This paper is about our attempt to solve a pair of related problems. The problems are, first, how to encourage use of the total range of software systems we have available at our installation, and, second, how to implement new stand-alone programs in such a way as to optimize their availability and usefulness. We have taken the approach of utilizing SAS to solve both of these problems by first creating a single comprehensive communication system with SAS at its center and, second, by adopting a policy of attempting to put up all new stand-alone programs as SAS procedures with conventional SAS syntax.

The situation

As instructors and user consultants at the College of Business Administration, and the University of South Carolina generally, we face many users. They are very diverse in their computing needs and desires. Our users have ready access to a large and capably-run machine which supports a wide range of statistical software. (Computer Services runs an IBM 370/168, with 5 MEG real storage, 36 disk spindles, 7 tape drives, and various peripherals, run under VS2. It currently supports 20 RJE terminals and approximately 80 timesharing and data base terminals spread over 15 campuses and 6 state agencies. Timesharing consists of IMS, ITS, APL-SV with TSO and batch job-entry capability, and BASIC, but we do not have TSO. Most applications which require storage of more than a small amount of data use magnetic tape; a 3850 Mass Storage System is being installed and tested, and it is expected that most tape storage applications will be eventually shifted to this more convenient medium.)

A few of the major pieces of statistically related software are: SAS72 and SAS76, SPSS, OSIRIS, BMD and BMDP, TSP, RAPFE, IMSL and SSP, CALCOMP and TECHRONIX graphics, and various APL applications including STAMPAC2, ECONOMETRIC PLANNING LANGUAGE and BOX-JENKINS. Besides these, there is, of course, the usual wide range of incompletely documented stand-alone programs (mostly written in FORTRAN) of varying quality.

The first problem

Most of the above mentioned software products are "systems". That is, they deal, at least conceptually, with the whole range of statistical data processing problems. These include data transformation, data selection and merging, and creation of a dictionary of variable naming and labeling information. Generally a set of related software products are used within the system to accomplish a series of possible tasks, utilizing a more or less coherent syntax for specifying control information.

*************

The toolbox approach

SAS76
SAS72
OSIRIS
SPSS
BMD
BMDP
TSP SPECIALISTS
RAPFE
IMSL & SSP
MPS
APL
other programs

(Illustration 1)

These software products are to a degree overlapping and redundant—but only to a degree. They each have their unique advantages and disadvantages which match up, to various degrees, with the characteristics of a particular user's current and past problems. At
the point when we see the user he usually already has an idea what basic approach he wishes to take; the more sophisticated the user, the more wedded he generally is to the use of a particular tool. Most users think of these software items as individual elements in a vague jumbled toolbox. As in the first illustration the user sees himself as a "specialist" with respect to the use of one or a few of the available "tools". The other implements in the software collection are regarded as interesting and curious, but the total cost of using them, considering most especially the users own learning time, is generally regarded as too high to justify investigation. This reaction is in no way irrational; the time investment certainly can be high. But almost all users are open to suggestions for alternative tools if the data processing barriers to their use can be minimized, and if the ties the user has already built to his preferred package need not be abandoned.

As an instance of this, to a SAS user the "DO REPEAT" facility and the pairwise deletion treatment of missing values might be attractive options of the SPSS package and might make it desirable to temporarily abandon the use of SAS for a particular analysis step. This step might easily require learning the JCL for production of a temporary disk file, the formatting of the output file, the creation of SPSS control cards to read this raw data file, and the learning of the SPSS syntax for the execution of the dictionary creation step (no easy task for someone used to SAS syntax.) The crucial and difficult (and unnecessary) step is the re-creation of the dictionary information concerning naming, labeling and value labeling of the variables, and the reformatting necessary to fill the needs of the various systems.

******** A SAS CENTERED NETWORK WITH COMMUNICATIONS IN BOTH DIRECTIONS

A SAS CENTERED NETWORK

WITH

COMMUNICATIONS IN BOTH DIRECTIONS

THE SOLUTION-A SINGLE SYSTEM

Our response to this problem has been to attempt the creation of a single software system, with SAS as its focal point and central communications link. In the second illustration we see the previous list of software positioned around SAS, with communications links to SAS.

A SAS CENTERED NETWORK

For the solution of the problem of multiple systems, we have developed output links to go from SAS to the other systems, utilizing a few simple SAS statements and special JCL proc to pass data and dictionary information between SAS and those systems. In addition to the CONVERT procedure, we have developed our own input routines to go from other systems into SAS.

With communication links developed going in both directions, a whole new dimension in software integration is possible. The quantitative increase in communication links generates a qualitative change in our attitude towards the software collection. With linkages in both directions it not only becomes possible to easily go from SAS to SPSS, or from APL to SAS, but it then becomes equally easy to go from SPSS to APL or from APL to OSIRIS, using SAS as a powerful communications switching and data conversion point. SAS becomes analogous to the telephone switching station of your local phone network.

AN EXAMPLE-APL TO SPSS

The following example shows how a user could easily go from APL to SPSS. While the example involves a trivial amount of data, the point of the example should be clear. The first lines of code show the essence of the APL conversation necessary to send the data from APL to SAS; the first leg of the hypothetical journey from APL to SPSS. (In this conversation, user entered information is in lower case letters for clarity):

**\begin{verbatim}
\textbf{\texttt{\textbf{XNames}}}
\texttt{SALES}
\texttt{PRICE}
\texttt{ADVER}
\texttt{x}
\end{verbatim}\**

\begin{verbatim}
2 3 7
4 5 14
6 7 3
12 -999 25
\end{verbatim}

************** INVEST IN ANY ENTRY POINT-BUY ACCESS TO ENTIRE NETWORK (Illustration 2) **************
WHAT NAME DO YOU WISH TO GIVE THE DATA SET? test
DO YOU HAVE A NAME MATRIX? yes
WHAT IS THE NAME OF YOUR MATRIX?name

YOUR DATA SET HAS BEEN CREATED, TO ACCESS IT, PREPARE THE FOLLOWING CARDS:

 JOB CARD
 EXEC SAS
 APLIN DD DISP=(OLD,DELETE),
 DSN=TSIO.AAAABDOG.TEST
 SYSIN DD *
 DATA; PROC APLIN;

*

Your attention is directed to the "-999" in the last line of the matrix "x", this is to be considered a missing value code for our example. Note also the simple nature of the conversation, where naming information was requested. When not supplied, the routine creates names of the type "v1", "v2", etc. The next lines show how this information could be passed through SAS and out to SPSS in a few lines:

 JOB CARD
 EXEC SAS
 APLIN DD DISP=(OLD,DELETE),
 DSN=TSIO.AAAABDOG.TEST
 SYSIN DD *
 DATA; PROC APLIN MISSING=-999;
 PROC SPSSOUT MISSING=-888;
*
 SPSS.SYSIN DD *
 GET FILE NONAME
 REGRESSION VARIABLES=SALES PRICE ADVER/
 REGRSSION=SALES WITH PRICE
 ADVER(2)/
 OPTIONS 2
 FINISH
*

In these few short lines we see the user passing his information through SAS, changing his missing value code from -999 to -888, and going on to SPSS to take advantage of the pairwise deletion option of the SPSS REGRESSION program. The SAS/SPSS JCL procedure is designed to shield the user from the irrelevant details of creating and passing the SPSS disk file from SAS to SPSS.

MORE ON THE NETWORK

With this network concept it becomes possible, as in any network, for the user to decide where in the total system he wishes to make his main investment of time and expertise, confident that at any time he may utilize any of the other features of the network with minimal expense. Investment in any of the multiple entry points buys access to the entire system. Manual dictionary creation is a one time exercise, and variable names can retain their significance and identity from one system to another—a name is a name is a name. This allows user decisions concerning data storage, data manipulation and data analysis to be made independently; data could feasibly be stored as a sum total of many systems, (utilizing the freedom to decide how to store variables and the large upper limit on number of variables of the OSIRIS format), manipulated using SAS (using the sort, merge and update facilities), and analyzed using SPSS (using some special feature, such as pairwise deletion). For each such decision the user would be taking advantage of the unique merits of each system, while avoiding the restrictions imposed by limiting himself to one system only. Also, once such a network is implemented, a facility programmed in one system may not need to be programmed in another.

Of course, there is a transmission cost involved in going through SAS in order to transmit information from one system node to another. Looking at the second illustration, this cost is the extra computer time to rewrite the data into a SAS format and read the SAS file to rewrite it for the destination system. But this is just what a machine is designed to do: replace scarce machine time for scarce human labor. The alternative is either to design one shot direct links, reconstructing the dictionary at the destination, or to forge special-purpose programs to read the files of one system and write the files of another in one step. This latter proposal might be useful if any given path became well traveled (for instance, the OSIRIS to SPSS path is used extensively, thus the system used for the direct link found in the SPSS "OSIRIS VARS" facility), but in general there are too many potential internode links, with the number of nodes growing constantly, to justify this (in general, it is the number of permutations of n systems taken 2 at a time; with 2 systems—2 paths, 3 systems—6 paths, 10 systems—90 paths).
THE SECOND PROBLEM

The problem of communication and labor time in going between systems described above is especially significant when it comes to the use of stand-alone programs. Not being imbedded in a system, these programs have their own unique overhead of implementation tasks. And it is a problem which continually grows, relieved by the steady expansion of the capabilities of the SAS, and SPSS type systems and fed by the steady source of new applications and new wrinkles on old applications. New stand-alone programs flow into our offices in a steady stream, accompanied by requests for implementation, and promises that "this will solve all of my problems if you can get it working!" These promises must be taken seriously since they sometimes come true. At which point the toolbox grows by one and the number of incomplete intersystem communication paths to this new program from the other systems grows by the number of systems. The problem is how to make the barriers to use minimal, so that other individuals besides the person who originated the request will not rule out either experimentation or use of a valid tool because of communication problems. But even more than the problem of communication between coherent systems, communication to and between individual programs offers unique problems. Some of these are fixed format control cards, irrational ordering restrictions on data, various labeling conventions (or worse yet, no variable labeling at all). Others have irritating implementation restrictions, (usually traceable to the use of FORTRAN static storage), and require various JCL tricks to get information to the program in a manner designed to make the program think it is getting all information from a deck of cards.

The result of barriers such as these is that we see the phenomena of persons treating programs as mysterious black boxes. Black boxes can be only partially opened by requiring publication and explanation of the theory and algorithm. Most users require experience with the working program to generate a full understanding of the tool. But if this user of the tool requires the commitment of a major dose of training time and expertise, the incentive to experiment may be overwhelmed. For instance, it has been demonstrated that the use of linear discriminant analysis is not optimal if the assumption of homogenous covariance matrices is rejected by means of a test. The Eisenhiae and Avery program MULDIS offers quadratic discriminant analysis and classification as a viable solution for this problem. However, the program defies causal use.

A GROWING SAS SYSTEM

For this second of the two problems, implementation of new applications, we have attempted to integrate the new applications into SAS. We have not attempted to reform all of the stand-alone programs which currently exist outside of our system. That task would be too big, and is being done steadily by the staffs and contributors to the larger systems. Instead, we have concentrated on the new applications or on those applications to which strong requests for reform have been directed. For instance, the MULDIS proc was adapted from the MULDIS stand-alone program in response to requests from several users complaining of the fallings of currently available discriminant procedures. The same is true of the three BOX-JENKINS procs and the ongoing work on PROBIT and LOGIT programs. The LP procedure was written to bring the powerful facilities of the Mathematical Programming System (MPS) within the understanding of the most casual student user.

As an example of this kind of work, the following control cards will generate a solution to a typical linear programming problem. The MPS (Mathematical Programming System) option and the SASMPS JCL proc cause the MPS control cards to be generated, passed to MPS and the MPS solution printed. Omitting the MPS option causes an internal routine to solve the problem and print out a less detailed solution, more suited to introductory students.

```plaintext
// JOB CARD
// EXEC SASMPS
DATA:;
PROC LP MINIMIZE MPS;
PARMCARDS;
P=3X1+2X2+5X3+4X4/
SUBJECT TO /
X2+X3<100/
4X1- X2<50/
3X1+X2-X3<10/
X3<50/
X1+X4>5/
X1+X4<10/
/
/
```

WHY A SINGLE SYSTEM?

With our single system concept, we thus see again the usage of the SINGLE DICTIONARY CONCEPT where a variable need be named, labeled, and value labeled only once and is available thereafter. Once in the center of our system, we
have the advantages of a central filing spot for all our software, with simple documentation standards, utilizing a simple clear syntax well known by our students and other users. In addition, when writing our programs we need not make provision for data storage, data transformation or data cleaning, limiting our analysis to the manipulation of clean standard matrices. We have thus separated application programming considerations from the data storage and data handling problems which so hamper the stand-alone program.

WHY SAS AT THE CENTER?

For handling these two problems, why should we place SAS at the center of the communications net?

SAS is an ideal software system. It utilizes a standard syntax. Always user oriented, it is possible to design new procedures with rational defaults which minimize the user's control card preparation burden without denying him full flexibility. For instance in the AUSCOU procedure, (to go from SAS to APL) the absence of the variables list is a signal that all of the variables are to be passed out to the APL system. This is, of course, the usual SAS convention. On the other hand, a large OSIRIS file with 8000 variables can be read with OSIRISEM by means of a set of free format punchcards which are scanned (using the syntax of SPSS, OSIRIS, or SAS) to obtain the names of the variables which are to be selected out to fit within the confines of the SAS system, giving the flexibility necessary to utilize such a large file within SAS. Because our users are used to the SAS convention of no variables list implying use of all variables and the flexibility of free format variables lists, it is easy to show a new user how to utilize these facilities.

At the same time, special JCL cataloged procedures can be placed on the system to serve the special needs of the user needing to do inter system communication, creating the necessary temporary disk files for data transmission, (JCL procs are cheap.) Thus we have the SAMS proc to enable the PROC LP user to access MPS with three cards. As shown in the example presented earlier, we can have a SASSPSS proc to speed information between SAS and SPSS and a SPSSSAS proc to get back again. We could even have a SPSSOSIRIS to go between SPSS and OSIRIS. All of these JCL procs would be tailored (like the FORTAN special procs FORTGC, FORTLG, FORTCL, etc.) for a particular purpose, no one of which was of any particular importance except for communication to the user.

SAS is one of the most powerful data manipulators available. All of this power can be brought to bear upon the data with which the new application program is to be used, without the necessity to explicitly include any of this power within the program. This was one of the most important considerations in deciding to utilize SAS to implement MULTIS. All of the data manipulation could be done ahead of time, within SAS, before accessing PROC MULTIS. In addition, there is no necessity to write the programs as multiple pass operations, since if the user needs to make multiple runs, he can give multiple invocations of the PROC statement. This can considerably simplify and clean up the conversion of stand-alone programs, as we have found while working the PRRTT procedures. Each new procedure can be designed as a single operator with which to preform a well-defined action upon a single well-defined data source.

As a data base manipulator, SAS obeys the principles of the "RELATIONAL" data base. That is, it portrays the relationships between observations by including in a single data set variables which have values which vary from one observation to the next and other variables whose values are constant over subgroups of observations. The SAS language allows very flexible selection and merging of information controlled by the values of this subsetting information. This advantage, while not unique, seems to be unique for statistical data processors.

It is easy to program new applications. This is no trivial statement. Development of the work done by us at the College of Business Administration was done by separate groups. Each subgroup did their development in separate libraries. There was little explicit coordination between groups and no coordination of effort with Computer Services, except for the provision of raw resources (disk space, computer time, willingness to place new cataloged JCL procedures on the system). No changes were made to the SAS source as it came from Raleigh. Updates to the SAS system and our work could be and were done completely independently. This was important, considering the amount of experimentation and the number of changes made. Thus we have all of the flexibility of independent researchers but the end result is as powerful as if we worked hand in glove with a large centralized staff. For after we finished testing a new proc, we placed it in a central library we maintained, which was concatenated to the STEPLIB of the usual SAS libraries. Simple. Even simpler is the prospect of using
standard FORTRAN subroutines from standard sources. For instance, several procedures we have worked on have utilized the IMSL, Library of FORTRAN subroutines to perform sensitive calculations. This is true of the LP proc, the NLPE proc and PRINCP. To keep these procedures up to date with the latest numerical analysis technology, the only thing we have to do is to periodically recompile and relink our source programs with the newest version of the IMSL load-module libraries. Static storage implementation restrictions can be minimized by either writing the routines in PL/I or by utilizing the storage allocation routines of SAS to get space for the variable dimension subroutines (such as the IMSL routines). In addition to this, we generally found that the subroutine calls and the keyword, option, variable lists and parameters of control information were powerful, complete, and easy to document.

As a system the multiple data set orientation of SAS gives it the power it needs to be catholic in its approach to other systems. This facility to handle multiple data sets (as opposed to, say, SPSS or BMDP.) combined with the ease of fitting in new procedures makes it possible to easily expand the system to communicate with other systems.

Finally, all of this can be done and supported with a small staff. For the past two years, most of the work was done by two persons, neither one full time (though there were times when full time work was required.) The key, of course, is the fact that we could build upon a large body of user familiarity with SAS. We are still building upon this base.

HOW ARE WE DOING?

It is a familiar story, but our task is as yet uncompleted. There are some communication links still unmade, though all of the major ones between SAS72, SAS76, SPSS, OSTRIS, BMDP and APL are in and working. We have had our share of implementation problems with the links we have forged. OSTRISIN is slow because of all the data conversions required, but it does work with the largest OSTRIS files. The SPSS links (written for SAS72) do not handle VARIABLE or VALUE LABELS. Currently, the conventions for the free format input lists used by the input routines are not completely compatible from one routine to another. SAS72 would not let us get alphanumeric names for DNNAMES from the PROC statement parameters. There are some other minor restrictions buried in a few of the routines. On the other hand, all of the programs are top-down structured programs, without GOTOs except LP where the use of GOTOs and label variables was the only practical method available. All of our procedures have been tested and subjected to user abuse.

The picture with respect to the inclusion of stand alone programs is not as complete. These programs have, up to now, been given lower priority than the intersystem communication linkages. Also, since some of the new applications imply the conversion of a current program, there is always the problem of understanding the contents of that program. That is hard work. Since there is always a long line of users who would like to have their programs worked on, we have at times been guilty of making more promises than we could keep with respect to including new capabilities in SAS. One commitment we have had to abandon at times is the commitment to put up "all" new applications in SAS. In fact all of the new applications that we have written from scratch are included in SAS. But several useful programs are still available only as stand-alone programs.

THE FUTURE

Some major tasks we would like to complete or start in the immediate future are: 1. Finish converting our SAS proc from SAS72 to SAS76. 2. Put up PROBIT and LOGIT procedures. 3. Implement Von Bard's NLPE program for estimation of nonlinear systems of equations. 4. Set up procs to easily do scatter plots and histograms on the flatbed or drum plotter. 5. Build a link to the COMPOSTAT data sets of information on large corporations (these data sets are very frequently used within the College of Business, and analysis often requires switching from cross-section to joint cross-section time-series modes of analysis). 6. Build links to the TIP and RAPPE systems of time series analysis programs. 7. Build a microfiche procedure to enable SAS users to send output directly to microfiche. 8. Enable SAS users to directly execute selected IMSL subroutines.

CONCLUSION

We have shown by our experience that SAS can be a practical programming tool to weave together software packages and new statistical tools into a single communications network. Because of the simple nature of the resulting user instructions, it can be truly said that the whole is greater than the sum of its parts.
SOFTWARE BALLOT SUMMARY

The sum of responses (number in parentheses) was taken to be the measure of question importance. Questions are ordered on decreasing value of this sum. The ballot was completed by 117 conference attendees.

Response:
1. The item has very low priority; it is of virtually no importance. (Lowest 20 percent.)
2. The item has low priority; there are others which have lower priority, but a large majority of the items on the ballot have higher priority. (20th-40th percentiles.)
3. The item has about average priority; about half the items are more important and about half are less important. (Middle 20 percent of the items.)
4. This item has higher priority than most of the ballot but is not in the top 20 percent. (60th-80th percentiles.)
5. This item is in the highest priority group; the SAS Institute should implement this suggestion (or solve this problem) as quickly as possible. (80th-100th percentiles of priority ratings.)

(Blank) - Leave the space blank if you have no opinion on the item or do not wish to express an opinion/rating.

(443) Need Do loops, Do groups.
(392) Need to provide an option for subscripted variables. Consider the following example:
T(1) = 'ATLANTAFALCONS';
T(2) = 'BaltimoreColts';
T(3) = 'CHICAGOBEARS';
TEAM = 3;
PUT & 1 T(Team);
CHICAGOBEARS

(389) Need a function to change a character value into numeric when the characters are valid numbers and the reverse.
e.g. CH = '76';
NU = FUNC(CH);
so NU = 76

(399) Need more ability to output statistics that procedures compute. That is output variables to a SAS data set.

(371) Need IF THEN ELSE Statement
(369) Statements of form XI-X10 = LOG(OP Y1-Y10) would be very useful.

(364) Need more non-parametric statistics, such as: median, interquartile range; Wilcoxon rank-sum tests for distinct distributions; Wilson's ANOVA.

(363) Need to implement SAS76 data set storage on tape; i.e., make it possible to copy a SAS76 data set directly to tape, without conversion to EBCDIC, and reread the data set.

(332) Need constant PROC PRINT format. PROC PRINT re-evaluates formats with each change in a BY variable. In many instances, this procedure causes several different formats on one page making the printout more difficult to read.

(329) Need to add median and parametric and non-parametric percentile estimates to PROC MEANS.

(317) Need a user-friendly organization of User’s Proc’s and their distribution and organized documentation of users library.

(316) Need to expand PROC GLM to generate contrasts for single degree of freedom comparisons. (e.g. LINEAR, QUADRATIC, CUBIC, etc.). Need to add option to output selected mean square terms, means, and adjusted means.

(316) Need variable labels in PROC PRINT.
(315) Need a DO WHILE statement.

(314) Need specifics on how SAS76 stores data sets.

(309) Include in "PROC PRINT" the option of TOTALS for all numeric variables unless a VAR statement is included. When the option is coded, the totals should be obtained in relation to "BY" variables. Perhaps an overall total at the bottom. If option is not coded, "PROC PRINT" should print standard output. Example:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OBS</td>
<td>BY VAR A = 1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>OBS</td>
<td>BY VAR A = 2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVERALL TOTAL</td>
</tr>
</tbody>
</table>

(308) In PROC MEANS, the addition of a confidence interval option would prove to be very time saving.

(307) All of the five regression packages need to have an output data set that would contain coefficient estimates, the variance-covariance matrix of estimates, MSE, R2 F and possibly other relevant information.

(306) Need to provide an option in GLM to test the usual assumption that the residuals are independent and follow a normal distribution. In addition, normal probability plots of residuals should be supplied (see BMDP7M, 1975).

(306) Description of GLM needs improvement. The manual is written for statisticians, not for applied users, not even for reasonably sophisticated applied users. Nor in this merely a matter of the level of sophistication of the terminology. To statisticians different disciplines use different terminologies. In this section on GLM, virtually no effort has been made to bridge this gap. This is particularly unfortunate since it is, after all, a "General Linear Model". Why it should be discussed using highly specific terminology alone is hard to fathom. Here are some examples:
(219) Add an option to PROC GLM for regression
(282) Need to expand the analysis of categorical

Here is one that may seem silly; but

virtually no information is given on the

and/or multiple range tests using mean

options to perform multiple comparisons

The print format for numeric variables in

PROCFREQ needs to be consistent. For

For example, it now prints:

```
  6.666667
  11.25
  12.41667
  40.5
  44
```

The first 44 pages need to be expanded and

the number of examples need to be tripled.

(202) Need more documentation of missing

(207) Need symbolics in Macros.

(209) Need a RSM procedure - to

(209) Need Macro storage.

(213) Need missing values in Matrix.

(218) Need to provide automatic contour plots.

(218) Use SAS under TSO, PROC MATRIX should

(220) Need to implement macro libraries, and do

it in such a way that macros are not auto-
matically printed.

(224) Need substring retrieval.

(229) Need more output capability in the form of

machine readable data files, i.e. means,

SS, correlation matrices, etc.

(234) Need to include variable labels in GLM.

(235) PROC PRINT; BY GROUP; currently does not

restart the observation counter with each

BY group. It would be useful if it did.

(239) Need option to suppress the SAS notes that

are interleaved with SAS statements.

(229) Need Global variable transformations in

compiler (_ALL_ = LOG(_ALL_);) with

attached exclude list.

(232) Need more output capability in the form of

PRINT statements,

(243) Need option to suppress the SAS notes that

are interleaved with SAS statements.

(249) Need to include variable labels in GLM.

(250) Need to implement macro libraries, and do

it in such a way that macros are not auto-
matically printed.

(252) Need more output capability in the form of

machine readable data files, i.e. means,

SS, correlation matrices, etc.

(254) Need option to suppress the SAS notes that

are interleaved with SAS statements.

(266) Need Array manipulation.

(269) When a macro is involved, make printing of

the macro text (in place) possible. At

present, only the macro name appears when

the macro is involved. One could have the

format

```
macro_name < LIST |
  NO LIST >
```

with NO_LIST as default if neither is

specified.

(270) Need more documentation of missing values—

the treatment of missing values is both

inadequate and poorly described. One has to

dig through procedures and finally finds out

that something called MANOVA is what is

needed, but even the write-up of MANOVA does

not communicate well exactly what it is.
Need labeling in Matrix.

Need to allow missing independent variables.

Need Procedure Rename statement as

LAGS - need easier creation of lagged

Need Variance-Covariance Matrix for the

Need labeling in Matrix.

Need to compute EMS in balanced data.

A feature COUNT,variable_name would be

The Normal and UNIFORM

New Qstimation procedures --

In PROe MEANS need to carry along variables

It would be very useful to be able to call

New estimation procedures --

Need Multi-dimensional scaling.

The DUNCAN's Multiple Range Procedure needs

In converting an SPSS file, if more than one

Useful

Need MANOVA -- given discriminant functions

New estimation procedures --

In PROC MEANS need to carry along variables

It would be very useful to be able to call

New estimation procedures -- Box-Jenkins

Need to be able to specify order of values

in FREQ procedure -- do not always want

them sorted.

Need to allow missing independent variables

in regression - most of the necessary
algebra has been done and may be found in:
Glasser, M., "Linear Regression with
Missing Observations Among the Independent

Need Variance-Covariance Matrix for the

estimates in GLM.

Need to be able to specify

Need to allow missing

also be found in:

Glasser, M., "Linear Regression with
Missing Observations Among the Independent

Need to compute EMS in balanced data.

A feature COUNT,variable_name would be

useful in conjunction with the SET and
MERGE when using BY variables. E.G.

FAMILY          COUNT,FAMILY
     A            1
     B            2

LAGS - need easier creation of lagged

variables; for example, in ESP (Econometric
Software Package) the form of the lag
creation statement is

XLAG1 = X(-1)  (1)

or

XLAG10 = X(-10) (2)

or

XLEAD5 = X(+5) (3)

XLAG1 is then X lagged one period, XLAG10 is X lagged 10 periods, XLEAD5 is X with 5
period lead. This, or something like it, would facilitate lagging and leading.

Need SPSS - like REGECU.

Need Procedure Rename statement as

described in the SAS Supplementary Pro­
cedures Guide of April, 1974.

The Normal and UNIFORM distributions should be

improved.

Need TABLE LOOK UP.

When using DATA _NULL, it would be useful

to know how many observations would be in
the data set if it were a non-_NULL_ data
set.

Need to permit saving/printing of (X'X)\(^{-1}\)

for regression/ANOVA and calculation of
Confidence Intervals at specific points.

Need to improve labeling capability.

Need a "DO" facility in compiler.

Need Ridge Regression.

Need an OPTION NOPRINT;

Need Multi-dimensional scaling. This would
be useful in the agricultural sciences now
as it has been historically in the social
sciences; since the data often has shown
similar configuration.

The DUNCAN's Multiple Range Procedure needs
improvement. The SAS76 procedures for
DUNCAN's Multiple Range or similar mean
comparison methods as provided by DSAD of
ARS, USDA are much less useful and much
more subject to errors than those used
with SAS72.

In PROC MEANS need to carry along variables
like was done in SAS72.

It would be very useful to be able to call
up SPSS procedures within SAS because of
some superior SPSS options.

New estimation procedures -- Box-Jenkins
time series analysis.
Need PROC XTAB to compute crosstabs where cells can be any combination of simple statistics (PROB, MEANS) (like VW's STATJOB).

New estimation procedures -- Random coefficients estimation.

DO NOT USE a new line for the DATE -- ABBREVIATE the date if need be, but do not use a new line, as sometimes happens now.

Need PROC LP (Interactive optimization) (Linear/Non-Linear Programming).

Cut the line skipped with a MACRO name in DATA section.

Need to allow the use of statement labels within a macro (avoid duplicate labels).

Need PROC EDIT (Apply tabular range check to selected variables. Flag and Print Errors.)

Need to add list to Service Bureaus with SAS.

OUTPUT DATA sets -- CLUSTER - create an output data set so that the "cluster" can be removed from it - could do it as follows:

OBSERVATIONS corresponded to the N of groups, observation 1 corresponds to placing original observations in 5 groups - VARIABLES are named CL001 to CLN where N is number of OBSERVATIONS the value of the i-th observation of this data set for variable CLj is the group # that original observation j belonged to when there were i groups - e.g., suppose there were four observations originally. The data set might appear as:

<table>
<thead>
<tr>
<th>OBS</th>
<th>CL1</th>
<th>CL2</th>
<th>CL3</th>
<th>CL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

This data set could then be used, for example, with another PROC or MATRIX to produce the output obtained by specifying NCLUSTERS = on the cluster PROC statement.

Need to add proc EXPLODE to SAS76, if not already present.

LAGS - need distribution lag procedure such as a generalized ALMON procedure with options on how to construct the Lagrangian interpolation coefficients - output would be estimates of the "ALMON variable" coefficients, the "biased" estimates of original lagged variable, sum of the coefficients, etc.

Need one sequential output data set for all PUT statements in the whole program.

APL Link to SAS.

Need a way to show the structure of control more clearly; e.g.:

Label: BEGIN;

statements

END Label;

Need to expand Data Set directory to include basic statistics, variable codes and labels, and marginal frequencies.

Would like to see MINITAB interface, similar to PROC BMDP.

Need compatibility with the Xerox Sigma 6 System.

Need one sequential output data set for all PUT statements in the whole program.

Need compatibility with the Xerox Sigma 6 System.