Teaching the SAS™ Programming Language to Programmers and Non-programmers

Wendy London
Pharmaceutical Research Associates, Inc.

INTRODUCTION: This paper reviews our efforts at Pharmaceutical Research Associates, Inc. (PRA) in the designing and teaching of a course in SAS Programming. There was (and still is) a high demand for timely instruction in SAS from a broad audience at PRA. The programming experience of the employees who wanted to take the course ranged from none in any language to good SAS proficiency. We attempted to address the needs of as many people as we could, as soon as we could, and to cover basic through advanced topics. The instructors of the course were SAS programmers and statisticians, most with no formal training in the field of education, and few with teaching experience. What a challenge! The whys, hows, and whats of our approach to this challenging task are presented here. The input for the discussions of improvements to the course was obtained from the students' evaluations of the course and the opinions of the instructors. Read on to discover why, with hindsight, we will probably divide the course into two levels in the future: a basic series for non-programmers and an advanced series for programmers.

Why do many PRA employees and the PRA management believe that a course in SAS programming needs to be offered? We have a growing volume of work that needs to be performed by using SAS. Currently, PRA applications for SAS include:

- clinical data analysis;
- FDA ready report generation;
- data tracking;
- project time keeping;
- data quality assurance review;
- data consistency checking; and,
- VMS accounting analysis.

As our usage of SAS for applications increases, more of our users see new ways that SAS may be applied to their problems. Personnel have been hired with varying degrees of proficiency in SAS programming. Some programmers need to extend or refine their SAS programming abilities. Statisticians who have advanced skills in the use of the statistical procedures would like to gain a greater understanding of programming in the DATA step. Other employees, programmers as well and non-programmers, who do not know SAS would like to learn it because they believe SAS would assist them in their work. Non-programmers would like to be more self-sufficient in their investigations of the consistency of the database and display of the data.

There clearly exists a need at PRA for instruction in the SAS programming language. We decided to design our own series of classes for several reasons:

1) We wanted to tailor our instruction so that students would learn to program according to the guidelines of PRA's SAS Coding Standard Operating Procedure (SOP).
2) We wished to emphasize the procedures and techniques most often needed in the work that PRA does.
3) We wanted to have the flexibility to slow down or speed up the classes depending on the ability of the students to learn the material.
4) We thought we could get more for our time and money than could be provided by a training package from SAS Institute.
5) We wanted to meet the challenge of addressing the needs of non-programmers and programmers in the same course.

When the instructors started designing the course, these reasons became identified as objectives of the course. In this first attempt at teaching the course, we addressed two of these objectives better than the others. Reasons 1) and 2) were fairly well addressed by the course material, but I do not think that we varied the pace of the course as much as we had intended as stated in 3). We are still assessing, as stated in 4), whether the time and efforts spent on course development were less expensive than using a course provided by SAS Institute would have been. Regardless of cost, I would still favor in-house course development because of the benefits of tailoring the instruction to the specific requirements of PRA's work. Plus, the more we teach the course, the more cost-effective it will become. Our approach to addressing objective 5) will be discussed later in the paper.

When students were surveyed on their course objectives, some of the non-programmers said that they wanted "to gain exposure to SAS", and others wanted "to obtain a working knowledge of SAS". The non-programmers also wanted to be able to communicate better with the programmers and to know where to look on the VAX for SAS programs and output. The objectives of the programmers ranged from obtaining "a working knowledge of SAS" to becoming "proficient in SAS". Some programmers wanted to gain a better understanding of the output from statistical procedures, and to learn "extra tidbits". In order to meet the objectives of the students, an additional objective of the instructors, probably their most
important objective, was to teach the students techniques that they wanted and needed to learn. The course objective from the perspective of PRA's management was for the students to benefit from the PRA sponsored class time so that the students would be more productive and more skilled employees.

At the end of the 12-week course, students were asked to complete course evaluations. The course evaluation form was completed and returned by 19 of the 20 students who participated in any aspect of the class. Students were asked, "How well was your objective (in taking the course) met?". Three students (2 Programmers (P) and 1 Non-Programmer (NP)) indicated that his/her objective was "poorly" met, one student (P) said "fairly", and four students (3 P, 1 NP) indicated that his/her objectives were "adequately" met. One of these latter programmers stated that they had done very little of the homework. Seven students (5 P, 2 NP) indicated that his/her objective had been "well" met, and four students (3 P, 1 NP) indicated that his/her objective had been "very well" met. Thus, 15 of the 19 students believed that their objectives were at least "adequately" met.

At the time the course was announced, there were 57 PRA employees; 26 of them expressed an interest in taking the course. We determined that we would be able to teach a class only half this size for several reasons: 1) inability to address the needs of the individual if the class was too big; 2) take too much of the instructor's time to check 26 homework programs each week; and, 3) classroom size limited the number of students who could comfortably be seated at one time. Since the majority of the students who wanted to take the course were employees in the Information Services Department, the Manager of Information Services selected 12 lucky students. Two additional students subsequently begged, bribed, or pulled rank to get into the course. Of these 14 starting students, seven were currently performing programming at PRA and seven were non-programmers. Six more programmers, fairly proficient in SAS, joined the course for the more advanced topics (starting at Class 9), and by then six students (two programmers and four non-programmers) had dropped out. There will be more discussion of the dropouts later in this paper.

There were 12 classes in the course, each one hour in length, and were scheduled each Thursday at 9:00 am. The topics covered, by class, were:

1. Introductory class for non-programmers - background on the "SAS System", computer and programming terminology, VAX directory structure.

2. "The Semicolon", DATA step vs. PROC step, INFILE, INPUT, permanent vs. work data sets, PRA utility software that creates permanent data sets from flat files, variable and data set naming, SET, OPTIONS, LIBNAME, comments, basic PROC PRINT.

3. Documentation in the header of a program, operators, IF THEN ELSE, ENDSAS, OUTPUT, KEEP and DROP.

4. RETAIN, PROC SORT, PROC CONTENTS, BY, FIRST.* and LAST.* variable groupings, MERGE, IN= variable, output to multiple data sets.

5. Missing values, ATTRIB, LENGTH, PUT and INPUT functions, SAS dates, PROC FORMAT, VALUE, PICTURE.

6. Functions: SUM, CEIL, FLOOR, SUBSTR, TRIM, SCAN. Presentation of PRA's SAS Coding SOP (good programming practices, formatting standards).

7. Debugging, subsetting, error prevention, testing, PQA (Program Quality Assurance - checking your code and that of other programmers), interpretation of log file, PROC DATASETS.

8. More on PROC PRINT, FILENAME, DATA _NULL_ (PUT, FILE), some standard PRA tables.

9. PROC FREQ, PROC MEANS, PROC UNIVARIATE, PROC TABULATE, PROC SUMMARY.

10. %INCLUDE, macro variables, %LET, SYMPUT, SYMGET.

11. PROC TTEST, PROC GLM, PROC LIFETEST, PROC ANOVA.

12. Graphics: PROC PLOT, PROC GPLOT, PROC ANNOTATE, PROC CHART, PROC GREPLAY.

This outline was revised as the course progressed. Instructors discovered that some topics had been omitted from the original outline, and these topics were added. Some material was shuffled from class to class as we determined how long it took to cover a particular topic in classroom discussion.

Regarding our original objective 5) above (simultaneously addressing the needs of programmers and non-programmers), we developed an approach to meet this challenge. We provided the students with the outline of the topics to be covered in the course so that they could select appropriate classes to attend. We informed the students that the degree of difficulty of the material increased as the classes progressed. We hoped that, by having the students select the classes based on their need to know, we would prevent the programmers from being bored with the basic material, and keep the non-programmers from being unnecessarily overwhelmed with difficult material. Students needed the agreement of their supervisor and the course coordinator before skipping any classes. The programmers with some previous SAS abilities skipped the introductory class and started at Class 2. In the opinion of
The classes were taught by five SAS programmers and two statisticians, employees at PRA. One of the programmers (myself) acted as the coordinator for the course, as well as teaching some classes. The decision to have several instructors was made because we wanted to distribute the workload of designing and teaching the classes (I sure didn't want to do it ALL myself!), and so that we could draw on many areas of expertise and experience that our instructor attempted to select the area or areas to teach in. Instructors was made because we wanted to distribute the workload of designing and teaching the classes (I sure didn't want to do it ALL myself!), and so that we could draw on many areas of expertise and experience that our programmers and statisticians could contribute. Each instructor attempted to select the area or areas to teach in which he or she was most proficient or knowledgeable. Instructors were limited in the amount of time they had available to prepare the handouts and lecture for the course. The instructors were expected to keep up with their normal workload, to continue to perform their primary job functions of programming and statistics in addition to doing course preparation work. Consequently, much of the preparation for the course was done on the instructors' own time. In cases where time was available, the instructor rehearsed the lecture before presenting the material to the class. A template for the class structure was provided to each instructor in an attempt to ensure consistency across the classes. The template looked like this:

- Presentation of the solution to the previous week’s homework (5 minutes)
- Questions and answers on previous homework (5 minutes)
- Introduction/overview of material, discussion of that week’s reading assignment (5 minutes)
- Explanation of concept (15 minutes)
- Application of concept, with examples (15 minutes)
- How this code fits into the guidelines of the SOP (5 minutes)
- Questions and Answers (10 minutes, anytime throughout the class)
- Summary - quick restatement of material presented (5 minutes)

At any time during the classes, we welcomed questions from the students.

Each class had a homework assignment associated with it. The homework included a reading assignment and a program or two to be written, debugged, and successfully executed. The homework was due the next class. We believe that the majority of the learning of the SAS programming language results from actually using the language. It is essential for students to apply what they learned in class by writing programs. For employees to remain in the class and to continue to benefit from instruction during company time, they had to complete the homework assignments on schedule. Homework was to be completed on the employee’s own time. For this reason, employee participation in the course had to be voluntary, though some supervisors strongly encouraged their employees to take the course. Some students were not able to complete their homework on time and a grace period was reluctantly given to these students. When some of them still failed to complete the homework, they voluntarily dropped out.

According to the students' evaluations of the homework assignments, most of the non-programmers thought that doing the homework contributed more to their learning of SAS than did the class lecture, the reading, or interaction with instructors and classmates. Most of the non-programmers found the assignments to be "very challenging", while most of the programmers thought the difficulty level was "just right". Most non-programmers thought that the homework "took a long time to do", while most programmers thought that the quantity was "just right". Most programmers found that the most valuable method for getting feedback on their homework was the classroom discussion of the solution code, while non-programmers preferred studying the solution code and comparing it to their programs on their own.

There were six dropouts from the class; four non-programmers and two programmers. In each case, the reasons that these students withdrew from the course related to their inability to complete the homework. Four out of the six students who dropped said that they did not have enough time to do the homework because they had so much PRA work to complete. These four same plus one student also said that they did not have enough time because of personal situations. Only one of the dropouts stated that the quantity of homework was too much for them to have been expected to complete. None of the dropouts stated that they had dropped because the class was not beneficial or that the material was too difficult.

We did not have a textbook for the course, though each student had a copy of SAS User’s Guide: Basics and access to SAS User’s Guide: Statistics. Because several of the instructors had had to learn SAS with only the SAS User’s Guide: Basics for reference and found it lacking as an instructional tool, we realized that we needed to supplement the User’s Guides with handouts. For each class, we created a handout that covered the material discussed in the lecture for that class. Examples of SAS code used in the handouts were often real applications of code in use at PRA. The handout also presented the SOP guidelines
applicable to the topic. Instructors spent more time developing the handouts than on any other single task in development of the course. I believe that this time was well spent because it forced each instructor to be sure that he/she had a thorough understanding of his/her topics.

As stated earlier, nineteen of the 20 students completed the course evaluations. One anonymous evaluation form returned had been completed using letters which had been cut from a magazine. The instructors refused input from terrorist students and so we disregarded that form; he/she didn't have anything good to say anyway. Actually, we took all students' comments very seriously. Here are some of the suggestions for improvement/criticisms we received about the course, as well as how we hope to implement/resolve them the next time around:

Comment:
Class went too fast for non-programmers. Grasp of basic programming concepts was assumed.
Resolution: Offer a separate, more basic class for non-programmers.

Comment:
Some topics were covered too quickly (arrays, merging, macros) or not at all (efficiency techniques, how SAS Supervisor works, user input from the terminal!).
Resolution: Slow down on certain topics, add others. Allow more time for difficult topics.

Comment:
Could a separate course be taught for each PRA working group? (make it even more PRA specific by group)
Resolution: Let's try having just two courses for next time, a basic and an advanced. If this still isn't sufficient, then we could consider it.

Comment:
Too much time was spent on the SOP and interpretation of the output of statistical procedures.
Resolution: Explain to the student why instructors felt this much time was necessary.

Comment:
Need more time spent in the classroom for discussion of homework solutions and more time for difficult topics.
Resolution: Get management to give the OK for employees to spend more time (extra half an hour per class or optional one hour tutorial per week) in class.

Comment:
Relationship between material taught that week and the homework assignment was not always clear.
Resolution: Improve homework assignments, discussion of assignment in class at the time assignment is made, make it clear to students that they should ask questions if instructions aren’t clear (just like the real world).

Comment:
Could handouts be provided at least one day before the day of class?
Resolution: Yes, we sure can try...especially now that they're all prepared.

Comment:
Give quizzes to assess the how much students have learned so that difficult material can be discussed further.
Resolution: Under consideration.

Comment:
Some instructors were poor communicators.
Resolution: Not sure this can be completely resolved - we’re programmers, not teachers. Instructors can be coached on teaching techniques.

Comment:
Not always prepared to cover homework solution at class time.
Resolution: Won't make this mistake again. (These students are HEARTLESS!)

Comment:
Not fair to modify the homework assignment in the middle of the week.
Resolution: Won't do this again unless we do it once intentionally for education on "when the user changes the specifications".

Comment:
Start using the program checking method of swapping with another programmer sooner or more frequently in the course.
Resolution: Can make this an option for whoever wants to find a checking buddy for any of the homework programs.

Comment:
Let new SAS users teach a couple of the more basic classes - learn a lot by teaching.
Resolution: We could try this if we get any volunteers.
Comment:
Be more specific when determining what the students' learning objectives are - provide them with a checklist of topics and let them choose what they want to learn about.
Resolution: Under consideration.

Comment:
Need better explanation of why and when to use certain PROCs.
Resolution: Can be better covered in more advanced classes. Consult a friendly statistician in the meantime.

In reviewing the praises of the course from the evaluations, we discovered that some of the positive feedback contradicted some of the views of students with negative feedback. Some students even contradicted themselves! Most of the students responded to most of the questions in a positive manner. In summary of those responses:

* The instructors presented the material "adequately" or "well".

* Valuable things learned in the course included: what a data set is, terminology, how a computer language actually works, SAS code structure, functions, macros, SYMPUT, SYMGET, DATA _NULL_, how to run a SAS program, stat PROCs, interpretation of log file, and "overall knowledge".

* The most enjoyable things learned were: DATA _NULL_, stat PROCs, tables and graphs, that SAS was now a usable tool, and the doughnuts we had with each class.

* Programmers and non-programmers felt that the topics which they wanted to learn about were "well" covered.

* Most of the students indicated that the pace at which the material was covered was "just right", though three programmers and three non-programmers thought that the material was covered "a little too fast" or "too fast".

* Programmers thought that the SAS User's Guide: Basics was "some help" or "a lot of help" in explaining SAS, while non-programmers thought it was "some help" or "no help at all". One programmer stated that "you really had to know where to look" in the guide for it to be helpful.

* Responses from programmers on how well the handouts covered the material ranged from "adequately" to "very well". Non-programmers thought the handouts covered the material "fairly" or "adequately".

In answer to the question, "Did you learn enough to have made it worthwhile to have taken the course?", 16 of the 19 students who completed the evaluation said, "Yes". The same three students who responded with an answer of "poorly" to the question about how well his/her objective had been met also said "No" to this question. One non-programmer said "No" because he/she didn't have enough programming background before starting the course. One of the programmers said "No" because advanced topics were not covered. The other programmer who said "No" did not provide any reasons why the course was not worthwhile to him/her.

The 16 students who did find the course worthwhile gave many reasons why:

* now have a more in-depth understanding of analysis of the clinical data as it relates to PRA's database.

* will be able to use SAS to supplement DATATRIEVE and determine when a SAS program would work more effectively instead.

* able to apply what was learned to increase efficiency on the job.

* learned shortcuts in SAS which will make future programs easier to write.

* learned a lot of new details or obtained an understanding of details in SAS which he/she had been using without knowing why or how.

* more SAS programmers now to discuss SAS issues with, share information with, and learn new tidbits from.

* provided a resource for answering SAS questions.

* clarified a lot of SAS concepts.

* covered most functions and helped to solidify knowledge.

* tailored to PRA specific standards.

* gained insight into why SAS does what it does.

* can now write useful SAS programs.

* gained confidence and ability by doing the homework (could have learned more by spending more time on homework).

* only went to three classes, but what was learned was valuable and a good start.

* anything learned is an improvement over nothing.

* now more conversant with SAS capabilities and terminology.
easier to communicate with programmers.
* better understanding of appropriate applications for SAS.
* the exposure alone was definitely worthwhile.

Unsolicited praises included:

* with a little effort, this course could be converted into a nice independent study course.
* much more impressed with this class as compared to other classes taught at PRA.
* felt very fortunate to have been a part of the class and can really appreciate the work that programmers have to do. Thank you!!
* preparing for instruction of this class could be a full time job. Given the amount of time the instructors had available, they did an excellent job.
* helped to have a variety of viewpoints and techniques to teach the class. Constructive criticism from other instructors during class was good.
* Good job!

In summary, PRA was faced with an immediate need for teaching the SAS programming language. We were challenged by the fact that those who wanted instruction were a mix of programmers and non-programmers. We designed our own course, aiming to teach SAS techniques tailored to PRA needs and standards. The material covered in the first couple of classes was basic information supplied mainly to meet the needs of the non-programmers, while the latter classes contained more advanced techniques for the programmers. Most of the students felt that their objectives in taking the course had been met. Sixteen of the 19 students who completed course evaluations indicated that he/she had learned enough to have made it worthwhile to have taken the course. Many reasons why the course was worthwhile were provided by the students. We also received many good suggestions for improvements to the course, and we plan to implement as many of these suggestions as possible for the next time the course is offered.

The criticisms most frequently voiced by the students all related to the disparity of the initial knowledge and capabilities of the programmers versus that of the non-programmers. Though we tried very hard to address that difference, the course still contained topics which were not covered at all for programmers. The homework proved to be rather difficult and time consuming for the non-programmers. Despite the fact that the course was worthwhile for most students, we have determined from this experience that the needs of the students would be even better addressed if the course were divided into a series of classes for programmers, and a separate series for non-programmers. This division will not entail a rewrite of the course material. Much of the same material could be presented in both class series, but the pace would be slower in the non-programmers' series and some material might not be covered at all in one series or the other. A more homogeneous audience would be addressed within each series. For the classes for non-programmers, each class as it currently exists could be split into two classes so that 1) time would be spent on programming concepts used in writing SAS code; 2) more time would be spent answering questions in class; and 3) less material would be covered by the homework assignment. For the series for programmers, more classes could be added at the end covering more advanced topics in detail.

References


Acknowledgements

I would like to thank the instructors for their hard work designing and teaching this course and the students for their input.

The Author

1290 Seminole Trail
Charlottesville, VA 22901
(804) 978-7730
fax (804) 978-7084

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.