VPLX: VARIANCE ESTIMATES FOR COMPLEX SAMPLES

Robert E. Fay, U.S. Bureau of the Census
Washington, DC 20233

Key Words: replication, jackknife, software

Abstract VPLX is a portable FORTRAN program in the public domain to estimate variances, covariances, and related statistics for complex samples through replication methods. The software has been designed particularly to address applications within the Census Bureau and by users of the Census Bureau's public use files, but the general nature of the program should make it of interest to many other survey researchers. VPLX accommodates variance calculations through replicate weights or factors, or through survey identifiers such as stratum, PSU, and secondary sampling unit. The user-oriented command language is similar in many respects to SPSS and SAS. The paper summarizes key features and future enhancements.

1. Introduction.

There are two major strategies for estimating variances for statistics from complex samples: Taylor-series linearization and replication. Recent advancements in general statistical packages for variance estimation through these two methods, including SUDAAN (Shah 1989), PC-CARP (Fuller *et. al.* 1986), and WESVAR (Flyer, Rust, and Morganstein 1989), have facilitated the routine estimation of variances from complex samples, helping to ameliorate the situation in survey research in which appropriate estimates of variance were often never calculated from many complex samples or were unavailable on a timely basis.

This paper describes development of another general system for variance estimation, VPLX. Like WESVAR, VPLX variance estimates are based on replication. As a late entry into the field, VPLX requires some justification for its appearance, and key features of VPLX will be described that should enable greater ability or flexibility to address special classes of problems. Most of the focus here will be on the capabilities of the current program, but an outline of projected additions will indicate how the system has been designed to grow.

The program is written in portable FORTRAN 77 and can easily be installed on a variety of mainframe, minicomputer, and microcomputer environments. The source code is in the public domain. Thus far, the program has in IBM CMS, VAX (VMS), IBM PC/DOS (with Microsoft Fortran 4.1 and 5.0), Univac, and Sun (under SunOS, a version of UNIX), environments, without encountering any significant problems in portability.

The program supports several different replication methods. VPLX will create simple or stratified jackknife replicates based on survey identifiers such as stratum, primary sampling unit, secondary sampling unit, and cluster, for a variety of one- and two-stage sample designs, including provision for finite population correction factors. Public-use files from the Census Bureau's Survey of Income and Program Participation (SIPP) may be analyzed with this option. Additionally, half-sample or random group replication can also be achieved by including replicate number identifiers on the data file.

Alternatively, replicates may be defined through replicate weights or replicate factors. This option provides a means to represent a diverse collection of replication methods, including the procedures now in use at the Census Bureau to estimate variances from the demographic surveys. This option also facilitates approaches to combine design-based variance estimation with multiple imputation to represent both the sampling variability and the variance due to estimation of missing data (Fay 1989).

Replication refers to a number of related variance estimation techniques, but VPLX organizes these calculations into a single common representation. For each estimate X_0 based on the full sample, estimates X_1, X_2, ..., X_R are derived by partial reuse of the sample. The manner in which these estimates are defined depends on the replication method. The underlying variance formula implemented by VPLX takes the form:

$$V^*(X_0) = \sum_{r=1}^{R} b_r (X_r - X_0)^2 \quad (1)$$

where b_r, r=1,...,R, denotes a set of coefficients that may depend on r but not on the specific characteristic

X_0. VPLX determines the appropriate b_r for the jackknife, stratified jackknife, random group and half-sample methods, but the user provides these coefficients as part of the command syntax for generalized replication.

Replicate calculations in VPLX are organized into a number of steps:

1. A CREATE step, which, on the basis of instructions from the command file, builds a file of replicate totals, called a VPLX tally file, used as input to the remaining steps. This file contains tallies for the overall sample and each replicate sample; names, labels, and characteristics of each variable; coefficients to be used in the variance estimation formula; and all other information critical for carrying out the calculations in the remaining steps.
2. The TRANSFORM step reads as input a VPLX tally file and can create a second VPLX tally file with any or all of the information from the first plus new variables created in the step as functions of estimated totals. A ratio estimate is one such statistic, but considerably more complex statistics are possible. In the current implementation, new variables are created through calls to user-supplied FORTRAN subroutines, but future versions of VPLX will also provide a number of standard functions of totals.
3. The DISPLAY step is able to process various requests for variances, covariances, correlations, and t-tests based on a VPLX tally file. The DISPLAY step may immediately follow a CREATE step or be run separately, as often as desired, provided that the VPLX tally file has been saved. The DISPLAY step normally writes results to the print file but can create formatted output files of results that can be readily input to spreadsheet programs, SAS, or other statistical or tabulation systems.
4. A CONTENTS step displays the variable names, variable labels, types of variables, labels for categorical levels, replication method, and other descriptive information stored on a VPLX tally file. This optional step may be run for a VPLX tally file at any point.

The steps communicate with each other through information stored on files rather than by retaining information from one step to the next during an execution of VPLX. A key feature of this design is that the steps may be run separately. Typical experience is that the CREATE step demands the most computer resources; once this step has been completed, then multiple executions of the DISPLAY step or combinations of the TRANSFORM and DISPLAY steps are possible.

This design resembles features of both SAS, SPSS-X, and some other statistical systems. In SAS, a key feature of the DATA step is to bring a user's data into the SAS system by creating a SAS system file. (Development of data engines in SAS make this only one of a number of alternative ways for SAS to deal with users' data, but the parallels to be offered here are based on SAS data sets.) The analogue in VPLX is the CREATE step, used to read a user's data and to create a VPLX tally file for further analysis. The chief parallels between the SAS DATA step and the CREATE step involve the description of input data, the definition of variable names and, optionally, variable labels, and the creation of an output file with the critical information for further analyses by the statistical system.

Although there are similarities between the SAS DATA step and the VPLX CREATE step, there are a number of differences as well. The key differences are:

1. The SAS DATA step creates a file with individual-level data, whereas the CREATE step produces a file with aggregate tallies organized for further replicate calculation.
2. All critical information required for variance calculation is identified in the CREATE step, and this information is retained on the VPLX tally file. In SAS, subsequent PROCs determine the procedures to be used for variance calculations, which generally assume simple random sampling.
3. In VPLX, different types of categorical data are distinguished within the CREATE step. Labels for levels of categorical data become associated with the VPLX tally file. SAS makes this association optional in the DATA step, since it offers the alternative of storing categorical information in separate FORMAT files.
4. Cross-classifications must be anticipated in the CREATE step, unlike the SAS DATA step. Variables in VPLX may be vectors storing cross-classified data as well as univariate characteristics. Basically, if cross-classifications are required in later steps, they must be formed within the

CREATE step. There are capabilities in both the TRANSFORM and DISPLAY step to reduce detailed cross-classifications later; for example, cross-classifications formed through declaring CLASS variables can be reduced to different marginal totals in both the TRANSFORM and DISPLAY steps. A cross-classification cannot be constructed later if the necessary detail is not obtained from the CREATE step, however. In SAS, cross-classifications are formed later in SAS procedures, such as PROC FREQ and PROC TABULATE.

The command language for the program is generally free-format and resembles in many respects the languages of SAS and SPSS-X. Key words for commands begin in column 1, and almost all statements may be extended onto an arbitrary number of lines by avoiding column 1 on the continuation lines. Variables may be named by up to 12 characters. No distinction is made between upper and lower case in interpreting statements or variable names; for example, V1 and v1 are treated as equivalent.

The next three sections of the paper will describe general characteristics of the CREATE, TRANSFORM, and DISPLAY steps, which will serve to illustrate the overall capabilities of the current system. The following section outlines anticipated enhancements. A concluding section briefly comments on the merits of the VPLX design with those of SUDAAN, PC-CARP, and WESVAR.

2. The CREATE Step

The specification of a CREATE step begins with a CREATE statement specifying the input and output files, for example,

```
create  in = d:\survey\datafile.dat
     out = d:\survey\datafile.vpl
```

Two additional statements are essential to the specification of a CREATE step: an INPUT statement listing names of variables to be read from the input file and a FORMAT statement providing a FORTRAN format by which the variables may be read. To illustrate, in

```
input  sexw2  agew2 degree  degreex tm8428
     racew2 workact waitw2
     totinc earnings  repf1 - repf100
format  (7x,f1.0,f3.0,3x,f1.0,1x,f1.0,f2.0,
         2f1.0,f8.5,2f10.1,12(/,8f10.7),/,4f10.7)
```

the INPUT statement defines 110 variables to be read from the input file, and the FORMAT statement that follows specifies their locations on a file with 14 records per case.

The choice of replication method is also determined during the CREATE step. There are sensible defaults: if VPLX encounters a variable name CLUSTER, it will implement a simple jackknife based on cluster membership defined by this variable, in the absence of other information indicating that a different method is intended. Similarly, the appearance of variable names STRATUM and CLUSTER will imply a choice of the stratified jackknife option. Explicit statements are used to specify other replication methods.

The CREATE step has capabilities to produce simple or extensive cross-classifications through a powerful but simple syntax. Incoming variables may be changed into categorical through statements such as:

```
cat  q103 - q117 q121 q123 (1/2/3/res) 'Yes',
   'No','Don''t know','Missing/no answer'
cat income into incomecat (low- -1/0/1-9999/
   10000-24999/25000-49000/50000-high) 'Loss'
   'None' '$1-9,999' '$10,000-24,999'
   '25,000-49,999' '$50,000+'
```

where the syntax specifies the variables to be operated on, the ranges to form the categories, and their labels. In the first example, a number of variables are simultaneously categorized according to the same rules. In the second, a new categorical variable, incomecat, may be created while retaining the original variable, income.

A CROSS statement in the VPLX syntax provides one means to construct cross-classifications. Real or categorical variables may be crossed by categorical variables. In addition, a crossed variable may be further crossed by other categorical variables, so that cross-classifications may be built up to an arbitrarily high dimension.

The CLASS statement provides a more powerful method of forming cross-classifications. In the absence of BLOCK statements in the CREATE step, all other variables, including crossed variables, will be cross-classified by the class variable. If two or more class variables are defined, the joint cross-classification will be produced.

The scope of CLASS statements may be limited by BLOCK statements. The VPLX specification may begin with global CLASS statements affecting all variables. After the appearance of a BLOCK statement, subsequent CLASS statements have a range of application only within the block of variables, although the same variable may be used as a class variable on more than one block. Thus, with a simple syntax, it is possible to build up a VPLX file as a series of matrices of different dimensions. Information on the dimensions of these cross-classifications is stored as part of the VPLX tally file.

A third means to cross-classify data is through the BY statement, analogous to similar features in SAS or SPSS-X. Use of this approach is encouraged only for extremely large problems in which the equivalent cross-classification cannot be obtained through a CLASS statement because of storage limitations.

Additional syntax in the CREATE step enables the labelling of variables; declaration of missing values; selection of cases, either globally or within a block; and specification of which variables to keep or drop from the output file.

3. The TRANSFORM Step

The VPLX file resulting from a CREATE step is essentially a file of sums, including replicate sums. This information is sufficient for the DISPLAY step to compute variances, covariances, and other statistics for estimates of totals, means, and proportions. Additionally, when the data have been cross-classified by class variables, the DISPLAY step also has the capability to estimate variances, etc., for any of the possible marginal tables. The DISPLAY step is not designed to compute statistics for more complex estimates, such as ratios and other functions of estimated totals, unless these values are included on the VPLX tally file for the full sample and the replicates.

The primary purpose of the TRANSFORM step is to provide the capability to create new variables in the VPLX tally file on the basis of estimated sums. Currently, the user provides one or more FORTRAN subroutines whose function is to compute, on the basis of the values of a set of the current estimates of totals, the values of one or more new variables. VPLX then calls this subroutine once for the total sample and once for each replicate. VPLX performs the necessary bookkeeping to add the variables to the new file. As a short illustration of the syntax:

```
transform  in = a:file1.vpl
    out=a:file2.vpl
user1
old  x1, x2 / class age*sex
old  x3 / class age
derived  ratio1 ratio2 / class age*sex
```

The example presupposes that the VPLX tally file a:file1.vpl contains variables x1, x2, and x3, where x1 and x2 are cross-classified by the class variables age and sex (although by possibly other class variables as well) and x3 is cross-classified by age (and possibly other class variables). The user-supplied FORTRAN subroutine will be passed FORTRAN matrices containing tallies for x1, x2, and x3 with the requested dimensions. New variables, ratio1 and ratio2, both cross-classified by age and sex, are to be created by the subroutine, and VPLX will add them to the outgoing file. If none of the existing blocks have age and sex as its defining class variables, then a new block will be formed with those dimensions, otherwise ratio1 and ratio2 will be placed in an appropriate existing block.

The TRANSFORM step also has capabilities such as adding labels to the file and keeping or dropping variables on the outgoing file.

4. The DISPLAY Step

The DISPLAY step is used to display estimates, estimated standard errors, covariances, correlations, and t-tests for variables on a VPLX file, and, optionally, to write the results to another file for use by other systems. The syntax permits distinctions between estimated means, totals, and proportions, and provides for flexibility in the display with respect to the class variables. For example, if the CREATE step had included three CLASS statements:

```
class  sex  (1/2)  'Males' 'Females'
class  race (1/2/res) 'White' 'Black' 'Other races'
class  age  (16-19/20-29/30-44/45-64/65-High)
```

then

```
display  in = temp$:[r_fay]w2vplx1.vpl
list  totinc earnings / class  total sex
    race(1,2)*sex(0,1,2)
        sex*age
```

would produce displays of the estimated means and standard errors for totinc and earnings for the total sample, by sex, for Whites and for Blacks for both sexes combined and by sex, and for sex by age.

5. Future Enhancements

The existing system has capabilities to produce variances for a number of applications, but further enhancements are envisioned. Briefly, these are:

1. To add capability to the TRANSFORM and DISPLAY steps to input a VPLX tally file of constants along with the primary VPLX tally file giving the replicate estimates. The VPLX CREATE step currently has the ability to create an output file of constants, that is, one for which there are no replicate estimates. Relatively simple modifications to the TRANSFORM and DISPLAY steps should permit this information to be merged with a regular VPLX tally file. This feature would facilitate the process of providing the TRANSFORM step constants to be used in estimation, such as population controls to be used in ratio estimation.
2. To develop a MERGE step to permit the combination of separate VPLX tally files.
3. To add a JACKKNIFE step which would compute the first-order bias corrections described in Fay (1984), a special case of which is the standard jackknife bias correction. This step could be used effectively with the simple or stratified jackknife, or, provided that they satisfy the conditions in Fay (1984), with half-sample or generalized replicates. An option would be added to the DISPLAY step to adjust the variance estimates for the effect of the bias correction.
4. To design a VPLX observation file and a step analogous to the CREATE step to form such files. A VPLX observation file would contain variable names, etc., and data for each observation in the sample universe, including replicate weights. These files would interact with the existing system, for example, the current CREATE step would be expanded to produce an output VPLX tally file from an input VPLX observation file, as an option. Another step would be added to reweight the replicate weights on a VPLX observation file according to factors stored on a VPLX tally file. Such an extension of the system would also enable consideration of procedures for logistic regression and other statistical analyses that cannot be performed from the summary data in a VPLX tally file.
5. To include a set of standard subroutines in the TRANSFORM step, so that transformations of totals involving simple operations, including ratio estimation, could be specified without recourse to user-supplied subroutines.
6. To add to the CREATE step the ability to form cross-product matrices from sets of the variables. Such matrices would then expand the ability of the TRANSFORM step to consider a variety of multivariate methods based on estimated means and cross-products, including linear regression.
7. To create a new version of the software CPLX for the log-linear analysis of categorical data. The new version would accept VPLX tally files as input and feature a user-friendly command language.

The fourth of these merits additional comment. One of the difficulties in producing variances from Census Bureau surveys has been reflect the impact of ratio estimation and other adjustments on the final estimates. Replication offers an answer to this problem, but performing the theoretically required full replication of the estimation for surveys, such as the Current Population Survey (CPS), has proven difficult and resource-intensive. The current version of VPLX has calculated variances from the CPS from replicate weights produced by the Census Bureau's existing variance system for the CPS. When this fourth enhancement has been added, however, it will then be possible to replicate the stages of CPS estimation and produce the final replicate weights within VPLX. Experience with the current version of VPLX suggests that a workstation environment may be adequate for this task, which would remove this burden from the Census Bureau's mainframe computers.

Current plans are to employ VPLX in the analysis of the 1990 Post-Enumeration Survey, which is designed to measure the undercount in the 1990 census. Investigation of numerous other applications of VPLX within the Census Bureau are anticipated.

6. Comparisons With Other Systems

The three other systems mentioned in the first section, SUDAAN, PC-CARP, and WESVAR, have capabilities not yet implemented in VPLX, including linear regression. The objective of duplicating the features of the other systems as quickly as possible has not driven the development of VPLX, even though VPLX may eventually replicate the other systems' capabilities. Rather, the purpose has been to design a general system to exploit the advantages of replication as a variance estimation technique. At the same time, it is hoped that many users may find VPLX as convenient or more convenient to use than other systems for standard analyses.

VPLX is the most portable of the systems, which is especially useful in computational environments with a variety of hardware, such as the environment at the Census Bureau.

Of the three other systems, only SUDAAN, which employs linearization, promises as much generality for complex statistics as VPLX. Replication offers distinct advantages for extremely large problems, favoring VPLX in those cases. For problems of lesser complexity, users with previous FORTRAN experience may prefer to implement their analyses in VPLX, while others may prefer the SUDAAN language.

PC-CARP, which also employs linearization, is the most interactive and friendly of the systems, and it is notable for its treatment of the errors-in-variables model. It is restricted to the IBM PC environment, and it has limited capability to deal with very large problems. Nonetheless, a number of users have carried out rather extensive calculations with it.

WESVAR, which implements replication, is methodologically close to VPLX. WESVAR is implemented within SAS, whereas VPLX is not, with the consequent advantages and disadvantages to these strategies.

The SAS environment has been unfavorable for the development of a PC version of WESVAR, whereas the PC version of VPLX is readily useful for the analyses of surveys on the order of thousands, or sometimes tens of thousands, of cases.

None of the four has the clear advantage over the others, but individual users may do well to consider the suitability of each to specific problems. The simplification of the calculation of design-based variances for complex samples is certain to yield improvements in the manner in which surveys are designed and analyzed.

References

Fay, R. (1984), "Some Properties of Estimators of Variance Based on Replication Methods," in *1984 Proceedings of the Section on Survey Research Methods*, American Statistical Association, Alexandria, VA, 495-500.

______ (1989), "Theory and Application of Replicate Weighting for Variance Calculations," in *1989 Proceedings of the Section on Survey Research Methods*, American Statistical Association, Alexandria, VA, 212-217.

Flyer, P., Rust K., and Morganstein D. (1989), "Complex Survey Variance Estimation and Contingency Table Analysis Using Replication," in *1989 Proceedings of the Section on Survey Research Methods*, American Statistical Association, Alexandria, VA, 110-119.

Fuller, W.A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H.J. (1986), *PC-CARP*, Statistical Laboratory, Iowa State University, Ames, IO.

Shah, B. V. (1989), "Compiler-Interpreter for Survey Data Analysis Language," in *1989 Proceedings of the Section on Survey Research Methods*, American Statistical Association, Alexandria, VA, 103-109.