United States
Department of
Agriculture

National
Agricultural
Statistics
Service

Washington,
DC.
20250

NASS Staff Report
Number SSB-88-10

December 1988

Allocations Requiring 100% Sampling In Some Strata

James Weldon Mergerson

Allocations Requiring 100% Sampling in Some Strata. James Weldon Mergerson, Research and Applications Division, National Agricultural Statistics Service, U.S. Department of Agriculture, Washington, D.C. 20250, December 1988. Staff Report Number SSB-88-10.

ABSTRACT

A general procedure for computing sample allocations for a stratified design, with simple random sampling within strata, based on a fixed desired level of precision is presented. The procedure can be applied when 100% sampling is required in some strata and costs among strata are unequal. A computer program, which implements the procedure, is also presented.

KEYWORDS: Stratified sampling, Optimal allocation, Fixed precision.

ACKNOWLEDGMENTS

Pat Thomas and *Beth Edwards* provided technical assistance which facilitated typesetting this report.

Table of Contents

Area Frame Section **Fairfax Circle Building** **Fairfax, Virginia 22030**

Allocations Requiring 100% Sampling In Some Strata

James Weldon Mergerson

1. INTRODUCTION

A univariate optimum allocation formula for a stratified sampling design with simple random sampling within strata may produce $n_h > N_h$ in some strata. Cochran (1977) gives a procedure for determining the optimum allocation for this situation when equal costs among strata are assumed and a fixed desired precision level is specified. When costs among strata differ this procedure is not applicable. An explicit procedure for this case does not seem to be available in sampling textbooks or statistical journals.

This paper, intended for NASS (National Agricultural Statistics Service) Mathematical Statisticians and limited distribution to the academic community, presents a general procedure and computer program for computing univariate optimal allocations. The procedure is applicable for determining optimum allocations for a stratified sampling design, with simple random sampling within strata, for a fixed desired level of precision when costs among strata are unequal. An example is presented to illustrate this situation. Allocations requiring 100% sampling in some strata can occur when the overall sampling fraction is substantial, when costs differ greatly among strata, when the population size of a stratum is small and when a stratum contains a very large percent of an item to be estimated. The problem has occurred in NASS. Some area frame strata are sampled 100%. Battaglia (1988) computed list frame allocations which required 100% sampling in some strata in order to conform with agency target coefficient of variation (CV) standards.

2. GENERAL PROCEDURE

When computing the allocation for a stratified sampling design with simple random sampling within strata, one or more n_h may initially be greater than the corresponding N_h. When this situation occurs the following procedure may be applied:

1. If $n_h > N_h$ set $n_h' = N_h$

2. Compute

$$n' = \frac{\left(\sum_{h \in (I-Q)} N_h s_h (c_h)^{1/2}\right)\left(\sum_{h \in (I-Q)} N_h s_h / (c_h)^{1/2}\right)}{N^2 D^2 + \sum_{h \in (I-Q)} N_h s_h^2}$$

where I is the index set {1, 2, $\cdots$,L} and Q is a set containing all h such that $n_h > N_h$.

3. Compute

$$n_h' = \frac{(n' N_h s_h)/(c_h)^{1/2}}{\sum_{h \in (I-Q)} N_h s_h / (c_h)^{1/2}}, \text{ for all } h \in (I-Q).$$

The optimum allocation is then $n_1', n_2', n_3', \cdots, n_L'$.

3. EXAMPLE

Given a desired coefficient of variation (CV) equal to 0.02, consider determining the optimum allocation using the following data:

Table 1. Allocation Example Data

h	N_h	c_h	s_h	$\bar{y}_h$
1	100	8	3.33	10.0
2	100	4	3.54	20.0
3	100	1	11.73	20.0

The formula for computing the total sample size is as follows:

$$n = [(\sum_{h=1}^{L} N_h s_h (c_h)^{1/2})(\sum_{h=1}^{L} N_h s_h/(c_h)^{1/2})]/[N^2 D^2 + \sum_{h=1}^{L} N_h s_h^2]. \qquad (2.1)$$

where

L - number of strata

N_h - population size of stratum h

s_h - standard deviation of data in stratum h

c_h - average data collection cost in stratum h

N - $\sum_{h=1}^{L} N_h$

D - desired size of $s_{\bar{y}_{st}}$

$\bar{y}_{st}$ - $\sum_{h=1}^{L} N_h \bar{y}_h / N$

$s_{\bar{y}_{st}}$ - standard error of the estimate of the population mean

$\bar{y}_h$ - mean of data in stratum h

For the given desired coefficient of variation, D is computed as follows:

$D = CV * \bar{y}_{st}$.

$\bar{y}_{st}$ = (100*10.0+100*20.0+100*20.0)/(100+100+100).

$\bar{y}_{st}$ = 50/3.

D = 0.02 * 50/3 = 1/3 .

The desired variance of the estimate of the population total (N^2D^2) is 300*300*(1/3)*(1/3) = 10,000. A layout of other computations follows:

Table 2. Some Allocation Related Computations

h	N_h	s_h	c_h	s_h^2	$N_h s_h (c_h)^{1/2}$	$N_h s_h/(c_h)^{1/2}$	$N_h s_h^2$
1	100	3.33	8	11.09	941.9	117.7	1,109
2	100	3.54	4	12.53	708.0	177.0	1,253
3	100	11.73	1	137.59	1,173.0	1,173.0	13,759
Totals					2,822.9	1,467.7	16,121

n = (2,822.9 * 1,467.7)/(10,000 + 16,121)

n = 159

The allocations to each stratum is computed using the formula:

$$n_h = [nN_h s_h/(c_h)^{1/2}]/[\sum_{h=1}^{L} N_h s_h/(c_h)^{1/2}]. \qquad (2.2)$$

n_1 = 159*117.7/1,467.7 = 13

n_2 = 159*177.0/1,467.7 = 19

n_3 = 159*1,173.0/1,467.7 = 127.

Since $n_3 > N_3$, set $n_3' = N_3$. The formula for computing the variance of the estimate of the population mean is as follows:

$$s_{\bar{y}_n}^2 = \frac{1}{N^2}\sum_{h=1}^{L} N_h (N_h - n_h)\frac{s_h^2}{n_h}. \tag{2.3}$$

The contribution to the standard error of the estimate of the population mean from stratum 3 will be zero. Strata 1 and 2 will account for all of the sample variance. The combined sample size for strata 1 and 2 must now be calculated. Since strata 1 and 2 will account for all of the variance, the quantity N^2D^2 in (2.1) remains the same. The formula for the sub-total sample size for strata 1 and 2 is:

$$n' = [(\sum_{h=1}^{2} N_h s_h (c_h)^{½})(\sum_{h=1}^{2} N_h s_h /(c_h)^{½})]/[N^2D^2 + \sum_{h=1}^{2} N_h s_h^2]. \tag{2.4}$$

Applying (2.4) to the example:

$n' = (1649.9 * 294.7) / (10,000 + 2,362)$

$n' = 40$

The allocation to each stratum is computed using the formula:

$$n_h' = [n' N_h s_h /(c_h)^{½}]/[\sum_{h=1}^{2} N_h s_h /(c_h)^{½}] .$$

$n_1' = 40 * 117.7/294.7 = 16$

$n_2' = 40 * 177.0/294.7 = 24$

The revised allocation is $n_1' = 16$, $n_2' = 24$ and $n_3' = 100$.

4. RECOMMENDATIONS

1. Agency statisticians should obtain and use the allocation program listed in Appendix B when performing univariate allocation analysis.

2. The general procedure should be extended to the multivariate case.

3. Multivariate allocation software should be modified to automatically compute the optimal allocation for the 100% sampling case.

5. REFERENCES

Battaglia, Robert (1988), *Methodology To Evaluate Sample Designs Of The Quarterly Agricultural Survey,* NASS Staff Report No. SSB-88-09, U.S. Department of Agriculture, National Agricultural Statistics Service.

Cochran, William (1977), *Sampling Techniques,* New York: John Wiley and Sons.

Hansen, Morris, Hurwitz, William and Madow, William (1953), *Sample Survey Methods and Theory,* New York: John Wiley and Sons.

Jessen, Raymond (1978), *Statistical Survey Techniques,* New York: John Wiley and Sons.

Kish, Leslie (1965), *Survey Sampling,* New York: John Wiley and Sons,

Raj, Des (1968), *Sampling Theory,* New York: McGraw-Hill.

Sukhatme, Pandurang, Sukhatme, Balkrishna, Sukhatme, Shashikala and Asok, C. (1984), *Sampling Theory of Surveys with Applications,* Ames, Iowa: Iowa State University Press.

APPENDIX A : Cochran's Procedure

An algorithmic form of Cochran's procedure for allocations requiring more than 100% sampling (equal costs among strata) is outlined:

i. Compute n

ii. Compute n_h for all $h \in I$

iii. Set $n_h' = n_h$ for all $h \in I$

1. If $n_h' > N_h$ set $n_h' = N_h$

2. Compute

$$n_h = \frac{(n - \sum_{k \in Q} N_k)(N_h s_h)}{(\sum_{h \in (I-Q)} N_h s_h)}, \quad \text{for all } h \in (I-Q)$$

and set $n_h' = n_h$.

I is the index set $\{1, 2, \cdots, L\}$ and

Q is a set containing all h such that $n_h > N_h$.

3. If $n_h' > N_h$ set $n_h' = N_h$ and go to step 2,

otherwise the optimum allocation is $n_1', n_2', n_3', \cdots, n_L'$.

APPENDIX B. PROGRAM LISTING

```
program expalc;
(* This program computes optimum univariate        *)
(* Sample Allocations for a stratified design.     *)
(* A generalized algorithm is implemented which    *)
(* will compute the correct allocation for the     *)
(* special case involving 100% sampling.           *)
(*                                                 *)
(*            JAMES WELDON MERGERSON               *)
(*            USDA-NASS-RAD-SSB-AFS                *)
(*             3251 Old Lee Highway                *)
(*              Fairfax, Virginia                  *)
(*                   22030                         *)
(*                703-235-3389                     *)
const
   size1 = 20;
type
   ivector = array [1..size1] of integer;
   rvector = array [1..size1] of real;
var
   cv, n, dv                       : real;
   cn, sd, cost, f, sn             : rvector;
   nstrata, est, i, j, redo        : integer;
   sum1, sum2, sum3                : real;
   isn, id                         : ivector;
(*                                                 *)
(* This program makes use of an advanced           *)
(* technique called mutual recursion.              *)
(* This technique requires making a FORWARD        *)
(* reference to the compiler.                      *)
(* The next line of code makes this                *)
(* FORWARD REFERENCE to the COMPILER.              *)
(*                                                 *)
procedure ComputeAllocations; forward;
(*                                                 *)
```

```
procedure CheckAllocations;
(* This procedure determine whether or not a          *)
(* computed sample size for a stratum exceeds         *)
(* the population size.                               *)
(* If the computed sample size is greater than        *)
(* the population size, the sample size for that      *)
(* stratum is set to equal the population size        *)
(* and the procedure ComputeAllocations is            *)
(* called.                                            *)
begin
   redo := 0;
   for i:=1 to nstrata do
    begin
       if sn[i] > cn[i] then
          begin
            sn[i] := cn[i];
            redo := 1;
            f[i] := 1
          end;
    end;
   if redo = 1 then ComputeAllocations
end;
(*                                                    *)
```

```
procedure ComputeAllocations;
(* This procedure computes optimum sample sizes.  *)
(* The procedure CheckAllocations is called to    *)
(* determine whether or not a sample size         *)
(* is greater than the corresponding population   *)
(* size for each stratum.                         *)
begin
   sum1 := 0;
   sum2 := 0;
   sum3 := 0;
   for j:=1 to nstrata do
       begin
         if f[j] = 0 then
            begin
              sum1 := sum1 + cn[j]*sd[j]*cost[j];
              sum2 := sum2 + cn[j]*sd[j]/cost[j];
              sum3 := sum3 + cn[j]*sd[j]*sd[j]
            end
       end;
   n := sum1*sum2/(dv + sum3);
   for j := 1 to nstrata do
       if f[j] = 0 then
          sn[j] := n*cn[j]*sd[j]/cost[j]/sum2;
   CheckAllocations
end;
(*                                                *)
```

```
procedure Initialize;
(*                                                    *)
begin
   writeln('enter number of strata');
   readln(nstrata);
   writeln('enter estimate and target cv');
   readln(est, cv);
   writeln('enter stratum-id cap-n cost std by stratum');
   writeln;
   begin
     for i := 1 to nstrata do
       begin
         readln(id[i], cn[i], cost[i], sd[i]);
         cost[i] := sqrt(cost[i]);
         f[i] := 0
       end;
   end;
   dv := cv*cv*est*est;
   writeln
end;
(*                                                    *)
procedure PrintAllocations;
(*                                                    *)
begin
   writeln;
   writeln('The OPTIMUM ALLOCATION is :');
   writeln;
   writeln('   strata   ' , 'sample size');
   for i := 1 to nstrata do
     begin
       isn[i] := round(sn[i]);
       writeln(id[i]:6, isn[i]:14)
     end
end;
(* -------------------------------------------- *)
(* -------------------------------------------- *)
begin (* Main Program *)
      Initialize;
      ComputeAllocations;
      PrintAllocations
end.  (* Expert Allocation Program *)
```