SUDAAN: A COMPREHENSIVE SOFTWARE PACKAGE FOR SURVEY DATA ANALYSIS

Lisa M. LaVange, Babubhai V. Shah, Beth G. Barnwell and Joyce F. Killinger
Lisa M. LaVange, Research Triangle Institute

KEYWORDS: variance estimation, user interface, C language

ABSTRACT

The development of a survey data analysis software package that is flexible enough to meet the needs of survey statisticians, as well as research scientists not necessarily schooled in survey sampling, is underway at the Research Triangle Institute. The system was designed to provide significant enhancements to RTI's existing software package, SUDAAN. The new SUDAAN system is written in the C language and provides greater efficiency and portability. The software development strategy consists of developing and debugging stand alone statistical procedures on an IBM PC with expanded capacity. The procedures are then implemented in two additional computing environments, VAX/VMS and IBM/OS. Variance estimation options and data analysis techniques not previously available in the RTI software package are among the many modifications incorporated into the new SUDAAN system. The purpose of this paper is to describe the design and development of this system.

INTRODUCTION

Scientists at the Research Triangle Institute have been involved in designing, developing, and maintaining software systems for the analysis of survey data for the past 17 years. Recently, RTI has been under contract to the Public Health Service (PHS) to develop a comprehensive software package that meets the needs of statistical analysts at the National Center for Health Statistics (NCHS) and PHS. In particular, this package must be flexible enough to handle most of the statistical designs used by these agencies. For this purpose, we have embarked on the task of developing a system that incorporates many of the features of RTI's existing survey data analysis system but also includes significant enhancements. The purpose of this paper is to describe in detail the design and development of this comprehensive software package. The immediate ojective of the SUDAAN system design was to develop a series of procedures capable of performing statistical data analysis for complex sample surveys. The ultimate objective was to have a system that could aid in the design and evaluation of new statistical techniques.

RTI's existing software system consists of a series of procedures that produce statistical analyses in the form of weighted cross-tabulations, generalized ratio estimators, and linear and logistic regressions. All of the procedures were constructed using a unified set of FORTRAN subroutines that execute in the SAS computing environment on an IBM mainframe. In the course of developing a new system, we went through the process of rethinking our current design. Four goals emerged:

- Portability
- Reliability/numerical accuracy
- Computational efficiency
- Ease of modification/enhancement.

The system design for the comprehensive survey data analysis software, SUDAAN, evolved with these goals in mind. The first goal is being met by writing the new software entirely in the C programming language and testing simultaneously on several systems. Compilers for the C language are available on a variety of operating systems. The initial programs are currently executing under the TURBO C compiler on an IBM personal computer, the SVC compiler on a 68020, and under the UNIX operating system on a GOULD computer. Versions are also being tested on the VAX/VMS and IBM/VS.

The second goal is being met by building checks into the system to assure that definitions of objects are meaningful and computations are feasible. We have implemented numerically stable procedures for the accumulation of sums of squares and cross products and for matrix operations, including matrix inversion.

For achieving the third goal of improved efficiency, special care has been taken in the design to isolate tasks that are repeated numerous times, such as a numerical calculation on each variable of each record. These sections of code are optimized to the extent possible without using assembly language. Some of the optimizing algorithms include processing of sparse matrices and vectors, replacement of multiplication by table lookup with addition, elimination of repetitive (in the loop) "if" tests by constructing separate do loops, and many others.

An important product of the SUDAAN design effort has been the creation of a high level statistical programming language. The SUDAAN language has been developed to provide the statistician/programmer with easy access to the software package, thus accomplishing the fourth goal listed above. Statistical procedures are currently being written in the SUDAAN language. Once they are finalized, some parts may be converted to C programs to maximize computing efficiency.

The SUDAAN design allows for four types of user interface as depicted in Figure 1. The general user will be able to access the system through user friendly procedures. Ultimately the procedures will be available through a statistical system, such as SAS, but the current design consists of stand-alone procedures that interface with a standard ASCII file format. The more sophisticated user can write SUDAAN programs for computations not available in the procedure library. This access feature is particularly well suited for testing modifications to existing procedures, such as the calculation of an alternative test statistic. A C programmer can access the system directly through C callable functions.

The key features of the SUDAAN system are described in detail in the following sections.

RTI SURVEY DATA ANALYSIS PROCEDURES

BACKGROUND

The collaboration of statisticians and

computer scientists at RTI, beginning in 1971 (e.g., Folsom, Bayless, and Shah, 1971) and continuing today, has resulted in a series of programs for survey data analysis that reflects state-of-the-art methodology in this field. All of RTI's analysis packages employ the Taylor series or Delta method of variance estimation for complex sample survey designs (Folsom, 1974). Each program currently marketed is designed to run within the SAS framework.

RTI is currently marketing the following procedures for general use:

- SESUDAAN
- RATIOEST
- RTIFREQS
- SURREGR
- RTILOGIT.

The SESUDAAN (Shah, 1981) procedure produces weighted estimates of means, totals, and proportions and their standard errors. RATIOEST (Shah, 1982) calculates weighted frequency distributions and percentages for cross-tabulations.

RTI's survey regression package, SURREGR (Holt, 1977), provides estimates of linear regression coefficients, their estimated variance-covariance matrix, and tests of hypotheses concerning the coefficients that are appropriate for a complex sample design. Logistic regression capabilities became available with the development of RTILOGIT (Shah, et al., 1984). This procedure runs in conjunction with SAS PROC LOGIST and produces pseudo maximum likelihood estimates of the parameters and tests of hypotheses about these parameters.

New System Procedures

The SUDAAN system design consists of two phases of software development, each resulting in a series of statistical procedures. The Phase I procedures are primarily concerned with descriptive data analysis, including cross-tabulations, generalized ratio estimation, and quantile estimation. The Phase II design, currently in the planning stages, will incorporate several analytical procedures, including linear and log-linear modeling and survivorship analysis of survey data.

Several features are available under the new SUDAAN system that were not previously available with RTI software. Key among these are multiple variance options. Sampling variances are computed for one of three design options invoked by the user. These are:

- With replacement sampling at the first stage
- Simple random sampling at each stage (with or without replacement)
- Without replacement, unequal probability sampling at the first stage and simple random sampling (with or without replacement) at subsequent stages.

The second option requires the input of stratum specific population and sample counts for each stage. In addition to these counts, the third option requires the input of joint probabilities of selection for each pair of primary sampling units within a stratum. These quantities are used to produce Yates-Grundy-Sen variance estimates for this design option. The three variance options cover a wide range of sample designs including sampling with probabilities proportional to size at the first stage. Unequal probabilities of selection at a subsequent stage can be handled by forming pseudo strata at that stage since probabilities of selection can vary from stratum to stratum. In addition to these variance options, the user may request that the variance-covariance matrix for estimates within a table be computed for all procedures allowing for cross-tabulations. Estimates of variance-covariance matrices were previously available only in the SURREGR and RTILOGIT procedures.

Phase I software consists of three procedures: CROSSTAB, DESCRIPT, and RATIO. The CROSSTAB procedure is the RTIFREQS equivalent in the new SUDAAN system. In addition to computing weighted frequency and percentage distributions, Chi-square tests of independence for a two-way contingency table are produced. The tests are computed using a Wald statistic according to methods advocated by Koch, Freeman, and Freeman (1976). During Phase II, the Rao-Scott Satterthwaite corrected test statistic (Thomas and Rao, 1984 and 1985) will be computed in addition to the Wald statistic for goodness of fit tests. This statistic has been shown by Thomas and Rao to have the best characteristics with respect to both power and significance level when compared with other statistics correcting for the liberalness of the Wald test.

The DESCRIPT procedure computes estimates of means, totals, proportions, geometric means, quantiles, and their sampling errors. Estimates can be requested for the subdomains of user-specified cross-tabulations. Options are also available to the user for standardized and/or post-stratified means and proportions and their sampling errors. The standardizing and/or post-stratifying variables and their distributions are required as input.

RTI has recently completed a simulation study comparing two methods for estimating the variances of quantiles for sample survey data (Wheeless, et al., 1988). The two methods considered were (1) the one proposed by Fuller and Francisco (1986) and currently used in the PC CARP software package and (2) a direct linearization method proposed by RTI. The simulation also compared two methods of estimating quantiles themselves. Based on the results of this study, the direct linearization method is currently being programmed for use in the DESCRIPT procedure.

The user can request estimates of linear contrasts among the domain estimates produced by DESCRIPT. Four types of contrast specifications are available. The user can specify a general linear contrast, all possible pairs of differences among levels of a domain variable, the differences between levels of a domain variable and the marginal value, and polymonial effects for an ordinal domain variable.

The RATIO procedure computes the estimates of generalized ratios for survey data. The user can specify numerator and denominator variables that are continuous or categorical. In the latter case, levels of the variables for which

ratios are to be estimated are required. This procedure, as the DESCRIPT procedure, can be executed in one of three modes: basic estimation, standardized estimation, or post-stratified estimation.

The current plan for Phase II software development consists of four statistical procedures written in the new SUDAAN framework. The first of these is a general categorical data analysis procedure that will offer log-linear modeling capabilities for survey data analysis. This procedure will be a survey data analogue to the SAS procedure CATMOD. Procedures two and three will be linear and logistic regression procedures, developed based on the existing procedures SURREGR and LOGIST. Each of these will be reprogrammed in the new SUDAAN system and various enhancements added. Enhancements to the linear regression procedure are being planned for stochastic regression coefficients model analysis of survey data. Software has been developed at RTI under a contract to the National Institute for Child Health and Human Development for this problem (LaVange, et al., 1988) that will be incorporated into the SUDAAN system. Fourth, a general procedure for survival data analysis is planned that will allow the analyst to perform Kaplan-Meir estimation and Cox proportional hazard model estimation using complex sample survey data.

SUDAAN SYSTEM DESIGN

The immediate objective of the SUDAAN system design was to develop a series of procedures capable of performing statistical data analysis for complex sample surveys. The ultimate objective was to have a system that could aid in the design and evaluation of new statistical techniques. The system design for SUDAAN consists of three layers:

(1) PROCs or procedures similar to SAS or BMDP procedures
(2) SUDAAN data structures and functions operating on these data structures
(3) A system level interpreter for (1) and (2).

These layers correspond to different views of the system. SUDAAN may be perceived quite differently by different groups of users. In this section we discuss these views and describe in detail the major components of the SUDAAN system.

System Views

A substantive researcher or data analyst will view SUDAAN as a collection of statistical procedures for applying alternative statistical techniques for survey data analysis. A statistician may view it as a convenient tool for evaluating new statistical formulas or analysis techniques. SUDAAN allows for easy conversion of formulas into executable statements, thereby enabling new procedures or enhancements to existing procedures to be programmed quickly and efficiently.

A system designer may view it as a compiler, written in C, for a very high level language in order to specify statistical analysis tasks. For the computer scientist, the development effort consisted of designing an optimizing compiler or interpreter with the potential for global optimization of all available resources. The SUDAAN program defines only what is to be done, not how. The system is free to select any of the possible sequences of operations in order to achieve the user-defined result.

The PROC interface is in SUDAAN for the convenience of the user who repeatedly applies the same procedures to different datasets or different variables within the same dataset. The systems programmer's view represents one of the many possible compiler designs for the implementation of SUDAAN.

However, the primary force behind the SUDAAN design is the statistician's view, namely a tool for translating mathematical formulas into a computer program capable of producing numerical results for a given data configuration. The statistician need not specify how a particular estimator is to be calculated. Rather his focus is on what is to be calculated. For the statistician, SUDAAN is a "nonprocedural" language. Given the definition of a statistical quantity in terms of a mathematical formula, the system determines the computational algorithm. Users have no knowledge of the representation of the data structures in memory or on an input/output device and no control of the program flow. In this sense the SUDAAN language is viewed as definitional and not procedural.

The basic elements of the system are as follows:

- Statistical data objects and abstract data structures
- Functions of and operations to data structures
- Functions to input, output, and print data structures.

Each of these is described in turn in the following subsections.

Statistical Data Structures

An example of a statistical data structure is the dataset resulting from a national household survey. Such a dataset might consist of records containing information on a sample of individuals, each selected from a sample of households within enumeration districts, counties, and states. The actual dataset contains a record for each individual, consisting of various attributes and item responses for that individual and household. Each record is identified by several hierarchical identifiers corresponding to the stages of sample selection: state, county, enumeration district, household, and individual. An example of a different type of data structure is a summary dataset of county-specific population counts of individuals by characteristics, such as race and sex. Such a dataset is a collection of two dimensional tables or matrices with the two hierarchical identifiers state and county.

In SUDAAN, statistical data structures are defined as collections of multidimensional arrays with nested hierarchical identifiers. These data structures are referred to as Balanced Array Trees (BATs). Each of the datasets described above is an example of a BAT, the

first with five nested identifiers and the second with only two. The SUDAAN language also consists of a series of functions and mathematical operators that are used to input and output BATs and to define BATs as functions of BATs and other input objects. The SUDAAN language itself is a mathematically closed set of BATs with respect to these functions. Applying a function to a BAT yields another BAT. The level of nested identifiers may change, but one or more BATs always result.

SUDAAN Program Examples

In order to illustrate the SUDAAN language concepts, consider the problem of computing the between primary sampling unit (PSU), within stratum mean square for an estimated population total. Let y(hijk) denote the observed values of the variable y collected from a nested sample design for the kth individual in the jth household, ith PSU, and hth stratum. The filename "EXAMPLE1" refers to an ASCII file with the nested sample identifiers included as variables one to four on the file. The variable y is the fifth variable on the file. The input file is stored in a standard file format with a codebook describing the variables and associated levels. Variable labels or formats are automatically read in and used for printing. In the following statements, X refers to the BAT consisting of these data values. The program uses the summation function, SIGMA, to compute totals of y at various levels. The first argument of SIGMA refers to the BAT containing data values to be summed, while the second argument refers to the nesting level up to which the summing takes place. This argument equals the number of nested identifiers in the resulting BAT that contains the sums. Comments, delimited by the strings "/*" and "*/", further clarify the logic of the following program statements shown in Figure 2.

Since the SUDAAN language deals with symbolic definition only, the above statements can easily be modified to obtain the mean square errors associated with more than one variable. Replacing the second statement with the following results in the calculation of mean square errors for variables statement 5, 6, 9, and 10 in this example.

```
X = SELECT(RECORDS, VECTOR(5,6,9,10));
```

Next, consider the problem of computing mean square errors for estimated population counts corresponding to the cells of a multidimensional table. In the following example statements, TABLES refers to a BAT consisting of a two-way and a three-way table. The first is the cross-classification of variables 5 and 6 on the input file and the second is the cross-classification of variables 5, 9, and 10. The EFFECTS function converts categorical variables with specified numbers of levels into dummy (0,1) vectors. The CROSSP function defines the tables as cross products of these effect vectors. SIGMAWG returns two arguments, the unweighted and weighted sums for each cell in the tables. Once XSUM2 is defined, the program statements in the above example follow to compute the mean square errors. The required statements are shown in Figure 3.:

SUDAAN Procedures

The task of developing new procedures in SUDAAN is fairly straightforward. The process involves writing a SUDAAN program and associating objects in the program with a well defined set of key words. The user-input to a PROC consists of a command level syntax, such as that used in IBM/TSO or SAS. The PROC processor interprets the user input and generates the necessary statements for the SUDAAN language parser. In short, writing a SUDAAN procedure is equivalent to defining a preprocessor for a SUDAAN program.

The major advantage of this approach is that the user has a SUDAAN program available for each procedure that serves as detailed documentation. The user can see the translation of input to output and the series of calculations that are performed once the procedure has been called. Modifications to the procedure can be made by modifying the accompanying SUDAAN program without writing a completely new procedure. The source for all of the SUDAAN procedures will be available to the user.

CONCLUDING REMARKS

The unique feature of SUDAAN is the entity defined as a BAT. Mathematically, a BAT represents a finite sequence of multidimensional arrays. The set of BATs represents a closed set with respect to binary operations and functions whose arguments are BATs. The functional syntax is "natural" for a mathematical statistician who can use the SUDAAN language to define the BATs of interest.

A SUDAAN user will have the SUDAAN source code available as part of the documentation. By referring to the SUDAAN program for a statistical procedure, the analyst will be able to determine exactly what underlying mathematical formulas are employed. When statistical research indicates new or improved formulas for a given analysis, the formulas can be readily translated into the SUDAAN language to yield a working computer program.

In writing a SUDAAN program, the user only defines the computational objects through function cells. The user does not need to specify how the calculations will be carried out. Increased reliability is achieved through checks built into the system to assure that the definitions are meaningful and that the objects required as output can be generated. The system is free to select the computational algorithm that is optimal given the dataset and available computing resources. The system also provides tremendous flexibility with respect to input and output. All input/output objects are defined as BATs. Any intermediate BAT can be saved or displayed at any time in the program.

The use of BATs, with the above stated advantages, has some restrictions and limitations. The user is limited to the BAT functions currently implemented, and all computations must be represented as BATs or functions thereof.

It is possible to test new functions independently before incorporating them into the system, thereby greatly facilitating the addition of functions to the system library on an as-needed basis.

Computations that cannot be represented as BATs will not be feasible in SUDAAN. The user must define all computations as functions of input BATs or data objects. The SUDAAN system is therefore not for general programming use, but should prove reasonably adequate for statistical analysis, and in particular, survey data analysis.

A major portion of the overall software development effort at RTI has been channeled into the SUDAAN system design. While this strategy has slowed the development of the statistical procedures initially, we are anticipating a tremendous savings in the long run due to the ease with which each procedure can be designed, implemented, and debugged. The end result of this effort will be not only a user-friendly software package that runs efficiently but also a valuable aid to the statistician for enhancing existing procedures and planning for additional procedures as new methodology becomes available.

REFERENCES

COX, D. R. (1972). "Regression Models and Life Tables" (with discussion), Journal of the Royal Statistical Society B, 34: 187-220.

FOLSOM, R. E., BAYLESS, D. L., and SHAH, B. V. (1971). "Jackknifing for Variance Components in Complex Sample Survey Designs." Proceedings of the American Statistical Association: 36-39.

FOLSOM, R. E. (1974). "National Assessment Approach to Sampling Error Estimation. Sampling Error Monograph. Prepared for the National Assessment of Educational Progress.

FRANCISCO, C. A. and FULLER, W. A. (1986) "Estimation of the Distribution Function with Data from a Complex Survey." Presented at the American Statistical Association Meetings, Chicago, Illinois.

HOLT, M. M. (1977). SURREGR: Standard Errors of Regression Coefficients from Sample Survey Data. Research Triangle Institute, Research Triangle Park, NC 27709.

KAPLAN, E. L. and MEIER, P. (1958). "Nonparametric Estimation from Incomplete Observations." J. of the American Statistical Association, 53: 457-481.

KOCH, G. G., FREEMAN, D. H., JR. and FREEMAN, J. L. (1975). "Strategies in the Multivariate Analysis of Data from Complex Surveys." Inter. Stat. Rev. 43, 59-78.

LAVANGE, L. M., PFEFFERMANN, D., SHAH, B. V., and GABEL, T. J. (1988). "The Analysis of Relationships Between Growth and Dietary Intake Using the NHANES Dataset." Final Project Report. Research Triangle Institute, Research Triangle Park, NC 27709.

SHAH, B. V., HOLT, M. M. and FOLSOM, R. E. (1977), "Inference about Regression Models from Sample Survey Data." Proceedings of the 3rd Meeting, International Association of Survey Statisticians, New Delhi.

SHAH, B. V. (1981). RATIOEST: Standard Errors Program for Computing Ratio Estimates from Sample Survey Data. Research Triangle Institute, Research Triangle Park, NC 27709.

SHAH, B. V. (1982). RTIFREQS: Program to Compute Weighted Frequencies, Percentages, and Their Standard Errors. Research Triangle Institute, Research Triangle Park, NC 27709.

SHAH, B. V. (1981). SESUDAAN: Standard Errors Program for Computing Standardized Rates from Sample Survey Data. Research Triangle Institute, Research Triangle Park, NC 27709.

SHAH, B. V. (1978). "SUDANN: Survey Data Analysis Software." Presented at the Joint Statistical Meetings of the American Statistical Association, San Diego.

SHAH, B. V. (1979). VCMPNLS: Program to Compute Variance Components. Research Triangle Institute, Research Triangle Park, NC 27709.

SHAH, B. V., FOLSOM, R.E. HARRELL, F. E. and DILLARD, C. N. (1984): "RTILOGIT: Survey Data Analysis Software for Logistic Regression," Final Report, Work Assignment 74, Research Triangle Institute, Research Triangle Park, NC.

THOMAS, D. R. and RAO, J. N. K. (1984). "A Monte Carlo Study of Exact Levels of Goodness-of-Fit Statistics Under Cluster Sampling." Proceedings of the Section on Survey Research Methods, American Statistical Association, Washington, DC.

THOMAS, D. R. and RAO, J. N. K. (1985). "On the Power of Some Goodness-of-Fit Tests Under Cluster Sampling." Proceedings of the Section on Survey Research Methods, American Statistical Association, Washington, DC.

WHEELESS, S. and SHAH, B. V. (1982), "RATIO2 and RATIO3: Computing Ratio Estimates with Two Data Files. Research Triangle Institute, Research Triangle Park, NC.

WHEELESS, S. C., SHAH, B. V., and LAVANGE, L. M. (1987). "Results of a Simulation for Comparing Two Methods for Estimating Quantiles and their Variances." Project Report. Research Triangle Institute, Research Triangle Park, NC 27709.

Figure 1. User Access to the SUDAAN System

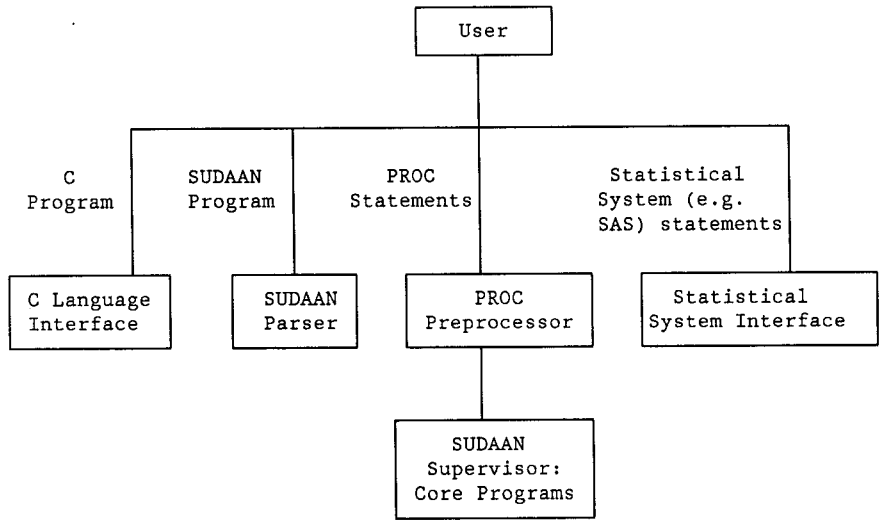

Figure 2

```
RECORDS = FICSFILE("EXAMPLE1",
          VECTOR(1,2,3,4));          /*RECORDS is a BAT containing the
                                       input dataset*/

X = SELECT(RECORDS,5);               /*X is a BAT of y values and id's */

PSULEVEL=2;
XSUM2=SIGMA(X,PSULEVEL);             /*Sum X over each PSU */
MH = SIGMA(CONSTANT(XSUM2,1),1);     /*Compute the number of PSUs
                                       per stratum by defining a vector
                                       of one's for each PSU in XSUM2*/

MHMIN1 = MH - 1;
STRATLEV = 1;
XSUM1 = SIGMA(XSUM2,STRATLEV);       /*Sum X over each stratum */
XSQR1 = SIGMA(XSUM2*XSUM2,
              STRATLEV);             /*Stratum sum of squares of
                                       PSU totals */
XSSO = SIGMA((MH * XSQR1 -
       XSUM1*XSUM1)/MHMIN1,0);       /*Compute sum of squared deviations
                                       over all strata */

MSE = SQRT(XSSO);
PRTABS(MSE);                         /*Print the result */
```

Figure 3

```
RECORDS = FICSFILE("EXAMPLE1", VECTOR(1,2,3,4));
SUBGROUPS = VECTOR(5,6,9,10);      /*Variables defining subgroups */
LEVELS = VECTOR(7,2,2,2);          /*Associated variable levels */
OPTIONS = VECTOR(0,0,0,0);         /*Four options are off */
EFFVECTORS = EFFECTS(RECORDS,SUBGROUPS,LEVELS,OPTIONS);
TAB1 = VECTOR(1,2);                /*Variable 5 by variable 6 */
TAB2 = VECTOR(1,3,4);              /*Variable 5 by var 9 by var 10 */
TABVEC = VECTOR(TAB1,TAB2);
TABLES = CROSSP(EFFVECTORS, TABVEC);
WEIGHT = SELECT(RECORDS,11)        /*Variable 11 is the weight */
NWSUM = SIGMAWG(TABLES, PSULEVEL, WEIGHT);
XSUM2 = NWSUM(2);
```