

## GSPP / HSPA Crash Course in SAS

Rob Letzler                      Course Materials will be posted at: [www.ocf.berkeley.edu/~rjl/stats/](http://www.ocf.berkeley.edu/~rjl/stats/)  
Helen Oliver  
Brian Quinn

Getting SAS: Berkeley students can get SAS for \$30 if they do some paperwork. The copy of SAS will run until September 2004. The instructions are at: <http://softdist.berkeley.edu/>

### Learning SAS:

We recommend *The Little SAS Book* by Lora Delwiche and Susan Slaughter. It describes what SAS can do, suggests the right questions to ask, and gives most of the right answers. The book is organized in two page segments so you can the topics that you need. (New versions of SAS largely automate importing data, so many of the skills in Chapter 2 are no longer necessary. The material in chapters 7 and 8 is simpler and more relevant material than chapter 6.)

When you have particular questions, try the SAS online reference at: <http://v9doc.sas.com>. The help system may ask you which parts of SAS to search for help on. It is wise to start by searching all of the documentation because some of the statistical tools you may use (and might expect to be in Base SAS) are in places like SAS / Stat or SAS / ETS (ETS has time series tools) which are part of the system that you get as a Berkeley student.

Ask for help – there are plenty of experienced SAS programmers around to help with problems that at first seem inexplicable. For example, former research assistants from places like the Urban Institute and Mathematica Policy Research are likely to be fluent in SAS, as are many econ grad students and professors. Most of us got help when we were first learning and are willing to return the favor.

SAS is built around a core language from the 1970's. Different SAS revisions came up with different solutions to similar problems, so there are fewer rules and more quirks and exceptions to rules than one would like. (For example, as a general rule Data steps change data and proc steps analyze and display data. But Proc Sort changes data and displays nothing.) Expect to never outgrow the need to consult documentation to look up exact syntax details.

### Basic ideas:

SAS saves and loads SAS data sets in “libraries”, which are nicknames that you give to locations on your disk. In your SAS program, you use the command libname to name a library:

#### For example:

```
libname analysis 'D:\docs\spring2004\apa\data\';
```

Tells SAS to read and write all data sets with names that start with `analysis.`, like `analysis.gasData` (notice that a period separates the library name from the data set name), to the directory `D:\docs\spring2004\apa\data\`. SAS can also work with temporary data sets. They have no libname and are erased when your SAS program finishes executing.

## Getting Data into SAS.

The first step in using SAS is to convert your data into a SAS data set. The SAS Import Data wizard, which is on the File menu makes this very easy. You will need to create a library, by double clicking on the “libraries” icon in the upper left hand corner of the screen and then right clicking the background of the “active libraries” explorer window and choosing “new”. (These instructions are based on SAS 8)

You will deal with 3 kinds of files when you work with SAS:

-Program files, have .SAS extensions. They contain batches of commands that tell SAS how to find, manipulate, and analyze your data. When you run the SAS program `programName.SAS`, SAS will create two files: `programName.LOG` and `programName.LST`. The LOG file tracks information like whether SAS encountered any errors running the program (and how it handled them) or generated any warnings about possible problems and the number of observations in each data set.

Databases – observations and variables.

Everything in SAS is a database, which consists of the same variables recorded for many observations. An example SAS dataset looks like this:

Observation #	Variable: Name	Variable: age	Variable: earnings	Variable: education
1	Dexter	40	1,000,000.45	21
2	Jeff	44	90,000.00	20

## **SAS distinguishes between changing observations using “Data Steps” and running procedures on the whole data base**

SAS makes a strong distinction between “Data Steps” and “Proc Steps”. Data steps are blocks of code containing instructions about where to find data, how to modify each observation, and where to write the resulting observations. “Proc steps” are procedures that analyze, display, or modify all of the data you send them as a unit.

For example, a basic SAS program that constructs a new variable on each observation and then prints all of the results would look like this:

```
libname analysis 'D:\docs\spring2004\apa\data\';

data results; /*this directs output to "results"*/
    /*the data statement loops until it runs out of data*/
    set analysis.input_data; /*set reads data in*/
    total_fruit = apples + bananas;

proc print data = results;
```