SAS COMMUNICATIONS

INSTITUTE OF STATISTICS, NORTH CAROLINA STATE UNIVERSITY, RALEIGH, NORTH CAROLINA #1 DECEMBER,1973

The first versions of the Statistical Analysis System remained pretty much within the domain of North Carolina State University. In the sixties, a typical SAS user worked at N.C.S.U. and had only to call or walk across campus to discuss the system with its developers. The user had full opportunity to find out about the then-current capabilities of SAS, the plans that Messrs. Barr and Goodnight had for the system, some of the out-of-the-ordinary tasks to which SAS could be put, and so forth.

Expanded and refined into what we believe to be the most powerful, flexible system extant for data management and analysis, SAS now boasts users at more than 100 installations across the United States and overseas. The informal network of communication is no longer adequate for serving the enlarged community of SAS users. We of the SAS project group are therefore launching SAS Communications. This newsletter will, we hope, involve users in the continuing development of SAS and aid them in their use of the current version. SAS Communications will be published at irregular intervals in an informal format, for we value the contributions that easy, spontaneous interaction between users and developers have already made and assuredly will make to the system.

The newsletter will highlight SAS features that are commonly overlooked. It will outline the activities of the SAS project group and the goals that the group intends to meet. Application notes -- special ways of using SAS -- will be featured too, as will notes on the algorithms embodied in SAS. We shall describe additions to the Supplementary Procedures Library, the collection of special-purpose procedures written by the staff and by users. Profiles of the people who make up the project group will be included. Other publications about SAS will be listed. We also plan to initiate a users' forum, wherein a user can question or comment to the staff and other users. Finally, every few issues, we will include an up-to-date index of the articles in SAS Communications.

We urge you, the users, to contribute to SAS Communications; it is only with your help that SAS can grow to meet the needs of the community it serves.

Jolayne Service

for the

SAS Project Group:

A. J. Barr, Systems
J. H. Goodnight, Procedures and Administration
J. Service, Documentation
C. G. Perkins, Assistant for Systems
J. Sall, Assistant for Procedures
B. Reeves, Administrative Assistant
H. J. Kirk, Consulting
S. L. Biggs, Consulting

SAS GROWS

A. J. Barr is currently rewriting the "systems" portion of SAS, that part which builds data sets and processes SAS statements. Multiple aims characterize his activity.

First, Mr. Barr is making it easier to expand the capabilities of the system. He is also making the internal workings of SAS clearer to other systems programmers, who might wish to make special-purpose modifications of SAS.

The rewritten system will handle data sets better, especially partitioned data sets. Less Job Control Language will have to be used to store and retrieve SAS data sets. Data sets on magnetic tape will acquire some of the flexibility now associated with disk data sets.

An installation will be able to "autobatch" SAS jobs; that is, to run series of small SAS jobs end-to-end very efficiently. Users familiar with autobatched versions of the programming language compilers WATFIV and PL/C will recnogize the advantages of autobatching. Also, SAS will be easier to invoke interactively under the Time Sharing Option (TSO) of the IBM operating system.

The new SAS will include report-writing capabilities. A user will be able to make SAS produce a customized, possibly annotated report, easily readable by people unfamiliar with computers.

Finally, Mr. Barr is making SAS statements more consistent and flexible. For example, where a user would now write

Q2 Q3 Q4 Q5 Q6 Q7

he will be able simply to write

Q2-Q7

Other members of the SAS project group are developing and modifying SAS Procedures. J. H. Goodnight is preparing a procedure to perform non-linear least-squares curve-fitting. Expanded capabilities are being added to DISCRIM, the discriminant analysis procedure. Carroll Perkins is working on a procedure for producing histograms.

NEW SPONSORS FOR SAS

The Directors of the Southern Regional Agricultural Experiment Stations approved in June, 1973, the project titled, "The Statistical Analysis System: Its Development and Maintenance." This project provides a new source of funding for SAS, insuring its growth for the next five years. Participating in the project are the State Agricultural Experiment Stations of Alabama (at Auburn University), Florida (at the University of Florida), Georgia (at the University of Georgia), Kentucky (at the University of Kentucky), Louisiana (at Louisiana State University), North Carolina (at North Carolina State University), Oklahoma (at Oklahoma State University), South Carolina (at Clemson University), Tennessee (at the University of Tennessee), Texas (at Texas A & M University) and Virginia (at Virginia Polytechnic Institute and State University), and the Agricultural Research Services of the United States Department of Agriculture. A Technical Committee was formed, which includes a representative from each participating organization and has Clyde Y. Cramer of Virginia as chairman, Robert D. Morrision of Oklahoma as vice-chairman, and Robert J. Monroe of North Carolina as secretary.

The subcommittee on price structure has set yearly fees for SAS installations Details about the charges and the services to which those who pay are entitled are available from J. H. Goodnight, Institute of Statistics, Raleigh, N. C. 27607.

The initial development of SAS was supported in part by grant RR-00011 from the Biotechnology Resources Branch, Division of Research Resources, NIH.

SUPPLEMENTARY PROCEDURES

This fall, we released the SAS supplementary procedure library, a collection of special-purpose routines. Some were contributed by SAS users; all are maintained by the project group. For those who have not yet seen the supplementary procedures guide, we list below the procedures presently documented.

PRTPCH: For punching or printing SAS data sets in a format determined by a user-written FORTRAN IV specification.

INBREED: For calculating inbreeding or covariance coefficients.

HARVEY: A SAS implementation of Dr. Walter Harvey's Least Squares and Maximum Likelihood General Purpose Program.

RENAME: For changing the names of variables in a SAS data set.

QUESTN: A quick procedure for producing frequency and contingency tables for data from questionnaires.

EXPLODE: For printing characters over one inch high.

PROBIT: For probit analysis of biological assay data.

STANDARD: For standardizing the values of numeric variables to a given mean and standard deviation.

A NOTE ON PRECISION

Some users have asked how precise are the results SAS produces. The answer, of course, depends somewhat on the data submitted to SAS and on the procedures used. We note, however, that all numeric data is stored and manipulated in double precision--about 16 decimal digits.

In 1967, James W. Longley published in the Journal of the American Statistical Association an article entitled "An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User." It presented some data for which regression analyses were to be performed and the results of using several then-popular computer programs for doing the calculations. He found that the error in the calculation of one typical regression coefficient ranged from .03% (in the most accurate program considered) to 375%.

This year, Terry Seaks of the University of North Carolina at Greensboro used SAS and several other newer programs for analyzing the Longley data. SAS virtually tied with another system for lowest error rate. Its answer for the regression coefficient mentioned was accurate to eight digits.

SAS PUBLICATIONS

Service, Jolayne. A User's Guide to the Statistical Analysis System. 1972.

Perkins, Carroll Gray. A Guide to the Supplementary Procedure Library for the Statistical Analysis System. 1973.

Barr, Anthony James, and James Howard Goodnight. SAS Programmer's Guide. 1972.

The user's guide, published by and available from the Student Supply Stores, North Carolina State University, Raleigh, North Carolina 27607, tells how to use SAS. The supplementary procedures guide describes special-purpose SAS procedures. Programmers who want to implement their own procedures under the umbrella of SAS will need to consult the programmer's guide. The latter two guides are available from the SAS project group, Institute of Statistics, North Carolina State University, Raleigh, North Carolina 27607.

ADDENDA TO THE USER'S GUIDE

A few facilities of the current version of SAS were omitted from the User's Guide. We circulated to all installations a memorandum detailing the omissions; here, we note those items for users who missed the memorandum.

Changes to the INPUT statement description

1. An INPUT statement can introduce packed decimal data into SAS. If the values of the variable AGE were in packed decimal form in positions 3 through 6 of the input records, one would write

   INPUT AGE PD 3-6;

2. The maximum number of characters per observation is 32000.

3. The number of cards per observation is not restricted.

4. Values of character variables may be no more than 80 characters long.

Changes to the descriptions of program statements

1. A KEEP statement of the form

   KEEP variable_1 < variable_2 ... variable_n>;

   can be used to retain only the listed variables in the data set being built.

2. In building a SAS data set, a numeric variable named ERRORSW is generated automatically but not included in the observations actually added to the data set. ERRORSW is set to zero before an observation is constructed; it is set to one if an invalid data element or an arithmetic error (like an invalid function argument) is encountered in forming that observation.

3. A Boolean operator YES can be used in place of NOT NO.

4. Two comparision operators can surround a quantity. For example,

   AGE > 12 AND AGE < 20

   can be abbreviated

   12 < AGE < 20

5. A STOP statment, written

   STOP;

   forces SAS to ignore the observation being processed and to cease altogether building the data set.

6. When an observation is to be formed from data on several records, one can use a LOSTCARD statement to insure that all "lost" records are detected. The statement terminates the formation of an observation and tells the INPUT statement to return to the records it just read, ignoring the first record and drawing in the appropriate number of records beginning with the second record previously encountered. For example, one might write

```
INPUT ID 1-3 S1 5-10 IDCHECK #2 1-3 S2 #2 5-10;
IF ID ¬=IDCHECK THEN ERROR ID IDCHECK;
IF ID ¬=IDCHECK THEN LOSTCARD;
```

Other Changes

1. Unless DUMMYB is specified in the MODEL statement in the REGR procedure, the rows and columns associated with dummy variables in the XPX, SWEPT, and INVERSE matrices will not be printed.

2. A new statement for use under TSO,

   RUN;

   forces SAS to execute all statements already entered but not executed. Additional SAS statements can be entered after the results of the execution are printed.

APPLICATION NOTE: RANDOMLY SAMPLING OBSERVATIONS IN A SAS DATA SET

Occasionally when a statistician must consider a very large data base, he finds it useful to investigate just a sample of the observations recorded. SAS's ability to generate pseudo-random numbers makes it easy to use the system to isolate a random sample.

To take a simple random sample of the observations in a data collection, the SAS user could create a variable which associates a pseudo-random number with each observation. The data set would be sorted according to those random numbers, thus effectively putting the observations in random order. Wanting a sample of size N, the user would finally create a subset of the sorted data set, the subset including only the first N of the randomly arranged observations.

Shown below is an example of selecting 50 observations randomly. Note that the use of the counting variable N is discussed on page 30 of the user's guide, and the UNIFORM function is described on page 246. We are supposing that the data collection BASE is stored on magnetic tape.

```
DATA POPULACE;
    INPUT DD=BASE ID 1-5 STATE $ 7-8 C 10-19 W $ 72-80;
    RANDOM=UNIFORM(43761);
PROC SORT;
    BY RANDOM;
DATA SAMPLE;
    SET POPULACE;
    N=N+1;
    IF N < = 50;
PROC PRINT;
    .
    .
    .
```

Taking a stratified random sample is more complicated. If we wish, say, to sample randomly 10% of the units of each state represented in BASE, we have to count how many observations are associated with each value of STATE. We show a sequence of SAS statements, annotated with COMMENT statements, that would accomplish the stratified random sampling. We recommend that a reader consult carefully the user's guide section on the MERGE statement, pages 47ff, if he desires a thorough understanding of the procedure.

```
        COMMENT
        WE BEGIN AS WE DID FOR SIMPLE RANDOM SAMPLING;
DATA POPULACE;
    INPUT DD=BASE ID 1-5 STATE $ 7-8 C 10-19 W $ 72-80;
    RANDOM=UNIFORM(34671);
        COMMENT
        AS WE SHALL TAKE 1/10 OF THE OBSERVATIONS OF EACH STATE,
        WE SORT "POPULACE" FIRST BY STATE.  THEN THE OBSERVATIONS
        WITHIN A STATE ARE PUT INTO RANDOM ORDER ACCORDING TO THE
        VALUES OF "RANDOM";
PROC SORT;
    BY STATE RANDOM;
        COMMENT
        NEXT, WE CREATE A PHONY DATA SET, CONTAINING NO OBSERVATIONS
        AT ALL, SO THAT WE CAN MERGE IT WITH "POPULACE" AND USE
        THE "LASTBY" FEATURES OF SAS'S MERGING OPERATION;
DATA PHONY;
    SET;
    STOP;
        COMMENT
        NOW, WE CREATE A VARIABLE "S_TOTAL" TO COUNT THE OBSERVATIONS
        WITHIN A STATE.  WHEN THE "LASTBY" VALUE TELLS US THAT WE HAVE
        REACHED THE LAST OBSERVATION IN A STATE, WE RECORD ON THE DATA
        SET "COUNTS" THE VALUE OF "S_TOTAL", THE NUMBER OF OBSERVATIONS
        IN THAT STATE, AND RESET "S_TOTAL" TO ZERO.  THUS "S_TOTAL"
        CAN BE USED TO COUNT THE OBSERVATIONS IN THE SUCCEEDING STATE.
        THE DATA SET "COUNTS" WILL HAVE ONE OBSERVATION PER STATE;
DATA COUNTS;
    MERGE PHONY POPULACE;
       BY STATE;
    S_TOTAL=S_TOTAL+1;
    IF LASTBY THEN OUTPUT;
    IF LASTBY THEN S_TOTAL=0;
        COMMENT
        FINALLY, WE USE THE MERGE STATEMENT AGAIN TO PROCEED THROUGH "POPULACE" STATE
        BY STATE.  WE PUT INTO "SAMPLE" AT LEAST ONE AND NO MORE THAN 10 PERCENT OF
        THE OBSERVATIONS IN EACH STATE;
DATA SAMPLE;
    MERGE COUNTS POPULACE;
       BY STATE;
    N=N+1;
    IF N=1 OR N<=.1*S_TOTAL THEN OUTPUT;
    IF LASTBY THEN N=0;
PROC PRINT;
```

.

.

.

If one wished further stratification of the sampling -- by values of STATE and C, for instance -- one need only substitute

```
    BY STATE C
```

everywhere

```
    BY STATE
```

appears above.

ANALYSIS OF VARIANCE IN SAS

The December, 1973, issue of the Journal of the American Statistical Association includes an article by Ivor Francis called "A Comparison of Several Analysis of Variance Programs." Mr. Francis describes a misuse of SAS's ANOVA procedure. We list below a few comments on the article.

1. The current documentation for ANOVA describes in detail the method used for calculating sums of squares (User's Guide, page 138 and pages 152-153). The method is considered appropriate for balanced factorial designs. The documentation does not claim it accurate for unbalanced factorial designs, such as the one treated in the article.* Mr. Francis, it should be noted, cites out-of-date SAS documentation in his criticism.

2. Mr. Francis claimed that no messages in the printout of the program suggested the cause of or solution to the problem noted. The ANOVA procedure, when confronted with an unbalanced design, always prints the message,
"WARNING: THE ANALYSIS MAY BE INCORRECT DUE TO NO DATA FOR SOME CELLS. CHECK YOUR DEGREES OF FREEDOM AND CELL FREQUENCIES TO VERIFY THE CORRECTNESS OF THE ANALYSIS."

3. The table of contents of the User's Guide, as well as the REGR description, indicates that the REGR procedure is appropriate for analysis of variance, without regard to the balance of the design. Towards the end of his article, Mr. Francis acknowledges the correctness of the analysis produced by REGR. It should be noted that the claims made for the versatility of the BMD program apply equally well to REGR, and, as Mr. Francis wrote, REGR is "very simple to use in this fasion."

4. Mr. Francis ascribed the higher cost of running REGR to the presence of the other SAS procedures. The overhead is due rather to the sophisticated data management facilities of SAS. Also, many SAS jobs may be run in a memory region of 100K.

ANOVA is an efficient procedure for processing balanced designs and hence has its place in SAS. Like any computer program, it is open to abuse by careless users. Though we would have wished that Mr. Francis had emphasized much earlier in his article the appropriateness and virtues of SAS-REGR, we hope that his article will be of benefit in forestalling the misuse of SAS.

---

*For extremely large data sets of very slight imbalance, ANOVA's approximate results may be the only ones feasible to obtain.

We ask that the SAS installations copy and distribute this issue of SAS Communications. Users wishing to receive further issues are encouraged to send us their names and addresses.

SAS COMMUNICATIONS

INSTITUTE OF STATISTICS, NORTH CAROLINA STATE UNIVERSITY, RALEIGH, NORTH CAROLINA #2 MAY, 1974

SAS SUBSYSTEM FOR PESTICIDE LICENSING

Concerned about injuries and deaths due to misuse of pesticides, the North Carolina General Assembly enacted a law requiring vendors and appliers of pesticides to be licensed. Herbert J. Kirk, in cooperation with the Pest Control Division of the North Carolina Department of Agriculture, has developed a SAS-based system for handling most of the information-processing aspects of the licensing function.

Mr. Kirk utilizes SAS's data management facilities heavily. His subsystem grades the tests that prospective licensees must take (there are tests to correspond to each area of specialization a pesticide applier might choose). Records of licensees are maintained as SAS data sets and updated with the MERGE facility. A custom-designed SAS procedure prints the licenses. The sorting and printing capabilities of SAS are used to furnish inspectors lists of licensees by county. Mr. Kirk claims that the integration of the necessary special-purpose routines into the framework of SAS not only saved program development time, but also reduced the time and staff needed to perform the licensing function day-to-day.

Other SAS subsystems have been created and used with equal success. A user interested in developing such a subsystem should consult the SAS Programmer's Guide.

USERS' FORUM

To The Editor:

I'd like to report a "bug" in the User's Guide write-up for the RANK procedure. It concerns the output for tied data when the FRACTION option is used.

When the FRACTION option is not used, and several observations have the same value for the variable, those values are assigned an average rank, as stated in the User's Guide. However, when the FRACTION option is used, the value returned is the fraction that pertains to the highest rank for which they are tied. In the example shown on Page 72 of the User's Guide, the two tied values would receive fractional ranks of $3/8 = 0.38$ rather than $2.5/8 = 0.31$.

This method of handling ties is reasonable if one is using the procedure to generate an empirical cumulative distribution function.

Sincerely yours,

Harvey J. Gold

√ We are grateful to Dr. Gold for pointing out both our mistake and an alternative use for the RANK procedure. Please make the appropriate changes to Page 73 of your copy of the User's Guide.

PROFILE: JAMES HOWARD GOODNIGHT

Jim Goodnight, co-developer of SAS, took a statistical programming course while an undergraduate at N. C. State University. In his work with SAS today, he is still exulting in the rewards of statistical programming that he discovered in that course. The satisfactions are direct, he claims. "You can see results, see programs grow and develop."

A native of North Carolina, Dr. Goodnight focused his interest on quantitative studies early. All his formal higher education was conducted at N. C. State: he was awarded a Bachelor's degree in Applied Mathematics in 1965, a Master of Experimental Statistics degree in 1968, and a Ph.D. in Statistics in 1972. Between college and graduate school, he worked in G. E.'s Apollo Support Department, first as an engineering programmer and later on information retrieval systems. Returning to N. C. State, he accepted a graduate research assistantship, then a faculty position as Assistant Statistician; he is now a Research Associate. He has been working on SAS since 1967 and doing additional consulting for corporate and university researchers.

Dr. Goodnight's area of greatest competence within statistics is regression and least-squares methodology. He likes the matrix orientation of that field, for he finds the essential simplicity of matrices appealing. In programming, he admits to dwelling on "the aesthetics" of the printed results. "I like nice output," he smiles.

Looking at the future of SAS, Dr. Goodnight states that one of his main concerns is the "caveat emptor" approach that developers of statistical systems -- even, to some extent, the developers of SAS -- have traditionally taken. He feels that statistical programmers should be looking into what they can do to protect users. "Right now, we can diagnose syntactical errors," he states. "We can't diagnose errors of application." Though there has not been enough research done in this area, he feels, some implementation is possible. For example, a lack-of-fit test for models might be built into a procedure and the analysis truncated if the test be significant. Tests of normality and homogeneity might be incorporated into routines whose validity depends on those assumptions. Dr. Goodnight questions both the availability of such screening techniques and the economy of including them, but he claims, "We must start taking these things into consideration."

SAS Communications is published by the SAS Project Group:

A. J. Barr, Systems
J. H. Goodnight, Procedures and Administration
J. Service, Documentation and Editing
C. G. Perkins, Assistant for Systems
J. Sall, Assistant for Procedures
H. J. Kirk, Consulting
S. L. Biggs, Consulting
J. Massengill, Secretary

Institute of Statistics
North Carolina State University
Raleigh, North Carolina 27607

APPLICATION NOTE: SIMULATING EXPERIMENTS FOR INSTRUCTIONAL PURPOSES

The past few years have seen a sharp rise in interest in using computers for instructional purposes. In statistical methods courses, student use of data analysis programs has helped shift the courses' emphasis from calculation of numeric measures (the "cookbook" approach) to interpretation of data. Beyond this, computers have made it feasible to illustrate principles -- like laws of large numbers, sampling properties, and Central Limit Theorems -- that require rather sophisticated mathematics to prove. S. G. Carmer and F. B. Cady, among others, have propounded yet another approach;

> In order to eliminate the often tedious search for 'real-life' research data meeting the specifications for a given teaching situation, a logical approach is to have a computer generate sets of data according to a model whose structure and parameters (including magnitude and distribution of random error) are controlled by the teacher. In a given problem the selected parameter values could be purely hypothetical in nature, and thus give rise to highly artificial data useful for illustrative purposes only. More realistic data could be generated by basing parameter values on research published in a subject matter (e.g., agriculture, biology, or engineering) journal or report where statistical summaries of real data, but not the data, appear.[1]

Below we show one way of generating in SAS individualized data sets suitable for simple linear regression or correlation analysis.

```
DATA CLASS; INPUT NAME $ 1-15 BETA0 16-20 BETA1 21-25 SIGMA 26-30;
     COMMENT      BE SURE THAT ALL THE NAMES DIFFER IN AT LEAST ONE
                  OF THE FIRST 8 CHARACTERS;
N = 0;
COUNT:  N = N + 1;
        X = 100 + 15*NORMAL(156983);
        Y = BETA0 + BETA1*X + SIGMA*NORMAL(23387);
        OUTPUT;
        IF N < 40 THEN GO TO COUNT;
CARDS;
ALLEN           0    .1   1
ANDERS          0    .1   10
BELL            0    10   1
BRYANT          0    10   10
BUFORD          10   .1   1
  .
  .
  .
ZIMMERMAN       5    1    10
```

1. S. G. Carmer and F. B. Cady, "Computerized Data Generation for Teaching Statistics," American Statistician, Vol. 23, No. 5 (1969), pp. 33 ff.

```
PROC PRINT PAGE; VAR X Y; BY NAME;
     COMMENT       STUDENTS' DATA;
PROC REGR C;  BY NAME;
     COMMENT       ANSWERS FOR INSTRUCTOR;
MODEL Y = X/P;
```

The following example illustrates what samples from an exponentially distributed population can be expected to look like and how well they conform to the Central Limit Theorems. We generate 50 samples of 80 observations each, produce a histogram for each sample, and produce a histogram of the sample means.

```
DATA EXP;
INPUT MU 1-10;
     COMMENT       MU IS THE POPULATION MEAN.  WE USE THE INVERSE OF THE
                   EXPONENTIAL CUMULATIVE DISTRIBUTION FUNCTION TO OBTAIN
                   EXPONENTIAL PSEUDO-RANDOM VARIATES FROM THE UNIFORM
                   PSEUDO-RANDOM VARIATES PROVIDED BY SAS;
SAMPLE = 0;
SET:  SAMPLE = SAMPLE + 1;
      N = 0;
      OBS:  N = N + 1;
            X = -MU*LOG(1-UNIFORM(33933));
            IF N < 80 THEN GO TO OBS;
      IF SAMPLE < 50 THEN GO TO SET;
CARDS;
1
PROC HIST; BY SAMPLE; VAR X;
     COMMENT       DON'T FORGET TO INCLUDE THE JOB CONTROL LANGUAGE
                   STATEMENT THAT MAKES THE SUPPLEMENTARY LIBRARY
                   PROCEDURES ACCESSIBLE;
PROC MEANS NOPRINT OUT=CLT; BY SAMPLE;
PROC HIST; VAR X;
     COMMENT       THIS LAST HISTOGRAM SHOWS THE EMPIRICAL SAMPLING
                   DISTRIBUTION OF THE SAMPLE MEANS;
```

SAS PUBLICATIONS

Service, Jolayne. A User's Guide to the Statistical Analysis System. 1972.

Perkins, Carroll Gray. A Guide to the Supplementary Procedure Library for the Statistical Analysis System. 1973.

Barr, Anthony James, and James Howard Goodnight. SAS Programmer's Guide. 1972.

The User's Guide, published by and available from the Student Supply Stores, North Carolina State University, Raleigh, North Carolina 27607, tells how to use SAS. (Because of a recently enacted statute, the Student Supply Stores will no longer be allowed to sell the User's Guide to anyone off campus other than university book stores. The guide will not be reprinted as it is being revised at this time and the new edition will be out in September. At the time the new edition is complete, new arrangements will be made for publication and distribution, and all SAS users will be notified.) The Supplementary Procedures Guide describes special-purpose SAS procedures. Programmers who want to implement their own procedures under the umbrella of SAS will need to consult the Programmer's Guide. The latter two guides are available from the SAS Project Group.

OPSCAN

A supplementary procedure called OPSCAN has been created to construct a SAS data set from the information on a magnetic tape produced by an OPSCAN/100 optical scanning reader. Developed by Carroll Perkins of the SAS Project Group, OPSCAN's first function was to facilitate processing of course-and-instructor-evaluation questionnaires. The procedure can handle information from the Optical Scanning Corporation's Standard Answer Form A; it also processes the forms of N. C. State's Student Government Faculty Evaluation.

Users of OPSCAN/100 equipment are invited to request further details from Mr. Perkins.

APPLICATION NOTE: OBTAINING REGRESSION COEFFICIENTS FOR FURTHER COMPUTATION

Several users have asked how to obtain estimated regression coefficients ("b-values") from the REGR procedure for use in the same run. Since REGR ignores observations for which relevant values are missing, one can insert "dummy" observations (with missing dependent variable values) into the data set to be analyzed. The OUTPUT statement can then pick out the b-values. In the example following, the first three observations are included in INFO for such a purpose. The data set FINAL will contain the b-values in every genuine observation along with X1, X2, Y and YHAT. A more realistic application would include using regression coefficients to generate "observations" for a plot of a regression line or surface. Such generation could be done by modifying slightly the program statements shown and adding statements like those shown in the User's Guide examples for the PLOT procedure.

```
DATA INFO;
INPUT  X1  1-5  X2  6-10  Y  11-15;
CARDS;
0    0              }  for this observation, the predicted Y will equal B0
1    0              }  for this observation, the predicted Y will equal B0 + B1
0    1              }  for this observation, the predicted Y will equal B0 + B2
1.46 2.93 20.9      }
1.98 2.60 24.8      }  this is
  .                 }  the data to
  .                 }  be analyzed
  .                 }
PROC REGR;
MODEL Y = X1 X2;
OUTPUT OUT=TEMP PREDICTED YHAT;
DATA FINAL; SET TEMP;
N = N + 1;
IF N = 1 THEN B0 = YHAT;
IF N = 2 THEN B1 = YHAT - B0;
IF N = 3 THEN B2 = YHAT - B0;
IF N > 3;
```

NEW SUPPLEMENTARY PROCEDURES

Two procedures have been added to the Supplementary Procedure Library, one for testing Normality of numeric data and the other for printing histograms. Installations having service agreements are receiving these additions. We are including in this newsletter the documentation for both procedures so that users will not have to reorder the Supplementary Procedures Guide to find out about the additions.

The KSLTEST Procedure

The KSLTEST procedure computes simple (univariate) statistics for testing normality of numeric variables in the input data set. The KSLTEST procedure is contributed to the Supplementary Procedure Library by Dr. Harvey J. Gold of the Biomathematics Program of the N. C. State University Department of Statistics.

The KSLTEST procedure first computes the sample mean and standard deviation for each variable. Next, the Kolmogorov statistic is computed as $D = MAX \mid S(X) - F(X) \mid$, where S is the sample cumulative distribution function and F is the Normal distribution function with the sample mean and variance. Then, the Kolmogorov statistic is compared to the Lilliefors table of critical values if N is not greater than 30 (reference 1). If N is greater than 30, then the Kolmogorov statistic is compared against asymptotic values.

The Fisher G-statistics are computed as measures of sample skewness and kurtosis (reference 2). These statistics may be used to test normality, but they have no power against some alternatives since they are based on the unbiased estimates of the first four cumulants. Also, a minimum of four observations from the input data set are needed to calculate these statistics.

OUTPUT

For each numeric variable, a table is output showing the number of observations, value of the Kolmogorov-Smirnov statistic, the Lilliefors significance level, the Fisher G-statistics, the standard errors of the G-statistics, the significance levels of the G-statistics, and the mean and standard deviation of the variable.

A minimum of four observations is needed to produce any statistics.

THE PROC KSLTEST STATEMENT

PROC KSLTEST <DATA=data_set_name>;

PROCEDURE INFORMATION STATEMENTS

The VARIABLES or DROP statement can be used to restrict the procedure to specific numeric variables.

The BY statement may be used to force KSLTEST to calculate test statistics for groups of observations. If a BY statement is included, the data set must already be sorted according to the variables in the BY statement.

TREATMENT OF MISSING VALUES

Missing values are omitted from the calculations.

EXAMPLE

```
DATA; INPUT X1 1-2 X2 4-6 1; CARDS;
19 371
93 719
72 391
28 829
45 341
65 281
18 371
 5 328
52 731
92 382
82 372
28 381
60 145
31  48
 3 291
 1 503
31 436
11 533
59 339
PROC KSLTEST;
```

REFERENCES

1. H. W. Lilliefors, Table of Critical Values from the Journal of the American Statistical Association, Volume 62, Page 399.

2. R. A. Fisher, Statistical Methods for Research Workers, Edition 13, Chapter 3.

The HIST Procedure

The HIST procedure will print one or more percentage frequency histograms. A histogram consists of vertical bars with class values printed below them, or horizontal bars with class values printed to the left. The percentage frequencies are printed perpendicular to the class values for all histograms. Optionally, the user may specify that a histogram's classes be broken down into subclassifications according to the value of one or more additional variables. Unless a VARIABLES statement is present, a histogram will be produced for every variable not appearing in a CLASSES, WEIGHT, BY, or DROP statement.

OUTPUT

The histogram for each variable is printed in the format that will fit best on the printer being used. If possible, a vertical histogram is printed; otherwise, a horizontal histogram is printed. For each numeric variable, the total frequency of all values, frequency of missing values, sum, arithmetic mean, uncorrected sum of squares, corrected sum of squares, variance, standard deviation, coefficient of variation, and standard error of the mean will also be printed.

The classes for a character variable will be the distinct character values of that variable.

The classes for a numeric variable will be either distinct values of that variable or intervals depending on the number of distinct values in that variable. If the maximum number of classes for a numeric variable is not specified by the user, the HIST procedure assumes that 10 or fewer classes should be used. All numeric variables with more distinct values than the maximum allowable number of classes will be subject to grouping. If the number of classes for this type of numeric variable is not specified by the user, the HIST procedure assumes that 1+FLOOR(3.3*LOG10(n)) intervals should be used as classes for the variable, where n is the total number of non-missing data elements in the variable.

A unique feature of the HIST procedure is the ability to sub-classify values within classes. The subclassifications are denoted by printing the leftmost character of a character subclassification value or the leftmost digit of a numeric subclassification value to form the bars representing each class. Scaling remains the same for all subclassifications, thus allowing within-class comparisons to be made.

THE PROCEDURE HIST STATEMENT

```
                    L=          W=          N=
PROC HIST <LOWMIDPT=__> <WIDTH=__> <NINT=__>

                    D=
          <DISCRETE=__> <DATA=data_set_name>;
```

The parameters for the HIST procedure pertain only to numeric variables. In the specifications below, a numeric variable whose values are to be grouped into intervals is called "continuous".

Parameters

LOWMIDPT=__ / L=__ — The LOWMIDPT parameter, which may also be written L, specifies the midpoint for the lowest interval for continuous numeric variables.

WIDTH=__ / W=__ — The WIDTH parameter, which may also be written W, specifies the width of the intervals for continuous numeric variables.

NINT=__ / N=__ — The NINT parameter, which may also be written N, specifies the number of intervals for continuous numeric variables.

DISCRETE=__ / D=__ — The DISCRETE parameter, which may also be written D, specifies the maximum number of distinct values a numeric variable may have before being considered continuous.

PROCEDURE INFORMATION STATEMENTS

WEIGHT Statement

The WEIGHT statement is of the form

```
WEIGHT var_1 <var_2 ... var_n>;
```

The WEIGHT statement is used to assign a predetermined weight to an observation, in effect causing the observation to appear more than once in the input data. If only one variable appears in the WEIGHT statement, its values will serve as weights for all the other variables. If more that one variable appears in the WEIGHT statement, a VARIABLES statement must appear also; these two statements must contain the same number of variables. The values of the first variable in the WEIGHT statement will be taken as weights for the first variable in the VARIABLES statement; the values of the second variable in the WEIGHT statement will be weights for the second variable in the VARIABLES statement, etc. All weighting variables must be numeric, and only the integer part of a weighting data element is used.

CLASSES Statement

The CLASSES statement is used to specify variables whose values will form subclassifications within each class. As with the WEIGHT statement, one subclassification variable may be assigned for all variables, or a separate subclassification variable may be assigned for each variable in the VARIABLES statement. The rules for specifying subclassification are directly analogous to the rules given above for specifying weighting variables. Subclassification variables may be numeric or character. Only the first character of a character CLASSES variable or the first digit of a numeric CLASSES variable is printed, even though the complete value of the variable is used in performing the subclassification operation.

TREATMENT OF MISSING VALUES

Missing values are a unique class (or unique subclassification) in each histogram.

EXAMPLE

In the following example, the HIST procedure is invoked three times.

```
DATA; INPUT GRADE $ 1 N 3-4 Y 6-8 1;
CARDS;
D 18  86
F 16  96
E 19  54
A 17  61
B 15  75
C 17  75
B 16  76
D 20  87
A 18  67
E 19  48
C 17  72
F 17  66
A 15  57
C 18  79
D 16  71
F 19  46
E 21  52
B 15  74
C 18  83
E 18  53
F 19  81
B 16  93
A 22  78
D 14  65
PROC HIST;
PROC HIST L=4 W=.5 N=14; VAR Y;
PROC HIST; VAR Y; WEIGHT N; CLASS GRADE;
```

SAS COMMUNICATIONS

INSTITUTE OF STATISTICS, NORTH CAROLINA STATE UNIVERSITY, RALEIGH, NORTH CAROLINA #3 DECEMBER, 197

SAS USERS DESCRIBE APPLICATIONS TO COMPUTER SYSTEM MANAGEMENT

Two papers were recently presented that describe using SAS to make understandable the computer performance data generated by IBM's System Management Facilities (SMF). In SHARE's Computer Measurement and Evaluation Project Session (B3Ø1; August 28, 1974), H. W. Barry Merrill of State Farm Mutual Auto Insurance Company, Inc., Bloomington, Illinois, discussed "Statistical Analysis of SMF Performance Data." At GUIDE 39, November 6, 1974, Session OPS-1, Steve Cullen, also of State Farm Mutual, and Soli Dastur of Procter and Gamble Company, Cincinnati, Ohio, jointly presented "Statistical Analysis of SMF Data."

Both papers detail how SAS is applied to SMF data. We quote the introduction to the paper of Messrs. Cullen and Dastur:

> SMF, despite its many shortcomings and omissions, is without a doubt the best source of performance data about a computer system because the voluminous amount of data is so easy to collect. But at a large installation like State Farm Mutual Automobile Insurance Company, the volume of data collected can be overwhelming. (Two 155s, a 165, and a 168 create in excess of 700,000 records weekly describing the processing of about 10,000 batch jobs and almost 1,800 TSO sessions.) Such a volume of data is impressive but of little value unless a means of reducing it to a reportable size and format is available. Our early attempts consisted of writing programs to reduce SMF records by type. The result was a group of "homegrown" programs which produced large amounts of paper. Fortunately, in 1972, we discovered a better way -- the Statistical Analysis System.... SAS gave us the facility to create rapidly (in 5-30 minutes of coding) the JCL and control statements to produce almost any SMF-based report. This paper is a discussion of our use of SAS to reduce SMF data into usable reports, as well as to analyze them statistically,... correlating various system parameters. Such usages have resulted in making our computer systems more effective and efficient.

Mr. Merrill's paper concludes, "SAS has proven itself as an extremely efficient and useful technique for analysis of this information, turning data into intuition."

We of the SAS Project Group are grateful to Messrs. Merrill, Cullen, and Dastur for communicating so widely and effectively their experiences with applying SAS to computer management, and we hope that other SAS installations may take advantage of the examples of SAS usage given in the bodies of the two papers.

PROFILE: A. J. BARR

Anthony James Barr, who has led the development of SAS since its inception, remains a most enthusiastic proponent of the further development and improvement of SAS.

Reared in New Jersey, Mr. Barr attended North Carolina State University and earned a B.S. degree in Applied Physics with honors. He was awarded a National Science Foundation fellowship and continued at NCSU, earning a M.S. degree in Physics. He began programming while working on his Master's degree, and computers quickly became his main professional interest. His first programming experience was the development of an analysis of variance program somewhat like the current ANOVA procedure. From 1964 to 1966, Mr. Barr worked for IBM, developing a military command and control system.

Returning to NCSU in 1966, primarily work on analysis of variance procedures for the Department of Statistics, he began compiling a record of outstandingly varied achievements. Working with the NCSU School of Forest Resources, he co-invented in 1972 the Yield-O-Matic (patent pending), a mini-computer-based, real-time system for scanning and marking boards so that the cutting of dimensioned parts is optimal. This achievement prompted the National Association of Furniture Manufacturers to give him a Special Challenger Award for his outstanding contribution to the industry. In the computer systems area, Mr. Barr pioneered both the linking loader -- which reduced program execution time substantially on the IBM 360 -- and multileaving communications between IBM computers and non-IBM terminals. He wrote HASP multileaving terminal programs for the University Computing Company's COPE terminal (1970) and the Singer Corporation's remote batch terminal (1971). Entering yet another area of computing, he developed in 1969 the ACME system, which the National Center for Health Statistics uses to process death records automatically and from which are produced all national mortality statistics. Mr. Barr's most extensive project, however, has been SAS.

Gleaning and combining ideas from his work with command-and-control systems and with least-squares programs, Mr. Barr began development of SAS in 1966; he was joined a year later by James H. Goodnight.

Mr. Barr claims, "People's problems with data are very similar, whether they are processing survey data, lab data, or the data that business or military installations accumulate." His goal is "to have one consistent language" which people of widely varying professional backgrounds can learn and apply to solve their problems. His greatest concerns are making SAS handle large problems easily and making the system easy to extend to customized applications. "Our SAS language is a language for describing simple programming operations as well as complex processes to be performed," says Mr. Barr. "It's a more problem-oriented language than PL/I or FORTRAN, which have been used on the same sort of file-processing problems. I feel that with SAS-75 we shall have a nearly complete system for processing and describing data sets which are organized as they are now in SAS."

Mr. Barr dreams, though, of another "generation" of SAS, which would introduce the concept of hierarchically organized data sets. In such a data set, an observation could consist not only of data elements (values of individual variables) but of data sets as well. This structure, believes Mr. Barr, would increase tremendously the flexibility of the data handling and the ease of programming and processing the output from procedures.

USER'S FORUM

Gentlemen:

I have been using your system of SAS for some time now . . . I would like to make one request, as a suggestion, not a criticism. The ability to lag variables, at various lengths of time, by my choice would be of great help to me in my use of regression analysis.

I would hope you will consider this suggestion, as many people who work in Economics, as I do, would really appreciate this addition to your program.

Thank you,

Danny Tatar
DeKalb, Illinois

Lagging is currently possible by the proper use of the OUTPUT statement.

Suppose you want two lags of X called XLAG1 and XLAG2. Then the following statements will suffice:

```
DATA; INPUT X etc.;
OUTPUT; XLAG2 = XLAG1; XLAG1 = X;
```

Notice that the computed variables, XLAG1 and XLAG2, have missing values until they are set to be non-missing. PROCEDURE REGR will throw away all observations with missing data. Also note that when you use an OUTPUT statement, there is no longer a default output at the end of the program statements or upon encountering a RETURN statement. Try this method out using PROCEDURE PRINT. The generalization to higher-order lags should be obvious now.

This feature has not been obvious, and we intend to show lagging examples in next year's edition of the SAS User's Guide.

John Sall
SAS Project

[Ed note: See also the "Retention of Values" section (pages 30-31) and the "OUTPUT Statements" section (pages 33-34) of the User's Guide.]

BUG

We have discovered that using more than 256 variables with the REGR, ANOVA, or FREQ procedure may cause the procedure to "blow up" or go into an infinite loop.

To avoid this difficulty, a user working with a data set of more than 256 variables can create one or more subsets of his original data set, using the DROP or KEEP statement to limit each subset to 256 or fewer variables. The subsets can then be processed by the REGR, ANOVA, or FREQ procedure.

NEW POLICY ON SAS PUBLICATIONS

Service, Jolayne. A User's Guide to the Statistical Analysis System. 1972.

The User's Guide is the basic reference for using SAS. Degree-granting educational institutions can order copies from the Student Supply Stores, North Carolina State University, P. O. Box 5245, Raleigh, NC 27607; other installations can order copies from Sparks Press, 115 Harrison Avenue, Raleigh, NC 27603. In either case, the cost is $3.95 per copy, plus 3% North Carolina sales tax plus 1% Wake County tax (residents only), plus mailing and handling charges ($0.39 for the first copy, $0.20 for each additional copy). Note: A revised edition of the User's Guide will be prepared for the new version of SAS.

Perkins, Carroll Gray. A Guide to the Supplementary Procedure Library for the Statistical Analysis System. Last revised April, 1974.

Barr, Anthony James, and James Howard Goodnight. SAS Programmer's Guide. 1972.

The supplementary procedures guide describes special-purpose SAS procedures. The programmer's guide tells programmers how to implement their own SAS procedures. One can obtain copies of the latter two guides from the SAS Project Group, Institute of Statistics, North Carolina State University, Raleigh, North Carolina 27607, at a total cost of $2.00 per copy.

SAS-75

In our first issue of SAS Communications, we described the SAS Project Group's activities, particularly the revision of the "systems" portion of SAS and the development of new and improved procedures. We stated several goals:

(1) to make the system easier to understand;

(2) to make the internal code clearer;

(3) to increase the efficiency of SAS, particularly in handling data sets;

(4) to make "autobatching" of SAS possible;

(5) to make SAS easier to use under the Time Sharing Option (TSO);

(6) to make it simple to produce easily readable reports;

(7) to make SAS statements more flexible; and of course

(8) to expand SAS's statistical capabilities.

We are currently working to have a provisional version of the new SAS -- called SAS-75 -- ready for distribution by February 15, 1975. SAS-75 as released then will not include all the present SAS procedures; many installations will hence wish to offer simultaneously the version of SAS now used and SAS-75. Users could then use all SAS's present capabilities and begin to try out some of SAS-75's important new features. All SAS's present capabilities will be incorporated as soon as possible into SAS-75.

Here, then, is an outline of SAS-75's new features:

I. Systems Features

A. New INPUT and PUT statements.

All INPUT statements presently valid in SAS will remain valid under SAS-75. The INPUT statement, though, will acquire new options, allowing the user to describe data using a format similar to PL/1 or FORTRAN formats. In addition, SAS will be able to handle free-form (list-type) data (a boon especially to people who use SAS interactively).

The PUT statement will acquire the same form as the INPUT statement and will be used to print or punch data, thus greatly enhancing SAS's report-writing capabilities.

B. Variable labeling.

A LABEL statement will associate a "label" of up to 40 characters with each variable name. Many of the procedures will use the label, if specified, rather than the variable name to identify items printed out. This feature too will contribute to SAS's role as a report generation tool.

C. Creation of several SAS data sets at one time.

The DATA statement will be expanded in function so that several SAS data sets can be created simultaneously as SAS passes through the input data. Program statements will determine which data elements are added to which SAS data set. This feature will increase greatly SAS's efficiency in processing large, complex collections of data.

D. Function to test the value of a variable in the next observation.

A new function, to be used in program statements, will check to see if, in the observation being processed, the value of a variable is different from the variable's value in the observation immediately following.

E. LENGTH statement.

Currently, every numeric data element stored as part of a SAS data set requires 8 characters of storage. This requirement has made it rather inefficient for users to store data which is mostly small integers as a SAS data set. The LENGTH statement will allow the user to specify how many characters (from 2 to 8) the values of a numeric variable are to take up on the storage medium. SAS's data management routines will then truncate each value to the specified number of characters before storing it. When such a stored variable is processed by a procedure, its data elements will each be expanded to 8 characters (without change of magnitude).

F. Conservation of paper.

SAS statements and messages will be printed separately from the output of procedures, thus eliminating the waste of paper which currently occurs (e.g., a single PROCEDURE statement and its procedure information statements, if any, now take up a whole page).

II. New and Revised Procedures

A. The FREQUENCY procedure.

A FREQUENCY procedure will produce multi-way frequency tables as a series of two-way tables. Three new algorithms will increase the efficiency of this procedure over that of the present procedure FREQ: a hash-coding algorithm will be used to produce one-way tables; a radix method (similar to the method used by compilers to handle multi-dimensional arrays) will be employed to produce multi-way tables; and if the data is sorted, special logic will be used to acquire the frequencies.

B. The CONTENTS procedure.

This new procedure will give a physical and logical description of a stored SAS data set.

C. The GLM procedure.

A GLM (for General Linear Models) procedure is being developed. It will combine the features of the present REGR and ANOVA procedures. For balanced designs, GLM will use the computational methods currently used in ANOVA; for unbalanced designs, it will use the techniques of REGR. Several new types of effects will be valid, including interactions and powers of continuous (numeric) variables, interactions between continuous variables and classification variables, and the effect of a continuous variable nested within a classification variable. One will be able to produce least-squares means as well as adjusted means. New missing-value rules will be established; the procedure will no longer omit an entire observation when one of the dependent variables has a missing value.

D. The NLIN procedure.

SAS-75 will include a procedure for least-squares estimation of the parameters of a non-linear model. The user will describe the model and derivatives with SAS statements that are similar in form to present SAS program statements. The procedure will be able to use any one (or a combination) of the following techniques:

(1) modified Gauss-Newton method,

(2) modified gradient method of Marquardt,

(3) parabolic approximation technique of Mason, and

(4) grid search to find starting values.

E. The new DISCRIM procedure.

In addition to its prexent capabilities, the DISCRIM procedure will allow the user to store calibration information and apply it to other data sets.

F. The new FACTOR procedure.

The FACTOR procedure is undergoing complete revision. The new version will offer several methods of communality estimation, several types of orthogonal and oblique factor rotation, and production of factor scores.

G. The SYSREG procedure.

This procedure will estimate the parameters of interdependent systems of equations. SYSREG can use any of the following estimation methods: classical least squares, two-stage least squares, three-stage least squares, and limited information maximum likelihood. Features include restricted estimation, tests of hypotheses, and creation of a data set containing residuals and predicted values.

A NOTE ON EFFICIENCY

Stored SAS data sets are kept on disk or tape as specified by the user; other SAS data sets created in the course of a job are stored on disk for the duration of the job. Hence every time a SAS data set is processed -- when it appears in a SET or MERGE statement or when a procedure is applied to it -- SAS must fetch the observations from the data storage unit and bring them into the central processing unit of the computer to work with them. This reading of observations from disk or tape can become quite costly when the data set has about 40,000 observations. The user is levied a charge each time SAS goes to fetch another group of observations. By increasing the blocksize of the SAS data set, however, the user can reduce the cost.

SAS reads SAS data sets block by block. Unless otherwise specified by the user, the blocksize of a SAS data set, stored or not, will be 2048. A SAS data set created just for the duration of the job is seen by the computer's operating system (OS) as a member of a partitioned OS data set whose data definition name (the name OS recognizes) is SASDATA. By telling OS that the blocksize of SASDATA is something larger than 2048, the user forces SAS to read larger blocks, containing more observations, than it ordinarily would. Furthermore, SAS is written so that the blocksize stated or implied for SASDATA is assumed to be the blocksize for all stored SAS data sets mentioned in that job as well. Thus the single Job Control Language statement

```
//SAS.SASDATA DD DCB=BLKSIZE=6400
```

appearing immediately after the

```
// EXEC SAS
```

card will make SAS take the number 6400 to be the blocksize of all the job's SAS data sets, whether stored or members of SASDATA.

When in a subsequent job one is processing a stored SAS data set with larger-than-2048 blocksize, one must be careful always to include the Job Control Language statement shown above, with BLKSIZE= a number at least as large as the blocksize with which the data set was originally stored.

PROCEDURE MEANS: EXPANDED OUTPUT OPTIONS

[Ed. note: We are grateful to PAUL D. HOPKINS of the Statistical Reporting Service, Research Division, United States Department of Agriculture, for preparing and allowing us to adapt and reprint this documentation of the expansion of SAS's MEANS procedure.]

The latest release of SAS has a heretofore undocumented feature in the MEANS procedure. It allows users to create a SAS data set with any combination of the descriptive statistics the procedure can produce. Users may create SAS data sets with the numbers of observations, means, standard deviations, variances, sums, corrected sums of square, smallest values, largest values, coefficients of variation, or any combination of these items.

The particular item or items to be put in the output data set are selected by options on the PROC MEANS statement. There is one option for each item to be included in the output data set. The SAS data set produced always has the same variables as the input data set. If more than one type of statistic is desired, SAS creates multiple observations and adds a new character variable OBS_TYPE to the data set. The values of OBS_TYPE are the options used on the PROC MEANS statement, which identify the type of data in the remainder of the observation.

BY variables and character variables are included in the output data set. A BY statement causes a new series of observations for each distinct value in the BY variable. The value of a character variable included is the first value encountered in any BY group.

The VARIABLES statement produces the following results in the output data set: those variables included in the VARIABLES statement are properly processed and included in the output data set; all variables not in the VARIABLES statement are also written out, but contain the first value received in any BY group.

The PROC MEANS statement now has the following form:

```
PROC MEANS  <NOPRINT> <OUT=data_set_2> <DATA=data_set_1>

            <N> <MEAN> <STD> <VAR> <SUM> <CS> <LOW> <HIGH> <CV>;
```

The NOPRINT and DATA options are described in the SAS User's Guide. The OUT option is expanded to include the options on the second line. Those options specify the type of items to be included in the output data set.

Option	Type of statistic in observation
N	Number of observations.
MEAN	Mean or average. If no options are specified but the OUT parameter is included, the procedure will include only means on the output data set, data_set_2. MEAN must be specified if means and one or more other statistics are to be included in data_set_2.
STD	Standard deviation.
VAR	Variance.
SUM	Sum or total.
CSS	Corrected sum of squares.
LOW	Smallest value.
HIGH	Largest value.
CV	Coefficient of variation.

Examples:

The following SAS statements produce the data set EXAMPLE from cards and print it.

```
DATA EXAMPLE;  INPUT ID $ 1 X 2 Y 3 Z 4;  CARDS;
A123
A234
A345
A456
A 67
A  9
B987
B876
B765
B54
B2
PROC PRINT  DATA=EXAMPLE;
```

OBS	ID	X	Y	Z
1	A	1	2	3
2	A	2	3	4
3	A	3	4	5
4	A	4	5	6
5	A		6	7
6	A			9
7	B	9	8	7
8	B	8	7	6
9	B	7	6	5
10	B	5	4	
11	B	2		

The following SAS statements make MEANS create a SAS data set with all available output items then print it out.

```
PROC MEANS NOPRINT OUT=STATS DATA=EXAMPLE N MEAN STD VAR SUM CSS LOW HIGH CV;
BY ID;
PROC PRINT DATA=STATS;
```

Results:

OBS	ID	X	Y	Z	OBS_TYPE
1	A	4.0000	5.0000	6.0000	N
2	A	2.5000	4.0000	5.6667	MEAN
3	A	1.2910	1.5811	2.1602	STD
4	A	1.6667	2.5000	4.6667	VAR
5	A	10.0000	20.0000	34.0000	SUM
6	A	5.0000	10.0000	23.3333	CSS
7	A	1.0000	2.0000	3.0000	LOW
8	A	4.0000	6.0000	9.0000	HIGH
9	A	51.6398	39.5285	38.1220	CV
10	B	5.0000	4.0000	3.0000	N
11	B	6.2000	6.2500	6.0000	MEAN
12	B	2.7749	1.7078	1.0000	STD
13	B	7.7000	2.9167	1.0000	VAR
14	B	31.0000	25.0000	18.0000	SUM
15	B	30.8000	8.7500	2.0000	CSS
16	B	2.0000	4.0000	5.0000	LOW
17	B	9.0000	8.0000	7.0000	HIGH
18	B	44.7562	27.3252	16.6667	CV

Note that the STATS data set has the same variables as EXAMPLE, plus OBS_TYPE. There is an observation created for each type of statistic selected in the option list. The value of OBS_TYPE identifies the type of statistic found in the observation, and the value of ID tells which BY group the observation belongs to. Missing values are deleted from computation.

The following SAS statements produce a data set of totals or sums and print it out.

```
PROC MEANS NOPRINT OUT=TOTALS DATA=EXAMPLE SUM;  BY ID;
PROC PRINT DATA=TOTALS;
```

Results:

OBS	ID	X	Y	Z
1	A	10	20	34
2	B	31	25	18

Note that when only a single type of statistic is wanted, the variable OBS_TYPE is not added to the output data set.

The next two examples show what happens to the output data set when a VARIABLES statement is used with the MEANS procedure. Remember that all the variables in the input data set appear in the output data set.

```
PROC MEANS NOPRINT OUT=TOTXY DATA=EXAMPLE SUM; BY ID;
VAR X Y;
PROC PRINT DATA=TOTXY;
```

Results:

OBS	ID	X	Y	Z
1	A	10	20	3
2	B	31	25	7

Even though Z was omitted from the VAR statement, it is included in the output data set. Also note that the value saved is the first Z in each BY group.

```
PROC MEANS NOPRINT OUT=TEST DATA=EXAMPLE N MEAN SUM;  BY ID;  VAR X Y;
PROC PRINT DATA=TEST;
```

Results:

OBS	ID	X	Y	Z	OBS_TYPE
1	A	4.0	5.00	3	N
2	A	2.5	4.00	3	MEAN
3	A	10.0	20.00	3	SUM
4	B	5.0	4.00	7	N
5	B	6.2	6.25	7	MEAN
6	B	31.0	25.00	7	SUM

Again note that the variable Z was not used in the computation process, but does retain the first value found in each BY group.

- -

SAS Communications is published by the SAS Project Group:

A. J. Barr, Systems and Administration
J. H. Goodnight, Procedures
J. Service, Documentation and Editing
C. G. Perkins, Assistant for Systems
J. Sall, Assistant for Procedures
S. B. Donaghy, Consulting
H. J. Kirk, Consulting
J. P. Massengill, Secretary and Administrative Assistant
D. Beckwith, Assistant

Institute of Statistics
North Carolina State University
Raleigh, North Carolina 27607

SAS COMMUNICATIONS

November 1975
Number 5

nstitute of Statistics North Carolina State University Raleigh, North Carolina

SAS-76

SAS-76 is expected to be ready this spring. GLM, the new general linear models procedure that encompasses the SAS-72 procedures REGR and ANOVA, will appear in SAS-76. The new FREQ procedure for frequencies and crosstabulation will also be included, and capabilities for SAS data sets on tape will be ready too. Of course, all the new features now available in SAS 75.2 will also appear in SAS-76.

SAS 75.1 and SAS 75.2 TEST VERSIONS RELEASED

SAS 75.1, a test version of the new SAS-76, was sent to 20 installations in June. The system was in a rough form and lacked such important features as the PUT statement and the SORT procedure. Provisional documentation accompanied SAS 75.1.

In October, SAS 75.2 was distributed to about 50 test sites. SAS 75.2 is a usable system, although it is not a replacement for SAS-72. Some of the new features in SAS 75.2:

- Variable names of the form ALPHAn-ALPHAnnn can be used: e.g.,

```
INPUT NAME $ 1-20  Q1-Q25 ;
```

- Several data sets can be created at the same time.
- The INPUT statement has many new capabilities. Free-format input data can be read: e.g., "INPUT X Y Z;" Zoned decimal, hexadecimal, and positive integer binary data can be read. Several INPUT statements can be used to read a file containing several record types (see "Application Note", p. 5). Repetitive formats can be given to read many variables with one format.
- The PUT statement, which corresponds to the INPUT statement, can be used to write reports, to write data on tape or disk, and to punch cards.
- The INFILE and FILE statements give information about the file read by the INPUT statement and the file written by the PUT statement, respectively.
- A number of new functions are available that give sample statistics.
- An ABORT statement lets you halt SAS execution immediately.
- The FORMAT statement lets you specify a format for a variable; SAS uses this format whenever it prints the values of the variable.
- The LABEL statement lets you give variable labels.
- The LENGTH statement allows you to specify the number of characters used to store variables.
- The MERGE statement has been renamed UPDATE and has some new capabilities.
- Up to 10 TITLE statements can now be given.

New procedures in SAS 75.1 and 75.2:

- AUTOREG is a new procedure for autoregression.
- BMDP calls any BMD P-Series program to analyze data in a SAS data set.
- CONTENTS prints descriptions and histories of your existing SAS data sets.
- CONVERT lets you convert SAS-72 data sets, SPSS system files, OSIRIS datasets, and BMDP save files to SAS data sets.
- FORMS can be used to print any continuous line printer forms: mailing labels, pre-printed forms, envelopes, external magnetic tape labels, and so forth.
- MATRIX, a comprehensive matrix-handling procedure, lets you use matrix operations and functions to solve a variety of problems. At NCSU, MATRIX has been used by students in two statistical methods courses to do homework exercises.

  Here is an example of using MATRIX to find the least squares estimates for a linear model:

```
PROC MATRIX PRINT;
X = 1  1  1 /
    1  1 -1 /
    1 -1  1 /
    1 -1 -1 ;
Y = 5 / 7 / 8 / 9;
BETA = INV(X'*X)*(X'*Y);
```

- NEIGHBOR performs nearest neighbor and nearest k-neighbor discriminant analysis.
- NLIN performs nonlinear regression, finding least squares or weighted least squares estimates of coefficients of a nonlinear model.
- PROBIT calculates maximum likelihood estimates for biological assay data.
- SCATTER replaces the SAS-72 procedure PLOT. A new OVERLAY option makes it easy to plot observed and predicted values.
- SPECTRA produces estimates of spectral and cross-spectral densities of a multivariate time series
- SAS72 calls any SAS-72 procedure not available in SAS 75.2. For example, you might use it to call REGR.
- TTEST computes t-statistics for two groups.
- VARCOMP computes estimates of the variance components in a random effects model.

In addition, major changes were made to the DISCRIM and FACTOR procedures.

Provisional documentation for SAS 75.2 is ready. Copies can be ordered from:

SAS Project
Institute of Statistics
North Carolina State University
P.O. Box 5457
Raleigh, North Carolina 27607

The price of $4.00 each includes postage. Please limit your order to five copies, since only a limited number is available.

SAS USERS' MEETINGS

A SAS Regional Users' Meeting, sponsored by Abbott Laboratories, was held in Chicago on July 21-22. More than 80 SAS users attended; papers were given by Jim Barr, Jim Goodnight, and eleven users. A demonstration of SAS 75.2 under TSO was presented. Attendees voted on a number of suggestions for future changes to SAS.

Because of the success of that meeting, an International SAS Users' Meeting is planned by Julian and Evey Horwich for the Orlando Hyatt House, Kissimmee, Florida on January 26-28, 1976. On Monday, January 26, Jim Goodnight, Jim Barr, and John Sall will provide progress reports, problem sessions, and tutorials. Tutorials planned include Procedure Writing, PROC MATRIX, SAS System Programming Features, PROC GLM, and Report Writing and Data Management. Users from the United States, Central and South America, England, and Canada have indicated probable attendance at the meeting. If you are interested in attending the conference, drop a note to:

SAS USERS' MEETING
1239 GLENCOE AVENUE
HIGHLAND PARK, ILLINOIS 60035

A tentative list of sessions scheduled for the SAS Users' Meeting in January begins on page 7.

SAS Communications is published by the SAS Project:

A. J. Barr, Systems and Administration
J. H. Goodnight, Procedures and Administration
J. P. Sall, Procedures
C. G. Perkins, Systems
J. H. Helwig, Communications
S. B. Donaghy, Consulting
H. J. Kirk, Consulting
J. P. Massengill, Administrative Assistant
D. K. Beckwith, Assistant

Institute of Statistics
North Carolina State University
Raleigh, North Carolina 27607

(919) 737-2585

SAS STAFF CHANGES

For almost four years, Jolayne Service has been a member of the SAS Project. Jo wrote A User's Guide to the Statistical Analysis System, which describes SAS-72, and revised it to describe SAS 75.1. Jo also edited SAS Communications. Although she worked with SAS only part-time, Jo's contributions have made SAS accessible to thousands of users.

Jo has resigned her position with the SAS Project to pursue full-time a doctorate in educational research and evaluation. Our thanks and best wishes go with her.

Jane Harriss Helwig has taken over Jo's duties and will edit all SAS manuals and SAS Communications. Jane, a SAS user since 1972, received a B.S. in chemistry from the University of North Carolina at Chapel Hill, where she has also done graduate work in economics. Jane comes to SAS from the User Services group at the UNC-CH Computation Center.

Carroll Perkins, who has been with the SAS Project since 1972, is leaving to become a systems analyst for the Corporate Data Center of Hanes Corporation in Winston-Salem, North Carolina. Carroll's contributions to SAS include several procedures, the Programmer's Guide, and the Supplementary Procedures Guide. We wish Carroll well in his new position.

John Sall, with SAS since 1973, is taking an increasingly larger role in the project. John is a heavy contributor to SAS-76: he has written twelve procedures, including MATRIX. John holds a B.A. from Beloit College, has a master's in economics from Northern Illinois University, and is working on his Ph.D. in statistics here at NCSU.

APPLICATION NOTE: THE TRAILING @

An @ as the last item in an INPUT statement tells SAS 75.2 to keep the current observation rather than reading a new one when the next INPUT statement is encountered. You can use this feature to select only records of a given type for processing. Here is an example:

```
DATA SMF4;
INFILE SMF;
INPUT @3 REC_TYPE PIB2. @;
IF REC_TYPE = 4 THEN GO TO A;
INPUT; DELETE;
A:  INPUT JOBNAME $ 15-22 @40 TIME PIB4.;
.
.
.
```

Note that when the record is not wanted, the statement "INPUT;" must appear before the DELETE statement. This INPUT statement removes the effect of the trailing @; the next INPUT statement will then read a new observation. If "INPUT;" were left out, the program would loop.

When you use the trailing @, remember that its use means the next INPUT statement executed will not read a new observation. So a program like

```
DATA;
INPUT ID 1-2 @; CARDS;
```

will read the first record over and over until time runs out.

On the other hand, the program

```
DATA;
INPUT ID @; CARDS;
```

reads each value that it sees on the first record as a value of ID. When the first record has been completely read, it continues on through the second and later records.

SAS ENHANCEMENT SUGGESTIONS

Suggestions for SAS enhancements were circulated at the Regional SAS Users' Meeting in July. Below are most of the suggestions on the list, along with our comments and the timetable for adding the features to SAS

	SUGGESTIONS	COMMENTS
Now available:	syntax-checking mode	add as first statement in SAS 75.2 program: OPTIONS OBS=∅;
	variable labels	use LABEL statement
	user-written functions	see 75.1 Programmer's Guide
	partial correlation procedure	use REGR to get partial correlations
	ANOVA with repeated measures	use either REGR or ANOVA
	less printout	SAS 75.2 log cuts down on paper use
	improved run times	SAS 75.2 runs faster than SAS-72
	bar graphs	in HISTOGRAM
	ability to halt program	use ABORT statement
	BMD P-Series interface	use BMDP and CONVERT procedures.
	VBS fixes	fixed in SAS 75.2
	HISTOGRAM fixes	fixed in SAS 75.2
to be in SAS-76:	mean separation techniques in REGR and ANOVA	GLM (General Linear Models) procedure in SAS-76 will have several mean separation techniques
	cumulative percentages in FREQ	will be in FREQ in SAS-76
	TEST statement in REGR	GLM in SAS-76 will have TEST statement
to be added in 1976:	SUBSTR operation	
	BIT variables	will be able to test bits of variable
	value labels	
	%INCLUDE from macro library	
	multiple data set merge	
	output regression coefficients	
	automatic plots of predicted vs. actual values	
	median, mode, confidence limits, n-tiles in MEANS	
	more about SAS internals and data set formats	
under consideration:	nonparametric statistics	
not under consideration at this time:	plotter and microfiche interface	
	near neighbor variance estimates	
	user profiles	
	IMS interface	
	PL/I Optimizer-compiled SAS	SAS would be dependent on software rented from IBM
	FORTRAN H-extended compiled SAS	only one FORTRAN procedure in SAS 75.2

SAS communications

Vol. II, No. 1 July 1976

SAS Institute Inc.

On July 1, total support for SAS was assumed by SAS Institute Inc., a private company devoted to the maintenance and further development of SAS. Anthony J. Barr, James H. Goodnight, John P. Sall, and Jane T. Helwig, former members of the SAS Project, are now with SAS Institute, along with Joyce Massengill and Debra Calloway.

Through agreement with the Institute of Statistics of North Carolina State University, SAS Institute now supports all current service agreements between the Institute of Statistics and SAS installations.

As SAS Institute, we intend to maintain SAS at the same high level as in the past. We will continue to do our best to make SAS as useful as possible for solving data analysis problems.

The offices of SAS Institute are at 2806 Hillsborough in Raleigh, across the street from North Carolina State University. Both Jim Barr and Jim Goodnight have adjunct positions on the NCSU faculty.

Of course, our address has changed. Please send all correspondence to

SAS Institute Inc.
Post Office Box 10066
Raleigh, North Carolina 27605

Note that our box has changed from 10522, the interim number, to 10066.

Our new telephone number is **(919) 834-4381.**

SAS•76 Released

We are happy to announce the official release of SAS•76. All installations that have current service agreements received SAS•76 packages, containing the 76.2 version of SAS, during the first part of July.

SAS•76 is the product of four years' improvements and additions to the successful 1972 version of SAS. Most procedures have been rewritten, and many new ones have been added. About 70,000 statements make up the source, which is written almost entirely in PL/I and assembler.

SAS•76 contains many new statistical capabilities, plus greatly enhanced data management facilities. Highlights are:

- GLM, the general linear models procedure
- MATRIX, a comprehensive matrix-handling procedure
- flexible and extensive facilities for reading, printing, sorting, merging, concatenating, updating, and storing data
- a new report-writing facility
- the ability to use SAS interactively under TSO.

New Manual Ready

A User's Guide to SAS•76 is now ready. The manual provides information needed to use SAS•76, with complete descriptions and extensive examples of data management abilities and statistical procedures.

An index is included, along with a glossary of computer and SAS terms. Appendixes deal with such topics as converting SAS•72 jobs to SAS•76; using SAS under TSO; and the theory behind the new GLM procedure.

Copies of **A User's Guide to SAS•76** may be ordered through your local bookstore or from:

SAS Institute Inc.
Post Office Box 10066
Raleigh, North Carolina 27605

The cost is $6.95 each, plus shipping charges. For overseas orders, please specify Air Freight or Surface Mail when ordering.

A User's Guide to SAS•76
Anthony J. Barr, James H. Goodnight,
John P. Sall, and Jane T. Helwig
SAS Institute Inc., Raleigh, 1976.
330 pp. (ISBN 0-917382-01-3)

Programmer's Guide Revision

The **SAS Programmer's Guide**, the manual for those who want to write their own SAS procedures, is now being revised. We expect that copies will be available around September 1.

New Orleans Is Meeting Site

At the January meeting in Orlando of SAS users, Dr. Ronald Helms of the University of North Carolina at Chapel Hill was chosen Chairperson of the Users' Group Steering Committee.

Other members of the Steering Committee are Dr. Michael P. Farrell of Miami University in Ohio; Kenneth Offord of the Mayo Clinic; Richard J. Olson of Oak Ridge National Laboratory; Dr. Philip Miller of Washington University in St. Louis; Cathy Milne of Procter and Gamble; and William Taylor of the Biometric Research Institute in Washington, D.C. Advisers to the committee are Julian Horwich of Abbott Laboratories and H.W. Barry Merrill of State Farm Mutual Insurance Company.

Dr. Helms reports that the Steering Committee is now planning the Second International SAS Users' Conference, to be held in late January or early February of 1977. The committee has selected New Orleans as the conference site.

Conference Proceedings Available

The **Proceedings** of the International SAS Users' Conference, held January 26-28 in Orlando, are now being published. Conference attendees will each receive a copy when they are ready.

Additional copies of the **Proceedings** are available from SAS Institute at $10.00 each.

Newsletter Quarterly

To keep in better touch with our users, SAS Institute will publish **SAS Communications** quarterly.

Let us know if you would like to see an article on a particular topic. If you want to share information about SAS, send it in.

Interesting Samples

One way to learn more about interesting SAS applications is to look at the sample SAS jobs found on each SAS istallation tape. Besides sample jobs to test each procedure, examples like these are included:

ANOVA2	analyzing a Latin-square split plot design
CENSUS	reading hierarchical files of the U.S. Census Bureau Public Use Sample tapes
HARRIS	reading Harris Poll tapes coded in column-binary format
IEHLIST2	reading a PDS directory
KIVIAT	plotting Kiviat graphs
LOADMAP	reading and mapping a load module
MAPDISK	reading a VTOC and mapping a disk pack
SMFPAGE	analyzing SMF type 71 records (courtesy J. Frank Chambers, Avco Financial Services)
TEACH	teaching your child arithmetic

The library is found on the SAS tape in unloaded-PDS form. Check with the person who installed SAS to find out how to access the library, since it may have been put on disk.

Here is a sample SAS job to print a copy of the TEACH program, mentioned above. In this example, the library is named SAS.SAMPLE and is a cataloged disk data set:

```
//COPY JOB account_code,username
//   EXEC SAS
//LIBRARY DD DSN=SAS.SAMPLE,DISP=SHR
//SYSIN DD *
DATA;
INFILE LIBRARY (TEACH);
INPUT;
LIST;
/*
```

Note that this example is written for SAS•76.

PROC PRINTTO

PRINTTO is a new SAS procedure that gives you control over the output of SAS procedures. Normally, this output goes to the printer. With PRINTTO, however, you can

- write SAS output on tape for COM (computer output to microfiche)
- selectively suppress SAS output
- print several copies of SAS output
- use SAS output as input data, in the same job.

PRINTTO transfers only the output pages produced by SAS procedures. The log of SAS statements that SAS•76 prints at the beginning of each job is not affected.

1. Here is an example that puts the output of PROC FREQ onto tape:

```
//COM JOB acct_code,username
//   EXEC SAS
//FT20F001 DD UNIT=2400,VOL=SER=TAPE05,DSN=TABLE,
//    DCB=(RECFM=FB,LRECL=133,BLKSIZE=3990)
DATA;
INPUT X Y Z;
CARDS;
...
PROC PRINTTO NEW UNIT=20;  * OUTPUT TO UNIT 20;
PROC FREQ; TABLES X*Y*Z;   * OUTPUT TO TAPE;
PROC PRINTTO;              * BACK TO PRINTER;
PROC CORR;                 * WILL BE PRINTED;
```

The statement

```
PROC PRINTTO NEW UNIT=nn;
```

changes the default output unit from the printer to FTnnF001. (In the example above, nn is 20.)

Subsequent SAS procedure output will be written on the file defined by the FT20F001 DD statement; in this case, that file is on a tape. The NEW option initializes the file, using the DCB attributes RECFM = VBA, LRECL=137, and BLKSIZE= 6391. They can be overridden by the JCL.

The statement

```
PROC PRINTTO;
```

resets the default output unit to the printer.

The statement

```
PROC PRINTTO UNIT=20;
```

does not appear in the example, but it could have been used later in the job to write more output on the file described by the FT20F001 DD statement.

2. This example prints four copies of the FREQ and MEANS output for a data set containing student grades. This output is written to FT20F001, a temporary disk file, which is then read back in and printed four times.

```
//GRADES JOB acct_code,username
// EXEC SAS
//FT20F001 DD UNIT=3330,SPACE=(TRK,(15,5))
DATA; INPUT SSN 1-9 GRADE 11-13; CARDS;
. . .
PROC PRINTTO UNIT=20 NEW;  * PRINT TO UNIT=20;
PROC FREQ; PROC MEANS;     * PRODUCE STATISTICS;
MACRO PR
DATA _NULL_; FILE PRINT NOPRINT;
INFILE FT20F001; INPUT;
PUT _INFILE_; %
PROC PRINTTO;              * SET PRINT BACK TO PRINTER;
PR PR PR PR                * EXECUTE MACRO 4 TIMES;
/*
```

3. Here is an example that puts GLM output on a temporary disk file. The output is then read back in and each line printed: this output will look the same as if the PROC PRINTTO statement had not appeared. Next, the temporary file is read again, and this time, beta values and error mean square values are read and put into a SAS data set. PROC MEANS is then used to find the mean of the ERRORMS values.

```
//GLM JOB acct_code,username
// EXEC SAS
//FT20F001 DD  UNIT=3330, SPACE=(TRK,(20,5))
DATA; INPUT X1-X10 Y1-Y5; CARDS;
. . .
PROC PRINTTO UNIT=20 NEW;       * SEND OUTPUT TO 20;
PROC GLM; MODEL Y1-Y5=X1-X10;   * GLM OUTPUT ON 20;
PROC PRINTTO;                   * PRINTER NOW DEFAULT;
DATA _NULL_; INFILE FT20F001; INPUT;
FILE PRINT NOPRINT; PUT _INFILE_;
DATA;
INFILE FT20F001;
INPUT @2 NAME $ @;
IF NAME='ERROR' THEN INPUT B1 B2 ERRORMS;
IF NAME='ERROR' THEN OUTPUT;
PROC PRINT;                     * PRINT BETAS AND ERRORMS;
PROC MEANS; VAR ERRORMS;        * GET MEAN OF ERRORMS;
```

PUT _INFILE_

INFILE, new in SAS•76, lets you use the current data line or card in PUT statements. For example, this SAS job prints all eighty columns of each data card:

```
DATA;
INPUT X 10-12 Y 24-28;
PUT _INFILE_;                  * PRINT ENTIRE INPUT CARD;
CARDS;
.
. data cards
.
```

You can use _INFILE_ in PUT statements as you would other variables. For example, this program pairs 80-byte input records to produce a file with an LRECL of 160:

```
DATA;
INFILE IN;
FILE OUT LRECL=160;
INPUT; PUT _INFILE_ @;
INPUT; PUT @81 _INFILE_;
```

You can also use other variables with _INFILE_ to rewrite parts of the input lines. For example, this SAS job duplicates a card deck, putting sequence numbers in columns 73-80:

```
DATA;
INPUT;
N + 100;                  * INCREMENT SEQUENCE NUMBER;
FILE PUNCH;
PUT _INFILE_ @73 N Z8. ;
CARDS;
.
. cards to be reproduced
.
```

Leading zeros will appear on the output cards because N values are written with the Z format. If columns 73-80 of the data cards contain any information, the sequence numbers will be "written over" it.

The LENGTH parameter of the INFILE statement can also extend _INFILE_ capabilities. For example, you can truncate _INFILE_ with a program like this:

```
DATA;
INFILE CARDS LENGTH=L;
INPUT;
L=L-20;
PUT _INFILE_;             * ONLY COLUMNS 1-60 PRINTED;
CARDS;
. . .
```

Another parameter of the INFILE statement that extends the usefulness of _INFILE_ is START (not documented in **A User's Guide to SAS•76**). You can define the starting column for _INFILE_ with a program like this:

```
DATA;
INFILE CARDS START=S;
INPUT;
S=20;
PUT _INFILE_;             * ONLY COLUMNS 20-80 PRINTED;
```

The LENGTH and START parameters can be used together. For example, this program will output _INFILE_ less the first 10 characters and the last 20 characters:

```
DATA;
INFILE TAPE START=S  LENGTH=L;
INPUT;
S=11; L=L-20;
FILE TAPEOUT;             * COLUMNS 11-60 PRINTED;
PUT _INFILE_;
```

Input Formats

Rules for input formats, given on pages 15-16 of **A User's Guide to SAS•76**, have been relaxed. One rule applies to numeric format items of the form w.d, where

w is the number of columns used to hold the value,

d is the number of places right of the decimal.

The d value, which formerly had to be less than w, now

has no limits. When the d value is greater than w, zeros are inserted to provide the proper number of decimal places.

For example, using

```
INPUT (X1-X10) (1.3);
```

to read the card

```
3276542201
```

produces an X1 value of .003, X2 value of .002, and so on.

The PIB., RB., IB., PD., and ZD. formats can now have decimal items with values ranging from 0 to 10. Thus, formats like these are valid:

RB4.2
PIB2.4
ZD8.2

Note that these changes apply only to **input** formats (used in INPUT and INFORMAT statements). For output formats (used in PUT and FORMAT statements), the number of decimal places must be smaller than the column width.

SAS Communications is published quarterly by SAS Institute Inc.

Anthony J. Barr, Systems
James H. Goodnight, Procedures
John P. Sall, Procedures
Jane T. Helwig, Communications
Joyce P. Massengill, Administration
Debra L. Calloway, Assistant

Address all correspondence to SAS Institute Inc., Post Office Box 10066, Raleigh, NC 27605.

SAS INSTITUTE INC.
Post Office Box 10066
Raleigh, North Carolina 27605

MR. RAYVON STURDIVANT
COMPUTATION CENTER
PHILLIPS HALL
UNIV. OF NORTH CAROLINA
CHAPEL HILL, NC 27514

SAS communications

Vol. II, No. 2 October 1976

It's January 3-5 In New Orleans

The SAS Users Group, International (SUGI) will hold its annual conference in New Orleans January 3-5, 1977. Those on the **SAS Communications** mailing list received conference and registration information a few weeks ago.

Dr. Ronald Helms, SUGI chairman, promises a conference that will be "informative, educational, useful to your work, and a great deal of fun." He encourages SAS users to submit abstracts of papers or "Darn Cute Code" to Ken Offord, Program Chairman, Mayo Clinic, Department of MEDRES, Rochester, Minnesota 55901.

For information about the conference, write or call Dr. Ronald Helms, Post Office Box 957, Chapel Hill, North Carolina 27514 (919/942-3175).

SAS Autobatch Is Ready

We have completed the SAS autobatch supervisor. Under autobatch mode, many SAS jobs are run sequentially with the system overhead of only one job. Thus, costs are often cut substantially. Autobatch is particularly useful in educational environments where students run many small SAS jobs.

If you want to run SAS in autobatch mode now, let us know and we will send you the SAS 76.4 interim release. The next general distribution tape, to be sent to all installations this winter, will include the autobatch supervisor.

SAS Distribution Tapes

Most SAS installations are now running SAS version 76.2. New installations are receiving SAS version 76.4, and this version is available to other installations that request it. (SAS 76.3 was a transient version that ran only locally.)

During the winter, SAS version 76.5 will be sent to all installations. Major new features included in SAS 76.5 will be facilities for storing SAS data sets on tape; the autobatch supervisor; means in GLM and ANOVA; a new plot procedure; a new histogram procedure; and date and time formats.

INQUIRE Interface Now Available

A mutual development effort by ICI-United States and SAS Institute has produced a versatile interface routine that converts an INQUIRE data base to a SAS data set.

The interface is written as a SAS procedure. Users can specify selected fields for conversion, or convert all fields.

The project was spearheaded by Dr. Robert Teichman of ICI. David Haynes and John Duffy of ICI cooperated with Dr. James Goodnight of SAS Institute in the implementation of the interface.

The INQUIRE-SAS interface is now available from SAS Institute on a one-time fee basis. Let us know if you are interested in acquiring it.

More New Formats

SAS 76.2 contains several formats that were added after the **User's Guide** went to press:

dollars
When the DOLLARw.d output format is used for numeric variables, a dollar sign is placed in front of the number. In addition, commas mark off thousands of dollars. The d value must be either 0 or 2. In calculating the w value, which can range from 2 to 32, allow for the dollar sign and any commas that will be needed.

words
The WORDSw. output format prints a numeric value in English: for example, -176 prints as "MINUS ONE HUNDRED SEVENTY-SIX." When the data contains one or two decimal places, they are used: 5.20 prints as "FIVE AND TWENTY HUNDREDTHS." Numbers greater than 99,999,999 print as "LARGE NUMBER." The w value can range from 8 to 200; the default is 10. When insufficient space is provided, the result in truncated on the right and the last character prints as "*".

column binary
Four new input formats provide SAS users the ability to read column binary data, also known as multi-punched data. Each card column contains several pieces of information rather than one as usual. Although column binary is rare and becoming obsolete, many important data libraries are in column binary form.

The column binary documentation is too lengthy to show here; if you are interested, let us know and we will send you a copy.

other new formats
Other new formats have been developed and will appear in the next general release:

- unsigned packed decimal
- zoned decimal in IBM 1410, 1401, and 1620 format
- Roman numerals
- fractions.

Next on the format development schedule are comprehensive date and time formats.

Field Report: SAS 76.2

Initial reports on SAS 76.2 have been good. Of the problems that have arisen, many have already been fixed by the zap cards that were sent to all installations. Others are being fixed for the next SAS distribution tape.

Listed below are problems that might affect the correctness of your results without warning:

- In GLM, when there is a continuous by class interaction, predicted values output to a SAS data set are in error. Printed predicted values are correct.
- In AUTOREG, the transformation of the first few observations is slightly off.
- In MATRIX, the DET function's result may have the wrong sign.

In addition, an incompatibility between SAS 76.2 and IBM 3340 and 3350 disk drives causes intermittent errors when these drives are used for SAS data sets.

These problems have been fixed for interim release SAS 76.4.

User's Guide Corrections

A User's Guide to SAS•76 is now in its third printing. In the second and third printings, we fixed several errors. Most of these were obvious to the reader. Listed below are those problems that a reader would not notice:

- p. 42, column 2, 10th line from bottom: the word "not" should not appear
- p. 219, 2nd line from bottom of input example should contain the statement
  MODEL Y=X1-X5 / START=3 STOP=4;
- p. 234, midway through column 2. Paragraph should read: If the IBM 2314 storage device is used, s must be **at least** n*v/50000.

In addition, PROC STANDARD's description is off slightly. PROC STANDARD's default action is to create a new data set containing the standardized values rather than replacing the original values with standardized values. This change was made to protect users from inadvertently destroying their original data.

One Site's SAS Use

At many installations, SAS is used for much more than statistical analysis. Recently, we received a letter from Walter J. Guthrie, Group Leader in the Systems Support Department of Cleveland Trust, giving some of the ways SAS is used in his department. Excerpts from Jim's letter are printed below.

> ...I have been using PROC CORR as an application program debugging tool. I read from SMF files the step CPU, the step elapsed time, and EXCP's by DSNAME. I then correlate CPU time and elapsed time against each file's EXCP's. I have seen correlations as high as 95%, which means that if the user can reduce the size of that file, or if the programmer can optimize that data set, program run time or CPU time can be cut significantly.
>
> ...we used SAS to combine information out of our SLACMON software monitor. The spooled printouts of SLACMON (run for one hour each) are read by SAS and SAS summarizes module usage.
>
> ...the programmer who maintains OS/MVT uses SAS to build cross-references from SMP data. SMP provides a list of PTF's, prerequisites, and successors, as well as affected module names. The data is sorted in different orders. We can go by module name and find all PTF's against that module.
>
> ...after SAS76 was installed, a programmer used it to do frequency distributions on the BankAmericard logtape. He printed activity by transaction type and by hour. He also did a distribution on dollar volume by hour.
>
> ...another programmer is installing UCC's Tape Management System (TMS) Version 4.2. A SAS program was written to read the Version 4.2 log and create transactions for the Version 3.1 update, in case we had to revert to the old version.
>
> ...all our files are on a SAS card system. Each piece of correspondence in our master file has a card with TO-FROM-SUBJECT-DATE-RETENTION. These cards are sorted and listed by each variable.
>
> ...our IBM manuals each have a card. They are sorted and printed in number and subject sequence.

From Cleveland Trust's report and from others like it we have received, it seems clear that SAS applications are limitless. We'd be interested in hearing from others on the uses that they have found for SAS.

Programmer's Guide Delayed

Revision of the **SAS Programmer's Guide,** the manual for those who want to write their own SAS procedures, was delayed until recently. We are now rewriting much of the **Programmer's Guide** and expect that it will be a more usable manual.

Publication date for the **Programmer's Guide** is now December 1.

SAS Training Courses

Several two-day SAS training courses have been given, and more are planned. Courses so far have been oriented to new SAS users. However, since we tailor each training course for the prospective audience, more advanced subjects (e.g.. writing SAS procedures) could be covered.

If your organization is interested in learning more about these SAS training courses, which are normally given at the customer's site, please let us know.

In addition to these courses, we plan to offer SAS short courses in central locations so that individual users can attend. Notice of these courses will be sent to all **SAS Communications** recipients.

Staff Additions

Users who call SAS Institute may hear a new voice answering the telephone: Kathy Fulp has joined us as secretary.

Also joining us is Ann Baggett, who is handling publication orders.

SAS communications

Vol. II, No. 3 January 1977

SAS Users Meet In New Orleans

Warm, sunny New Orleans was host to more than 200 SAS users in January for the second annual meeting of the SAS Users Group, International (SUGI). Meeting participants learned new uses of SAS, shared knowledge of the system, and heard news about the future of SAS. In addition, users voted on SAS enhancement priorities and gave many suggestions to the SAS staff.

Conference proceedings are being readied by Ken Offord of the Mayo Clinic, Program Chairman. They will be published by SAS Institute and a copy sent to each attendee. Additional copies will be available for $10.00 each.

New SUGI officers were chosen in New Orleans: co-chairmen for the coming year are Dr. Michael Farrell of Miami University in Ohio and Dr. Rodney Strand of Oak Ridge National Laboratory in Tennessee.

A complete report of the software ballot results will appear in the conference proceedings. To summarize these results, the top items are listed below:

- SAS data sets on tape
- DO loops; DO groups
- a histogram procedure
- assignment statements of the form
  `X1-X9=LOG(Y1-Y9);`
- more output data sets from procedures
- procedures for non-parametric statistics
- subscripted variables
- character-numeric conversions
- organization of supplementary procedures and distribution of documentation
- IF-THEN/ELSE statements
- contrasts and means in GLM.

Plans for next year's SUGI conference, which will include tutorials on using SAS, are underway, and Las Vegas is the tentative site. The April issue of **SAS Communications** will contain more information, as well as the call for papers.

SUGI Corner

SUGI Co-chairman Rod Strand reports that a new column, SUGI Corner, will begin in the April issue of **SAS Communications.** Editor of the SUGI Corner will be Helene Cavior of the Federal Bureau of Prisons in Pleasanton, California.

Contributions of SAS users to the SUGI Corner are welcomed. Dr. Strand suggests the following as examples of contributions:

- short notes on SAS
- "darn cute code"
- correspondence of interest to SUGI members
- reports on projects in which SAS was particularly useful
- ways in which SAS is used for various applications.

Please send your SUGI Corner contributions to:

SUGI Corner
c/o SAS Institute Inc.
Post Office Box 10066
Raleigh, North Carolina 27605

They will be forwarded to the SUGI Corner editor.

SAS Plans For The Future

At the SUGI meeting in New Orleans, the SAS staff presented timetables showing anticipated SAS developments during the next six months. Our next major goal is the release of SAS 76.5 in early spring. New features for SAS 76.5 include:

- PROC PLOT, a comprehensive plotting procedure that replaces SCATTER
- MEANS and WEIGHT statements in PROC GLM
- a facility for storing SAS data sets on tape
- enhancements to PROC MATRIX
- new formats and functions for time and date
- PROC NPAR1WAY, a new procedure for one-way rank tests
- **SAS Supplementary Library,** a short manual containing interim documentation of the above features and descriptions of supplementary procedures.

continued on page 2

continued from page 1

Work scheduled at SAS Institute for the spring and summer includes:

- a new histogram procedure
- adjusted means, contrasts, and expected mean squares in GLM
- enhancements to MACRO capabilities
- an internal editor for SAS under TSO
- a categorical linear models procedures
- improvements to the FREQ procedure
- a **SAS Primer** to help new users learn SAS.

System Notes

Syntax Checking with OBS=0

What happens when you specify

```
OPTIONS OBS=0;
```

for syntax checking, or when an error occurs and SAS sets OBS to zero?

SAS continues through the SAS statements in the job, actually executing each DATA and PROC statement. Since the number of observations to be processed is zero, procedures produce a message noting that the input data set contains zero observations. Procedures and DATA statements create any requested SAS data sets, although these data sets will contain zero observations. Since data sets are created as requested in PROC and DATA statements, SAS continues through the job, not stopping because a specified data set does not exist.

Thus, although no observations are actually processed, the syntax of the SAS statements is checked and any syntax errors are caught.

Because SAS does create data sets as requested in the job, using statements of the form

```
DATA OLD.X;
SET OLD.X;
```

to "update in place" is dangerous. If OBS=0 were in effect when these statements were executed, the data set OLD.X would be recreated with zero observations. The original contents of OLD.X would be lost. Using statements of this form is therefore not recommended when the original data cannot be easily regenerated.

The LENGTH Statement

Although not explicitly stated in the **User's Guide**, the LENGTH statement is only effective when it appears **before** the first definition of a variable. Thus, it should appear before an INPUT statement referencing variables listed in the LENGTH statement. For data sets created with SET, MERGE, or UPDATE, the LENGTH statement should appear before the SET, MERGE, or UPDATE statement.

Staff Changes

Kathy Fulp has moved with her husband to Wichita, Kansas. Our new administrative assistant is Tressa Gates.

Recoding

Lawrence Muhlbaier of Duke University has suggested a useful mechanism for recoding SAS variables. This method relies on the fact that logical expressions are evaluated as either 1 (true) or 0 (false). For example, the logical expression

```
A < B
```

has a value of 1 if A is less than B; 0 otherwise.

Consider the following situation. A data set contains a variable called AGE, whose values range from 0 up. We want to create a new variable, AGEGROUP, whose value will be 1 if the subject's age is under 25; 2 if the age is between 26 and 75; and 3 if the age is over 75.

This statement accomplishes the recode:

```
AGEGROUP = 1*(AGE<=25) + 2*(26<=AGE<=75) + 3*(AGE> 75) + AGE*(AGE=.);
```

Note that the first three logical expressions define the three groups. When an AGE value falls into, say, the third group, the logical expression AGE>75 will be true and will have a value of 1. This value is then multiplied by 3. Since the other logical expressions all have a value of 0 when AGE is greater than 75, the AGEGROUP value will be 3.

The last expression AGE*(AGE=.) assures that a missing AGE value results in a missing AGEGROUP value. When the logical expression is true, multiplying it by AGE (a missing value in this case) produces another missing value.

An alternate method:

```
AGEGROUP= 1 + (AGE > 25) + (AGE > 75) + AGE*(AGE=.);
```

The four statements below could also be used to produce the same result:

```
IF AGE <=25 THEN AGEGROUP=1;
IF 26 <=AGE <=75 THEN AGEGROUP=2;
IF AGE > 75 THEN AGEGROUP=3;
IF AGE=. THEN AGEGROUP=.;
```

Testing Bits

Testing bit values in SAS can be accomplished with two extra steps. The first step shifts the bit in question to the rightmost position, and the second isolates it for testing.

For example, to test bit 5 (where bits are numbered 0 to 7) of a one-byte numeric variable I, use these SAS statements:

```
INPUT I PIB1;
J=INT(I/4);          * SHIFT BIT 2 POSITIONS RIGHT;
BIT=MOD(J,2);        * ISOLATE BIT;
IF BIT THEN...;      * TEST BIT;
```

Note that the value of the constant in the second statement above depends on the bit to be tested. Use the value 2^n, where n is the number of positions to shift the bit. Above, the constant is 4 since the bit is being shifted 2 positions. If we were testing bit 4, we would use the constant 2^3=8 in the second statement.

More About Hexadecimal Formats

The numeric format item HEX is useful for converting decimal numbers to hexadecimal numbers, and vice versa. For example, the SAS statements

```
DATA;
INPUT D 2. X HEX2.;
PUT D= 2. 'IN DECIMAL'   @20 D= HEX2.  'IN HEX' /
    X= HEX2. 'IN HEX'    @20 X= 3.  'IN DECIMAL' /;
CARDS;
11 A1
16 16
```

produce this output:

```
D=11 IN DECIMAL  D=0B IN HEX
X=A1 IN HEX      X=161 IN DECIMAL
D=16 IN DECIMAL  D=10 IN HEX
X=16 IN HEX      X=22 IN DECIMAL
```

Similarly, the character format item $HEX converts character values to their hexadecimal representations. When you use $HEX, remember that the hexadecimal representation of a single character takes two columns to display. Thus, for output, the width specified should be twice the number of characters in the variable. For example, the SAS statements

```
DATA;
INPUT A $1. B $2.;
PUT A=$1.  @15 A= $HEX2.  'IN HEX' /
    B=$2.  @15 B= $HEX4.  'IN HEX' //;
CARDS;
P TO
9 22
```

produce this output:

```
A=P      A=D7 IN HEX
A=TO     B=E3D6 IN HEX

A=9      A=F9 IN HEX
B=22     B=F2F2 IN HEX
```

The HEX format is often useful for handling systems data. For example, the system completion code (SCC) is stored in SMF records as a two-byte binary field, although it is usually printed in its hex representation. The SAS statements

```
INPUT SCC PIB2.;
FORMAT SCC HEX3.;
```

read the SCC and assign it a hex format, so that any subsequent printing of the SCC will use the hex representation.

Kruskal-Wallis and Friedman Tests in SAS

You may not be aware that many non-parametric tests can be performed using PROC RANK with PROC ANOVA or PROC GLM. Frank Harrell and Bill Gjertsen of the Lipids Program at UNC-Chapel Hill have pointed out several such uses.

The Kruskal-Wallis test is a one-way analysis of variance on ranks. If you have a balanced one-way layout, with Y as the response variable and TMT as the classification variable, run these SAS statements:

```
PROC RANK; VARIABLES Y; RANKS YR;
PROC ANOVA; CLASSES TMT; MODEL YR=TMT;
```

Then compute

```
H=SST*12/(N*(N+1))
```

where SST is the sum of squares for TMT in ANOVA and N is the number of observations. Critical values or significance probabilities for H, the Kruskal-Wallis statistic, can be found in tables, or a chi-square approximate value can be used.

The Friedman test is a two-way analysis of variance on ranks. If you have a two-way layout of blocks (BLOCK) and treatments (TMT) with one observation per cell and Y the response variable, run these SAS statements:

```
PROC SORT; BY BLOCK;
PROC RANK; BY BLOCK; VAR Y; RANKS YR;
PROC ANOVA; CLASSES BLOCK TMT;
MODEL YR=BLOCK TMT;
```

Then compute

```
F=SST*12/(T*(T+1))
```

where SST is the sum of squares in ANOVA for TMT, and T is the number of treatments. Friedman's statistic F is tabled; it is approximately chi-square with T-1 degrees of freedom.

Converting to a chi-square test above is not absolutely necessary: the F tests that PROC ANOVA produces are asymptotically valid, and may be reasonable even in fairly small samples.

SAS Publication News

The **SAS Programmer's Guide**, containing much new material and a new organization, is now available from SAS Institute. Single copies are $6.95 plus $.25 postage.

The **Programmer's Guide** gives directions for writing SAS procedures using FORTRAN or PL/I, as well as brief descriptions of the macros and subroutines used in SAS. Note that adding new SAS procedures does **not** mean maintaining a separate version of SAS on disk; you need only concatenate your load library to the main SAS library.

Also available is a brochure on SAS called "SAS: A New Concept in Statistical Software." Universities and service bureaus have found this booklet useful for describing SAS to prospective users. We are happy to send out single copies on request. In quantity, we make the brochure available at our cost of $.35 per copy.

We are beginning work on the **SAS Primer**, a new publication that will introduce new users to SAS and get them started. We expect to have a draft available for review in early May, with publication scheduled for summer. Please send us any handouts you have used to introduce SAS to new users if you think these handouts would be helpful to us in writing the **Primer**.

continued on page 4

continued from page 3

SAS Supplementary Library should be available in April. This guide will include descriptions of the supplementary procedures we now have, as well as documentation of the new SAS 76.5 features. If you have any locally written SAS procedures that you would like to include in the SAS Supplementary Library, we encourage you to submit them to us, along with complete documentation and sample output.

SAS Makes Honor Roll

DATAPRO's annual survey of software users showed that SAS was among the top 38 software products, placing on the DATAPRO Honor Roll. This was our first mention among the DATAPRO results, and we are gratified that SAS entered the competition at a high level.

SMF Samples Available

For some time Barry Merrill, formerly with State Farm and now with Suntech, has been making available to other SAS users a tape containing his SAS programs to process SMF (System Management Facilities) data.

SAS Institute has taken over distribution of this SMF tape, and copies are available at a charge of $25.00 to cover tape and copying costs. Let us know if you would like a copy of the tape.

Note that neither Barry Merrill nor SAS Institute assumes any responsibility for the programs on the tape; they are presented as examples that will probably be helpful to people doing SMF analysis.

SAS Communications is published quarterly by SAS Institute Inc.

Anthony J. Barr, Systems
James H. Goodnight, Procedures
John P. Sall, Procedures
Jane T. Helwig, Communications

Joyce P. Massengill, Administration
Ann W. Baggett, Publications
Tressa A. Gates, Administrative Assistant
Debra L. Calloway, Assistant

Address all correspondence to **SAS Institute Inc.,** Post Office Box 10066, Raleigh, NC 27605.

SAS INSTITUTE INC.
Post Office Box 10066
Raleigh, North Carolina 27605

MR. RAYVON STURDIVANT
COMPUTATION CENTER
PHILLIPS HALL
UNIV. OF NORTH CAROLINA
CHAPEL HILL, NC 27514

SAS communications

Vol. II, No. 4 Spring 1977

Take A SAS Short Course

The SAS short course is being offered in two locations this year. This course is designed for data analysts with some statistical training who want to learn more about using SAS for practical applications, and includes:

- basic steps in SAS jobs
- basic SAS procedures
- the statistical model procedures (GLM, ANOVA, NLIN, SYSREG)
- matrix applications (MATRIX)
- data management within SAS
- writing custom-formatted reports
- new features of SAS 76.5.

One course will be given August 31-September 1 in New Brunswick, N.J.; the other September 26-27 in Chicago. Jim Goodnight, John Sall, and Bill Gjertsen will teach the courses, and materials will be provided. Course registration fee is $295.

To register, and for more information, call or write

SAS Institute Inc.
P.O. Box 10066
Raleigh, N.C. 27605
(919) 834-4381

SAS 76.5

Finishing touches are being applied to SAS 76.5, and a tape containing this new version will be sent to all current installations as soon as these final stages are complete.

A copy of **SAS Supplemental Library User's Guide** will be included with each tape. This manual contains documentation of eight user-contributed procedures, nine SAS procedures, and the new features in SAS 76.5. Additional copies will be available from SAS Institute at $6.95 plus $.25 postage.

SUGI Conference Plans

Next year's SAS Users Group International (SUGI) conference will be held January 31-February 2 in Las Vegas, Nevada, at Caesar's Palace. Conference co-chairmen Mike Farrell and Rodney Strand report that plans for the conference are well underway. The summer issue of SAS **Communications** will contain more information about the format of the conference, and will include a call for papers in such areas as teaching applications, SMF analysis, general linear models, research data management, and medical applications.

SUGI Proceedings at Printer

The **Proceedings** of the second SUGI meeting held in New Orleans in January 1977 are now at the printer, and a copy will be sent to each conference attendee in June. Additional copies may be ordered from SAS Institute for $10.00 each plus $.35 postage.

PL/I Optimizer Version of SAS

The SAS 76.5 tape will include two separate versions of the main SAS library. In one version, the PL/I parts of SAS are compiled with the PL/I (F) compiler; in the other, with the PL/I Optimizing Compiler. The PL/I source code, which makes up about half of SAS's 80,000 source statements, is the same for both libraries.

Demand for an Optimizer-compiled SAS has been strong for several years. However, such a version requires that the installation have the PL/I Resident and Transient Libraries available, and these must be leased from IBM. We did not want to make the presence of these libraries a prerequisite for SAS.

Distributing both versions on the SAS tape gives those installations that have the libraries the ability to run the optimized version, while other installations may continue to run the PL/I (F) version.

More New Features In SAS 76.5

In addition to the new features described in the January issue of **SAS Communications** (PROC PLOT; MEANS and WEIGHT statements in PROC GLM; tape data sets; time and date formats), SAS 76.5 will include some additional capabilities.

PROC FORMAT is a new procedure that lets you define formats to print value labels with SAS procedure output and PUT statements.

PROC PDSCOPY is a valuable addition to the utility procedures in SAS. PDSCOPY copies load modules and libraries from disk to disk, or from disk to tape, reblocking them if desired.

PDSCOPY was developed to make distribution of SAS easier. With PDSCOPY, a very large, efficient blocksize can be used on the SAS distribution tape, and each installation can then reblock the libraries to an optimal size for its direct access devices.

In testing PDSCOPY, we discovered that its blocking algorithm is much more efficient than the linkage editor's. **This efficiency results in a savings of around 15% in the track space used by load module libraries.**

For example, the main SAS 76.4 library takes 330 tracks on a 3330 disk. After copying with PDSCOPY using the NE option, the library required only 237 tracks. That's a savings of 28%. Trimming the library as suggested in the installation instructions would reduce it even further to 203 tracks, a savings of 36% in track space!

Treatments vs. Control

Suppose that one of the treatment levels of an effect is a control group; that is, no treatment. You may want to test this control group against the other treatments, then later test among the other treatments.

This analysis can be set up for GLM by providing a special control vs. treatments indicator variable, and nesting treatments in this variable. For example:

```
DATA;
INPUT Y TMT;
CONTROL=0;
IF TMT=0 THEN CONTROL=1;
CARDS;
. data lines
.
PROC GLM;
CLASSES CONTROL TMT;
MODEL Y = CONTROL  TMT(CONTROL);
```

This technique also applies to a second crossed factor, such as "method of application," which is not meaningful for the control group. For example:

```
PROC GLM;
CLASSES CONTROL TMT METHOD;
MODEL Y = CONTROL  TMT(CONTROL)  METHOD(CONTROL)
          TMT*METHOD(CONTROL);
```

The user can verify that these statements produce the expected tests by trying them with the E3 or E4 option in the MODEL statement.

Rumors

We hear that GLM runs much faster these days...that MATRIX handles character variables...that you can do Duncan's multiple range tests in GLM...that HARVEY and EXPLODE run under SAS76...that GLM prints the X'X and X'X inverse matrices...that you can print exact inverses of matrices in PROC MATRIX... that Caroline Grayson Helwig arrived April 29...that Gary Nemeth's **Structured Programming in BAL** uses SAS...that SAS runs on a CDC Omega?

Contrasts By Transformation For Balanced Data

For testing a set of orthogonal contrasts, it is easy to set up your own regressors:

```
TEST1=(TMT=1) - (TMT¬=1)/3;
TEST2=.5*(TMT=2 | TMT=3) - (TMT=4);
TEST3=(TMT=2) - (TMT=3);
```

The t-tests on the parameters provide the contrast results.

For non-orthogonal contrasts, be sure to "invert" before using them to transform the data. Consider the reparameterization

$$E\mathbf{y} = X\beta = XM^{-1}M\beta = Z\delta$$

The reparameterization $\delta = M\beta$ is done by transforming the data:

$$Z = XM^{-1}$$

(If M is orthonormal, $M^{-1} = M'$, so this extra step is not necessary.)

Consider

$$M\beta = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 \\ 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}$$

$$M^{-1} = 1/12 \begin{pmatrix} 3 & 3 & 6 & 0 \\ 3 & -1 & -2 & 8 \\ 3 & 7 & -10 & -8 \\ 3 & -9 & 6 & 0 \end{pmatrix}$$

So a suitable set of dummy variables would be

```
TEST1 = 3*(TMT=1) - (TMT=2) + 7*(TMT=3) - 9*(TMT=4);
TEST2 = 6*(TMT=1) - 2*(TMT=2) - 10*(TMT=3) + 6*(TMT=4);
TEST3 = 8*(TMT=2) - 8*(TMT=3);
```

The t-tests of the parameters do the corresponding contrasts:

TEST1 does 1 vs. 4
TEST2 does 1 vs. 2 and 3
TEST3 does 2 vs. 3 and 4

Problems In SAS 76.2 and 76.4

The problems below exist in versions 76.2 and 76.4 of SAS. They have all been fixed for SAS 76.5.

- CORR: when the NOMISS option is used, the procedure cannot distinguish between zeros and missing values.
- AUTOREG, BMDP: BY variables are not handled correctly.
- SYSREG: in the third stage, the standard errors are off by a factor of $((N-K)/N)^2$; the RESTRICT constant's sign might be reversed; and occasional allocation problems occur when there are large BLOCKs and few MODELs.
- DUNCAN: do not use both the WALLER and DUNCAN options in the same run.
- MATRIX: do not use the GINV and SVD commands, since a poor choice of tolerance reduces the accuracy of their results.

Programmer's Guide Corrections

These errors in the **SAS Programmer's Guide** were corrected for the second printing:

- on page 20, the last line should be deleted
- on page 45, the two CALLs to the ONAMES routine should read

```
CALL ONAMES(NTYPE,OUTVEC(1)); and
CALL ONAMES(NTYPE,OUTVEC (1),OBSNO);
```

- on page 66, INPUT subroutine, the IEND parameter is FIXED BIN(31) or INTEGER*4
- on page 75, OPENL routine, variable TYPE should be initialized to 'O ' (letter O followed by 7 blanks)
- on page 78, POINT subroutine, the IOBS parameter is **not** returned

The errors below will be corrected for the third printing:

- page 71, NAMEV routine, the NTYPE parameter is FIXED BIN (15) or INTEGER*2
- page 101, PRT macro, the MVC statement should read

```
MVC SCC+16(8),NAME
```

Another problem in the **Programmer's Guide** concerns the SASFT, SASFHX, and SASFG1 routines, which establish the environment for SAS procedures written in FORTRAN. Starting with SAS 76.5, use only the SASFT routine. It has been modified to handle the FORTRAN G1 and H Extended compilers as well as FORTRAN G and H.

Error Term for Latin Square Whole Plots

Consider a split plot experiment where the whole plot is arranged in a Latin Square. It is easy to find the whole plot error by hand—but how do you specify the error term as an effect for a TEST statement in SAS?

Consider the keyout:

Whole Plot: **ROW**
COL
VARIETY
ROW*COL – VARIETY

Split Plot: **TMT**
TMT*ROW
TMT*COL
TMT*VARIETY
residual

The effect "ROW*COL-VARIETY" expresses the idea of VARIETY as a portion of the ROW*COL interaction, with error being the remainder of the interaction. But there is no minus (—) operator in SAS ANOVA.

Consider the term ROW*COL*VARIETY. ANOVA computes SS by obtaining the row SS, then subtracting out the "contained" SS's. The partition for the ROW*COL*VARIETY is the same as that for ROW*COL, and thus the raw SS is the same. But the VARIETY SS is subtracted from ROW*COL*VARIETY; it acts as ROW*COL — VARIETY, the term we want. Consequently, we use ROW*COL*VARIETY in our model and test with:

```
TEST H=ROW COL VARIETY E=ROW*COL*VARIETY;
```

Staff Additions

A recent addition to the SAS Institute staff is William R. Gjertsen. Bill's current responsibilities include consulting and education; in addition, he is handling much of our marketing effort.

Bill comes to SAS from the Lipids Program of the Department of Biostatistics at UNC-Chapel Hill, where he worked as a statistician and SAS programmer. He received his undergraduate degree from Renssalaer Polytechnic Institute and an M.S. from Carnegie Tech, both in electrical engineering. After teaching mathematics for two years in Turkey with the Peace Corps, he earned an M.A. in math from Duke. Bill then began work at UNC-Chapel Hill toward his Ph.D. in biostatistics, which he expects to complete within the next year.

Susan King has also joined the staff as an administrative assistant.

SAS communications

Vol. III, No.1 Summer 1977

SUGI Call for Papers

From Rod Strand, SUGI Co-Chairman

The third annual conference of the SAS Users Group, International (SUGI) will be held January 30, 31 and February 1 at Caesar's Palace in Las Vegas, Nevada.

The upcoming conference has been organized into six sessions: 1) Linear Model Applications, 2) Medical Applications, 3) Statistical Applications, 4) Research Data Management, 5) Business Applications, and 6) Teaching Applications. Each session will be chaired by a specific individual who will be arranging for invited speakers as well as serving as the receiver of contributed papers.

Individuals interested in presenting a paper should submit an abstract, not exceeding 200 words, to the appropriate session chairman listed below by October 15, 1977. Do not submit the same contributed paper to more than one Session Chairman; if not considered suitable for that Session, it will be automatically forwarded to the appropriate Session. Presentation of contributed papers will not exceed 20 minutes, including discussion. Please include a complete address and phone number of the presenter with the abstract. Letters of acceptance will be sent, if requested by the presenter at the time of submission of the abstract.

SESSION	CHAIRMAN
I. LINEAR MODEL APPLICATIONS	Ramon C. Littell Department of Statistics University of Florida Gainesville, Florida 32611
II. MEDICAL APPLICATIONS	Ken Offord Mayo Clinic Medical Research Statistics 200 First Street, S.W. Rochester, Minnesota 55901 Phone (507) 282-2511, ext. 2180
III. STATISTICAL APPLICATIONS	Robert Anderson Deering Milliken Service Corp. P.O. Box 1926 Spartanburg, South Carolina 29304 Phone (803) 573-2020
IV. RESEARCH DATA MANAGEMENT	Wendell Smith Department of Biostatistics School of Public Health University of North Carolina Chapel Hill, North Carolina 27514 Phone (919) 933-6971
V. BUSINESS APPLICATIONS (e.g., SMF, marketing, accounting)	Terry Flynn Occidental Petroleum 5 Greenway Plaza E. BI Level Houston, Texas 77046 Phone (713) 629-3268
VI. TEACHING APPLICATIONS	Rudolph J. Freund Institute of Statistics Texas A&M University College Station, Texas 77843 Phone (713) 845-3141

New FREQ Statistics

On page 35 of the **Supplemental Library User's Guide**, the ALL option of the TABLES statement for PROC FREQ is listed as producing "all additional undocumented statistics."

These statistics, which are produced for two-way tables, are:

- chi-square
- phi
- Cramer's V
- likelihood ratio
- gamma
- Kendall's tau-b
- Stuart's tau-c
- contingency coefficient
- Somer's D
- product moment correlation
- Spearman correlation
- lambda, asymmetric and symmetric
- uncertainty coefficient.

Supplementary Procedures

If you are working on a SAS procedure, you may want to share it with other SAS users by sending SAS Institute a copy. To find out the procedure for sending in a supplemental procedure, write us for a copy of **Contributing SAS Procedures**, which details the necessary steps.

SAS 76.5 Notes

By now, all current SAS customers have received SAS 76.5 installation packages. After SAS 76.5 is installed, you will be able to take advantage of many new SAS abilities. Among these are:

- SAS data sets on tape
- PLOT, a new plotting procedure
- MEANS and WEIGHT statements in GLM
- the FORMAT procedure, to define value labels
- the PDSCOPY procedure, to copy and reblock load module libraries
- functions and formats for date and time values.

All the features above are described in the **SAS Supplemental Library User's Guide**. One copy was included in each installation package, and you can order additional copies from:

SAS Institute Inc.
Post Office Box 10066
Raleigh, North Carolina 27605

The cost is $6.95 each; postage for one copy is .41.

When you start using SAS 76.5, it is important to keep in mind this fact:

ONCE YOU USE SAS 76.5 TO CREATE A DATA SET OR TO READ A SAS 76.4 OR 76.2 DATA SET, YOU MUST CONTINUE USING SAS 76.5 WITH THAT DATA SET.

In other words, you can read SAS 76.4 or SAS 76.2 data sets with SAS 76.5. However, you can't access SAS 76.5 data sets with SAS 76.4 or 76.2.

We have found these problems in SAS 76.5:

- The new functions DATETIME and TIME are not working properly.
- The NOMISS option in PROC CORR should not be used.

PROC FORMAT Correction

In the PROC FORMAT description on page 66 of the **Supplemental Library User's Guide**, users are cautioned to make sure that data sets created with user-defined format items always have the format library available when the data set is used. However, instead of the "disastrous errors" promised if the library is not available, SAS abends with a completion code of 806. The message

IEA7031 806-4 jobname stepname MODULE ACCESSED format_name

appears on the Job Console Log of the job. The data set containing the variable with the user-defined format is not changed, and the job will work properly when the format library is added to the JCL.

Reduction Operators in MATRIX

Suppose you want column sums of a matrix. Before now, you had two choices:

- premultiply by a vector of 1's, then use the ROWSUM function
- transpose the matrix, then use the ROWSUM function.

Neither option is very elegant.

APL has a generalized reduction operator that works with a dyadic operator. Now, the SAS MATRIX procedure also has reduction operators that work in the framework of the subscript syntax.

Recall that subscripting in MATRIX can be used to refer to a submatrix or a single value in a matrix. For example, the expression X(2,3) refers to the element in the second row, third column of the matrix X. The expression X(*,1) (or X(,1)) refers to the first column of X. The expression X(2 3,5 6) refers to the submatrix formed by the intersection of the second and third rows of X with the fifth and sixth columns.

Instead of using subscripts to ask for certain rows or columns, you can use reduction operators to ask for a reduction across all rows and/or columns. For example, if you wanted column sums (summing the row elements, thereby reducing the row dimension to 1), you could specify X(+,). The elements in each column will be added and the new matrix will consist of one row containing the column sums.

Note that the second subscript was omitted, and its place marked by a comma. The column dimension was therefore not changed. The first subscript, +, means that summation reduction will take place across the rows.

MATRIX now allows eight operators for subscript reduction:

+ addition
 # multiplication
<> maximum
>< minimum
<:> index of maximum
>:< index of minimum
. mean
sum of squares

These operators can be used to reduce either rows or columns or both. When both rows are reduced, the row reduction is done first.

For example, the expression A(+,<>) is the maximum of the column sums. To get the sum of the row maxima, you would say A(,<>) (+,).

You can combine subscript reduction with regular subscripts; for example, A(2 3,+) first selects the second and third rows of X, then finds the row sums of that submatrix.

Limit on MACROs

In a single SAS job, no more than 40 MACROs with different names may be defined. It is possible to use the same MACRO name twice, however. The most recent definition of a MACRO is the one used.

Options and TSO

If you use SAS under TSO, you can specify SAS options and parameters for your TSO session with the OPTIONS keyword of the SAS command. For example, to specify the NOCENTER option, you would use the command

```
SAS OPTIONS(NOCENTER)
```

If you want to specify more than one option or parameter, the list of options and parameter must be enclosed in single quotes:

```
SAS OPTIONS('NOCENTER PS=24')
```

This form, instead of the one given on page 286 of **A User's Guide to SAS 76**, is necessary because of TSO restrictions on strings passed to programs.

BY-Variables

The section on page 31 of **A User's Guide to SAS 76** describing "Two or More BY-Variables" states that

> When the value of the first variable given in the BY statement changes, all FIRST. variables are automatically set to 1.

This sentence should be replaced by one that reads

> When the value of any variable given in the BY statement changes, all FIRST. variables corresponding to BY-variables that appear to the right of the changed variable in the BY statement are automatically set to 1.

In other words, FIRST. and LAST. variables are set to 1 when the value of the corresponding BY-variable changes, or **when any of the variables appearing before it in the BY statement change.**

New Character-Handling Features

With SAS 76.5, it is now possible to perform such character-handling tasks as substringing, concatenation, indexing, and numeric-to-character conversions. A new SAS statement, CALL, is used to call the routines that perform these operations, which are documented below.

As currently implemented, only predefined variables may appear as arguments to these functions. A better implementation of character-handling operations is planned for the future.

substring

```
CALL SUBSTR(C1, C2, P, L);
```

where C1 and C2 are character variables
P and L are numeric variables

Like the PL/I function SUBSTR, the SUBSTR routine moves characters from one value to another. A string of L characters, beginning in position P of variable C2, is moved to variable C1. If the string being moved is shorter than C1, blanks are added on the right. L should not be longer than C1, and P+L−1 must be less than the length of C2.

For example, the SAS statements

```
DATA;
C2='BLACKJACK';
C1='          ';
N6=6;
N4=4;
CALL SUBSTR(C1,C2,N6,N4);
PUT C1;
```

produce the line

```
JACK
```

reverse substring

```
CALL STRSUB(C1, C2, P, L);
```

where C1 and C2 are character values
P and L are numeric values

Like the PL/I pseudovariable SUBSTR, the CALL STRSUB routine moves characters from one value into another. A string of L characters, beginning in position 1 of variable C2, is moved into variable C1, starting at position P. If the string being moved is shorter than C1, characters to the right of P+L in C1 are not changed. L must be less than or equal to C2's length and P+L-1 must be less than the length of C1.

SUBSTR and STRSUB differ in what happens when the string being moved is shorter than C1. With SUBSTR, blanks are added to the string to make it the length of C2, replacing whatever value C1 has. With STRSUB, the L characters of C1 starting in position P are replaced, but other characters are not changed.

For example, the SAS statements

```
DATA;
C1='BLACKJACK';
C2='APPLE';
N1=1;
N5=5;
CALL STRSUB(C1,C2,N1,N5);
PUT C1;
```

produce the line

```
APPLEJACK
```

concatenation

```
CALL CONCAT(C, A, B);
```

where C, A, and B are character variables
C is returned

Like the PL/I concatenation operator ||, the CALL CONCAT routine joins character values so that the last character of one is followed by the first character of the other. The value of A, up to the last blank, is joined to B's value, and the result becomes the value of C. There are no length restrictions. If C's length is shorter than the length of the joined string of characters, the string is truncated on the right to C's length. If C is longer than the joined characters' length, blanks are added on the right.

For example, the SAS statements

```
DATA;
A='BLACK';
B='JACK';
LENGTH C $ 9;
CALL CONCAT(C, A, B);
PUT C;
```

produce the line

```
BLACKJACK
```

indexing

```
CALL INDEX(X, A, B);
```

where X is a returned numeric variable
A and B are character variables

Like the PL/I INDEX function, the CALL INDEX routine searches the value of A for an occurence of the value of B. If the character in B occur in A, the position where they begin is returned in X. If the characters in B do not occur in A, a zero is returned in X.

For example, the SAS statements

```
DATA;
A='BLACKJACK';
B='JACK';
X=0;
CALL INDEX(X, A, B);
PUT X;
```

produce the line

```
6
```

producing formatted values

```
CALL FMT(C, X);
```

where C is usually a character variable
X is usually a numeric variable

The FMT routine provides a capability like PUT SUBSTR in PL/I, and can be used for numeric-to-character conversions. The value of X is formatted with its associated format, and the result becomes the value of C. Formats may be associated with X either with the FORMAT statement, the FORMAT procedure, or by using X in a PUT statement with a format. If X has no associated format, the BEST. format is used. If the formatted length of X is greater than C's length, the value is truncated on the right.

For example, the SAS statements

```
DATA;
FORMAT X 7.;
C='ABCDEFG';
X=123;
CALL FMT(C,X);
```

result in the character variable C having the value '123'.

retrieving formatted values

```
CALL INFMT(X, C);
```

where X is usually a numeric variable
C is usually a character variable

The INFMT routine provides a capability like GET SUBSTR in PL/I, and can be used for character-to-numeric conversion. Using the input format associated with X, C's value is placed in X. If X has no associated input format, and X is numeric, a numeric input format w., where w is C's length, is used. If X is a character variable, a character input format $w. is used, where w is X's length.

For example, the SAS statements

```
DATA;
X=.;
C='456';
CALL INFMT(X,C);
```

result in the numeric variable X having the value 456.

Making It Clear

A misconception about calling SAS72 procedures from SAS has arisen. SAS72 procedures with the same name as SAS76 procedures can be called from SAS76. (In the early release SAS75.2, they could not be.)

If you try to call a SAS72 procedure not in the SAS library, the call will not succeed. Most installations keep only a few SAS72 procedures in the SAS72 library; ANOVA, CANCORR, INBREED, QUESTN, and REGR are usually available.

Another misconception holds that a numeric sort key must have a length of 8 or 4. This is not true. In the past, SYNCSORT was unable to handle SAS variables of other lengths as sort keys, but Whitlow fixed this problem last year.

Finally, contrary to yet another misconception, secondary requests for space may be included in the DD statement that allocates space for SAS data sets.

Rumors

SAS does the relational DBMS operations described by Date in **Datamation**[1]...SAS Institute is moving 50 feet south...Owen Richard Gjertsen arrived August 2...Manson-Barr-Goodnight blackjack paper[2] is in hot demand by Las Vegas-bound SAS users.

Character Comparisons

When you compare two character values in SAS, normally SAS extends the shorter value with blanks to the length of the longer value for the comparison. For example, if the value of the variable MISS is '999', the SAS statement

```
IF MISS='9' THEN DELETE;
```

is equivalent to

```
IF MISS='9  ' THEN DELETE;
```

The observation will not be deleted, since '999' is not equal to '9 '. You can reverse this process by using a colon : after the equals sign in the comparison. SAS will then truncate the longer value to the length of the shorter value for the comparison. For example, if the value of MISS is '999', the SAS statement

```
IF MISS=:'9' THEN DELETE;
```

will compare the first character of MISS and '9'. The observation will be deleted, since the characters are equal.

Note that SAS truncates and extends values only during the comparison. The values themselves keep their lengths.

Saving Work Data Sets

If you use SAS interactively under TSO, you might want to end a TSO session and yet be able to pick up your SAS processing at another time, with all your work data sets intact.

You can pick up your processing in this way if you allocate an OS data set with the filename USER before you enter the first SAS command. For example:

```
ALLOCATE FILE(USER) BLOCK(13030) SPACE(10 10) DATASET(SAS.WORKFILE)
```

SAS will then automatically add the first-level name USER (instead of WORK) to all the SAS data sets you create. These data sets will be stored in the OS data set you allocated.

When you are ready to resume processing, allocate the OS data set again with the filename USER, and enter the SASGO command:

```
ALLOCATE FILE(USER) DATASET(SAS.WORKFILE)
SASGO
```

All the data sets that you created in the earlier session are then available to you.

1. Date, C.J., "Relational Data Base Concepts," Datamation, Volume 22, Number 4 (April 1976).
2. Manson, A.R., A.J. Barr, and J.H. Goodnight, "Optimum Zero-Memory Strategy and Exact Probabilities for 4-Deck Blackjack," The American Statistician, Volume 29, Number 2 (May 1975).

Calling Other Programs from SAS

Did you know you could call any OS program from SAS by putting its name in a PROC statement? For example, if you want to call a program of yours named RECORD in the middle of a SAS job, you can use the SAS statement

```
PROC RECORD;
```

Your program will then be executed.

When you are using this feature, keep in mind these facts:

- The SAS statement

  ```
  OPTIONS PROC;
  ```

  must appear before the PROC statement.
- The library containing your program must be concatenated to the STEPLIB DD statement in the SAS cataloged procedure.
- You can check the return code issued by your program by adding a CC parameter, giving the maximum acceptable return code, to your PROC statement:

  ```
  PROC RECORD CC=4;
  ```

  If the return code issued by your program is greater than the CC value, SAS will stop executing your SAS statements and will syntax-check the rest of your job.
- You may want to add the CLEANUP option to the OPTIONS PROC statement, so that your program will be called via the ATTACH macro. Then SAS will not blow up or lose memory if your program malfunctions.

  ```
  OPTIONS PROC CLEANUP;
  ```

Communicating Problems

Several of the university computing center newsletters that come to SAS Institute have listed SAS documentation corrections and system problems. We are pleased that SAS users at universities are being notified of corrections and warned of problems.

However, a few of the problems were news to us as well. Some stemmed from IBM bugs, some were errors corrected months ago, and some reflected misunderstandings. For example, a SAS problem that occurred under one level of IBM's sort was described in an installation's newsletter. The description of this problem was published in several other newsletters, even though the problem did not occur at those sites.

To reduce the chances of such situations, give SAS Institute a call to check on the status of SAS problems that you read about for the first time in other installations' newsletters. Always report any problem that you discover to SAS Institute, so that we can send prompt notification to all our customers.

HEX Listings

You may not realize that SAS can produce a very readable listing of data on tape or disk. If the data includes unprintable characters, SAS prints them in hexadecimal.

For example, you could use the job below to print the first 100 records in a tape file:

```
//jobname JOB acct,programmer
// EXEC SAS
//TAPE  DD DSN=DEPT13.STUDY,VOL=SER=350123,DISP=OLD,
// UNIT=TAPE
DATA;
INFILE TAPE OBS=100;
INPUT;
LIST;
```

For a standard-labeled (SL) tape, no DCB information need be specified, since SAS gets this information from the label. For non-labeled (NL) tapes, give the DCB information either in the JCL or with the BLKSIZE, LRECL, and RECFM parameters of the INFILE statement.

If you do not know the DCB specifications for an NL tape, omit them—SAS assumes BLKSIZE=32767 and RECFM=U. Although the records will not be deblocked, you will get a readable dump of the tape contents.

More SAS Short Courses

Raleigh and San Francisco will be the locations of the next SAS two-day short courses. The course will be offered in Raleigh on December 7 and 8, and the San Francisco course is scheduled for January 26 and 27.

Registration information for these short courses will be sent to everyone on the SAS Communications mailing list. For more information, give SAS Institute a call at (919) 834-4381.

SAS Communications is published quarterly by SAS Institute Inc.

Anthony J. Barr, Systems
James H. Goodnight, Procedures
John P. Sall, Procedures
Jane T. Helwig, Communications
William R. Gjertsen, Consulting
Joyce P. Massengill, Administration
Ann W. Baggett, Publications
Tressa A. Gates, Administrative Assistant
Susan D. King, Administrative Assistant
Ramey G. Powell, Assistant

Address all correspondence to **SAS Institute Inc.,** Post Office Box 10066, Raleigh, NC 27605.

SAS INSTITUTE INC.
Post Office Box 10066
Raleigh, North Carolina 27605

MS. JUDY HALLMAN
COMPUTATION CENTER
PHILLIPS HALL
UNIV. OF NORTH CAROLINA
CHAPEL HILL, NC 27514

Ode to SAS 76.5 by Ogden Crash*

From North Carolina we present to you
An update of SAS 76.2.
This pristine release is better and newer,
Abends and errors and problems are fewer.
Its PL/I is now optimized,
The blocking default is brand-newly sized.
Some options were altered, and some were unchanged,
Internally, SAS is now neatly arranged.
Several procedures, often requested,
Now are available, ready and tested.
PROC PDSCOPY deals with utilities,
It makes former MOVEs naught but futilities.
PROC FORMAT's a new one that lets you provide
Your own value labels at your SAS output's side.
We are all proud of this latest release,
and we hope that our users will find it a breeze.
You're advised to consult the new green SLUG,†
Which lists the new goodies, but nary a bug.
So let's say welcome, we're glad it's arrived,
Long life to SAS 76.5.

Reprinted with permission from *Communications*, the newsletter of the University Computer Center, City University of New York.

*Ogden Crash is Harriet Schabes' pen name.

†SLUG is the acronym for the *SAS Supplemental Library User's Guide*, the documentation for new features in SAS 76.5.

SAS communications

SAS Institute Inc • Post Office Box 10066 • Raleigh, North Carolina 27605 • (919) 834-4381

Fall 1977 • volume III number 2

courses

San Francisco in January

Make a SAS week of it by attending the SAS Short Course January 26-27 in San Francisco before you go to Las Vegas for the SUGI meeting January 29.

The beautiful Hyatt Regency San Francisco is the location for the Short Course. Instructors will be Jim Goodnight, John Sall, and Bill Gjertsen. Topics to be covered include basic SAS concepts, data management, modeling procedures, SAS for data processing, and a potpourri of new SAS 76.5 features.

To register, call SAS Institute at (919) 834-4381.

conference

SAS users to meet in Las Vegas

"A maelstrom of organization" was the way one observer described preparations for the SAS Users Group, International (SUGI) conference to be held January 30-February 1 in Las Vegas.

If you haven't sent in your registration, send it today. Don't forget to send the Caesar's Palace registration card too.

If you didn't receive any information about the conference, give SAS Institute a call and we'll get it to you immediately.

Co-chairmen Rodney Strand and Michael Farrell report that over sixty papers have been accepted for presentation at Las Vegas. These papers are listed below.

Papers accepted for SUGI conference, Las Vegas:

Business Applications

Terry Flynn, Occidental Petroleum Company, Chairman

Invited Papers

H. W. "Barry" Merrill, Suntech, Inc.
Using SAS to Tune MVS

Mario Morino, Morino Associates, Inc.
SAS: The Necessary Utility for Computer Performance Evaluation and Control

Kenneth W. Kolence, Institute for Software Engineering
Using Software Physics to Size Workloads with SAS

Contributed Paper

Maxine Potter, Armco Steel Corporation
History Problems and Practical Applications of SAS in the Large Industrial Organization

Research Data Management

Wendell Smith, University of Carolina at Chapel Hill, Chairman

continued

Invited Papers

Rudolph C. Mendelssohn, U. S. Department of Labor, Bureau of Labor Statistics
The Data Base Approach at the Bureau of Labor Statistics

Frank E. Harrell, Jr., University of North Carolina
Capabilities of SAS as a Clinical Research Data Management and Analysis System

Glen A. Augustine, Mayo Clinic
SAS as a Management System for Routine Echocardiograms

Contributed Papers

Joel Achtenberg and J. Phillip Miller, Washington University
Interfacing MUMPS-based Data Entry System to SAS

Mark R. Schultz, TASC
Pinpointing Data with 'DARTS'

Helene Enid Cavior, Federal Bureau of Prisons, Federal Correctional Institute
Simulating a Hierarchical Master File Structure Using SAS

Max F. Ellis, 3-M Company
A Logical Data Definition Capability in SAS

Jane Beth Markley, UNC-Chapel Hill
A SAS Data File Management System for Data from Kidney Dialysis Patients

Frank J. Potter, Merck & Company
SAS as a User-Maintained Data Base System

J. Phillip Miller, D. Powell and C. R. Cloninger, Washington University
The Use of SAS for the Analysis of Family Studies

Lynn B. Eggering, Washington University
Adherence Monitoring at the Clinic Level with the Use of SAS

R. J. Olson and Rodney H. Strand, Oak Ridge National Labs
Management of Diverse Environmental Data with SAS

Robert Bronstein, U.S.V. Pharmaceutical Corporation
Stability Scheduling Using SAS

Sharon Bucci, TASC
Data Base and Analysis of Power Plant Cooling Systems

E. N. Caldwell, WVNET
SAS's Vital Role in a Statewide Fuel Conversion and Energy Conservation Study

Michael S. Lajiness, T. P. Tesar and P. L. Kemp, Upjohn Company
SAS and the Analysis of Clinical Data at the Upjohn Company

Paul D. Hopkins, Research & Development Branch, Statistical Reporting Service, U. S. Department of Agriculture
PROC CODIN - A Procedure for Coded Data Entry

Peggy Pearsall, USDA-ARS
The KWIKPLOT Procedure

Linear Models

Ramon C. Littell, University of Florida, Chairman

Invited Papers

Walter R. Harvey, Ohio State University (with L. A. Swiger)
Orthogonal Polynomial Fitting with Arbitrary Spacings and Correlated Means

William L. Sanders, University of Tennessee
Analysis of a Repeated Measures Experiment with Incomplete Data

Contributed Papers

Kareem Nakkash and Bradford T. Greene, Richard Katon Associates, Inc.
Computer Program for Ridge Regression

I. K. Hwang, Merck & Company
Using SAS-GLM in Kinetic Modeling

John A. Courtright, Cleveland State University
Repeated Measures in SAS

H. I. Patel, Louisiana State University
SAS Procedure for the Analysis of Covariance with Intra-Class Regression Model in One- and Two-way Classifications

James A. Bolognese, Merck & Company
Adjusted Means from SAS76

Michael Conlon and Jean B. Holzer, University of Florida
Use of the ABSORB Statement in SAS

Medical Applications

Ken Offord, Mayo Clinic, Chairman

Invited Papers

Professor David Hurst, Biostatistics Department, University of Alabama
Some Statistical Problems in Biomedical Data Analysis

Contributed Papers

M. L. Moeschberger and Ernest Hilderbrand, University of Missouri
Use of SAS Procedures in Estimating Survival Curves

Chaio Yeh, ICI
Unsymmetrical Parallel Line Bioassay Using SAS

Michael S. Lajiness, Upjohn Company
Evaluating Drug Reinforcement Using SAS

Edward M. Bosanac, West Virginia Department of Finance & Administration
A SAS-Based Small-Area Data Profile System: Its Use in Primary Care Resource Development

Edward L. Spitznagel, Jr. and W. D. Owens, Washington University
Maintenance and Analysis of Anesthesial Surgery Data with SAS

J. D. Quick, University of Rochester Medical Center; Merwyn R. Greenlick, Kaiser Health Services Research Center
Multivariate Prediction of Neonatal Mortality with Emphasis on Health Care Inputs

Statistical Applications

Robert Anderson, Milliken Service Corporation, Chairman

Invited Papers

Kenneth L. Koonce (co-author: Emilio Icaza), Louisiana State University
Some Applications of the Discriminant Procedure

Kenneth White, Rice University
Applications in Econometrics - Problems and Procedures

Contributed Papers

Robert Parks and M. McBride, Washington University
Principal Component Regression with PROC MATRIX

Robert Rogers, North Central Forest Experiment Station, University of Missouri
A SAS Macro for Calculating Coefficients of Ridge Regression

Paul Slater, West Virginia University
Response Surface Contour Plotting in SAS

C. Bean and James Dunn, University of Arkansas
A SAS Macro for Maximum Likelihood Probit Analysis with Multiple Predictors

J. Phillip Miller, Washington University
Maximum Likelihood Estimation of the Multivariate Logistic

Don Henderson, USDA-ARS
KRUWALC, A Macro for Categorical Data

H. Hill, F. Morgan and Richard Nelson, Clemson University
'Lifetest' Procedure for Analyzing Reliability Data

Don Henderson, USDA-ARS
Macro Commands for Tukey Type Smoothing

Samuel Carmer, University of Illinois
Use of MATRIX Proc for Single Degree of Freedom Contrasts in Factorial Experiments

Jerry Oglesby and Lowell Bahner, University of West Florida
Test of Model for Predicting Kepone Accumulation in Selected Estuarine Species

William Reynolds, University of North Carolina, Chapel Hill
SAS Supplemental Procedure SPSS

Nat H. Wooding, Virginia Electric & Power Company
Improving SAS Procs Without Rewriting SAS

Edward L. Spitznagel, Jr., Washington University
KWIC Indexes with SAS

Teaching Applications

Rudolph J. Freund, Texas A & M, Chairman

continued

Invited Papers

Thomas A. Bubolz (co-author: James E. Gentle), Iowa State University
SAS Applications in a Three-Course Sequence in Statistical Computing

William J. Wilson, University of North Florida
Use of SAS in a Statistical Methodology Course

Robert D. Morrison, Oklahoma State University
Use of SAS in Graduate Level Service Courses in Statistics: A Survey of Practices in the Southern Region

Contributed Papers

Ione Cockrell and Dorsey Glenn, S. C. Commission on Alcohol and Drug Abuse
Interactive Teaching with SAS

Dan Chilko and E. J. Hamer, West Virginia University
Generating Multivariate Normal Data in SAS

technical fall 1977

Help SAS improve diagnostics

Several SAS users have asked for a directory of SAS error messages, including the corrective action for each problem.

We try to include all this information in SAS diagnostic messages. Doing this means that you don't have to spend your time poring over error message manuals.

If you feel that a message is ambiguous or incomplete, we want to know about it. Please send to SAS Institute the computer output for any diagnostic messages that you feel need improvement, along with your comments and suggestions.

SAS clinic

Recently the following SAS statement came across our desk:

```
INPUT P 1-3 PH 4 FIELD1 13   STREAM1 14   TIME1 15
                 FIELD2 16   STREAM2 17   TIME2 18
                 . . . . . .
                 FIELD10 40  STREAM10 41  TIME10 42;
```

This SAS user could have saved himself time by using format lists to read the three sets of variables

FIELD1-FIELD10, STREAM1-STREAM10, and TIME1-TIME10:

```
INPUT P 1-3 PH 4 @13 (FIELD1-FIELD10) (1. +2)
                 @14 (STREAM1-STREAM10) (1. +2)
                 @15 (TIME1-TIME10) (1. +2);
```

What does this INPUT statement do?

- reads in the P and PH variables just as the first INPUT statement did
- reads in all the FIELD variables by starting at column 13, reading FIELD1 with a format of 1., moving the pointer over two columns to 16, reading FIELD2, and so on
- reads the STREAM and TIME variables in the same manner.

Rearranging FREQ tables

When you analyze questionnaire data, it's often useful to see table values from PROC FREQ come out in the same order they had on the questionnaire.

For example, let's say that one item in your questionnaire was:

17. What is your attitude toward overtime work?
    - ☐ hate extra work
    - ☐ willing to work extra
    - ☐ enjoy extra work

You code the answer as A, B, or C, and use PROC FORMAT to assign value labels:

```
PROC FORMAT;
  VALUE $ATT A=HATE
             B=WILLING
             C=ENJOY;
```

Next you print tables with PROC FREQ. Since the values are alphabetized before the table is printed, the answers appear in the order ENJOY, HATE, and WILLING. You want them in the order they had on the questionnaire.

The solution lies in the value label: assign the value A) HATE to the value A, and so on:

```
PROC FORMAT;
  VALUE $ATT A=A) HATE
             B=B) WILLING
             C=C) ENJOY;
```

Now the values will be alphabetized just as you want them, and the frequency tables can be compared directly to the questionnaire.

Are you missing any decimal places?

How column input really works

Veteran SAS users are familiar with both column input

```
INPUT X 10-12;
```

and formatted input

```
INPUT @10 X 3.;
```

In fact, the effects of these two statements are identical.

When SAS encounters an INPUT statement containing column input, it translates column to formatted input:

- the first column number becomes the pointer direction @'s value
  ```
  @10
  ```
- the second column value minus the first column plus 1 becomes the width for the numeric format w.
  ```
  3.
  ```
- when a $ appears,
  ```
  INPUT A $ 10-15;
  ```
- SAS uses the $w. format
  ```
  INPUT @10 A $6.;
  ```
- if a decimal specification appears,
  ```
  INPUT X 23-27 2;
  ```
  the decimal specification becomes the d value in the numeric format w.d
  ```
  INPUT @23 X 5.2;
  ```

This situation affects SAS users most often when they use PROC PRINT or PROC FREQ to print numeric values containing decimal places. Data values that include decimal points are always read correctly. However, if the INPUT statement did not include a decimal specification for the variable, the variable's implied format (see p. 49, *SAS User's Guide*) also does not include a decimal part. Consequently, when PROC PRINT uses the implied format to print the values, the decimal part is not printed.

This problem can be solved by using either a decimal specification to read the data

```
INPUT X 10-13 1;
```

or a FORMAT statement after the INPUT statement

```
FORMAT X 4.1;
```

Exact-size random sampling

Researchers often need a random sample of the observations in a data set. When the precise number of observations in the sample isn't important, these SAS statements serve:

```
DATA SAMPLE; SET FULL;
  IF UNIFORM(0)>.2 THEN DELETE;
```

For each observation, the UNIFORM function produces a random number between 0 and 1. This is a 20% sample, and the random number's value will fall between 0 and .2 for about 20% of the observations. These observations will be included in the sample data set.

The problem with this method is that you can't specify the exact number of observations for the sample data set. You can only say that the number will be **about** 20%.

For an exact-sized random sample, these SAS statements can be used:

```
DATA SAMPLE; SET FULL;
    RETAIN NOBS 101 NSAMP 20;
    NOBS=NOBS-1;
    IF UNIFORM(0)<NSAMP/NOBS;
    NSAMP=NSAMP - 1;
```

This method requires that you know the exact number of observations in your data set. You must specify this number, plus 1, in the RETAIN statement. The number observations you want in the sample data set must also b specified in the RETAIN statement. Counters are decremented each time an observation is read and each time an observation is selected for the sample. The probability of selection will thus be equal for all observations, and an exact number of observations will be selected.

Note that the subsetting IF statement causes SAS to return for another observation when the current observation isn't selected; NSAMP is not decremented in this case.

Solving eigenvalue systems

Several callers have recently asked us about getting eigenvalues of nonsymmetric matrices. These users want to solve the eigenvalue problem

$$(A - \lambda B)x = 0$$

where A and B are positive-definite symmetric, with λ and x to be determined. The obvious next step is multiplication by B^{-1} to get

$$(B^{-1}A - \lambda I)x = 0$$

However, now you have a nonsymmetric matrix $B^{-1}A$. You can reduce the problem to a symmetric one with a trick that has been around a long time but isn't obvious.

Let U be the upper Choleski root (see the MATRIX function HALF, *SAS User's Guide*, p. 165) such that U'U=B. Then

$$(A - \lambda B)x = 0$$

Let B = LU, where

```
U = HALF(B) and L=U'
```

Therefore

$$(A - \lambda LU)x = 0$$
$$(L^{-1}A - \lambda U)x = 0 \quad \text{multiply by } L^{-1}$$
$$(L^{-1}AU^{-1} - \lambda I)Ux = 0 \quad \text{factor out U}$$

Note that the roots of $L^{-1}AU^{-1}$ are the same as the roots of $B^{-1}A$, and that the characteristic vectors of $B^{-1}A$ are U^{-1} times the characterictic vectors of $L^{-1}AU^{-1}$.

The MATRIX statements that correspond to these operations are:

```
UI=INV(HALF(B));
W=UI'*A*UI;
EIGEN LAMBDA X W;
X=UI*X;
```

CALL documentation

Jeanne

It's possible to write your own routines that process observations in a SAS program and then invoke the routines with a CALL statement. The string-handling routines, described in the Summer 1977 issue of *SAS Communications*, work this way.

For documentation of the CALL facility, write or call SAS Institute.

inside SAS

staff additions

Camp, Council join SAS

Michael Camp has joined SAS Institute. Michael, who graduated from Stetson University with a degree in mathematics, comes to SAS from Informatics, Inc., where he was Eastern Regional Manager of Technical Services. Michael's responsibilities at SAS include consulting and marketing.

Another new staff member is Kathy Council. Kathy is a graduate of UNC-Greensboro, and received her master's in statistics from North Carolina State University. Before joining SAS, Kathy was Coordinator of Institutional [illegible]dies at NCSU. At SAS Institute, Kathy handles consulting and marketing communications.

Billie Parrish has also joined the staff as an administrative assistant.

New quarters

The expanding SAS staff recently moved to another building in the same complex. The new offices occupy an entire floor, doubling our square footage. The physical address, 2806 Hillsborough Street, remains the same.

publications

SAS Views

The SAS publications rainbow gained an orange band recently in *SAS Views*. The SAS Short Course notes, rewritten and reduced, make up *SAS Views*. These notes are taken from the overhead slides used in the course: hence the name *SAS Views*.

SAS Views, a valuable supplement to the SAS literature, contains many insightful examples of using SAS profitably. The best way to get a copy is to take the SAS Short Course. If you can't do this, a few copies of the 276-page book are available from SAS Institute for $40.00 each.

SAS Introductory Guide

SAS Views isn't a substitute for the *SAS Introductory Guide*, also known as the *SAS Primer*. The *Introductory Guide*, although delayed, promises to be the perfect vehicle for teaching SAS to novices. In fact, familiarity with the *SAS Introductory Guide* will soon be a prerequisite for the SAS Short Course.

Publication date for the *Introductory Guide* is March 1978.

352 installations

SAS communications

SAS Institute Inc • Post Office Box 10066 • Raleigh, North Carolina 27605 • 919/834-4381
winter/spring 1978 • volume III number 3

sugi news

SUGI '78

by Helene Cavior

Caesars Palace in Las Vegas was host for the successful third annual SAS Users Group International (SUGI) meeting January 30-February 1. Despite the enticements of Las Vegas, the 241 SAS users registered were evident at all the sessions. Special thanks go to Rod Strand and Mike Farrell, conference co-chairmen, and to the session chairmen and SAS Institute for a job well done.

SUGI '78 opened with a discussion by SAS Institute staff of 1977 accomplishments and 1978 plans. Results of the 1978 Software Ballot (see below), which SAS Institute will use in planning priorities for the coming year, were presented.

Monday afternoon saw SAS users giving papers in business, medical, and teaching applications, while Tuesday's sessions featured linear models, research data management, and statistical applications. Deciding which session to attend was a difficulty that most attendees encountered!

The *Proceedings* of SUGI '78 are scheduled for May publication.

SUGI '79

SUGI '79 is scheduled for the Sheraton-Sand Key in Clearwater, Florida on January 29-31, 1979. Co-chairing the conference will be Dr. Ramon Littell of the University of Florida and Dr. William Wilson of the University of North Florida. As plans for SUGI '79 are finalized, they will be reported in *SAS Communications.*

San Antonio is the very tentative site for SUGI '80.

SUGI and SAS

At the Wednesday general session of SUGI '78, Mike Farrell discussed the relationship between SUGI and SAS Institute: SUGI is a growing organization that is dependent on staff and financial support provided by SAS Institute. Acting as a technical support group for the conference co-chairmen, SAS Institute handles local arrangements and logistics for the meeting, including pre-conference publicity.

The SUGI executive committee has as a long-term goal the chartering of SUGI as an independent organization; until that time, the support provided by SAS Institute is very much needed and appreciated. Executive committee members include Ronald Helms of the University of North Carolina, Ken Offord of the Mayo Clinic, Rod Strand of Oak Ridge National Laboratories, Michael Farrell of Miami University and Waterways Experiment Station, Ramon Littell of the University of Florida, William Wilson of the University of North Florida, Barry Merrill of Sun Information Services, Kenneth Koonce of Louisiana State University, Rudi Freund of Texas A & M, Helene Enid Cavior of California State Bureau of Prisons, Phillip Miller of Washington University, and James Goodnight, Anthony J. Barr, John Sall, and Jane Helwig of SAS Institute.

SUGI software index

SUGI is compiling an index of all user-contributed SAS procedures and macros, which will be published annually as an attachment to *SAS Communications.* Descriptions will be published in this SUGI News section of the *Communications.*

Your contributions should have this format:

procedure or macro name
author's name, address, and phone
description
(in no more than 50 words, explain what it does, how it does it, any additional software or hardware required, and your cost for distribution)
keywords (up to 5)
reference (if already published)

Mail your contributions to

Helene Enid Cavior,
SUGI Editor
1921 Glenhaven Avenue
Walnut Creek, CA 94595

SUGI news: what is it?

SUGI News will publish descriptions of new contributions to the Software Index, programming hints, short cuts, solutions to special problems, and other information of interest to SUGI members. SUGI News will appear in each issue of *SAS Communications.* Designed as a service to you, the users of SAS, SUGI News needs your support. Please send contributions or suggestions to the SUGI Editor, Helene Cavior, or call her between 7:30 and 4, Pacific time,

with your ideas: 415/347-0721. We will act as a clearinghouse, not as a distribution center.

Code critique service: Among the services planned is an evaluation of user code. This is not a debugging service, but should make code that you use frequently more efficient. Volunteers to critique the code are needed; Lou Partridge of the Academy of Natural Sciences has already offered his services. If you are interested in critiquing code or want to submit your code for critique, notify the SUGI Editor.

Ballot results

More than 500 SAS users participated in setting priorities for SAS improvements by sending software ballots to SAS Institute.

The ballots were more like shopping lists this year: 163 possible SAS enhancements were listed, and users were asked to allocate $500 among the various items. After all the ballots were received, the total dollars spent on each item was tallied. The top twenty:

1. IF-THEN, ELSE, and DO
2. book of SAS examples
3. subscripted variables
4. SAS primer
5. least squares means
6. improved GLM documentation
7. IMS interface
8. reenter erroneous lines under TSO
9. refer to range of variables
10. loglinear contingency table
11. totals in PROC PRINT
12. improved error documentation
13. Box-Jenkins procedure
14. more statistical methodology
15. explanation of procedure output
16. graphics
17. X1-X10=LOG(OF Y1-Y10);
18. categorical linear models
19. homogeneity of variances
20. MACRO library

Books mention SAS

Two books that have recently been published include examples of using SAS for statistical analysis.

Lyman Ott's book, *An Introduction to Statistical Methods and Data Analysis*, published in 1977 by Duxbury Press, uses examples from many disciplines to illustrate the statistical principles explained in the text, and shows solutions using SAS and other systems.

Richard Moore offers an overv of using the computer for data analysis in his book, *Introduction to the Use of Computer Packages for Statistical Analyses*. This short book, published in 1978 by Prentice-Hall, leads the novice step-by-step through some simple examples, including several SAS jobs.

courses

Chicago short course

Make plans now to attend the SAS Short Course to be offered May 18-19 in Des Plaines, Illinois, just outside Chicago. Instructors will be John Sall, Jim Goodnight, and Dan Chilko.

Three additional public short courses will be held this year in Atlanta, Dallas, and New York. More information will be coming to you soon on these courses.

To register for the Chicago course, call SAS Institute at 919/834-4381.

technical winter/spring 1978

SAS 76.6 available by request

The interim release SAS 76.6 is now available to all current SAS installations.

Major new capabilities in SAS 76.6 include the CHART procedure for bar charts and histograms, least squares means in the GLM procedure, and a more efficient BMDP procedure. Many small additions to SAS are also included that will contribute to the ease and convenience of using SAS.

The white insert in this newsletter contains the SAS 76.6 documentation. This information, along with the SAS 76.5 documentation contained in the *SAS Supplemental Library User's Guide*, will be merged into the *SAS User's Guide* for the next major release of SAS, in 1979.

If you would like to use SAS 76.6, ask your SAS installation representative to request it from SAS Institute.

PROC BMDP improvements

Cooperation between the developers of BMDP and SAS Institute recently resulted in a stronger PROC BMDP. BMDP programs now obtain data from SAS data sets directly, rather than requiring that the SAS data set first be converted to a BMDP save file.

The cost of this SAS-to-BMDP file conversion has been a deterrent to supplementing an analysis using SAS with a BMDP program. Now, this cost is eliminated and users can economically use the features in BMDP that are not available in SAS.

Because changes to both BMDP and SAS were necessary for this interface, you must have both the 1977 version of BMDP and SAS 76.6 in order to u it. PROC BMDP still works, however, for older versions of BMDP by converting SAS data sets to BMDP save files.

How SAS sorts

Especially when you're sorting very large SAS data sets, it's useful to know how SAS handles sorting. If your sorted data set will have the same name as the original data set, do you need space in your OS data set for two copies of the SAS data set?

The answer is yes, you do. SAS does not delete the original data set until the sort has successfully completed. This protects you against system crashes in mid-sort that might destroy your data.

Speed differences

Some SAS procedures, such as GLM, are designed to be very general, and these procedures are efficient considering the many kinds of problems they can handle. However, if you are only doing ordinary regression and are concerned about saving computer time, you might consider alternative regression procedures.

You can use such procedures as STEPWISE, SYSREG, and AUTOREG with specifications that are close to GLM's, although the outputs are formatted somewhat differently.

We tried running a 30-variable, 500-observation regression problem through four procedures with these results:

procedure	CPU time	region
GLM	3.77 sec	172K
STEPWISE	2.13	142K
SYSREG	1.26	164K
AUTOREG	1.21	140K

The CORR procedure, like GLM, is general: it handles rectangular problems and pairwise missing values, computes significance probabilities, and outputs data sets. A procedure not encumbered with these features, such as PROC FACTOR, will run faster. We ran comparisons for a 30-variable, 200-observation problem:

method	CPU time	region
PROC CORR;	2.93 sec	176K
PROC CORR NOMISS;	2.36	136K
PROC FACTOR METHOD= INPUT;	1.07	192K
PROC FACTOR METHOD= INPUT NOCORR NOSIMPLE OUT=B;	.82	192

We don't encourage users to worry about these performance considerations, since usually the computer expense is small compared to the labor cost. Nevertheless, there may be circumstances where these hints may be worth remembering—for example, if you are doing benchmarks of SAS.

Dates in PROC PLOT

When you want to produce a plot showing dates along one axis, chances are you'd like to have the same day of each month shown along the axis. For example, suppose you want to plot your company's income for the first four months of 1976. If you do not specify the dates you want, SAS will scale the values so that an equal number of days is included in each interval on the date axis. Since months do not all have an equal number of days, the dates printed would be something like 01JAN76, 31JAN76, 01MAR76, and 31MAR76.

To explicitly specify dates in PROC PLOT, you must use the internal representation of dates, which is the number of days since January 1, 1960. Here is an example:

```
PROC PLOT; FORMAT DATE DATE7. ;
 PLOT INCOME*DATE / HAXIS=5844 5875 5904 5935;
```

In this example, the variable DATE is to be plotted along the horizontal axis. Before SAS prints the horizontal axis values of 5844, 5875, 5904, and 5935 that you specified, it looks for a format associated with the variable DATE (in this case, DATE7.), and passes the date values through that format so that they are printed as 01JAN76, 01FEB76, 01MAR76, and 01APR76.

You can easily find the internal representation for dates:

```
DATA; INPUT DATE DATE7;
 PUT DATE   DATE DATE7.; CARDS;
01JAN76
01FEB76
01MAR76
...
01APR78
```

This program produces these results:

```
5844 01JAN76
5875 01FEB76
5904 01MAR76
...
6665 01APR78
```

When DATE is specified on the PUT statement without a format, SAS prints the values as the number of days since January 1, 1960. This first number in the column can be used to specify the date in PROC PLOT, as in the example above.

We are planning to allow date, time, and date-time literals in a future version of SAS. After this is implemented, you won't have to use the internal representations for the dates.

inside SAS

staff additions

Chilko joins SAS

A recent addition to the SAS Institute staff is Daniel M. Chilko. Dan received his undergraduate degree in math from Washington and Jefferson College, and his M.S. in statistics from Rutgers University. He comes to SAS from West Virginia University, where he taught in the department of statistics and computer science and consulted with SAS users at the university's computer center.

Dan's responsibilities at SAS Institute include consulting, teaching, and programming. He really is a SAS veteran, since WVU was among the first installations to receive SAS in 1970.

Mary Mason and Patsy Cantrell have also joined the staff as administrative assistants.

Datapro honor roll

SAS users recently honored SAS by putting it on the Datapro Hon Roll for the second consecutive year. SAS was one of 32 software products on the 1978 Honor Roll.

443 installations

SAS communications is published quarterly by SAS Institute Inc.

Address all correspondence to:
SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

MR. RAYVON STURDIVANT
COMPUTATION CENTER
PHILLIPS HALL
UNIV. OF NORTH CAROLINA
CHAPEL HILL, NC 27514

SAS communications

SAS Institute Inc. • Post Office Box 10066 • Raleigh, North Carolina 27605 USA • 919/834-4381
summer 1978 • volume iv number 1

primer

Introductory Guide Is Here

The *SAS Introductory Guide* is now available. This 83-page book introduces the new user to SAS and guides him through the steps necessary to create a SAS data set. Several of the most valuable SAS procedures, including SORT, PRINT, FREQ, CHART, CORR, MEANS, ANOVA, and GLM, are explained.

You can order the *SAS Introductory Guide* from SAS Institute; call our Publications Department at 919/834-4381.

SAS Introductory Guide. Jane T. Helwig. Raleigh: SAS Institute Inc., 1978. 83 pp. ISBN 0-917382-05-3. $4.95

short courses

Ottawa and New York next

Ottawa, Canada and New York City will be the locations of the next SAS two-day short courses. The Ottawa course will be held July 20-21, and the New York course is scheduled for September 14 and 15.

Registration forms for these two courses have been mailed to everyone on the *SAS Communications* mailing list. If you'd like additional forms or more information, please call us at 919/834-4381.

Two additional courses will round out the short course schedule for 1978: November 2-3 in Dallas, Texas and December 7-8 in Detroit, Michigan. Registration information for these courses will be sent to you soon.

SAS in San Diego

Watch for SAS at the annual meeting of the American Statistical Association in San Diego in August. SAS Institute will be exhibiting documentation and publications at a booth in the Town and Country Hotel. Stop by and see the *SAS Introductory Guide!*

Using SAS for CICS monitoring?

We are interested in finding out more about using SAS for CICS performance monitoring. If you've got any stories to tell about your use of SAS in this area, write or call Michael Camp at SAS Institute.

SUGI '79 call for papers

Co-chairmen William Wilson, of the University of North Florida, and Ramon Littell, of the University of Florida, have extended a call for papers for the fourth annual SAS Users Group, International (SUGI) conference, to be held January 29-31, 1979, at the Sheraton-Sand Key in Clearwater, Florida.

If you are interested in presenting a paper at SUGI '79, send an abstract of 200 words or less to any one of the session chairmen listed below. Send your abstract by October 15, and send it to only one chairman. If it's not suitable for that session, it will be forwarded to the appropriate chairman.

Plan on about twenty minutes for presenting the paper in Florida, including discussion. Include your complete address and phone number with the abstract.

SUGI '79 will focus on the use of SAS in six areas:

business (CPE, accounting, marketing, etc.)
Jim Guthrie
Department 625
The Cleveland Trust Company
900 Euclid Avenue
Cleveland, Ohio 44101
phone 216/687-5000

linear models
Dal Kratzer
The Upjohn Company
Kalamazoo, Michigan 49001
phone 616/385-6517

what's inside

medical applications

Jane Abel
Dept. 062
Abbott Laboratories
North Chicago, Illinois 60064
phone 312/688-8808

statistics

I.K. Hwang
Merck Sharp & Dohme
Building R86-222
P.O. Box 2000
Rahway, New Jersey 07065
phone 201/574-4000

research data management

Robert Bronstein
USV Pharmaceutical Corporation
1 Scarsdale Road
Tuckahoe, New York 10707
phone 914/779-6300

SAS in the university

Rudolph J. Freund
Institute of Statistics
Texas A&M University
College Station, Texas 77843
phone 713/845-3141

Important note for SAS users!

It's time to send your suggestions for SAS improvements to SAS Software Ballot, Box 10066, Raleigh, N.C. 27605. Your suggestions will be used to create the 1979 SAS Software Ballot, to be included in the SUGI conference information you will receive later this year.

Now is your chance to be heard!

sugi

Helene Cavior, editor

Using SAS under TSO?

Interested in exchanging ideas on the interactive use of SAS? Send your comments or suggestions to

Richard Israel
Time-Sharing Coordinator
Coca-Cola USA
P.O. Drawer 1734
Atlanta, Georgia 30301

Code critique service

You may have missed the item in last quarter's *SAS Communications* about the SUGI code critique service. This service is intended to help you make your SAS code as efficient as possible. Lou Partridge of the Academy of Natural Sciences and Michael Lajiness of Upjohn are the critiquers; other volunteers are needed. If you want to take advantage of the code critique service, let Helene Cavior know. Her address is 1921 Glenhaven, Walnut Creek, California 94595.

SUGI software index

Descriptions of the SAS procedures, macros, and functions below have been sent to the SUGI editor. If you are interested in using any of the software, get in touch with the author. Thanks to all who have contributed to the SUGI software index!

AOVMEAN
procedure

One-way analysis of variance using group sizes, means, and standard deviations as input. T-test for three types of comparisons: all groups with first group; all possible pairs of groups; and user-specified contrasts. Uses 38K.

T.P. Tesar
7293-32-1
The Upjohn Company
Kalamazoo, Michigan 49001

CCPLOT
procedure

Calcomp plots with up to five functions on one set of axes. Each function is defined by a pair of variables; user controls plot size, symbols for each function, axis labels, legends, and scales. Uses 49K; requires Calcomp plotter and associated software.

T.P. Tesar
7293-32-1
The Upjohn Company
Kalamazoo, Michigan 49001

DISPLAY
procedure

Plots two-dimensional data. Linear or logarithmic scales, up to 500 points, axes labels, titles, and up to four lines per graph. Uses 4K; requires DISSPLA software and Calcomp plotter.

R.J. Olson, R.H. Strand, and D.K. Kumar
P.O. Box X, Building 1505, ORNL
Oak Ridge, Tennessee 37830

GETBIT and PUTBIT
functions

Bit manipulation routines allowing users to store or retrieve any bit-string subset of a SAS variable. Uses 4K.

Frank Harrell
UNC Department of Biostatistics
Chapel Hill, North Carolina 27514

KM
macro

Kaplan-Meir estimation of survival distributions. Statements take ½ page of code.

Frank Harrell
UNC Department of Biostatistics
Chapel Hill, North Carolina 27514

PLOTTER
procedure

Line/point plotting for Calcomp plotter. Uses 110K; dependent on TUCC's modified Calcomp plotting subroutines; requires Calcomp plotter.

Frank Harrell
UNC Department of Biostatistics
Chapel Hill, North Carolina 27514

RIDGREGR
macro

Calculates coefficients of ridge regression and furnishes information so user can select appropriate coefficient values.

Robert Rogers and Ernest Hilderbrand
North Central Forest Experiment Station
1-26 Agriculture Building
University of Missouri
Columbia, Missouri 65201

TABULAR
macro

Uses tabular method of analysis to compute user-specified single-degree-of-freedom linear contrasts on treatment means from balanced factorial experiments. Available at $7.50 for the 85 card images until 1FEB80; see SUGI '78 Proceedings.

Samuel G. Carmer
Department of Agronomy
University of Illinois
Urbana, Illinois 61801

WILCOX
procedure

Distribution-free estimates of the ratio of two random variables; confidence intervals are also estimated. Uses 4K.

K. Deva Kumar and R.H. Strand
P.O. Box X, Building 1505, ORNL
Oak Ridge, TN, 37830

Supplemental procedures in SAS 76.6

SAS 76.6 includes several user-contributed SAS procedures. A brief description of each procedure is given below; for a copy of the documentation, write or call SAS Institute.

TPLSAS

Designed to transmit TPL (Table-Producing Language) table cell values to SAS; transforms the output of up to ten TPL table statements into SAS data sets. With this linkage program, TPL users can easily use SAS to perform statistical analyses on their tabular output.
Kenneth D. Buckley, *Bureau of Labor Statistics*

IPFPHC

Performs a cluster analysis of the units of a transaction flow table, which is a square matrix with the same row and column labels, frequently found in the social sciences. The ij entry is the number of items that moved or "flowed" from the i'th unit to the j'th unit over a period of time. Examples are migration, occupational mobility, trip distribution, citation, and input-output tables.
Daniel M. Chilko, *West Virginia University and SAS Institute*

LAV

Uses the least absolute values (LAV) criterion to fit a linear model. The procedure produces the LAV estimates of the coefficients, with an indication of whether or not the estimates are unique; the residual and total sums of absolute deviations; and residuals and predicted values, if requested.
James Gentle and William Lee, *Iowa State University*

RSP

Produces response contour plots for up to third-order response surface equations. A response surface equation is a formula involving a set of variables which results in a value, Y. As the variables vary over a range of values, the resulting Y's define a surface in three or more dimensions.
Daniel M. Chilko, *West Virginia University and SAS Institute*

Converting BMDP save files

If you have several BMDP save files in an OS data set, you may have been frustrated because you could only read the first save file in the data set with PROC CONVERT.

It is possible to use PROC CONVERT with any of the BMDP save files in an OS data set, since several options are available to specify which file you want.

For example, if you have a BMDP save file whose CODE is JUDGES, but it's not the first save file in the data set, you can convert it to a SAS data set by using these statements:

```
//  EXEC SAS
//SAVE DD DSN=save.file, DISP=SHR
PROC CONVERT
  BMDP=SAVE(CODE=JUDGES.);
```

In the PROC CONVERT statement, the new option CODE and the CODE name appear in parentheses after the DDname describing the BMDP save file data set. The CODE name must be followed by a period for SAS 76.5 and 76.6. Future versions of SAS will allow the period, but not require it.

If you need to further specify the BMDP save file, you can also include the CONTENT option inside the parentheses. For example, if the CODE=JUDGES file had a CONTENT of DATA, you could use this statement:

```
PROC CONVERT BMDP=SAVE
  (CODE=JUDGES.  CONTENT=DATA.);
```

Don't forget the period!

Undocumented feature of FILE statement

You can use the PUT statement to write to a member of a partitioned data set (PDS), although the *User's Guide* does not explicitly describe this feature. Add the member name in parentheses after your DD name in the FILE statement:

FILE *ddname(member name);*

You need not include the member name in the DD statement.

This feature is especially useful when more than one member is to be stored on the same partitioned data set in the same SAS job. In this case, one DD statement can be used for both FILE statements.

RETAIN statement subtleties

what usually happens When you create a SAS data set using an INPUT statement to describe the data, SAS begins the processing for each observation by setting all numeric variables to missing, all characters variables to blank. Then it reads the input lines. If you want a variable to retain the value it had for the previous observation, you give its name in a RETAIN statement. SAS won't set a value to missing when the variable appears in a RETAIN statement.

But what happens if you use a SET, MERGE, or UPDATE statement instead of INPUT? Does SAS still set each variable's value to missing when it begins the processing for each observation?

what happens when you use SET, MERGE, or UPDATE Whether SAS sets variable values to missing when you use a SET, MERGE, or UPDATE statement depends

on whether a BY statement appears. If a BY statement is present, SAS sets the variables to missing when the value of one of the BY-variables changes. Otherwise, the variables are treated as though a RETAIN statement was used. When the BY statement is not present, variable values are not set to missing; the effect is the same as if a RETAIN statement including all the variables was used.

When a SET statement is present, SAS also sets variables to missing before reading the first observation from a new data set, as well as when a value of a BY-variable changes.

RETAIN statement's effect What is the effect of a RETAIN statement when you use a SET, MERGE, or UPDATE statement? It is ignored, and the values are retained unless a BY-variable's value changes or a new data set is begun.

What does all this mean to you? When you use an UPDATE statement, each observation in your master file has a different BY-variable value, so variables are set to missing for each new observation in the master file. When you use a MERGE statement, the variables are set to missing for each new BY-group; within the BY-group, values are retained. If you use a SET statement with several data sets, remember that values won't be retained when SAS begins processing the next data set, even if its first observation has the same BY-values as the last observation of the previous data set.

IN-variables At the same time that SAS sets variables to missing, it sets the IN-variables (defined with the IN= option in the SET, MERGE, or UPDATE statement) to 0. The IN= variables are set to 1 when an observation from the corresponding data set is read.

Constructing joint tests

When you want to make joint tests of significance by combining several effects in a model, put these effects last in your MODEL statement. Then you can add the Type I sum of squares for these effects and construct your own F test using this total sum of squares as the numerator.

Suppose, for example, that you wanted to run a regression equation to predict the first-year grade-point average of incoming freshmen at XYZ University, using Scholastic Achievement Test scores (SAT) and high school grade-point average (HSGPA) as independent variables. You want to run separate regression equations for men and women, and then test to see if these equations are the same.

Use the following SAS statements:

```
PROC GLM;
CLASSES SEX;
MODEL GPA=SAT   HSGPA   SEX
  SEX*SAT   SEX*HSGPA;
```

The last three effects in the MODEL statement—SEX, SEX*SAT, and SEX*HSGPA, can be used to test the equality of the intercept, SAT parameters, and HSGPA parameters respectively. Before examining each effect individually, you can perform a joint test of these three effects.

Type I sum of squares are additive and sequential. This means that you can start at the end of the model and add sums of squares for the effects that you want to combine into a joint test. The F test in our example would be

$$F = (SS_1(SEX) + SS_1(SEX*SAT) + SS_1(SEX*HSGPA)) / (3*MSE)$$

which has numerator degrees of freedom equal to the sum of the degrees of freedom for each of the effects in the numerator, and denominator degrees of freedom equal to the degrees of freedom for error in the model.

If the result of this joint test is significant, then you can consult the type IV tests shown in your output to find which individual effects are significantly different for males and females.

Date values and BY statements

The following situation occurred when a user had two data sets, both containing the variable DATE. In one data set, the variable DATE was considered a SAS date value. In the second data set, DATE was an ordinary numeric variable. Here's what happened when an UPDATE was performed using DATE as the BY variable:

```
DATA A;  INPUT DATE YYMMDD6. A;
  FORMAT DATE YYMMDD6.;
  CARDS;
770101 6
770102 6
DATA B;  INPUT DATE 1-6   A;
  CARDS;
770101 7
770102 8
DATA C;  UPDATE   A   B;   BY DATE;
PROC PRINT;
```

```
data set C:
770101 6
770102 6
680618 7
680619 8
```

What happened?

- In data set A, the date values were read in and stored internally as SAS date values, the number of days since January 1, 1960. Thus, the date 770101 was stored as 9855.
- In data set B, the values for DATE were read in and stored as numeric values, so that the internal representation for 770101 was just 770101.
- In the UPDATE operation, these internal values were compared, and no match occurred.

- When data set C, the UPDATE data set, was printed, SAS printed both of these internal values using the YYMMDD6. output format. Thus, 9855 was printed as 770101, representing January 1, 1977; and the internal value 770101 was treated as the date June 18, 4068, and printed as 680618.

Transposing data sets

Often users want to transpose large data sets so that the variables in the old data set correspond to observations in the new, transposed data set.

For example, suppose data set A has 80 variables and 300 observations and looks like this:

OBS	X1	X2	X3	. . .	X80
1	1	1	1	. . .	1
2	2	2	2	. . .	2
3	3	3	3	. . .	3
.					
.					
.					
300	300	300	300	. . .	300

You want to transpose data set A into a data set having 300 variables and 80 observations that looks like this:

OBS	Y1	Y2	Y3	. . .	Y300
1	1	2	3	. . .	300
2	1	2	3	. . .	300
.					
.					
.					
80	1	2	3	. . .	300

The following SAS statements accomplish this:

```
//  EXEC SAS
//TEMP DD UNIT=SYSDA,SPACE=(TRK,(50,50))
DATA _NULL_;
  SET A;
  FILE TEMP;
  PUT (X1-X80) (RB4.) @;
DATA B;
  INFILE TEMP;
  INPUT X RB4. @@;
  N=MOD(_N_-1, 80);
PROC SORT;  BY N;
DATA _NULL_;
  SET B;
  FILE TEMP;
  PUT X RB4. @;
DATA C;  INFILE TEMP;
  INPUT (Y1-Y300) (RB4.) @@;
```

In this program, data set A is temporarily stored on disk (FILE TEMP) in the order X1-X80 of observation 1, followed by X1-X80 of observation 2, and so on.

For data set B, each of these values is read one at a time, given the variable name X, and associated with a new variable N that has the values 0 through 79 for the first 80 values read, 0 through 79 for the next 80 values read, and so on until all the values are read.

Data set B is then sorted by this variable N, which arranges the variables in the proper order for the transposed data set. In other words, when data set B is sorted, all 300 values for X1 from data set A occur first in the sorted data set, followed by all X2 values, and so on. The sorted data set is then stored on disk where it is read as data set C. Each set of 300 values is read as variables Y1-Y300.

If some of your variables are character variables, the procedure becomes a little more complicated since you must keep up with which variables are character and which are numeric in the INPUT statements.

If you have a small data set, you can transpose it using MATRIX as shown in example 2 on page 42 of the *SAS Supplemental Library User's Guide*.

Subsetting IF and automatic variables

When you use a subsetting IF statement in a DATA step, keep in mind how SAS carries out this statement. For example, consider this DATA step:

```
DATA;
  INPUT NAME $ AGE;
  IF AGE GT 18;
  PUT NAME= AGE= ;
  CARDS;
  data lines
```

SAS carries out the subsetting IF statement for each observation by comparing the observation's AGE value to 18. When the IF-condition is true and AGE is greater than 18, SAS continues executing statements for that observation. When the IF-condition is false and AGE isn't greater than 18, SAS stops executing statements for the observation, doesn't add it to the data set being created, and returns for the next observation.

Thus, the program above prints the NAME and AGE values only for observations where the AGE value is greater than 18.

Some users have fallen into the trap of using the automatic FIRST. and LAST. variables after a subsetting IF statement. The values of the FIRST. and LAST. variables are set at the beginning of the DATA step for each observation. A subsetting IF may delete the first or last observation in a BY group, and thus any subsequent programming statements which refer to FIRST. and LAST. variables may never be processed.

If you need to use both a subsetting IF and FIRST. or LAST. variables in a DATA step, be sure to put any subsetting IF statements after statements that include FIRST. and LAST. variables.

inside SAS

SAS updates

Since the last *SAS Communications*, two updates have been mailed to those installations that are running SAS 76.6. Updates are now identified with letters following the release number: for example, the second update was 76.6B. The letter is now included in the release identification printed at the top of the SAS log. If both of these updates have been applied at your installation to SAS 76.6, the top line of the log will say that your program has been run under SAS 76.6B.

Staff changes

Several changes have taken place this spring in the staff at SAS Institute. Tressa Gates moved with her husband to Mt. Vernon, Ohio. Ann Rutledge has joined our marketing department, and Robin Layland has also joined the marketing staff.

Ann Baggett has retired from her job as publications manager, and Patsy Cantrell is now in charge of publication orders.

Wayne Lindsey has come to SAS to operate our new phototypesetter; Wayne will handle preparation of all our printed material between the writer and the printer.

Eve Cooper has also joined our staff as an administrative assistant.

Phototypesetter arrives

We're happy to announce the arrival at SAS Institute of a Compugraphic EditWriter 7500 phototypesetter. All future SAS publications will be typeset here at SAS Institute, helping us produce quality publications speedily and accurately.

The first major project for the phototypesetter is the *SAS User's Guide*, 1979 edition. We're hard at work on the new *User's Guide*, and now is the time to send us your suggestions and comments for improving it.

502 installations

SAS communications is published quarterly by SAS Institute Inc.

Address all correspondence to:
SAS Institute Inc.
P.O. Box 10066
Raleigh, N.C. 27605

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

MS. JUDY HALLMAN
COMPUTATION CENTER
PHILLIPS HALL
UNIV. OF NORTH CAROLINA
CHAPEL HILL, NC 27514

U.S. Postage
PAID
Raleigh, N.C.
Permit No. 977

SAS communications

SAS Institute Inc. • PO Box 10066 • Raleigh, NC 27605 USA • cable SASINST • 919/834-4381
fall 1978 • volume iv number 2

last chance

Call for papers

Now that summer is over and you're making plans for the coming months, consider presenting some of your SAS applications at the annual SUGI conference in January. Many of you are using SAS in ways that would be of interest to others, and we hope you'll share your ideas with us.

Co-chairmen Ray Littell and Bill Wilson are extending a final call for papers for SUGI '79. If you're interested in presenting a paper, send a 200-word abstract to one of the session chairmen listed below by October 15. Include your address and phone number with your abstract.

business (SMF, accounting, marketing, etc.)
Jim Guthrie
Department 625
The Cleveland Trust Company
900 Euclid Avenue
Cleveland, Ohio 44101
phone 216/687-5563

linear models
Dal Kratzer
The Upjohn Company
Kalamazoo, Michigan 49001
phone 616/385-6517

medical applications
Jane Abel
Dept. 062
Abbott Laboratories
North Chicago, Illinois 60064
phone 312/688-8808

statistics
K. Hwang
Merck Sharp & Dohme
Building R86-222
P.O. Box 2000
Rahway, New Jersey 07065
phone 201/574-4000

research data management
Robert Bronstein
USV Pharmaceutical Corporation
1 Scarsdale Road
Tuckahoe, New York 10707
phone 914/779-6300

SAS in the university
Rudolph J. Freund
Institute of Statistics
Texas A&M University
College Station, Texas 77843
phone 713/845-3141

january sunshine

Come to SUGI '79

It's time to make your plans to come to Florida in January. The 1979 meeting of the SAS Users Group International (SUGI) will be held January 29-31 in Clearwater, Florida, at the Sheraton-Sand Key Hotel on the Gulf of Mexico. All SAS users are cordially invited to attend.

The six SUGI '79 session chairmen have scheduled an outstanding roster of invited speakers. Contributed papers from SAS users are also flowing in, and these titles will be listed in the next issue of SAS Communications.

This year will be the fourth annual SUGI meeting. Those who have attended earlier meetings have found them valuable for learning more about SAS, meeting with the SAS staff, and interacting with other SAS users to improve problem-solving effectiveness.

Since January represents the height of the Florida season, you may want to make your airline reservations now. Likewise, you'll need to send your hotel reservation in earlier this year. More information on SUGI '79, including complete registration material, will be coming to you within the next few weeks.

The invited speakers for SUGI '79 are given below. The titles of their papers give you a taste of what SUGI '79 has to offer:

linear models
S.R. Searle, Ph.D.
Cornell University
Relationships between the estimable functions of the SAS GLM output for unbalanced data and the hypothesis tested by traditional F-tests.

continued on page 2

short courses

Dallas, Detroit

Courses in Dallas and Detroit round out the SAS short course schedule for 1978. The Dallas course will be held November 2-3, and the Detroit course is scheduled for December 7-8 at the new Renaissance Center.

If you're on the *SAS Communications* mailing list, you should have received registration information for these courses. If you need more registration folders or additional information, call SAS Institute at 919/834-4381.

Courses scheduled for early 1979 will be in Los Angeles and Washington, D.C. Look for more information in the winter issue of *SAS Communications.*

J.H. Goodnight, Ph.D.
SAS Institute
New features in GLM and VARCOMP.

scussants:
William L. Sanders, Ph.D.
University of Tennessee

F. M. Speed, Ph.D.
Mississippi State University

business

H.W. Barry Merrill
Sun Information Services
A new tool for the evaluation of SMF data with SAS.

medical applications

Daniel H. Freeman, Ph.D.
Yale University School of Medicine
Spinal cord injuries in the hospital discharge survey 1971-1975: a SAS/AUTOGROUP/BMD interface.

Harji I. Patel, Ph.D.
CIBA-GEIGY
Analysis of covariance of incomplete ta in experiments with repeated measures in clinical trials.

Carl Metzler, Ph.D.
The Upjohn Company
An evaluation of nonlinear regression programs.

statistics

A. Lawrence Gould, Ph.D.
Merck Sharp & Dohme Research Labs
Multivariate multiple comparisons using SAS.

Bernard F. McDonagh, Ph.D.
3M Company
Statistical applications using SAS at 3M Company.

research data management

Rodney H. Strand, Ph.D.
Oak Ridge National Laboratories
Environmental data: management and analysis considerations.

John Duffy
ICI Americas
The "critical mass" concept.

Herbert Kirk, Ph.D.
North Carolina State University
The use of SAS in administrative computing at both the university and federal levels.

SAS in the university

James E. Gentle, Ph.D.
Iowa State University
The role of packaged programs in statistical methods courses.

Kenneth L. Koonce, Ph.D.
Louisiana State University
The impact of SAS at the university.

sugi

Helene Cavior, editor

sasware index
Send your procs

The SUGI SASware index now includes eleven contributions. To be a useful service to SAS users, we must make the SUGI index as complete as possible. If you have written any SAS procedures, macros, or functions that you'd like to share with other users, send a description to the SUGI editor, Helene Cavior. Include the procedure or macro name; your name, address and phone; a 50-word description of the procedure and whether it requires other software or specific hardware; up to 5 keywords; and a reference if it's already been published. Helene's address and phone number are

Helene Enid Cavior
1921 Glenhaven Avenue
Walnut Creek, California 94595
415/347-0721

If you act now, your procedure can be included in the first SUGI SASware Index. This index will be available at the January SUGI meeting in Florida, thanks to Rod Strand of Oak Ridge National Laboratories, who has written the SAS program to provide the listing.

technical

fall 1978

documentation
SAS technical reports

Papers in the SAS Technical Report Series are now available from SAS Institute. These papers document work-in-progress, describe new supplemental procedures, and cover a variety of application areas. Some of the features mentioned in the reports are still experimental: for example, the ARIMA procedure, described in report P-101, has not yet been distributed.

The reports now available are listed below. The charge for each is also given; this charge reflects only our production costs. Mailing costs are extra. If you'd like to order one or more of the reports, call or write the SAS Publications Department at P.O. Box 10066, Raleigh, N.C. 27605 (919/834-4381).

A-101 *SAS merge applications.* John P. Sall, 43 pp., $3.98.
A-102 *SAS regression applications.* John P. Sall, 96 pp., $7.16.
A-103 *SAS update applications.* John P. Sall, 20 pp., $2.60.
A-104 *Using SAS for large data sets.* John P. Sall, 9 pp., $2.00.
A-105 *Elements of SAS style.* John P. Sall, 6 pp., $2.00.
R-101 *Tests of hypotheses in fixed-effects linear models.* J.H. Goodnight, 16 pp., $2.36.
R-102 *Computing expected mean squares.* J.H. Goodnight and F.M. Speed, 6 pp., $2.00.
R-103 *Least squares means in the fixed-effects general linear model.* J.H. Goodnight and Walter R. Harvey, 9 pp., $2.00.
R-104 *A simplified algorithm for the W-transformation in variance component estimation.* J.H. Goodnight and W.J. Hemmerle, 11 pp., $2.06.
R-105 *Computing MIVQUE0 estimates of variance components.* J.H. Goodnight, 10 pp., $2.00.
R-106 *The sweep operator: its importance in statistical computing.* James H. Goodnight, 48 pp., $4.98.

P-101 *The ARIMA procedure.* John P. Sall, 24 pp., $2.84. (Box- Jenkins time-series models)

P-102 *The SIMLIN procedure.* John P. Sall, 9 pp., $2.00. (Forecasting models from SYSREG)

P-103 *Nonlinear modeling procedures.* John P. Sall, 28 pp., $3.08. (Estimation and simulation of nonlinear simultaneous equation models)

P-104 *The COPY procedure.* Anthony J. Barr, 3 pp., $2.00. (Copying SAS data sets)

P-105 *The TAPELABEL procedure.* Daniel M. Chilko, 4 pp., $2.00. (Dumping tape label information)

P-106 *The X11 procedure.* Daniel M. Chilko, 15 pp., $2.30. (Seasonal adjustment)

S-101 *The LAV procedure.* James Gentle and William Lee, 5 pp., $2.00. (Least absolute values (L1) regression)

S-102 *The IPFPHC procedure.* Daniel M. Chilko, 8 pp., $2.00. (Cluster analysis of transaction flow table)

S-103 *The RSP procedure.* Daniel M. Chilko, 10 pp., $2.00. (Response surface plots)

S-104 *The TPLSAS procedure.* Kenneth D. Buckley, 17 pp., $2.42. (TPL/SAS interface)

S-105 *The PREDICT procedure.* M. Martha McCrate and J. Philip Miller, 11 pp., $2.06. (Multivariate logistic model)

T-101 *Accessing SAS data sets with other programs.* John P. Sall, 9 pp., $2.00.

T-102 *SAS library assembler routines ($-series).* John P. Sall, 13 pp., $2.18.

G-101 *SAS—a unified language for computer performance evaluation.* William R. Gjertsen and Robert M. Gaddy, 10 pp., $2.00.

Key to first letter of paper number: A for application-oriented paper; R for research paper; P for paper describing new procedure; S for paper describing new supplemental procedure; G for general-information paper.

time and space
Consultant's corner

Here's a typical beginning of a phone conversation between a SAS Institute consultant and a caller:

MASON: This is Mason Nichols, may I help you?

CALLER: Yes. I'm getting ready to run this SAS job with 83,429 records and I need to know how much time and space to give it.

Let us say that you are the caller. How will Mason help you?

Her first suggestion will be to run the job on a subset of the records, say 1000. Then you can extrapolate to estimate the run time for the entire file. For example, say that the job uses 7 seconds to process 1000 records. Since the entire file contains 83,000-odd records, you multiply 7 seconds by 83 to get an estimated run time of 581 seconds, or 9.68 minutes. This figure is probably high, since the test job has a larger percentage of start-up time, but conservative estimates are recommended. Thus a time value of 10 minutes would be reasonable in this case.

Why can't Mason estimate the time required? That time depends on many factors: the computer configuration where the job is run, what the job does, how many variables are used, how many records are actually processed. The easiest and most reliable way to estimate the time is to run a sample job. Although this means you must run two jobs, you are spending a little to save a lot. If your job should bomb because of time after running for many minutes, the total expense would be large indeed. In addition, it's good programming practice to test your program before trying it out on a huge file.

The best way to get a 1000-observation subset is to add OBS=1000 to the INFILE statement, or to add the statement

```
OPTIONS OBS = 1000;
```

as the first line of your program.

When you are dealing with very large files, it's a good idea to make sure that the SAS WORK file is large enough to hold the SAS data sets created by your program. On the SAS log from the sample run, you will see the number of observations per track for each SAS data set created. Divide that number into the total number of observations to get the number of tracks needed. Then total the tracks required for each data set, adding 2 for the directory. (Also, remember that PROC SORT makes a copy of each data set rather than sorting in-place, even when both input and output data sets have the same name.) Now check the WORK DD statement in your installation's cataloged SAS procedure to see how many tracks are routinely allocated for WORK—usually, 240 with a secondary allocation of 60. If your job requires more than 240 tracks, override the WORK DD statement to give WORK a larger SPACE allocation.

If you use the SORT procedure in the job, you should also make sure that enough sort workspace will be available. Use the formula on page 234 of the User's Guide to determine the space required, and add a SORT parameter to your EXEC statement if necessary.

SAS

SAS is a package that always is moving,
Never content to sit idly by.
Its authors in Raleigh forever are proving
That theirs are procedures of quality high.

Updates, revisions, both minor and major,
Are faithfully sent to us every few weeks.
From North Carolina we stand in no danger
Of missing the latest statistic techniques.

The latest release, by new functions augmented,
Has some that are simple and some quite complex.
And so, to our users, we proudly present it—
The coming of SAS76.6.

Harriet Schabes

SAS Institute proudly presents the latest SAS poem from Harriet Schabes, a.k.a. Ogden Crash. Reprinted with permission from *Communications*, the newsletter of the University Computer Center, City University of New York.

proc format
Don't use TO in ranges

the SAS 76.6 documentation found in the 1978 winter/spring issue of *SAS Communications*, the PROC FORMAT changes included the keyword TO in value ranges in the VALUE statement. For example, the VALUE statement below was legal:

```
PROC FORMAT;
  VALUE Q 1 TO 5 = A
          6,7 TO 9 = B;
```

Unfortunately, allowing the TO keyword caused more problems to our users than it solved for them, so the feature has been removed. After the 76.6C update has been installed on your system, the word TO in VALUE ranges must be replaced with a dash "-" as before:

```
PROC FORMAT;
  VALUE Q 1-5 = A
          6,7-9 = B;
```

saving·money
Autobatch SAS

If you use SAS at a university, you can sometimes reduce b costs substantially with autobatch SAS.

what is autobatch? Computer programs run by students are often very short. For these jobs, the setup time needed to bring the language processor into the computer's main storage and to allocate work files is often large compared to the time that the job actually executes. If several jobs that use the same compiler are "batched" at regular intervals and run one after another, these setup costs are shared. WATFIV and PL/C, student compilers for FORTRAN and PL/I, are two well-known examples of autobatch processors.

SAS can also be used as an autobatch processor, and the savings can be significant. To illustrate, two small SAS jobs were first run as standard batch jobs and then as autobatch jobs. Since costs differ among computer centers, the jobs were run at both Texas A & M and North Carolina State University.

	batch cost	autobatch cost	savings
Texas A & M			
GLM job	$0.77	$0.25	67%
MATRIX job	.73	.23	68%
NCSU			
GLM job	.47	.24	48%
MATRIX job	.38	.17	55%

So an average savings of 62% was achieved by using autobatch SAS for these jobs at these two installations. Improved turnaround time is another benefit, since autobatch processors are scheduled to run a batch of jobs every few minutes. At most installations, adding another autobatch processor in no way impacts computer performance.

Autobatch isn't a universal solution. Only jobs that require no disk or tape files other than those in the cataloged procedure are eligible, and most installations restrict job execution time and output pages to small quantities, typically 10 seconds and 20 pages. For centers that run many small SAS jobs, though, autobatch SAS can reduce user costs.

proc freq
Percent variable in output data set

You can use PROC FREQ in SAS 76.6 to create an output data set containing cell percentages as well as frequency counts. For example, say you want to analyze survey data on political preferences. You use PROC FREQ to tabulate party by age, and create an output data set:

```
PROC FREQ;
  TABLES PARTY*AGE / OUT = NEW;
```

FREQ prints a crosstabulation table showing the number of persons affiliated with each political party at each age level:

TABLE OF PARTY BY AGE

PARTY / AGE	18	19	TOTAL
DEMOCRAT	42	50	92
FREQUENCY			
PERCENT	25.15	29.94	55.09
ROW PCT	45.65	54.35	
COL PCT	52.50	57.47	
REPUBLICAN	37	35	72
FREQUENCY			
PERCENT	22.16	20.96	43.11
ROW PCT	51.39	48.61	
COL PCT	46.25	40.23	
OTHER	1	2	3
FREQUENCY			
PERCENT	0.60	1.20	1.80
ROW PCT	33.33	66.67	
COL PCT	1.25	2.30	
TOTAL	80	87	167
	47.90	52.10	100.00

The output data set created by FREQ contains eight observations, one for each cell of the crosstabulation table. The COUNT variable's values are the frequency counts for the cells, and the PERCENT variable's values are the percentage of the total count represented by the cell:

PARTY	COUNT	PERCENT	AGE
DEMOCRAT	42	25.1497	18
DEMOCRAT	50	29.9401	19
REPUBLICAN	37	22.1557	18
REPUBLICAN	35	20.9581	19
OTHER	1	0.5988	18
OTHER	2	1.1976	19

statistics
Repeated measures

Although the 1976 edition of the *SAS User's Guide* does not mention repeated measure designs, it is easy to analyze these designs with the ANOVA procedure.

You can think of repeated measure designs as split-plot designs with subjects forming the whole plots. In agriculture, the whole plots in an experiment can be subdivided, with additional treatments applied to the subplots. In a psychological or medical experiment, each subject can be measured repeatedly, at different times and possibly under different experimental conditions. The analysis is the same.

You must use a TEST statement, as with split-plot designs, if you want to test effects between subjects (whole plots), so that the sum of squares for the effect is compared with the proper error term for that effect. Effects within subjects are usually tested against the residual error.

Elements of the split-plot/repeated measures analogy:

split-plot experiment		repeated measures experiment	
whole plot	A	between subjects	A
	REP (A)		SUBJECT (A)
split-plot	B	within subjects	B
	A*B		A*B
	REP(A B)		SUBJECT (A B)

The analysis:

```
PROC ANOVA;
  CLASSES A B SUBJECT;
  MODEL Y = A SUBJECT(A) B A*B;
  TEST H = A     E = SUBJECT(A);
```

568 installations

efficiency
Time trade-offs

When timing considerations are important—for example, if you're processing many thousands of observations—you can often improve SAS performance significantly by doing a little more work yourself.

A good example is a simple data listing. You'd ordinarily use PROC PRINT for a data listing, since PRINT chooses optimal formats for your variables, arranges the data nicely on the page, and requires only a single SAS statement. But perhaps you could reduce the time required for PROC PRINT by giving it a format to use rather than letting it choose one.

The investigation To test this idea, we first generated a data set containing 1000 observations and 40 variables. Then the PRINT procedure was used twice, once without specifying formats and once using a FORMAT statement:

```
PROC PRINT DATA=TEST;
PROC PRINT DATA=TEST;
  FORMAT X1-X40 6.2;
```

As an additional check, we used a PUT statement to print the same data set:

```
DATA _NULL_;
  SET TEST;
  FILE PRINT;
  PUT (X1-X40) (8.2);
```

The results Without a FORMAT statement, PROC PRINT used 21.38 cpu seconds. With a FORMAT statement, only 9.51 seconds were required. Using a DATA step with SET and PUT statements cut the time by more than half to 3.77 cpu seconds.

The moral Although the simple statement

```
PROC PRINT;
```

is the easiest for you, SAS does more work. If you assume some of this work by writing

```
PROC PRINT;
  FORMAT X1-X40 6.2;
```

SAS does less work, saving you money. If you take the time to write the four SAS statements

```
DATA _NULL_;
  SET TEST;
  FILE PRINT;
  PUT (X1-X40) (8.2);
```

SAS does only half as much work, although your list won't be as attractive.

Keep in mind that worrying about such performance issues is generally important only when you are processing large files. The value of the time you spend optimizing one-shot programs might be far in excess of the cost savings.

The efficiency of the PRINT procedure, which was written in PL/I, will increase in the next major release of SAS, since it is now being rewritten in assembler. With the new version, the simple statement PROC PRINT; should take no more than 50% more time than a PUT statement requires.

inside SAS

Blair, Nichols
New SAS staff

William Blair has joined the systems staff at SAS Institute, bringing added expertise in computer science, statistics, and systems programming. William graduated from the University of North Carolina at Chapel Hill with a B.S. in bacteriology, and has completed course work for an M.S. in computer science at UNC. He comes to SAS from First Computer Services in Charlotte, where he was a technical specialist; he was also manager of the JES2 Project of GUIDE's Operating Systems Division MVS Group.

Mason Nichols has joined our consulting staff to provide additional telephone support. Mason's B.S. in mathematics is from East Carolina University, and she has eight years' experience as a SAS data analyst and assembler programmer at the North Carolina Science and Technology Research Center in the Research Triangle Park.

Patti Reinhardt augments the SAS communications staff as a technical editor. Patti received her B.A. in English from the University of North Carolina at Chapel Hill, where she also studied computer science and statistics.

Marsha Russo has joined the administrative staff and is working with customer contracts and renewals. Marsha's last position was with Digital Equipment Corporation as a branch financial representative.

Marti Dominick, who comes to SAS with nine years' experience in cost accounting and vendor relations, is another new member of the administrative staff.

sas updates
Summer mailings

In early July, a copy of the *SAS Introductory Guide* went to each current SAS installation; on September 7, update 76.6C went to all SAS installations running the 76.6 version. If your installation's SAS representative hasn't received these, he should let SAS Institute know.

SAS communications is published quarterly by SAS Institute Inc.

Address all correspondence to:
SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

U.S. Postage
PAID
Raleigh, N.C.
Permit No. 977

MR. GRADY MEEHAN
IRSS
MANNING HALL
UNC-CH
CHAPEL HILL, NC 27514

SAS communications

SAS Institute Inc. • PO Box 10066 • Raleigh, NC 27605 USA • telex 802505 • 919/834-4381
spring 1979 • volume iv number 4

conference

SAS users meet

Enthusiasm and conversation were high at the fourth SUGI (SAS Users Group International) meeting in January, held this year in Clearwater Beach, Florida. SUGI '79 was the biggest and best yet, with almost 500 attendees and nearly 100 papers. Some scenes: registration desk swamped by waves of SAS users . . . SAS consultant Mason Nichols presenting Dr. Phil Miller with the User Feedback Award, an engraved plaque featuring a dead bug . . . standing room only for S.R. Searle's presentation . . . Jim Goodnight, Rudi Freund, and Herb Council pushing a catamaran through the surf . . . LAST.HELWIG unexpectedly introducing herself from the back row . . . Bill Blair breaking transparency-flipping speed records while condensing his presentation of new SAS features into 34 minutes . . . Rod Strand rushing among sessions to present all three of his papers . . . SUGI-green t-shirts provoking a stampede . . . the Gulf sunset and spirited conversation keeping 500 SAS users warm at the poolside mixer . . . join us next year!

survey

Technical environment at SAS sites

We wanted to learn more about the technical environment—data base management systems, operating systems, interactive services, and graphic devices—at SAS sites, so last fall we sent questionnaires to more than 500 SAS representatives. We also asked about the extent and nature of SAS usage at each site.

Almost three hundred usable responses were returned to us, and we would like to thank those who completed and returned the questionnaire.

Some highlights of our findings:

- Over half (52%) of those responding have or plan to acquire the IMS data base management system.
- MVS is the operating system used by a majority (56%) of those responding.
- Almost two-thirds (64%) have TSO.
- The most common graphics devices are the Tektronix tube (45%) and the Calcomp plotter (40%).
- In general application areas, SAS is used often for statistical analysis by 71% of those responding; for descriptive statistics by 68%; for data manipulation by 61%; for retrieval by 47%; and for report writing by 41%.
- In specific areas, SAS is used often for computer performance evaluation by 38%; for analysis of designed experiments by 36%; for social research by 28%; for forecasting and econometrics by 22%; and for business data processing by 21%.
- 94% reported that SAS usage is growing at their installations.

SAS Institute plans to use the results of this survey in making decisions about the future growth and development of SAS. If you'd like more specific information about the results of our survey, give SAS Institute a call.

users decide

Ballot results

Over 800 SAS users participated in setting priorities for SAS improvements by sending SASware ballots to SAS Institute.

Users were asked to allocate $500 among 185 possible SAS enhancements. The top twenty items, based on total dollars spent, are listed below:

1 book of SAS examples
2 extended graphic capabilities
3 DO-WHILE statement
4 ability to reenter lines under TSO
5 IMS interface
6 table look-up or recode facility
7 ability to run under CMS
8 indexing system for SAS data sets
9 ability to read VSAM and BDAM files
10 improved understandability of GLM documentation
11 ability to define functions as in PL/I and FORTRAN
12 median, percentile estimates
13 more statistical methodology in documentation
14 symbolic parameters in MACROs
15 MACRO library facility
16 selection of observations within PROCs
17 TSO user's guide
18 more multiple comparison methods
19 guide to SMF processing
20 in PROC MEANS, ability to output in following format:

	X	Y	Z
MEAN	x.x	x.x	x.x
N	x.x	x.x	x.x

courses

Chicago and Atlanta

Chicago, Illinois and Atlanta, Georgia are the locations for the next SAS two-day short courses. The Chicago course will be held April 2-3 at the Continental Plaza; the Hyatt Regency is the

(continued on next page)

location for the Atlanta course May 10-11.

Registration forms for these two courses have been mailed to everyone on the *SAS Communications* mailing list. If you'd like additional forms or more information, please call us at 919/834-4381.

Courses planned for the summer are in Washington, D.C. June 14-15, in Boston July 30-31, and in Raleigh August 20-21.

courses

Writing procedures

In addition to the courses mentioned above, a short course in writing SAS procedures is scheduled for Raleigh, North Carolina on May 17-18. This course is more technical in nature than the regular two-day course, and is intended for the programmer with a knowledge of FORTRAN or PL/I who wants to write his own SAS procedures.

Registration forms for the procedure-writing course will be mailed in early April.

Who to call at SAS

Many SAS users need to call SAS Institute from time to time. You can make your telephone call shorter by asking directly for the person who can solve your problem.

If you want to . . .	ask for . . .
order manuals	Patsy Cantrell
register for a public short course	Billie Parrish
arrange an in-house short course	Herb Kirk
discuss your contract	Marsha Russo
add or change a name on our mailing list	Mary Mason
get help with a technical question	a technical consultant

The SAS Institute phone number is 919/834-4381.

unbalanced data

Annotated output for GLM

Information is now available on the meaning of SAS GLM output for unequal subclass numbers (unbalanced) data. In a project directed by S.R. Searle at Cornell University, output generated by a series of small sets of unbalanced data processed through GLM and HARVEY has been extensively annotated with an explanation of what the output values are. These annotated computer outputs, in 8½ x 11 form, are available for $5 each for GLM and HARVEY, and also for BMDP2V, SPSS ANOVA, and GENSTAT ANOVA. Orders will be mailed post-free from:

Biometrics Unit
339 Warren Hall
Cornell University
Ithaca, NY 14853

upon receipt of $5 (us) for each annotated output ordered. Checks should be made payable to Cornell University.

SUGI *Helene Cavior, editor*

SUGI '79

The fourth annual SAS Users Group International (SUGI) conference was held in Clearwater Beach, Florida, January 29-31, 1979. SAS's growth over the past year was reflected in the conference registration—462 people compared with 241 at last year's conference. Special thanks go to the conference chairmen, William Wilson of the University of North Florida and Ramon Littell of the University of Florida; the session chairmen; and to SAS Institute, for a most informative and enjoyable conference.

Fourteen invited papers and 70 contributed papers were presented, and this year awards were given for the best presentation in each of the six sessions. The winners were:

Business
William M. Parker
North Central Texas Council of Governments

Medical Applications
Richard A. Enz and Doron Steger
American Hoechst Corporation

SAS in the University
Douglas Cockrell and Ione Cockrell
University of South Carolina

Linear Models
Paul Stewart and Ronald W. Helms
University of North Carolina

Research Data Management
Michael P. Farrell, A. Dale Magoun, and Karen Daniels
U.S. Army Corps of Engineers

Statistics
Edward R. Jones
Chevron Research Company

Honorable mentions were also awarded to Terry Therneau of the Mayo Clinic, in the Medical Applications session; Samuel G. Carmer and W.T. Hsieh of the University of Illinois, in the Linear Models session; and Marilyn Bodow of the College of Medicine and Dentistry of New Jersey, in the Research Data Management session.

The *SUGI '79 Proceedings* will be published in early spring. A copy will be mailed to each conference attende call Patsy Cantrell at SAS Institute to order additional copies.

Performance measurement group

Evening mixers at SUGI '79 and birds-of-a-feather sessions gave additional opportunities for exchange of ideas. The birds-of-a-feather session on Performance Measurement and Capacity Planning resulted in the formation of a group to coordinate a SASware exchange of programs in this area. Anyone interested in participating should contact:

Pete Rikard
User Services, Computing Center
Virginia Commonwealth University
1015 Floyd Avenue
Richmond, VA 23285
804/257-1582

or

Pete Kuchnicki
Southwestern Bell
St. Louis, MO 63102
314/427-7178

or J. Michael Camp at SAS Institute.

Power users group

The birds-of-a-feather session at SUGI '79 for users in the utility industry also resulted in the formation of a group to address specific application areas of concern, such as environmental data analysis, load forecasting, and support. Initially, a directory of utility SAS users will be distributed. Anyone interested in participating in this group should contact:

Joe Donaghy
Carolina Power & Light
411 Fayetteville Street
PO Box 1551
Raleigh, NC 27602
919/836-7502

SUGI '80

SUGI '80 is planned for San Antonio, Texas and will be chaired by Rudolph J. Freund of Texas A & M University. The sessions will be organized around usage areas rather than interest groups to minimize the problem of simultaneous sessions of interest. Next year's session chairmen:

Statistics
Ed R. Jones, Chevron Research Company
Don Henderson, USDA

Computer Performance
Pete Kuchnicki, Southwestern Bell

Information Systems
Ruth Ingram, Procter and Gamble

Training and Support of SAS
Jan Curley, Ford Motor Company

In addition, SUGI '80 will feature a general poster session that will be coordinated by Helene Cavior of the Bureau of Prisons. The poster session will provide a more appropriate format for presentations describing procedures or macros.

Individuals wishing to participate in next year's conference should submit an abstract to the appropriate session chairman, who may recommend the poster session as an alternative to an oral presentation. It will remain the option of the author as to his method of presentation.

SASware index

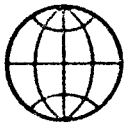
Copies of the SASware Index were distributed at the conference and are also being distributed, upon request, by the SUGI editor.

Interest in copies of the SASware Index has been great, but the contributions have been slow in arriving. In order to increase the Index's usefulness, a committtee has been formed to solicit contributions for all macros and procedures appearing in the SUGI proceedings. Helene Cavior heads the committee, and several SUGI members have volunteered to work with her. You can save the committee some work by sending a description of your proc or macro to:

Helen Cavior, SUGI Editor
1921 Glenhaven Avenue
Walnut Creek, CA 94595
415/347-0725 (9:30-6:00 Pacific time)

Please use the following format:

PROC or MACRO name
Author's name and address
Description (up to 50 words)
(include what it does, how it does it, additional software or hardware required, and the cost for reproduction/distribution)
Keywords (up to 5)
Reference (if already published)



SAS around the world

Agnar Hoskuldsson, European editor

(NOTE: This column is the first contribution from Agnar Hoskuldsson of the Technical University of Denmark, who is the European editor for *SAS Communications*.)

Saving intermediate results

You can use intermediate results from SAS procedures to compute statistics that aren't otherwise available in SAS. These results can be stored on disk or tape, then used as input to other PROC steps. Two examples of using intermediate results are shown below: computing confidence intervals for mean values, and producing normal probability plots of residuals.

Confidence intervals for mean values: example 1

The first lines in the SAS job contain JCL and the SAS statements necessary to read in the data from cards. Each data line contains the variables REGION and X1-X6. The OPTIONS statement requests that all results be printed using a linesize of 72.

```
//jobname JOB acct, name
//   EXEC SAS
//A   DD   DSN=NAME1, DISP=OLD
OPTIONS LS=72;
DATA D;
  INPUT REGION X1-X6;
  CARDS;
  data lines
```

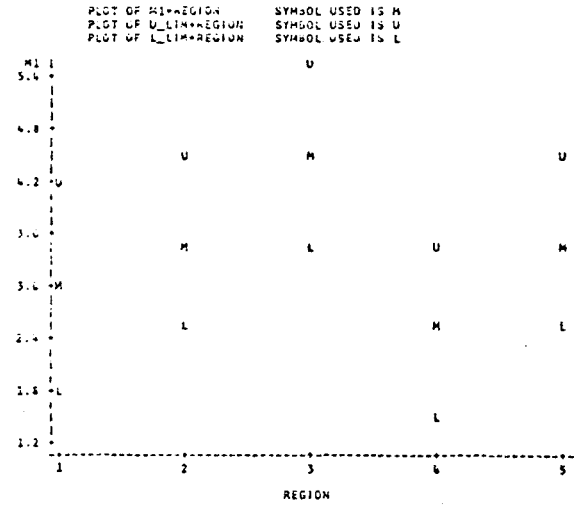
In the next step, you sort the data, and compute means, standard errors, and the number of nonmissing values for each REGION. Since the goal is to save these results for later use, no printout from the MEANS procedure is needed.

```
PROC SORT; BY REGION;
PROC MEANS NOPRINT; BY REGION;
  OUTPUT OUT=A.REGMEAN   N=N1-N6
    MEAN=M1-M6  STDERR=S1-S6;
```

The statistics you want are stored in the SAS data set REGMEAN in the OS data set with DSN=NAME1. You can now use this SAS data set containing means and standard errors to approximate confidence intervals for the mean value of X1 within each region:

```
DATA;
  SET A.REGMEAN;
  U_LIM=M1+1.96*S1;
  L_LIM=M1-1.96*S1;
PROC PLOT;
  PLOT M1*REGION='M'
       U_LIM*REGION='U'
       L_LIM*REGION='L'/ OVERLAY
         VPOS=30;
```

The resulting plot is shown below.



Normal probability plot of residuals: example 2

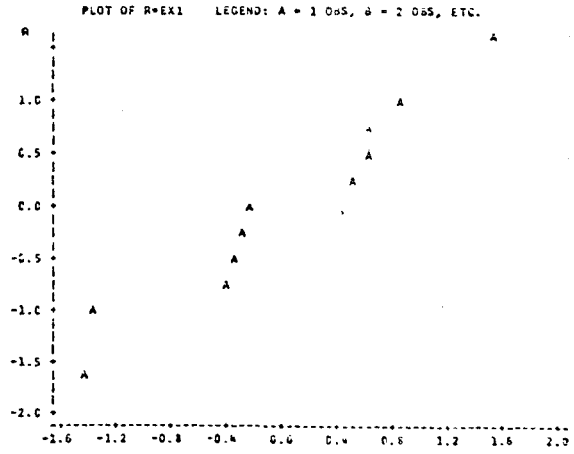
Suppose you want to study how your data fits the simple model X1=a+b*X3+c*X6. You can use GLM, outputting predicted and residual values into a SAS data set:

```
PROC GLM DATA=D;
  MODEL X1=X3 X6 /P;
  OUTPUT OUT=NEW1 PREDICTED=PX1
    RESIDUAL=EX1;
```

The predicted values of X1, called PX1, are stored in the SAS data set NEW1. Residual values, the differences X1-PX1, are given the name EX1. The P option on the MODEL statement requests that individual predicted values and residuals be printed, making it easy to spot numerically large residuals. A normal probability plot of the residuals is produced by:

```
PROC RANK NORMAL=BLOM   OUT=NEW2;
  VAR EX1;
  RANKS R;
PROC PLOT;
  PLOT R*EX1 / VPOS=30;
```

If the residuals follow a normal distribution, the points would approximately follow a straight line. The probability plot is shown below.



Iteratively reweighted least squares using NLIN

SAS's NLIN procedure can be used to perform iteratively reweighted least squares (IRLS), a technique that is used to obtain more robust estimates than those obtained using other least squares methods.

One class of weight functions that has been proposed uses a power of the residuals of the last iteration:

$W = ABS(RESIDUAL)^{P-2}$ *where P is between 1 and 2*

This weight function results in estimates that minimize the L_p norm of the residuals:

$$\underset{\beta}{MINIMIZE} \sum_{i=1}^{n} |y_i - X_i \beta|^p$$

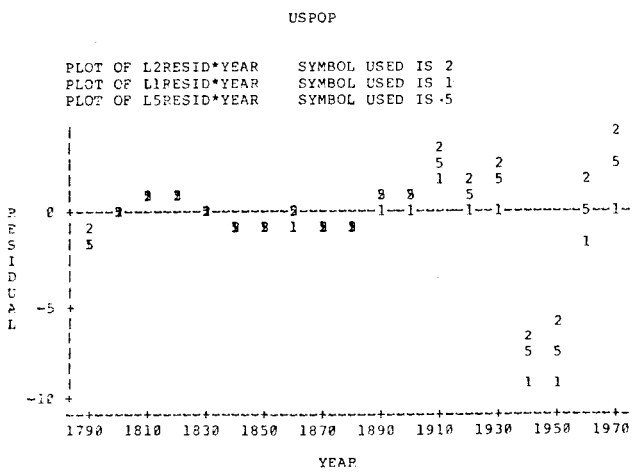
The L_2 norm estimates have weights equal to 1, and are the usual least squares estimates produced by GLM. The L_1 norm estimates minimize the sum of the absolute values of the residuals, and are the estimates computed by the user-contributed SAS procedure LAV. The $L_{1.5}$ norm estimates a compromise between the L_2 least squares estimates and the more robust L_1 LAV estimates.

The __WEIGHT__ statement in NLIN can be used to specify weights that are functions of the predicted or residual values from the previous iteration.

IRLS was used to compute the $L_{1.5}$ estimates of a quadratic time trend model of the United States population from 1790 to 1970. In order to compare estimates, L_2 estimates were computed using GLM, and L_1 estimates using LAV:

```
PROC GLM;
  MODEL POP = YEAR YEARSQ;
PROC LAV;
  MODEL POP = YEAR YEARSQ;
PROC NLIN;
  PARMS B0 = 0   B1 = 0   B2 = 0;
  FITTED = B0 + B1*YEAR + B2*YEARSQ;
  MODEL POP = FITTED;
  DER.B0 = 1;   DER.B1 = YEAR;
    DER.B2 = YEARSQ;
  __WEIGHT__ = 1/SQRT(ABS(POP - FITTED));
```

The IRLS process converged in 13 iterations. A residual plot was generated by running the procedures and creating output data sets, then merging the residual values. In the plot, L_1 residuals are marked "1", L_2 residuals are marked "2", and $L_{1.5}$ residuals are marked "5". Note the YEAR values of 1940 and 1950 were outliers in the population trend. These outliers influence the L_2 fit more than the other more robust fits.



For more information about iteratively reweighted least squares, see:

Holland and Welsch, "Robust Regression Using Iteratively Reweighted Least Squares," *Communications in Statistics* A6(9), 1977, pp 813-827.

and

Kennedy and Gentle, "Comparisons of Algorithms for Minimum L_p Norm Regression," *Tenth Annual Symposium on the Interface*, NBS Publication 503, 1977.

For details on the user-contributed LAV procedure, see Gentle and Lee, SAS Technical Report S-101.

Consultant's corner

Q: I am writing a SAS program to produce a weekly report. Each week, the job reads data from a file and creates separate SAS data sets, each with a different value of the variable TYPE. When I run this job, some values of TYPE will not be present in that week's data file. I still want to print a report, but with a message saying that there are no records for that week. How can I handle this problem using SAS?

A: Consider the following job:

```
//jobname JOB acct, name
//   EXEC SAS
//IN DD DSN=...
DATA A B C;
  INFILE IN;
  INPUT TYPE $1.   ...;
  IF TYPE='A' THEN OUTPUT A;
  IF TYPE='B' THEN OUTPUT B;
  IF TYPE='C' THEN OUTPUT C;
DATA __NULL__;
  FILE PRINT;
  IF EOF AND __N__=1 THEN
    PUT 'NO OBSERVATIONS OF TYPE=A';
  SET A END=EOF;
  PUT ...;
```

similar report-writing steps for data sets B and C

In this example, three SAS data sets are created from the weekly file. If there are no TYPE=A observations in a given week, the SAS data set A will be empty. The END= option causes the variable EOF to be set to 1 when the next execution of the SET statement will fail, and so when there are no observations the value of EOF is set to 1 initially. Thus, when both EOF=1 and __N__=1, the message 'NO OBSERVATIONS OF TYPE=A' is printed on the report.

This technique works for all methods of input: SET, MERGE, UPDATE, and INFILE. The rule is:

> SAS continues to execute statements in a DATA step until a SET, MERGE, UPDATE, or INPUT statement fails because the end of the data set or file has been reached.

Staff changes

As SAS's popularity grows, the need for new SAS staff members increases, and several people have recently joined the SAS staff.

Heading the Education and Support Division is Dr. Herbert Kirk, who last year taught SAS short courses on a part-time basis. Herb's Ph.D. in statistics and horticulture is from North Carolina State University, where he was an assistant professor in the Department of Statistics.

Joining Herb in the Education Division is Ginger Kelly, most recently responsible for packaged software support—including SAS—at Rice University. Ginger's undergraduate degree in physics is from Rice, and she has also done graduate work in biostatistics at UNC-Chapel Hill.

The use of SAS for financial applications is growing, and Loren Harrell has joined the Applications Division, where design is underway for a financial reporting system. Loren has a B.S. in computer science from NCSU, and for the past five years has been a technical representative for General Electric Information Services Company.

Mary Edeburn, who has joined the SAS marketing staff, was formerly a marketing representative with Dustin, Inc., which develops systems for Wang computers. Mary, who has an English degree from Marshall University, also has programming and data analysis experience.

Anthony J. Barr, whose concept of a total system for data analysis became the framework of SAS, and who single-handedly designed and implemented the SAS parsing mechanism and the supervisor, has begun a new venture called Barr Systems, Inc. Jim leaves SAS after twelve years, taking with him our best wishes for success in the future.

Daniel Chilko, who spent a year's sabbatical with SAS Institute, has returned to West Virginia University. In addition to teaching and consulting at SAS, Dan wrote several SAS procedures and functions, including the new UNIVARIATE procedure in SAS•79.

678 installations

SAS ***communications*** is published quarterly by SAS Institute Inc.

Address all correspondence to:

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

U.S. Postage
PAID
Raleigh, N.C.
Permit No. 977

MR. FRANK DIORIO
MANNING HALL
IRSS
UNC-CH
CHAPEL HILL, NC 27514

SUGI program issue

SAS communications

SAS Institute Inc. • PO Box 10066 • Raleigh, NC 27605 USA • telex 802505 • 919/834-4381
winter 1979 • volume iv number 3

power users

SUGI specialty groups

A special meeting is planned at SUGI '79 for SAS users involved in the utility industry. Joe Donaghy, Senior Scientist at Carolina Power & Light Company, writes:

Dear SAS User:

Nat Wooding at Virginia Electric and Power Company and I have expressed an interest to SAS Institute in forming a group of utility industry SAS users.

One specific interest is environmental applications, including statistical, data management, and SAS problems. We feel, however, that other areas in the utility industry could benefit from such a group.

We would like to meet with you on Monday evening after the mixer at the January 29-31 SAS conference to discuss the formation of such a group and any subgroups, such as environmental applications. We also suggest that you bring examples of your SAS usages, any statistical problems, etc. for an informal discussion period.

Yours very truly,
Joe Donaghy

For more information about the meeting, contact Joe or Nat:

Joe Donaghy
Senior Scientist, CP&L
P.O. Box 1551
Raleigh, NC 27602
(919/836-7502)

Nat Wooding
Environmental Services, VEPCO
P.O. Box 26666
Richmond, VA 23261
(804/771-4457)

conference

SAS users in Clearwater

Preparations for the SAS Users Group, International (SUGI) conference to be held January 29-31 in Clearwater Beach, Florida are well underway.

If you haven't already mailed your conference registration, send it today. Don't forget to make your hotel reservations directly to the Sheraton Sand-Key.

If you haven't received your conference brochure and registration form, give SAS Institute a call and we'll get it to you immediately.

Co-chairmen William Wilson and Raymond Littell report that over eighty papers have been accepted for presentation at the conference. These papers are listed below.

Papers accepted for SUGI '79:

Linear Models

Dal Kratzer, The Upjohn Company, Chairman

Invited Papers

S.R. Searle, Cornell University
Relationships between the Estimable Functions of the SAS GLM Output for Unbalanced Data and the Hypothesis Tested by Traditional-style F-tests

J.H. Goodnight, SAS Institute
New Features in GLM and VARCOMP

Discussants:
William L. Sanders, University of Tennessee
F.M. Speed, Mississippi State University

Contributed Papers

Carlos A. Gonzalez, EMBRAPA
SAS Macros for Analyzing Growth Curves

Jerome Busemeyer,
University of South Carolina
Coding Subjects for Analyzing Unbalanced Repeated Measures Designs in SAS

Emilio A. Icaza and Kenneth L. Koonce,
Louisiana State University
Using U-Model Techniques to Test Hypotheses in the Linear Model

Ronald W. Helms and Imogene McCanless,
University of North Carolina at Chapel Hill
Using GLMM5 for General Linear Models Analysis

David Christiansen and Ronald W. Helms,
University of North Carolina at Chapel Hill
Plotting Confidence Bounds for Parameters for General Linear Models: A GLMM5 Case Study

James D. Hosking and Ronald W. Helms,
University of North Carolina at Chapel Hill
Analyzing Large Data Sets Using PROC SSCP and GLMM5

Agam N. Sinha and Kenneth Hardy,
University of North Carolina at Chapel Hill
A SAS Macro for Ridge Analysis of Multivariate General Linear Models

S.G. Carmer and W.T. Hsieh,
University of Illinois
Exploring Biased Regression with SAS

Shawki A. Salem,
Smith Kline & French Laboratories
SAS in the Analysis of Categorical Data

James E. Dunn and Gloria Cappy,
University of Arkansas
A Class of Invertible Functions for Analysis of Categorical Response Using Linear Models

Keith Muller, James D. Hosking, and Ronald W. Helms, University of North Carolina at Chapel Hill
Using GLMM5 as a Tool for Simultaneous Study of Structure of the Covariance Matrix and Experimental Design Effects

Ronald W. Helms, David Christiansen, and James D. Hosking, University of North Carolina at Chapel Hill
The GLMM5 Algorithms

Paul Stewart, Imogene McCanless, and Ronald W. Helms, University of North Carolina at Chapel Hill
GLMM5 Analysis of Growth and Dose-Response Curves

Business Applications

Jim Guthrie, The Cleveland Trust Company, Chairman

Invited Papers

H.W. "Barry" Merrill,
Sun Information Services
The Analysis of SMF and RMF Data with SAS

Robert L. Anderson, Milliken & Company
Use of SAS for Corporate Operations Research

Contributed Papers

Jan Curley and Douglas Garber,
Ford Motor Company
Analyzing and Reporting Employee Attitude Survey Findings in a Large Multi-Divisional Company

Franklin Young, United Airlines
Estimating Airline Passenger Demands from Truncated Samples

William Ingram, University of Florida
PAMS: A Project Activity Management System Implemented in SAS

Rodney H. Strand,
Oak Ridge National Laboratory
Cost Accounting of an RJE Station with SAS

William M. Parker,
North Central Texas Council of Governments
Providing Data Processing Support with SAS

Edward L. Spitznagel,
Washington University
Interfacing SAS with Mark IV

William R. Gjertsen, SAS Institute
Graphic, Calendar, and U.S. Map Reports for Various CPE and Marketing Applications

Walter J. Guthrie,
The Cleveland Trust Company
A SAS Program to Graphically Schedule the Tasks Assigned to Each Employee

Steven R. Borbash and Arup K. Mallik,
West Virginia University
A SAS Procedure for Capital Investment Analysis

Nancy Begin,
Public Employment Relations Commission, and Barbara E. Kemmerer, Rutgers University
Development and Implementation of a Computerized Case Management System

David J. Cowen and Sandra T. Cowen,
University of South Carolina
SAS-VSPC Interface for Handling Library Functions

Jean B. Holzer, University of Florida
File Definition Macros for Census Public Use Samples

Statistics

Irving K. Hwang, Merck Sharp & Dohme Research Labs, Chairman

Invited Papers

A. Lawrence Gould,
Merck Sharp & Dohme Research Labs
Multivariate Multiple Comparisons Using SAS

Bernard F. McDonagh, 3M Company
Statistical Applications Using SAS at 3M Company

Contributed Papers

John P. Sall, SAS Institute
Procedures for Dynamic Nonlinear Simultaneous Equation Models

Karen Kral, Thomas P. Capizzi, and R.D. Small, Academy of Natural Sciences of Philadelphia
Estimating the Parameters of Biological Models Using SAS

Chandu M. Patel, CIBA-GEIGY Corporation
Application of Stepwise MANOVA and MANCOV Using SAS

Donald Henderson,
U.S. Department of Agriculture
Trimmed Means, Medians, and Robust Tests of Homogeneity of Variance

William G. Jackson, Jr. and Richard C. McNee, USAF School of Aerospace Medicine
PROC REPT1FMD—A SAS Procedure for Single Factor Repeated Measurements with Missing Data

Frederick W. Morgan, L. Andy Litteral, and Richard Nelson, Clemson University
The LIFETEST Procedures for Analyzing Reliability Data

Agam N. Sinha and Kenneth A. Hardy,
University of North Carolina at Chapel Hill
A SAS Macro for Plotting Pair Charts and Obtaining Certain Two Sample Statistics

Mark McBride and R. Parks,
Washington University
The Procedure DURBIN

Charles McElrea, Academy of Natural Sciences of Philadelphia
A SAS Macro for Robust Multiple Regression

Robert Rogers and Sandra Novinger,
U.S. Department of Agriculture Forest Service
A SAS Macro for a Distribution-Free Test for the Parallelism of Two Regression Lines

James P. Summe and R. Clifton Bailey,
Naval Medical Research Institute
SAS Procedures for Estimating the Parameters of the Makeham Survival Model

Edward R. Jones, Chevron Research Co.
DESIGN, a Macro for Constructing Optimum Experimental Designs

W.S. Cash and K. Martin, Vick R&D
A SAS Macro for Selecting A Random Sample of Data Points from a SAS Data Base

Richard E. Cooper,
U.S. Department of Agriculture
Responsibilities of a SAS User

Moacir Redroso, Jr., EMBRAPA
The CLMEANS Procedure

Research Data Management

Robert J. Bronstein, USV Pharmaceutical Corporation, Chairman

Invited Papers

John F. Duffy, ICI Americas, Inc.
The Critical Mass Concept

Herbert J. Kirk,
North Carolina State University
The Use of SAS in Administrative Computing at Both the University and Federal Levels

Rodney H. Strand,
Oak Ridge National Laboratory
Environmental Data: Management and Analysis Considerations

Contributed Papers

Michael P. Farrell and A. Dale Magoun,
Environmental Laboratory, U.S. Army Corps of Engineers
Graphical Report Generation Using PROC VIVIPLOT

Michael P. Farrell and A. Dale Magoun,
Environmental Laboratory, U.S. Army Corps of Engineers
Management of Evolving Ecological Data Sets with SAS

Louis G. Partridge, Academy of Natural Sciences of Philadelphia
Writing Simple SAS Procedures

William Ingram, College of Medicine, University of Florida
Implementing Relational Data Base Management Techniques in SAS

J. Philip Miller, Washington University
The Role of SAS in the Support of Clinical Studies in Medicine

John N. Hubbell, Jr., Asian Vegetable Research and Development Center
Development of a SAS Data Management System for the Germplasm Collection at the Asian Vegetable Research and Development Center (AVRDC)

Richard D. Langston and David H. Culver,
University of North Carolina at Chapel Hill
Retrieval of Value Labels from SAS Files for SAS Use and Creation and Maintenance of a SAS Format Library

Cynthia Deitz, Michael C. Choban, and C. William Cox, Jr., West Virginia University
Use of SAS in the Development of a Management Information System in a Social Service Agency

Marilyn Bodow, College of Medicine and Dentistry of New Jersey
Correcting and Updating EBCDIC Files with SAS

Tina Brezenoff, College of Medicine and Dentistry of New Jersey
Building a Research Data Base through Clerical Staff Using SAS

Dalton F. de Andrade, Gustavo M. Pimentel, Enedino C. da Silva, and Cicero A. de Silva, EMBRAPA
The Statistical Analysis System as Implemented at the Department of Quantitative Methods of EMBRAPA

Robert J. Bronstein,
USV Pharmaceutical Corporation
Modular Report Writing in a Clinical Environment Using SAS

Robert J. Bronstein,
USV Pharmaceutical Corporation
Automatic Job Scheduling Using SAS

Peter Svendsen, Mayo Foundation
An Unsophisticated and Easy-to-use Edit-Update Program in Support of Research Data

Medical Applications

Jane Abel, Abbott Laboratories, Chairman

Invited Papers

Harji Patel, CIBA-GEIGY Corporation
Analysis of Covariance of Incomplete Data in Experiments with Repeated Measures in Clinical Trials

Daniel H. Freeman, Jr., Michael B. Bracken, and Robert F. Elia, Yale University
A SAS-AUTO GRP Interface for the Analysis of Hospital Discharge Data

Carl Metzler, The Upjohn Company
An Evaluation of Nonlinear Regression Programs

Contributed Papers

M. Clinton Miller, III, Edmund Murphy, and Paul Underwood, Medical University of South Carolina
The South Carolina Papmobile Program: A SAS Application

David W. Johnson and R.L. Brunelle, Lilly Research Laboratories
A Multivariate Approach to the Analysis of a Repeated Measures Experiment Using SAS

Rodney H. Strand,
Oak Ridge National Laboratory
The Preparation and Publication of a Toxic Effects Document Using SAS and COM

Terry M. Therneau, Mayo Clinic
Procedures for the Analysis of Survival Data

Helen Wojnikisz, College of Medicine & Dentistry of New Jersey
Use of SAS Procedures to Establish Criteria for Determining Misclassified Disease Cases

Richard A. Enz,
Hoechst-Roussel Pharmaceuticals, Inc.
The Clinical Data Management System at Hoechst-Roussel Pharmaceuticals, Inc.

John L. Stedl, ICI Americas, Inc.
The Production of Automatic Randomization Schedules via SAS

Chiao Yeh, ICI Americas, Inc.
Testing for Trends in Drug Research Data Using SAS

William Ingram, University of Florida
A Tumor Registry Information System Application Developed in SAS

William Ingram, Warren Curry, and Dennis Lezotte, University of Florida
SAS Used for Data Management and Mathematical Analysis of a Flow Microflourometry Study

William L. Landrum, Wayne State University, and Delores Higgins, Michigan Cancer Foundation Registry
Maintenance of Cancer Patient Data with SAS

SAS in the University

Rudolph J. Freund, Texas A&M University, Chairman

Invited Papers

Thomas A. Bubolz and James E. Gentle, Iowa State University
The Role of Package Programs in Statistical Methods Courses

Kenneth L. Koonce and Emilio A. Icaza, Louisiana State University
Applications of SAS at Educational Institutions

Contributed Papers

Douglas Cockrell and Ione Cockrell, University of South Carolina
A Recode Facility for SAS

Edward L. Spitznagel, Jr., Washington University
SAS as a Management Tool for Course Registration and Grading

Susan E. Bengtson, Byron C. Lewis, and Richard Nelson, Clemson University
On-line System Report Generator

Susan E. Bengston, Byron C. Lewis, and Richard Nelson, Clemson University
Project Control System

Audi R. Brewton, University of Utah, and Douglas L. Anderton, Utah State Bureau of Health Statistics
A Short Note on Discrete Event Simulation in SAS

Phillip Gallagher, University of North Carolina at Chapel Hill
SAS in a Course for Beginning Computer Users

Hoke S. Hill and Jeannie N. Currin, Clemson University
Using SAS for Soil Test Reporting

SUGI *Helene Cavior, editor*

sasware index New entries

Below are descriptions of four macros which have been added to the SUGI SASware Index. The first listing of the SASware Index, which currently contains 14 entries, will be available at SUGI '79. If you cannot attend the conference but would like a copy of the listing, contact Helene Cavior, SUGI Editor. Helene's address and phone number are

Helene Enid Cavior
1921 Glenhaven Avenue
Walnut Creek, California 94595
(415/347-0721)

RECODE
macro

Recodes a specified value of indicated variables on all observations in a SAS data set to a second specified value (12 card images).

Don Henderson
Data Systems Application Division
National Agricultural Library Building
Beltsville, MD 20705

DATALOOK
macro

A series of macros for univariate descriptive statistics providing: (a) listing of moment and percentile information; (b) outliers (far-out values) report; and (c) one or more plots. Output is flexibly controlled by the user. Reference: SUGI '77 Proceedings (Revisions available from author.)

William R. Gjertsen
SAS Institute Inc.
P.O. Box 10066
Raleigh, NC 27605

KS1SAMP
macro

Kolmogorov-Smirnov one-sample test. Outputs sample size N, test statistic DSUP, and, when N GE 30, selected asymptotic critical values for determining p-value. Reference: SUGI '77 Proceedings

William R. Gjertsen

KS2SAMP
macro

Kolmogorov-Smirnov two-sample test. Outputs sample sizes N_1 and N_2, test statistic DSUP, and, when N_1 and N_2 GE 30, selected asymptotic critical values for determining p-values. Reference: SUGI 77 Proceedings

William R. Gjertsen

Reduction of GTF and SMF data?

If you are involved in the reduction of GTF and/or SMF data using SAS and would be willing to share your expertise with another SAS user, contact Richard Denny, British Columbia Hydro, 7th floor, 970 Burrard St., Vancouver, BC V6Z1Y3, Canada (606/663-2472).

626 installations

documentation
SAS technical reports

In the fall newsletter, we introduced the SAS Technical Report Series. These papers document work-in-progress, describe new supplemental procedures, and cover a variety of application areas.

Three new reports have been added to the series. These reports and their costs are listed below; mailing costs are extra. To order one or more reports, call or write the SAS Publications Department at P.O. Box 10066, Raleigh, N.C. 27605 (919/834-4381).

A-106 *Probability plotting.* Daniel M. Chilko, 18 pp., $2.48.

P-107 *The FUNCAT procedure.* John P. Sall, 28 pp., $3.08.

T-103 *A compiler that generates analytic derivatives.* John P. Sall, 13 pp., $2.18.

Note: The P-series reports describe features in experimental form that are not yet available in SAS.

best contributed papers
$25 award

A $25 award will be given to the speaker presenting the best contributed paper at each session at this year's SUGI conference. Judges at each session will include the session chairman and a member of the SAS staff. Best of luck to each of you presenting papers!

short courses
Los Angeles & D.C.

Courses in Los Angeles and Washington, D.C. open the 1979 SAS short course year. The first course, to be held at the Hyatt Regency Los Angeles, will be taught January 8-9. The Washington course will be February 15-16 at the Arlington Hyatt House.

Registration forms for these two courses have been sent to everyone on the *SAS Communications* mailing list. If you need additional information, please call us at 919/834-4381.

The next courses scheduled will be April 2-3 in Chicago and May 10-11 in Atlanta. Look for more information in the spring issue of *SAS Communications*.

SAS **communications** is published quarterly by SAS Institute Inc.

Address all correspondence to:

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

SAS Institute Inc.
P. O. Box 10066
Raleigh, N. C. 27605

U.S. Postage
PAID
Raleigh, N.C.
Permit No. 977

MR. KEN HARDY
INST. - RES. IN SOCIAL STUDIES
MANNING HALL
UNIV. OF NORTH CAROLINA
CHAPEL HILL, NC 27514