Published in the Journal of Computational Statistics and Data Analysis in 2020
DOI: 10.1016/j.csda.2020.106971

Sampling a two dimensional matrix

Louis-Paul Rivest & Sergio Ewane Ebouele

Department of Mathematics and Statistics, Université Laval, Québec, Canada

Abstract

A new sampling design for populations whose units can be arranged as an $N \times M$ matrix is proposed. The sample must satisfy some constraints: row and column sample sizes are set in advance. The proposed sampling method gives the same selection probability to all the sample matrices that satisfy the constraints. Three algorithms to select a sample uniformly in the feasible set are presented: an exact algorithm based on the multivariate hypergeometric distribution, an MCMC algorithm, and the cube method. Their performances are evaluated using Monte Carlo simulations. The designs for sampling elements in a given row or a given column are investigated and the single inclusion and joint selection probabilities under the proposed design are evaluated. Several variance estimators are proposed for the Horvitz-Thompson estimator of the population mean of the survey variable y and their performances are compared in a Monte Carlo study. A numerical example dealing with a creel survey of fishermen found at 9 sites over 36 days is presented.

Keywords: Balanced sampling, Creel survey, Cube method, Multivariate hypergeometric distribution, Monte Carlo Markov Chain

2010 MSC: 62D05, 62P12

*Corresponding author: Louis-Paul Rivest, Department of Mathematics and Statistics, Université Laval, Quebec Canada, G1V 0A6: Louis-Paul.Rivest@mat.ulaval.ca

[1] A dataset, R functions with documentation, and supplementary material are available as annexes in the electronic version of the manuscript.

1. Introduction

In the sample survey literature, Ohlsson [15] introduced cross-classified sampling for populations having a matrix structure. It selects samples of rows and columns of the population matrix independently and considers only the population units in the rows and the columns that have been selected. This design is considered in Vos [22], Skinner [17], and Juillard et al. [13]. This work investigates a two dimensional population matrix where entry (i, j) of the matrix has a single unit. The population to be sampled is a $N \times M$ matrix containing MN units and the objective is to estimate the mean of the survey variable y over these MN units. Let $\mathbf{Z} = \{Z_{ij} : i = 1, \ldots, N; j = 1, \ldots M\}$ be the matrix of sample indicators where $Z_{ij} = 1$ if unit (i, j) is sampled and 0 otherwise. The goal is to select a sample where the row totals $\{Z_{i\bullet}\}$ and the column totals $\{Z_{\bullet j}\}$ are fixed. We consider samples where $Z_{i\bullet}$ varies with i, to account for the rows varying importance, and where the column sample sizes are fixed, $Z_{\bullet j} = n$ for all columns as illustrated in Table 1.

Table 1: Representation of an $N \times M$ sample matrix $\mathbf{Z}$ with row totals equal to $\{m_i\}$ and column totals equal to n.

	Col 1	Col 2	...	...	Col M	Tot
Row 1	Z_{11}	Z_{12}	$\cdot$	$\cdot$	Z_{1M}	$Z_{1\bullet} = m_1$
Row 2	Z_{21}	Z_{22}	$\ddots$	$\cdot$	Z_{2M}	$Z_{2\bullet} = m_2$
$\vdots$	$\cdot$	$\cdot$	$\ddots$	$\ddots$	$\cdot$	$\vdots$
Row N	Z_{N1}	Z_{N2}	$\cdot$	$\ddots$	Z_{NM}	$Z_{N\bullet} = m_N$
Tot	$Z_{\bullet 1} = n$	$Z_{\bullet 2} = n$	$\cdot$	$\cdot$	$Z_{\bullet M} = n$	Mn

A motivating example for the design considered in this paper is the creel survey in Ida et al. [11] to estimate fishing effort; the rows of the population matrix are sites and the columns are days. On a given day, only n of the N sites can be sampled and visited. The sites might differ in their importance and a sample where $Z_{i\bullet} = m_i$ where m_i is a measure of the importance of site i is of interest, see Table 1. This design is also relevant when sampling repeatedly the same population since the fixed $\{Z_{i\bullet}\}$ allow to control the sampling effort

for the N population units. Data matrices with 0-1 entries also occur when carrying out *null model* analysis in ecology [8]. In this context $Z_{ij} = 1$ if species i is found on site j, for $i = 1, \ldots, N; j = 1, \ldots M$. Two dimensional sampling is used to create random alternatives to the observed matrix $\mathbf{Z}$ and to evaluate the null distributions of statistics measuring aspects of the species assemblage, such as its nestedness and the level of species co-occurrence, using Monte Carlo techniques [9]. The matrix $\mathbf{Z}$ can be also be looked at as an incomplete block design: M blocks are available to compare N treatments and a single block can only contain n of the N treatments. The matrix $\mathbf{Z}$ identifies the treatments that are run in each block. Selecting the matrix $\mathbf{Z}$ randomly is not optimal in this context and focussing on balanced or partially balanced matrices provides better estimators of the treatment effects [12].

This work studies a new sampling algorithm, based on the hypergeometric distribution, and a well known MCMC algorithm to select a sample matrix $\mathbf{Z}$ uniformly among the set of 0-1 matrices with the required row and column totals. It also investigates whether the cube method algorithm of [5] selects uniformly in that set. This is done by comparing this general algorithm with the first two that are specific to the problem of sampling a matrix. These algorithms are presented in Section 2; comparisons between sampling algorithms are presented in Section 5. This work also studies the properties of the new sampling design: single inclusion and joint selection probabilities are provided in Section 3 with a closed form expression for the variance of the Horvitz-Thompson estimator of the mean of survey variable y. Approximately unbiased estimators for this variance are given. Monte Carlo comparisons between the new variance estimators and an omnibus variance estimator for the cube method [6] are presented in Section 5. The analysis of a creel survey data set illustrates the methodology presented in this work.

2. Sampling algorithms

This section suggests two methods to draw a sample matrix uniformly among all possible matrices with predetermined row and column totals. They rely on the multivariate hypergeometric distribution and on a Markov chain defined on the set on feasible samples respectively. It also discusses the cube method of Deville and Tillé [5], a general algorithm for sampling under constraints, that does not target directed the set of feasible matrices.

2.1. Multivariate hypergeometric sampling

Consider a two dimensional random matrix $\mathbf{X}$ whose entries are $\{X_{ij} : i = 1, \ldots, r \ ; \ j = 1, \ldots, c\}$ are non negative integers, and let $\{X_{i\bullet}\}$ and $\{X_{\bullet j}\}$ be the marginal row and column sums. The matrix $\mathbf{X}$ is said to have a multivariate hypergeometric distribution if its row and column totals are fixed and equal to $\{X_{i\bullet}\}$ and $\{X_{\bullet j}\}$ respectively, and if the joint probabilities of the entries $\{X_{ij}\}$ are given by

$$Pr(X_{ij} = x_{ij} \ i = 1, \ldots, r \ j = 1, \ldots, c) = \frac{\prod_{i=1}^{r} X_{i\bullet}! \prod_{j=1}^{c} X_{\bullet j}!}{X_{\bullet\bullet}! \prod_{i,j} x_{ij}!}, \qquad (1)$$

for any matrix $\{x_{ij}\}$ of non negative integers meeting the row and column constraints, where $X_{\bullet\bullet} = \sum_i X_{i\bullet}$. This distribution occurs when testing for independence in a two dimensional contingency table. Under the null hypothesis, the conditional distribution of the entries of the table, given the marginal row and column totals, is a multivariate hypergeometric. A random matrix $\mathbf{X}$ with a multivariate hypergeometric distribution can be simulated from a random permutation matrix $\mathbf{P}$ of size $X_{\bullet\bullet} \times X_{\bullet\bullet}$. It suffices to let X_{ij} be the sum of entries in the sub-matrix of $\mathbf{P}$ consisting of the rows $\sum_{i'=1}^{i-1} X_{i'\bullet} + 1$ to $\sum_{i'=1}^{i} X_{i'\bullet}$ and of the columns $\sum_{j'=1}^{j-1} X_{\bullet j'} + 1$ to $\sum_{j'=1}^{j} X_{\bullet j'}$, see Agresti et al. [1, p.77-78] for details. The multivariate hypergeometric distribution provides an exact test of fit for the independence model in a two-way contingency table. Exact tests of fit for general multi-way tables can be constructed with the MCMC algorithms presented in Diaconis and Sturmfels [7]; in the special case of a two-way table,

it provides an alternative to the algorithm of Agresti et al. [1] to simulate a multivariate hypergeometric distribution.

Considering (1), all matrices with entries x_{ij} equal to either 0 or 1 and marginal row and column totals equal to $\{X_{i\bullet}\}$ and $\{X_{\bullet j}\}$ have the same probability under the multivariate hypergeometric distribution. Thus a random 0-1 matrix simulation procedure consists in simulating repeatedly $N \times M$ hypergeometric random matrices with row totals $\{m_i\}$ and column totals n until a 0-1 matrix is found. When the column totals are $n = 1$, a single iteration of the algorithm is needed as any random hypergeometric matrix $\mathbf{X}$ is a 0-1 matrix. In large problems, with $n \gg 1$, the waiting time to get a 0-1 matrix can be very long and another procedure is needed.

2.2. Markov chain Monte-Carlo sampling

The first step of this algorithm is to simulate $\mathbf{X}_0$ according to the multivariate hypergeometric distribution introduced in Section 2.1. This matrix meets the constraints on the row and the column totals. However, it can have entries larger than 1. To get a 0-1 matrix we use the sum of squares reduction algorithm of Miklós and Podani [14]. When an entry $X_{ij} > 1$ is found, this algorithm looks for a cell (k, ℓ) such that $X_{k\ell} \geq 1$, and $X_{i\ell} = X_{kj} = 0$. These 4 entries are then changed to $(X_{ij} - 1, X_{k\ell} - 1, X_{i\ell} + 1, X_{kj} + 1)$, thereby reducing the sum of the squares of the 4 entries. Repeated applications of this procedure to entries larger than 1 creates a 0-1 matrix without altering the marginal totals. Non stochastic algorithms are available to construct the matrix $\mathbf{X}_0$, such a North-West rule, that systematically fills the row of $\mathbf{X}_0$ one after the other. These might be faster; however, for repeated applications of this algorithm, we elected to use random starting values as they produce independent replications.

This first step yields $\mathbf{Z}_0$, a random 0-1 matrix, whose distribution is not necessarily uniform on the set of matrices with fixed row and column totals. To get a uniformly distributed matrix we use $\mathbf{Z}_0$ as the initial value of a Markov chain defined on the set of 0-1 matrices, with fixed row and column totals, whose stationary distribution is uniform on that set. Then we run the chain

long enough for stationarity to be reached. Several ways to construct such a Markov chain are available, see [2], [14], and [18]. This section considers the swapping algorithm proposed in Besag and Clifford [2].

Let $\mathbf{Z}_t$ be an $N \times M$ matrix giving the state of the chain at time t. To make the transition to time $t+1$ one randomly picks two rows and two columns of $\mathbf{Z}_t$. If the corresponding 2×2 sub-matrix has the form of a "checkerboard", namely

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{2}$$

then one swaps the 0 and the 1 in the selected sub-matrix to create $\mathbf{Z}_{t+1}$. If the sub-matrix selected differs from the above, then one simply sets $\mathbf{Z}_{t+1} = \mathbf{Z}_t$. Swapping leaves the marginal row and column totals unchanged. This "trial swap" algorithm yields a reversible Markov chain with a uniform stationary distribution. A random matrix $\mathbf{Z}$ is obtained by running the chain long enough, starting at the initial value $\mathbf{Z}_0$ defined above.

The next step is to determine T, the burn-in needed for the Markov chain to reach its stationary distribution. At each step there are $N(N-1)M(M-1)/4$ possible choices for the 2×2 sub-matrix. The number of occurrences of (2) is a random variable with stationary expectation $E(C)$. This expectation is calculated under a model where all possible matrices $\mathbf{Z}$ have the same probability of occurrence. An approximation to the number of steps that the Markov chain takes to leave the current one is $N \times (N-1) \times M \times (M-1)/\{4E(C)\}$. If $Z_{\bullet\bullet} = Mn$ denotes the total sample size, then we suggest running the chain for $T = Z_{\bullet\bullet} \times N \times (N-1) \times M \times (M-1)/\{4E(C)\}$ steps before selecting the sample as $\mathbf{Z}_t$. The proposal is to run the chain long enough for the expected number of successful swaps to be equal to the number of ones in the matrix. See Miklós and Podani [14] and von Gagern et al. [21] for similar recommendations. A conservative approximation for $E(C)$ is given in the next proposition that is proved in the appendix.

Proposition 1. *A lower bound for the expectation of the number of occurrences*

of (2), evaluated under a model where all possible matrices $\mathbf{Z}$ *have the same probability of occurrence, is*

$$E(C) > \frac{M}{2(M-1)} \left(Mn - \sum_{i=1}^{N} \frac{m_i^2}{M} \right)^2 .$$

This lower bound leads to the following value for the burn-in of the chain,

$$T = \frac{N(N-1)(M-1)^2}{2Mn\{1 - \sum_{i=1}^{N} m_i^2/(nM^2)\}^2}. \tag{3}$$

One objective of this work is to investigate whether the burn-in (3) is large enough to allow the Markov chain to reach its stationary distribution. This is investigated numerically in Sections 4 and 5. To implement this algorithm in R we developped a C++ program for the sum of squares reduction algorithm [14] that was used with the trial swap Markov chain in the R-package `vegan` [16].

2.3. The cube method

A sample satisfying the constraints on the row and the column totals can be selected using the cube method implementation of balanced sampling [5]. The population to sample has NM units and the single selection probabilities $\{\pi_{ij}\}$ satify the contraints, that is $\pi_{\bullet j} = \sum_i \pi_{ij} = n$ and $\pi_{i\bullet} = \sum_j \pi_{ij} = m_i$, $i = 1, \ldots, N$. As shown in Deville and Tillé [6], the fixed row and column totals are $N + M$ conditions involving the row auxiliary variables $x^{k,(r)}$, $k = 1, \ldots, N$, defined by $x_{ij}^{k,(r)} = \pi_{kj}$ if $i = k$ and 0 otherwise, and the column auxiliary variables $x^{\ell,(c)}$, $\ell = 1, \ldots, M$, defined by $x_{ij}^{\ell,(c)} = \pi_{i\ell}$ if $j = \ell$ and 0 otherwise. The Horvitz-Thompson estimator for the total of $x^{k,(r)}$ is $\sum_{j=1}^{M} Z_{kj} x_{kj}^{k,(r)}/\pi_{kj} = Z_{k\bullet}$. Requesting this to be equal to the total of $x^{k,(r)}$, m_k for $k = 1, \ldots, N$, insures that the row sums are equal to $\{m_i\}$. In a similar way calibrating on the totals of $\{x^{\ell,(c)}\}$ forces all column totals to be equal to n.

As explained in Deville and Tillé [5], the cube method algorithm has two phases. The first one, called the flight phase, uses a discrete martingale whose value at time 0 is the vector of selection probabilities $\{\pi_{ij}\}$. This martingale

creates a random vector of probabilities $\{\pi_{ij}^M\}$. This vector satisfies the constraints $\sum_{i,j} \pi_{ij}^M x_{ij}/\pi_{ij} = x_{\bullet\bullet}$ for all balancing variables x_{ij}, also its expectation is equal to the original selection probabilities $\{\pi_{ij}\}$ and most of its entries are either 0 or 1. This phase leads to a true sample if all the entries of $\{\pi_{ij}^M\}$ are either 0 or 1. If they are not a second step, called the landing phase, is applied to the vector $\{\pi_{ij}^M\}$ to obtain a 0-1 vector of sample indicators.

The set of matrices with fixed row and column totals is the set of balanced samples for this problem. To investigate whether the cube method gives a sample uniformly distributed on that set, it is compared the algorithms of Sections 2.1 and 2.2 that target directly the set of possible samples.

3. Some properties of the proposed design

This section derives the selection probabilities under the proposed uniform sampling design. It investigates the standard Horvitz-Thompson estimator of the population mean of survey variable y and provides a closed form expression for its variance and suggests variance estimators. The sample size for row i is $m_i = Z_{i\bullet}$ and that for column j is $Z_{\bullet j} = n$. Considering Table 1, m_i, n and M satisfy $\sum_{i=1}^{N} m_i = Mn$. In this section ij and $k\ell$ represent two population units using their row and column numbers; their single inclusion and joint selection probabilities are respectively given by π_{ij}, $\pi_{k\ell}$, and $\pi_{ij,k\ell}$.

3.1. Joint selection probabilities

Under the proposed sampling design, the probability that a particular 0-1 matrix $\mathbf{Z}$ is the sample is $1/\mathcal{N}$, where $\mathcal{N}$ is the number of possible samples. The design for sampling elements within the i'th row gives vector $Z^{(i)} = (Z_1^{(i)}, \ldots, Z_M^{(i)})$ satisfying $\sum_{j=1}^{M} Z_j^{(i)} = m_i$ a probability equal to $\mathcal{N}_{Z^{(i)}}/\mathcal{N}$, where $\mathcal{N}_{Z^{(i)}}$ is the number of possible matrices $\mathbf{Z}$ with row i equal to $Z^{(i)}$. The design for sampling elements within a column is defined in a similar way.

Proposition 2. *The sampling design that selects a sample matrix uniformly among all possible matrices satisfies the following properties:*

i) The design for sampling elements within the i'th row is without replacement random sampling of m_i units among M and the first order selection probabilities are $\{\pi_{ij} = m_i/M\}$;

ii) The designs for sampling elements within a column are the same for all columns;

iii) The joint selection probabilities are given by

$$\pi_{ij,k\ell} = \begin{cases} \gamma_{ik} & i \neq k, j = \ell, \\ m_i(m_i - 1)/\{M(M-1)\} & i = k, j \neq \ell, \\ \frac{m_i m_k}{M(M-1)} - \frac{\gamma_{ik}}{M-1} & i \neq k, j \neq \ell, \end{cases}$$

where $\{\gamma_{ik}\}$ are the joint selection probabilities when sampling units within a column.

Proof: The proof of *i*) and *ii*) relies on the observation that the distribution of a randomly selected $\mathbf{Z}$ is invariant with respect to a fixed permutation of its columns. This implies that the probability of drawing a $\mathbf{Z}$ with a given sample for row i is the same for all possible samples in row i. Thus the sampling design for row i is without replacement simple random sampling of m_i units among M. This also shows that the column sampling designs are all the same. To get the joint selection probabilities let $i \neq k$ and observe that, by the row-sum constraint,

$$\begin{aligned} m_i m_k &= \sum_{j=1}^{M}\sum_{\ell=1}^{M} E(Z_{ij}Z_{k\ell}), \\ &= \sum_{j=1}^{M} E(Z_{ij}Z_{kj}) + \sum_{j=1}^{M}\sum_{\ell=1\,\ell\neq j}^{M} E(Z_{ij}Z_{k\ell}), \\ &= M\gamma_{ik} + M(M-1)\pi_{i1,k2}, \end{aligned}$$

where the last equation comes the observation that, when $i \neq k$, $\pi_{ij,k\ell}$ is the same for all $j \neq \ell$. This is true because the number of matrices $\mathbf{Z}$ with $Z_{ij} = Z_{k\ell} = 1$ is the same as that with $Z_{i1} = Z_{k2} = 1$, for any $j \neq \ell$. Solving the

above equation for $\pi_{i1,k2}$ gives the result. □

3.2. Sampling properties of the Horvitz-Thompson estimator

The population mean of y is $\bar{y}_U = \sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}/(NM)$ and its Horvitz-Thompson estimator is

$$\widehat{\bar{y}} = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{ij}y_{ij}/(m_i N) = \sum_{i=1}^{N} \bar{y}_{i,s}/N, \tag{4}$$

where $\bar{y}_{i,s}$ is the sample mean for the ith row. Clearly $\bar{y}_{i,s}$ is an unbiased estimator for $\bar{y}_{i\bullet} = \sum_{j=1}^{M} y_{ij}/M$ and $\widehat{\bar{y}}$ is unbiased for $\bar{y}_U$. The calculation of the variance of $\widehat{\bar{y}}$ involves the $N \times N$ matrix Γ containing the second order selection probabilities γ_{ij} defined in Proposition 2 with $\gamma_{ii} = m_i/M$. This matrix is unknown and has to be evaluated by Monte Carlo simulations as illustrated in Section 5. The variance of $\widehat{\bar{y}}$ is given in the next proposition whose proof is given in the appendix.

Proposition 3. *The following results hold,*

- *The covariance between the sample means for rows i and k is given by $Cov(\bar{y}_{i,s}, \bar{y}_{k,s}) = \Delta_{ik}S_{ik}, \quad i,k = 1,\ldots,N$, where Δ_{ik} is the (i,k) entry of*

$$\mathbf{\Delta} = M\, diag(\mathbf{m})^{-1}\mathbf{\Gamma}\, diag(\mathbf{m})^{-1} - \mathbf{1}_N\mathbf{1}_N^{\top}/M, \tag{5}$$

  $\mathbf{1}_N$ is the $N \times 1$ vector of 1 and diag($\mathbf{m}$) is the diagonal matrix for $\mathbf{m}$, the vector of the row sample sizes m_i, and S_{ik} is the covariance beween rows i and k,

$$S_{ik} = \sum_{j=1}^{M}(y_{ij} - \bar{y}_{i\bullet})(y_{kj} - \bar{y}_{k\bullet})/(M-1) \tag{6}$$

- *The variance of the Horvitz-Thompson estimator (4) is*

$$Var(\widehat{\bar{y}}) = \frac{tr(\mathbf{S\Delta})}{N^2}, \tag{7}$$

where $\mathbf{\Delta}$ *is given in (5),* $tr(\cdot)$ *denotes the trace operator and* $\mathbf{S}$ *is the* $N \times N$ *matrix of the* S_{ik}*, see (6).*

Observe that since $\sum_{k=1}^{N} \gamma_{ik} = E(\sum_{k=1}^{N} Z_{ij} Z_{kj}) = nm_i/M$ one has $\mathbf{\Delta m} = 0$. This is used in Section 5.3. If the rows were sampled independently one of the other then the sampling design would be stratified random sampling; the variance of (4) would then be $\mathrm{Var}_{str}(\widehat{\bar{y}}) = \sum_{i=1}^{N} \Delta_{ii} S_{ii}/N^2$, as, considering (5), $\Delta_{ii} = 1/m_i - 1/M$. This is equal to (7) if $\Delta_{ik} = 0$ for $i \neq k$. In general it is conjectured that $\Delta_{ik} < 0$ (i.e. $\gamma_{ik} < m_i m_k/M^2$) and the proposed uniform sampling design provides a more precise estimator of $\bar{y}_U$ than a stratified design if there is a positive correlation between rows, that is if $S_{ik} > 0$.

When the row sample sizes m_i are equal to m, $\mathbf{m} = m\mathbf{1}_N$ and $\gamma_{ik} = n(n-1)/\{N(N-1)\}$ for $i \neq k$, then (5) gives $\mathbf{\Delta} = (N-n)(\mathbf{I}_N - \mathbf{1}_N \mathbf{1}_N^\top/N)/\{m(N-1)\}$, where $\mathbf{I}_N$ is an $N \times N$ identity matrix; (7) then simplifies to

$$\mathrm{Var}\{\widehat{\bar{y}}\} \quad = \quad \frac{1 - m/M}{Nm} \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (y_{ij} - \bar{y}_{\bullet j} - \bar{y}_{i\bullet} + \bar{y}_U)^2}{(M-1)(N-1)}.$$

This expression is very similar to the variance of the sample mean in simple random sampling. It is equal to one minus the sampling fraction ($m/M = n/N$) divided by the sample size ($nM = mN$) multiplied by the variance of the ANOVA residuals with a divisor of $(N-1)(M-1)$.

Estimators for the means of y in row i, $\bar{y}_{i\bullet} = \sum_{j=1}^{M} y_{ij}/M$, or column j, $\bar{y}_{\bullet j} = \sum_{i=1}^{N} y_{ij}/N$, and their variances are easily evaluated. The estimator for $\bar{y}_{i\bullet}$ is $\bar{y}_{i,s}$ with variance $\Delta_{ii} S_{ii}$ while that for $\bar{y}_{\bullet j}$ is $\bar{y}_{j,HT} = M \sum_{i=1}^{N} Z_{ij} y_{ij}/(m_i N)$ with variance $\mathbf{y}_j^\top \mathbf{\Delta} \mathbf{y}_j/(MN^2)$, where $\mathbf{y}_j$ i the $N \times 1$ vector $y_{ij} : \ i = 1, \ldots, N$.

3.3. Variance estimation

Simple unbiased variance estimators are available for the estimators of the mean in row i, $\bar{y}_{i,s}$, and for that in column j, $\bar{y}_{j,HT}$: the variance for the sample mean in simple random sampling for $\bar{y}_{i,s}$ and the Sen-Yates-Grundy estimators for $\bar{y}_{j,HT}$. The estimation of the variance of (4) is more complicated as it involves

the covariances between rows. The next proposition proposes a conditionally unbiased estimator.

Proposition 4. *The following results hold:*

- *If* $n_{ik} = \sum_{j=1}^{M} Z_{ij} Z_{kj}$*, the number of columns sampled in both row* i *and row* k*, is greater than 1, then the estimator of* S_{ik} *given by*

$$s_{ik} = \frac{\sum_{j=1}^{M} Z_{ij} Z_{kj} y_{ij} y_{kj} - \sum_{j=1}^{M} Z_{ij} Z_{kj} y_{ij} \sum_{j=1}^{M} Z_{ij} Z_{kj} y_{kj} / n_{jk}}{n_{ik} - 1} \tag{8}$$

  is conditionally unbiased, given the sample size n_{ik}*, for* $i, k = 1, \ldots, N$*.*

- *If the matrix* $\widehat{\mathbf{S}}$ *is defined by* $\hat{S}_{ik} = s_{ik}$ *if* $n_{ik} > 1$ *and 0 otherwise, for* $i, k = 1, \ldots, N$ *then he plug-in estimator* $v_{PI}(\widehat{\bar{y}}) = tr(\widehat{\mathbf{S}}\boldsymbol{\Delta})/N^2$ *is unbiased for (7) provided that all the joint sample sizes* n_{ik}*,* $i, k = 1, \ldots, N$ *are larger than 1.*

The proof of this result comes from the column exchangeability highlighted in Proposition 2: given n_{ik}, the columns that are sampled in both row i and row k is a simple random sample of size n_{ik} and (8) gives an unbiased covariance estimator under simple random sampling.

The condition $n_{ik} > 1,\ i, k = 1, \ldots, N$ for $v_{PI}(\widehat{\bar{y}})$ to be unbiased is fairly restrictive. When it is not met, some of the entries of $\widehat{\mathbf{S}}$ are 0 and $v_{PI}(\widehat{\bar{y}})$ might overestimate (7). An alternative estimator assumes an equal correlation between all the rows. A simple estimator of this correlation and the "equal correlation" estimator of the covariance S_{ik} are given by

$$\hat{\rho} = \frac{\sum_{i,k:\ n_{ik}>1} s_{ik}}{\sum_{i,k:\ n_{ik}>1} \sqrt{s_{ii} s_{kk}}} \quad \text{and} \quad s_{ik}^{(ec)} = \hat{\rho}\sqrt{s_{ii} s_{kk}},\ i \neq k.$$

This leads to the equal correlation estimator $v_{ec}(\widehat{\bar{y}}) = \mathrm{tr}(\widehat{\mathbf{S}}^{(ec)}\boldsymbol{\Delta})/N^2$, where $\widehat{\mathbf{S}}^{(ec)}$ is the matrix with diagonal elements given by the row variances $\{s_{ii}\}$ and off-diagonal elements equal to $s_{ik}^{(ec)}$.

Two other variance estimators are considered in section 5. The stratified variance estimator, $v_{str}(\widehat{\bar{y}})$, is obtained by setting $\hat{\rho} = 0$ in the formula for

$v_{ec}(\widehat{\bar{y}})$. Deville and Tillé [6] proposed omnibus variance estimators developed for an arbitrary balanced sampling design. Their second approximation, see also equation (15) of Breidt and Chauvet [3], is given by

$$v_{DT}(\widehat{\bar{y}}) = \frac{nM}{M^2N^2(nM-N-M+1)} \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{Z_{ij}(1-\pi_{ij})}{\pi_{ij}^2}(y_{ij}-\hat{y}_{ij})^2, \qquad (9)$$

where $\hat{y}_{ij}$ is the predicted value for y_{ij} in a linear model where the row and column indicators, $x^{k,(r)}$ and $x^{\ell,(c)}$ defined in Section 2.1, are the explanatory variables and where a weight of $(1-\pi_{ij})/\pi_{ij}^2 = M(M-m_i)/m_i^2$ is given to sample data point ij. This estimator does not depend on the joint inclusion probabilities derived in Proposition 2. Note that the Sen-Yates-Grundy variance estimator for (4) is not investigated in this work as, considering Proposition 2, the condition for it to be positive, $\pi_{ij,k\ell} - m_i m_k/M^2 > 0$, fails.

4. A detailed example

This section considers a simple example with $N = 4$ rows and $M = 3$ columns with a column sample size of $n = 2$ and row totals $m_1 = m_2 = 1$ and $m_3 = m_4 = 2$. The possible sample matrices $\mathbf{Z}$ for these marginal totals are enumerated. The waiting time for the hypergeometric algorithm of Section 2.1 and the suitability of the burn-in (3) for the MCMC algorithm are investigated. This section also checks whether the joint selection probabilities meet the constraint $m_i m_k/M^2 > \gamma_{ik}$, that insures that $\Delta_{ik} < 0$ in (7).

There are six possible samples of size 2, $s = (s_1, s_2, s_3, s_4)$, where $s_i = 1$ if units i is in the sample and 0 otherwise. Let $\{n_s\}$ be the frequency of s in $\mathbf{Z}$. The 4 row constraints can be expressed as $\sum s_i n_s = m_i,\ i = 1, \ldots, 4$. There are three solutions to these equations, namely $\{n_{1,0,1,0} = 1, n_{0,1,0,1} = 1, n_{0,0,1,1} = 1\}$, $\{n_{1,0,0,1} = 1, n_{0,1,1,0} = 1, n_{0,0,1,1} = 1\}$, and $\{n_{1,1,0,0} = 1, n_{0,0,1,1} = 2\}$. Matrices

$\mathbf{Z}$ for each of the three solutions are given by

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \mathbf{Z}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad \mathbf{Z}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Permuting the columns of these matrices gives a total of $\mathcal{N} = 15$ possible samples that are listed in the Supplementary Material. The column joint selection probabilities γ_{ik} are evaluated by a direct enumeration using the 45 possible column samples. They are given by $\gamma_{12}^{(3)} = 1/15 = .067$, $\gamma_{13}^{(3)} = \gamma_{14}^{(3)} = \gamma_{23}^{(3)} = \gamma_{24}^{(3)} = 2/15 = .133$ and $\gamma_{34}^{(3)} = 2/5 = .4$. They are all smaller than $m_i m_k/9$ so $\Delta_{ik} < 0$ for this special case.

The probability that the multivariate hypergeometric distribution, see (2.1), with marginal totals $(1, 1, 2, 2)$ and $(2, 2, 2)$ be equal to one of the 15 possible sample matrix is $2^5/6! = 2/45$. Thus the probability that it gives a 0-1 matrix is $15 \cdot 2/45 = 2/3$; therefore the waiting time for the algorithm of Section 2.1 is short for this small example. The Supplementary Material gives the 15×15 transition matrix P of the Markov chain of Section MCMC. In this example (3) gives a burn-in of $T = 21$. The Supplementary Material gives the matrix P^{21} and shows that each row of this matrix gives accurate evaluations of the joint selections probabilities γ_{ik} as their largest relative errors is 2.5 10^{-4}. Thus (3) gives an adequate burn-in for this small problem.

We next investigate how the design for sampling units within a column varies with M. To do this we consider $\mathbf{Z}$ matrices constructed with the following specifications: $N = 4$, $n = 2$, and $M = 3K$ with row totals $m_1 = m_2 = K$ and $m_3 = m_4 = 2\ K$ for an arbitrary positive integer K. The entries in Table 2 are obtained by evaluating exact expressions for $\gamma_{ik}^{(3K)}$ provided in the appendix. Note that for each K the condition, $\gamma_{ik}^{(3K)} < m_i m_k/9K^2$, is met.

In Table 2, the $K = \infty$ column gives the joint selection probabilities for a conditional Poisson sampling design with $N = 4$ and $n = 2$, see Tillé [19]. This

Table 2: Joint selection probabilities, $\gamma_{ik}^{(3K)}$, of the design that samples units within a column for various values of K.

Sample	$K=1$	$K=2$	$K=5$	$K=10$	$K=15$	$K=\infty$
$\{1,2\}$	0.0667	0.0566	0.0529	0.0522	0.0520	0.0516
$\{1,3\}$	0.1333	0.1384	0.1402	0.1406	0.1407	0.1409
$\{3,4\}$	0.4	0.3899	0.3862	0.3855	0.3853	0.3849

design gives a probability proportional to $\exp(\beta_i + \beta_k)$ to sample $\{i,k\}$, where the β_i's are selected in such a way that the marginal selection probabilities are given by $(1/3, 1/3, 2/3, 2/3)$. Elementary calculations give $\exp(\beta_1) = \exp(\beta_2) = x = (\sqrt{3}-1)/2$ and $\exp(\beta_3) = \exp(\beta_4) = 1$ as solutions to these equations. The joint conditional Poisson selection probabilities for units 1 and 2 is, for instance, equal $x^2/(1+4x+x^2) = .0516$, in agreement with the entry for γ_{12} in the $K=\infty$ column of Table 2. We conjecture that this result holds in general: when N is fixed, as M goes to infinity the design for sampling elements within a column converges to a conditional Poisson sampling design.

5. Numerical investigations

This section considers a creel survey for measuring fishing effort for striped bass (*Morone saxatilis*), described in Ida et al. [11], that uses sampling in two dimensions. It compares, using Monte Carlo methods, the three sampling algorithms of Section 2 and the four variance estimators proposed in Section 3.3.

5.1. A creel survey for striped bass

The survey involves $N = 9$ sites and lasts $M = 36$ days. On each day $n = 2$ sites can be visited and, as the sites do not have the same importance, the site sample sizes are $m_1 = 10,\ m_2 = 11,\ m_3 = 10,\ m_4 = 11,\ m_5 = 7,\ m_6 = m_7 = m_8 = 6,\ m_9 = 5$. The study variable y is the fishing effort in number of fishing hours over a 2 hours observation period [4]. Thus on a given site-day, $y = 3$ if three fishermen have been fishing for one hour each or if two fishermen fished for 90 minutes each. In creel survey this is estimated separately of success rate, the average number of fish caught by hours of fishing. We treat the fishing

effort data set as if it had been collected using the uniform sampling design considered in this paper. It is given in the Supplementary Material together with R-codes for the evaluation of the estimators of Section 3. The Horvitz-Thompson estimate is $\widehat{\overline{y}} = 8.3$ and the 4 estimates of standard errors proposed in Section 3.3 are $se_{PI} = 0.80,\ se_{str} = 0.85,\ se_{DT} = 0.56$, see (9), and $se_{ec} = 0.75$. For this data set the correlation estimate of Section 3.3 is $\hat{\rho} = .25$ and the sites are positively correlated within days, thus the stratified estimate se_{str} has a positive bias. More than half of the joint sample sizes n_{ik} in Proposition 3 are less than 1 so many covariances S_{ik} cannot be estimated and the plug-in estimator $se_{PI} = 0.80$, that sets them equal to 0, might also have a positive bias. This is further investigated using a simulation study in Section 5.3.

5.2. Comparisons of three sampling algorithms of Section 2

This section investigates whether the three sampling algorithms of Section 2 are equivalent for the creel survey of Section 5.1. The hypergeometric algorithm is the gold standard to which the other 2 sampling algorithms are compared. For the MCMC algorithm we investigate whether the burn-in (3) of $T = 1063$ is large enough for the Markov chain to reach its stationary distribution. For balanced sampling we use the cube method implemented in the function `samplecube` of the package `sampling` [20]. This function starts with a random permutation of the MN population units; this insures that the design for sampling units within a row is simple random sampling and that all the column sampling designs are the same. The question is whether these three algorithms give the same joint selection probabilities $\{\gamma_{ik}\}$ when sampling units within a column.

A total of $B = 10^4$ Monte Carlo repetitions were run for the 3 algorithms. A Monte Carlo evaluation of the $N \times N$ matrix $\mathbf{\Gamma}$ of the joint selection probabilities $\{\gamma_{ik}\}$ was calculated using

$$\mathbf{\Gamma} = \frac{1}{MB} \sum_{b=1}^{B} \mathbf{Z}_b \mathbf{Z}_b^\top, \tag{10}$$

with $M = 36$. The Monte Carlo estimates of γ_{ij} for the three algorithms are

presented in Table 3, the variances of the Monte Carlo samples are reported in the v columns. The statistics $z_{12} = \sqrt{10^4}(\gamma_{ik}^{hyp} - \gamma_{ik}^{MCMC})/\sqrt{v^{hyp} + v^{MCMC}}$ tests the null hypothesis that the hypergeometric and the MCMC algorithms give the same estimation for γ_{ij}; they are reported in the z_{12} column of Table 3. The z_{13} column of Table 3 compares the cube method and the hypergeometric estimates. The test statistics of Table 3 do not allow to reject the null hypothesis that the three sampling algorithms have the same joint selection probabilities: the largest absolute z statistic of 2.17 in Table 3 is not extreme, considering that 36 z statistics are calculated for each of the two algorithms tested. With 10^4 repetitions, the coefficients of variation of the estimates of the joint selection probabilities in Table 3 range between 0.6% and 1.4% so these estimates are precise.

To validate the findings in Table 3 a second experiment was carried out with $M = 72$, $N = 9$, $n = 2$ and row samples sizes of $\{2m_i\}$. The burn-in (3) for the MCMC algorithm is $T = 2187$. As shown in Table 4, the hypergeometric algorithm took more than 5 hours to simulate the 10 000 matrices: with $M = 72$ and $n = 2$ the probability for an hypergeometric matrix to be a 0-1 matrix is small and this algorithm cannot be used to simulate large matrices. The results are presented in the Supplementary Material; they reveal no differences between the hypergeometric and the MCMC joint selection probabilities: the burn-in (3) is large enough for the stationary distribution to be reached in the two examples considered in this section. This second set of simulations used the C++ implementation of the cube method in Grafström and Lisic [10] as that in Tillé and Matei [20] sometimes failed to converge. Some slight differences between the cube method and the hypergeometric joint selection probabilities can be noted as 5 of the 36 absolute z-statistics are larger than 2. Thus, for this larger problem, the sampling design associated with the cube method might not be exactly the uniform distribution in the set of feasible samples.

The $M = 72$ and $M = 36$ hypergeometric estimates of $\mathbf{\Gamma}$ are not significantly different since the absolute z statistics comparing the two sets of estimates are all less than 2. Thus $M = 36$ is large enough for γ_{ik} to reach its limiting value as M

Table 3: Evaluation of the joint selection probabilities $\{\gamma_{ik}\}$ using three sampling algorithms for a 9×36 population matrix and test statistics comparing hypergeometric estimates to those obtained with MCMC (z_{12}) and the cube method (z_{13})

		Hypergeometric		MCMC		Cube Method			
i	k	γ_{ik}^{hyp}	v^{hyp}	γ_{ik}^{MCMC}	v^{MCMC}	γ_{ik}^{bal}	v^{bal}	z_{12}	z_{13}
1	2	0.05162	0.00097	0.05192	0.00097	0.05182	0.00094	-0.66972	-0.44595
1	3	0.04583	0.00087	0.04539	0.00084	0.04546	0.00088	1.08042	0.90415
1	4	0.05133	0.00095	0.05184	0.00097	0.05107	0.00091	-1.18005	0.59841
1	5	0.03046	0.00062	0.03047	0.00065	0.03080	0.00063	-0.03886	-0.97304
1	6	0.02603	0.00056	0.02572	0.00056	0.02567	0.00056	0.91365	1.07861
1	7	0.02569	0.00054	0.02612	0.00055	0.02578	0.00055	-1.33423	-0.27823
1	8	0.02569	0.00055	0.02547	0.00055	0.02588	0.00055	0.67011	-0.55234
1	9	0.02112	0.00045	0.02083	0.00046	0.02131	0.00047	0.95764	-0.60542
2	3	0.05174	0.00095	0.05254	0.00094	0.05200	0.00093	-1.86069	-0.60223
2	4	0.05823	0.00102	0.05725	0.00101	0.05826	0.00102	2.17432	-0.06146
2	5	0.03417	0.00070	0.03407	0.00069	0.03383	0.00068	0.28346	0.91328
2	6	0.02846	0.00058	0.02861	0.00060	0.02831	0.00059	-0.42854	0.45336
2	7	0.02906	0.00060	0.02832	0.00059	0.02888	0.00060	2.13382	0.51306
2	8	0.02889	0.00060	0.02924	0.00060	0.02871	0.00059	-1.02681	0.53153
2	9	0.02338	0.00050	0.02359	0.00051	0.02375	0.00050	-0.68167	-1.17755
3	4	0.05110	0.00094	0.05150	0.00095	0.05137	0.00095	-0.92594	-0.62659
3	5	0.03008	0.00063	0.03025	0.00063	0.03053	0.00063	-0.47790	-1.26910
3	6	0.02602	0.00055	0.02561	0.00055	0.02585	0.00055	1.25146	0.51035
3	7	0.02578	0.00054	0.02543	0.00056	0.02539	0.00055	1.05749	1.18001
3	8	0.02582	0.00055	0.02550	0.00055	0.02611	0.00057	0.96265	-0.87468
3	9	0.02141	0.00047	0.02156	0.00047	0.02107	0.00046	-0.48939	1.10968
4	5	0.03447	0.00069	0.03465	0.00071	0.03421	0.00068	-0.47633	0.70550
4	6	0.02909	0.00061	0.02905	0.00060	0.02937	0.00060	0.12777	-0.78241
4	7	0.02874	0.00059	0.02894	0.00061	0.02922	0.00060	-0.58433	-1.39901
4	8	0.02874	0.00061	0.02896	0.00060	0.02837	0.00059	-0.62311	1.07566
4	9	0.02385	0.00051	0.02336	0.00050	0.02369	0.00051	1.54552	0.51415
5	6	0.01709	0.00040	0.01701	0.00040	0.01718	0.00039	0.26519	-0.31687
5	7	0.01704	0.00040	0.01727	0.00040	0.01693	0.00039	-0.83654	0.38591
5	8	0.01737	0.00040	0.01683	0.00038	0.01697	0.00038	1.94778	1.43352
5	9	0.01376	0.00032	0.01389	0.00033	0.01399	0.00033	-0.50266	-0.88041
6	7	0.01424	0.00033	0.01469	0.00034	0.01439	0.00033	-1.74705	-0.57052
6	8	0.01404	0.00033	0.01436	0.00034	0.01424	0.00034	-1.25816	-0.78741
6	9	0.01169	0.00028	0.01161	0.00028	0.01167	0.00028	0.36548	0.10611
7	8	0.01428	0.00033	0.01407	0.00033	0.01452	0.00033	0.83010	-0.93642
7	9	0.01184	0.00028	0.01181	0.00028	0.01156	0.00027	0.12893	1.22565
8	9	0.01183	0.00028	0.01224	0.00029	0.01187	0.00028	-1.68644	-0.15262

becomes large as illustrated in Table 2. Indeed the matrix $\mathbf{\Gamma}$ estimated in Table 3 is that for a conditional Poisson sampling design with marginal probabilities $\{m_i/36\}$: the conjecture presented at the end of Section 4 holds for the creel survey considered in this section.

Table 4 provides the computer time required to simulate 10 000 samples $\mathbf{Z}$ for several values of M with the algorithms proposed in Section 2. This was evaluated using the R-function `Sys.time` In all simulations $N = 9$, $n = 2$ and the first order selection probabilities were equal to those for the creel survey of Section 5.1. Table 4 shows that the hypergeometric algorithm is not appropriate for large problems as the waiting time to get a sample matrix is too long. Thus the simulations of the next section uses the MCMC algorithm with burn-in (3). In the Supplementary Material tests comparing the MCMC Monte Carlo estimates of the matrix Γ obtained with $M = 144$ and $M = 288$ with the $M = 72$ hypergeometric estimate are not significant. This suggests that the burn-in (3) is adequate for these larger problems.

Table 4: Computer time to simulate 10 000 samples with the three algorithms of Section 2 for various values of M.

Algorithm	$M = 36$	$M = 72$	$M = 144$	$M = 288$
Cube	14 seconds	29.6 minutes	NA	NA
Hypergeometric	1.1 minutes	5.6 hours	NA	NA
MCMC	26 seconds	30 seconds	42 seconds	2.3 minutes

5.3. Comparisons of the variance estimators of Section 3.3

This section suggests a log-linear model for the fishing effort of Section 5.1. It derives the expectation of $\bar{y}_U$ and $\mathbf{S}$ under the proposed model. It also investigates, by Monte Carlo simulations, the bias and the variance of the four variance estimators proposed in Section 3.3 under repeated sampling from 12 populations simulated using the proposed model.

The proposed model for y_{ij} is

$$y_{ij} = m_i a_j e_{ij},\ i = 1, \ldots, N;\ j = 1, \ldots, M, \tag{11}$$

where m_i is the sample size for row i and $\{a_j\}$ and $\{e_{ij}\}$ have independent log-normal distributions with respective parameters $(-\sigma_a^2/2, \sigma_a^2)$ and $(-\sigma^2/2, \sigma^2)$. With this parametrization the log-normal random variables a_j and e_{ij} have an expectation of 1 and respective variances of $\exp(\sigma_a^2) - 1$ and $\exp(\sigma^2) - 1$. Under this model, the expectation of $\bar{y}_U$ is $\bar{m}$ the mean of the m_i's. The variance covariance matrix of the $N \times 1$ vector $\mathbf{y}_j$ of the y entries in column j is

$$\begin{aligned} \mathbf{\Sigma} = \mathrm{Var}_m(\mathbf{y}_j) &= E_m\{\mathrm{Var}_m(\mathbf{y}_j|a_j)\} + \mathrm{Var}_m\{E_m(\mathbf{y}_j|a_j)\}, \\ &= \mathrm{diag}(\mathbf{m})^2 \exp(\sigma_a^2)\{\exp(\sigma^2) - 1\} + \{\exp(\sigma_a^2) - 1\}\mathbf{m}\mathbf{m}^\top, \end{aligned}$$

where index m means that the moments are taken with respect to model (11). The correlations between the N components of $\mathbf{y}_j$ are all equal to $\rho = \{\exp(\sigma_a^2) - 1\}/\{\exp(\sigma_a^2 + \sigma^2) - 1\}$. Since $E_m(\mathbf{S}) = \mathbf{\Sigma}$, the expectation of (7) under the proposed model is

$$\begin{aligned} E_m\{\mathrm{Var}(\widehat{\bar{y}})\} &= \frac{1}{N^2}\mathrm{tr}(\mathbf{\Delta}\mathbf{\Sigma}), \\ &= \frac{1}{N^2}\exp(\sigma_a^2)\{\exp(\sigma^2) - 1\}\sum_{i=1}^{N} m_i\left(1 - \frac{m_i}{M}\right), \end{aligned}$$

since $\mathbf{\Delta}\mathbf{m} = 0$, as stated after Proposition 3. Under (11) the random effects $\{a_j\}$ create a positive dependency between rows. This makes $E_m\{\mathrm{Var}(\widehat{\bar{y}})\}$ smaller that $E_m\{\mathrm{Var}_{str}(\widehat{\bar{y}})\}$, the expected variance under a stratified design that has independent samples in the N rows, as $E_m\{\mathrm{Var}_{str}(\widehat{\bar{y}})\}/E_m\{\mathrm{Var}(\widehat{\bar{y}})\} = 1/(1-\rho)$.

The simulations are carried out in the setting of the creel survey, with $M = 36 \cdot 2^{K-1}$, $N = 9$, $n = 2$ and $\{m_i^{(K)} = 2^{K-1} \cdot m_i\}$, $K = 1, 2, 3, 4$ where the vector $\{m_i\}$ is given in Section 5.1. For all the values of K, a y population matrix was generated using (11) for three sets of variance components, (σ_a^2, σ^2), given by $\{(0, 0.8), (0.2, 0.6), (0.4, 0.4)\}$ corresponding to correlations ρ of 0, 0.18, and 0.4 respectively. Note that for i and j fixed, y_{ij} has the same marginal log-normal distribution for the three sets of variance components. Thus Table 5 reports results for 12 simulated populations. From each one, 10 000 samples

were drawn using matrices $\mathbf{Z}$ generated using the MCMC algorithm of Section 2.2. The results are presented in Table 5. As stated in Section 5.2, the joint selection probability matrix Γ is constant in K and $\Delta^{(K)}$, the matrix (5) for the simulation with $M = 36 \cdot 2^{K-1}$ is $\Delta^{(K)} = \Delta/2^{K-1}$ where Δ is the $M = 36$ matrix calculated with the hypergeometric estimates of Table 3. Each row of this table is obtained with a single y data matrix and $\mathrm{Var}(\widehat{\overline{y}})$, the variance of $\widehat{\overline{y}}$ for this data matrix, is evaluated using (7).

Table 5: Comparison of four variance estimators, the stratified estimator (str), the plug-in estimator (PI), the equal correlation estimator (ec) and the Deville and Tillé estimator (DT). The expectations of the variance estimators are reported together with their standard deviations, between parenthesis.

M	ρ	$\mathrm{Var}(\widehat{\overline{y}})$	v_{str}	v_{PI}	v_{ec}	v_{DT}
36	0	0.829	0.803 (0.280)	0.814 (0.297)	0.821 (0.317)	1.382 (0.733)
36	0.18	0.653	0.795 (0.261)	0.754 (0.253)	0.722 (0.264)	0.897 (0.452)
36	0.40	0.544	0.738 (0.309)	0.678 (0.307)	0.618 (0.324)	0.643 (0.313)
72	0	0.486	0.503 (0.225)	0.494 (0.225)	0.498 (0.224)	0.787 (0.422)
72	0.18	0.350	0.424 (0.136)	0.372 (0.127)	0.368 (0.123)	0.593 (0.247)
72	0.40	0.333	0.566 (0.184)	0.409 (0.152)	0.426 (0.157)	0.315 (0.109)
144	0	0.243	0.246 (0.131)	0.242 (0.126)	0.245 (0.129)	0.463 (0.378)
144	0.18	0.195	0.237 (0.086)	0.201 (0.079)	0.213 (0.079)	0.291 (0.117)
144	0.40	0.106	0.198 (0.040)	0.114 (0.034)	0.141 (0.032)	0.133 (0.032)
288	0	0.118	0.117 (0.027)	0.118 (0.029)	0.118 (0.029)	0.194 (0.067)
288	0.18	0.106	0.127 (0.032)	0.107 (0.030)	0.114 (0.030)	0.146 (0.038)
288	0.40	0.062	0.120 (0.029)	0.063 (0.025)	0.080 (0.021)	0.079 (0.015)

When $M = 36$, the expectations $\{M\gamma_{ik}\}$ of the joint sample sizes $\{n_{ik}\}$ range between 0.42 and 2.1. Thus most of the covariance estimators (8) are undefined as many joint sample sizes n_{ik} are less than 2. For the small M simulations, the estimator $v_{PI}(\widehat{\overline{y}})$ that sets the non-estimable covariances to 0 has a positive bias. In Table 5 $v_{ec}(\widehat{\overline{y}})$ is, as expected, less biased than $v_{PI}(\widehat{\overline{y}})$ when $M = 36$; it is the best variance estimator for the creel data of Section 5.1. In Table 5, $M = 288$ is needed for all the covariances to be estimable with a high probability and for $v_{PI}(\widehat{\overline{y}})$ to be an unbiased variance estimator. In Table 5, the Deville-Tillé variance estimator has an erratic behavior; estimators designed specifically for the problem of sampling in two dimensions do better than this omnibus estimator.

6. Discussion

For small matrices, the three algorithms of Section 2 are nearly equivalent for sampling in two dimensions. For larger problems, the MCMC algorithm is the best choice as it is fast and samples uniformly in the set of possible matrices. Variance estimation for the Horvitz-Thompson estimator is challenging for this design as the between row covariance matrix needs to be estimated. The variance estimators proposed in Section 3.3 address this problem.

The uniform sampling algorithms presented in Sections 2.1 and 2.2 work with unequal row and column totals. Allowing both sets of totals to vary creates a relatively complex design: balanced sampling cannot be used because there are no closed form expressions for the selection probabilities π_{ij} and the designs for sampling units within a row or a column are relatively complex. Indeed, changing a single row or column total alters the selection probabilities of the MN units of the population. This generalized sampling design will be considered in future work.

7. Acknowledgements

The financial supports of the Sentinelle Nord, a grant form the Canada First Research Excellence Fund, and the Natural Sciences and Engineering Research Council of Canada are gratefully acknowledged. We also want to thank the referees and the Associate Editor whose comments helped to improve the content of the paper.

References

[1] Agresti, A., Wackerly, D., Boyett, J. M., 1979. Exact conditional tests for cross-classifications: Approximation of attained significance levels. Psychometrika 44, 75–83.

[2] Besag, J., Clifford, P., 1989. Generalized monte carlo significance tests. Biometrika 76 (4), 633–642.

[3] Breidt, F. J., Chauvet, G., 2011. Improved variance estimation for balanced samples drawn via the cube method. Journal of Statistical Planning and Inference 141 (1), 479–487.

[4] Daigle, G., Crépeau, H., Bujold, V., Legault, M., 2015. Enquête de la pêche sportive au bar rayé en gaspésie en 2015. Tech. rep., Service de consultation statistique de l'Université Laval, Ministère du Développement durable, de l'Environnement, de la Faune et des Parcs du Québec, Québec.

[5] Deville, J.-C., Tillé, Y., 2004. Efficient balanced sampling: the cube method. Biometrika 91 (4), 893–912.

[6] Deville, J.-C., Tillé, Y., 2005. Variance approximation under balanced sampling. Journal of Statistical Planning and Inference 128 (2), 569–591.

[7] Diaconis, P., Sturmfels, B., 1998. Algebraic algorithms for sampling from conditional distributions. The Annals of statistics 26 (1), 363–397.

[8] Gotelli, N. J., Graves, G. R., 1996. Null models in ecology. Smithsonian Institution Press, University of Michigan USA.

[9] Gotelli, N. J., Ulrich, W., 2012. Statistical challenges in null model analysis. Oikos 121, 171–180.

[10] Grafström, A., Lisic, J., 2018. BalancedSampling: Balanced and Spatially Balanced Sampling. R package version 1.5.4.
URL `http://www.antongrafstrom.se/balancedsampling`

[11] Ida, I. O., Rivest, L.-P., Daigle, G., 2018. Using balanced sampling in creel surveys. Survey Methodology 44, 239–252.

[12] John, P. W., 1998. Statistical design and analysis of experiments. SIAM.

[13] Juillard, H., Chauvet, G., Ruiz-Gazen, A., 2017. Estimation under cross-classified sampling with application to a childhood survey. Journal of the American Statistical Association 112, 850–858.

[14] Miklós, I., Podani, J., 2004. Randomization of presence–absence matrices: comments and new algorithms. Ecology 85 (1), 86–92.

[15] Ohlsson, E., 1996. Cross-classified sampling. Journal of Official Statistics 12, 241–251.

[16] Oksanen, J., Blanchet, F. G., Kindt, R., Legendre, P., Minchin, P. R., Ohara, R., Simpson, G. L., Solymos, P., Stevens, M. H. H., Wagner, H., et al., 2013. Package vegan. Community ecology package, version 2 (9).

[17] Skinner, C. J., 2015. Cross-classified sampling: some estimation theory. Statistics and Probability Letters 104, 163–168.

[18] Strona, G., Nappo, D., Boccacci, F., Fattorini, S., San-Miguel-Ayanz, J., 2014. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. Nature communications 5, 4114.

[19] Tillé, Y., 2006. Sampling Algorithms. New York: Springer Science&Business Media.

[20] Tillé, Y., Matei, A., 2015. sampling: Survey Sampling. R package version 2.7.
URL `http://CRAN.R-project.org/package=sampling`

[21] von Gagern, M., von Gagern, M., Schmitz Ornès, A., 2015. Problems with bins: A critical reassessment of gotelli and ulrich's bayes approach using bird data. Acta Oecologica 69, 137–145.

[22] Vos, J. W. E., 1964. Sampling in space and time. Review of the International Statistical Institute 32, 226–241.

Appendix

Proof of Proposition 1: Since $Z_{ij}(1-Z_{i\ell})(1-Z_{kj})Z_{kl} = Z_{ij}Z_{kl} - Z_{ij}Z_{kl}Z_{i\ell} - Z_{ij}Z_{kl}Z_{kj} + Z_{ij}Z_{kl}Z_{i\ell}Z_{kj}$, the number of checkerboard 2×2 matrix is

$$\begin{aligned} C &= \frac{1}{2}\sum_{i,k=1}^{N}\sum_{j,\ell=1}^{M}\left\{Z_{ij}Z_{kl} - Z_{ij}Z_{kl}Z_{i\ell} - Z_{ij}Z_{kl}Z_{kj} + Z_{ij}Z_{kl}Z_{i\ell}Z_{kj}\right\}, \\ &= \frac{1}{2}\sum_{j,\ell=1}^{M}(Z_{\bullet j} - C_{j\ell})(Z_{\bullet \ell} - C_{j\ell}), \end{aligned}$$

where $C_{j\ell} = \sum_{i=1}^{N} Z_{ij}Z_{i\ell}$ is the number of joint occurrences in the samples for columns j and ℓ if $j \neq \ell$ and $C_{jj} = Z_{\bullet j} = n$. Thus

$$\begin{aligned} E(C) &= \frac{1}{2}\sum_{j\neq\ell=1}^{M}\{Z_{\bullet j} - E(C_{j\ell})\}\{Z_{\bullet \ell} - E(C_{j\ell})\} + \frac{1}{2}\sum_{j\neq\ell=1}^{M}\mathrm{Var}(C_{j\ell}), \\ &> \frac{M(M-1)}{2}\{n - E(C_{12})\}^2, \end{aligned}$$

as $E(C_{j\ell})$ is the same for all pairs of columns. Using result i) in Proposition 2 the probability that, in row i, columns 1 and 2 are sampled is $m_i(m_i-1)/\{M(M-1)\}$, one has

$$E(C_{12}) = \sum_{i=1}^{N}\frac{m_i(m_i-1)}{M(M-1)} = \sum_{i=1}^{N}\frac{m_i^2}{M(M-1)} - \frac{n}{M-1}.$$

Thus

$$E(C) > \frac{M}{2(M-1)}\left(Mn - \sum_{i=1}^{N}\frac{m_i^2}{M}\right)^2.$$

Proof of Proposition 3: We have

$$\mathrm{Cov}(\bar{y}_{i,s}, \bar{y}_{k,s}) = \frac{1}{m_i m_k}\left(\sum_{j=1}^{M} y_{ij}y_{kj}\mathrm{Cov}(Z_{ij}, Z_{kj}) + \sum_{j\neq\ell}^{M} y_{ij}y_{k\ell}\mathrm{Cov}(Z_{ij}, Z_{k\ell})\right).$$

Now, using Proposition 2, $\mathrm{Cov}(Z_{ij}, Z_{kj}) = \gamma_{ik} - m_i m_k/M^2$ and $\mathrm{Cov}(Z_{ij}, Z_{k\ell}) =$

$-\mathrm{Cov}(Z_{ij}, Z_{kj})/(M-1)$, for $j \neq \ell$. Thus

$$\mathrm{Cov}(\bar{y}_{i,s}, \bar{y}_{k,s}) = \frac{\mathrm{Cov}(Z_{ij}, Z_{kj})}{m_i m_k} \frac{M}{M-1} \sum_{j=1}^{M} (y_{ij} - \bar{y}_{iU})(y_{kj} - \bar{y}_{kU}) = \left(\frac{M\gamma_{ik}}{m_i m_k} - \frac{1}{M} \right) S_{ik}.$$

From (4), the variance of the Horvitz-Thompson estimator is

$$\mathrm{Var}(\widehat{\bar{y}}) \quad = \quad \frac{1}{N^2} \left\{ \sum_{i=1}^{N} \mathrm{Var}(\bar{y}_{is}) + \sum_{i \neq k}^{N} \mathrm{Cov}(\bar{y}_{is}, \bar{y}_{ks}) \right\} = \frac{\mathrm{tr}(\mathbf{\Sigma\Delta})}{N^2}.$$

Derivation of the joint probabilities presented in Table 2: The frequencies of the $3K$ column samples must satisfy

$$\begin{aligned} n_{1,1,0,0} + n_{1,0,1,0} + n_{1,0,0,1} &= n_{1,1,0,0} + n_{0,1,1,0} + n_{0,1,0,1} = K, \\ n_{1,0,0,1} + n_{0,1,0,1} + n_{0,0,1,1} &= n_{1,0,1,0} + n_{0,1,1,0} + n_{0,0,1,1} = 2K. \end{aligned}$$

These constraints can be reformulated as $n_{0,0,1,1} = K + n_{1,1,0,0}$, $n_{1,0,1,0} = n_{0,1,0,1}$, $n_{1,0,0,1} = n_{0,1,1,0}$. Thus the possible number of matrices $\mathbf{Z}$ is completely determined by $(n_{1,1,0,0}, n_{1,0,1,0})$. The total number of matrices with $n_{1,1,0,0} = i$ and $n_{1,0,1,0} = j$, with $i + j \leq K$, obtained by permuting the $3K$ columns of $\mathbf{Z}$ is equal to $(3K)!/\{i!(i+K)!(j!)^2\{(K-i-j)!\}^2\}$, as permuting columns that are equal does not change $\mathbf{Z}$. Summing gives

$$\mathcal{N} \quad = \quad \sum_{i+j \leq K} \frac{(3K)!}{i!(i+K)!(j!)^2\{(K-i-j)!\}^2}.$$

For $K = 1$, one gets $\mathcal{N} = 3! \cdot (1/2 + 1 + 1) = 15$ as found earlier. In a single matrix, the number of occurrences of $\{1, 2\}$ is $i = n_{1,1,0,0}$ and the joint probability $\gamma_{12}^{(3K)}$ is the total number of occurrences of $(1, 2)$ divided by the total number of columns, $3K \cdot \mathcal{N}$. This is given by

$$\gamma_{12}^{(3k)} \quad = \quad \frac{\sum_{i+j \leq K} i/[i!(i+K)!(j!)^2\{(K-i-j)!\}^2]}{3K \ \sum_{i+j \leq K} 1/[i!(i+K)!(j!)^2\{(K-i-j)!\}^2]}. \qquad (12)$$

For $K = 1$, one has $\gamma_{12}^{(3)} = 0.5/\{3\cdot(0.5{+}1{+}1)\} = 1/15$, as calculated earlier. The formula for $\gamma_{13}^{(3K)}$ is similar. Since $n_{0,0,1,1} = K + n_{1,1,0,0}$, $\gamma_{34}^{(3K)} = \gamma_{12}^{(3K)} + 1/3$.
515 Table 2 gives the values of $\gamma_{ij}^{(3K)}$ for various values of K, obtained by evaluating expressions such as (12).

Supplementary Material for "Sampling a two dimensional matrix"

Louis-Paul Rivest & Sergio Ewane Ebouele

Department of Mathematics and Statistics, Université Laval, 1045 avenue de la médecine, Quebec, QC, G1V 0A6 Canada

4 A detailed example

The 15 possible 4×3 sample matrices are

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \mathbf{Z}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \mathbf{Z}_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix},$$

$$\mathbf{Z}_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \mathbf{Z}_5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad \mathbf{Z}_6 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix},$$

$$\mathbf{Z}_7 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad \mathbf{Z}_8 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad \mathbf{Z}_9 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix},$$

$$\mathbf{Z}_{10} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad \mathbf{Z}_{11} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad \mathbf{Z}_{12} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\mathbf{Z}_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad \mathbf{Z}_{14} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad \mathbf{Z}_{15} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

The total number of 2×2 sub-matrices that can be investigated for swapping is $M(M-1)N(N-1)/4 = 18$. For the first three $\mathbf{Z}$ matrices a swap is possible for 8 sub-matrices while the other matrices only have 6 allowable swaps. The 15×15 transition matrix for the Markov chain of Section 2.2 is given by

$$P = \frac{1}{18} \left(\begin{array}{ccc|cccccc|cccccc} 10 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 10 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 10 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 12 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 12 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 12 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 12 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 12 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 12 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 12 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 12 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 12 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 12 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 12 \end{array} \right)$$

Row i gives the vector of probabilities for going from $\mathbf{Z}_i$ to $\mathbf{Z}_j$, $j = 1, \ldots, 15$. For instance the entry 1 at position $(10, 4)$ means that it is possible to go from $\mathbf{Z}_{10}$ to $\mathbf{Z}_4$, provided that rows 3 and 4 and columns 1 and 2 are selected for swapping when the chain is in state $\mathbf{Z}_{10}$. To check whether the burn-in

of $T = 21$ is large enough, we evaluate $P^{21} - 1/15$. This is equal to

$$10^{-3}\begin{pmatrix}
0.7 & -0.3 & -0.3 & 0.3 & 0.3 & 0.3 & 0.3 & -0.7 & -0.7 & 0.3 & 0.3 & 0.3 & 0.3 & -0.7 & -0.7 \\
-0.3 & 0.7 & -0.3 & 0.3 & -0.7 & 0.3 & -0.7 & 0.3 & 0.3 & 0.3 & -0.7 & 0.3 & -0.7 & 0.3 & 0.3 \\
-0.3 & -0.3 & 0.7 & -0.7 & 0.3 & -0.7 & 0.3 & 0.3 & 0.3 & -0.7 & 0.3 & -0.7 & 0.3 & 0.3 & 0.3 \\
\hline
0.3 & 0.3 & -0.7 & 1.1 & -0.1 & 0.3 & -0.5 & -0.5 & -0.1 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 & -0.4 \\
0.3 & -0.7 & 0.3 & -0.1 & 1.1 & -0.5 & 0.3 & -0.1 & -0.5 & -0.4 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 \\
0.3 & 0.3 & -0.7 & 0.3 & -0.5 & 1.1 & -0.1 & -0.1 & -0.5 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 & -0.4 \\
0.3 & -0.7 & 0.3 & -0.5 & 0.3 & -0.1 & 1.1 & -0.5 & -0.1 & -0.4 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 \\
-0.7 & 0.3 & 0.3 & -0.5 & -0.1 & -0.1 & -0.5 & 1.1 & 0.3 & -0.4 & -0.4 & -0.4 & -0.4 & 0.7 & 0.7 \\
-0.7 & 0.3 & 0.3 & -0.1 & -0.5 & -0.5 & -0.1 & 0.3 & 1.1 & -0.4 & -0.4 & -0.4 & -0.4 & 0.7 & 0.7 \\
\hline
0.3 & 0.3 & -0.7 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 & -0.4 & 1.1 & -0.1 & 0.3 & -0.5 & -0.5 & -0.1 \\
0.3 & -0.7 & 0.3 & -0.4 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 & -0.1 & 1.1 & -0.5 & 0.3 & -0.1 & -0.5 \\
0.3 & 0.3 & -0.7 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 & -0.4 & 0.3 & -0.5 & 1.1 & -0.1 & -0.1 & -0.5 \\
0.3 & -0.7 & 0.3 & -0.4 & 0.7 & -0.4 & 0.7 & -0.4 & -0.4 & -0.5 & 0.3 & -0.1 & 1.1 & -0.5 & -0.1 \\
-0.7 & 0.3 & 0.3 & -0.4 & -0.4 & -0.4 & -0.4 & 0.7 & 0.7 & -0.5 & -0.1 & -0.1 & -0.5 & 1.1 & 0.3 \\
-0.7 & 0.3 & 0.3 & -0.4 & -0.4 & -0.4 & -0.4 & 0.7 & 0.7 & -0.1 & -0.5 & -0.5 & -0.1 & 0.3 & 1.1
\end{pmatrix}$$

Each row of P^{21} gives its own evaluation of the joint selection probabilities $\{\gamma_{ik}\}$. That for γ_{12} is, for instance, the average of the entries in the first three columns. The largest relative error between these evaluations and their true values is 2.5×10^{-4}. Thus the proposed burn-in is adequate for this small problem.

Some R-code for Section 4

Construction of the matrix P and evaluation of $P^{21} - 1/15$:

```
trans<-matrix(0,15,15);diag(trans)<-c(rep(10/18,3),rep(12/18,12))
trans[1,c(4,5,6,7,10,11,12,13)]<-1/18;trans[2,c(4,6,8,9,10,12,14,15)]<-1/18
trans[3,c(5,7,8,9,11,13,14,15)]<-1/18;trans[4,c(1,2,5,9,10,12)]<-1/18
trans[5,c(1,3,4,8,11,13)]<-1/18;trans[6,c(1,2,7,8,10,12)]<-1/18
trans[7,c(1,3,6,9,11,13)]<-1/18;trans[8,c(2,3,5,6,14,15)]<-1/18
trans[9,c(2,3,4,7,14,15)]<-1/18;trans[10,c(1,2,4,6,11,15)]<-1/18
trans[11,c(1,3,5,7,10,14)]<-1/18;trans[12,c(1,2,4,6,13,14)]<-1/18
trans[13,c(1,3,5,7,12,15)]<-1/18;trans[14,c(2,3,8,9,11,12)]<-1/18
```

```
trans[15,c(2,3,8,9,10,13)]<-1/18
#
xx<-eigen(trans)
trans21<-xx$vector%*%diag(xx$values)^21%*%t(xx$vector)
#
# Evaluation of the relative errors for the estimation
#of $\{\gamma_{ik}\}$ obtained with $M^{21}$
#
vec12<-c(1,1,1,rep(0,12))
max(abs(trans21%*%vec12/3-rep(1,15)/15))*15 #  RE 1.6e-7% for gamma_12
vec13<-c(rep(0,3),rep(1,6),rep(0,6))
max(abs(trans21%*%vec13/3-rep(2,15)/15))*(15/2)  #RE 2.5e-4% for gamma_13
vec14<-c(rep(0,9),rep(1,6))
max(abs(trans21%*%vec14/3-rep(2,15)/15))*(15/2) #RE 2.5e-4% for gamma_14
vec34<-c(rep(2,3),rep(1,6),rep(1,6))
max(abs(trans21%*%vec34/3-rep(6,15)/15))*(15/6) #RE 2.7e-08 for gamma_34
```

5.1 A Creel survey for stripped bass

The data set is given in Table 1.

Table 1: Creel survey data set analysed in Section 5. This gives the fishing effort (FE) measured on 72 day×site.

Site	Day	FE	Site	Day	FE	Site	Day	FE
1	4	8.283	3	17	16.986	5	36	3.043
1	5	6.416	3	18	3.661	6	4	6.771
1	12	4.662	3	20	13.541	6	7	2.391
1	19	2.715	3	25	12.603	6	8	1.862
1	20	3.644	3	26	3.766	6	10	5.904
1	21	5.123	3	33	0.701	6	21	5.128
1	25	25.088	3	34	7.646	6	26	1.229
1	30	19.464	4	1	7.243	7	6	2.011
1	33	67.688	4	3	10.177	7	11	4.877
1	34	3.941	4	6	15.187	7	22	9.754
2	1	4.412	4	9	9.576	7	24	3.508
2	3	4.696	4	11	19.688	7	29	2.777
2	9	2.388	4	13	32.779	7	35	3.998
2	14	3.516	4	14	14.435	8	7	0.865
2	19	8.300	4	16	10.236	8	10	2.664
2	22	5.928	4	27	2.963	8	15	5.178
2	24	3.751	4	32	5.378	8	28	0.633
2	27	3.990	4	35	7.115	8	31	1.531
2	29	4.353	5	2	1.091	8	36	0.800
2	30	8.713	5	13	3.498	9	2	3.839
2	32	15.149	5	16	2.781	9	15	1.385
3	5	7.875	5	23	1.278	9	17	2.402
3	8	9.690	5	28	17.733	9	18	5.608
3	12	6.206	5	31	4.942	9	23	3.641

Some R-code for calculating the Horvitz-Thompson estimate and the four variance estimates of Section 3.2

```
# Definition of the covariance function
#
 covar<-function(x,y){sum((x-mean(x))*(y-mean(y)))/(length(x)-1)}
 #
EstiS2<-function(data , echan, Delta){
# This function calculates the Horvitz-Thompson estimate (4)
# and the 4 variance estimates of Section 3.3
# data is the observed NxM data matrix with zeros for
# units that have not been sampled
# echan is the NxM matrix of sample indicators
# Delta is NxN matrix entering the variance calculations
  datav=as.vector(t(data))[as.vector(t(echan))==1]
M=dim(echan)[2];N=dim(echan)[1]; mi<-rowSums(echan);n=sum(mi)/M
Pselect<-outer(mi/M,rep(1,M)) # NxM matrix of selection probabilities
    HT<-(1/(M*N))*sum(data*echan/Pselect)   #  HT estimator
# Estimation of the NxN y variance covariance matrix using (8)
    sigma<-matrix(0,N,N)
    for (i in (1:(N-1))){
      sigma[i,i]<-var(data[i,echan[i,]==1])
      for (j in (i+1):N){
    if(sum(echan[j,]*echan[i,]==1)>1){
sigma[j,i]<-covar(data[i,echan[j,]*echan[i,]==1],data[j,echan[j,]*echan[i,]==1])}
     sigma[i,j]<-sigma[j,i] }
      sigma[9,9]<-var(data[9,echan[9,]==1])
    }
# Calculation for the correlation rho and of
# the equal correlation variance covariance matrix
#
    rho<-0; rhonum=rhoden=0
    for (i in (1:(N-1))){
      for (j in (i+1):N){
        if(sum(echan[j,]*echan[i,]==1)>1){rhonum<-rhonum+sigma[i,j];
        rhoden<- rhoden+sqrt(sigma[i,i]*sigma[i,i])}
      }}
rho=rhonum/rhoden
```

```
dvar<-diag(sqrt(diag(sigma)))
sigmaec<-dvar%*%((1-rho)*diag(rep(1,N))+rho*outer(rep(1,N),rep(1,N)))%*%dvar
varPI<-sum(diag(Delta%*%sigma))/N^2;   varec<-sum(diag(Delta%*%sigmaec))/N^2
varstr<-sum(diag(Delta)*diag(sigma))/N^2
#
# Construction of the balancing variables and evaluation of the
# Deville Tille variance estimator
#
cube.data<-cbind(kronecker(diag(mi),rep(1,M)),kronecker(mi,diag(1,M)))
dataDE<-cbind(datav,cube.data[as.vector(t(echan))==1,])
prob2<-Pselect[as.vector(t(echan))==1]
xx<-lm(dataDE[,1]~-1+ dataDE[,-c(1,2)],weight=(1-prob2)/prob2^2)
varDT<-(n*M/(N^2*M^2))*sum(xx$residuals^2*(1-prob2)/prob2^2)/(M*n-M-N+1)
out<-c(HT,varstr,varPI,varec,varDT); out
}
```

5.2 Comparisons of three sampling algorithms of Section 2

Table 2 gives the Monte Carlo estimates for the joint selection probabilities obtained with the three sampling algorithms in a design with $M = 72$, $N = 9$, $n = 2$ and an m vector given by $(20, 22, 20, 22, 14, 12, 12, 12, 10)$. All the z-statistics comparing MCMC and hypergeometric joint selection probabilities are less than 2 in absolute value. The `BalancedSampling` implementation of the flight phase for balanced sampling stops when the number of undecided units is equal to the number of constraints, that is $N + M$ for our problem, and the sample obtained at the end might not meet the required constraints. To overcome this problem we reran the flight phase several times after removing useless constraints; we are grateful to Yves Tillé for suggesting an algorithm that does this. Still, in 16% of the simulations, the cube method sample did not meet the row and column constraints. The cube method empirical joint selection probabilities reported in Table 2 were evaluated using the samples that met the constraints only. Among

the z statistics comparing the hypergeometric and the cube method joint selection probabilities five z statistics are large than 2 in absolute value in Table 2, showing small differences between the hypergeometric and the cube method algorithms for this problem.

Table 2: Evaluation of the joint selection probabilities $\{\gamma_{ik}\}$ using three sampling algorithms in a 9×72 population matrix and test statistics comparing hypergeometric estimates to those obtained with MCMC (z_{12}) and the cube method (z_{13}).

		Hypergeometric		MCMC		Cube Method			
i	k	γ_{ik}^{hyp}	v^{hyp}	γ_{ik}^{MCMC}	v^{MCMC}	γ_{ik}^{bal}	v^{bal}	z_{12}	z_{13}
1	2	0.05208	0.00048	0.05204	0.00048	0.05162	0.00046	0.11686	1.44045
1	3	0.04554	0.00043	0.04608	0.00043	0.04554	0.00043	-1.84128	0.00884
1	4	0.05144	0.00046	0.05106	0.00046	0.05148	0.00046	1.27044	-0.13553
1	5	0.03044	0.00032	0.03038	0.00031	0.03071	0.00032	0.23203	-1.00298
1	6	0.02572	0.00028	0.02557	0.00027	0.02591	0.00028	0.63445	-0.76115
1	7	0.02602	0.00027	0.02590	0.00027	0.02607	0.00027	0.53521	-0.18490
1	8	0.02547	0.00027	0.02566	0.00027	0.02607	0.00027	-0.83025	-2.46364
1	9	0.02106	0.00022	0.02108	0.00023	0.02120	0.00023	-0.10451	-0.61187
2	3	0.05195	0.00047	0.05160	0.00047	0.05164	0.00046	1.13200	0.96643
2	4	0.05770	0.00052	0.05773	0.00051	0.05811	0.00050	-0.09539	-1.21472
2	5	0.03375	0.00034	0.03401	0.00034	0.03423	0.00034	-1.01330	-1.76194
2	6	0.02889	0.00030	0.02867	0.00029	0.02910	0.00030	0.88512	-0.82651
2	7	0.02882	0.00029	0.02859	0.00030	0.02901	0.00030	0.94025	-0.72930
2	8	0.02879	0.00030	0.02926	0.00029	0.02871	0.00029	-1.92306	0.32219
2	9	0.02358	0.00025	0.02365	0.00026	0.02397	0.00025	-0.29170	-1.64693
3	4	0.05127	0.00046	0.05159	0.00047	0.05213	0.00047	-1.06771	-2.70378
3	5	0.03067	0.00032	0.03065	0.00031	0.03062	0.00031	0.07739	0.17481
3	6	0.02556	0.00027	0.02565	0.00027	0.02595	0.00028	-0.39401	-1.59799
3	7	0.02591	0.00027	0.02557	0.00027	0.02580	0.00028	1.43491	0.46046
3	8	0.02573	0.00027	0.02567	0.00027	0.02580	0.00027	0.27531	-0.29547
3	9	0.02115	0.00023	0.02096	0.00023	0.02103	0.00023	0.89179	0.54299
4	5	0.03469	0.00035	0.03461	0.00035	0.03393	0.00034	0.30941	2.75851
4	6	0.02898	0.00030	0.02922	0.00030	0.02896	0.00030	-0.98114	0.08750
4	7	0.02865	0.00030	0.02897	0.00030	0.02914	0.00030	-1.31829	-1.89553
4	8	0.02897	0.00030	0.02877	0.00030	0.02898	0.00030	0.80309	-0.04233
4	9	0.02385	0.00025	0.02359	0.00025	0.02367	0.00025	1.13139	0.76791
5	6	0.01718	0.00019	0.01709	0.00019	0.01696	0.00019	0.40830	1.07582
5	7	0.01671	0.00019	0.01689	0.00019	0.01733	0.00020	-0.95795	-3.01811
5	8	0.01703	0.00020	0.01695	0.00019	0.01727	0.00020	0.42402	-1.16704
5	9	0.01398	0.00016	0.01385	0.00016	0.01412	0.00016	0.71117	-0.72752
6	7	0.01422	0.00017	0.01429	0.00017	0.01425	0.00017	-0.40278	-0.16237
6	8	0.01440	0.00017	0.01433	0.00016	0.01436	0.00017	0.40473	0.20450
6	9	0.01172	0.00014	0.01183	0.00014	0.01185	0.00014	-0.67013	-0.75507
7	8	0.01453	0.00017	0.01427	0.00016	0.01411	0.00016	1.40614	2.23663
7	9	0.01181	0.00014	0.01217	0.00014	0.01163	0.00014	-2.16049	1.05412
8	9	0.01175	0.00014	0.01176	0.00014	0.01195	0.00014	-0.06681	-1.11958

Table 3 compares three sets of estimates of the joint selection probabilities for the creel survey example. One is obtained using the hypergeometric algorithm and uses $M = 72$ (as reported in Table 2). The other two used the MCMC sampling algorithm to simulate matrices with $N = 9$, $M = 144$ ($M = 288$) and single unit selection probabilities as given in the creel example of Section 5.1. The three sets of selection probabilities are not significantly different as only 3 of the 72 absolute z statistics are larger than 2. This suggests that the MCMC burn-in given in equation (3) is adequate for large problems

Some R-code for Section 5.2

```
# The R and C++ code for the implementation of the MCMC algorithm
# of Section 2.2 is available from the authors
#
# HYPERGEOMETRIC SAMPLING
#
rmat<-function(R,S, bb=1000)
  # Fonction that simulates an array of bb random 0-1 matrices
  # using hypergeometric sampling
  # R is the vector or row totals
  # S is the vector of column totals
  # The two vectors sum to the same total
{
  m<-length(R)
  cR<-c(0,cumsum(R))
  cS<-c(0,cumsum(S))
  n<-length(S)
  nn<-sum(R)
  matR<-matrix(0,m,nn)
  matS<-matrix(0,n,nn)
  for (i in (1:m)){matR[i,(cR[i]+1):(cR[i+1])]<-1}
  for (i in (1:n)){matS[i,(cS[i]+1):(cS[i+1])]<-1}
  mat0<-diag(rep(1,nn))[sample(nn),]
  mat1<-matR%*%mat0%*%t(matS)
```

Table 3: Comparison of three evaluations of the joint selection probabilities $\{\gamma_{ik}\}$: the first one is obtained by sampling 9×72 matrices with the hypergeometric algorithm, the second and the third ones are obtained by simulating matrices with $M = 144$ and $M = 288$ columns using the MCMC algorithm and test statistics comparing hypergeometric estimates to those obtained with the $M = 144$ (z_{12}) and $M = 288$ (z_{13}) MCMC algorithm

		Hypergeometric		MCMC, $M = 144$		MCMC, $M = 288$			
i	k	γ_{ik}^{hyp}	v^{hyp}	γ_{ik}^{144}	v^{144}	γ_{ik}^{288}	v^{288}	z_{12}	z_{13}
1	2	0.05208	0.00048	0.05170	0.000241	0.05173	0.000118	1.41243	1.43097
1	3	0.04554	0.00043	0.04582	0.000215	0.04586	0.000104	-1.12193	-1.38392
1	4	0.05144	0.00046	0.05155	0.000233	0.05163	0.000116	-0.41772	-0.78681
1	5	0.03044	0.00032	0.03045	0.000152	0.03044	0.000078	-0.03325	0.00208
1	6	0.02572	0.00028	0.02568	0.000137	0.02565	0.000067	0.21352	0.40194
1	7	0.02602	0.00027	0.02567	0.000133	0.02561	0.000066	1.73639	2.22205
1	8	0.02547	0.00027	0.02570	0.000135	0.02566	0.000067	-1.15325	-1.05122
1	9	0.02106	0.00022	0.02120	0.000111	0.02120	0.000057	-0.79585	-0.83465
2	3	0.05195	0.00047	0.05176	0.000234	0.05187	0.000115	0.72226	0.32296
2	4	0.05770	0.00052	0.05779	0.000255	0.05772	0.000127	-0.32437	-0.07237
2	5	0.03375	0.00034	0.03418	0.000172	0.03423	0.000085	-1.89982	-2.31713
2	6	0.02889	0.00030	0.02874	0.000146	0.02901	0.000074	0.72540	-0.64092
2	7	0.02882	0.00029	0.02880	0.000148	0.02874	0.000074	0.10221	0.39590
2	8	0.02879	0.00030	0.02895	0.000147	0.02866	0.000072	-0.75012	0.65891
2	9	0.02358	0.00025	0.02364	0.000122	0.02359	0.000062	-0.32328	-0.03460
3	4	0.05127	0.00046	0.05169	0.000230	0.05162	0.000119	-1.58118	-1.46557
3	5	0.03067	0.00032	0.03042	0.000155	0.03042	0.000078	1.12777	1.23679
3	6	0.02556	0.00027	0.02574	0.000133	0.02558	0.000066	-0.91159	-0.11591
3	7	0.02591	0.00027	0.02578	0.000139	0.02568	0.000068	0.65073	1.22911
3	8	0.02573	0.00027	0.02553	0.000136	0.02570	0.000067	0.97955	0.14441
3	9	0.02115	0.00023	0.02103	0.000115	0.02103	0.000057	0.64348	0.69866
4	5	0.03469	0.00035	0.03422	0.000167	0.03432	0.000086	2.06063	1.78609
4	6	0.02898	0.00030	0.02882	0.000144	0.02891	0.000076	0.76220	0.34112
4	7	0.02865	0.00030	0.02892	0.000150	0.02884	0.000074	-1.28609	-0.97912
4	8	0.02897	0.00030	0.02887	0.000150	0.02882	0.000073	0.47427	0.78167
4	9	0.02385	0.00025	0.02370	0.000126	0.02370	0.000063	0.79168	0.86736
5	6	0.01718	0.00019	0.01710	0.000096	0.01694	0.000046	0.48980	1.57931
5	7	0.01671	0.00019	0.01710	0.000096	0.01698	0.000047	-2.28499	-1.74554
5	8	0.01703	0.00020	0.01711	0.000097	0.01710	0.000048	-0.49103	-0.45101
5	9	0.01398	0.00016	0.01386	0.000079	0.01402	0.000040	0.75485	-0.28143
6	7	0.01422	0.00017	0.01434	0.000084	0.01443	0.000041	-0.77622	-1.46474
6	8	0.01440	0.00017	0.01443	0.000084	0.01442	0.000043	-0.20054	-0.12624
6	9	0.01172	0.00014	0.01182	0.000071	0.01172	0.000035	-0.67578	-0.03258
7	8	0.01453	0.00017	0.01424	0.000083	0.01452	0.000042	1.79474	0.06057
7	9	0.01181	0.00014	0.01181	0.000071	0.01185	0.000035	-0.00287	-0.33137
8	9	0.01175	0.00014	0.01182	0.000068	0.01178	0.000034	-0.50507	-0.21054

```
   res<-array(0,c(m,n,bb))
  resi<-rep(0,bb)
  for(j in (1:bb)){
    mat0<-diag(rep(1,nn))[sample(nn),]
    mat1<-matR%*%mat0%*%t(matS)
    while(max(mat1)>1){
      mat0<-diag(rep(1,nn))[sample(nn),]
      mat1<-matR%*%mat0%*%t(matS)
    }
 res[,,j]<-mat1
 }
 res
}
#
#  BALANCED SAMPLING
#
#Call to BalancedSampling to simulate a data matrix with M=72 and N=9
#Calculation of the selection probabilities
mi<-c(10,11,10,11,7,6,6,6,5);proba<-rep(2*mi/72,each=72); N<-9;M<-72
# Evaluation of the balancing variables
cube.data<-cbind(kronecker(diag(2*mi),rep(1,M)),kronecker(2*mi,diag(1,M)))
# Flight phase
 pikstar=flightphase(proba,cube.data)
 #This has N+M undecided units with probabilities in (0,1)
 #We clean up the balancing variables to reduce this number
  EPS=0.00000001
  T1=rep(1,length(proba))
  T=pikstar>EPS & pikstar < 1-EPS
  ii=0
  while(sum(T)!=sum(T1))
  {
    ii=ii+1;CC=cube.data[T,]
    svde=svd(CC);
    cube.datan=svde$u[,svde$d>EPS]%*%diag(svde$d[svde$d>EPS])
    pikstar[T]=flightphase(pikstar[T],(pikstar[T]/proba[T])*cube.datan)
    T1=T; T=pikstar>EPS & pikstar < 1-EPS
  }
   # Landing phase
  sample=landingphase(proba,pikstar,cube.data)
  samp2<-matrix(sample,nrow=9,ncol=72,byrow=TRUE)
```