Ask and Ye Shall Receive: Getting the Most from SAS-L

Howard Schreier, U.S. Dept. of Commerce, Washington DC

ABSTRACT

SAS-L is a worldwide online community of SAS® software users. Participants discuss various aspects of SAS and help one another solve SAS-related problems. This paper offers a number of suggestions, primarily directed at those asking questions on SAS-L. Following these suggestions may make questions clearer and easier to analyze, thus increasing the chances that prompt, appropriate, and useful responses will follow.

INTRODUCTION

SAS-L is a worldwide online community of SAS software users. Participants discuss various aspects of SAS and help one another solve SAS-related problems.

Participants include many of the world's most experienced and knowledgeable SAS users, some of whom are presenting highly anticipated papers at this conference. Others are newcomers to SAS, or users with experience in one area who are moving into new SAS products or usages. Customarily, "newbies" are treated well; their questions are taken seriously, and it's not unusual to see responses providing tested code and extensive explanations.

SAS-L is a LISTSERV® e-mail list. However, the subject of this paper is the content rather than the mechanism. This content can actually be accessed in a number of ways: electronic mail (via LISTSERV), Usenet (newsgroup comp.soft-sys.sas), and web interfaces for both LISTSERV and Usenet. See the Appendix and the References. Although this paper refers repeatedly to SAS-L, the points generally apply as well to users of the other channels.

I assume that you are familiar with both (1) the mechanics of your preferred access mechanism (LISTSERV, Usenet, or Web) and (2) "netiquette" and, more generally, the culture of online discussions. If you are new to SAS-L, and especially if you are new to online applied technology discussions in general, I suggest that you get a sense of the nature of SAS-L by looking at some of the postings, in either the SAS-L archives or the newsgroup archives (available through Google™ Groups). See the Appendix and the References.

This paper is intended primarily for people who wish to present SAS problems in the hope of getting practical and useful solutions to these problems. It offers a number of suggestions which can increase the likelihood of such a positive outcome.

If you have a problem to present to the the SAS-L community, here are some questions to consider:

- Have you done your homework?
- Is SAS-L the best place to post your question?
- Will your posting be noticed?
- **Can** your problem statement be understood? That is, is it reasonably complete?
- **Will** your problem statement be understood? That is, have you made it reasonably easy for people to focus on the essence of the problem?
- Have you made it easy for people to replicate your results and to experiment with your code and data?

These questions pertain to the initial posting. It's also important to monitor the response and, under some circumstances, to make follow-up postings.

DO YOUR HOMEWORK

Why is this important? Nobody is paid to monitor SAS-L and to provide answers to people's questions. It's strictly a voluntary, peer-based effort. So it's reasonable to expect people to post only those problems which they cannot solve on their own.

HOMEWORK, FIGURATIVELY …

The answer to your problem is probably in one (or more) of many widely available resources. Most of these resources are freely available via the Web. Among them are:

- SAS software documentation
- SAS-L discussion archives
- proceedings of this conference and of past NESUG and SUGI conferences and
- technical and user-support materials on the SAS Institute web site.

See the References.

Of course just because the relevant information is out there does not mean that it's easily found, or recognized, or understood. Moreover, solving a SAS problem often involves synthesis, combining two or more SAS capabilities which are separately documented. But it's always a good idea to look. If you still need help from SAS-L, mention the subjects on which you did research. That will illuminate your thought processes and help others to better understand the problem. It will also earn you credibility and sympathy. Here's an example:

> "I looked at the examples in the PROC FREQ section of the Procedures Guide, but none of them showed how to left justify the output."

Another form of "homework" is experimentation. Often the best way to find out if SAS permits some code construct, or if SAS behaves in some speculated way, is to try it (of course with suitable precautions to avoid destructive or disruptive consequences to your production environment). If such testing fails to answer your question, post the question on SAS-L and include a summary of your experiment(s).

AND HOMEWORK, LITERALLY

If you are taking a SAS class (or being taught SAS as part of some other course), don't just post your homework problems on SAS-L. You will learn more by attempting to devise your own solutions. Of course, if you are on the way to the solution of a problem and encounter some particular difficulty, and if your ground rules do not prohibit doing so, go ahead and post a help request on SAS-L. Be sure to show what you have done, mention things you have considered doing, and explain your reasoning.

IS SAS-L THE RIGHT PLACE?

Why is this important? There are many, many online discussion spaces. SAS-L is the one which takes SAS as its subject matter. If your question has no connection to SAS, you should probably look for a more appropriate list or newsgroup, where you are more likely to find relevant expertise.

WILL YOUR POSTING BE NOTICED?

Why is this important? SAS-L is a fairly busy list, with several dozen postings on the average day. Moreover, most people deal with a lot of message traffic in addition to that coming from SAS-L. As a consequence, many people decide whether or not to read a SAS-L posting on the basis of a quick inspection of the subject line alone. If your subject line is uninformative (in the worst case, blank), or hard to understand, or if it completely misses the essence of the problem, you may not even gain the attention of the people who could solve your problem.

So being noticed depends greatly on composing a good (clear and informative) subject line for your posting. That's easy to say, but it's a little difficult to give advice that's terribly helpful.

I can start by enumerating some things to avoid. Anything like

```
Subject: Please Help
```

is very uninformative, since much of the message traffic on SAS-L consists of requests for help, and the subsequent responses. Something in the vein of

```
Subject: SAS Question
```

is almost as uninformative, for the same reason.

"SAS Question" as a subject is worth discussing a bit more. Consider that a SAS-L subject line will appear in three contexts: the sender's, the recipients', and that of the archives. The sender's "Out" box almost certainly contains messages which are not questions and which have nothing to do with SAS, and the sender may only rarely ask SAS-related questions. So, in the sender's context, "SAS Question" may be useful as a subject header. It's in the other contexts that it is less than helpful. But, if you are going to the trouble of asking a question, and hoping for useful replies, shouldn't you go out of your way to be accommodating?

Now let's turn to what makes a **good** subject line. Be specific, if the specificity relates to SAS. So if you are having a problem with a particular SAS procedure or statement or function, etc., use that in the subject. Example:

```
Subject: PROC SUMMARY Options
```

Your thinking may center on a task rather than on a particular SAS tool. Suppose you work in some esoteric field, where "bluedoo" is part of your arcane vocabulary, and you have to count your bluedoos, so you come up with

```
Subject: Counting Bluedoos
```

The reference to counting is informative, but you can do even better by coupling the widely understood term "Counting" with some phrase which will convey the nature of the problem, such as

```
Subject: Conditional Counting
```

In reality, it can be hard to come up with a subject line which is both accurate and precise. Often, if you could, you would be halfway to solving the underlying problem. But it is important, and thus worth attention and effort.

IS THE PROBLEM STATEMENT COMPLETE?

Why is this important? At this point, the reader of your posting is at a crossroads. Without a good roadmap, s/he may go off in the wrong direction, or simply give up on the journey.

I'll divide the process of stating the problem into three components: defining the environment, describing the problem, and illustrating the problem.

ENVIRONMENT

SAS runs in a wide variety of environments, and that has to be considered in the problem-solving process. So it helps to include environment information in a problem statement. Pertinent information items may include

- version number of SAS software being used
- SAS products which are licensed
- SAS system options in effect

- host operating system
- hardware platform
- network configuration and
- specifics on non-SAS software products which are expected to interoperate with SAS or which are otherwise involved in the problem.

Another aspect of the environment which may be important consists of constraints, such as client specifications or regulatory restrictions or other aspects of the problem which are beyond your control.

Not all of these information items are known and relevant and important in all cases.

DESCRIPTION

Narrative explanation of the data in hand, requirements, efforts made, difficulties encountered, and so forth is essential. Numerous specific suggestions appear elsewhere in this paper. Try to be consistent with SAS documentation in your use of terminology.

ILLUSTRATION

A concrete narrative description of the problem is of course valuable, but it is rarely adequate by itself. People will be able to grasp the problem much better if shown details and evidence. Include most if not all of the following:

- input data
- SAS code
- excerpts from the SAS log (ERROR and WARNING messages, as well as unusual or mysterious NOTE lines)
- results which were generated and
- results which are needed or expected.

As much as possible, copy and paste directly from a SAS session, to avoid introducing anomalies and errors. But take the time to edit the log messages so that you don't present a "needle in a haystack". The last item (needed/expected results, to the degree they differ from those produced) would of course be mocked up rather than copied.

It is not always necessary to include all of these things. For example, if a step fails, there is no output; if the question is one of performance alone, then the output produced is as needed and expected.

If your question is a hypothetical one, you won't have a ready-made example from your work. Consider fabricating one.

Keep in mind that good illustrative examples are particularly important because of the international nature of SAS-L. Nearly all SAS-L messages are in English, but for many participants English is a second language.

IS THE PROBLEM STATEMENT FOCUSED?

Why is this important? As noted above, having an informative subject header can be essential to gaining the attention of people who might have solutions to your problem. Suppose that such a person opens your message, only to encounter a massive presentation of data and code, and perhaps a lot of unfamiliar terminology drawn not from SAS but from the subject matter of the problem. He or she may not have the time and patience to wade through all of this material.

There are a number of ways in which you can streamline your presentation and thus help people to focus on the essence of the problem.

- Reduce the scale of the dataset(s) involved.
- Cut away processes which are upstream or downstream from the problem, as well as details which are not germane.
- Eliminate terminology which is not widely understood.

Unfortunately, each of these actions can have undesirable consequences, so there are tradeoffs to evaluate and compensating measures to consider.

MINIATURIZE

Earlier, I explained the value of illustrative examples including input data, code, log excerpts, actual output, and mockup output. This can take a lot of space. Apart from bandwidth considerations, the reader would have a hard time taking it all in. Often, the solution is to construct a miniaturized version. Reduce the number of observations, the number of variables, the number of keys (BY variables) and the number of key combinations (BY groups).

There are basically two ways to do this. One is to start with your real-life data and do some subsetting. The other is to completely fabricate your miniature example. Of course if your data are private or confidential or in any way sensitive, your only choice is to fabricate.

In shrinking your example, don't lose generality. For example, if there are BY groups in the actual problem, have at least two BY groups in the example, and if a particular variable in the real data has some missing values, represent it in the miniature with a variable having some missing values.

As helpful as miniaturization is, the real-life scale of a problem can be important. This is most obviously the case if the issue is one of performance (speed). But in any case, if you miniaturize your problem, it's usually a good idea to include in your description a summary of the actual scale

(the number of observations, variables, BY groups, etc.). For example:

> "The real data set has seven categorical variables, 40 response variables, and about 2 million observations. Two of the categorical variables and most of the response variables have some missing values."

ZOOM IN

In addition to reducing the scale of the problem for presentation, it can help to simplify it by whittling away extraneous details or preliminaries. For example, if your code is along the lines of

```
data ds1;
...
proc this data=ds1 out=ds2;
...
proc that data=ds2 out=ds3;
...
proc etc  data=ds3 out=ds4;
...
data ds5;
set ds4;
...
```

and your question is about the final DATA step, don't provide test data for DS1. Instead, provide a DATA step which loads a small but representative sample version of DS4. That way attention is focused on the DATA step where the problem is. SAS-L readers would see only

```
data ds4;
...
cards;
...
...
;
data ds5;
set ds4;
...
```

Unfortunately, if you present only what you perceive to be the immediate problem, with little or no context, you may not get the best answer. If somebody can see not only where DS4 comes from, but also why DS5 is needed and what the overall purpose is, s/he may be able to suggest an altogether different (better, more efficient) approach to your overall task, one which avoids the immediate problem altogether.

So there is a real tradeoff, between focus and context. The best course depends on circumstances, but in general I recommend full presentation (with sample data, code, etc.) for the immediate obstacle, with a narrative explanation of the overall process.

For example, suppose that you are having difficulty building some dataset, because you want to combine a series of daily totals for each department in your company with year-to-date totals for each **other** department. You might present your example to show only the step which attempts to bring about the combination, but include narrative along the lines of

> "The input dataset is extracted from a remote transactions database by a scheduled job which stores a date-stamped flat file on our LAN early each morning. The summary file I'm trying to build will be used to generate a set of tables in our management monitor system. Each department likes to see its own day-to-day track but only needs current-year cumulatives for other departments."

Notice that this narrative covers not only the provenance of the data, but also the purpose (**why** your are building the dataset which is causing you difficulty).

BE GENERIC

Often a problem statement is clearer and more easily grasped if you use generic names for variables rather than names which are meaningful only to specialists in your line of business or subject matter. For example, suppose that your work entails a concept called "self-identified cohort" and that you use a variable recording this classification to group your data. So, you may have a variable named "SIC", but naming it "GroupID" when presenting your problem on SAS-L will make things easier for most people to follow.

On the other hand, the SAS-L community includes people who have experience in many fields and with many types of data. By preserving the substantive context of your problem, you may elicit some extremely valuable advice which would otherwise not be forthcoming. So this is another point where there is no simple rule. A good course might be to use generic names in the code you present on SAS-L, but indicate actual sense in your narrative. For example:

> "The real data set is drawn from a credit card transactions database. The group ID variables are based on age brackets, as self-reported on a sweepstakes entry form."

ENABLE CPR (COPY/PASTE/RUN)

When you present the input items (data and code) in an illustrative example, it's a good idea to do so in a self-contained package which readers can quickly copy, paste into a SAS session, and run to reproduce your results. Why is this important? It makes it easier for people to experiment, and perhaps discover or develop solutions to your problem. It also makes it possible to expose information which is not in your posting (for example, by running PROC CONTENTS against output datasets to reveal variable attributes).

So don't present code with blind references to pre-existing SAS datasets or external files. Introduce input data in the

form of data-loading DATA steps, rather than by pasting in PROC PRINT output.

For example, instead of presenting an input dataset as

```
Customer    Transaction
   Num         Date       Amount

   22        20030330       230
   21        20030401        88
   22        20030401      1000
   21        20030402       335
```

present it as

```
data start_with;
input CustomerNum
      TransactionDate :date9.
      Amount;
format TransactionDate yymmddn8.;
cards;
22 30MAR2003 230
21 01APR2003 88
22 01APR2003 1000
21 02APR2003 335
;
```

Consider what this accomplishes. It allows a reader to quickly set up a test bed to work on the problem at hand. In addition, it adds information. For example, look at the middle column (the date variable) in the tabular presentation. What does it represent? There are three possibilities: (1) a numeric variable following the SAS convention for representing dates (days elapsed since January 1, 1960) and displayed via the YYMMDDN8. format, (2) a numeric integer of magnitude 20 million, or (3) a character variable. In the second presentation, there is no such ambiguity; one can see that TransactionDate is a SAS date variable.

If input data are voluminous and it is not practical, or not desirable, to miniaturize, consider using one of these techniques:

- creating a data generator (typically a DATA step with an OUTPUT statement within a loop, perhaps using pseudorandom number functions to introduce variation)
- offering to e-mail the data to people directly or
- posting the data online (such as on your personal website).

Here is an example of a data generator:

```
data start_with;
format TransactionDate date9.;
do TransactionDate =
 '01JAN2002'd to '31DEC2004'd;
   if ranuni(1)<0.98 then
    do CustomerNum = 300 to 499;
      if ranuni(2)<0.05 then do;
         Amount =
          round(1+999*ranuni(3),
                0.01);
         output;
         end;
      end;
   end;
run;
```

If you e-mail data or post it on a web site, don't use SAS native datasets. There are too many version and platform compatibility obstacles.

FOLLOW-UP POSTINGS

There are several circumstances under which it is appropriate to post follow-up messages.

- If you have seen no responses, the original problem statement was probably not as clear as it needed to be, so go to the trouble of restating it and elaborating.
- If the responses you've seen are going in the wrong direction, offer clarification.
- If there have been responses asking questions or requesting details, address these.
- If you have made use of suggestions offered in response to the original posting, but still find yourself short of a full solution to the problem, post a mid-course progress report.
- If you have received useful suggestions via private e-mail, share these with the group.

One final point about follow-ups: don't change the subject line unless there is a really compelling reason to do so. When a discussion continues under a different subject, the "threading" is disrupted and people may have trouble keeping track. This is true even with the tiniest changes (such as removing an extraneous space in the middle of a word), so either use "reply" functionality or copy and paste subject lines to preserve them verbatim.

APPENDIX: MECHANICS

There are several ways to participate in the online SAS discussion. Subscribe to SAS-L by sending a message like

```
subscribe sas-L Yourfirstname Your-
lastname
```

to

```
listserv@listserv.uga.edu
```

or use the web subscription form at
http://listserv.uga.edu/cgi-bin/wa?SUBED1=sas-l

It's also possible to use the archive site by itself as a mechanism for full participation. That's because the archives are maintained almost continuously. Usually, postings are available there within moments. The archive web site also includes a form for posting messages. See the References.

If you have a Usenet feed, you can participate by subscribing to the newsgroup

```
comp.soft-sys.sas
```

Google Groups provides a full-service web alternative to the use of a newsreader to access the newsgroup. See the References.

CONCLUSION

There are things you can do to greatly increase the likelihood that the SAS-L community can and will help you solve your problem.

If you attempt to follow all of the guidelines and suggestions in this paper, the asking of a simple question may become an overly elaborate exercise. That's because all of the points are not applicable or important in every situation. So use your judgment.

REFERENCES

ARCHIVES OF ONLINE SAS DISCUSSIONS

Google Groups, comp.soft-sys.sas,
http://groups.google.com/groups?group=comp.soft-sys.sas

University of Georgia, Archives of SAS-L,
http://listserv.uga.edu/archives/sas-l.html

OTHER ONLINE SAS RESOURCES

SAS Institute Inc., "Available Documentation" (includes product manuals and technical support materials),
http://support.sas.com/documentation/onlinedoc/

NorthEast SAS Users Group (NESUG), *Proceedings*,
http://www.nesug.org

SAS Users Group International (SUGI), *Proceedings*,
http://support.sas.com/usergroups/sugi/proceedings/

INFORMATION ABOUT PARTICIPATION MECHANISMS

Internet FAQ Consortium, *Usenet References*,
http://www.faqs.org/usenet/

L-Soft international, Inc., *General User's Guide to LISTSERV*,
http://www.lsoft.com/manuals/1.8d/user/user.html

ACKNOWLEDGMENTS

The development of this paper benefited greatly from the ideas and examples contributed by countless SAS-L contributors over my 14 years of participation in this community. That is what SAS-L is about.

REVISION HISTORY

Revised for NESUG 16 (Sept. 2003). Explicated distinction between "can" and "will" the problem statement be understood. Changed seed arguments passed to RANUNI function. Inserted additional examples. Other, minor, editorial changes.

First presented at SUGI 28 (March 2003).

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Howard Schreier
U.S. Department of Commerce
Stop H-2815
Washington DC 20230

(202) 482-4180

Howard.Schreier@mail.doc.gov