

The Power of Know-How: Pump Up Your Professional Value by Refining Your SAS® Skills

Gina Huff, Lumeris, St. Louis, Missouri

ABSTRACT

If you have been programming in SAS® for any length of time, you are quite familiar with the slogan, “Power to Know.” After more than 15 years of programming in this robust solution, it goes without saying that I am a firm believer in the truth of this statement. Undoubtedly though, there is a second unspoken and critical truth tied to the success of SAS within an organization—the power of SAS is nothing without the users’ “Power of Know-How.” Developing a strong foundation in programming best practices; writing stable, efficient code; and building automated solutions are all ways in which you can begin to fully appreciate the power of SAS (and make yourself more appealing to employers to boot). There is a difference between having a working knowledge of SAS and being able to really program in SAS. As someone who has grown as a SAS programmer over the years and now teaches SAS in an online graduate certificate program, I have either lived or witnessed many of the hurdles and stumbling blocks that folks can encounter when they want to transition from an ad hoc user of SAS to a more technical practitioner. This paper provides practical tips on how to add to your SAS toolbox and how to apply that knowledge to create production-ready code. A deeper understanding of key SAS concepts helps you become more effective in your current organization and it might even land you the ideal job.

INTRODUCTION

My first order of business is to correct a calculation error I made in the abstract. After checking multiple times though, alas, I cannot. I have indeed been coding for 15 years. Although it's hard to believe that I have been glued to a computer screen for that long, it is certainly a rewarding journey, albeit an occasionally frustrating one. There are days that I am doing a happy dance at my desk because I solved a difficult problem and others when I am but a second away from heaving my computer out the window. That being said, I hope this paper will assist beginner and/or casual users of SAS who wish to grow their competency to enjoy more “happy dances” with fewer frustrations. I have a similar wish for managers who are helping cultivate the next generation of SAS developers—for you, I also hope to increase the number of good days but, perhaps more importantly, to decrease the number of days you would like to throw the computer (and perhaps the programmer) right out the window.

I have been fortunate to have had several different experiences throughout my career. I spent many years in academia, which is where I learned the craft. I was in an environment that embraced education and encouraged me to learn, explore, and find new ways to present data. One drawback of working in education, however, was the undeniable fact that we were not as competitive when it came to salary. The atmosphere and the benefits were incredible, but being unable to compete financially, we found that 1) we simply couldn't attract experienced SAS programmers and 2) once a hire was with us for a few years, they would often leave for more lucrative offers. We knew that in order to ensure our office remained successful, we had to get new hires up and going quickly and ensure projects were standardized for ongoing viability.

Additionally, I had the opportunity to begin teaching foundational and advanced SAS courses for an online graduate program as a result of my practical work experience. This has afforded me another perspective. Seeing firsthand the knowledge acquired by students, I appreciate the foundation that is set in these programs, but I also see some of the gaps that must be filled when a student fresh from such a program enters the workforce. In the interest of time and sticking with the curriculum, students often get very distilled data and problems to work with. Let's be honest, real data and projects are messy. Sometimes the hardest part of programming has nothing to do with the actual code on the page; the hardest part could be finding

out what the problem is to begin with. These are the gaps that we must strive to fill when transitioning from a SAS beginner to one with “know-how.”

I now work at a firm that is committed to improving healthcare, something in which I passionately believe. This new experience has afforded me yet another perspective. The corporate world is fast-paced, the data big, and programs complex. I have met people with shared passion, but not necessarily shared coding experience. I hope to continue to assist my colleagues to understand the difference between coding for exploration and coding for consistent production. Concepts like scalability, automation, and efficiency are what we strive to improve upon. My desire is that this paper shares tips to help you grow as well.

Now, let me tell you what this paper is not. Although I will bring up many coding skills that SAS developers should acquaint themselves with, I won't be going into much detail about them. After all, many of the concepts have entire books and/or papers dedicated to them. This is just a collection of ideas to help guide your journey. What I will do, however, is direct you to some of the resources that I found helpful in my own journey. Additionally, this paper is not for someone who lacks a basic understanding of SAS. If you don't know what a data step is, have no idea what a function does, or think a macro is a course you struggled with in economics, then go ahead and set this paper to the side, read a few SAS texts, take a course or watch some online SAS training, and get a basic understanding before you read on. If you have that foundation and have decided to stick with me though, I hope that you enjoy some of the SAS “truths” I have discovered over the years and use them to expand your own SAS “know-how.”

SAS TRUTH #1: CUTTING CORNERS WON'T SAVE TIME, IT WILL WASTE IT

I am guilty of it myself—a deadline is looming and so I take shortcuts and/or don't use the techniques I have learned over the years just to “get something out.” I have found that although this may satiate a need in the short-term, it never saves time in the long run. These first tips may seem basic, but I assure you, they are SAS coding essentials. Anyone that has ever seen code that is lacking in any (or heaven-forbid all) of these elements will vehemently agree. Trust me, if you want to anger your colleagues and frustrate your manager, write ugly code. If you want to get ahead, however, learn how to write logically sound and aesthetically pleasing code.

“You must spend time to save it.” —Gina Huff (that's me!)

COMMENTS ARE CRUCIAL

I know, I know, you're thinking that this Gina person just jumped right in and started lecturing, right? Maybe a little, but I promise that I am trying to save you from some of the same bumps in the road that I have stumbled over through the years. Commenting is SUPER IMPORTANT! **Always** start your program with a Header that lists key information and be super diligent about placing comments throughout the code. I comment before **each** new step, explaining what that step does. Even if you think it is self-explanatory, comments help save time in the long run making second-leveling and maintaining the code so much easier.

I can't tell you the number of times that I have inherited someone else's code or even looked AT MY OWN CODE and had no idea what the purpose of a step was, had to track logic back multiple steps like a maze, or had to research the intent of a process because it wasn't documented. Reviewing and understanding undocumented code can take hours to decipher. Take a few minutes up front to comment and save loads of trouble down the road.

While in the process of writing, you have an intimate knowledge and understanding of the “why”—so DOCUMENT IT RIGHT AWAY. If there is any logic that is unusual at all, make sure you explain why you coded it that way. The next person to use your code isn’t a mind reader and I guarantee that he/she will thank you for the guidance, even if that next person is YOU. If you think you will remember the “why” a month from now or a year from now, I assure you, you won’t. Document, document, document.



TIPS AND TIDBITS: Comment why ya’ did; comment why ya’ didn’t!

I have certain procedures or steps that I automatically include options with. A quick example is using the MISSING option in the Tabulate Procedure. I typically include it because I want to capture all observations and see if there is missing data. If I elect not to use that option, it is a conscious decision that should be documented. Otherwise, I or someone else may go through and assume that failing to include the MISSING option was an oversight. If it is something you ‘always’ do, ‘always’ document when you choose not to use it.

```
Comment when you DON'T use a standard option:  
proc tabulate data=data1 /*didn't use  
MISSING because requirements specified  
to exclude all null values*/  
class ...
```

HIT THE BEAUTY BUTTON

Oh, how I wish there was a beauty button that organized and structured your code with the press of a button. In lieu of a magic button, get into good habits when laying out your code. Just like when you were growing up, your mom strongly encouraged you to keep a tidy room, this SAS Mom recommends that you maintain tidy code. If code is messy, inconsistent, and hard to follow, it will become more prone to errors, less manageable, and harder to troubleshoot or update. Furthermore, this SAS instructor is adamant about good coding habits because I cannot stand grading ugly code.

When you code, think vertically (it is so much easier to see) and take advantage of indentions. A colleague of mine just introduced me to automatic indention, which is a great help. If you highlight your code and press CTRL+i, then SAS will “automagically” line break and indent the code. Although it doesn’t do it exactly the way I would, it is a start and certainly much better than having an entire step on one line! (Ick-my left eye just twitched a little thinking about it.)

This is just the tip of the iceberg. Using standard naming conventions, order, flow, and organization are all issues that you need to consider when developing your coding style. I recommend the following paper to research this topic further. It is an oldie but a goodie. I actually attended this session in 2006 as a relatively new programmer. I immediately went back to the office and completely changed the way I wrote code. The physical structure and standardizations made a huge difference in making the code clear and consistent. The “readability” factor is an undeniably effective timesaver.

Levin, Lois (2006). “SAS Programming Guidelines.” *Proceedings of SUGI 31*. Available at <https://support.sas.com/resources/papers/proceedings/proceedings/sugi31/123-31.pdf>.

If you are interested in a more recent take, this is another a fantastic paper that reinforces much of what Lois stressed in her paper. You will see that proper coding structure is timeless.

Laufner, Kirk Paul and Rosenbloom, Mary (2017). “Best Practice Programming Techniques for SAS® Users.” *Proceedings of the 2017 SAS Global Forum Conference*. Available at <https://support.sas.com/resources/papers/proceedings17/0175-2017.pdf>.



TIPS AND TIDBITS: Let Your Commas Dangle

This is a little programming standardization tip that has helped me save time looking for syntax errors. It is particularly useful in PROC SQL code. In the past, I was probably the victim of missing comma errors as much or more than missing semicolons in my code. However, by putting the commas FIRST in a new line of code rather than the end of the previous line, I can quickly check to make sure all my commas are in place. It provides the added benefit of allowing me to easily comment out individual variables if needed.

```
SQL Code with dangling commas:  
proc sql;  
  create table tip1 as  
  select  
    name  
    , id  
    , address  
    , gender  
  from data1;  
quit;
```

TAKE THE TIME TO SHORTEN YOUR LETTER

Mark Twain once wrote a letter apologizing for its long length, stating that he would have written a shorter one if he'd only had the time to do so. Short and simple is not synonymous with basic or easy. It is often "easier" to write long, complicated code. I encourage you to take the time to lay out what your goals are for the project and then build a roadmap to accomplish that goal. I **never** start a project by pounding on the keyboard. First, I get the requirements and make sure I understand the objectives. I determine what data are needed and make sure I can access and understand the datasets I am using. Next, I develop a plan of action on how to code for a solution. If a visualization is needed, then I draw out a template of how the report should appear. I strive to have a definitive picture of the result before I open SAS. Actual coding is one of the last steps.

Wouldn't it save time to just start coding you ask? The short answer is no. You may get started faster, but in doing so, you will push the finish line out further because the odds of you taking the shortest path is nominal. If you don't know where you are going before you begin, then the likelihood of taking a wrong turn, going on wild goose chases (been on plenty of those!), or taking the scenic route to get there is great. Surprisingly, I have inherited code where a dataset was created and then never even used. How does that happen? Easy. No road map.

Take it from me: map first, code second. As you get more proficient and depending on the complexity of the task, it may just be a road map in your head—for example, you probably don't have to reference a physical map to get to work every day. Similarly, if I need to do a quick filter of some data and output a cross tabular report, then I can just plan that out in my head and execute. If the task is new and complex, however, writing out the plan becomes more crucial—I wouldn't take off to a brand new city across the country without mapping my route. By the same token, if I take on a project that requires me to ingest millions of records from brand new data sources with data transformations, multiple joins, and a complex series of reports to output, then I'm heading to the whiteboard to brainstorm and map.

"In God we trust; all others must bring data." — W. Edwards Deming

KNOW THY DATA

I debated at first even including this—it's common sense, right? But I just don't think it can be said enough. I still have students that will complete an exercise completely wrong and had they just LOOKED at the data, they would have known in an instant that they had made a logical error. If you are asked to categorize by ranges, for example, double check your answer by reviewing the data. If your report lists that there are 5 stores selling between \$100,000-\$200,000 in red high heel shoes during May but the highest store sales amount you can find in the data is \$2,500, guess what? You coded something wrong.

You need to review the data, understand variables, variable types, primary keys, code values and where to find their descriptions. The best way to do this is with a data dictionary. If one exists, study it. If one does not and this is data that you will be using on an ongoing basis, create one. It takes time now but will pay dividends later. And it doesn't hurt for you to be deemed the go-to guy for the data either—get yourself noticed. If you have any question about where to start, look no farther than KNOW THY DATA!



TIPS AND TIDBITS: PROC CONTENTS to Kick-Start your Data Dictionary

Anytime I create a data dictionary, I use PROC CONTENTS (with the VARNUM option) as my starting point. CONTENTS provides a wealth of information including names, labels, data type, length, formatting, and indexing and by using varnum, you get the variables listed in the order of appearance in the table. You can write the results to a dataset using OUT= and add additional information as needed. Some variables that I commonly add are source of data, format reference (if there is a format that can be used but isn't already assigned to it), and date added. Use the output delivery system (ODS) to write out the data dictionary and people will think you worked for hours on it. As an added benefit, maintenance and update is a breeze.

It should come as no surprise that there is a wealth of information out there about exploring your data. Here's a cool paper to get you started:

Kuligowski, Andrew T and Shankar, Charu (2013). "Know Thy Data: Techniques for Data Exploration." *Proceedings of the 2013 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings13/145-2013.pdf>.

I would be remiss if I failed to mention the next book as a valuable resource, both to get to know the data as well as to clean it. Great read and even greater resource. Purchase it so you can highlight, bookmark, and write notes all over it!

Cody, Ron (2017). *Cody's Data Cleaning Techniques Using SAS*. SAS Institute, Third Edition.

SAS TRUTH #2: THE SMARTEST CODERS ARE THE LAZY ONES

What??? Is that a typo in the second SAS Truth? Actually, it isn't. I say all the time that programmers are lazy and that's a GOOD thing. I don't want a developer that is perfectly content having to manually type the same code over and over again. I don't want a co-worker to enjoy creating multiple Excel reports from scratch every month using data derived from SAS output. Heck, I don't even want to have to remember to run a program at a specified interval if I can let SAS do it for me. Yes, I am a lazy programmer and proud of it. (And if you don't like the tongue and cheek quote, that's OK, you can just say you prefer to "work smarter, not harder"—same difference.)

"Progress is made by lazy men trying to find easier ways to do something." — Robert Heinlein

IF YOU HAVE TO TOUCH IT, THEN IT ISN'T AUTOMATED

I'm sure some folks will disagree with that statement and I certainly admit that there are different levels of automation. It isn't always possible to fully automate in every situation. However, let's agree that our goal should be to automate as fully as possible. I will expand that further to say that in numerous situations full automation is completely possible and when possible, almost always ideal.

If you know that you need to run a program every Monday at 8 a.m., then put it on a scheduler to run "automagically." If you need to run a report for every client, don't you dare run it n times, use a macro and loop through each client. Let SAS do the heavy lifting.

That, of course, leads me to the best path toward automation, my number one recommendation: SAS MACROS. Macro programming affords you the opportunity to build code once in a way that is flexible enough to expand or shrink as demands change; to be able to define clients, date ranges, and categories on the fly; and to be able to loop through and create data on multiple clients, date ranges, and/or categories at once.

Macros can be as easy as defining %LET statements at the top of the code so that you can define parameters one time and then let the macro resolve to that specification throughout the code. This process is simple but already quite powerful. Master that basic technique first, but please don't stop there. Macros will add so much to your SAS know-how—they are truly SAS power tools. Who wants to grab a screwdriver to do a big job when with a little effort, you can learn to use a power drill? I guarantee it's worth the hard work and time you invest in learning about macros to truly embrace a lazy programmer mentality.

I'm going straight to the books on this one. Just read them cover-to-cover. You're welcome.

Burlew, Michele (2014). *SAS Macro Programming Made Easy*. SAS Institute, Third Edition.

Carpenter, Art (2016). *Carpenter's Complete Guide to the SAS Macro Language*. SAS Institute, Third Edition.

This last recommendation is thanks to the teacher inside me. It is the Advanced Certification Guide. It has great examples, quizzes to test your understanding, and will prepare you for the Certification test (Obviously, get your Base Certification first, but then focus heavily on the SQL and Macro portions of the advanced book—you will be a SAS programming powerhouse before you know it!)

SAS Institute (2014). SAS Certification Prep Guide: Advanced Programming for SAS 9. Fourth Edition.

IF ALL YOU HAVE IS A HAMMER, THEN EVERYTHING LOOKS LIKE A NAIL

If there's one thing that I've learned about SAS over the years is that there are multiple ways to solve a problem. If, for example, you need to join data, there are a multitude of options: a PROC SQL join, a DATA merge, hash, formatting, etc. Which is best? My answer to most questions like this is, "It depends." Sometimes a simple merge or join is the quickest and most efficient solution, other times the power of hash will greatly improve run time, and still other times creating a user-defined format and a PUT function will fit the bill. Often, more than one solution is equally effective, so consideration of other data manipulation and/or transformation requirements will help determine the best join strategy. This brings me to my next tip—learn multiple solutions so that you can solve problems effectively with the BEST tool possible. Don't just bring a hammer to the build site because when you get there, you may need a sledgehammer instead. I have seen too many programmers content to let a program run for hours and hours because all they have in their toolkit is the join or merge, when using hash would save hours in runtime. Conversely, though, don't hold on to that sledgehammer too tight, because it doesn't really pound a staple into wood well. Use a staple gun for that. If you are simply trying to add one variable to a table with a single primary key, then is hash code really the most effective use of your time? Maybe not. Using a format is a clever and viable solution. Again, it is a matter of having more than one tool and then KNOWING how to use them that is key...it's all about the power of know-how. My husband always says that it is important to have the right tool for the right job—it makes the job easier!

Clearly, I am a fan of both hash and format. I say those are key elements that you should know and use, so how about some suggestions for further reading? The first paper will get you started with hash:

Schacherer, Chris (2015). "Introduction to SAS Hash Objects." *Proceedings of the 2015 SAS Global Forum Conference*. Available at <https://support.sas.com/resources/papers/proceedings15/3024-2015.pdf>.

The next paper is on PROC FORMAT. This is a paper I printed off and had beside my desk for years: an obvious favorite of mine as evidenced by its crumpled, used appearance and maybe even a diet coke stain or two. I remember thinking that his paper was so informative that he could write a book. Well, low and behold, he did: a short book (now out of print unfortunately) that is jam packed with information called *The Power of PROC Format*. Trust me when I say, PROC FORMAT may be touted for making things "pretty," but don't be fooled. Underneath the "pretty" is one powerful procedure.

Bilenas, Jonas (2008). "I Can Do That With PROC FORMAT." *Proceedings of the 2008 SAS Global Forum Conference*. Available at <https://support.sas.com/resources/papers/proceedings/pdfs/sgf2008/174-2008.pdf>.

IS THERE SUCH A THING AS TOO MUCH CHOICE?

OK, so now I am contradicting myself, right? I just got finished with a section talking about understanding all the options and adding to your toolkit, but I will say that all good things have limits. SAS definitely has a lot of ways to solve a problem, but at times and particularly for a new programmer, that choice can become a bit overwhelming. It is for this reason that I suggest that you find a few things that you want to master first and do just that—master them. Strive for depth of knowledge in niche areas and be satisfied with a more superficial understanding in areas that you don't need as often.

If, for example, you know that your job entails projects where you need to create complex cross-tab reports, then commit yourself to learning the ins and outs of PROC TABULATE, PROC FORMAT, and the Output Delivery System (ODS). If you are going to be working with big data and need to find ways to pull in data faster, then start researching MPCConnect and parallel processing. Research ways to sample your data so that you can maximize efficiency. Be the master of that process. Find the things you use often, excel at them, and commit them to memory. Don't stress the rest; we are lazy programmers after all.

DON'T REINVENT THE WHEEL

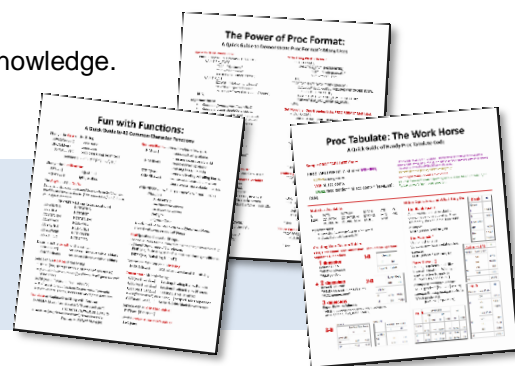
The great thing about SAS is that you can constantly add to and build on what you know. You can reuse code and do so effectively by creating a code library that you can reference as you work on different projects (even better to build some of the code as macros and create a macro library but take it one step at a time). I am also a big fan of “cheat sheets.” For example, I have a functions cheat sheet that lists the character functions I use most, along with the options I need to be able to use them. Again, it was an investment of time, but I—and some of my colleagues both past and present—have been using them for years, making that investment payoff huge! As an added benefit, you will position yourself as that “go-to” guy, with helpful resources and SAS knowledge.

Long story short, BUILD on what you know and develop a depth of knowledge.



TIPS AND TIDBITS: Cheat Sheet Your Way to the Top.

SAS is too vast to remember everything and as good as their resources are, I have found it INCREDIBLY helpful to compile the elements I use most in a given component and develop a one page “cheat sheet.” Make it short, concise, and include those elements you use often to be of the most benefit.



SOMETHING OLD, SOMETHING NEW, SOMETHING BORROWED, SOMETHING BLUE

OK, we've already got the something blue—right in the SAS logo color (Pantone 285 to be exact), but what do I mean about the something old, new, and borrowed? Let's start with two at once—something old and something borrowed. Don't be afraid to use legacy code as a starting point. You don't have to reinvent the wheel. “Borrow” code snippets and logic from people you trust. Review how someone else solves a specific problem and use that as your jumping off point.

Don't forget the something new, however. Make sure that the something old is just your starting point. Oftentimes, legacy code can be improved upon and as SAS is always adding new options, features, and functions there may be a better way to do things. It doesn't mean that the original code was wrong, it just stresses the importance of staying up on upgrades. This topic reminded a colleague of mine of previous work experience. Even though ODS Excel had been in production for quite some time, many people were still using DDE because they weren't aware of the change—they didn't embrace the “something new.” Similarly, I recently encountered a large block of code that included macros and accessing dictionary tables to get a list of variable names. I was able to replace those many lines of code with the vname function—this a function that pulls variable names. If I hadn't stayed on top of function options and new updates, then I may not have been aware that existed. (I must add here, with hundreds of SAS functions available, if you think it should be possible, it likely is. Check the functions list in SAS Help. Here's a link to 9.4 functions: <https://documentation.sas.com/?docsetId=lefunctionsref&docsetTarget=n01f5qrjoh9h4hn1olbdpb5pr2td.htm&docsetVersion=9.4&locale=en>)

SAS TRUTH #3: THE BEST PROGRAMMERS KNOW HOW DUMB THEY ARE

Oh my goodness, now I've resorted to insulting people. Certainly not! My third SAS truth is just a feeble attempt to humorously express how no one is ever "done" learning. It signifies that no programmer knows everything there is to know about SAS. The best developers embrace learning every single day and constantly strive to improve their code and processes. Just yesterday a gentleman at work was overly generous by calling me a "SAS Expert", a compliment I kindly rescinded. I, too, recognize just how "dumb" I am, remembering a statement made often by one of my professors, "The sign of a smart man is that he knows what he is dumb about."

"An investment in knowledge pays the best interest." — Benjamin Franklin

THE PROTÉGÉ EFFECT IS NOT A STEAMY ROMANCE NOVEL

Oh no, the protégé effect is quite real. This term suggests that if one teaches (or even just intends) to teach a subject to others, they learn it better than someone who does not as evidenced by test performance. If you are serious about learning SAS, then I suggest you find a way to teach it to someone else. In your company, that may mean volunteering to put together the aforementioned cheat sheets and then presenting them at a staff meeting or even better a "lunch and learn" or similar training session. It could mean that you help onboard new employees and give them a brief overview of common SAS techniques used at your company.

Expanding outside your business, it could mean volunteering at a local user group, submitting and presenting a paper at SAS Global Forum, or even volunteering to speak at your local high school's technology club. The main thing is that it motivates you to learn more about the topic, you are forced to organize and structure your thoughts, and you communicate good techniques to achieve your goal. Learning by teaching is truly a win-win!

If you want to present at a Global Forum, I have another oldie but goodie for you. Some advice is just timeless:

Rhodes, Dianne (2004). "So You Want To Write A SUGI Paper? That Paper About Writing A Paper." *Proceedings of SUGI 29*. Available at <https://support.sas.com/resources/papers/proceedings/proceedings/sugi29/145-29.pdf>.

A more recent publication that is also good (but quite different, so read both):

Billings, Thomas (2014). "Suggestions for Writing Better SAS User Group Papers." *Proceedings of the 2014 WUSS (Western Users of SAS Software) Conference*. Available at https://www.lexjansen.com/wuss/2014/5_Final_Paper_PDF.pdf.

(BTW, please send me an email if you elect to present at the next Global Forum—I would love to attend and cheer you on!)

THE TRUTH IS OUT THERE

There is a wealth of information right at your fingertips and it is yours for the taking. The best takeaway I can give you is to use the vast amount of information that is at your disposal to customize your SAS education. There are formal certification programs, a wealth of SAS training, online tutorials, and face-to-face seminars that can help you build a foundation in SAS. Additionally, there are numerous books, papers, presentations, and websites that delve into specific topics and provide practice examples and applications. SAS Support and the SAS Community should be in your browser favorites and yes, I'll say it, Google is your friend. I often just google a topic and find that the very first result provides the answer to my question. A unique option is the online game *SAS Sensei*, my friend and co-worker is a bit of a Sensei addict I'm afraid, and that is yet another effective and quite fun way to broaden your SAS skillset. The fact that you have read

this paper proves that you are committed to learning and you have already started that journey by exploring Global Forum. Kudos to you!

Here is an exceptional paper that provides an abundance of SAS learning tools and educational resources:

Laufler, Kirk Paul (2016). "Differentiate Yourself." *Proceedings of the 2016 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings16/2742-2016.pdf>.

Keep learning and keep trying. Remember that there are people out there ready and willing to help as long as you continue to push yourself and be proactive. You cannot learn SAS in a vacuum and you cannot learn SAS by osmosis. Actively and aggressively pursue learning through all the means mentioned above.

CONCLUSION

Thank you so much for going on this little journey of SAS Truths with me and I beg your indulgence if the topics seemed scattered or incomplete. SAS is such a wonderful, robust language that it is hard—impossible really—to distill everything you want to say down to a few pages of text. I encourage you to let this be but a jumping off point for your SAS journey as you begin to build your tool kit (and/or help others build theirs). I trust that you will find SAS to be as powerful tool as I have over the years and hope that you will choose to share your SAS journey with others. It is a wonderful community and it is the creativity and ingenuity of its members that have helped make SAS the success it is. SAS truly possesses the "Power to Know" as long as it has users' continued commitment to harnessing the "Power of Know-How."

REFERENCES

All references were also identified in the main paper as suggested further reading for the applicable topic.

Bilenas, Jonas (2008). "I Can Do That With PROC FORMAT." *Proceedings of the 2008 SAS Global Forum Conference*. Available at <https://support.sas.com/resources/papers/proceedings/pdfs/sgf2008/174-2008.pdf>.

Billings, Thomas (2014). "Suggestions for Writing Better SAS User Group Papers." *Proceedings of the 2014 WUSS (Western Users of SAS Software) Conference*. Available at https://www.lexjansen.com/wuss/2014/5_Final_Paper_PDF.pdf.

Burlew, Michele (2014). *SAS Macro Programming Made Easy*. SAS Institute, Third Edition.

Carpenter, Art (2016). *Carpenter's Complete Guide to the SAS Macro Language*. SAS Institute, Third Edition.

Cody, Ron (2017). *Cody's Data Cleaning Techniques Using SAS*. SAS Institute, Third Edition.

Kuligowski, Andrew T and Shankar, Charu (2013). "Know Thy Data: Techniques for Data Exploration." *Proceedings of the 2013 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings13/145-2013.pdf>.

Laufler, Kirk Paul and Rosenbloom, Mary (2017). "Best Practice Programming Techniques for SAS® Users." *Proceedings of the 2017 SAS Global Forum Conference*. Available at <https://support.sas.com/resources/papers/proceedings17/0175-2017.pdf>.

Laufler, Kirk Paul (2016). "Differentiate Yourself." *Proceedings of the 2016 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings16/2742-2016.pdf>.

Levin, Lois (2006). "SAS Programming Guidelines." *Proceedings of SUGI 31*. Available at <https://support.sas.com/resources/papers/proceedings/proceedings/sugi31/123-31.pdf>.

Rhodes, Dianne (2004). "So You Want To Write A SUGI Paper? That Paper About Writing A Paper." *Proceedings of SUGI 29*. Available at <https://support.sas.com/resources/papers/proceedings/proceedings/sugi29/145-29.pdf>.

SAS Institute (2014). *SAS Certification Prep Guide: Advanced Programming for SAS 9*. Fourth Edition.

Schacherer, Chris (2015). "Introduction to SAS Hash Objects." *Proceedings of the 2015 SAS Global Forum Conference*. Available at <https://support.sas.com/resources/papers/proceedings15/3024-2015.pdf>.

Some Other Papers by Gina Huff:

Huff, Gina (2016). "An Excellent Journey: Exploring the New ODS EXCEL Statement." *Proceedings of the 2016 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings16/2780-2016.pdf>.

Helbig, Tuesdi and Huff, Gina (2014). "From Providing Support to Driving Decisions: Improving the Value of Institutional Research." *Proceedings of the 2014 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings14/1448-2014.pdf>.

Huff, Gina (2014). "Simple ODS Tips to Get RWI (Really Wonderful Information) from your RWI (Report Writing Interface)." *Proceedings of the 2014 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings14/1689-2014.pdf>.

Huff, Gina (2011). "Absolutely Fabulous: Tips on Creating a Publication-Ready Report Using ODS Absolute Layout Functionality." *Proceedings of the 2011 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings11/293-2011.pdf>.

ACKNOWLEDGMENTS

A special thank you goes out to the SAS team at Lumeris. Thank you so much for being such an incredible group of colleagues and for passionately and generously helping me brainstorm on pertinent SAS topics. **Team Goodnight** (great name, right?) includes **Chris James, Chandra Ailneni, Jimmy Ardiansyah, Taylor Blaetz, Diane Nieman, Tom Platt, and Oleg Zakharov**. Again, many thanks!

CONTACT INFORMATION

Gina Huff
Principal SAS Engineer
13900 Riverport Dr.
Maryland Heights, MO 63043
ghuff@lumeris.com



SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.