PharmaSUG 2017 - Paper TT04

SAS ® Studio: We Program

Jim Box, SAS Institute, Cary, North Carolina
Matt Becker, SAS Institute, Cary, North Carolina

ABSTRACT

Have you investigated SAS® Studio? From the 1980s into the 2010s I used SAS Display Manager (PC SAS front end) for all of my clinical table, listing, figure and database program development. I became accustomed to the program editor, log window, output window and being able to view my working and saved datasets via this programming IDE. I resisted new coding editors through the years UNTIL SAS® Studio came to fruition.

SAS Studio is a web-based application that accesses your SAS environment – cloud, local server(s), grid or PC. With the environment, you can access your data libraries, files and existing programs … and write new programs! Additionally, SAS Studio contains pre-defined tasks that generate code for you. Have a specific set of clinical programming code you always use? Snippets! Want to define a personal or global task for AE table summarization? Define it within SAS Studio!

INTRODUCTION

SAS Studio is the web browser development interface that allows you to access your data and run code from any machine, anywhere. It's designed with programmers in mind, and has many features that make programming more efficient that we will discuss in this paper. Highlights for clinical programmers are: (1) a dataset viewer that allows you to hide columns, perform elaborate filters, and save the code that executed the filter, (2) a coding editor with auto-formatting, logs that link directly to the errors, and a submission history to roll-back submitted code, (3) a code snippet system that comes with several existing code blocks and allows you to store and share your own, and (4) a task system that generates SAS code for you based on helpful wizards.

SAS Studio was first released in March 2014, and is included in installs of SAS version 9.4. It can run locally, with a URL that points to your local SAS install, or users can access a server (or the grid manager) by nearly any web browser from any operating system by using their server credentials. SAS Studio can also be integrated with a metadata server, allowing granular permissions for data access and for metadata defined library access.

DATASET BROWSING

Let's start by looking at Studio's dataset browser (Figure 1). In the left-hand panel, you can browse libraries like you would in a file system. When you select a library, it expands to show you all of the datasets. When you expand the dataset, you see all of the variables.

Once you double click to open a dataset, a new tab will appear showing the Column information and the data table. In the column information panel, you can uncheck columns to remove them from the dataset view, and you can view the column metadata for a selected column (label, name, length, type, format, and informat). If you prefer, you can collapse the column information panel to give yourself more viewing area for the data.

When the dataset is open, the first 100 rows will be visible on the screen, and you will be able to page through the rest of the data in pages of 100 records. This is done to speed up the loading of data if you are accessing a server. The total number of rows and columns is always displayed. You can also easily apply filters to the data view, either by clicking the filter funnel and writing valid SAS expressions or by right-clicking a column name and adding it to the filter. You can filter multiple columns very easily. The number of filtered rows will be displayed, as well as information about the filter. Studio will keep this filter on the data until you clear it, even if you exit the Studio session. You can also view the SQL query that was used to create the filtered view, in case you want to save the code to apply to a program.

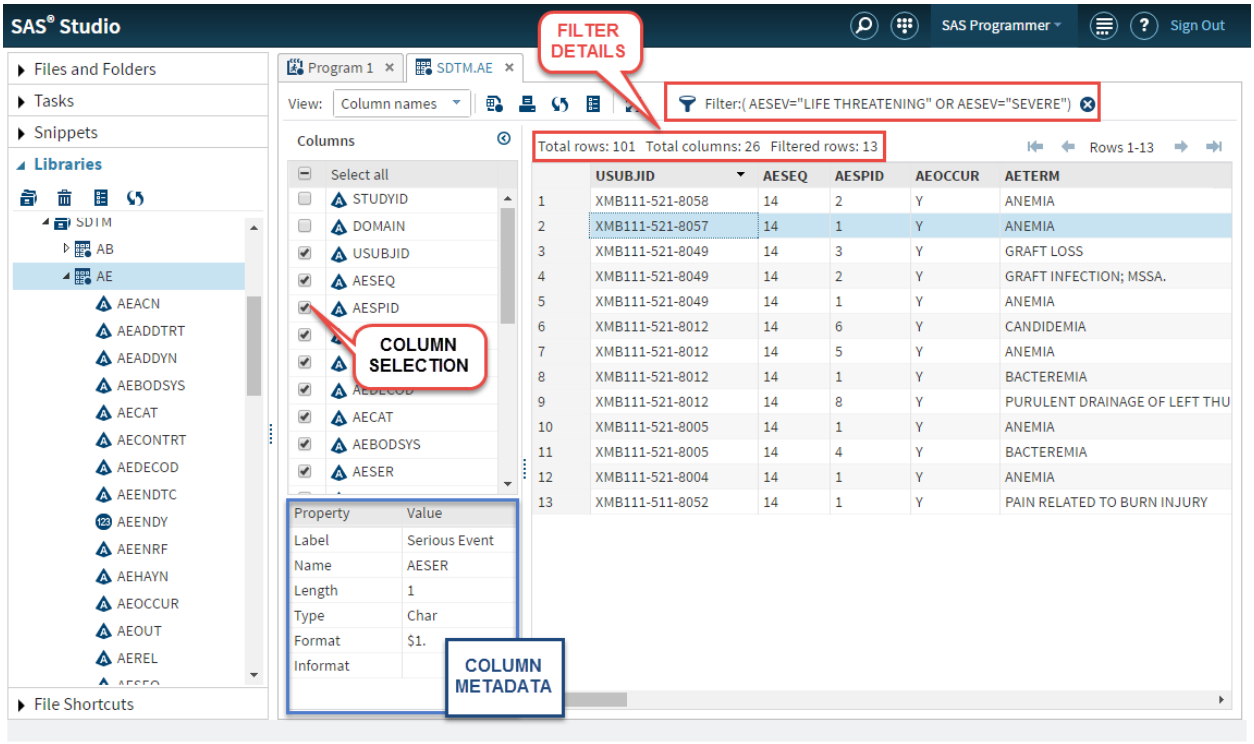

Figure 1. Library and Dataset Browser

CODE EDITOR

The code editor is where you will spend most of your time, and there are some very nice features in Studio to make programming easier. Figure 2 shows the code editor. Note that the code, log and results windows are part of a tabbed display system, making it very easy to switch between them. If you have the screen real estate, you can also separate the tabs and view two in the same window. The code uses the same syntax highlighting as the Windows Display Manager. In Figure 2, I've called out one of my favorite features – the code formatting button. As programmers, we know that properly formatted code is much easier to read and understand, but being busy means we don't always take the time to properly format the code. SAS Studio will do that for you at the press of a button, as you can see in Figure 3.

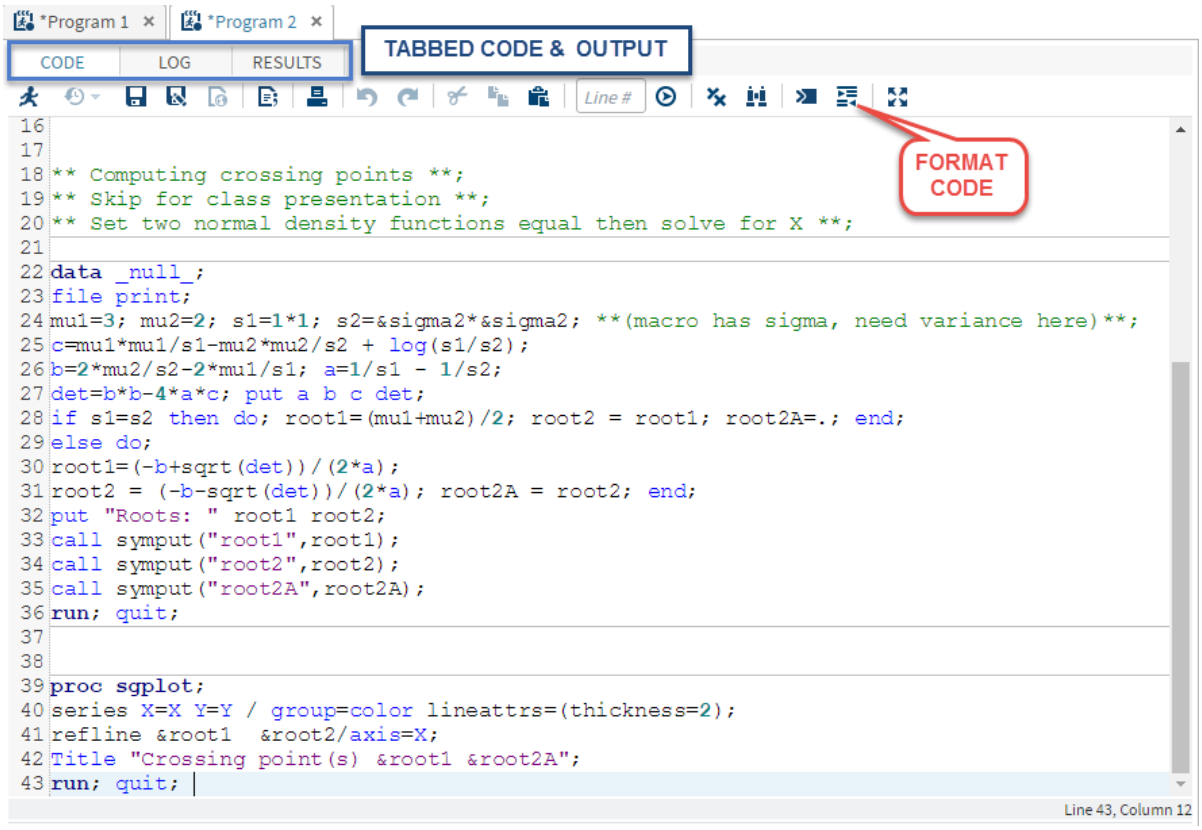

Figure 2. Code Editor with Unformatted Code

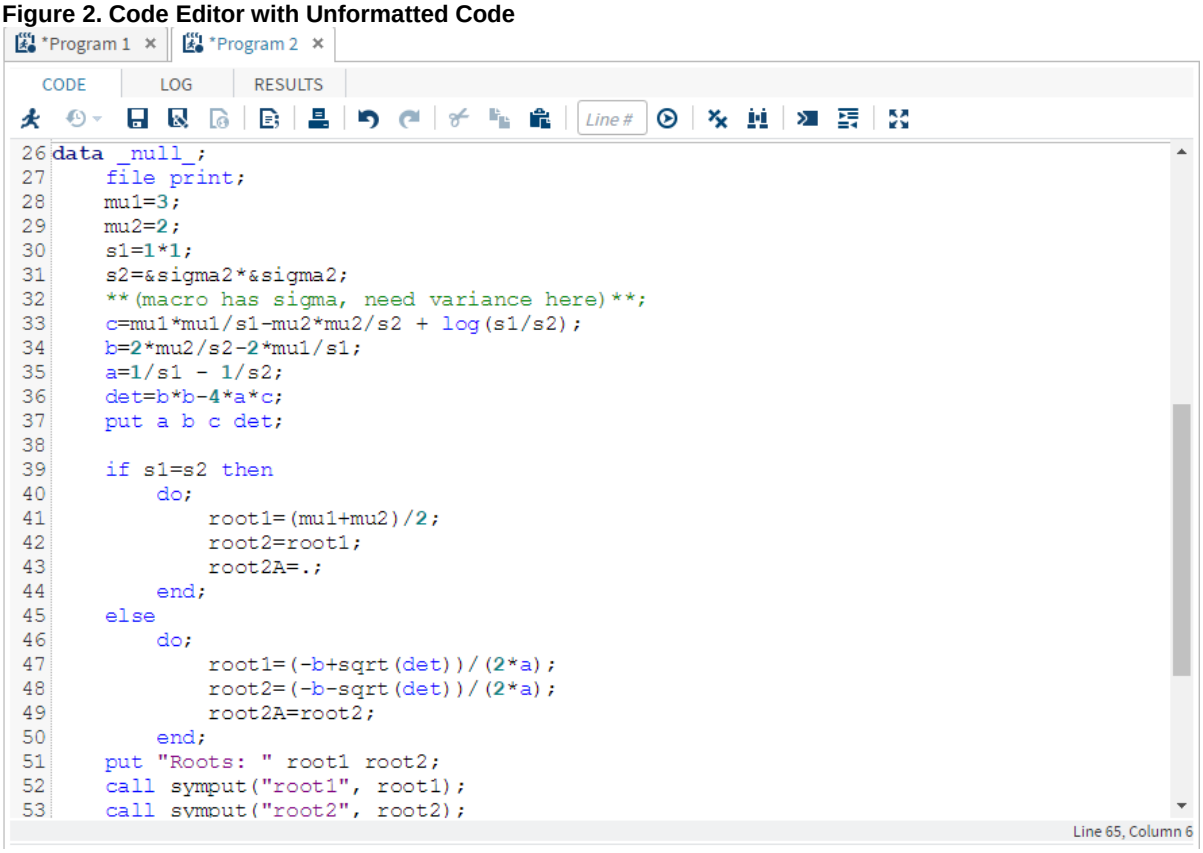

Figure 3. Code Editor with Auto-Formatted Code

Figure 4 shows some code that has been submitted and has returned an error. Before we get to checking the log, I wanted to point out that when writing the code, I can drag variable names from the library panel directly into the code editor, so I don't have to worry about spelling the variables correctly. As you'll see with the error message, I should have done that with the dataset name, too. Note that the red x in the program tab tells me that I have an error. To investigate, I just need to click on the Log tab.

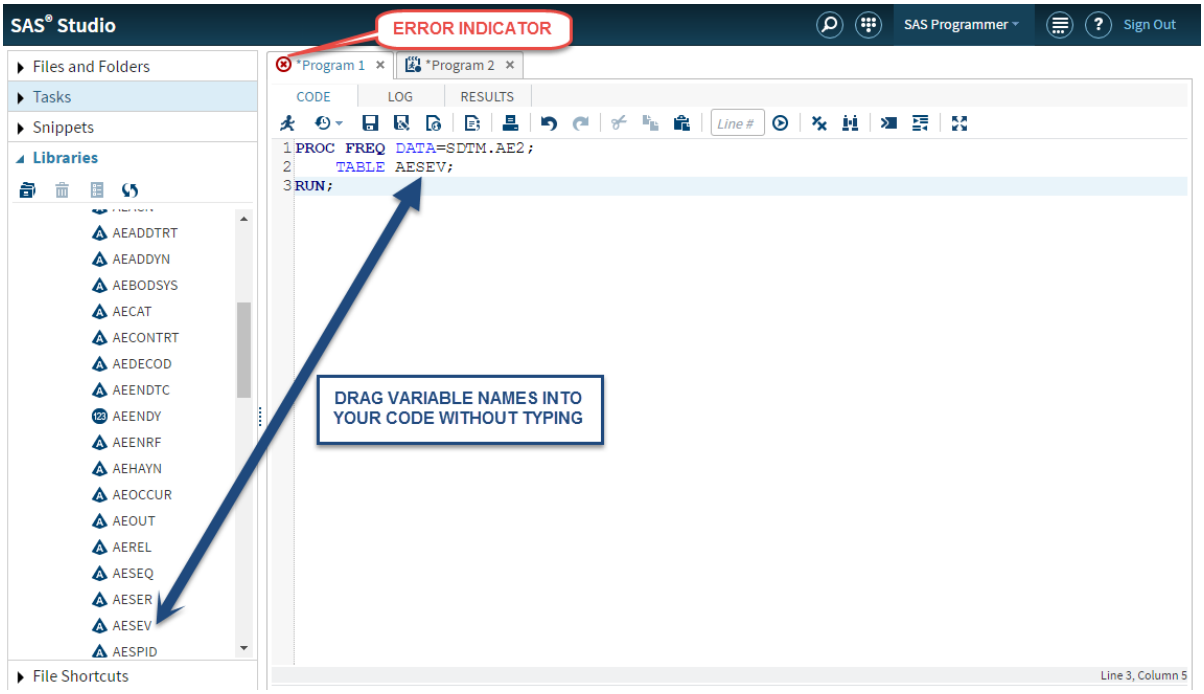

Figure 4. Code Editor with Error

Figure 5 shows what happens when I view the log. The top part of the display, in the grey box, gives me a count of errors, warnings and notes. When I expand the errors section, I see a list of all of the error messages. When I click on the error message in that section, I am linked directly to the part of the log where the error shows up. No more CTRL-F to find the Errors!

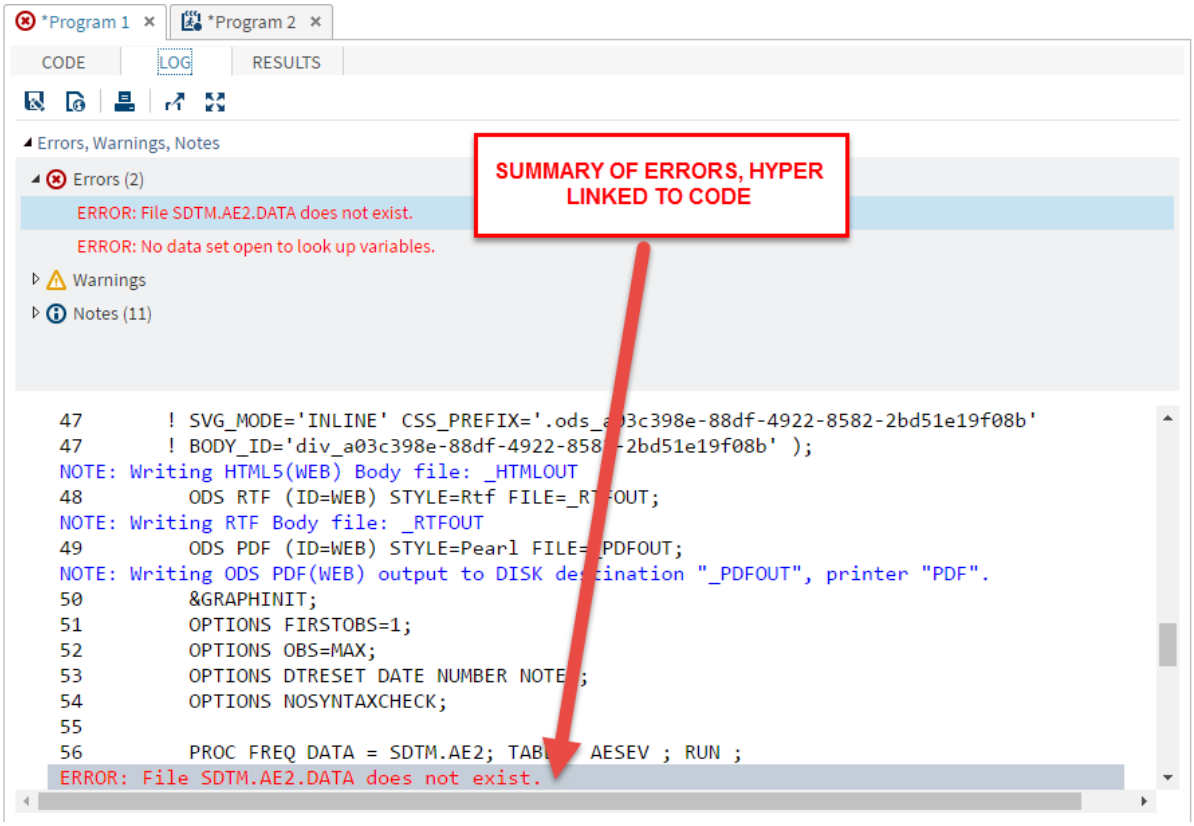

Figure 5. Log Viewer

Another handy feature is the submission history button (Figure 6). Studio keeps a record of every time you submit code during a session, and you can call up the previous sessions. Make some changes to the code and realize you liked it better on the previous submission? Go to the submission history and call up a previous run. It will open in a new program tab as a read-only file and you can cut and paste the changes you want into your current program. Note that this history will clear out when you close the Studio session.

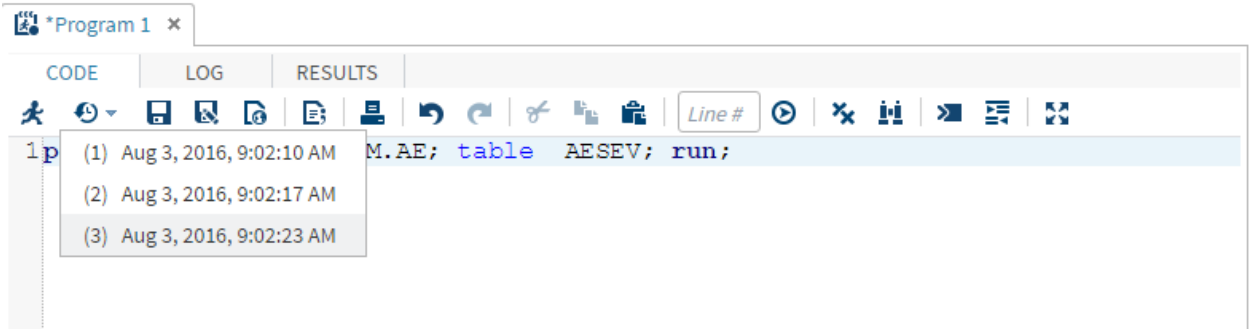

Figure 6. Submission History

CODE SNIPPETS

If you are like me, somewhere (or maybe several places) you have a document with some examples of cool or really useful code you have written and want to be able to go back to and use again. Once you get more than just a few of these code blocks, it can get really hard to manage them. SAS Studio has a built in way to manage these code snippets. It comes with several well documented snippets that cover things like importing and exporting files, creating graphs, writing macros and even IML functions. As Figure 7 shows, all you have to do is drag a snippet over to your programming window, and you add the code to your existing program. In the Import CSV example, all you have to do is put in the file name, change the output dataset, and remove the section about the Unix filename, and you are all set.

In addition to the included snippets, you can create your own collection, so you always have your coding toolset with you no matter where you are accessing the server from. You can also set up shared locations for snippets for study specific ones to share across the team (like the program header, for example).

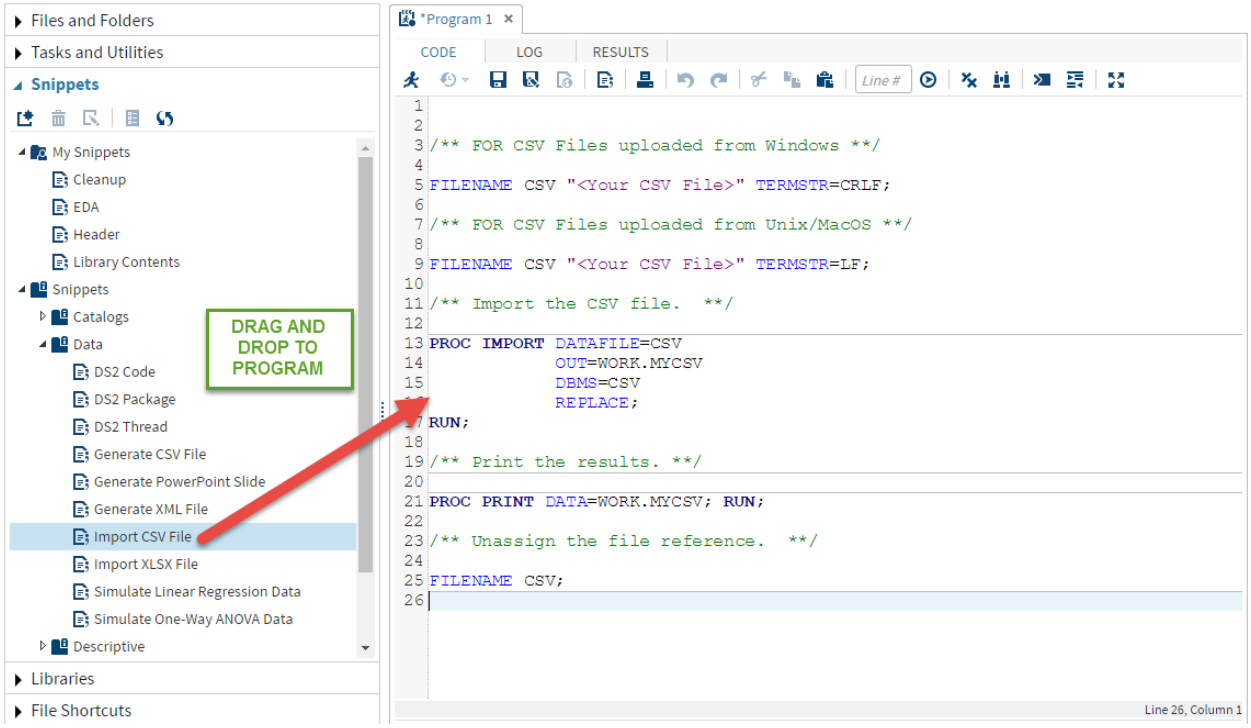

Figure 7. Code Snippets

TASKS

SAS Studio also comes equipped with pre-defined tasks to help you generate code. These are very helpful when it comes to using procedures and methods you are unfamiliar with. The included tasks cover areas such as data summarization, graphing, probability, statistics, high-performance statistics, power and sample size, multivariate analysis, econometrics, forecasting, statistical process control, and data mining. Let's look at a graphing one as an example.

Figure 8 shows the task for building a bar chart. In the settings panel, you select the dataset to use from a drop down, and this allows you to select the different variables. In the Options tab, you can set the headers and footers, and specify data labels. While you are making your selections, code is generated in the code window. When you submit the code, the results are presented as in Figure 9. Note that once you have the graphical output, you can easily save it out in one of several formats.

Tasks are an excellent way to learn new procedures or options. Once you have the output working the way you want, you can just cut and paste the automatically generated code into your own program.

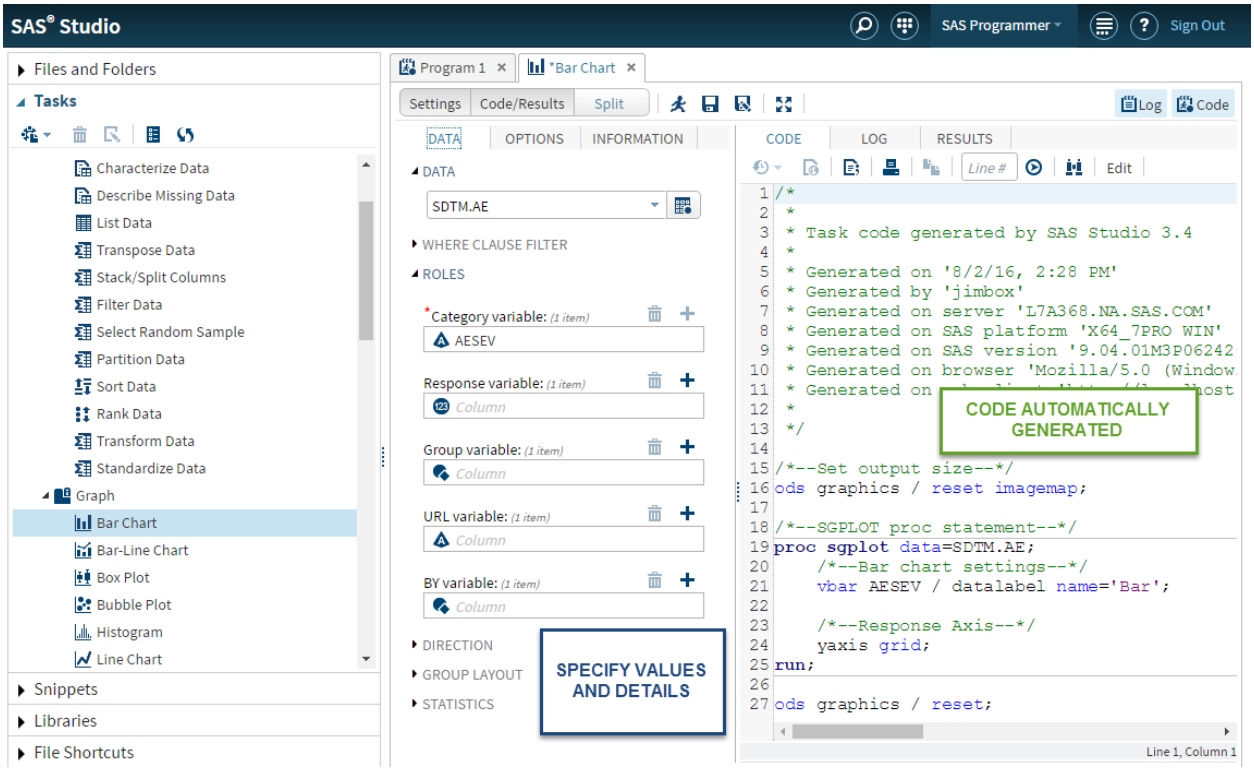

Figure 8. Using Built-In Tasks

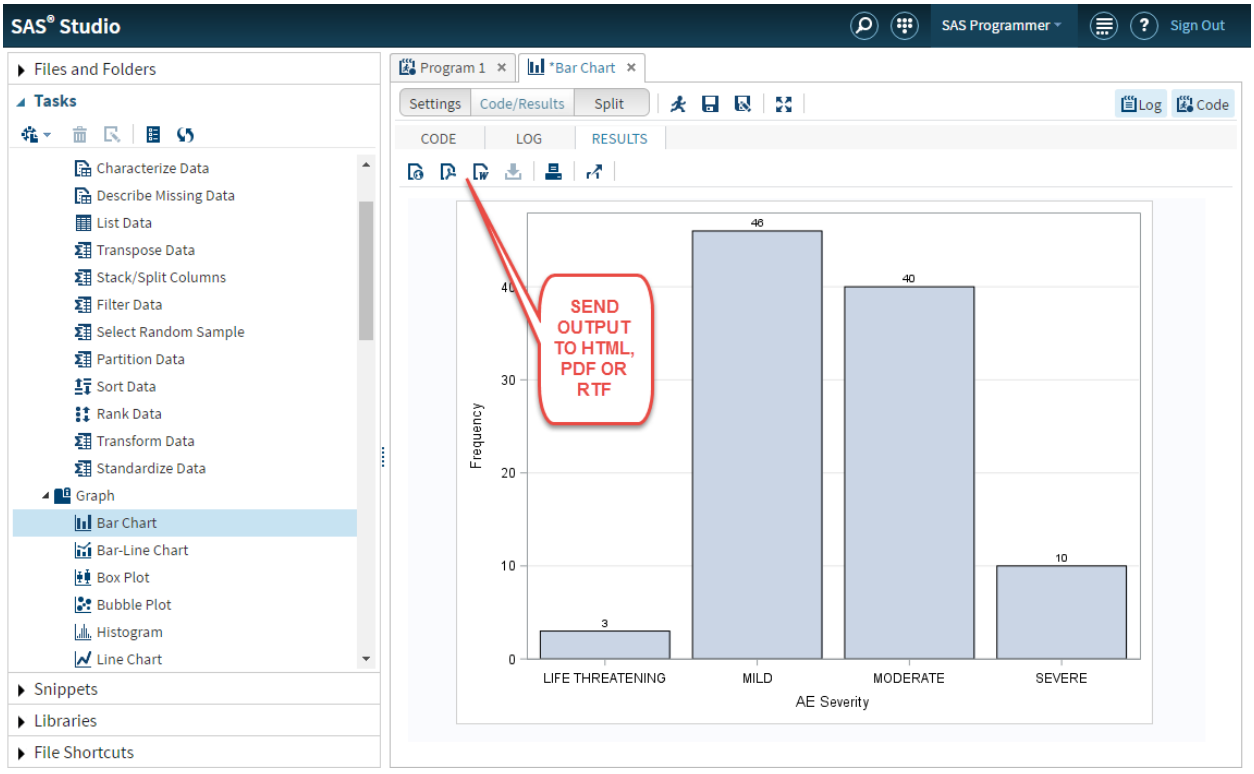

Figure 9. Task Results

You can also create your own tasks to do just about anything. You just have to set up the options panel using code that is very much like XML to capture the inputs you want. You can then treat those input as essentially macro variables to pass into your code for execution. Studio comes equipped with simple and advanced task templates for you to see how to edit the options.

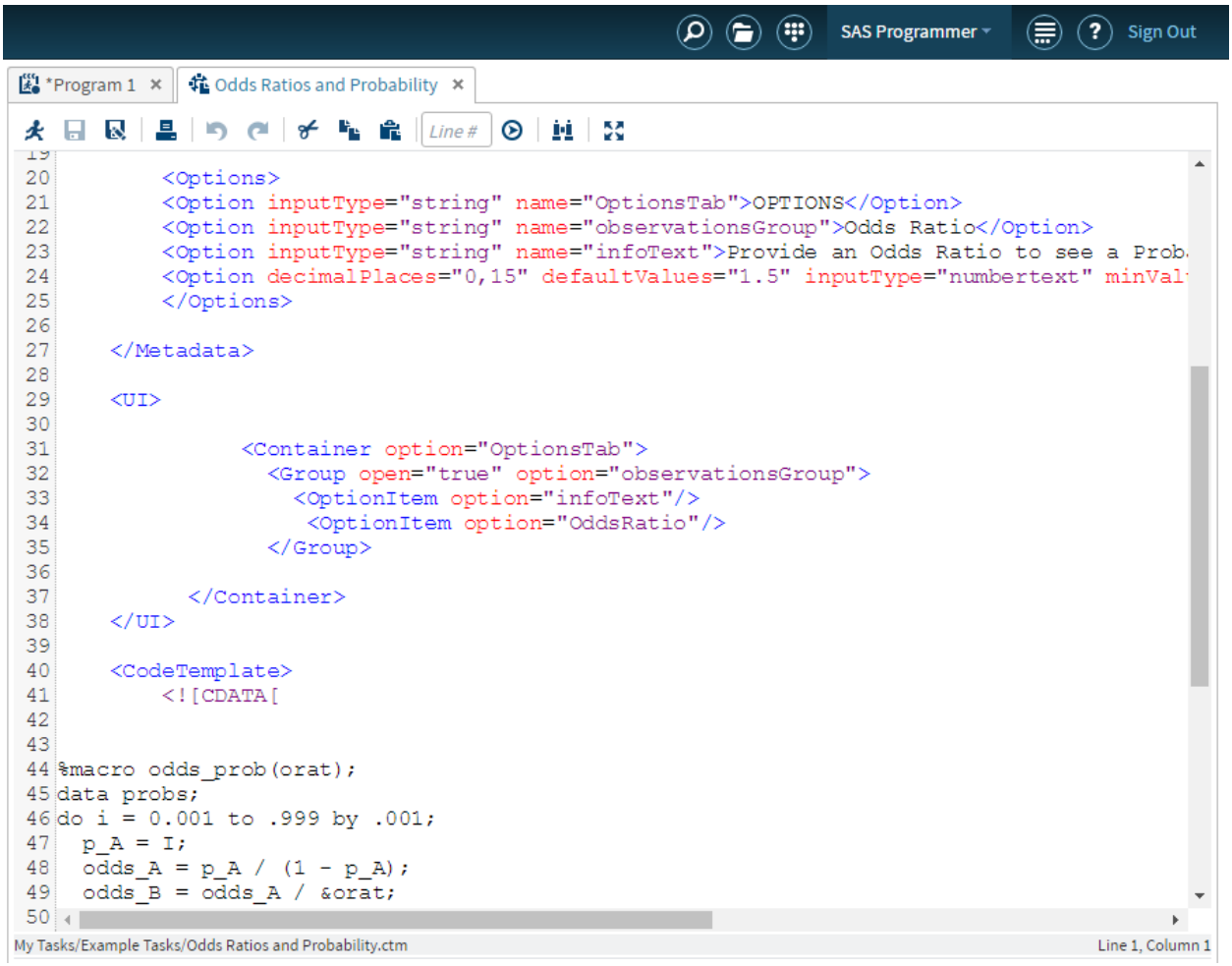

Figure 20. Task Editor

CONCLUSION

SAS Studio is a tool designed specifically to make writing programs easier. Manage your snippets, use tasks to help generate code, and access your SAS environment from any computer. Once you try it out and see how easy it is to write and format your code, you'll wonder how you programmed without it.

RECOMMENDED READING

- SAS® Studio Video Library: http://support.sas.com/training/tutorial/studio/index.html
- Teach Them to Fish—How to Use Tasks in SAS® Studio to Enable Coworkers to Run Your Reports Themselves http://support.sas.com/resources/papers/proceedings15/SAS1831-2015.pdf
- SAS® Studio Knowledge Base: http://support.sas.com/software/products/sasstudio/#s1=2

CONTACT INFORMATION

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

Jim Box
SAS Institute
100 SAS Campus Drive
Cary, NC 27513
Jim.Box@sas.com

Matt Becker
SAS Institute
100 SAS Campus Drive
Cary, NC 27513
Matt.Becker@sas.com