

SAS Toolkit for running cleaner statistical analyses: Using macros to declutter your code

Luke W. Johnston

2014-08-22

Contents

Purpose	2
Installation and usage	2
Version numbering and meaning	2
Documentations	2
HTML	2
PDF	2
Contents	2
doc folder:	3
src folder:	3
Style guide for writing comments and code	3

Purpose

I developed this toolkit to consolidate macros that I use in my research and also to share my efforts with anyone interested, but in particular to share with my fellow labmates and graduate students. As a disclaimer, not all macros are finished and some are very specific and situational, so I encourage anyone to update the files for readability, user-friendliness, flexibility, and parsimony.

Installation and usage

In order to use this repo, run the following shell and git commands (this should run if using Git Bash or other git shell on Windows):

```
cd ~  
git clone git://github.com/lwjohnst86/sasToolkit.git
```

Then, include the following code into the top of your SAS file/script and run it:

```
filename macrolib '~/SAS/src/';  
options maautosource sasautos=(sasautos macrolib);
```

At this time, the macro.sas file will need to be sourced in the SAS file/script, using this command:

```
%inc '~/SAS/src/macros.sas';
```

Version numbering and meaning

Documentations

HTML

Contains the documentations generated by DocItOut. Use the index.html file in the doc/html/ as well as the parent folder to view the documentation.

PDF

Contents

There are two folders in this toolkit.

doc folder:

The doc folder contains the documentation for the macros. So far, I haven't found many documentation generators for SAS, one of which is called DocItOut that is pretty good at generating html pages for looking at the comments and parameters of the custom macros. Open the `index.html` file (in the parent directory) in a web browser (for example, Firefox) to view the documentation.

I will likely be making some type of parser to generate a pdf of the code documentation for those who would prefer viewing a pdf doc.

src folder:

This folder contains all macro files. The format to how the comments are structured are important and need to be consistently adhered to in order for DocItOut to work. Please, if you plan to add to the macros and submit a pull request, be consistent with the way the code is structured (see below).

Style guide for writing comments and code

For writing comments, I based much of what I will be describing below on `DocItOut.pdf` (found in the `doc/` folder). In addition, I also used the website [“How to write comments for the Javadoc tool”](#), which provides a more comprehensive documentation on how to write comments that are similar to Java (SAS has similar syntax to Java). For more reference on things to keep in mind when writing code/documentations, see the files in the `doc/` folder. The file `WritingCodeThatLasts.pdf` by Sandra Minjoe is an excellent resource on practices and techniques to make code have greater longevity. The file `WritingMacros.pdf` (slides) by Patrick Breheny is a great resource to learning how to learning how to write macros. Also, I have a few blog posts on my [website](#) that go through step-by-step on learning to do basic SAS macros, at least to get started.

When writing up SAS macros, always include documentation. To make it quicker to type of formatted documentation, use the table below for using “quick tags” that can be converted later.

Quick tag	Converted html tag
“word (two backticks)	<code>
word“ (two backticks)	</code>
\\	<p>
[word	
word]	
‘&’ (space, then & at the end of line)	

Table 1: Quick tags to use when writing up SAS comment documentation, which can be converted to the html tags by using the `sasQuickTags.sh` in my [git bin repo](#). These html tags are used by DocItOut when creating the documentation.

Code within the SAS macro should always be indented properly and a new line should be started after the SAS semicolon “;” or if the code is longer than 80 characters, whichever comes first.