

SAS Macros for Sample Selection and Estimation

N. I. Lyons and K. Tam

Department of Statistics
University of Georgia
Athens, GA 30602 USA

ABSTRACT

A collection of SASTM macros for the selection of samples from finite populations is developed for most of the designs encountered in a typical course in survey sampling. The package also includes macros for estimation of means, totals, and proportions, for each of the designs, as well as for sample size determination and allocation. The macros are easy to use and to incorporate into other SAS programs. The package allows classroom demonstration of the significance of sampling error and error bounds, and allows students to compare the results of selecting samples under different sampling designs.

INTRODUCTION

Sample survey courses have been taught the same way for many years. Various sampling designs are introduced and formulas for estimates and standard errors are presented on the blackboard. In textbook exercises, student participation amounts to routine substitution of numbers into formulas. Many instructors assign projects which involve students in actual data collection and analysis. Although projects are extremely beneficial, they can take several weeks to complete. Thus only designs discussed early in the term can be used, and each student has first-hand experience with only one of many possible designs.

Chang, et al (1992) develop an interactive Fortran program called SURVEY which allows students to simulate the selection of a sample of households from a county. Any of several different sampling designs can be employed, and the output from the program is a data set with rows representing sampled households, and columns representing responses to a questionnaire. The authors recommend the use of a

statistical package to obtain estimates and standard errors.

The advantage of a program like this is that students are able to compare estimates of parameters obtained from data sampled under many different sampling designs in a short period of time. These estimates can be compared with known population values, and repeated performance of the sampling experiment demonstrates the importance of standard errors and error bounds. These concepts are fundamental to the subject area and, except in rare instances, can only be demonstrated by simulation.

The Chang program applies to a specific population of households in a fictitious county. It is beneficial to have a more versatile software package which could simulate sampling from many different types of populations. One option is to sample from a frame which is stored in a data file. If this approach is taken then one software package can be used to both simulate the sample selection, and obtain estimates and standard errors. Because of the versatility of the SAS language and procedures, it is often used in many upper-level statistical methods courses. Thus students are often already familiar with the language.

In this article we present a library of SAS macros which perform the sample selection and computation of estimates for all of the basic sampling designs encountered in an introductory survey sampling course. These macros can be used to analyze data for homework, for class projects, and classroom demonstrations.

SAMPLE SELECTION MACROS

There are three basic algorithms required to select samples according to most of the designs encountered in an

introductory sample survey course. The macros for each of these designs are included here in detail. Variations of these macros for more complicated designs and for two-stage designs may be obtained with only slight modification. The macros were written in a straight forward manner, so that they may be explained easily to students who have some knowledge of the SAS language, even though they may not be familiar with the SAS macro language. Each macro invokes the keyword type parameter list. We have found that this method is more user-friendly than the positional parameter method, because the parameters may be listed in any order, and because some parameters may optionally be omitted. Meaningful choices of the macro variable keywords actually helps students better understand how to invoke the macros. In each algorithm a sample of size n is selected from a finite population of size N .

Simple Random Sample Designs

A simple random sampling design (SRS) is any selection scheme in which every sample of size n has probability of selection equal to $1/C(N,n)$. The algorithm of Bissell (1986) was used because it is the most efficient when the population size is known. It is ideally suited to the SAS data step, because it requires only one pass through the frame for selection of the sample. Future work will involve developing similar macros for selection when the population size is unknown. Bissell's algorithm is as follows:

Step 1: For a given N and n , set $r = N - n$, and set dummy variables N' and r' initially to N and r , respectively.

Step 2: Generate $x = U(0,1)$ and set $p' = 1$.

Step 3: Replace p' by $p'r'/N'$.

Step 4: If $p' \leq x$, select item number $N - N' + 1$. Reduce N' by one. Retain r' at its current value. Return to step 2.

Step 5: If $p' > x$, reduce both N' and r' by one. Return to step 3.

```
%MACRO SRS(FRAME=, SETUP=, POPSIZE=,
  N=, IDNOS=, OUTSRS=, SEED=);
%IF %LENGTH(&OUTSRS) = 0 %THEN
  %LET OUTSRS = SAMPLE;
```

```
%IF %LENGTH(&POPSIZE) = 0 %THEN
  %LET POPSIZE=%STR(POPSIZE);
%IF %LENGTH(&FRAME) = 0 %THEN
  %LET FRAME = %STR(FRAME);
%IF %LENGTH(&SEED) = 0 %THEN
  %LET SEED = 0;
%IF %LENGTH(&SETUP) = 0 %THEN
  %LET SETIT = %STR();
%ELSE %LET SETIT = %STR(; SET
  &SETUP);
DATA TEMP &SETIT;
RETAIN NPRIME 0 PPRIME 1 RPRIME 0
  U 0;
FLAG = 1;
DO I = 1 TO &POPSIZE;

  IF I = 1 THEN DO;
    NPRIME = &POPSIZE;
    RPRIME = &POPSIZE - &N;
    U = RANUNI(&SEED);
    PPRIME = PPRIME*RPRIME/NPRIME;
  END;
  IF PPRIME > U THEN DO;
    RPRIME = RPRIME - 1;
    NPRIME = NPRIME - 1;
    IF NPRIME > 0 THEN DO;
      PPRIME = PPRIME*RPRIME/NPRIME;
    END;
  END;
  ELSE DO;
    OUTPUT;
    NPRIME = NPRIME - 1;
    PPRIME = 1;
    U = RANUNI(&SEED);
    IF NPRIME > 0 THEN DO;
      PPRIME = PPRIME*RPRIME/NPRIME;
    END;
  END;

  END;
  KEEP I FLAG;
%IF %LENGTH(&IDNOS) > 0 %THEN %DO;
  DATA &IDNOS; SET TEMP;
  ID = I;
  KEEP ID;
%END;
%ELSE %DO;
DATA &FRAME; SET &FRAME; I + 1;
FLAG = 0;
DATA &OUTSRS &FRAME;
MERGE &FRAME TEMP; BY I;
IF FLAG = 1 THEN OUTPUT &OUTSRS;
OUTPUT &FRAME;
DROP I FLAG;
PROC PRINT DATA = &OUTSRS;
TITLE1 "SIMPLE RANDOM SAMPLE DESIGN";
TITLE2 "FINAL SAMPLE SELECTED";
TITLE3 "OUTPUT DATASET = &OUTSRS";
DATA TEMP; SET TEMP; DELETE;
%END;
%MEND SRS;
```

Probability Proportional to Size Designs

In a probability proportional to size (PPS) design, the sampling units are selected with probability proportional to the value of some non-negative auxiliary variable X , called the size. The size variable must be observable for every element in the frame before the sample is selected. Sampling is done with replacement, so that a unit may appear in the sample more than once. For example, if unit i has auxiliary variable $x(i)$, then unit i is selected with probability equal to $x(i)$ divided by the sum of all $x(i)$, $i=1, \dots, N$. This is accomplished here by generating n random uniform numbers, $\{u(j) \mid j = 1, \dots, n\}$, and sorting them from smallest to largest. Then unit i is selected if $p(i-1) < u(j) \leq p(i)$, for each j , ($j=1, \dots, n$), with $p(0) = 0$. The SAS code to perform this sample selection is as follows:

```
%MACRO PPS(FRAME=, SETUP=, POPSIZE=,
  N=, SIZE=, OUTPPS=, OUTPI=, SEED=);
%IF %LENGTH(&POPSIZE) = 0 %THEN
  %LET POPSIZE = %STR(POPSIZE);
%IF %LENGTH(&OUTPPS) = 0 %THEN
  %LET OUTPPS = %STR(SAMPLE);
%IF %LENGTH(&FRAME) = 0 %THEN
  %LET FRAME = %STR(FRAME);
%IF %LENGTH(&OUTPI) = 0 %THEN
  %LET OUTPI = PI;
%IF %LENGTH(&SEED) = 0 %THEN
  %LET SEED = 0;
%IF %LENGTH(&SETUP) = 0 %THEN
  %LET SETIT = %STR();
%ELSE %LET SETIT = %STR(; SET
&SETUP);
DATA &FRAME; SET &FRAME;
  CUM + &SIZE;
DATA _TOTAL; SET &FRAME END =
  EOF;
  IF EOF = 1 THEN DO;
    TOT = _CUM_;
    OUTPUT;
  END;
  KEEP _TOT_;
DATA &FRAME; SET &FRAME;
  IF _N_ = 1 THEN SET TOTAL;
  U = _CUM_ / _TOT_;
  &OUTPI = &SIZE / _TOT_;
  DROP _TOT_;
DATA _UNIFS_ &SETIT;
  DO _I_ = 1 TO &N;
    U = RANUNI(&SEED);
    OUTPUT;
  END;
  KEEP _U_;
PROC SORT DATA = _UNIFS_;
  BY _U_;
```

```
DATA &OUTPPS;
  SET &FRAME _UNIFS_ BY _U_;
DATA &OUTPPS; SET &OUTPPS;
  RETAIN _COUNT_ 0;
  IF _CUM_ = . THEN
    _COUNT_ = _COUNT_ + 1;
  ELSE DO;
    IF _COUNT_ > 0 THEN DO;
      DO _I_ = 1 TO _COUNT_;
        OUTPUT;
      END;
      _COUNT_ = 0;
    END;
  END;
  DROP _COUNT_ _CUM_ _U_ _I_;
DATA &FRAME; SET &FRAME; DROP
  CUM _U_;
PROC PRINT DATA = &OUTPPS;
  TITLE1 "PPS DESIGN";
  TITLE2 "FINAL SAMPLE
  SELECTED";
  TITLE3 "OUTPUT DATA SET =
  &OUTPPS";
%MEND PPS;
```

Systematic Sampling Designs

A systematic 1-in- k sampling design (SYS) is selected by choosing a random starting point, and then selecting every k -th element in the frame, beginning with this random start, until the end of the frame. The design assumes that there is some ordering of the units in the population, such as the order the units were listed in the data set containing the frame.

The SAS code is given below for macro SYS. The user can enter either the value of k , or the desired sample size. The SAS program determines k in either case, chooses a random start, and selects the sample nearest n , determined by the combination of values of n and k .

```
%MACRO SYS(FRAME=, POPSIZE=, N=, K=,
  OUTSYS=, SEED=);
%IF %LENGTH(&FRAME) = 0 %THEN
  %LET FRAME = %STR(FRAME);
%IF %LENGTH(&POPSIZE) = 0 %THEN
  %LET POPSIZE = %STR(POPSIZE);
%IF %LENGTH(&OUTSYS) = 0 %THEN
  %LET OUTSYS = SAMPLE;
%IF %LENGTH(&SEED) = 0 %THEN
  %LET SEED = 0;
DATA _TEMP_;
%IF %LENGTH(&K) = 0 %THEN
  %LET K = %STR( INT(&POPSIZE/&N
  +.5) );
START = INT(RANUNI(&SEED)*&K)
  + 1;
```

```

N = INT((&POPSIZE-START)/&K)
  + 1;
DATA &OUTSYS; SET &FRAME
  END=EOF;
  IF _N_ = 1 THEN SET _TEMP_;
  IF _N_ = START THEN DO;
    START + &K;
    OUTPUT;
  END;
  DROP START N;
PROC PRINT DATA = &OUTSYS;
  TITLE1 "SYSTEMATIC SAMPLE";
  TITLE2 "FINAL SAMPLE
  SELECTED";
  TITLE3 "OUTPUT DATASET =
  &OUTSYS";
%MEND SYS;

```

EXAMPLE

To illustrate the simplicity of the main program a simple example is given. Suppose a list of population elements, or frame, is stored in a data file called reasearch.dat. The population has N = 400 elements (observations in the SAS data set), and we wish to select a SRS of n = 30 elements. Then the main program is as follows:

```

DATA POP_DAT;
  INFILE 'RESEARCH.DAT';
  INPUT 'list of variables in
  research.dat';

%SRS(FRAME=POP_DAT,
  POPSIZE=400, N=30);
RUN;

```

The result is a SAS data set called SAMPLE (by default) containing a SRS of 30 elements from pop_dat.

ESTIMATION AND ALLOCATION MACROS

Tam (1994) gives a complete SAS macro for determining sample size and allocation to strata, for several different allocation schemes, for the stratified random sampling design. SAS macros for the estimation of means, totals, and proportions for all of the designs are straightforward applications of MEANS and DATA step statements, and are not included here.

SAS is a registered trademark of SAS Institute, Inc. in the USA and other countries.

REFERENCES

- Bissell, A. (1986). Ordered random selection with replacement, *Applied Statistics*, 35: 73-75.
- Chang, T. C., Lohr, S. L., and McLaren, C. G. (1992). Teaching survey sampling using simulation, *The American Statistician*. 46: 232-237.
- Tam, K. K. (1994). SAS Macros for Sampling Methods. Masters Thesis, Department of Statistics, University of Georgia, Athens, GA.
- SAS Institute Inc. (1990). SAS Guide to Macro Processing, Version 6, Second Edition. Cary, N.C: SAS Institute, Inc.