INTRODUCTION

In some statistical applications, base SAS® software and SAS/STAT™ are not good choices for performing some types of statistical computations. The C programming language is more than an adequate alternative for some statistical computations. This paper illustrates a type of statistical computation in which C is a much better choice than base SAS® software or SAS/STAT™.

DISCUSSION

Suppose a sample is selected strictly proportional to the sizes of the sampling units and strictly without replacement. Let $x_i$ denote the probability that unit $U_i$ is included in the sample. Let $D$ denote the sample size and $p_i$ denote the probability that unit $U_i$ is selected on the first draw. A general formula [1] for computing $x_i$ is presented below:

$$x_i = p_i + \sum_{i \neq j} p_j (1 - p_j) + \sum_{i \neq j \neq k} p_k (1 - p_k) (1 - p_j) + \cdots + \sum_{i \neq j \neq k \neq \ldots} p_l (1 - p_l) (1 - p_k) (1 - p_j) \cdots (1 - p_i).$$

The inclusion probabilities cannot be computed using SAS/STAT™. The use of base SAS® software is a very poor choice. The algorithm is easily programmed in C. A generalized program (1<n<6) for performing the computations is presented in the appendix.

REFERENCE


APPENDIX. PROGRAM LISTING

/* Compute Inclusion Probabilities */
#include <stdio.h>
#define max 100
main ( ){

/* INITIALIZATION AND INPUT */
int icnt,capn,srn,ni,n2,n3,n4,n5;
int i1,i2,i3,i4,i5;
float pi(max), p(max);
double sum1,sum2,sum3,sum4;
double temp1,temp2,temp3,temp4;
int i=0; ni=0; n2=0; n3=0; n4=0; n5=0; icnt=0;
printf("enter capn "); scanf("%d", &capn);
printf(" enter srnn "); scanf("%d", &srnn);
printf(" enter srnn = %d ");
if (srnn>=1) ni=capn;
if (srnn>=2) n2=capn;
if (srnn>=3) n3=capn;
if (srnn>=4) n4=capn;
if (srnn>=5) n5=capn;
i=1; while (i1<capn) {
    printf("enter p[%d] "); scanf("%f", &p[i1]);
i1++;
}

/* MAIN ALGORITHM */
i1=1; while (i1 <= capn) { icnt++;
    temp1=pi[1]=0.0;
    sum1=pi[1]=temp1;
    if (i1 != i3) {
        temp2=p[i1]*i1; sum2=sum2+temp2;
    }
    i2=1; while (i2 <= n2) { icnt++;
        temp3=p[i1]*i2; sum3=sum3+temp3;
        if (i2 != i3) {
            temp4=p[i1]*i2*temp3;
            sum4=sum4+temp4;
        }
    }
    if (i1 <= n5) {
        pi[i1]=pi[i1]*sum1*sum2*sum3*sum4;
        i1++;
    }
}

/* OUTPUT */
for (i1=1; i1 <= capn; i1++) { sum1 += pi[i1];
    printf("sum = %8.4f, ", sum1);
    printf("iterations = %d 
"), icnt);
for (i1=1; i1 <= capn; i1++) {
    sum1 = pi[i1];
    printf("pi[%d] = %8.4f ", i1, sum1);
    printf(" ");
}