

# 1 An overview of SAS system and how to run it.

## What is SAS?

SAS (Statistical analysis system) is many things. Among other things, it can be used for data management and report generation, as a statistical package including descriptive and inferential statistics (for a wide variety of problems) and as a general programming environment, including a portion called IML which allows for easy handling of matrices. The complete package is extremely comprehensive, and continues to expand at a rapid rate. This course provides an introduction to SAS with the primary focus on statistical computing (as opposed to data management and report generation) including how to program to handle customized analyses.

Important links through OIT (available through course website):

- Info on OIT Classrooms (which have SAS): <http://www.oit.umass.edu/classrooms/index.html>
- Info on SAS site licenses if you want to buy your own copy.: <http://www.oit.umass.edu/software/site/index.html>
- Basic instructions for using SAS <http://www-unix.oit.umass.edu/statdata/software/index.html>
- Online documentation for SAS 9.1.3: <http://support.sas.com/onlinedoc/913/docMainpage.jsp>

## Where do I run SAS?

- On a PC with Windows: You can run SAS in any of the OIT computer classrooms or you can buy a copy (renewable each year) for your own machine.
- The PC's in the Math-Stat rooms for graduate students are running under linux rather than windows, so do not support SAS. You can run SAS from these machines though in a way that looks much like if you were running it on the PC (it is actually running on the department's unix machine).
- Using Unix.

For graduate students in Math-Stat, you can run SAS interactively from the workstations in 1537. Or you can sign into the dept. machine remotely and use SAS but you need X-windows emulation. For anyone with an OIT account, you can also run SAS on the OIT unix machines interactively but again you need X-windows emulation. On any unix machine, you can also run SAS in batch mode; see later discussion.

## Where do I read about things?

There are a large number of manuals and updates associated with SAS. It is best to use the online documentation. More about the online documentation in class. Many books have been written on using SAS in various ways. A list of books is available at

<http://www.sas.com/apps/pubscat/booklist.jsp?attr=category&val=SAS+PRESS>

Here is a link to a page at UCLA that looks helpful.

<http://www.ats.ucla.edu/stat/sas/>

## 1.1 How do I run SAS?

**On a PC**, SAS is run interactively in the windows environment. When SAS is invoked there are various sub-windows, the three main ones being the program window, the output window and the log window. When you are actually using programming statements (as opposed to click and go as in Analyst, etc.) the programming statements are put in the program window. The program is then submitted (by clicking on

the running figure), with output going to the output window and the log window containing information (including error messages) about the job. To run only portions of the program in the program window, you highlight that portion and then submit. You can open and save files from the various windows. Various other aspects of running in this environment will be demonstrated in class and are described in the on-line help.

Running SAS **on unix** is a bit different. If you want to run SAS interactively with the various sub-windows then this requires an X-terminal emulator on a computer with a direct connection to the machine where SAS is running. It can be easier to run SAS on a unix machine in **batch mode**, without an Xterminal emulator (but this is tedious when you are first writing programs that could need a lot of debugging). Some editor (e.g., emacs or VI) is used to create control files, examine and edit output files, etc. To do this first, the control file is created using the editor and saved as *xxx.sas* where *xxx* is a name of your choosing. To submit the control file in batch mode one simply types *sas xxx* or *sas xxx &* where the version with the & runs the job in background. When the job is done you get a return to the system prompt (or a done message if you submitted it to background). Two files will have been created *xxx.log* which is the log file and *xxx.lst* which is the output file (but *xxx.lst* will be empty if there were errors that prevented the execution). You can now read and edit these as usual using an editor. It is always a good idea to look over the .log file first as one can often get output in the .lst file even though some errors might have occurred.

## 1.2 The general structure of running SAS using programming statements versus “click and go”.

### Using programming statements.

The **data step** creates a SAS data file. The data step might involve reading data from a file in a particular way or consist of programming statements which generate a data file (or a combination of the two). SAS data sets can also be created by importing data.

Once a SAS data set is in place, there is a large number of pre-programmed procedures (procs) that can be run on that data. These are invoked with a **proc** statement. For the most part, each proc has a fixed set of rules and options that must be followed. There is one major exception to this: proc IML is really not a procedure in the sense that the other are; rather it activates a rich programming environment. IML will be covered in ST597D.

### Running SAS without “programming”.

There are various options now available in SAS for running SAS without programming, but instead using drop down menus and “click and go”. We will cover certain aspects of this in the PC environment primarily through the use of the Analyst within the Analysis menu. The analyst is easy to use (although if you are proficient with SAS language, it is sometimes faster and easier to program directly), creates the accompanying SAS code, produces graphs and has some statistical tools not available in other parts of SAS (such as power and sample size calculations). At the same time, it does not have all of the options available through the procs.