

# Connect

Information Technology at NYU



Search This Site

Spring 2003 Edition

Read the article below, or select an option from this menu:



Current Issue



Archives



About Connect

Social Science, Statistics & Mapping

## SAS for SPSS Programmers

By [Frank LoPresti](#)

SAS, like SPSS, is a popular statistical software package. These two statistical packages are the top ranked among the many used at NYU. Both packages allow researchers to handle and investigate data in a user-friendly fashion. For example, U.S. Census data, which can easily be downloaded from the Internet, comes with programs that prepare the data for use within SAS or SPSS.

After teaching SPSS and SAS introductory tutorials for many years, I have several observations that I feel may help first-time SAS users. I have watched some very advanced SPSS users hit a wall when they moved to SAS, and I think I understand why. (If you already know all about SAS, you can go straight to the heart of this article in the bottom section: "So Why Is SAS So Hard?")

This article is a simple overview of SAS, with insights intended to assist new or "challenged" SAS users. The sample commands included in this article would be written and submitted within a SAS windows session. To actually run these SAS commands, read the article "SAS for Windows," by Bob Yaffee, available on the ITS Social Science, Statistics and Mapping Group's website at [http://www.nyu.edu/its/socsci/Docs/sas\\_windows.html](http://www.nyu.edu/its/socsci/Docs/sas_windows.html), or "How to Import Data Files into SAS," by Marc Grayson, at <http://www.nyu.edu/its/socsci/Docs/Import2SAS.pdf>.

### SAS Commands

There are two different types of SAS commands: DATA steps and PROCs. Think of them as small programs. DATA steps and PROCs are very different and very separate from each other. They do not need to be used during the same SAS session, though they frequently are. DATA steps, which allow us to handle data, are the most difficult part of using SAS. On the other hand, PROC commands simply request a statistic or a report, and usually do not affect the data.

Since PROCs are so easy, let's get them out of the way first. Let's assume that you have a set of data formatted for use within SAS. Perhaps your instructor has given you a file and told you that it is a SAS dataset. Your homework assignment is to get frequencies and means for all variables in that dataset where it is appropriate. Easy as pie, right?

Say the data on your floppy is in a SAS file called "census" in the directory sas\_files. You would simply submit the following lines to get a frequency table of the variable "sex":

```
LIBNAME      play 'a:\sas_files;
PROC FREQ    data = play.census; tables sex;
```

Or to get the mean of the variable "age":

```
PROC MEANS data = play.census; VAR age;
```

The LIBNAME statement gives the SAS program the path to your data.

Other simple useful PROCs are:

```
PROC PRINT data = play.census;
or
PROC CONTENTS data = play.census;
```

The first command prints out your data, while the second reports the contents (variable names, count of cases) of your data file.

Given that we started with a SAS dataset, these PROCs, and other very advanced statistical PROCs such as PROC TIMESERIES, are short, simple programs. So long as you understand the statistics you are using, the most advanced procedures will run in a straightforward manner on your SAS dataset.

The difficult part of using SAS involves the manipulation of data. Unless you are given a dataset formatted for either SPSS or SAS, you must get your data into a dataset. Conceptually, SAS and SPSS bring data into their respective datasets using similar commands (see fig. #1). In SPSS, we could write a "DATA LIST" statement, while in SAS we could use an "INPUT" statement. Both of these statements are used to make raw data into datasets. They tell the programs the names of our variables and let the programs sort out which numbers in the raw data (Bucket o' Data) are associated with each variable. See the Little SAS Book (Delwiche and Slaughter), or any other SAS introduction for information on the syntax of the "INPUT" statement. See the SPSS 11 Syntax Reference Guide (SPSS, Inc.), or any other version manual, for details of the "DATA LIST" statement.

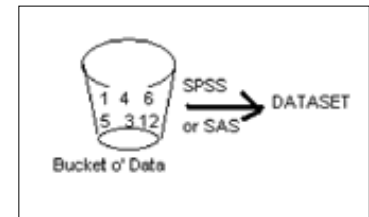


Figure 1. Creating a Dataset from Raw Data Using SAS or SPSS.

### So Why Is SAS So Hard?

After teaching these two programs for years, I had an epiphany. I realized that the order of the DATA step leads to confusion--confusion which is often never cleared up. Following is a typical SAS DATA step (program) that reads a file of raw data (see fig. 1) and creates a SAS dataset. Explaining the exact syntax of each line is not important. However, the flow of the program is important, and demonstrates what I think is the fatal flaw of SAS:

```
DATA play.test1;
INFILE 'c:\frank_sas\my_data.txt';
INPUT id x y ;
```

This program actually starts on the second line, not the first. The second line, the INFILE statement, identifies the file to be read in. In this example, the file is named "my\_data.txt", and contains the raw data shown in figure 2. The third line of this program is an input statement, which tells the program to read the first line and give id the value 1, x the value 23 and y the value 45.

```
1 23 45
2 44 56
5 46 75
```

Figure 2. Raw data file "my\_data.txt" read into the above program.

Then the program starts again, reading the second line and the third line, then reading in another line from the raw data file: 2 44 56 . The program then goes to the top and writes another line 2 44 56 into the formatted SAS dataset. Finally, the program will repeat once more, reading in and writing the last line of the raw data file (see fig. #3).

When the program completes, we have our original raw data file, "my\_data.txt", and a new SAS-formatted dataset, "test1", with the file extension chosen by SAS. If you used version 8 or 9 of SAS, the file would be called test1.sas7bdat.



Figure 3.

Thus--and here's the key--the above lines:

```
DATA          play.test1;
INFILE        'c:\frank_sas\my_data.txt';
INPUT         id x y ;
```

Should really be:

```
INFILE        'c:\frank_sas\my_data.txt'; (READ)
INPUT         id x y ;
DATA          play.test1; (WRITE)
```

Trivial observation? I don't think so! I have known some smart old SPSS programmers (myself included) who never really felt comfortable with SAS DATA steps. Although I have never seen a SAS publication observe that the DATA line (the first line of the DATA step) should be thought of as the last line of the program, I strongly suspect that this line order confusion is the root of the problem.

---

### Author Biography

*Frank LoPresti heads the Social Science, Statistics, and Mapping Group of Academic Computing Services at ITS. He can be reached at [frank.lopresti@nyu.edu](mailto:frank.lopresti@nyu.edu).*