Paper 10082-2016

A Pseudo-Interactive Approach to Teaching SAS® Programming

Hunter Glanz, Statistics Department, California Polytechnic State University, San Luis Obispo, California

ABSTRACT

With the advent of the exciting new hybrid field of Data Science, programming and data management skills are in greater demand than ever and have never been easier to attain. Online resources like codecademy and w3schools offer a host of tutorials and assistance to those looking to develop their programming abilities and knowledge. Though their content is limited to languages and tools suited mostly for web developers, the value and quality of these sites are undeniable. To this end, similar tutorials for other free-to-use software applications are springing up. The interactivity of these tutorials elevates them above most, if not all, other out-of-classroom learning tools. The process of learning programming or a new language can be quite disjointed when trying to pair a textbook or similar walk-through material with matching coding tasks and problems. These sites unify these pieces for users by presenting them with a series of short, simple lessons that always require the user to demonstrate their understanding in a coding exercise before progressing. After teaching SAS® in a classroom environment, I became fascinated by the potential for a similar student-driven approach to learning SAS. This could afford me more time to provide individualized attention, as well as open up additional class time to more advanced topics. In this talk, I discuss my development of a series of SAS scripts that walk the user through learning the basics of SAS and that involve programming at every step of the process. This collection of scripts should serve as a self-contained, pseudo-interactive course in SAS basics that students could be asked to complete on their own in a few weeks, leaving the remainder of the term to be spent on more challenging, realistic tasks.

INTRODUCTION

As the science of data, Statistics provides an indispensable set of tools and methods to countless other fields of study. While these methods possess rigorous mathematical foundations, their implementation has become increasingly computational in nature. The tidal influx of data and development of the new hybrid field of Data Science over the last decade created an unprecedented demand for skills in data analysis and management.

General quantitative skills and a basic statistical literacy used to give job seekers a noticeable advantage. Recently, these special qualifications have been amended with programming skills and a degree of literacy in computational thinking [10]. The demand for these skills coupled with the ubiquity of internet access and personal computing power gave rise to a plethora of free online web tutorials and courses like those at Code School [1], Codecademy [7], and w3schools [11]. These websites tend to offer more courses in general programming and web development languages. They do also teach about the R and Python software packages, which have recently gained prominence in the Statistics and Data Science communities.

These resources stand out as free courses about free-to-use programming languages. Additionally, the style and format of these courses differ substantially from traditional classroom or textbook experiences. Concepts and skills are presented in small, manageable pieces of an over-arching narrative built to motivate the learning of "the basics" of that language. While university courses can involve much the same thing, here learning and progression are completely student-driven and dependent. Latter topics and tasks cannot be unlocked without first completing preceding material via short programming exercises.

The alternatives to these online learning tools usually do not include university courses, but instead activities like watching video tutorials or reading through a textbook. As clear and thorough as videos and textbooks can be, they are passive learning tools. Putting things into practice and finding exercises appropriate for each step in the learning process (or chapter in the book you are reading) can be immensely challenging; effectively forcing participants to simultaneously be the teacher and the student which adds unnecessary difficulty and distraction to learning the new language.

The effectiveness of "learn-by-doing" [4] seems to explain the reason for these online courses directly involving the user and requiring evidence of some sort of understanding before allowing the user to proceed. Though SAS markedly differs from these other software packages, it maintains a rich community of support and resources. Despite its depth, however, it lacks a learning tool comparable to the interactive, online courses described above.

CURRENT RESOURCES FOR LEARNING SAS®

Staples in the arsenal of books available from the SAS Institute are *The Little SAS book: a primer* [3] and *Learning SAS by example: a programmer's guide* [2]. In the Statistics Department at Cal Poly, in fact, we use the *The Little SAS book: a primer* as the textbook for our required course in statistical computing with SAS. The introduction of the freely available SAS University Edition was a big step towards bringing SAS software to a wider group of people and at no cost. Beyond the multitude of books already sold by the SAS Institute about a host of topics, a number of e-courses and video tutorials were developed to accompany the SAS University Edition. In 2003, Mirjana Stojanovic even catalogued the free, third-party resources for learning about programming in and using SAS [8].

These books, videos, and third-party resources maintain the same issues described above. Despite all of their benefits and potentially high quality, they remain mostly passive forms of learning the material. One of the best answers to this weakness only just arrived in *Exercises and Projects for The Little SAS Book, Fifth Edition* [5]. This book of exercises and projects explicitly complements *The Little SAS book: a primer, Fifth Edition*, thus providing a unified collection of exercises and practice concerning basic programming and statistical analyses in SAS.

This pairing suits our Cal Poly SAS course well, where instructors usually fall into two camps concerning how class periods are run. One camp consists of more traditional lecture with practice exercises woven throughout; while the other camp consists of shorter initial lectures and then a large programming activity meant to be completed within that class period. The latter of these requires students to do research and teach themselves more so than the former, but allows for the instructor to spend more time giving individual attention. I usually employ the first approach: lecturing about a SAS topic, demonstrating its use and then asking students to complete a short task involving that topic for multiple topics in a given class period. Both approaches could make excellent use of the exercises and projects meant to accompany the textbook already in use.

This pairing also suits individuals in industry or academia aiming to learn about SAS, as they avoid the need to construct the course for themselves. One of the biggest challenges then consists of maintaining the discipline and motivation to complete the exercises and projects alongside their corresponding reading, instead of slipping into using just one or the other.

I am creating and developing a series of SAS scripts that emulate the courses offered by Code School and Codecademy, which will transform the learning of basic programming and statistical analyses in SAS.

NEW INTERACTIVE COURSE FOR LEARNING SAS®

The seamless integration of programming experience and practice into the learning process fills the hole in the collection of SAS resources mentioned above. To best learn programming, students need to *be* programming. My proposed collection of SAS scripts will provide a one-stop, holistic experience to learning the basics of programming in SAS:

- Interface to SAS and SAS Studio
- Introduction to the DATA and PROC steps
- The program data vector
- Statements
- Types of variables
- Arrays
- SAS functions
- Conditional logic and DO loops
- Introduction to MACROS
- Reading external data files
- LIST, COLUMN, and FORMATTED input
- Basic PROC's
- Concatenating, merging, and interleaving data sets
- ODS graphics
- Basic statistics

Creating an over-arching story that requires the use of each of these skills will engage and drive users to fully complete each lesson and exercise. The careful organization of this story into short, basic concepts will make it easy for users to easily track progress, review what they have been through, and skip around to specific topics. While not strictly prohibiting users from progressing until they complete preceding material (hence “pseudo-interactive” instead of fully interactive), these lessons will briefly introduce a new tool, demonstrate its use, and then ask for users to demonstrate their understanding in a short programming exercise that incorporates both the current and previous material.

A sample script introducing the use of variables and simple arithmetic within the DATA step is included at the end of this paper.

SELF-CONTAINED COURSE

For individuals in industry and academia, taking the learning of SAS into their own hands, this new free course should serve as a completely self-contained method of instruction. Of course the aforementioned resources including textbooks and online documentation will complement it nicely as detailed reference material, but will not be necessary for its completion.

With an extensive collection of these scripts established and vetted for teaching the basics of SAS, I envision them being packaged as an e-Book to be downloaded by anyone interested in learning SAS who has access to at least SAS Studio.

COMPONENT OF UNIVERSITY COURSE

The use of these SAS scripts to augment a regular classroom experience could significantly enhance students' learning. The integration of it into this environment could take many forms, but I am envisioning using them as homework assignments prior to the students' exposure to concepts during class. That is, for a class period on ODS graphics students will come in to class having completed the SAS script lessons associated with this topic ready to spend that period on a heavier, more challenging activity on ODS graphics.

Alternatively, students could be asked to complete the *self-contained course* (the entire collection of SAS scripts) as homework in the first few weeks of class, with accompanying assistance and guidance during class. The remaining 70+% of the course could then be spent on more advanced statistics and programming topics.

In the Fall 2015 and Winter 2016 Quarters at Cal Poly I implemented the first of the two plans mentioned above. I assigned the first draft of a small collection of these SAS scripts as pre-class activities or homework assignments, on topics that would be discussed the next class period. The scripts were short and simple, which I think contributed to near 100% completion of them by every student. Students were briefly quizzed at the beginning of each class period on the material covered in these pre-class homework scripts, for which there was also accompanying textbook reading. After a short lecture, students were assigned a programming lab in class consisting of a more challenging task than in the homework scripts.

It is perhaps impossible to attribute students' performance to these homework scripts independent of the rest of the course, but I was able to incorporate a couple custom questions concerning these scripts into the students' course evaluations. When asked to rate the quality of the homework scripts on a traditional (A, B, C, D, F) scale, 81% (34 of 42) responded with an 'A' or a 'B'. The initial reaction, at least, seems quite positive and I was very satisfied with my students' performance overall.

CONCLUSION

The immense increase in data and vast advancement of computing power over the past decade continues today, and brings with it an unprecedented demand for knowledge in statistical computing. Free courses in just about anything flood the internet, making these new proficiencies easier to acquire than ever before. While the SAS Institute and its community boast an impressive collection of documentation and learning material, a free-to-use and completely self-contained course in the basics of SAS that has users learn by doing is non-existent. I am building such course in the form of a series of SAS scripts which introduce basic concepts, demonstrate their use, and ask the user to prove their understanding by completing programming tasks themselves within the same scripts. As a self-contained home course or supplement to a classroom experience, these scripts provide those interested in SAS with a one-stop resource for learning the basics in a synergetic, hands-on way.

Initial implementations of a subset of the proposed scripts have been positively received and successful in their mission, within a university classroom.

SAMPLE SCRIPT

```
* One of the first things you probably used a computer or device for
was numerical computations. In other words, using a calculator to
perform mathematical computations more efficiently.;

* SAS is capable of some complex and quite amazing things, but we
should
understand how to at least use it as a calculator first.;

* We can use traditional notation in SAS to perform computations: +, -
, *, /.
SAS adheres to the traditional order of operations (PEMDAS) as well:
parentheses (), exponents **, multiplication *, division /, addition
+,
 subtraction -.;

data math;

* Notice that we can use variables to compute the values of other
variables.
```

```
As long as we're doing math we should be using numeric variables.;
annualsalary = 60000;
monthlyrent = 1600;
annualrent = monthlyrent * 12;

* Proportion of salary used for rent.;
rentprop = annualrent / annualsalary;

run;

proc print data = math;
run;

* In the space below write your own DATA and PROC PRINT steps to store
or
compute the following:
- your annual salary
- your montly and annual rent
- your monthly and annual utilities
- your monthly and annual food cost/budget
- the proportion of your salary that each of the above three things
make up;





* When your DATA and PROC PRINT steps look good, go ahead and run this
SAS
script. In the first lesson you should have noted the presence of the
Editor,
Log, and Output windows. So far we've only looked at the Editor and
Output
windows. The Log window is useful for checking to make sure SAS did
not
encounter any errors in running the script. If your output looks
different
than you expect, the Log is a good place to look first.;
```

REFERENCES

1. "Code School." URL = *http://www.codeschool.com/* (2015).
2. Cody, Ron. *Learning SAS by example: a programmer's guide*. SAS Institute, 2007.
3. Delwiche, Lora D., and Susan J. Slaughter. *The Little SAS book: a primer*. SAS Institute, 2012.
4. Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics.*Proceedings of the National Academy of Sciences of the United States of America*, *111*(23), 8410–5. doi:10.1073/pnas.1319030111
5. Ottesen, Rebecca A., Delwiche, Lora D., and Susan J. Slaughter. *Exercises and Projects for The Little SAS Book, Fifth Edition.* SAS Intitute, 2015.

6. Resnick, Mitchel, et al. "Scratch: programming for all." *Communications of the ACM* 52.11 (2009): 60-67.
7. Sims, Zach, and C. Bubinski. "Codecademy." *URL= http://www.codecademy.com* (2011).
8. Stojanovic, Mirjana. "Efficient Way to Learn SAS® with Virtually No Cost."*SESUG 2003 Proceedings* (2003).
9. Thompson, Clive. "How Khan Academy is changing the rules of education."*Wired Magazine* 126 (2011).
10. Wing, Jeannette M. "Computational thinking and thinking about computing."*Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 366.1881 (2008): 3717-3725.
11. "W3Schools Online Web Tutorials [online]. 2014 [cit. 18.10. 2014]." *Dostupné z: http://www.w3schools. com/* (2014).

CONTACT INFORMATION

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

Name: Hunter Glanz
Enterprise: Statistics Department, California Polytechnic State University
Address: 1 Grand Avenue
City, State ZIP: San Luis Obispo, CA 93407
Work Phone: 805-756-2792
E-mail: hglanz@calpoly.edu