

Paper 145-30

Everybody Loves SAS®! They Just Don't Know It Yet.

Denise A Figliozzi, SunTrust Bank, Atlanta, Georgia
David Steves, SunTrust Bank, Atlanta, Georgia

ABSTRACT

Believe it or not, there are people in this world that resist and avoid learning SAS unless absolutely necessary. (These are the non-technical people of the world and they like it that way.) I have had the pleasure of introducing SAS to some of the non-technical business analysts that work in my group. SAS can be cumbersome and intimidating to a user that has adapted to a point and click query software. Often these users have never programmed in any language before but now need to know BASE SAS to access the data for their analysis.

This paper discusses training non-technical people how to use SAS and will suggest BASE SAS procedures (aka. procs) and handy SAS tips that an analyst should learn in order to get their job done. Procs and navigation (through SAS software) that are almost intuitive to a technical person need to be explained and introduced. At times, advanced programmers think they are explaining the basics. In reality, they have not gone back to the beginning concepts. This paper will help you to see SAS through the eyes of a non-technical first-time user and will in turn make you a more effective trainer.

INTRODUCTION

A new analyst joins the group and needs to access data remotely via SAS. There is no time for the course of "Programming I: Essentials" given by the SAS Institute. The course takes three days and the analysis is due in two days. Consequently, the analyst recruits you, the experienced programmer, for help.

I have provided 8 survival procs/tips that a non-technical user needs to learn about SAS in order to efficiently and effectively retrieve data from SAS tables. Additionally, there are 4 procs/tips for more intricate analysis and processing.

CUSTOMIZING TRAINING

First and foremost, verify that the analyst's connectivity is working. Connectivity is the ability for that particular user and/or computer to have access to a server that houses the data to be accessed. Explain why they need to run a signon script to do a remote submit and what exactly connects once the script has executed. Once they are connected, it is time to figure out what they need to know.

A good starting point is to talk with the user to see what kind of software they are comfortable with or prefer to use on a regular basis. It may be Microsoft Excel®, Microsoft Access®, or some other GUI (graphical user interface) type software. This can tell you a lot about what the user would prefer to use in SAS. For instance, if the person creates queries by using the GUI part of Microsoft Access®, they may have explored the SQL code behind those table queries. This would tell you that PROC SQL would make that particular user relate to SAS on some level. The idea is to draw similarities between a SAS environment and what the person already knows; find their comfort zone. Quite often, users don't even realize how capable they already are with SAS. Most of them do intricate things with Microsoft Excel® that I have never seen before. Remind the user of the learning process involved when learning all of the Microsoft Excel functionality. They have mastered that software and they too can master SAS concepts.

Another important piece of information is to find out what they need to do with the data once it is retrieved. If they will be summarizing or doing calculations, PROC FREQ saved to a dataset is a great method for someone to get counts. PROC SQL can be a quick and dirty way to calculate.

MAKING SAS TANGIBLE

"Where is the data?" and "How do I get to it?" are the most important basic questions a user needs to have answered. In our shop, we store our data on a UNIX platform. It is necessary to explain to the user that the data is stored in different "directories" and the way to point to the correct data location or "directory" is through a libname statement. This is also a good time to explain about viewing the data that is created via the data step. A common mistake that is made when learning SAS is to view the data through a PROC DOWNLOAD, then re-run the dataset with different criteria. A new user does not intuitively know that if the PROC DOWNLOAD is not re-run, they will be viewing old data. This can be frustrating.

See DIAGRAM A for a simple yet enlightening sketch of this basic concept.

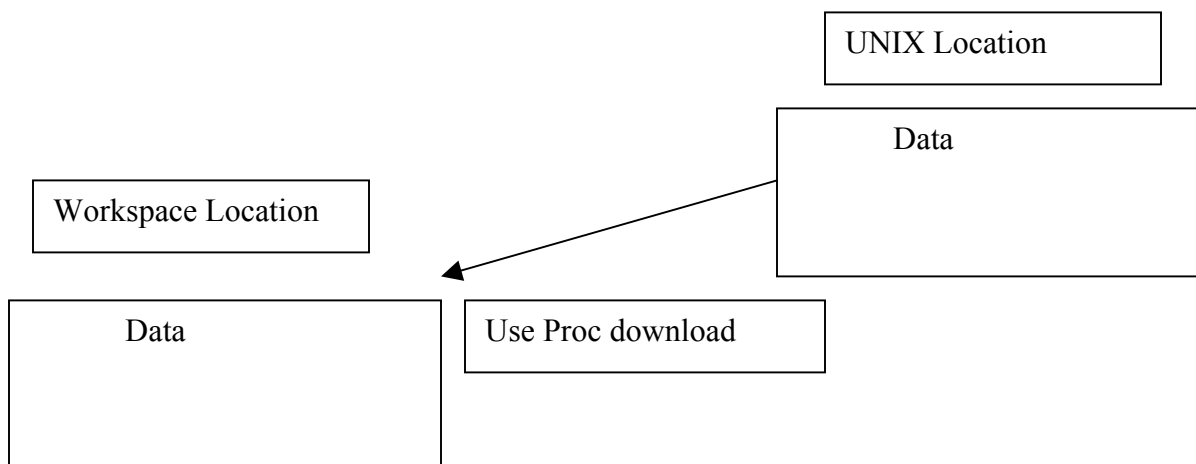


DIAGRAM A

Another important tool for the new user is the contents procedure. This starts to help them bridge the gap between the non-technical and technical lingo. Something to recognize is that when a PROC CONTENTS executes, the SAS system automatically activates the output window. A user doesn't have to search to find the results from the code that has been run. Conversely, after the download procedure has executed, they need to know how to access the results. A familiar way to bring the user to the data is via the Explorer SAS column on the left hand side. Most computer users are familiar with a Windows Explorer® format, so the SAS Explorer will be somewhat intuitive for them.

Explaining data locations with a quick sketch and explaining where the user can find data results are two important steps in making access to the data tangible for the user.

MANIPULATING RAW DATA

A very useful procedure for analysts is the PROC FREQ. Analysts tend to view value distributions as one of the first steps in researching customer behavior. Normally, the PROC FREQ automatically populates via the "output" window. The analyst may get more use out of the PROC FREQ if the results are stored in a dataset. This is achieved by adding a few commands on the "table" line.

```

Ex.   PROC FREQ data=lib.testdata;
       Table var1 / out=mylib.permdata;
       Run;
  
```

** This example saves the frequency counts from lib.testdata to a permanent dataset called permdata.

The data step is certainly an integral part of understanding how to retrieve data and perform data manipulations. A novice is not going to be thinking about efficiency of coding, but a novice will definitely feel the lack of efficiency if he codes something and it takes hours instead of minutes to run. Therefore, it is up to the trainer to introduce the use of a "where" clause vs. an "if" clause and the impact it has on the run time of a query. Keeping with the efficiency trend, do not be afraid to introduce the trainee to the "obs = " syntax. I usually start the trainee off with using it in a data step and then evaluate if confusion will ensue if I mention the "options" line. Remember, the analyst only wants to know what is essential at first. Throwing too many choices into the mix may only complicate things in their head.

The easiest way to inflict a panic attack on an analyst is to show him a SAS date. Time periods are essential for analysis and dates are used frequently. This is a situation where the data step may seem complicated by the introduction of the "format" statement, but once the analyst sees that "date number" transform into "mm/dd/yyyy" format, they won't mind. Once they become more advanced, show them how to format the entire date column in a PROC DOWNLOAD for viewing purposes. Easing their date anxiety by showing them '15apr2005'd avoids many headaches.

I find that when looking at more than 20 lines of output, it is best to view through a PROC DOWNLOAD. The most useful feature of the PROC DOWNLOAD view is the query box in the left hand corner. The analyst has the ability to focus on the pertinent records of the output and then quickly change the requirements. This is done by typing "where <var> = '<value>'" in the query box. However, if the analyst insists that it must be viewed in Microsoft Excel®, it is time to introduce the "Export Wizard" functionality. This way, there is no more coding to be learned. Just a series of clicks and the code is automatically generated behind the scenes. The analyst will probably prefer to export the data until they become more comfortable using SAS.

One valuable tool that my trainees appreciate is the skeleton data steps and procs for future use. It's a template of syntax that is easily accessed. My template looks something like this:

```
rsubmit;
libname gs sasspds "g32005" host="99.99.999.99" serv="5150"
user="everyone" password="pass11";
libname gsd sasspds "gsdata" host="99.99.999.99"
serv="5150" user="everyone" password="pass11";
libname ext sasspds
"extracts"host="99.99.999.99"serv="5150"user="theboss"passw
ord="pass22";
libname dee '/temp/';
libname tran '/codedir';
endrsubmit;
```

```
*** Use this to view the variables that are present in the
dataset;
rsubmit;
proc contents data=          ;
run;
endrsubmit;
```

```
*** Use this MODEL to structure your query;
rsubmit;
data <the new dataset>;
set gs.<source data> (obs=10);
where <variable> = '<value>';
run;
endrsubmit;
```

```
*** Use this to view the data like a spreadsheet;
rsubmit;
proc download data=          ;
run;
endrsubmit;
```

```
*** Use this to view the data with a few records;
rsubmit;
proc print data=            ;
run;
endrsubmit;
```

```
*** Use this MODEL to run a frequency and save it to a
table.
```

```
rsubmit;
proc freq data=lib.testdata;
table <variable> / out=mylib.permdata;
run;
endrsubmit;
```

```
*** Use this MODEL to query using SAS dates;
rsubmit;
data <the new dataset>;
set <source data> (obs=10);
where <date var> = '15apr2005'd;
run;
endrsubmit;
```

FINE TUNING THE PROCESSING

Once the analyst is comfortable with SAS, it is time to show him some tricks. These tips are definitely dependant upon the specific analytical needs of the user. If the analyst is attempting to produce reports, it may be time for an introduction on "if first / if last" processing. Most reports require some sort of summarization function. As SAS programmers, we are partial to use PROC MEANS but remember the background of the users. If they have programmed with Microsoft Access®, they may be more comfortable with the PROC SQL method.

Much to my own surprise, one of my novice users has a need for PROC UNIVARIATE. Although this procedure's results are very detailed, the analyst needed to find the median calculation of many variables. PROC UNIVARIATE was the easiest way to obtain this data.

CONCLUSION

The most crucial point that I want to emphasize in this paper is the importance of letting the user's background dictate the direction of the training. It is not about showing them 10 different ways to do one task. It is about the necessity of data retrieval via SAS.

As an avid user and promoter of SAS, it saddens me to see my peers get so frustrated with their work when SAS is so useful and efficient. I have made myself readily available to the novice user, just so that they do not become overwhelmed. By attempting to keep things simple and making the user feel comfortable, it's just a matter of time before the sounds of "I love SAS!" are beaming from another office.

The analyst's "survival guide" includes:

- 1) Explanation of libnames
- 2) Explanation of data locations
- 3) PROC PRINT / PROC DOWNLOAD
- 4) Obs=
- 5) "Where" statement
- 6) PROC FREQ (saving to a dataset)
- 7) Export Wizard
- 8) Date Queries using SAS dates

For the more advanced analyst:

- 9) PROC SQL
- 10) PROC UNIVARIATE
- 11) "If First" processing
- 12) PROC MEANS / PROC SUMMARY

CONTACT INFORMATION

Denise A. Figliozi
SunTrust Bank
303 Peachtree Ctr. N.E.
Atlanta, GA 30303
(V) 404.230.5547
(F) 404.658.4083
denise.figliozi@suntrust.com

David Steves
SunTrust Bank
303 Peachtree Ctr. N.E.
Atlanta, GA 30303
(V) 404.827.6581
(F) 404.658.4083
david.steves@suntrust.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 registered trademarks or trademarks of their respective companies.