An Algorithm for Small Area Estimation under Not Missing At Random Non-response October 2022

Michael Sverchkov
Bureau of Labor Statistics, Washington DC, USA

Abstract

Sverchkov and Pfeffermann (2018, 2019), consider small area estimation (SAE) under informative probability sampling of areas and within the sampled areas, and not missing at random (NMAR) nonresponse.

To account for the nonresponse, the authors assume a given response model and estimate the corresponding response probabilities by application of the Missing Information Principle, which consists of defining the likelihood as if there was complete response and then integrating out the unobserved outcomes from the likelihood, employing the relationship between the sample and sample-complement distributions. Sverchkov and Pfeffermann (2018) illustrate the use of this approach by a simulation study and by application to a real discreate data set. In this paper we suggest an algorithm of estimating the response probabilities for the case of continuous data.

Key words: population distribution, respondents model, sample distribution.

1. Notation and Models

Let $\{y_{ij}, \mathbf{x}_{ij}; i = 1,...,M,\ j = 1,...,N_i\}$ represent the data in a finite population of N units belonging to M areas with N_i units in area i, $\sum_{i=1}^{M} N_i = N$, where y_{ij} is the value of the study variable for unit j in area i and $\mathbf{x}'_{ij} = (x_{ij,1},...,x_{ij,K})$ is a vector of corresponding K covariates. We assume that the covariates are known for every unit in the population. Suppose that the outcome values follow the generic two-level population model:

$$\begin{aligned} &y_{ij} \mid \mathbf{x}_{ij}, u_i^U \sim f(y_{ij} \mid \mathbf{x}_{ij}, u_i^U),\ i = 1,...,M,\ j = 1,...,N_i \\ &u_i^U \sim f(u_i^U);\ E(u_i^U) = 0,\ V(u_i^U) = \sigma^2_{u^U}, \end{aligned} \tag{1}$$

where u_i^U is the i[th] area level random effect under this model. The target is to estimate the area means $\bar{Y}_i = N_i^{-1} \sum_{j=1}^{N_i} y_{ij}, i = 1,...,M$, based on a sample obtained by the following two-stage sampling scheme: select a sample s of m out of the M population areas with inclusion probabilities $\pi_i = \Pr(i \in s)$; select a sample s_i of $n_i > 0$ units from selected area i with probabilities $\pi_{j|i} = \Pr(j \in s_i \mid i \in s)$. Denote by I_i, I_{ij} the sample indicators; $I_i = 1$ if area i is selected in the first stage and 0 otherwise, $I_{ij} = 1$ if unit j of selected area i is sampled in the second stage and $I_{ij} = 0$ otherwise. Let $w_i = 1/\pi_i$, $w_{j|i} = 1/\pi_{j|i}$ denote the first- and second-stage base sampling weights.

In practice, not every unit in the sample responds. Define the response indicator; $R_{ij} = 1$ if unit $j \in s_i$ responds and $R_{ij} = 0$ otherwise. The sample of respondents is thus $R = \{(i, j) : I_i = 1, I_{ij} = 1, R_{ij} = 1\}$ and the sample of nonrespondents among the sampled units is $R^c = \{(i,k) : I_i = 1, I_{ik} = 1, R_{ik} = 0\}$. We assume $\sum_{j=1}^{n_i} R_{ij} > 0$ for all the sampled areas. The sample of respondents can thus be viewed as the result of a two-stage sampling process where in the first stage the sample is selected from the population with known inclusion probabilities, and in the second stage the sample is "self-selected" with unknown response probabilities (Särndal and Swensson, 1987).

Define, $u_i = u_i^U - E(u_i^U \mid i \in s)$. Then, under the population model (1), the observed data follow the two-level 'respondents' model:

$$f_R(y_{ij} \mid \mathbf{x}_{ij}, u_i) = f(y_{ij} \mid \mathbf{x}_{ij}, u_i, (i, j) \in R);\ u_i \sim f(u_i \mid i \in s), E(u_i \mid i \in s) = 0 . \quad (2)$$

The model in (2) is again general and all that we state at this stage is that under informative sampling and/or NMAR nonresponse, the population and the respondents models differ, $f_R(y_{ij} \mid \mathbf{x}_{ij}, u_i) \neq f(y_{ij} \mid \mathbf{x}_{ij}, u_i^U)$.

Remark 1. The respondents' model refers to the observed data and hence can be estimated and tested by standard small area estimation (SAE) methods. See Pfeffermann (2013) and Rao and Molina (2015) for estimation and testing procedures in SAE, with references.

Let $p_r(y_{ij}, \mathbf{x}_{ij}) = \Pr[R_{ij} = 1 \mid y_{ij}, \mathbf{x}_{ij}, i \in s, j \in s_i]$. If the probabilities $p_r(y_{ij}, \mathbf{x}_{ij})$ were known, the sample of respondents could be considered as a two-stage sample from the finite population with known sampling probabilities π_i and $\tilde{\pi}_{j|i} = \pi_{j|i} p_r(y_{ij}, \mathbf{x}_{ij})$. Then the area means $\bar{Y}_i = N_i^{-1} \sum_{j=1}^{N_i} y_{ij}, i = 1, ..., M$ can be estimated as in Pfeffermann and Sverchkov (2007). Also, if known, the response probabilities could be used for imputation of the missing data within the selected areas, by application of the relationship between the sample and sample-complement distributions, (Sverchkov and Pfeffermann, 2004),

$$f(y_{ij} \mid \mathbf{x}_{ij}, u_i, (i, j) \in R^c) = \frac{[p_r^{-1}(y_{ij}, \mathbf{x}_{ij}) - 1] f(y_{ij} \mid \mathbf{x}_{ij}, u_i, (i, j) \in R)}{E\{[p_r^{-1}(y_{ij}, \mathbf{x}_{ij}) - 1] \mid \mathbf{x}_{ij}, u_i, (i, j) \in R\}} . \quad (3)$$

See Pfeffermann and Sverchkov (2019) for details.

2. Estimation of Response Probabilities

In what follows we assume a parametric model for the response probabilities, which depends on an unknown vector parameter $\boldsymbol{\gamma}$; $p_r(y_{ij}, \mathbf{x}_{ij}) = p_r(y_{ij}, \mathbf{x}_{ij}; \boldsymbol{\gamma}) = \Pr[R_{ij} = 1 \mid y_{ij}, \mathbf{x}_{ij}, i \in s, j \in s_i; \boldsymbol{\gamma}]$. We assume that $p_r(y_{ij}, x_{ij}; \boldsymbol{\gamma})$ is differentiable with respect to $\boldsymbol{\gamma}$ and satisfies the same general regularity conditions as in Sverchkov and Pfeffermann (2018).

Under these assumptions, if the missing outcome values were actually observed, $\boldsymbol{\gamma}$ could be estimated by solving the likelihood equations:

$$\sum_{(i,j)\in R}\frac{\partial \log p_r(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}+\sum_{(i,\mathrm{k})\in R^c}\frac{\partial \log[1-p_r(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma})]}{\partial \boldsymbol{\gamma}}=0\,. \tag{4}$$

In practice, the missing data are unobserved and hence the likelihood equations (4) are not operational. However, one may apply in this case the missing information principle:

Missing Information Principle (Cepillini et al. 1955, Orchard and Woodbury, 1972)**:** since no observations are available for $(i,k)\in R^c$, solve instead,

$$E_U\left\{\left[\sum_{(i,j)\in R}\frac{\partial \log p_r(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}+\sum_{(i,\mathrm{k})\in R^c}\frac{\partial \log[1-p_r(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma})]}{\partial \boldsymbol{\gamma}}\right]\middle| O\right\}$$

$$\overset{by(3)}{=}\sum_{(i,j)\in R}\frac{\partial \log p_r(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}$$

$$+\sum_{(i,k)\in R^c}E_s\left(\frac{E_{re}\left\{[p_r^{-1}(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma})-1]\dfrac{\partial \log[1-p_r(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma})]}{\partial \boldsymbol{\gamma}}\middle|\,\mathbf{x}_{ik},u_i,(i,k)\in R\right\}}{E_{re}\{[p_r^{-1}(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma})-1]\,|\,\mathbf{x}_{ik},u_i,(i,k)\in R\}}\middle| O\right)=0\,, \tag{5}$$

where $O=\{y_{ij},\pi_{j|i},\pi_i,n_i,(i,j)\in R;\ \mathbf{x}_{ht},h=1,...,M,\ \mathrm{t}=1,...,N_i\}$ represents all the observed data, see Sverchkov (2008) and Sverchkov and Pfeffermann (2018) for derivation of (5). In (5) $E_U,\mathrm{E}_s,\mathrm{E}_{re},E_{nre}$ define respectively expectations with respect to the population distribution, the sample distribution, the respondents' distribution and the non-respondents' distribution. Notice that the internal expectations in the last expression are with respect to the model holding for the observed data for the respondents.

In practice, the vector parameter $\boldsymbol{\gamma}$ is estimated by replacing u_i by $\hat{u}_i$ obtained by estimating of the observed model (2) and dropping the external expectation. We propose to solve the resulting equations by minimizing the log-likelihood leading to them, i.e., minimizing,

$$\sum_{(i,j)\in R}\log p_r(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})$$

$$+\sum_{(i,k)\in R^c}E_s\left(\frac{E_{re}\{[p_r^{-1}(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma}^*)-1]\log[1-p_r(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma})]\,|\,\mathbf{x}_{ik},u_i,(i,k)\in R\}}{E_{re}\{[p_r^{-1}(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma}^*)-1]\,|\,\mathbf{x}_{ik},u_i,(i,k)\in R\}}\middle| O\right). \tag{6}$$

We distinguish between $\boldsymbol{\gamma}^*$ and $\boldsymbol{\gamma}$ because by (5), the derivatives should only be taken with respect to $\boldsymbol{\gamma}$. The minimization is thus carried out iteratively by minimizing the (q+1) iteration of the function,

$$\sum_{(i,j)\in R}\log p_r(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma}^{(q+1)})$$

$$+\sum_{(i,k)\in R^c}E_s\left(\frac{E_{re}\{[p_r^{-1}(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma}^{(q)})-1]\log[1-p_r(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma}^{(q+1)})]\,|\,\mathbf{x}_{ik},u_i,(i,k)\in R\}}{E_{re}\{[p_r^{-1}(y_{ik},\mathbf{x}_{ik};\boldsymbol{\gamma}^{(q)})-1]\,|\,\mathbf{x}_{ik},u_i,(i,k)\in R\}}\middle| O\right) \tag{7}$$

with respect to $\boldsymbol{\gamma}^{(q+1)}$.

Example (Sverchkov and Pfeffermann 2018)**:** *Mixed logistic model for the outcome variable*.

Suppose that the model fitted to the observed data of the respondents is the mixed generalized logistic model,

$$p_y(x_{ij},u_i)=\Pr(y_{ij}=1\,|\,x_{ij},u_i,(i,j)\in R;\boldsymbol{\beta})=\frac{\exp(\beta_0+\beta_1x_{ij}+u_i)}{1+\exp(\beta_0+\beta_1x_{ij}+u_i)},\ u_i\overset{\text{i.i.d.}}{\sim}N(0,\sigma_u^2). \qquad (8)$$

Consider a generic response model, $p_r(y_{ij},x_{ij};\boldsymbol{\gamma})=\Pr[R_{ij}=1\,|\,y_{ij},x_{ij},i\in s,\,j\in s_i;\boldsymbol{\gamma}]$.

The components of (8) can be written in this case as,

$$E_{re}\left\{[p_r^{-1}(y_{ij},x_{ij};\boldsymbol{\gamma})-1]\frac{\partial\log[1-p_r(y_{ij},x_{ij};\boldsymbol{\gamma})]}{\partial\boldsymbol{\gamma}}\,\middle|\,x_{ij},u_i,(i,j)\in R\right\}=$$

$$p_y(x_{ij},u_i)[p_r^{-1}(1,x_{ij};\boldsymbol{\gamma})-1]\frac{\partial\log[1-p_r(1,x_{ij};\boldsymbol{\gamma})]}{\partial\gamma}+$$

$$[1-p_y(x_{ij},u_i)][p_r^{-1}(0,x_{ij};\boldsymbol{\gamma})-1]\frac{\partial\log[1-p_r(0,x_{ij};\boldsymbol{\gamma})]}{\partial\boldsymbol{\gamma}}; \qquad (9)$$

$$E_{re}\{[p_r^{-1}(y_{ij},x_{ij};\boldsymbol{\gamma})-1]\,|\,x_{ij},u_i,(i,j)\in R\}=p_y(x_{ij},u_i)[p_r^{-1}(1,x_{ij};\boldsymbol{\gamma})-1]+$$

$$[1-p_y(x_{ij},u_i)][p_r^{-1}(0,x_{ij};\boldsymbol{\gamma})-1]. \qquad (10)$$

The random effects u_i and the logistic probabilities $p_y(x_{ij},\mathrm{u}_i)$ can be estimated by use of the SAS procedure PROC NLMIX.

3. An Algorithm for SAE Under NMAR, Continuous Data

In the example above, the outcomes follow a discrete distribution. In this section, we consider continues outcomes. The proposed algorithm consists of three parts:

Part 1: Fit (estimate) the model (2). The output of Part 1 (input for Part 2) contains the model (2) parameter estimates and estimates of random effect, $\hat{u}_i$, estimates of $p_y^{(l)}(\mathbf{x}_{ij},\hat{u}_i)=P_R(a_l\le y_{ij}<a_{l+1}\,|\,\mathbf{x}_{ij},\hat{u}_i,(i,j)\in R)$, $l=0,...,L+2$, $a_0=-\infty$, $a_{L+2}=\infty$, $a_l=\min(y_{ij})+(l-1)\dfrac{\max(y_{ij})-\min(y_{ij})}{L}$, (where max and min is over all observed y_{ij}), $l=1,...,L+1$, for each $(i,j)\in R$ (see the following SAS example for details).

Part 2: Approximate the expectations in (7) similarly to (9) and (10),

$$E_{re}\left\{[p_r^{-1}(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})-1]\frac{\partial\log[1-p_r(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})]}{\partial\boldsymbol{\gamma}}\,\middle|\,\mathbf{x}_{ij},u_i,(i,j)\in R\right\}\cong$$

$$\sum_{l=1}^{L+1}p_y^{(l)}(\mathbf{x}_{ij},\hat{u}_i)[p_r^{-1}(a_l,\mathbf{x}_{ij};\boldsymbol{\gamma})-1]\frac{\partial\log[1-p_r(a_l,\mathbf{x}_{ij};\boldsymbol{\gamma})]}{\partial\boldsymbol{\gamma}}, \qquad (11)$$

$$E_{re}\left\{[p_r^{-1}(y_{ij},\mathbf{x}_{ij};\boldsymbol{\gamma})-1]\,\middle|\,\mathbf{x}_{ij},u_i,(i,j)\in R\right\}\cong\sum_{l=1}^{L+1}p_y^{(l)}(\mathbf{x}_{ij},\hat{u}_i)[p_r^{-1}(a_l,\mathbf{x}_{ij};\boldsymbol{\gamma})-1], \qquad (12)$$

substitute (11) and (12) into (7) and estimate $\boldsymbol{\gamma}$ by iteratively minimizing (7).

Part 3: Finally, $\bar{Y}_i = N_i^{-1}\sum_{j=1}^{N_i} y_{ij}, i = 1,...,M$ can be estimated based on (3) as,

$$\hat{\bar{Y}}_i = N_i^{-1}[\sum_{j:(i,j)\in R} y_{ij} + \sum_{j:(i,j)\in R^c} \frac{\sum_{l=1}^{L+1} a_l p_y^{(l)}(\mathbf{x}_{ij}, \hat{u}_i)[p_r^{-1}(a_l, \mathbf{x}_{ij}; \hat{\boldsymbol{\gamma}}) - 1]}{\sum_{l=1}^{L+1} p_y^{(l)}(\mathbf{x}_{ij}, \hat{u}_i)[p_r^{-1}(a_l, \mathbf{x}_{ij}; \hat{\boldsymbol{\gamma}}) - 1]}]. \quad (13)$$

Note. The choice of L depends on the distribution in (2). For example, from simulation studies (not reported in this paper), I found that if the working model in (2) is Normal, then L=100 gives very good approximation for the expectations in (2). On the other hand, if the working model in (2) follows the Gamma distribution, then it is better to use L=1000.

SAS example.

In the following example, 100 populations are generated from the nested error regression model, $y_{ij} \mid x_{ij}, u_i \sim N(x_{ij} + u_i, 1),\ i = 1...100,\ j = 1...20$, $u_i \sim N(0,1)$. From each simulated population a sample of respondents is selected by Poisson sample with response probabilities, $p_r(y_{ij}, x_{ij}) = \frac{\exp(-.5x_{ij} + 2y_{ij})}{1 + \exp(-.5x_{ij} + 2y_{ij})}$.

For each simulation, the area means are estimated by the proposed three parts procedure, where the respondent data is assumed to follow the nested error model, $y_{ij} \mid x_{ij}, u_i \sim N(\theta_0 + \theta_1 x_{ij} + u_i, \theta_2),\ i = 1...100,\ j = 1...20$, $u_i \sim N(0, \sigma_u^2)$, (note that this working model is incorrect under the population and response models above), and working parametric model for the response probabilities, $p_r(y_{ij}, x_{ij}; \boldsymbol{\gamma}) = \frac{\exp(\gamma_0 + \gamma_1 x_{ij} + \gamma_2 y_{ij})}{1 + \exp(\gamma_0 + \gamma_1 x_{ij} + \gamma_2 y_{ij})}$.

In addition, the area means are estimated assuming MAR response. Finally, the empirical Bias, MSE's and relative Bias of the suggested and MAR area estimates over the 100 simulations are calculated.

```
libname lib 'c:\jsm2022';
options ls=100 nocenter nonotes;

%let gamma0=0; %let gamma1= -0.5; %let gamma2=2;
                                   /* - parameters of the response function (logistics) */
%let sigma2_u=1; /* - variance of the random effect */
%let theta0=0; %let theta1=1; %let theta2=1;
                                /*- parameters of the population model of the outcome*/
%let Mcap=100; /* - number of areas */
%let Ncap=20;  /*- number of units within the area */
%let seed=-1;
%let cp=0.01; %let co=0.01; /*- convergence parameters for Proc NLIN */
%let num_darboux=100;   /*- number of intervals used for approximation expectations in (7),
                             (11) and (12)  by respective sums */
%let num_nlin_iter=30;  /*- number of iterations in estimation of gamma parameters (7) */
```

```
%let numit=100; /*- number of simulation runs */

%macro a;
%do it=1 %to &numit;

/* Part 0: simulation of population and calculation of true area means, simulation of response*/

data pop0; do i=1 to &Mcap;it=&it; ui=sqrt(&sigma2_u)*max(-2.5,min(2.5,rannor(-1))); output;
end; run;
data pop; set pop0; do j=1 to &Ncap;
x=2*ranuni(&seed);
y=x*&theta1+ui+&theta0+&theta2*rannor(&seed); output; end; run;
proc means data=pop noprint; var y; by it i;
output out=true(keep= it i meany) mean=meany; run; /*- calculate true area means for each area,
                used for final accuracy statistics, empirical bias and empirical MSE calculations  */

/*End simulation of population*/

/*Simulation of response */

data pop; set pop; indr=0; if
ranuni(&seed)<exp(&gamma0+&gamma1*x+&gamma2*y)/(1+exp(&gamma0+&gamma1*x+&
gamma2*y)) then indr=1; run;    /*- here response model is logit, with intercept, x and y */

/*End simulation of response */
data resp; set pop; if indr=0 then delete; run;
data samp_resp; set pop; if indr=1 then delete; run;

/* End of Part 0 */

/* Part 1: estimation of observed model (2) */

/* Here the observed working model is two level model with random effect distributed normally
and outcomes given random effect having normal distribution. */

proc nlmixed data=resp;
parms beta0=0 beta1=0 beta2=1 s2u=1; eta=u;
ll=log((1/sqrt(2*3.14159265358979*beta2**2))*exp(-(y-beta0-x*beta1-u)**2/(2*beta2**2)));
     model y ~ general(ll);
     random u ~ normal(0,s2u) subject=i;
     predict eta out=resp;
ods output ParameterEstimates=parms_resp; run;

data t0; set parms_resp(obs=1); beta0=estimate; it=&it; keep it beta0; run;
data t1; set parms_resp(firstobs=2 obs=2); beta1=estimate; it=&it; keep it beta1; run;
data t2; set parms_resp(firstobs=3 obs=3); beta2=estimate; it=&it; keep it beta2; run;
data t3; set parms_resp(firstobs=4 obs=4); s2u=estimate; it=&it; keep it s2u; run;
data samp_resp; merge samp_resp t0 - t3; by it; run;
data resp; merge resp t0 - t3; by it; run;

/* Following 4 lines read random effect estimates and add them to data resp and data samp_esp */
```

```
data hatu; set resp; hat_u=pred; run;
proc means data=hatu noprint; by it i; var hat_u; output out=hat_u(keep=it i hat_u) mean=hat_u;
run;
data samp_resp; merge samp_resp hat_u; by it i; Pred=888; run; data samp_resp; set samp_resp;
if y=. then delete; run;
data resp; merge resp hat_u; by it i; keep it i j y x indr beta0 - beta2 s2u hat_u pred ui; run;
proc means data=resp noprint; var y; by it; output out=t(keep=it yd0 ydend) min=yd0
max=ydend; run;   /*- max and min of y used for expectations approximation in (11) – (12) */

proc append base=resp data=samp_resp force; run;
data resp; merge resp t; by it; run;

/*  End estimation of observed model */

/* Start calculating interval probabilities needed for approximation the expectations in (11)-(12)
*/

data resp; set resp;
array yd yd1-yd&num_darboux.;
array pyd pyd1-pyd&num_darboux.;
do d=1 to &num_darboux;
yd{d}=yd0+d*(ydend-yd0)/&num_darboux;   /*- define interval points for (11)-(12)
                                            approximation */
if d=1 then pyd{d}=probnorm((yd0-beta0-x*beta1-hat_u)/beta2);
if d>1 then pyd{d}=probnorm((yd{d}-beta0-x*beta1-hat_u)/beta2)-probnorm((yd{d-1}-beta0-
x*beta1-hat_u)/beta2);
end;
run;

/* End interval probabilities needed for approximation the integrals (expectations) in (11)-(12) */

/* End of Part 1*/

/* Part 2: Estimating parameters of the response model*/

/* Starting values (assuming ignorable model)*/;
proc logistic data=resp outest=parms noprint; model indr(event='1')=x;
data igntheta; set parms; ignb0=intercept; ignb1=x; it=&it;
 gs0=ignb0; gs1=ignb1; gs2=1.1;
call symput("ignb0",ignb0); call symput("ignb1",ignb1);
call symput("gs0",gs0); call symput("gs1",gs1); call symput("gs2",gs2);
keep it ignb0 ignb1; run;
/* End Starting values*/

/* Start of estimating response model parameters: Gamma estimates */

data resp; set resp; null=0; run;
data resptemp; set resp; if indr=0 then y=1/99; run;
/* y=1/99 added for checking, outcomes for non-respondents, indr=0, do not influence the
program if it is written correctly*/
```

```
%do k=1 %to &num_nlin_iter;  /* &num_nlin_iter depends on the data, in real data case one
need to run the program for big &num_nlin_iter and look when the gamma estimates stop to
change or use other criteria */
proc nlin data=resptemp outest=gamma_est convergeparm=&cp convergeobj=&co noprint
method=gauss save;
parms g0=&gs0, g1=&gs1 g2=-5 to 5 by 0.1;
bounds  -15<g0<15, -100<g1<100, -15<g2<15;
array yd yd1-yd&num_darboux.;
array pyd pyd1-pyd&num_darboux.;
nominator=0; denominator=0;
do d=1 to &num_darboux;
nominator=nominator+((1+exp(&gs0+&gs1*x+&gs2*yd{d}))/exp(&gs0+&gs1*x+&gs2*yd{d})
-1)*log(1/(1+exp(g0+g1*x+g2*yd{d})))*pyd{d};
denominator=denominator+((1+exp(&gs0+&gs1*x+&gs2*yd{d}))/exp(&gs0+&gs1*x+&gs2*yd
{d})-1)*pyd{d};
end;
Eclog=nominator/denominator;
f=(indr*(g0+g1*x+g2*y-log(1+exp(g0+g1*x+g2*y)))+(1-indr)*Eclog); /*- f estimates (7) */
model null=(-f)**0.5;
run;

/*Following 5 lines read the intermediate estimate and redefine the starting values for the next
cycle of estimation */

data gamma; set gamma_est; if _TYPE_ ne 'FINAL' then delete; it=&it;  keep it g0 g1 g2; run;
data gamma; set gamma; gs0=g0; gs1=g1; gs2=g2;
call symput("gs0",gs0); call symput("gs1",gs1); call symput("gs2",gs2); run;
data resptemp; merge resptemp gamma; by it; run;
data resptemp; set resptemp; drop g0 g1 g2; run;

%if &it<8 %then %do; proc print data=gamma; run; %end;

%end;

%if &it=1 %then %do; data lib.gamma; set gamma(obs=0); run; %end;
proc append base=lib.gamma data=gamma force; run;  /*- save final gamma estimates into
                                                          data set*/

/*   End of Part 2*/

/* Part 3: Final SAE estimates  */
data resptemp; set resptemp;
array yd yd1-yd&num_darboux.;
array pyd pyd1-pyd&num_darboux.;
l_nominator=0; nominator=0; denominator=0; Eresp=0;
do d=1 to &num_darboux;
l_nominator=l_nominator+((1+exp(&gs0+&gs1*x+&gs2*yd{d}))/exp(&gs0+&gs1*x+&gs2*yd
{d})-1)*log(1/(1+exp(&gs0+&gs1*x+&gs2*yd{d})))*pyd{d};
nominator=nominator+((1+exp(&gs0+&gs1*x+&gs2*yd{d}))/exp(&gs0+&gs1*x+&gs2*yd{d})
-1)*yd{d}*pyd{d};
```

```
denominator=denominator+((1+exp(&gs0+&gs1*x+&gs2*yd{d}))/exp(&gs0+&gs1*x+&gs2*yd
{d})-1)*pyd{d};
Eresp=Eresp+yd{d}*pyd{d}; /*** E_re(y|x,u)  ****/
end;
Enonrep=nominator/denominator; /*** E_nre(y|x,u)  ****/
ign=beta0+beta1*x+hat_u; ign=Eresp; /*- predictor from proc nlmixed (ignoring response) */
ymy=indr*y+(1-indr)*nominator/denominator;  /*- Suggested estimator (13) */
yign=indr*y+(1-indr)*ign; /*- estimator assuming MAR, ignoring informativity of response
                              (biased in this case) */
log_l=(indr*(&gs0+&gs1*x+&gs2*y-log(1+exp(&gs0+&gs1*x+&gs2*y)))+(1-
indr)*l_nominator/denominator); /*- Log Likelihood, can be used for response model selection */
run;
proc sort data=resptemp; by it i; run;
proc means data=resptemp noprint; var ymy yign log_l; by it i; output out=sae_est(keep=it i
s_sae_my s_sae_ign log_l) sum=s_sae_my s_sae_ign  log_l; run;
data sae_est; set sae_est; sae_my=s_sae_my/&Ncap; sae_ign=s_sae_ign/&Ncap; run;

/* End Part 3*/

/*  Bias and MSE empirical statistics */
data sae_est; merge sae_est true; by it i; run;
data sae_est; set sae_est; bias_my=meany-sae_my; mse_my=bias_my**2; bias_ign=meany-
sae_ign; mse_ign=bias_ign**2; run;
%if &it=1 %then %do; data lib.sae_est; set sae_est(obs=0); run; %end;
proc append base=lib.sae_est data=sae_est force; run;
%end; %mend; %a;



/*Results over simulation studies */

proc means data=lib.gamma; var g0 - g2; run;
proc sort data=lib.sae_est; by i; run;
proc means data=lib.sae_est noprint; var bias_my bias_ign mse_my mse_ign; by i; output out=t
(keep=i bias_my bias_ign mse_my mse_ign) mean=bias_my bias_ign mse_my mse_ign; run;

data t; set t; rel_bias_my=bias_my/sqrt(mse_my); rel_bias_ign=bias_ign/sqrt(mse_ign); run;
proc means data=t; run;
```

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
g0	100	0.2881809	0.6151212	-0.2970055	2.9734545
g1	100	-0.6179596	0.2473706	-1.4768553	-0.1356483
g2	100	2.2611048	0.6993000	1.5879451	5.4610828

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
i	100	50.5000000	29.0114920	1.0000000	100.0000000
bias_my	100	0.0149357	0.0191400	-0.0357148	0.0513812
bias_ign	100	-0.4711682	0.0452364	-0.5975926	-0.3825212
mse_my	100	0.0410990	0.0116800	0.0221409	0.0777839
mse_ign	100	0.4002557	0.0810996	0.2722546	0.6182223
rel_bias_my	100	0.0828893	0.1004781	-0.1451939	0.3026021
rel_bias_ign	100	-0.7493356	0.0278722	-0.8029488	-0.6682278

Acknowledgment: I thank Danny Pfeffermann for many valuable comments and suggestions.

The opinions expressed in this paper are of the author and do not necessarily represent the policies of the U.S. Bureau of Labor Statistics

References

Cepillini, R., Siniscialco, M., and Smith, C.A.B. (1955). The estimation of gene frequencies in a random mating population. *Annals of Human Genetics*, **20**, 97-115.

Orchard, T., and Woodbury, M.A. (1972). A missing information principle: theory and application. *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 697-715.

Pfeffermann, D. (2013). New Important Developments in Small Area Estimation. *Statistical Science*, **28**, 40-68.

Pfeffermann, D., and Sverchkov, M. (2007). Small-Area Estimation under Informative Probability Sampling of Areas and Within Selected Areas. *Journal of the American Statistical Association*, **102**, 1427-1439.

Pfeffermann, D., and Sverchkov, M. (2009). Inference under Informative Sampling. In: Handbook of Statistics 29B; *Sample Surveys: Inference and Analysis*. Eds. D. Pfeffermann and C.R. Rao. North Holland. pp. 455-487.

Pfeffermann, D. and Sverchkov, M. (2019). Multivariate small area estimation under nonignorable nonresponse, *Statistical Theory and Related Fields*, **3**, pp. 213-223

Rao, J.N.K., and Molina, I. (2015), *Small Area Estimation*, 2nd Edition, Wiley.

Särndal, C.E., and Swensson B. (1987). A general view of estimation for two phases of selection with applications to two-phase sampling and nonresponse. *International Statistical Review*, **55**, 279-294.

Sverchkov, M. (2008). A new approach to estimation of response probabilities when missing data are not missing at random. *Joint Statistical Meetings, Proceedings of the Section on Survey Research Methods*, 867-874.

Sverchkov, M., and Pfeffermann, D. (2004). Prediction of Finite Population Totals Based on the Sample Distribution. *Survey Methodology*, **30**, 79-92.

Sverchkov, M. and Pfeffermann, D. (2018). Small area estimation under informative sampling and not missing at random non-response. *Journal of Royal Statistical Society, ser. A*, 181, *Part* 4, *pp.* 981–1008.