design-based simulation is underway.
Finally, any of the estimators explored in this paper can be modified to estimate population covariances. The simulation results above may still apply if the population correlation between each of the responses and the auxiliary variable are identical or similar. Otherwise new simulations should be used to judge the effect of different population-correlations.

3.9 References


Chapter 4

Summary

This chapter summarizes the entire thesis and discusses topics for further research. Both Chapters 2 and 3 contain their own conclusion and discussion so no further implications are developed here. Rather I tie the two chapters together and discuss immediate further research.

We explored two related topics in environmental statistics, methods for data exploration and estimators for a population variance. Researchers in the environmental statistics community have acknowledged the need for methods for data discovery. Non-linear methods such as projection-pursuit guided tours may prove especially important for environmental applications because of the complex nature of environmental data. The direct-search method explored in Chapter 2 proves to be a competent method supplementing the tools already available for data exploration. Chapter 3 explored the difficult problem of population variance estimation under complex sample designs. Surprisingly not much research has been done in this area of statistics. Because environmental monitoring is a controversial topic we wish to use design-based methods. The results of our study supply practical advice for estimator selection over a variety of population and design characteristics. Both of these topics are important for environmental monitoring.

Consequently these methods are likely to be used concurrently. Suppose a manager or researcher wishes to design a survey to study the status of a resource in
a region. The first question is likely to be: How much material do we have to collect in order to get good estimates? Chapter 3 outlined how population variance components provide answers to sample size inquiries. Further suppose the researcher conducts a pilot survey where he or she, because of a lack of a good frame, collects data with a complex design. Then the methods in Chapter 3 are applicable for the estimation of the variance. These methods though, demand some knowledge of population relationships. Specifically, is there a relationship between the inclusion probabilities and the response but also suppose population health cannot be directly measured but a surrogate or index response needs to be created. Indexes such as this are usually nonlinear functions of measured variables; further many different promising indexes can be derived. Which index is least variable? and how are the different indexes related? How are the indexes related to the design inclusion probabilities? The projection-pursuit guided tour tools explored in Chapter 2 will help answer these questions. Thus the tools explored in this thesis could be used concurrently in environmental survey design.

As usual in scientific inquiry the questions answered in this thesis invoke new questions. The next step for the visualization is implementation. No method will be used unless there exists a practical mode of use. This means guided-tour software that includes the pattern search method as a tool. XGobi mentioned in Chapter 2 is a model for new software but what is needed is software for personal computers not UNIX based workstations. The new ORCA software under construction at the University of Washington is an example (see http://pyrite.cfas.washington.edu/orca).
The findings in Chapter 3 also suggest topics for further research. The most obvious concerns estimating variance components under more complex population structure. Population "models" that lead to variance components are the result of population structure and survey objectives rather than assumed sample structure. How are we going to estimate components of variance when our method of moment estimator proved not to be a good estimator? Generalization of the Horvitz-Thompson estimator or the Yates Grundy estimator may be possible. We also failed to explore an estimator suggested by the SAS software that already has a straightforward generalization. Finally all the variance estimators have covariance forms. How do they compare when we are estimating a covariance between two populations? What if one of our responses is positively related to the inclusion probabilities and the second is negatively related? With additional work the solutions in Chapter 3 might be extended to these situations.
Bibliography


APPENDIX
Appendix

Listing of Computer Code

This appendix lists the source code wrote to perform the simulations presented in Chapter 3. The simulations were implemented in SAS© (SAS Institute, Cary NC) and relied heavily on the SAS macro facility and Proc IML, the matrix procedure in SAS. I will give electronic copies of the simulation files away freely.

/* ######################################################### */
/* This file needs to be in a location that all the modules can reach for my computer it is d:\ */
/* droot needs to be the same as sishome in simulate.sas */

%let droot = d:\pip\finite\sassim;
/* IML modules */
libname simprocs "&droot\simprocs";
/* simulation results */
libname ... 
/* other simulation files */
libname simu "&droot";

/* +++++++++++++++++++++++++++++++++++++++++++++++++++++++ */
/* This will run the simulation for a set of simulation parameters Simulate.sas and the other modules should be executed before this can be. */

options nosource2 nomprint nosymbolgen;
proc printto log = "&sirnhome/simlog.log";

%let cor1 = -0.95;
%let cor2 = -0.9;
%let cor3 = -0.7;
%let cor4 = -0.5;
%let cor5 = -0.4;
%let cor6 = -0.3;
%let cor7 = -0.2;
%let cor8 = -0.1;
%let cor9 = 0.0;
%let cor10 = 0.1;
%let cor11 = 0.2;
%let cor12 = 0.3;
%let cor13 = 0.4;
%let cor14 = 0.5;
%let cor15 = 0.7;
%let cor16 = 0.9;
%let cor17 = 0.95;

%let alpha1 = 0.532;
%let alpha2 = 0.263;
%let alpha3 = 0.178;
%let alpha4 = 0.1313;

%let an1 = a;
%let an2 = b;
%let an3 = c;
%let an4 = d;

%let nm1 = _h;
%let nm2 = _9;
%let nm3 = _7;
%let nm4 = _5;
%let nm5 = _4;
%let nm6 = _3;
%let nm7 = _2;
%let nm8 = _1;
%let nm9 = 00;
%let nm10 = 01;
%let nm11 = 02;
%let nm12 = 03;
%let nm13 = 04;
%let nm14 = 05;
%let nm15 = 07;
%let nm16 = 09;
%let nm17 = 0h;

%macro runsim;
%do i = 0 %to 0;
   %put &i;
   %do n = 1 %to 17;
      %put &n;
      %do m = 1 %to 4;
         %put &m;
      %end;
   %end;
%let sd = 865804;
%let u = 0;
%syscall ranuni(sd,u);
%let d1 = %sysevalf(&u * 1000000000,floor);
%syscall ranuni(sd,u);
%let d2 = %sysevalf(&u * 1000000000,floor);
%syscall ranuni(sd,u);
%let d3 = %sysevalf(&u * 1000000000,floor);

%put &&cor
%put &&alpha

%GOSIM( &d1, &d2, &d3, 
1000, &cor, &alpha, 6, 
3, 50, 
10000, 
0, 
g&&an..&&nm..6&i..txt, 
gpr&&an..&&nm..6&i..txt, 
gxy&&an..&&nm..6&i..txt, 
g2o&&an..&&nm..6&i..txt, 
g2h&&an..&&nm..6&i..txt, 
gsm&&an..&&nm..6&i..txt); 
%end;

%end;

%mend runsim;

%runsim;

proc printto;
run;

/* #---------------------------------------------------------------------------*/
/* +--------------------------------------------------------------------------*/
/* SIMULATE.SAS */
/* Start up file for simulations. This is the first file executed. It contains all the parameters. */
/* I usually submit this file line by line (or section by section) to avoid errors. */
/* Pip Courbois 9/22/99 */
/* +--------------------------------------------------------------------------*/
/* Simulation root directory. The file librefs.sas describes the directories under this root directory. These commands need to */
/* be changed for each platform. */

%let simhome=c:\pip\sassim3;

/* Librefs.sas needs to be in a location that all the sas programs and modules can find */
%include 'librefs.sas';

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++ */
/* START MACRO: start here if this is the first run of the */
/* simulation or some of these modules have changed. */
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

%macro startup;

/* Include the population module or script */
%include "&sirnhome\makedata.sas";
%include "&sirnhome\families.sas";
%include "&sirnhome\psingle.sas";

/* Include the modules for sampling */
%include "&sirnhome\fgrs.sas";
%include "&sirnhome\scalepi.sas";
%include "&sirnhome\sample.sas";
%include "&sirnhome\iprobs.sas";
%include "&sirnhome\jointo.sas";
%include "&sirnhome\jointhr.sas";

/* Include the modules for estimation */
%include "&sirnhome\estvar2.sas";

%mend startup;

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++ */
/* START 2: Start here if the modules above have already */
/* been defined. */
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* Macro to open storage catalogs where the modules are */
/* found */
%let opencats %quote( reset storage = simprocs.sprocs;
    load module = (sample scalepi
        fgrs sgrs jointo
        jointhr iprobs estvar);
);
%let openpops %quote( reset storage = simprocs.sprocs;
    load module = (makedata gamfam unifnorm mixture
        gamnorm params grouped invfam););

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++ */
/* Macro to set parameters and run simulation. */
%macro GOSIM(Popseed, smpseed, xseed, capN, p, alpha, muY, swPOP, sampsize, niter, long, results, parmeter, xandy, pi20, pi2hr, sumfile);
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* SIMULATION parameters. Set at the beginning of each simulation. */
/*---------------------------------------------------*/
/* The random seed: */
/*---------------------------------------------------*/
%macro setseed;
   /* to duplicate a sample or a population SAS must be 
   restarted or the variables SMPseed and POPseed must 
   be given new names. Each of these starts a random 
   number sequence */
   c = 1;
   Popseed = &popseed;
   call rannor(Popseed,c);
   SMPseed = &smpseed;
   call ranuni(SMPseed,c);
%mend setseed;

/*---------------------------------------------------*/
/* The population: */
/*---------------------------------------------------*/
/* There are several different populations here. The variable 
   swPOP is a switch to choose between the different families of 
   populations. The choosing is done in main.sas. Here we 
   simply define different families. The variable p is to 
   choose members of the families. */
%macro popnread;
   /* Read in the population from a SAS dataset. */
   use simu.marcia;
   read all var(L_ENGLIS DURATION) where(L_ENGLIS ^= . & PORT = (M2 & YEAR = 96 ) into marcia;
   popvals = marcia[,1] ;
   x = marcia[,2];
   capN = nrow(popvals);
%mend popnread;

/*---------------------------------------------------*/
/* create random data according to the parameters 
in module makedata */
/* The following keeps the auxiliary variable the same over multiple simulations */

xseed = &xseed;
call rannor(xseed,c);

run makedata(popvals,x,&capN,&p, &alpha, &muY, &swpop,Ppopseed,Xseed);
done = {done with makedata};
print done;

print popvals;
%mend randdata;

%macro calcpar;
/* This macro calculates the parameters. */

/* single stage sampling */
run psingle(sigmapop, popvals);
done = {Done with psingle};
print done;
%mend calcpar;

/* ------------------------------------------------------------- */
/* Sample design: */
/* ------------------------------------------------------------- */

/* VRS (swVRS = 1) or SRS (swVRS = 0)*/
%let swVRS = 1;

/* VRS ordered (swORD = 1) or random (swORD = 0) (either for SRS) */
%let swORD = 0;

/* Insert the order vector here, needs to be (NX1) If SRS or random VRS use ord = 0 */
%let ord = 0;

%let pids = iprobs(pi1, pi2o, pi2hr, x, &swVRS, &sampsize);

%let design = FGRS(s, pi1, &sampsize, &sword, &ord, smpseed);

%macro design;
/* do the actual sampling */
call &design;
%mend design;

/* ------------------------------- */
/* Estimation: */
/* ------------------------------- */

%macro estimate;
  /* this macro calculates the estimators */
  call estvar(sigma, s, popvals, pil, PIPiPrime, pi20, pi2hr);
  %mend estimate;

/* ------------------------------- */
/* Other utilities: */
/* ------------------------------- */

/* set the number of iterations */
%let niter = &niters;

/* Store the results */
%macro results;
  /* the second term in rsigma2 declaration tells how many estimators we are comparing +1 for iteration number. */
  nest = 1 + 10;
  rsigma2 = J(&niter,nest,0);
  %mend results;

/* summary.sas includes summarizing the estimates */
%include "&simhome\sumstat.sas";

/* ------------------------------- */
/* Run the simulation: Main.sas has the simulation loop. */
/* ------------------------------- */

%include "&simhome\main.sas";

%mend GOSIM;

/* #################################################################### */
/* +------------------------------------------------------------------+*/
MAIN.SAS
main file for the simulation.
Pip Courbois 9/22/99.
+------------------------------------------------------------------*/

%include "librefs.sas";
proc iml symsize = 524;
reset log;

/* show modules; */
&opencats;
&openpops;

/* set the random seed */
%setseed;

/* Create the population */
if &swpop = 0 then
    do;
        %popnread;
    end; /* show names; */
else
    do;
        %randdata;
    end;

%calcpar;

/* inclusion probabilities. */
call &pids;
    PIPIprime = pil * pil';
done = (done with pids);
print done;

/* Output */
%results;

/* iteration loop */
do iterate = 1 to &niteri;
/* sample according to the design */
    free s;
%design

/* compute the statistics */
%estimate; /* resume; */
/* save the results */
    results = iterate || sigma;
    rsigma2[iterate,] = results;
    print(iterate);
end;

show space;

filename rparam "&simhome\simresults\&parmeter";

file rparam;
  seeds = &popseed | &smpseed | &xseed | &p | &alpha | &muY | bigN | nA;
  do i = 1 to ncol(sigmapop);
    put (sigmapop[i]) 6.4 +2 @;
  end;
  do i = 1 to ncol(seeds);
    put (seeds[i]) 8.0 +2 @;
  end;
  /* put &p 6.2; does not work for -0.3 */
  put;
closefile rparam;

filename reslts "&simhome\simresults\results";
file reslts;
  do i = 1 to nrow(rsigma2);
    do j = 1 to ncol(rsigma2);
      put (rsigma2[i,j]) 6.4 +2 @;
    end;
  put;
end;
closefile reslts;

filename xandy "&simhome\simresults\xandy";
file xandy;
  do i = 1 to nrow(popvals);
    put (popvals[i]) 6.4 +2 @;
    put (x[i]) 6.4 +2 @;
    put (pi1[i]) 6.4;
  end;
closefile xandy;

if (&long = 1) then
do;

filename pi2o "&simhome\simresults\pi2o";
file pi2o;
  do i = 1 to nrow(popvals);
    do j = 1 to nrow(popvals);
      put(pi2o[i,j]) 8.6 ;
    end;
e
end;
closefile pi2o;

filename pi2hr "&simhome\simresults\&pi2hr";
file pi2hr;
do i = 1 to nrow(popvals);
do j = 1 to nrow(popvals);
   put(pi2hr[i,j]) 8.6;
end;
end;
closefile pi2hr;
end;
%
summary;
quit;

/* ################################################################################ */
/* ################################################################################ */
/* MAKEDATA.SAS  --  This script creates random data for bi-sampling. No interaction effect.  9/22/99 */
/* ################################################################################ */
/* ################################################################################ */

/* MAKEDATA is a module that creates a population for use in the simulation. */

INPUTS: Population parameters.
   bigN -- population size.
   rho, alpha, muY,
   swpop -- to choose between populations.
   Xseed -- the seed for the auxiliary variables.

OUTPUTS:
   popvals - population responses.
   x - population auxiliary variables
*/
%
include "librefs.sas";

proc iml;

start makedata(popvals,x, bigN, rho, alpha, muY , swpop, popseed, Xseed);

/* gamma-normal population */
if swpop = 1 then
do;
muX=0;
beta = 1;
run gammnorm(popvals,x,bigN, alpha, beta, muX, rho, muY,popseed,xseed);
end;

if swpop = 2 then
do;
muX=0;
run unifnorm(popvals,x,bigN, muX, rho, muY,popseed,xseed);
end;

if swpop = 3 then
do;
muX=0;
run grouped(popvals,x,bigN, alpha, muX, rho, muY, popseed);
end;

if swpop = 4 then
do;
muX=4;
beta = 1;
run mixture(popvals,x,bigN, alpha, beta, muX, rho, muY,popseed,xseed);
end;

finish makedata;
reset storage=simprocs.sprocs;
store module=makedata;
quit;

/* save as sasdataset
reset storage = simu.simdata;
create lakes var{popvals qa qb};
append;
close lakes;
*/

/* Families.SAS -- This script creates random auxiliary
and response vectors.  

version 2  

+FAMILIES is a module that creates pairs of random vectors for use in the simulation.  

**INPUTS:**  
capN - The population size.  
parameters - vary according to the family, see below.  
Xseed - random seed to create auxiliary variable.  
popseed - random seed for responses.  

**OUTPUTS:**  
y - population responses.  
x - auxiliary variable values.  

%include "librefs.sas";  
proc iml;  

start gamnorm(y,x,capN,alpha,beta,muX,rho,muy,popseed,Xseed);  

/* x gamma, y normal, linear regression relationship */  

x = J(capN,1,1);  
u = J(capN,1,1);  
run rangam(xseed,alpha,x);  
x = beta * x;  
ones = J(capN,1,1);  
varX = (x'(*I(capN)-(1/capN)*ones*ones')*x)/(6apN - l);  
x = sqrt(l/varX) * x;  
call rannor(Popseed,u);  
/* normalize */  
varU = (u'(*I(capN)-(1/capN)*ones*ones')*u) / (capN - 1);  
u = sqrt((1-rho*rho)/varU) * u;  
/* simple regression relationship */  
y = rho*x + u;  
varY = (y'(*I(capN)-(1/capN)*ones*ones')*y) / (capN - 1);  
y = sqrt(1/varY) * y - ((1/capN)*ones*ones'y) + muy;  
x = x + muX;
finish gamnorm;
reset storage=simprocs.sprocs;
store module=gamnorm;

start unifnorm(y,x,capN,muX,rho,muY,popseed,Xseed);

/* x uniform, y normal, linear regression relationship */
x = J(capN,1,1);
u = J(capN,1,1);

run ranuni(xseed,x);

ones = J(capN,1,1);
varX = (x'*(I(capN)-(1/capN)*ones*ones')*x)/(capN - 1);
x = sqrt(1/varX) * x;
call rannor(Popseed,u);

/* normalize */
varU = (u'*(I(capN)-(1/capN)*ones*ones' )*u) / (capN - 1);
u = sqrt((1-rho*rho)/varU) * u;

/* simple regression relationship */
y = rho*x + u;

varY = (y'*(I(capN)-(1/capN)*ones*ones')*y) / (capN - 1);
y = sqrt(1/varY) * y - ((1/capN)*ones*ones'*y) + muY;
x = x + muX;

finish unifnorm;
reset storage=simprocs.sprocs;
store module=unifnorm;

start grouped(y,x,capN,alpha,muX,rho,muY,popseed);

/* x gamma, y normal, linear regression relationship */
capN=1000;
u = J(capN,1,1);
x1= J(620,1,1.5);
x2= J(168,1,7.5);
x3 = J(160,1,25);
x4 = J(45,1,275);
x5 = J(6,1,2750);
x6 = J(1,1,15000);

x = x1//x2//x3//x4//x5//x6;

ones = J(capN,1,1);
varX = (x'*(I(capN)-(1/capN)*ones*ones')*x)/(capN - 1);
x = sqrt(1/varX) * x;

call rannor(Popseed,u);

/* normalize */

varU = (u'*(I(capN)-(1/capN)*ones*ones')*u) / (capN - 1);
u = sqrt((1-rho*rho)/varU) * u;

/* simple regression relationship */

y = rho*x + u;

varY = (y'*(I(capN)-(1/capN)*ones*ones')*y) / (capN - 1);
y = sqrt(1/varY) * y - ((1/capN)*ones*ones'*y) + muY;

if min(y)<0 then y = y + abs(min(y));

x= x + alpha;

finish grouped;

reset storage=simprocs.sprocs;
store module=groupedi;

start mixture(y,x,capN,alpha,beta,muX,rho,muY,popseed,Xseed);

/* x mixture of normals, y normal, linear regression relationship */

x = J(capN,1,1);
u = J(capN,1,1);

run rannor(xseed,x);
run rannor(xseed,u);

ju = loc(u<alpha);
x[ju] = x[ju] + muX;
ju = loc(u>alpha);
x[ju] = x[ju] + muX + 5;

x = beta * x;
ones = J(capN,1,1); 
varX = (x*(I(capN)-(1/capN)*ones*ones')*x)/(capN - 1); 
x = sqrt(1/varX) * x; 

u = J(capN,1,1); 
call rannor(Popseed,u); 

/* normalize */ 
varU = (u*(I(capN)-(1/capN)*ones*ones')*u) / (capN - 1); 
u = sqrt((1-rho*rho)/varU) * u; 

/* simple regression relationship */ 
y = rho*x + u; 

varY = (y*(I(capN)-(1/capN)*ones*ones')*y) / (capN - 1); 
y = sqrt(1/varY) * y - ((1/capN)*ones*ones'*y) + muY; 

finish mixture; 
reset storage=simprocs.sprocs; 
store module=mixture; 

proc iml; 
quit; 

/ * PSINGLE.SAS - IML module to calculate the population parameters (variance component) for a single stage design. 
Pip Courbois 9/22/99 */ 
/* psingle is a module that calculates the population variance. 
INPUTS: 
y - population responses. 
OUTPUTS: 
popsigma - population variance component parameter. */ 
%include "librefs.sas"; 
proc iml; 
start Psingle(popsigma, y);
/* explain here */
/* MSB first */
N = nrow(y);
eye = I(N);
ones = J(N,N,1);
popsigma = 1/(N-1) * y'*(eye - ones/N)*y;

finish;
reset storage=simprocs.sprocs;
store module=psingle;
quit;

/** #*--------------------------------- */
/** * FGRS.SAS - IML module to collect a generalized random sample. */
/** *--------------------------------------------------------------- */
/** FGRS.SAS - factor generalized random sample. This module in conjunction with the module sample collects a generalized random sample of the levels for a factor. */
/**--------------------------------------------------------------- */

/ * FGRS.SAS - factor generalized random sample. This module in conjunction with the module sample collects a generalized random sample of the levels for a factor. */

INPUTS:

  nf - sample size.
  pil - 1st order inclusion probabilities.
  swordf - switch to specify an order for VRS. If swordf = 0 then the program picks a randomized VRS. If swordf = 1 then the program picks an ordered VRS.
  ordf - Vector of same length as the population denoting the order for ordered VRS. If the xf are in the correct order or the design is randomized VRS then ordf = 0.
  smpseed - seed for sampling.

OUTPUTS:

  sf - population level indices of selected units.

*/

%include "librefs.sas";
proc iml;
start FGRS(sf, pil, nf, swordf, ordf, smpseed);
popsize = nrow(pi1);

call sample(sf, pil, popsize, nf, swordf, ordf, smpseed);

finish;

reset storage=simprocs.sprocs;
store module=FGRS;
quit;

/*/ *******************************************************************************/
/*  ---------------------------------------------------------------------------
SCALEPI.SAS
IML module that scales auxiliary data into first order inclusion probabilities.
from Trent McDonald.
-----------------------------------------------------------------------------*/

%include 'librefs.sas';
proc iml;
start scalepi( n, x);
/* The purpose of scalepi is to create inclusion probabilities for 
 pi_p_x 
 design (prob. inclusion proportional to x). The pi vector 
 should sum 
to n the sample size and have no elements > 1.

Inputs:  n - sample size 
        x - vector of auxiliary variables (for the 
 population).

Output:  Saves the pi vector into the simu directory.
*/

_pi1 = n # (x / sum(x));
do while( any(_pi1 > 1) )
   _i = loc(_pi1 >= 1);
   _k = loc(_pi1 < 1);
   _pi1[_i] = j( nrow(_i),1 );
   _pi1[_k] = (n - nrow(_i)) * (_pi1[_k] / sum(_pi1[_k]));
end;

return( _pi1 );

finish scalepi;

reset storage = simprocs.sprocs;
store module = scalepi;
quit;
SAMPLE.SAS
IML module that collects a sample from the population indices. Adapted from Trent McDonald's routine, see his thesis.


Sample collects a VRS sample. Note this is different than Trent's in that the first order inclusion probabilities must be supplied already scaled.

INPUTS:
- popsize - size of the population.
- pils - 1st order inclusion probabilities for population elements. See iprobs.sas.
- sampsize - sample size.
- swords - switch for ordered VRS. if swords = 0 then sample collects a randomized VRS, if swords =1 then sample collects an ordered VRS.
- ords - the order vector for ordered VRS. Note this should be set = 0 if the population is already ordered correctly or swords = 0.

OUTPUTS:
- ss - sampled indexes.

/*
%include 'librefs.sas';
proc iml;
start sample(ss, pils, popsize, sampsize, swords, ords, smpseed);

/* Do not return randomized probabilities */
pils0 = pils;

/* Check error conditions
   if sampsize > popsize then
      do;
         file log;
         put "SAMPLE ERROR: Sample size cannot be greater than population size";
         return;
      end;
   else if (nrow(pils) ^= popsize) then
      do;
         file log;
         put "SAMPLE ERROR: Pi_1 vector size not equal to population size";
         return;
*/
end;

/* initialize */
popu = (1:popsize)';

/* If systematic (sw_ordered) and vector order is given, 
sort labels 1:N by the order vector. This gives a way 
to sample across some set of 'X' values. */
if (swords) & (ords ^= 0) then do;
    if (nrow(ords) ^= popsize) then do;
        file log;
        put "SAMPLE ERROR: Order vector not 
same size as population size";
        return;
    end;
    popu = popu || ords;
    popu = sortmat(popu,2)[,1];
end;

if (sum(pils) - sampsize > 0.5) then do;
    file log;
    put "SAMPLE ERROR: sum of inclusion probabilities 
not equal to sample size.";
    return;
end;
if (any(pils) > 1) then do;
    file log;
    put "SAMPLE ERROR: First order inclusion probabilities 
not scaled. At least one > 1.";
    return;
end;
if swords = 0 then */
    do;
        /* Randomly sort the population indices */
        /* NOTE: This uses the smpseed random number sequence that 
should be set outside of this routine. */
        m = j(popsize,1,0);
call ranuni(smpseed,m);

        /* This is the old routine that used sortmat() 
        popu = popu || pils || m; 
        popu = sortmat(popu,3); */
        pils0 = pils[rank(m)];
        popu = popu[rank(m)];
end;
/* construct the pointer vector */
m = 0;
call ranuni(smpseed,m);
pv = m + (0:(sampsize -1)');
/* compute cumulative sum of pi_1 vector */
cupi = cusum(pi1s0);
/* find all indices in pi(i) such that
   pi(i) < pv[j] < pi[i+1] for some j */
cupi1 = shape(cupi1, sampsize, popsize)';
pv = shape(pv, popsize, sampsize);
pop_ind = (pv > cupi1)[+,] + 1;
/* extract unit indices from population */
ss = popu[pop_ind];
finish;
reset storage =simprocs.sprocs;
store module = sample;
quit;

/* ################################################################### */
/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* IPROBS.SAS - IML module that calculates the first- and second-order inclusion probabilities for fixed sample size randomized VRS designs. 
   Pip Courbois 9/22/99 */
/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* iprobs.sas */
INPUTS
   x - auxiliary variable that the sampling will be based. Must be the same length as the popn.
   swVRS - switch for VRS = 1 or SRS = 0.
   n - sample size.
RETURNS
   pi1 - 1st order inclusion probabilities.
   pi20 - 2nd order inclusion probabilities
         0 approximation.
   pi2hr - 2nd order inclusion probabilities
          hr approximation.
*/
%include "librefs.sas";
proc iml;
start iprobs(pi1, pi20, pi2hr, x, swvrs, n);
bigN = nrow(x);
if n > bigN then
do;
file log;
   put "iprob error 1. Sample size is larger than population size."
   return;
end;

if swVRS = 0 then
   do;
      pil = shape(n/bigN, bigN, 1);
      if (bigN = 1) then pi2o = J(bigN, bigN, 0);
      else pi2o = shape(n*(n-1)/(bigN*(bigN-1)), bigN, bigN);
      pi2hr = pi2o;
   end;
else
   do;
      pil = scalepi(n,x);
      /* Overton's approximation */
      call jointo(pi2o, pil, n);
      /* Hartley and Rao's approximation */
      call jointhr(pi2hr, pil, n);
   end;

pi2o = pi2o # (j(bigN, bigN)-i(bigN)) + diag(pil);
pi2hr = pi2hr # (j(bigN, bigN)-i(bigN)) + diag(pil);

finish iprobs;
reset storage = simprocs.sprocs;
store module = iprobs;
quit;

/* ################################################################### */
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */
/* Jointo.sas IML module */
/* Calculate the joint inclusion probabilities for a generalized random sample. This uses the Overton's approximation Pip Courbois 9/22/99 */
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* pi2 is an NxN matrix of second order inclusion probabilities
   pil is the Nx1 vector of first order inclusion probs.
   n sample size */
%include "librefs.sas";
proc iml;
   start jointo(pi2, pil, n);
      nh = sum(pil);
if (n - nh > 0.1) then
  do;
    file log;
    print nh; print n;
    print pil;
    put "Error 1 in jointo.sas";
    return;
  end;

  popsize = nrow(pil);
  ones = J(popsize,popsize,1);

  if (n = 1) then
    pi2 = J(popsize,popsize,0);
  else
    pi2 = (2*(n-1)*pil*pil')/(2*n*ones-
      shape(pil,popsize,popsize)
      -shape(pil,popsize,popsize)');

  finish jointo;

reset storage = simprocs.sprocs;
store module=jointo;
quit;

/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */
/* Calculate the joint inclusion probabilities for a generalized random sample. This uses the Hartley and Rao's O(N^-4) approximation. */
/* PiP Courbois 9/23/99 */
/* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ */

/* pi2 is an N x N matrix of second order inclusion probabilities
   pil is the N x 1 vector of first order inclusion probs.
   n sample size */

@include "librefs.sas";

proc iml;
start jointhr(pi2hr,pil,n);
  if (n - sum(pil) > 0.5) then
    do;
      print n;
      ns=n-sum(pil); print ns;
put "Error 1 in joinHR.sas";
return;
end;

popsize = nrow(pi1);

if (n = 1) then pi2hr = J(popsize,popsize,0);
else
do;

ones = J(popsize,popsize,1);
pisquare = pi1 ## 2;
picube = pi1 ## 3;

n2 = n ## 2; n3 = n ## 3; n4 = n ## 4; n5 = n ## 5;

pi2hr = (n-1)/n * pi1*pi1' + (n-1)/n2 *
(pisquare*pi1` + pi1*pisquare') - (n-1)/n3 * sum(pisquare) * pi1*pi1` + 2*(n-1)/n3 * (picube * pi1` + pi1 * picube') + pisquare * pisquare') - 3*(n-1)/n4 * sum(pisquare) * (pisquare * pi1` + pi1 * pisquare') + 3*(n-1)/n5 * (sum(pisquare)) ## 2 * pi1*pi1` - 2*(n-1)/n4 * sum(picube) * pi1 * pi1`;
end;

finish jointhr;

reset storage = simprocs.sprocs;
store module=jointhr;
quit;

/* ********************************************************** */
/* ----- ---------------------------------------------------*/
/* estVAR2.SAS - IML module to calculate the estimators 
for the variance component in single stage sampling. 
Pip Courbois 9/22/99 
version 2 */
/* ----------------------------------------------------------- */

/* estVAR.SAS.

INPUTS:
s - population level index of sampled units. (population level implies that popvals[s] is the sampled responses).
popvals - the responses.
pi1 - population 1st order inclusion probabilities.
PIPIprime - pi1*pi1';

OUTPUTS:
sigma2 - vector of estimates of variance components. See below for specific estimators.
*/
%include "librefs.sas";

proc iml;
start estVAR(sigma2, s, y, pi1, PIPIprime, pi2o, pi2hr);

counter = {I am here};

bigN = nrow(y);
n = nrow(s);

ys = y[s];
pis = pi1[s];
pi2os = pi2o[s,s];
PIPIs = PIPIprime[s,s];

onepop = J(n,1,1);
Ipop = I(n);
Jpop = J(n,n,1);

/* print counter; */
invpipi = Jpop / PIPIs;
/* print counter; */

/* naive estimator */
s21 = ys` * (Ipop-Jpop/n) * ys / (n-1);

/* estimate of population size */
Nhat = sum(onepop/pis);

/* Overton weighted estimator of population total squared */
NNhat = onepop` * (Ipop / pi2os) * onepop;

/* HT estimator of variance */
S2pi = ys` * ( inv(diag(pis)) - invpipi /bigN ) * ys / (bigN-1);

/* Weighted HT estimator of variance, SSW 5.9.11. */
S2R = ys` * ( inv(diag(pis)) - invpipi /Nhat ) * ys / (Nhat-1);

/* Generalized weighted HT estimator of variance */
S2GR = ys` * ( inv(diag(pis))/(Nhat-1) - invpipi /(NNhat - Nhat) ) * ys ;

/* Method of moments estimator */
S2MOMe = ys` * ( ( Ipop - 1/bigN * Jpop ) / pi2os ) * ys / (bigN-1) ;

/* Weighted Method of moments estimator */
S2MOMRe = ys' * ((Ipop - 1/Nhat * Jpop) / pi2os) * ys / (Nhat-1);

/* Generalized weighted method of moment estimator */
S2MOMGRe = ys' * ((Ipop/(Nhat-1) - Jpop/(NNhat - Nhat)) / pi2os) * ys;

/* Yates Grundy estimator */
sumPij = (Jpop/Pi2os)[,+];
S2YGp = ys' * ((diag(sumPij)) - Jpop / pi2os) * ys;
S2YG = S2YGp / (bigN * (bigN-1));

/* Weighted Yates Grundy estimator */
S2YGR = S2YGp / (Nhat * (Nhat-1));

/* Generalized weighted Yates Grundy estimator */
S2YGR = S2YGp / ((NNhat - Nhat));

/* this will be one row of the results matrix */
sigma2 = S21 || S2pi || S2R || S2GR || S2MOME || S2MOMRe || S2MOMGRe || S2YG || S2YGR || S2YGR;

/* This file calculates the summary statistics of the result matrix.
Here rsigma2 is the result matrix (need to make this a module but how do I pass the filename?) */
*/ 

%macro summary;

meanS = rsigma2[:,(2,3,4,5,6,7,8,9,10,11)];
maxS = rsigma2[<>,(2,3,4,5,6,7,8,9,10,11)];
minS = rsigma2[>>, (2,3,4,5,6,7,8,9,10,11)];
vmat = (rsigma2[,(2,3,4,5,6,7,8,9,10,11]) - (J(&niter,1,1) @ meanS))##2/(&niter-1);
varS = vmat[+,,];
stdS = varS ## (1/2);

estimators =
( naive, HT, HT_R, HT_GR, MOME, MOME_R, MOME_GR, YG, YG_R, YG_GR);
rho = J(ncol(minS),1,&p);
alpha = J(ncol(minS),1,&alpha);
mY = J(ncol(minS),1,&muY);
nn = J(ncol(minS),1,&sampsize);

summat = rho || alpha || mY || nn || minS' || meanS' || maxS' ||
varS' || stdS';

filename sumfile "&simhome\simresults\sumfile";
file sumfile;
do i = 1 to nrow(summat);
put(estimators[i]) $char8. @;
do j = 1 to ncol(summat);
put(summat[i,j]) 8.4 +2 @;
end;
put;
end;
closefile sumfile;

%mend summary;