Workpackage 6
Variance Estimation for Unequal Probability Designs

Deliverable 6.2
List of contributors:
Yves Berger, Chris Skinner, University of Southampton.

Main responsibility:
Yves Berger, University of Southampton.

IST–2000–26057–DACSEIS

The DACSEIS research project is financially supported within the IST programme of the European Commission. Research activities take place in close collaboration with Eurostat.

http://europa.eu.int/comm/eurostat/research/
http://www.cordis.lu/ist/
http://www.dacseis.de/
Preface

Sampling textbooks often refer to the Horvitz and Thompson (1952) and Yates and Grundy (1953) variance estimator for use in the presence of general unequal probability designs. In practice, these variance estimators are often only implemented in special cases, because of the complexity of determining joint inclusion probabilities for general designs. In this workpackage, simpler variance estimators are investigated, which depend only on the first-order inclusion probabilities.

Yves Berger and Chris Skinner

Southampton, October 2002
Contents

1 Introduction ........................................ 1
   1.1 Sampling design ................................ 1

2 A variance estimator for unequal probability sampling with replacement .... 3

3 A general variance estimator for unequal probability sampling without replacement ........................................ 5

4 A variance estimator for Poisson sampling ........................................ 7

5 Variance estimation for fixed size sampling ........................................ 9
   5.1 Variance estimation by a with-replacement approximation ................. 10
   5.2 The Hájek variance estimator ........................................ 10

6 Systematic sampling ........................................ 15
   6.1 The systematic sampling design ........................................ 15
   6.2 Variance estimation under systematic sampling design ................. 16

A S-Plus® Functions ........................................ 19

References ........................................ 29
Chapter 1

Introduction

Variance estimation with unequal probability sampling is common in survey sampling. Unequal probability sampling was first suggested by Hansen and Hurwitz (1943) in the context of with-replacement sampling. Horvitz and Thompson (1952) developed the corresponding theory for without-replacement sampling.

Consider a population $U$ containing $N$ units:

$$U = \{1, \ldots, i, \ldots, N\}.$$

We identify the $i$-th unit of $U$ by its label $i$. In this report, we assume that the parameter of interest is a population total

$$\tau = \sum_{i \in U} y_i,$$

$y_i$ being the value of a study variable for the $i$-th unit. In this report, we focus on the variance estimation of the Horvitz and Thompson (1952) estimator of $\tau$. The variance of the Horvitz-Thompson estimator plays an important role in variance estimation, as most estimators of interest can be linearized as Horvitz-Thompson totals of linearized variables. For example, with the regression estimator (Brewer, 1963), the $y_i$ are residuals of the regression between the study variable and the auxiliary variables (Särndal et al., 1992, p. 234).

1.1 Sampling design

Let $p(s)$ be the probability of selecting a sample $s \subset U$. This probability distribution is called the sampling design. Let $n_s$ denotes the sample size; that is, the number of distinct units of $s$. This size depends on the sampling design and may be fixed (non-random) or not fixed (random). We will use the notation $n$ when the sample size $n_s$ is fixed.

Throughout this report, we suppose that the sampling design is a single stage stratified sampling design with strata $U_1, \ldots, U_H$; where $\bigcup_{h=1}^H U_h = U$. We suppose that a sample $s_h$ of size $n_{sh}$ is selected within each stratum $U_h$ of size $N_h$. We will use the notation
\( n_h \) when the stratum sample size \( n_{sh} \) is fixed. We have \( \bigcup_{h=1}^{H} s_h = s \), \( \sum_{h=1}^{H} n_h = n \) and \( \sum_{h=1}^{H} n_{sh} = n_s \).

In this report, we use the design-based approach. We do not assume any modelling of the study variable. For simplicity, complete response is assumed.
Chapter 2

A variance estimator for unequal probability sampling with replacement

Sampling with unequal probability with replacement was first suggested by Hansen and Hurwitz (1943). In this chapter, we define this sampling design, the Hansen and Hurwitz (1943) point estimator and the Hansen and Hurwitz (1943) variance estimator. Suppose that \{p_1, \ldots, p_N\} are given positive numbers such that

\[ \sum_{i \in U_h} p_i = 1 \quad \text{for } h = 1, \ldots, H. \]

The selection of each sample \(s_h\) is carried out by drawing a first unit in such a way that the probability of drawing unit \(i \in U_h\) is equal to \(p_i\). The same set of probabilities is used to draw the second unit, \ldots and the \(m_h\)-th unit. The \(m_h\) draws are independent and \(s_h\) is the set of distinct units drawn. As each unit can be selected more than once, the strata sample sizes is random and given by \(n_{sh} = \sum_{i \in s_h} 1 \leq m_h\). For a small sampling fraction, \(n_{sh} \approx m_h\).

The Hansen and Hurwitz (1943) estimator

\[ \hat{\tau}_{sur} = \sum_{i \in s} \frac{y_i}{\tilde{\pi}_i} M_i. \] (2.1)

is an unbiased estimator of \(\tau\); where \(\tilde{\pi}_i = m_h p_i\) for \(i \in U_h\) and \(M_i\) is the number of times the unit \(i\) is drawn. The variance of \(\hat{\tau}_{sur}\) is given by

\[ \text{var}_{sur}(\hat{\tau}_{sur}) = \sum_{h=1}^{H} \sum_{i \in U_h} \tilde{\pi}_i \left( \frac{y_i}{\tilde{\pi}_i} - \frac{\tau_h}{m_h} \right)^2 \]

where \(\tau_h = \sum_{i \in U_h} y_i\). An unbiased estimator of \(\text{var}_{sur}(\hat{\tau}_{ysur})\) is given by the Hansen and Hurwitz (1943) variance estimator

\[ \widehat{\text{var}}_{sur}(\hat{\tau}_{sur}) = \sum_{h=1}^{H} \frac{m_h}{m_h - 1} \sum_{i \in s_h} M_i \left( \frac{y_i}{\tilde{\pi}_i} - \frac{\hat{\tau}_h}{m_h} \right)^2 \] (2.2)

where \(\hat{\tau}_h = \sum_{i \in s_h} y_i \tilde{\pi}_i^{-1} M_i\). We see that the variance estimator (2.2) is simple to compute, as it involves simple sums and does not depend on unknown quantities.
Chapter 3

A general variance estimator for unequal probability sampling without replacement

For sampling without replacement, the unbiased Horvitz and Thompson (1952) estimator of $\tau$ is usually preferred. This estimator is defined by

$$\hat{\tau} = \sum_{i \in s} \bar{y}_i,$$  \hspace{1cm} (3.1)

where $s$ is a sample of $U$ and $\bar{y}_i = y_i \pi_i^{-1}$ and where $\pi_i$ is the first-order inclusion probabilities of unit $i$; that is, the probability for unit $i$ to be selected in the sample. We note that the estimators (2.1) and (3.1) give the same point estimates if $M_i = 1$ (for all $i \in s$), $i = \pi_i$, and $m_h = n_h$. This is often the case for small sampling fractions $f_h = n_h/N_h$.

The sampling variance of $\hat{\tau}$ is given by

$$\text{var}(\hat{\tau}) = \sum_{i \in U} \sum_{j \in U} (\pi_{ij} - \pi_i \pi_j) \bar{y}_i \bar{y}_j.$$  \hspace{1cm} (3.2)

The quantities $\pi_{ij}$ are the joint inclusion probabilities of unit $i$ and $j$; that is, the probability that both units $i$ and $j$ are selected. An unbiased design-based estimator of var($\hat{\tau}$) is given by the unbiased Yates-Grundy (1953) estimator

$$\widehat{\text{var}}(\hat{\tau}) = \frac{1}{2} \sum_{i \in s} \sum_{j \in s} \left( \frac{\pi_i \pi_j - \pi_{ij}}{\pi_{ij}} \right) (\bar{y}_i - \bar{y}_j)^2.$$  \hspace{1cm} (3.3)

This estimator can be implemented with any sampling design, as long as the $\pi_{ij}$ are known. However, these probabilities are often unknown except for special cases such as stratified simple random sampling.

Chapter 5 shows that the actual computation of the $\pi_{ij}$ is not really necessary for variance estimation. Nevertheless, the $\pi_{ij}$ are not totally irrelevant, as analytic expressions $i \in U_h$ for the $\pi_{ij}$ might be useful when we have to chose a $\pi$s sampling design. For example, we may want the $\pi_{ij}$ to be such that $\pi_{ij} > 0$ and $\pi_{ij} \leq \pi_i \pi_j$ ($i \neq j$). The first property guarantees an unbiased variance estimator and the second one a positive variance estimator. However, finding a suitable sampling design and finding a variance estimator are two different issues.
Chapter 4

A variance estimator for Poisson sampling

The Poisson sampling design is defined as follows. Let $\epsilon_1, \ldots, \epsilon_N$ be independent random numbers drawn from the uniform distribution $U(0,1)$. If $\epsilon_i < \pi_i$, unit $i$ is selected, otherwise not ($i = 1, \ldots, N$). The resulting sample is a Poisson sample. Thus, the Poisson sampling design is such that the probability of selecting a sample $s$ is

$$P(s) = \prod_{i \in s} \pi_i \prod_{j \notin s} (1 - \pi_j). \quad (4.1)$$

The first-order inclusion probabilities of the Poisson sampling design are given by $\pi_i$. It is worth noticing that stratification is only used to compute the $\pi_i$. The stratification information is not necessary for the selection of a Poisson sample, as the sampling design consisting of selecting Poisson samples in each stratum is given by (4.1).

As each unit of a Poisson sample is selected independently,

$$\pi_{ij} = \pi_i \pi_j$$

for any $i \neq j$. By plugging these $\pi_{ij}$ in (3.2) and in (3.3), we obtain the following variance and variance estimator

$$\text{var}_{\text{Poiss}}(\hat{\tau}) = \sum_{i \in \mathcal{U}} \pi_i (1 - \pi_i) \hat{y}_i^2$$

$$\overline{\text{var}}_{\text{Poiss}}(\hat{\tau}) = \sum_{i \in s} (1 - \pi_i) \hat{y}_i^2. \quad (4.2)$$

The information about the strata is not necessary for variance estimation. We see that (4.2) is simple to compute.

Under Poisson sampling, the sample size $n_s$ is random, with expectation

$$E(n_s) = \sum_{i \in \mathcal{U}} \pi_i.$$
It is easy to show that the variance expression in (3.2) for any unequal probability sampling design without replacement can be expresses alternatively as

$$\text{var}(\hat{\tau}) = \sum_{i \in U} \pi_i (1 - \pi_i) \hat{y}_i^2 - \sum_{i \in U} \sum_{j \in U, j \neq i} (\pi_i \pi_j - \pi_{ij}) \hat{y}_i \hat{y}_j.$$ 

Thus, if $\pi_{ij} < \pi_i \pi_j$, for all $i$ and $j$ then

$$\text{var}_{\text{Poisson}}(\hat{\tau}) \geq \text{var}(\hat{\tau}). \quad (4.3)$$

This means that the Poisson sampling design is less accurate than sampling designs for which $\pi_{ij} < \pi_i \pi_j$, such as fixed size sampling design. Thus the random size decreases the accuracy of $\hat{\tau}$. This is the reason why the Poisson sampling design is considered to be an inefficient sampling design.
Chapter 5

Variance estimation for fixed size sampling

Practitioners avoid designs in which the sample size varies extensively, like Poisson designs. One reason is that variable sample size will cause an increase in variance for certain type of estimator (see (4.3)). More importantly, survey statisticians dislike being in a situation where the number of observations is highly unpredictable when the survey is planned. Furthermore, it is common practice to sample without-replacement to increase the efficiency of the estimators compared to sampling with-replacement (Gabler, 1984) and to avoid selecting the same unit more than once.

A πps sampling design is one where the inclusion probabilities π_i are specified to be proportional to a given size variable (πps denotes π_i proportional to size) and where the sample size is fixed, so that whenever p(s) > 0, the sample s contain a fixed number of distinct units. Hence n_s = n. That is, a sample is selectable under a πps sampling design only if its size is exactly n. But not all samples of size n need to be selectable to have a πps sampling design. This is the case for systematic sampling (see Chapter 6). There are several methods for drawing a πps sample. Brewer and Hanif (1980, 1983) provide a long list of πps samplings. Deville and Tillé (1998) provide a constructive method for generating πps samples based on a splitting procedure.

In this chapter, we address the issue of variance estimation with fixed size unequal probability sampling without-replacement (πps sampling).

Unequal probability sampling was first suggested by Hansen and Hurwitz (1943) in the context of with-replacement sampling. Horvitz and Thompson (1952) developed the corresponding theory for πps sampling. However it was not until 1984 that Gabler (1984) proposed a sufficient condition for πps sampling to be better than with-replacement sampling. This condition is fulfilled by the Deville and Tillé (1998) class of πps sampling designs. As most πps sampling designs used in practice belong to this class, sampling without-replacement is usually better than sampling with-replacement. This is particularly true for large stratum sampling fractions f_h = n_h/N_h which are common in business surveys.

Variance estimation for sampling with-replacement is straightforward (see (2.2)) and very easy to implement. However, for πps sampling, the variance estimator in (3.3) is hard
to compute because of the joint inclusion probabilities $\pi_{ij}$. Although exact computation of the $\pi_{ij}$ is possible for some sampling designs like the CHAO (1982) sampling design, the calculation of the $\pi_{ij}$ becomes practically impossible when the sample size is large. Moreover, standard statistical packages like SPSS, SAS, STATA do not deal with the $\pi_{ij}$. Specialized software like Sudaan needs to be used. However, even specialized softwares do not include actual computation of these probabilities. They often need to be specified by the user. Generally, it is inconceivable to provide the $\pi_{ij}$ in a sample data file. Thus, in practice, the $\pi_{ij}$ are almost never available for variance estimation.

The double sum feature in (3.3) is computationally inconvenient for large samples. A solution might be to implement a $\pi$ps sampling design that provides easy computation of the $\pi_{ij}$. This can be done with the TILLÉ (1996) sampling design. Unfortunately, even with this design, the computation of the $\pi_{ij}$ requires the values of the $\pi_i$ for all the units of the population, whereas it is common to know the value of $\pi_i$ only for the sampled units. In this case, the $\pi_{ij}$ cannot be computed. There are alternative methods like replication methods that do not involve the $\pi_{ij}$. We refer to SMITH (2001) for an overview of possible methods of variance estimation. In the next section, we show that the variance can be easily estimated without computing the $\pi_{ij}$.

5.1 Variance estimation by a with-replacement approximation

Variance estimation is greatly simplified by treating the sample as if the units were sampled with replacement. In this case, expression (2.2) may be used after replacing $\tilde{\pi}_i$ by $\pi_i$, $M_i$ by 1 and $m_h$ by $n_h$. This approach is often adopted in practice. However this approximation leads to overestimation of the variance. This positive bias is particularly large for large sampling fractions. Thus, we need an estimator that takes the effect of the sampling fraction into account and which is as simple as the with-replacement variance estimator.

5.2 The Hájek variance estimator

In this section, the results of BERGER (2004) are presented. The aim is to show that it is possible to estimate the variance without computing $\pi_{ij}$ with the HÁJEK (1964) variance estimator. The aim of this section is to show the practical interest of this estimator and how this estimator can be easily implemented with weighted least squares regression, which makes it easy to implement with standard statistical packages. We give an alternative expression for the Hájek estimator, as a weighted sum of residuals. This alternative expression is computationally simpler than the Yates-Grundy (1953) variance estimator (3.3) and does not require computation of $\pi_{ij}$. Moreover, it leads to values close to the Yates-Grundy variance estimates (BERGER, 2004). We will see that this alternative expression reveals the main effects of the sampling design on the variance: the stratification effect, the unequal probabilities effect, the finite population correction effect and the effect of the correction of degree of freedom. This allows us to quantify the impact of these effects on the variance.
HÁJEK (1981) proposed the following approximation for the $\pi_{ij}$ of $\pi_{ps}$ stratified sampling design.

$$
\pi_{ij} \approx \pi_i \pi_j \left[1 - N_h (N_h - 1)^{-1} (1 - \pi_i)(1 - \pi_j) d_h^{-1}\right]
$$

(5.1)

when $i, j \in U_h$ and where

$$
d_h = \sum_{i \in U_h} \pi_i (1 - \pi_i).
$$

If we replace $\pi_i$ by $f_h = n_h N_h^{-1}$ ($i \in U_h$) in (5.1), we get the $\pi_{ij}$ of the simple stratified sampling (STSRS) design. The approximation (5.1) is therefore exact for STSRS. HÁJEK (1964, 1981) shows that this approximation is valid when rejective sampling is implemented in each stratum. BERGER (1998) shows that this approximation can be used for a larger class of highly randomised sampling designs such as rejective (HÁJEK 1964), successive (HÁJEK, 1964) and Rao-Sampford (RAO, 1965 and SAMPFORD, 1967) sampling designs.

In order to derive a variance estimator, we can replace in (3.3), the $\pi_{ij}$ given by (5.1). However, this does not give a suitable estimator, as the approximation (5.1) and the double sum in (3.3) give unstable variance estimates. This is due to the fact that $\hat{\text{var}}(\hat{\tau})$ is a double sum containing $n^2$ term that estimates the population double sum (3.2) containing $N^2$ terms. If we see $\hat{\text{var}}(\hat{\tau})$ as a total, the estimator $\hat{\text{var}}(\hat{\tau})$ is based on sampling fraction $n^2 N^{-2} = f^2$ which is too small to give stable estimates. Thus, such an estimator involving an approximation of the $\pi_{ij}$ can be unstable. In this section, we propose a better option that consists on estimating the approximation of the variance obtained by replacing (5.1) in (3.2).

By substituting (5.1) into (3.2), we get the following approximation of the variance

$$
\text{var}_{Haj}(\hat{\tau}) = \sum_{h=1}^{H} N_h (N_h - 1)^{-1} \sum_{i \in U_h} \pi_i (1 - \pi_i) (\bar{y}_i - \beta_h)^2
$$

where

$$
\beta_h = d_h^{-1} \sum_{i \in U_h} \pi_i (1 - \pi_i) \bar{y}_i.
$$

As (5.1) is exact for STSRS, $\text{var}_{Haj}(\hat{\tau})$ is the usual variance for STSRS design when $\pi_i = f_h$ ($i \in U_h$).

An alternative expression for $\beta_h$ is the regression coefficient

$$
\beta_h = \left(\sum_{j \in U} c_j z_{jh}^2\right)^{-1} \sum_{i \in U} c_i \bar{y}_i z_{ih}
$$

where $c_i = N_h (N_h - 1)^{-1} \pi_i (1 - \pi_i)$ ($i \in U_h$) and where $z_{ih} = 1$ if $i \in U_h$ and otherwise $z_{ih} = 0$. The $z_{ih}$ are the indicators for the strata. This leads to the following alternative expression for $\text{var}_{Haj}(\hat{\tau})$.

$$
\text{var}_{Haj}(\hat{\tau}) = \sum_{i \in U} c_i^2 e_i^2;
$$

(5.2)
where \( e_i = \hat{y}_i - \beta_h \) \((i \in U_h)\). From the definition of \( z_{ih} \), we may re-write \( e_i \) as

\[
e_i = \hat{y}_i - \sum_{h=1}^{H} \beta_h z_{ih};
\]

that is, the \( e_i \) are the weighted least square residuals of the working regression

\[
\hat{y}_i = \sum_{h=1}^{H} \beta_h z_{ih} + e_i;
\] (5.3)

The term “working regression” is used to emphasize the fact that we do not assume that the model behind (5.3) is true. This regression is just used to produce the residuals \( e_i \). The fact that the \( e_i \) can be interpreted as residuals is a consequence of approximation (5.1) of the joint-inclusion probabilities. Even if the \( e_i \) can be interpreted as the residuals of a super-population model, (5.3) is not necessarily a super-population model assumed to be true. It does not matter if this model is wrong or not. If the model is wrong, \( \text{var}_{H} (\hat{\tau}) \) is still a correct approximation of the variance.

A natural estimator of \( \text{var}_{H} (\hat{\tau}) \) is given by

\[
\hat{\text{var}}_{H} (\hat{\tau}) = \sum_{i \in s} \hat{c}_i \hat{e}_i^2; \tag{5.4}
\]

where \( \hat{c}_i = n_h (n_h - 1)^{-1} (1 - \pi_i) \) with \( i \in U_h \) and where the \( \hat{e}_i \) are the weighted least square residuals

\[
\hat{e}_i = \hat{y}_i - \sum_{h=1}^{H} \hat{\beta}_h z_{ih}
\]

where

\[
\hat{\beta}_h = \left( \sum_{j \in s} \hat{c}_j z_{jh}^2 \right)^{-1} \sum_{i \in s} \hat{c}_i \hat{y}_i z_{ih}.
\]

A series of simulations in BERGER (2004) shows that (5.4) is an accurate variance estimator.

In \( \hat{c}_i \), we prefer to use \( n_h (n_h - 1)^{-1} \) instead of \( N_h (N_h - 1)^{-1} \), as in this case, \( \hat{\text{var}}_{H} (\hat{\tau}) \) gives the usual stratum by stratum variance estimator for STSRS (BERGER, 1998). Thus, when the units are selected with equal probabilities in each stratum, \( \hat{\text{var}}_{H} (\hat{\tau}) = \hat{\text{var}} (\hat{\tau}) \). Therefore, \( \hat{\text{var}}_{H} (\hat{\tau}) \approx \hat{\text{var}} (\hat{\tau}) \) if units are selected with different probabilities.

If we approximate \( n_h (n_h - 1)^{-1} \) by 1, (5.4) is the Hájek variance estimator (see Formula (8.12) in HÁJEK, 1964). Thus, (5.4) is not exactly the Hájek variance estimator. However in order to show the practical value of the Hájek variance estimator, we prefer to write the estimator in the more convenient form (5.4). HÁJEK (1964) considers the single stratum case \((H = 1)\). For more than one stratum, by using the Formula (8.12) in HÁJEK (1964) within each stratum, we obtain a formula close to (5.4). It is worth noticing that if \( \hat{c}_i = (1 - \pi_i) \log(1 - \pi_i) \pi_i^{-1} \), (5.4) is the Rösén (1991) estimator which is similar to the Hájek estimator.
In practice, (5.4) is a simple estimator to compute, as it does not require the \( \pi_{ij} \). As we know in which stratum each units belongs, it is easy to specify the set of stratification variables \( z_{ih} \). As \( \hat{\beta}_h \) is the usual weighted least squares estimate of a regression coefficient weighted by the \( \hat{c}_i \), any standard statistical package can be used to compute the \( \hat{\beta}_h \) and the set of residuals \( \hat{e}_i \). The variance \( \hat{\text{var}}_{\text{Haj}}(\hat{\tau}) \) is just a weighted sum of these residuals. The merit of this method is the fact that the variance estimator is only computed through a set of residuals and only requires the values of \( \pi_i \) for the sampled units.

If a small number of units are selected per stratum, the \( \pi_i \) are small and \( (1 - \pi_i) \) is close to 1. If we replace \( (1 - \pi_i) \) by 1 in (5.4), we get \( \hat{\text{var}}_{\text{swr}}(\hat{\tau}_{\text{swr}}) \) defined by (2.2) with \( \tilde{\pi}_i = \pi_i \), \( M_i = 1 \) and \( n_{sh} = n_h \). This is natural as when few units are selected per stratum, there is not a significant difference between sampling with or without replacement. We note that \( \hat{\text{var}}_{\text{Haj}}(\hat{\tau}) \) is as simple as \( \hat{\text{var}}_{\text{swr}}(\hat{\tau}) \) to compute, as it involves singles sums and does not depend on unknown quantities.

Other effects of the sampling design are also included in (5.4). The effect of stratification is included in the residuals \( \hat{e}_i \) as the working regression (5.3) uses the stratification variables as regressors. This is the major effect of the sampling design. The effect of the \( \pi_i \) is also included in the residuals as the independent variables \( \tilde{y}_i \) in (5.3) is the study variable \( y_i \) divided by \( \pi_i \). The estimator \( \hat{\text{var}}_{\text{Haj}}(\hat{\tau}) \) splits these effects into the residuals and the \( \hat{c}_i \) and therefore provides an elegant interpretation of these effects. The effects of most \( \pi ps \) sampling designs can be split this way. This is the reason why we expect a similar variance for \( \pi ps \) sampling designs which have the same set of \( \pi_i \).

There is a remainder effect that is not included in (5.4). The remainder effect explains the slight differences between (3.3) and (5.4) due to the method of sampling used. This remainder effect is negligible for highly randomised sampling designs (BERGER, 1998). Systematic sampling is not highly randomised because most of the fixed size samples cannot be drawn. Nevertheless, in Chapter 6, we show that the estimator (5.4) can still be used if we include in the working model (5.3), additional regressors that takes the systematic sampling into account.

Although \( \hat{\text{var}}_{\text{Haj}}(\hat{\tau}) \) is applicable under single stage stratified sampling, it can be generalized to multi-stage sampling. For two stage, the variance is usually estimated by a variance between the primary sampling units (PSU), as such variance estimator has a positive bias that incorporates the within PSU variance (SKINNER et al., 1989). Thus, for two stage, (5.4) can be implemented with \( i \) representing the PSU label and \( y_i \) an estimate of the total of the \( i \)-th PSU.
Chapter 6

Systematic sampling

The systematic sampling (SYS) design is widely used by statistical offices due to its simplicity and its efficiency (Bellhouse and Rao, 1975). This sampling design has been studied since the early years of survey sampling: Madow and Madow (1944), Madow (1949), Cochran (1946). An overview of the literature can be found in Iachan (1982). In this chapter, a variance estimator for the Horvitz and Thompson (1952) estimator is proposed, when the sample is selected by a SYS design with unequal probabilities.

Under SYS, there is no unbiased estimator of the design variance (Cassel et al., 1977, page 68). A partial solution is to assume that the units are arranged randomly, giving to each permutation the same probability, the random ordering being part of the sampling design. This sampling design is called the randomised SYS (RSYS) design, suggested first by Madow (1949). Under RSYS, it is possible to derive a variance estimator with a small bias (Hartley and Rao, 1962; Hájek, 1981). However, these estimators are inaccurate under SYS, as the variance under RSYS is generally larger than the variance under SYS (Särndal et al., 1992, page 81) especially when there is a trend in the survey variable (Bellhouse and Rao, 1975). In fact, SYS is particularly efficient if there is a trend, as a trend tends to decrease the variance. Unfortunately, this reduction in the accuracy is difficult to estimate. This is due to the fact that the more accurate sampling design is, the more difficult the variance estimation is.

However, in many practical situations there are good reasons for letting the population have a predetermined (non-random) order. For example, the population can be sorted by a size variable, by regions, socio-economic group and postal sectors or in some other ways. Our concern is the variance estimation in these situations. In this chapter, we do not assume RSYS. Instead we assume a predetermined (non-random) order of the population. We will use a methodology based on Hájek (1981) (see Section 5.2). We show that (5.4) can be generalized to SYS.

6.1 The systematic sampling design

For simplicity, suppose that the population is composed of a single stratum. Consider a finite population \( U = \{1, \ldots, k, \ldots, N\} \) of \( N \) units in a given order. Any order can be
considered. A systematic πps sample (SYS sample) is defined by
\[ s = \left\{ i \in U : \pi_i^{(c)} < m_j \leq \pi_i^{(c)}; \ j = 1, \ldots, n \right\} \quad (6.1) \]

where
\[ m_j = u + j - 1, \quad (6.2) \]
\[ \pi_i^{(c)} = \sum_{j \in U; j \leq i} \pi_j, \quad (6.3) \]

\( \pi_0^{(c)} = 0 \) and \( u \) is a random number from a uniform distribution \( U(0, 1) \). We assume the \( \pi_i \) known for all \( i \in U \). Several SYS samples can be selected depending on the value of \( u \). As \( u \) is random, the sample is therefore a random variable. The SYS design is the probability distribution of \( s \). Systematic samples contain \( n \) or \( n - 1 \) units, where \( n = \sum_{i=1}^{N} \pi_i \). The SYS design is therefore not a fixed size sampling design. The methodology proposed will take this feature into account.

6.2 Variance estimation under systematic sampling design

In this section, we introduce briefly Berger’s (2003) results without any proofs.

Consider the set of variables
\[ a_{iq} = \tilde{\pi}_i^{(c)}(I\{i \in G_q\} - I\{i \in G_1\}) \quad \text{with} \quad q = 1, \ldots, n - 1 \]

where \( I\{\Psi\} = 1 \) if \( \Psi \) is true and \( I\{\Psi\} = 0 \) otherwise. The groups \( G_q \) are defined by
\[ G_q = \left\{ i \in U : \tilde{\pi}_i^{(c)} \leq q \text{ and } \tilde{\pi}_i^{(c)} > q - 1 \right\} \]
\[ \tilde{\pi}_k^{(c)} = \pi_k^{(c)} - \pi_k/2 \]

Consider the variables
\[ x_{i1} = 1 \]
\[ x_{i2} = \sum_{q=1}^{n-1} a_{iq} \]

for \( i = 1, \ldots, n \). BERGER (2003) shows that the following modified version of (5.4) can be used for systematic sampling:
\[ \hat{\text{var}}_{\text{sys}}(\hat{\tau}) = \sum_{i \in s} \frac{C_i}{\pi_i^{(c)^2}} \frac{x_{i2}^2}{\pi_{\text{sys};i}}, \quad (6.4) \]
where the $\hat{e}_{sys,i}$ are the sample residuals

$$
\hat{e}_{sys,i} = \frac{y_i}{\pi_i} - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2}.
$$

where

$$
\hat{\beta}_p = \left( \sum_{i \in s} \frac{c_i}{\pi_i^2} y_k x_{kp} \right) \left( \sum_{j \in s} \frac{c_j}{\pi_j} x_{jp}^2 \right)^{-1} \quad p = 1, 2.
$$

and the $c_i$ are defined in BERGER (2003). A method to compute the $c_i$ in (6.4) is proposed in BERGER (2003). A S-Plus® function that computes these $c_i$ is given in Appendix (see function COMPUTE.Vect.c()). A series of simulation in BERGER (2003) shows that (6.4) is more accurate than (5.4).
Appendix A

S-Plus® Functions

In this appendix, we have a series of functions that compute the estimates of variance introduced in this report.

- **VARIANCE.HANSEN()**: This function computes the Hansen and Hurwitz (1943) variance estimator in (2.2).
- **VARIANCE.HAJEK()**: This function computes the Hájek (1964) variance estimator in (5.4) using Berger (2004) method.
- **VARIANCE.SYST()**: This function computes the Berger (2003) variance estimator in (6.4) for a pps systematic sampling design.

```r
# FUNCTION VARIANCE.HANSEN
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
# This function computes the Hansen and Hurwitz (1943) variance estimator
# INPUT:
#   * Vect.Y: a nx1 vector that gives the value of the study variable.
#   * Vect.Pi: the nx1 vector containing the first-order inclusion probabilities.
#   * Stratum.Indicator: a nx1 vector describing the strata. This vector contains the labels specifying the strata.
# OUTPUT:
#   * The variance estimate.
# REMARK:
#   * We must have more than one unit per stratum.
#   * There is no finite population correction.

VARIANCE.HANSEN <- function(Vect.Y, Vect.Pi, Stratum.Indicator) {
  Vect.Y.Breve <- Vect.Y / Vect.Pi
  # other code...
}
```
Appendix A. S-Plus® Functions

# FIND THE LABEL USED FOR THE STRATA
Table <- table(Stratum.Indicator)
Set.Of.Modalities <- dimnames(Table)[[1]]
model(Set.Of.Modalities) <- 'numeric'
NB.Strata <- as.numeric(length(Set.Of.Modalities))

# COMPUTE THE ESTIMATES OF VARIANCE
Est.Variance <- 0
for(h in 1:NB.Strata)){

# COMPUTE THE ESTIMATES OF VARIANCE in STRATUM h
Stratum.Label <- Set.Of.Modalities[h]
In.Stratum <- (Stratum.Indicator==Stratum.Label)
Sample.Stratumize.h <- sum(as.numeric(In.Stratum))
Vect.Y.Breve.h <- Vect.Y.Breve[In.Stratum]
Mean.h <- sum(Vect.Y.Breve.h)/Sample.Stratumize.h
Variance.h <- (Sample.Stratumize.h/(Sample.Stratumize.h-1)) * sum((Vect.Y.Breve.h - Mean.h)^2)
Est.Variance <- Est.Variance + Variance.h

if(Sample.Stratumize.h <= 1){
cat("ERROR: A stratum contain only one sampled unit.\n")
}
}

# OUTPUT
Est.Variance

# FUNCTION VARIANCE.HAJEK
#
# This function computes the citehajek64 variance estimator
#
# INPUT:
# * Vect.Y: a nx1 vector that gives the value of the study variable.
# * Vect.Pi: a nx1 vector containing the first-order inclusion probabilities.
# * Stratum.Indicator: a nx1 vector describing the strata.
# This variable contains the labels specifying the strata.
#
# OUTPUT:
# * The variance estimate.
#
# REMARK:
# * We must have more than one unit per stratum.
#
VARIANCE.HAJEK <- function(Vect.Y, Vect.Pi, Stratum.Indicator){
Sample.Size <- length(Vect.Y)

© http://www.DACBEIS.de
Vect.Y.Breve <- Vect.Y / Vect.Pi

# FIND THE LABEL USED FOR THE STRATA
Table <- table(Stratum.Indicator)
Set.Of.Modalities <- dimnames(Table)[[1]]
mode(Set.Of.Modalities) <- "numeric"
NB.Strata <- as.numeric(length(Set.Of.Modalities))

# CREATE THE MATRIX OF DESIGN VARIABLES (STRATIFICATION)
Mat.Design <- matrix(rep(0, times=NB.Strata*Sample.Size),
                      ncol=NB.Strata, nrow=Sample.Size)
for(j in 1:NB.Strata){
  Modalities <- Set.Of.Modalities[j]
  Mat.Design[,j] <- as.numeric(Stratum.Indicator == Modalities)
}

# COMPUTE VARIANCE ESTIMATE
Weights <- as.vector(1 - Vect.Pi)
Est.Variance <- sum.square.weighted.residuals(Vect.Y.Breve,
                                            Mat.Design, Weights, Weights)

# OUTPUT
  Est.Variance

# # FUNCTION VARIANCE.SYST
# #
# # This function computes the Berger (2003) variance estimator
# # for a pps systematic sampling design.
# # INPUT:
# #  * Vect.Y: a nx1 vector that gives the value of the study variable.
# #  * Vect.Pi: a nx1 vector containing the first-order
# #    inclusion probabilities.
# #  * Sample: a N1 vector containing the information about the
# #    sample: the i-th component of Sample is equal to 1 if the i-th
# #    unit is selected, otherwise, the i-th component is equal to 0.
# # OUTPUT:
# #  * The variance estimate.
# # IMPORTANT REMARKS
# #  * We must initialise the function 'VARIANCE.SYST()' by
# #    the following two commands:
# #    Mat.Constraints <- INIT.VAR.SYST(Vect.Pi, Sample.Size,
# #                                       NB.Eigen.Vector)
# #    Vect.c <- COMPUTE.Vect.c(Vect.Pi, Tolerance)
Appendix A. S-Plus® Functions

# * 'INIT.VAR.SYST' must be called before 'COMPUTE.Vect.c'.
# * If there are several estimation of the variance based
# on different samples. One call of 'INIT.VAR.SYST' and
# 'COMPUTE.Vect.c' is necessary before the selection
# of the first sample.
# * We need to call these functions if one or more of the
# following parameters change: the sample size,
# the population size, the first-order inclusion
# probabilities and the number of eigen value.
# * NB. Eigen.Vector can be set equal to zero
# (see Berger 2003).
#
VARIANCE.SYST <- function(Vect.Y, Vect.Pi, Sample) {
  Sample.Size <- length(Vect.Y)
  Vect.Y.Breve <- Vect.Y / Vect.Pi
  Sample.Logical <- (Sample == 1)
  Mat.Constraints.s <- Mat.Constraints[Sample.Logical ,]
  # COMPUTE VARIANCE ESTIMATE
  Weights <- Vect.c[Sample.Logical]/ Vect.Pi
  Est.Variance <- SUM.SQUARE.WEIGHTED.RESIDUALS(Vect.Y.Breve,
                                               Mat.Constraints.s,
                                               Weights)
  # OUTPUT
  Est.Variance
}

#
# FUNCTION INIT.VAR.SYST
# = = = = = = = = = = = = = = = = = = = = = =
#
# This is the first initialisation function for 'VARIANCE.SYST()'.
#
# INPUT:
# * Sample.Size: the sample size, n.
# * Pop.Size: the population size, N.
# * Vect.Pi: a Nz1 vector containing the first-order
# inclusion probabilities.
# * NB. Eigen.Vector: the number of eigen vectors.
#
INIT.VAR.SYST <- function(Vect.Pi, Sample.Size, NB.Eigen.Vector) {
  Pop.Size <- length(Vect.Pi)
  # COMPUTE THE VECTOR OF THE CUMULATIVE
  # FIRST-ORDER INCLUSION PROBABILITIES (Cum.Vect.Pi)
}
for (i in 2:(Pop.Size+1)){
}

K <- 0

    K <- K + 1
}

# CREATION OF THE DIFFERENT STRATA
Cumulative <- Cum.Vect.Pi[2:(Pop.Size+1)]
Transpose.Mat.Strata <- matrix(rep(0, times=Sample.Size*Pop.Size),
                                Sample.Size, Pop.Size)

Maximum.Value <- 0
K <- 1
L <- 0

for (i in (1:Pop.Size)){
    if (Cumulative[i]>Maximum.Value){
        In.The.Previous.Stratum <- Maximum.Value - Cumulative[i-1]

        if (In.The.Next.Stratum >= In.The.Previous.Stratum){
            Maximum.Value <- Maximum.Value + 1
            K <- K + 1
        }
    }
    L <- L + 1
    Transpose.Mat.Strata[K,L] <- 1
}

# CREATION OF THE VECTOR WITH 0 AND CUMULATIVES
Cumulative.for.Mat.Constraints <- Cumulative - 0.5 * Vect.Pi
Mat.Diag.Cumulative <- diag(Cumulative)
Mat.Differences <- matrix(rep(0, times=(Sample.Size-1)*Sample.Size),
                          Sample.Size-1, Sample.Size)

for (i in (1:(Sample.Size-1))){
    Mat.Differences[i,(i+1)] <- 1
    Mat.Differences[i,1] <- -1
}

# CREATION OF THE TRANSPOSE OF THE CONSTRAINT MATRIX

# CREATION OF MATRIX SUM DIFFERENCES
Mat.Sum.Diff <- matrix(rep(1, times=(Sample.Size-1)), nrow=1, ncol=(Sample.Size-1))

# CREATION OF THE CONSTRAINT VARIABLE WITH THE SUM METHOD
Appendix A. S-Plus® Functions


# CREATION OF THE CONSTRAINT VARIABLES WITH EIGEN VALUES METHOD
if(NB.Eigen.Vector>0){

# COMPUTE THE EIGEN VALUES AND EIGEN VECTORS
  Mat.Constraints <- t(Transpose.Mat.Constraints)  
  Mat.C.Pi <- diag(as.vector(Vect.Pi*(1-Vect.Pi)))  
  Var.Cov.A <- Transpose.Mat.Constraints %*% Mat.C.Pi  
  Mat.Constraints  
  Eigen <- eigen(Var.Cov.A)  
  Eigen.Values <- Eigen$values  
  Eigen.Values <- Eigen.Values /sum(Eigen.Values)  
  Eigen.Vectors <- Eigen$vecors

# SORT BY THE EIGEN VALUES (DECREASE ORDER)
  Order.Eigen.Values <- order((-1)*Eigen.Values)  

# CHOOSE THE FIRST NB.Eigen.Vector EIGEN VECTORS
  Mat.Constraints <- Mat.Constraints %*% Eigen.Vectors

# ADD THE VARIABLE RELATED TO THE SUM METHOD
  Mat.Constraints <- as.matrix(cbind(Mat.Constraints.Sum,Mat.Constraints))
  if(NB.Eigen.Vector <= 0){  
    Mat.Constraints <- Mat.Constraints.Sum
  }

# ADD THE FIXED SIZE CONSTRAINT
  Vect.1.U <- matrix(rep(1,times=Pop.Size),ncol=1,nrow=Pop.Size)  
  Mat.Constraints <- as.matrix(cbind(Vect.1.U,Mat.Constraints))

# OUTPUT  
  Mat.Constraints

# # FUNCTION COMPUTE.Vect.c
# # ===========
# # This is the first initialisation function for "VARIANCE.SYST()".
# # INPUT:
# #   * Vect.Pi: a N1 vector containing the first-order inclusion probabilities.

© http://www.DAC5E5.de
# * Tolerance: the tolerance for the convergence.
# OUTPUT:
# The vector Vect.c
#
COMPUTE. Vect.c <- function(Vect.Pi, Tolerance){
  Pop.Size <- length(Vect.Pi)

# INITIALISATION FOR COMPUTATION OF VECTOR Vect.c
  Mat.Delta <- Mat.Joint - Vect.Pi \%\%t(Vect.Pi)
  Mat.C.Pi <- diag(as.vector(Vect.Pi*(1-Vect.Pi))
  Vect.1.U <- matrix(rep(1, times=Pop.Size), ncol=1, nrow=Pop.Size)
  Mat.C.U <- Mat.C.Pi
  Identity.U <- diag(as.vector(Vect.1.U))
  Transpose.Mat.Constraints <- t(Mat.Constraints)

# COMPUTE VECTOR Vect.c
  Found <- FALSE
  Number.Of.Iterations <- 0
  while(!Found){
    Number.Of.Iterations <- Number.Of.Iterations + 1
    Old.Mat.C.U <- diag(Mat.C.U)
    Part.1 <- Transpose.Mat.Constraints %*% Mat.C.U
    Projection.Mat <- Mat.Constraints %*% GINVERSE(Part.1 %*% Mat.Constraints) %*% Part.1
    Mat.Pearson <- t(Projection.Mat) %*% Mat.Delta %*% Projection.Mat
    Mat.Pearson <- diag(as.vector(diag(Mat.Pearson)))
    Mat.C.U <- (Mat.C.Pi - Mat.Pearson) %*% GINVERSE(Identity.U - diag(as.vector(diag(Projection.Mat))))
    Distance <- max(abs(Old.Mat.C.U-diag(Mat.C.U)))
    Found <- (Distance <= Tolerance)
    if(Number.Of.Iterations >= 100){
      Found <- TRUE
      Mat.C.U <- Sample.Size * (Sample.Size - 1) ^(-1) * Mat.C.Pi
    }
  }

# OUTPUT
  Vect.c <- diag(Mat.C.U)
  Vect.c
}

# FUNCTION Joint.Proba.Syst
# = = = = = = = = = = = = = = = = = = = = = = = = =
# This function computes the joint inclusion probabilities
# of a systematic sampling design given an order
#
# INPUT:
* Vect.Pi: a N x 1 vector containing the first-order inclusion probabilities.

# OUTPUT:
* The N x N matrix containing the joint-inclusion probabilities.

Joint.Proba.Syst <- function(Vect.Pi) {
  Sample.Size <- round(sum(Vect.Pi), digits=0)[1]
  Pop.Size <- length(Vect.Pi)
  
  # INITIALISATION
  Mat.Joint <- matrix(rep(0, times=Pop.Size*Pop.Size), Pop.Size, Pop.Size)
  
  # CREATION Mat.V
  First.Time <- TRUE
  for(k in (1:(Pop.Size-1))) {
    for(l in ((k+1):Pop.Size)) {
      V.k.l <- sum(Vect.Pi[(k:(l-1))])
      Delta.k.l <- V.k.l - floor(V.k.l)
      P.k <- Vect.Pi[k]
      P.l <- Vect.Pi[l]
      P.k.l <- min(max(0,(P.k-Delta.k.l)),P.l) +
                min(P.k,max(0,(Delta.k.l+P.l-1)))
      Mat.Joint[k,l] <- P.k.l
    }
  }
  Mat.Joint <- Mat.Joint + t(Mat.Joint)
  diag(Mat.Joint) <- Vect.Pi
  
  # OUTPUT
  Mat.Joint
}

# FUNCTION SUM.SQUARE.WEIGHTED.RESIDUALS
# ===============
# This function computes the weighted sum of squares of residuals:
# SUM c_i (e_i)^2
# Where
# e_i = y_i - x_i Beta
#
# * Vect.Y: The n x 1 vector containing the values y_i.
# * Mat.Regressor: the sample value of the Regressor (X) variables (a n x p matrix).
# * Weights.of.Residuals: the n x 1 vector containing

# PUT THE DATA IN A SINGLE DATA FRAME
Data <- data.frame(cbind(Vect.Y, Mat.Regressor))

# CREATE THE NAME AND THE FORMULA FOR THE WEIGHTED LEAST SQUARE FIT
NB.Regressor <- as.numeric(dim(Mat.Regressor)[2])
List.Name.Variables <- c("Y")
Formula <- "Y \symbol{126}1"
for(i in (1:NB.Regressor)){
    Name.New.Variable <- paste("X", i, sep="")
    Formula <- paste(Formula, " + " , Name.New.Variable, sep="")
}

Formula <- as.formula(Formula)
List.Blank <- dimnames(Data)[[1]]
Names.List <- list(List.Blank, List.Name.Variables)
dimnames(Data) <- Names.List

# THE WEIGHTED LEAST SQUARE FIT
Model.Fit <- lm(formula=Formula, data=Data, weights=Weights.for.Coeff.Regression)

# THE RESIDUALS
Residuals <- residuals(Model.Fit)

# OUTPUT
Sum.Square.Residuals <- sum(Weights.of.Residuals * Residuals^2)
Sum.Square.Residuals
}
References


