T110
30/8/88

~~1998~~ 88 4

15

Contents :-

INDIAN STATISTICAL INSTITUTE LIBRARY
No. 110
Date 30 AUG 1988
CALCUTTA

1. Introduction to 'Deep Stratification'

If the study variables are all related to single supplementary variable (x) for which information is available for all population units then stratification and allocation may be done in an optimum fashion using the data on x. But all the characteristics of interest may not be related to one supplementary variable but to two or more auxiliary variables. In such a situation, the units may first be grouped into primary strata with respect to the most important of the stratification variables and then within each of the primary strata so formed, secondary or sub-strata may be constructed according to another supplementary variable, and so on. This procedure is known as multiple stratification or deep stratification.

However, our problem is associated to two-way stratification of the population units.

Suppose a population consists of N units and these N units are grouped into p x q cells of a matrix of order p by q. So each cell constitutes one stratum. p denotes the number of levels of first stratification variable and q denotes the number of levels of second stratification variable.

Let b_{ij}, $i = 1(1)p$, $j = 1(1)q$ denote the number of population units in the (i,j)th cell. Let U_{ijk}, $k = 1(1)b_{ij}$

denote the k-th population unit of the (i, j)th cell. Our problem is to draw a sample of n units from the population considering the following restrictions :-

(i) MINR(i) $\leq$ total number of sample units in the i-th row $\leq$ MAXR(i), for all i = 1(1)p

(ii) MINC(j) $\leq$ total number of sample units in the j-th column $\leq$ MAXC(j) for all j = 1(1)q

where MINR(i), MAXR(i), MINC(j), MAXC(j), n are given.

So our problem is to select a matrix $\underset{\sim}{A} = ((a_{ij}))$, i = 1(1)p, j = 1(1)q following the restrictions :-

(A) $\text{MINR}(i) \leq \sum_{j=1}^{q} a_{ij} \leq \text{MAXR}(i)$, for all i = 1(1)p

(B) $\text{MINC}(j) \leq \sum_{i=1}^{p} a_{ij} \leq \text{MAXC}(j)$, for all j = 1(1)q

(C) $a_{ij} \leq b_{ij} \; \forall \; i = 1(1)p, \; j = 1(1)q$

and

(D) $\sum_{i=1}^{p} \sum_{j=1}^{q} a_{ij} = n$ (given).

We may get a large number of solution matrices $\underset{\sim}{A}$ satisfying the above criteria. Suppose PNO denotes the possible number of solution matrices. Select one matrix out of PNO matrices randomly.

For a selected matrix $\underset{\sim}{A} = ((a_{ij}))$, if $a_{ij} > 0$, draw a sample of a_{ij} units from b_{ij} population units of the (i,j)th cell with SRSWOR sampling scheme, $i = 1(1)p$, $j = 1(1)q$.

Suppose P_i is the probability that one particular unit of the cell numbered i, will be included in the sample and P_{ij} is the probability that one particular unit of the cell numbered i and another particular unit of the cell numbered j will be included in the sample. The (i,j)th cell will be numbered as s, where s will be obtained by the formula

$$s = q(i - 1) + j.$$

An unbiased estimate for the population total and an unbiased estimate of the variance of the unbiased estimator of population total can be obtained by Horvitz-Thompson method using P_i's and P_{ij}'s

2. Description about the computer program :

2.1 Objectives :

The computer program has been written in PASCAL language to get the following things :

(i) to generate all possible solution matrices

(ii) to compute the inclusion probabilities P_i's and P_{ij}'s

(iii) to select one of the solution matrices randomly

(iv) to select sample units corresponding to the selected solution matrix

(v) to estimate population total and also an estimate of the variance of the estimator.

2.2 Input data :

(i) row-dimension = p, column-dimension = q

(ii) row-minimum(i) = MINR(i), row-maximum(i) = MAXR(i), $\forall i = 1(1)p$

(iii) column-minimum(j) = MINC(j), column-maximum(j) = MAXC(j), $\forall j = 1(1)q$

(iv) total = n

(v) row-priority(i), $\forall$ i = 1(1)p

(vi) column-priority(j), $\forall$ j = 1(1)q

(vii) population-matrix (i, j) = b_{ij}, if $b_{ij} > 0$ then enter the value of Y_{ijk}, $k = 1(1)b_{ij}$, $\forall\ i = 1(1)p$, $j = 1(1)q$.

Where Y_{ijk} denotes the value of the k-th population unit of the (i, j)th cell.

2.3 Output :

(i) possible number of generated matrices

(ii) tabulation of inclusion probabilities P_i's and P_{ij}'s

(iii) randomly selected solution matrix $\underset{\sim}{A}$

(iv) listing of population units

(v) listing of sample units

(vi) estimate of population total and population mean

(vii) exact population total and population mean

(viii) estimate of variance of the estimator of population total and population mean

(ix) exact variance of the estimator of population total and population mean.

2.4 Subprograms used :

Three main procedures have been used in this program viz., 'matrix-stratification', 'row-stratification', 'column-stratification'.

If you call the procedure 'matrix-stratification' then you will get the output under two-way (deep) stratification. If you call the procedure 'row-stratification' then you will get the output under one-way (row) stratification (each row constitutes a single stratum). If you call the procedure 'column-stratification' then you will get the output under one-way (column) stratification (each column constitutes a single stratum).

The above three main procedures involve the following procedures :

(i) iexact-row

(ii) iexact-column

(iii) vector

(iv) incr

(v) incrrr-row

(vi) incrrr-column

(vii) transformation-matrix

(viii) inverse-matrix

(ix) matrix-print

(x) random

(xi) select-matrix [inside the procedure 'matrix-stratification']

(xii) check-on-piei [inside the procedure 'matrix-stratification']

(xiii) check-on-pieij [inside the procedure 'matrix-stratification']

(xiv) select-row [inside the procedure 'row-stratification']

(xv) check-on-piei [inside the procedure 'row-stratification']

(xvi) select-column [inside the procedure 'column-stratification']

(xvii) check-on-piei [inside the procedure 'column-stratification']

3. Method to generate all possible solution matrix $\underset{\sim}{A}$:

We have to generate all possible matrices $\underset{\sim}{A} = ((a_{ij}))$ such that the following conditions hold :

(i) $\text{MINR}(i) \leq \sum_{j=1}^{q} a_{ij} \leq \text{MAXR}(i), \quad \forall\ i = 1(1)p$

(ii) $\text{MINC}(j) \leq \sum_{i=1}^{p} a_{ij} \leq \text{MAXC}(j), \quad \forall\ j = 1(1)q$

(iii) $\sum_{i=1}^{p} \sum_{j=1}^{q} a_{ij} = n$

(iv) $a_{ij} \leq b_{ij}\ , \quad \forall\ i = 1(1)p, \quad j = 1(1)q$

where MINR(i), MAXR(i), MINC(j), MAXC(j), n, $((b_{ij}))$ are given through input.

Suppose $\underset{\sim}{C} = (c_1 c_2 \ldots\ldots c_s)$ and $\underset{\sim}{D} = (d_1 d_2 \ldots\ldots d_s)$ are two vectors such that all c_i's and d_i's are non-negative integers and $\sum_{i=1}^{s} c_i = \sum_{i=1}^{s} d_i$. We shall say that the vector $\underset{\sim}{C}$ is greater than the vector $\underset{\sim}{D}$ (or equivalently $\underset{\sim}{D}$ is less than $\underset{\sim}{C}$) if $\quad\quad i \in \{1, 2, 3, \ldots\ldots, s\}$ such that $c_i > d_i$ and $c_k = d_k\ \forall\ k < i$.

We shall say that $\underset{\sim}{C}$ is 'just greater' than $\underset{\sim}{D}$ if it is not possible to find a vector $\underset{\sim}{V}$ such that $\underset{\sim}{C}$ is greater than $\underset{\sim}{V}$ and $\underset{\sim}{V}$ is greater than $\underset{\sim}{D}$.

If the vector $\underset{\sim}{C}$ is just greater than $\underset{\sim}{D}$ then denote it by the notation $\underset{\sim}{C} \gtrsim \underset{\sim}{D}$.

Step - I

Call the procedure 'iexact-row', which will give an output vector $\underset{\sim}{R} = (R_1 R_2 \ldots\ldots R_p)$ satisfying the conditions

$$\left.\begin{array}{ll} & \underline{\text{MINR}}(i) \leq R_i \leq \underline{\text{MAXR}}(i), \quad \forall\ i = 1(1)p \\ \text{and} & \sum_{i=1}^{p} R_i = n \end{array}\right\} \quad \ldots \quad (A)$$

This vector $\underset{\sim}{F}$ is the minimal in the sense that there is no other vector $\underset{\sim}{R}'$ which is less than $\underset{\sim}{R}$ and which satisfies the conditions given by (A).

Step - II

Set i = 1 and call the procedure 'vector' by fixing the argument 'it' = R_i, which will give an output vector $\underset{\sim}{X}_i = (x_{i1}\ x_{i2} \ldots\ldots x_{iq})$ satisfying the conditions :

$$\left.\begin{array}{ll} & \sum_{j=1}^{q} x_{ij} = R_i \\ \text{and} & x_{ij} \leq b_{ij} \quad \forall\ j = 1(1)q \end{array}\right\} \quad \ldots \quad (B)$$

This vector $(x_{i1}\ x_{i2} \ldots\ldots x_{iq})$ is the minimal in the sense that there is no other vector $(x'_{i1}\ x'_{i2} \ldots\ldots x'_{iq})$ which is less than $(x_{i1}\ x_{i2}\ \ldots.\ x_{iq})$ and satisfies the conditions given by (B).

Step - III

If i = p then go to Step VI.

Increase i by 1 and call the procedure 'vector' by fixing the argument 'aux' = R_i. If it is possible to find a vector $(x_{i1}\ x_{i2}\ \ldots.\ x_{iq})$ (minimal) satisfying (B) then go to Step - III otherwise
set $(x_{i1}\ x_{i2}\ \ldots.\ x_{iq}) = (0\ 0\ \ldots.\ 0)$ and go to Step IV.

Step - IV

If i = 1 then go to Step - V.

Decrease i by 1 and call the procedure 'incr' taking arguments 'aux' = $\underset{\sim}{X}_i$ and 'it' = R_i. The procedure 'incr' will try to find a vector $\underset{\sim}{C}_i$ such that $\underset{\sim}{C}_i \geqslant \underset{\sim}{X}_i$. If it is possible to find a vector $(c_{i1}\ c_{i2}\ \ldots.\ c_{iq}) \geqslant (x_{i1}\ x_{i2}\ \ldots.\ x_{iq})$ then set $(x_{i1}\ x_{i2}\ \ldots.\ x_{iq}) = (c_{i1}\ c_{i2}\ \ldots.\ c_{iq})$ and go to Step - III otherwise set $(x_{i1}\ x_{i2}\ \ldots.\ x_{iq}) = (0\ 0\ \ldots.\ 0)$ and go to Step - IV.

Step - V

Call the procedure 'incrrr-row' setting the argument 'aux' = $\underset{\sim}{R}$. This procedure will try to find a vector $\underset{\sim}{D}$ such that $\underset{\sim}{D} \succcurlyeq \underset{\sim}{R}$. If such a vector $\underset{\sim}{D}$ can be found then set $\underset{\sim}{R} = \underset{\sim}{D}$ and go to the Step - II otherwise terminate the process.

Step - VI

So we get a matrix $\underset{\sim}{X} = ((x_{ij}))$, $i = 1(1)p$, $j = 1(1)q$. Now if $\text{MINC}(j) = \sum_{i=1}^{p} x_{ij} \leq \text{MAXC}(j)$, $\forall\ j = 1(1)q$ then this matrix $\underset{\sim}{X}$ is an occurrence of $\underset{\sim}{A}$. Set $\underset{\sim}{A} = \underset{\sim}{X}$. Setting $i = p$, call the procedure 'incr' taking arguments 'aux' $= \underset{\sim}{X}_i$ and 'it' $= R_i$. If we can find a vector $\underset{\sim}{C}$ such that $\underset{\sim}{C} \succcurlyeq \underset{\sim}{X}_i$ then set $\underset{\sim}{X}_i - \underset{\sim}{C}$ and go to Step - VI otherwise set $\underset{\sim}{X}_i = \underset{\sim}{0}$ and go to Step - IV.

DISS/88/01/4

STATISTICAL INSTITUTE LIBRARY
No. 110
Date 3.0 AUG 1988
CALCUTTA

4. Computation of P_s and P_{st} :

Suppose $\underset{\sim}{A}^{(k)} = ((a_{ij}^{(k)}))$, $i = 1(1)p$, $j = 1(1)q$ be the k-th generated solution matrix for $\underset{\sim}{A}$, $k = 1,2,3, \ldots, PNO$, where PNO denotes the number of all possible solution matrices. Suppose s and t denote respectively the cell number of (i,j)th and (i',j')th cell, i.e., s and t are related to i, j, i', j' by

$$s = q(i - 1) + j, \qquad t = q(i' - 1) + j' .$$

Then $$P_s = \frac{1}{PNO} \sum_{k=1}^{PNO} a_{ij}^{(k)}/b_{ij}$$

$$P_{st}^{(k)} = \frac{a_{ij}^{(k)}}{b_{ij}} \frac{(a_{ij}^{(k)} - 1)}{(b_{ij} - 1)} \text{ if } i = i', j = j' \text{ and } a_{ij}^{(k)} > 1$$

$$= 0 \text{ if } i = i', j = j' \text{ and } a_{ij}^{(k)} \leq 1$$

$$= \frac{a_{ij}^{(k)} \times a_{i'j'}^{(k)}}{b_{ij} \times b_{i'j'}} \text{ if } (i,j) \neq (i',j') \text{ and } a_{ij}^{(k)} \geq 1, \quad a_{i'j'}^{(k)} \geq 1$$

$$= 0 \text{ if } (i,j) \neq (i',j') \text{ and } a_{ij}^{(k)} = 0 \text{ or } a_{i'j'}^{(k)} = 0$$

$$P_{st} = \frac{1}{PNO} \sum_{k=1}^{PNO} P_{st}^{(k)}$$

5. Method of selection of sample units :

Suppose PNO is the number of possible solutions of $\underset{\sim}{A}$. Select a random number from 1 to PNO by calling the procedure 'random' with argument 'n' = PNO. Suppose the random number drawn is s. Select the s-th solution matrix $\underset{\sim}{A} = ((a_{ij}))$. Now for each cell (i,j), if $a_{ij} > 0$ then draw a sample of a_{ij} units from b_{ij} population units of the (i,j)th cell by SRSWOR sampling method by calling the procedure 'select-matrix' taking arguments i = i, j = j, k = a_{ij} .

[Note : For one-way(row) stratification call the procedure 'select-row' and for one-way(column) stratification call the procedure 'select-column'.]

6. Estimation of population total and mean :

U_{ijk} = k-th population unit of the (i,j)th cell,

$$k = 1, 2, \ldots, b_{ij}, \quad \sum_{i=1}^{p} \sum_{j=1}^{q} b_{ij} = N.$$

Y : variable under study.

Y_{ijk} : Y-value of U_{ijk}.

y_{ijk} : Y-value of the k-th sample unit of the (i,j)th cell

$$k = 1, 2, \ldots, a_{ij}, \quad \sum_{i=1}^{p} \sum_{j=1}^{q} a_{ij} = n.$$

$$T = \sum_{i=1}^{p} \sum_{j=1}^{q} \sum_{k=1}^{b_{ij}} Y_{ijk} = \text{population total.}$$

$\bar{Y} = T/N$ = population mean.

Consider a vector of sample units

$$\underset{\sim}{Z} = (y_{111}, y_{112}, \ldots, y_{11a_{11}}, y_{121}, y_{122}, \ldots, y_{12a_{12}}, \ldots, y_{pq1}, y_{pq2}, \ldots, y_{pqa_{pq}})$$

$$= (Z_1, Z_2, \ldots, Z_n)$$

i.e., Z_s denotes the s-th sample unit, $s = 1(1)n$.

Define π_s = inclusion probability of s-th sample unit in the sample.

Then $\pi_s = P_h$ if Z_s sample unit comes from the cell numbered h.

$\pi_{ss'}$ = joint inclusion prob. of s-th sample unit and s'-th sample unit in the sample.

Then $\pi_{ss'} = P_{hh'}$ if Z_s comes from the cell numbered h and $Z_{s'}$ comes from the cell numbered h'.

Consider a vector of population units -

$$\underset{\sim}{Z}^* = (Y_{111}, Y_{112}, \ldots, Y_{11b_{11}}, Y_{121}, Y_{122}, \ldots, Y_{12b_{12}}, \ldots, Y_{pq1}, Y_{pq2}, \ldots, Y_{pqb_{pq}})$$

$$= Z_1^*, Z_2^*, \ldots\ldots, Z_N^*$$

i.e., Z_s^* den^ces the s-th population unit, $s = 1, 2, \ldots, N$.

Define

π_s^* = prob. that s-th population unit will be included in the sample.

$= P_h$ if Z_s^* population unit belongs to the cell numbered h.

$\pi_{ss'}^*$ = prob. that s-th population unit and s'-th population unit will be included in the sample.

Then $\pi_{ss'}^* = P_{hh'}$ if Z_s^* belongs to the cell numbered h and $Z_{s'}^*$ belongs to the cell numbered h'.

Horvitz-Thompson estimate of population total is given by

$$\hat{T}_{HT} = \sum_{i=1}^{n} Z_i/\pi_i$$

Horvitz-Thompson estimate of population mean is given by

$$\hat{\bar{Y}}_{HT} = \hat{T}_{HT}/N \ .$$

Variance of $\hat{T}_{HT}$ is given by

$$V(\ \hat{T}_{HT}\ ) = \sum_{i<j=1}^{N}\sum \ (\pi_i^* \ \pi_j^* - \pi_{ij}^*)(\frac{Z_i^*}{\pi_i^*} - \frac{Z_j^*}{\pi_j^*})^2$$

Variance of the $\hat{\bar{Y}}_{HT}$ is given by

$$V(\ \hat{\bar{Y}}_{HT}\ ) = V(\ \hat{T}_{HT}\ )/N^2 \ .$$

An unbiased estimatt of $V(\ \hat{T}_{HT}\ )$ is given by

$$\hat{V}\ (\ \hat{T}_{HT}\ ) = \sum_{i<j=1}^{n}\sum \ (\frac{\pi_i\pi_j - \pi_{ij}}{\pi_{ij}})(\frac{Z_i}{\pi_i} - \frac{Z_j}{\pi_j})^2$$

An unbiased estimate of $V(\ \hat{\bar{Y}}_{HT}\ )$ is given by

$$\hat{V}\ (\ \hat{\bar{Y}}_{HT}\ ) = \hat{V}\ (\ \hat{T}_{HT}\ )/N^2 \ .$$

7. Properties of π_i^* 's and π_{ij}^* 's :

π_i^* = prob. that i-th population unit will be included in the sample.

π_{ij}^* = prob. that i-th population unit and j-th population unit will be included in the sample.

The inclusion probabilities π_i^* 's and π_{ij}^* 's follow the following relations :-

(i) $\sum_{i=1}^{N} \pi_i^* = n$, (ii) $\sum_{j(\neq i)=1}^{N} \pi_{ij}^* = (n-1)\ \pi_i^*$,

(iii) $\sum\sum_{i\neq j=1}^{N} \pi_{ij}^* = n\ (n-1)$

where n is the sample size.

In my program, the condition (i) has been checked by the procedure 'check-on-piei' and the conditions (ii) and (iii) have been verified by the procedure 'check-on-pieij'.

8. Comments on $\hat{V}(\hat{T}_{HT})$:

We know that the variance of any estimator is positive quantity, so it is desirable that an estimate of the variance should be positive. But depending upon the inclusion probabilities π_i's and π_{ij}'s the Horvitz-Thompson estimate for the variance may be negative quantity. If π_i's and π_{ij}'s are such that $\pi_i \pi_j > \pi_{ij}$, $\forall$ $i, j \in \{1, 2, \ldots, n\}$, then the Horvitz-Thompson estimate of the variance will be positive. But if the condition $\pi_i \pi_j > \pi_{ij}$ does not hold for some (i, j) combinations, then there is no guarantee that the estimate of the variance will be positive. There are n_{c_2} number of (i, j) combinations. We can expect that the probability of being positive estimated variance will be high if there are less number of (i, j) combinations for which the condition $\pi_i \pi_j > \pi_{ij}$ does not hold.

9. Method of generation of random numbers :

Z_i = i-th generated random number.

$Z_i = a \, Z_{i-1} \pmod{m}$

where a and m are constants and Z_o is an initial number (specified).

In my program $a = 16807$

$m = 32767 = 2^{15} - 1$

To get a random number (x) from 1 to r, call the procedure 'random(x, r)'.

10. Computer outputs :

Computer outputs for different sets of data are attached after the last page of this report.

11. References :

1. Sampling Theory and Methods - M.N. Murthy

2. Sampling Techniques - W.G. Cochran