

Keep it Organized- SAS tips for a research project

Elisa L. Priest
The Cooper Institute, Dallas, Texas
University of North Texas School of Public Health, Fort Worth, Texas

ABSTRACT

Students learn the basics of SAS programming in class or on their own and then embark on a research project with these newly developed skills. At the end of the project, you generally end up with multiple pages of SAS code, multiple .SAS files, and many datasets. I will present examples to help students keep their SAS code and SAS datasets organized from their first research project on.

The primary topics to be covered will include: documentation of SAS code, using comments, program headers, tables of contents, and organization of SAS files. This will allow the student researcher to remember what they did, how they did it, and where the results are. This diligence will be a critical timesaver in the final revisions of a research project or dissertation.

INTRODUCTION

My first research project using SAS was completed in the graduate school computer lab using example code written by my professor. After many days struggling to get my analyses correct, I printed off my results and put them in a folder on my desk. After a couple of months, with a deadline quickly approaching, I began to write up my results. My professor reviewed my rough draft and suggested additional analyses to conduct. I returned to the computer lab to modify my SAS code only to be faced with the following questions: Where did I store the code? What does this code mean? What are all these data sets? And finally...Why are these results different from the ones I printed out before?

WHERE DID I STORE THE CODE?

Store all of your SAS files on your personal computer, on a CD, or a flash drive and always keep a backup copy of your files in a separate location. Do not store your data on public computers! Data sets may contain confidential information or may be deleted by technical support or other persons.

The key to finding your SAS files months after you've last used them is to store them in folders with descriptive and logical names that you will remember. Store all SAS files in a folder called "SAS" within the main project folder. For example, if I have a folder labeled "DISSERTATION", all my SAS files for the dissertation are located at "DISSERTATION\SAS". Another option is to have a folder called "SAS" with subfolders for each project. The first method facilitates backup and archiving of the project, though either method works as long as you are consistent with the organization. Similarly, keep SAS files (.SAS), SAS data sets (.SD2 or .SAS7BDAT), and SAS output (.LST) in separate directories. This avoids clutter and keeps similar files together. Finally, store only the final drafts of the code or data sets in the main directories. All

other drafts should be cleared out at the end of the project and placed in a folder called “TO DELETE”. These files are saved for reference or ‘just in case’.

WHAT DOES THIS CODE MEAN?

A key to refreshing your memory about the purpose of your SAS code is to include documentation while you are writing it and still remember what it means! Use comments to create a main program header at the start of the SAS file to quickly give you an overview of the code. Several authors have provided examples of different header styles (Winn 2004, Levin 2004, Martin 2000). Standard information found in main program headers includes the program name, a description of the purpose of the program, the author of the program, the date written, input and output files, and a listing of modifications made to the code. It does not matter what header you use for your program, but pick one style and stick with it for all of your programs to make them consistent.

In addition, a table of contents at the end of the main program header can describe the major components of the program succinctly. Create section headers with comments that correspond to the table of contents. An example of a main program header with a table of contents and section header is presented below.

```
*****
*****
TITLE: DISSERTATION Blood Analysis
FOLDER: DISSERTATION\SAS\BloodAnalysis09152006.SAS
AUTHOR: Elisa L. Priest
CREATED: Tuesday, November 9 2004
LAST MODIFIED: Thursday, July 06, 2006

INPUTS: ACCESS DATABASE "\DISSERTATION\DISSERTATIONDATABASE.mdb" TABLE:
BLOODTEST
OUTPUTS: DISSERTATION.BLOODANALYSIS09152006
         "DISSERTATION\SAS\OUTPUT\BLOODANALYSIS09152006.LST"
TEMPORARY: WORK.BLOODANALYSIS
MACROS USED: NONE
EXCEPTIONAL CONDITIONS TO PREVENT ERRORS: NONE

*****TABLE OF CONTENTS*****

1. Import Blood Table
2. Recode Blood Table
3. Import Insulin Data
4. Recode Insulin Data
5. Merge and Label

*****
*****1. IMPORT BLOODTABLE*****
*****;
```

Comments should also be used liberally throughout the program to describe the actions of the program. One suggestion is to include a comment before every major DATA or PROC step (Levin 2004). However, if the code is repetitious, limit comments to the first

instance of the code and to code that is tricky or non-standard. Comments can also be useful for documenting the rationale or reasoning behind code. It is very helpful to document that your professor suggested that you analyze the data in a specific manner!

WHAT ARE ALL THESE DATASETS?

The amount of data sets produced for a project may require additional organization beyond the basic 'DATA' folder. If needed, additional folders can be created to categorize specific types of datasets, such as input or analysis datasets. While many permanent datasets may be necessary, using the temporary SAS work library can reduce the number of datasets that require organization. Keep input datasets as permanent datasets as well as the final analysis dataset. Never save over your input datasets. Instead, use the temporary datasets for creating and modifying variables during the SAS session. The final analysis dataset should contain all recoded variables for all analyses, if possible. This will reduce the number of datasets needed to recreate an analysis if necessary.

To easily find and understand the contents of datasets, document the meaning of dataset and variable names. Dataset labels and variable labels to describe the data and the dataset and both can be assigned during a DATA step. All labels should be concise and descriptive. Labels should provide additional information beyond the variable names, such as the unit of measure. Both dataset and variable labels can be viewed using PROC CONTENTS which can then be used as a codebook.

```
Data Dissertation.BloodAnalysis09152006 (Label="Final Analysis Dataset  
09/15/2006");  
set work.BloodAnalysis;
```

```
Label  
BloodMonth0="baseline BLOOD MONTH"  
BloodDay0="baseline BLOOD DAY"  
BloodYear0="baseline BLOOD YEAR"  
Cholesterol0="baseline CHOLESTEROL (mg/dL)"  
HDL0="baseline HDL (mg/dL)"  
LDL0="baseline LDL (mg/dL)"  
VLDL0="baseline VLDL (mg/dL)"  
;  
Run;
```

```
Proc contents data=Dissertation.Analysis09152006;  
Run;
```

WHY ARE THESE RESULTS DIFFERENT FROM THE ONES I PRINTED BEFORE?

One way to avoid changing results is to save all the files required to recreate the data analysis. The important files to save include the input datasets, SAS code used to create the analysis dataset, the analysis dataset, and the output window (.LST) file with the relevant results. With all of this information, there is no need to re-run an analysis unless you discover a mistake in the coding that created the analysis dataset. In addition, if there

are multiple versions of analysis datasets and outputs, save each .SAS file associated with the different versions. If you have documented the inputs and outputs to the code in the header, you should have no problem recreating your analysis with the correct code and datasets or modifying your analysis if your professor requests it!

CONCLUSION

Student researchers are often on their own for their first project. The previous guidelines have successfully kept this student researcher organized and able to reproduce results for a project quickly and without frustration.

Thus, using the previous examples, the files saved for this project would easily be found:

Code: DISSERTATION\SAS\BloodAnalysis09152006.sas

Analysis Dataset: DISSERTATION\SAS\DATA\BloodAnalysis09152006.sas7bdat

Output: DISSERTATION\SAS\OUTPUT\BloodAnalysis09152006.lst

REFERENCES:

Levin, L. (2006). SAS Programming Guidelines. SUGI 31 Proceedings: Paper 123-31.

Martin, C and Martin, L. (2000). Clean-up, Comments and Code- Making it Maintainable. NESUG Proceedings.

Rhodes, D. (2004). Programming Standards, Style Sheets, and Peer Reviews: A Practical Guide. SUGI 29 Proceedings: Paper 135-29.

Winn, T. (2004) Guidelines for coding of SAS Programs. SUGI 29 Proceedings: Paper 258-29.

CONTACT INFORMATION

Elisa L. Priest
Director, Clinical Trials Data Management
The Cooper Institute
12330 Preston Road
Dallas, TX 75230
972-341-3254
elisapriest@hotmail.com