SAS® GLOBAL FORUM 2020 #SASGF

Paper 4652-2020

The Past, Present, and Future of Training SAS® Professionals in a University Program

Jonathan W. Duggins, NC State University;
Jim Blum, UNC Wilmington

ABSTRACT

University students who are fortunate enough to encounter SAS® typically do so in a variety of courses and contexts. A few programs offer courses designed specifically to build SAS programming fundamentals before moving on to courses with specific applications. However, many curricula merely include offerings that target the applications of immediate interest without building a broad skill base to prepare students for careers that rely heavily on programming. This session discusses past and current practices at several institutions, the philosophies behind them, and where they succeed and fail. As SAS and its role in the marketplace evolve, a university curriculum must evolve with it. Therefore, this session closes with a discussion of how SAS and universities can work together to provide the training necessary to continue preparing high-quality programmers.

INTRODUCTION

As the importance of statistical programming continues to grow, there is an expected effect on the curricula in statistics and data science programs. However, a wide range of other disciplines from business, engineering, and various natural and social sciences such as sociology, psychology, epidemiology, and ecology also include such topics may also feel these effects. Thus, the inclusion of SAS in university courses and curricula manifests itself in a wide variety of forms. Determining how best to provide students with a strong programming foundation that develops good programming practices requires university programs to make several challenging decisions on how they can best include SAS in their curricula. This paper looks at the two prevalent philosophies - standalone instruction, where courses cover SAS exclusively, and integrated instruction, where SAS instruction is in concert with other concepts. Courses that give exclusive coverage to SAS may lean heavily toward using procedures or may lean more toward data step programming (and other programming-oriented) concepts, but may also be some combination of the two. A curricular program may then contain any number of these courses of any type and, when more than one is present in a curriculum, some prerequisite structure may be placed on students' path through them.

In the face of all of this, what follows is an effort to describe the various types of courses that use SAS, and their pros and cons, along with how those tend to be integrated into various programs (also noting pros and cons), with some attention paid to the history of such instruction and what it may look like in the future. Of course, much of how teaching SAS has developed and will develop is influenced by the relationships between the SAS Institute and various academic institutions, so considerable attention to the development of that relationship is also considered.

SAS IN INDIVIDUAL COURSES--TWO PHILOSOPHIES

As noted previously, there are two fundamental philosophies for including SAS in university courses: standalone vs integrated instruction. Standalone instruction involves courses that exclusively cover SAS concepts and other general programming concepts that are also required to effectively use SAS (e.g. path or directory structure, file types and attributes, display resolution). This is different than integrated courses which typically focus only on the portions of SAS that are directly applicable to the discipline-specific course content. For example, a general statistics course may include procedures like MEANS, UNIVARIATE, and FREQ, while a regression and/or ANOVA course is likely to include some review of the REG, GLM, and MIXED procedures. Both of the previous examples would likely include some instruction on data visualization, such as PROC SGPLOT, and managing results, such as with ODS statements. Both the standalone and integrated approaches have benefits and pitfalls, some of which are pedagogical while others are institutional, and this section focuses on those.

INTEGRATED INSTRUCTION

Integrated courses have the major advantage of directly introducing SAS tools as they are needed in the course curriculum. This just-in-time approach is a well-respected pedagogical technique; however, the major disadvantage of integrated courses is that they typically do not (and often cannot) include broader programming concepts necessary for setting up data for input, or refining results for output and reporting. The effects of this disadvantage are actually increasing as university curricula evolve. Instructors and programs are moving toward integrating realistic data, requiring more data setup skills, and requiring communication of results consistent with practices of professionals in the field which requires advanced reporting and graphing concepts.

Another substantial drawback of the integrated approach is that by distributing the instruction across multiple courses, the responsibility for the quality of the instruction is also distributed. This requires the discipline-specific courses that are part of the integrated approach to have instructors that are comfortable teaching programming, debugging programs, and assessing students on their programming skills. This also requires the instructors have time to instruct students in both the discipline-specific content and the programming content. In most cases, these drawbacks will prevent the realization of the full set of benefits unless there is careful consideration when designing the degree program's curriculum. The negative aspects can be minimized, and positive aspects maximized, with buy-in from department heads and the instructors, though this does not appear to be common. As part of that curriculum planning, it is important to note that the positive aspects of this approach are enhanced when it is preceded by a standalone course that focuses on the whole SAS language, as described below.

STANDALONE INSTRUCTION

Standalone programming courses typically are structured in one of two ways, with the pros and cons of each being somewhat different. One common implementation uses an approach that focuses more on what code is necessary to carry out analyses relevant to the other courses in the curriculum. In a SAS-focused course, a strict focus on analyses results in a heavily PROC-based course. The other common implementation follows a model for teaching SAS like a language-specific computer science course (like C, Python, or Java), broadly covering syntax, the compilation and execution processes, and good programming practices.

In a PROC-based course, most learning objectives are related to selecting the correct PROC for an analysis, developing knowledge of the syntax for the PROC, and understanding output the PROC generates. Data handling is less likely to be a focus of this type of course, and thus the DATA step and its associated utilities for reading and manipulating data, are generally left out. The direct result is one of the more prevalent issues with this approach - students finish the course without much, if any, ability to clean and/or transform data prior to subjecting it to the appropriate analysis. Also, the focus on achieving the analysis result with a PROC often leaves out utilities (such as ODS) for producing quality reporting of the results. Thus, students are often left treating the software as a black box for calculations, failing to understand the utility of SAS as a whole, and are deprived of the opportunity to develop the critically important skill of writing a SAS program from end-to-end.

Additionally, the standalone, PROC-based course suffers from a logistical issue regarding where this course would appear in a curriculum. If the PROC-based course covers procedures like REG, GLM, LOGISTIC, etc., and is offered before students take courses in regression, ANOVA and others, the coverage has to be very thin, as the techniques learned in the class are new and potentially confusing. However, if the curriculum is structured so students take this course after they have taken their content-specific courses, they would take it only after they had completed all of those content courses, and therefore would not be using SAS to apply the techniques in the content coursework. Overall, there is not an optimal time to place a standalone course focused on analysis techniques that does not suffer from this timing issue. In general, as discussed later, instruction on using SAS for analyses of any significant complexity is best done as an integrated component of the content courses.

From an institutional standpoint, the standalone, PROC-based course also requires departmental coordination in setting the course content. Specifically, that course needs to teach the fundamental analysis techniques covered in other content courses under the expectation that instruction in the content courses uses the same SAS methods to carry out analyses. For example, if the standalone course teaches PROC GLM for certain analyses, but a content course uses PROC ANOVA and PROC TTEST for the same analysis, then the benefit of the standalone course is muted by the substitution of new and different syntax in the content course.

For a standalone course that teaches SAS from a whole-language perspective, learning objectives may still include some focus on analytical skills, but the primary learning objectives are designed to give students a solid foundation in the details. These would include: syntax, compilation and execution, good programming practices, and a holistic view of the utility of the language. In a course such as this, there is limited time to introduce anything beyond the most basic of analysis procedures, which is both a potential strength and a potential weakness. It is a strength because it does not suffer from the logistical drawback of when to offer it in the curriculum; it can be offered early because students need little prior content-specific knowledge to complete this version of the standalone course. Therefore, a wide variety of students can take advantage of this type of course to learn the foundations of effective programming and good programming practices.

The major drawback to this version of the standalone course is that its ultimate utility is dependent on how the remainder of the curriculum builds upon it. In such a course, students are not learning, at least not directly, the analytical tools needed for their advanced content courses. As a result, if subsequent content courses do not build upon the concepts from the foundational course, students will not connect the skills gained in this course to the tasks they are asked to complete in other courses. Therefore, the opportunity to reinforce the view of SAS as an end-to-end tool for data analysis, irrespective of the analysis type, is lost. Thus, even with a standalone course on the SAS language early in the curriculum, a full understanding of the role of SAS as an analysis tool is dependent on its use in subsequent courses in the curriculum.

Thus, whether using the PROC-based or general programming flavor of the standalone course, the common drawback is the potential of a silo effect when students are exposed to these skills only in the standalone course, without the same skills being reinforced in subsequent courses. This is entirely the consequence of institutional barriers, and can only be avoided by getting multiple instructors to agree to, enforce, and assess a set of programming competencies that build upon each other throughout the curriculum.

OPTIMIZING THE CURRICULUM

Both the integrated and standalone models of SAS instruction are common in various undergraduate and graduate programs, but often as part of curricula where programming is an afterthought rather than as a skill in its own right. In many degree programs, as computing became more important, programming skills were added at some points in the curricula by integrating them into the existing coursework. In some curricula, the extent of SAS instruction has never evolved past this structure. In some other cases, a basic, standalone course was added to the beginning of the curriculum to attempt to provide some foundational content for the other courses. These patchwork additions of programming to the curriculum naturally lead to sub-optimal results and, unfortunately, are often a persistent structure in present-day curricula.

In order to maximize the benefits of SAS instruction, we advocate a curricular structure built upon a foundational course that is designed with specific intent. Defining the end-to-end programming process in three major components:

1. Data cleaning and preparation
2. Data summary, analysis, and modeling
3. Reporting and discussion of results

the ideal curriculum design includes a standalone SAS programming course early in the plan of study that is heavily weighted toward components 1 and 3. Therefore, this course spends considerable time on cleaning and managing data with the DATA step and its associated functions and utilities, along with tools like ODS that allow for building custom reports. Basic analysis methods are included to provide students the opportunity to write end-to-end programs. In this course it is expected that ample time is devoted to understanding the syntax and operational details of the SAS language, along with development of good programming practices.

In the applied courses, as students are developing analytical skills, they are introduced to the SAS methods needed to implement those analyses. This allows instructors to use the just-in-time approach inherent in the integrated philosophy for building on component 2 in the end-to-end process, while relying on the foundational course to ensure skills for components 1 and 3 are in place. However, it is critical that examples in these courses do, in fact, use non-trivial skills for components 1 and 3 in order to complete the assigned tasks. Further, it is important that assessment of these tasks include direct assessment of all programming tasks in the end-to-end process. Therefore, while the integrated courses are typically seen as the opportunity to build up skills for the analysis and modeling component in the end-to-end process, it is essential that they simultaneously reinforce the skills needed for the first and last components as well.

Other key components need to be in place to ensure the success of this approach. First, buy-in from faculty within the program is crucial since most, if not all, faculty must be sufficiently versed in SAS to support the development of these SAS skills throughout the curriculum. In addition, one or more faculty must be comfortable teaching SAS at a detailed level for the foundational course(s) to be successful. Next, access to training materials is highly beneficial to the development of a faculty that can implement such a program, and partnerships between SAS and the academic institutions can provide such resources. It is also vital for students and faculty to have access to SAS software that meets not only their educational needs, but also parallels what their future employers use. Finally, access to software and training materials requires support; thus, a strong, close liaison between SAS and the institutions is of significant benefit to both, as is discussed in the next section.

ACADEMIA-SAS INSTITUTE PARTNERSHIP

To aid in describing the value of a strong relationship between SAS and a particular institution, we start with a bit of personal history. Many years ago, Jim received an email from a former student thanking him for teaching her SAS programming as it had helped her to get a job. The course in question was a regression course following a fairly weak form of the integrated model, including several software packages in addition to SAS. The student's email immediately raised a question: How is it possible that such a rudimentary treatment was influential in the student landing a job? Fortunately, this question was quickly supplanted by another: If that training was of value to an employer, what better training could be offered and how much better could the program do?

It was well-known that the partnership between SAS and the University of North Carolina system included full access to SAS software, but less well-known is that (among other things) it also included the opportunity to attend training courses at SAS campus free of charge. Taking advantage of this, Jim attended several courses and used that training to revamp an introductory course to the standalone model discussed previously. Later, after attending more training courses, a second standalone course was introduced covering advanced topics such as macro language and SQL. Over the years, these courses have formed the basis for training dozens of SAS programmers employed in the Wilmington, North Carolina region and beyond--and none of it would have been possible without the software and training support provided by SAS to the university.

Over the years, students from Jonathan's classes have provided similar feedback stating that they continue to use their lecture notes from class as a resource in their job or that the skills learned in their SAS courses were integral to landing a job. Similarly, students who have continued on into graduate programs have indicated that they were much better prepared for the programming aspects of their Master's or doctoral program than their classmates who did not have such courses in undergraduate. These anecdotes have provided valuable insight into the utility of the material as well as ideas on how to improve the course. Students in our undergraduate program are exposed to R, Python, and SAS but it is common for current students to report that due to their comfort level with SAS, they go on to choose SAS when given an option of software in future coursework in our program. While this is not a formal, qualitative analysis - student feedback for this course is overwhelmingly positive despite the course consistently ranking as one of the most challenging for students. In fact, the success of the course lead to student requests for a second SAS programming course - a course that is now offered and covers advanced certification exam topics such as macros and SQL as well as additional skills necessary for statisticians such as IML, customized templates, quantile regression, and bootstrapping. This second course has gotten similar feedback and reviews, confirming that the standalone programming courses play an important role in the modern training of statisticians and data scientists.

Relationships like this are win-win prospects for both SAS and academia. The ability of instructors to offer relevant training to their students is greatly enhanced, and those real-world skills are powerful recruiting tools for any program. For SAS, a potentially greater number of SAS-trained people will subsequently join the workforce, both with employers who desire SAS skills or with those who have not yet learned the value SAS can provide to their organization. Also, since both SAS and the academy are interested in offering training, a synergy between the two can speed the evolution of training materials and methods. But there is a fundamental obstacle to providing such relationships: the ability to scale them across several departments and universities.

The number of universities, and the number of departments within each, that are potential users of SAS is quite large. Thus, providing services of significant scale to all of these partners is a daunting task, but a necessary one. If SAS cannot be properly deployed at the partner universities, or if access to training is difficult or unavailable, the curriculum cannot be expected to include SAS effectively. Therefore, continuing development on how SAS and universities work together is paramount.

Campuses that use SAS typically have one or more liaisons assigned to the task of working with SAS to disseminate the services SAS provides to their partner universities. Unfortunately, the quality of these liaison services varies widely, and is often a function of the level of resourcing and commitment provided for the university liaison. One way to mitigate this problem would be for SAS to publish and promote clear guidelines for what the relationship entails. Further, it would be of great benefit to have training and documentation available to those university employees who wish to participate as liaisons, no matter what role they play at the university – from IT staff to faculty. This would allow SAS to be connected to all interested academic partners with a minimal resource outlay, while giving the partner universities a means to effectively scale their commitment to suit their individual needs.

As industry needs evolve and SAS evolves to match those needs, it is vital that academia is kept up to speed on this progress. For example, as the Viya platform becomes more prominent, training in university programs will have to match this migration. Additional training will not only be required for faculty, but also for support personnel who manage and maintain the software deployments. The migration to new platforms, along with the introduction of new features and components into existing platforms, needs to occur seamlessly lest there be a loss of interest in continuing to use SAS software. Therefore, the links between SAS and the academy must be built not only to handle current needs, but also with an eye toward the future.

CONCLUSION

College and university partnerships with SAS have resulted in many successes; however, as with all things, there is room to build on those successes and improve results. Any university program will benefit from an internal review of what SAS skills they most want to develop and how those can best be inserted into the curriculum through standalone SAS courses and courses that integrate SAS skills into the existing coursework. Achieving the desired outcomes of such a curriculum requires a strong relationship with SAS for software access, deployment, support, and training. As these discussions continue, it is our hope that models for successful implementations of these ideas are developed and disseminated to all interested academic programs.

ACKNOWLEDGMENTS

Thank you to Liz Moran, Senior Academic Program Manager at SAS Education, for providing thoughtful discussion during the development of this paper.

CONTACT INFORMATION

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

Jonathan W. Duggins
NC State University
jwduggin@ncsu.edu
https://duggins.wordpress.ncsu.edu/