

Introduction to SAS

Statistics 514 Fall 2007

Statistical analyses, in practice, are carried out by computer software. In this course, we will primarily use SAS/STAT, an integral component of SAS, to perform statistical analysis. SAS is a statistical software package that allows the user to manipulate and analyze data in many different ways. Because of its capabilities, it is used in many disciplines, including medical sciences, biological sciences, and social sciences. This document focuses on using SAS installed on personal computers, specifically SAS for Windows (PC-SAS Version 8.02). You can find more introductory material at the URL

<http://www.stat.purdue.edu/scs/help/SASshortcourse.pdf>

In addition, the book *Applied Statistics and the SAS Programming Language, Fourth Edition* by Ronald P. Cody and Jeffrey K. Smith (published by Prentice-Hall) is a very useful reference, which is often used as a supplemental text for the STAT 512 class.

Getting Started

1. Version 8 of SAS is available on the PC's in ITaP labs. You must have a Purdue University Computing Center **career account** in order to use ITaP facilities. If you do not have a career account, take your Purdue ID to any ITaP PC lab. You will keep this account as long as you are at Purdue.
2. Purdue's SAS license allows any student or staff member to get a copy of PC-SAS. If your department has PC's or if you have a PC, you can check out the SAS CD's and install SAS on your machine. South Campus Courts Building C has CD's available free of charge to students, faculty and staff to install on your home computer or laptop. (Remember to take your student ID to sign out CD's over night.)

Installation hints. SAS complete occupies more than 500 megabytes of disk space. The parts of SAS used in STAT 514 occupy just over 300 megabytes of hard disk space. To keep SAS to "only" 300 meg: (1) Ignore the CD's titled "Online Doc" and "Client-side Components" in the installation package. (2) Say "No" when asked if you want to install help in "Simple HTML." (3) Choose Custom Installation and in the window that eventually appears, check only these components: Base SAS, Core of the SAS System, SAS/Graph, SAS/QC, and SAS/Stat.

Using ITaP Labs

You will want to find a lab that has space available at times you want to work. Schedules for individual labs are usually posted on the door. You can find complete lab schedules at the ITaP information Web site. Most labs are scheduled for classes during the day.

Logging in. The screen should show a box with the message "Please login to use this machine." Type your career account ID in the **Login** field. Use the tab key or the mouse (point and click once) to move the **Password** field. Type your career account password (it will not show on the screen) and hit the enter key. Once logged in, you can access the course materials from the Stat 514 web page and save material on your home directory (H: drive) or floppy disk.

Logging out. After saving your work either on disk or on the H: drive, log out from the PC by clicking in the **logout** box at the bottom of the screen. **Be sure you do this.** Otherwise, anyone can use your account for any evil purpose.

Other Computing Options

ITaP (<http://www.itap.purdue.edu/>) has several other options for accessing SAS. In particular, the Distributed Academic Computing Services (DACS) allows you to (with a Purdue account and password) access SAS from any machine with a Web browser. In addition, setting up a Virtual Private Network or VPN (<http://www.itap.purdue.edu/telecom/vpn/>) allows one to set up a connection that provides access to Purdue network drives (such as your ITaP drive).

Using SAS for Windows

You can launch SAS from its program icon or by double-clicking on a SAS program file – that is, any file with the .sas extension. On ITaP lab PC's, you can launch SAS by **Start menu** → **Standard Software** → **Statistical Packages** → **The SAS System** → **SAS System for Windows V8**. If you install SAS on your own machine, it will have a similar entry in your **Start** menu.

Program windows in SAS

After SAS is activated, you will see three windows: the program editor, an explorer window, and the log window. Also, there is an output window that is hidden until you actually have output. The program editor is where you will type the program that you will eventually run. It works almost exactly like Microsoft Word. The enhanced program **Editor** will give you color-coded procedures, statements, and options that will help you find errors in your program before you run it. The **Log** window will inform you of any errors in your program and the reason for the errors. Always check it to see if your program ran properly. The **Output** window is where, once you run your program from the program editor, your output will appear. (You can also cut, paste, etc. from the log and output windows.) With the **Explorer** window, you can open and view data you have read into SAS. Click on library, then the work folder, and this will show you any data sets you have read into or created in SAS for that session.

A sample SAS file

The file `tensile.dat` is the data set on page 62 of the text (Table 3-1). This data set is based on an experiment investigating the tensile strength of a new synthetic fiber blended with different percentages of cotton. Open the file `tensile.dat` using WordPad or any word processor like WORD. It should appear as follows:

```
15  7  15
15  7  19
15 15  25
15 11  12
:   :   :
```

This follows the basic format for data files. Each line contains data for a different run of the experiment. There are three variables/columns. By examining Table 3.1 and page 61, you should see that the first variable is the cotton weight percentage, the second is the tensile strength, and the third is the test sequence. Having a header/label row in a .dat file is usually avoided.

A sample SAS program

The SAS program file named `tensile.dat` contains a SAS program that reads and analyzes the data in `tensile.dat`. Here is the program:

```
options ls=75 ps=60 nocenter;
goptions colors=(none) device=win target=winprtm rotate=landscape ftext=swiss
         hsize=8.0in vsize=6.0in htext=1.5 htitle=1.5 hpos=60 vpos=60
```

```

    horigin=0.5in vorigin=0.5in;

data one;
    infile = 'C:\saswork\data\tensile.dat';
    input percent strength time;

title1 'Chapter 3 Example';
proc print data=one; run;

symbol1 v=circle i=none;
title1 'Plot of Strength vs Percent Blend';
proc gplot data=one; plot strength*percent/frame; run;

proc boxplot;
    plot strength*percent/boxstyle=skeletal pctldef=4;

proc glm;
    class percent; model strength=percent;
    output out=oneres p=pred r=res; run;

proc sort; by pred;
symbol1 v=circle i=sm50; title1 'Residual Plot';
proc gplot; plot res*pred/frame; run;

proc univariate data=oneres pctldef=4;
    var res; qqplot res / normal (L=1 mu=est sigma=est);
    histogram res / normal; run;

symbol1 v=circle i=none;
title1 'Plot of residuals vs time';
proc gplot; plot res*time / vref=0 vaxis=-6 to 6 by 1;
run; quit;

```

To **run** this SAS program, first double-click the file: the `.sas` extension links to the SAS program, which will open. You will see several windows. One is the **Editor** in which you can create or modify SAS programs. This is a simple word processor. The program in `tensile.sas` is automatically entered into the Editor because you started SAS from the program file. The other way to open this file is to start SAS and then, with the Editor window highlighted, click on File menu → **Open**.

With the Editor window highlighted, click the running figure icon in the toolbar (or do Run menu → **Submit**). This tells SAS to run the program in the Editor window. (Note: if you have `tensile.dat` in a different directory, you must edit the `infile` statement before it will run properly.)

The results of the program appear automatically in several other windows. The **Log** window is a step-by-step account of what SAS did. Use this to find errors in your programs. Special graphics appear in a separate **Graph** window which will probably be on top: use the Page Up and Page Down keys to view the graphs one by one. The **Output** window has the text output from your program. Compare the output with the commands in the SAS program as we go through them one by one in the next section. You may want to maximize the Output window (click the maximize box in the upper right, as usual in Windows).

Depending on the size of your screen, it may be hard to see everything at once. You can select any of these windows from the **Window** menu at the top of the SAS screen or from the Window toolbar at the

bottom. **Hint:** When you write a SAS program, you will probably need several attempts. It is much easier to see your results if you clear both the Log and Output windows before running the program a second time. In the Log window, right-click to bring up a contextual menu. Then do **Edit**→**Clear All**. For the Output window, the most effective approach is to go to the results summary (left-most window) and highlight the results main directory. Then click on the *X* button or do **Edit**→**Clear All**. Save your work by **File menu**→**Save** and exit from SAS by **File menu**→**Exit**.

You can print the contents of any window (Editor, Output, Graph) by using **File menu**→**Print** with the window you want highlighted. SAS tends, however, to generate too many pages of output and it is better to move the **Output** window contents into a word processor like Word. This can be done by saving the entire output file as an .rtf file and then import it into Word. To save the **Output** window as an .rtf file, highlight the **Output** window and select **File menu**→**Save As...**

The graphics can also be cut and pasted into Word documents. I've found this is easiest to do when you are in the graphics editor. With the graphics window highlighted and the graphic of interest displayed, click the **Edit Graph** button in the toolbar. Once in the graphics editor, you can add to or edit the graphic. To copy the graphic to Word, select **Edit**→**Select**→**All** and then **Copy**. You can also export the graphic as an image (.bmp, .gif, .jpeg, or .ps) and import them into Word. In this case, you cannot edit them once in Word.

Meet SAS: Basics of SAS Programming

Note very carefully that *all SAS program lines must end with a semicolon*. The indented and blank lines just make the program easier to read; they are not required. SAS executes each command when it sees the next command, so every program must end with “**run;**” in order to execute the final command. I also add a “**quit;**” to make clearing out the Output window easier.

Note also that *names in SAS should be no more than 8 characters long, should contain only letters and number, and should begin with a letter*. This applies to the names you assign to both variables and data sets. In more recent versions of SAS, this restriction appears to have been dropped, but I will continue to follow this rule so my files can be used in all version. Now let's look at the commands in the `tensile.sas` program.

options ls=75 restricts the width of the output to 75 characters. The **ps=60** restricts each page of output to be 60 lines and the **nocenter** tells SAS not to center the output.

goptions specifies various options for the graphics. These settings hopefully create graphics that look nice in Word. You may have to tweak them to get graphics that fit. Feel free to play around with the various options. They can be found in the SAS help. The **colors = (none)** option tells SAS to only use black and white in the graphics.

title1 prints a title on each page of your output to help you identify it later. You should always do this. You can print more than one line by adding *title2*, *title3*, and so on. The actual title *must* be enclosed with a single *right quote* symbol at each end of the text. The last title will be used on all subsequent graphs. To turn the last title off, you need the statement **goptions rest=title;**

data one: SAS programs usually consist of *data steps* and *procedures*. A **data** statement names a data set. The lines following a data statement create the data set. This program has one data statement and creates a SAS data set named **one** containing three variables.

infile and **input:** We read data from a file. The **infile** statement tells SAS what file to read and where that file is located. Be sure to put a single right quote symbol on either end of the file's name. The **input** statement describes the data. We name the three variables *percent*, *strength*, and *time* from left to right in the data file. The other method of inputting data is to use the *cards* or *datalines* statements and include the data set in the .sas file. This is demonstrated in `tpower.sas`.

proc: Lines that say **proc** tell SAS to run some procedure on the data. If you omit the **data=** in a *proc* statement, SAS will use the last data set created. The general form of procedure commands is SAS is

```

proc procname options;
  statement / statement options;
  statement / statement options;
  .
  .
  .

```

This program uses six procedures: `proc print`, `proc boxplot`, `proc gplot`, `proc glm`, `proc sort`, and `proc univariate`. The first procedure in this program is `proc print`, which just prints the data to verify that they were read correctly. The `data=one` is unnecessary because the data set `one` is the last data set created. This is the first command in the program that produces output.

`proc gplot` makes a scatterplot. Note that the y (vertical) variable is given first. The `symbol1` command sets the shape of the symbol to be used in the plot. The `frame` option puts a box around the plot. It is also used later on to generate a scatterplot of predicted vs. residual values from a linear model and a time series plot of the residuals.

`proc boxplot` creates boxplots of the data. Note again that the y (vertical) variable is given first. The `skeletal` option means that the whiskers of each box extend to the minimum and maximum values. The `pctldef=4` option tells SAS to use the Moore and McCabe described method of computing quantiles.

`proc glm` (and `proc mixed`) are the two linear models commands that you will need for many of your homeworks. Please consult the SAS Help for more details. We will go over the structure and meaning of this program/output in class when we get to this problem. The `model` statement has the form

response variable = list of predictor variables

The equal signs can be interpreted “is explained by”. The `output` statement enables you to save results for further analysis. This creates a new file name `oneres` that contains all the original data plus additional variables. Here the new variables are the predicted (`p=pred`) and residuals (`r=res`) values.

`proc sort` sorts the data according to a specific variable(s). In this case, the data is sorted from smallest to largest according to the predicted values from the linear model.

`proc univariate` gives basic numerical descriptions for each variable you request. If you leave out the `var` statement, SAS describes all the numeric variables in the data set. Including the `qqplot` statement adds a normal quantile plot and including the `histogram` statement adds a histogram and overlays, in this case, a normal distribution. We will again discuss these in more detail when we get to Chapter 3.

SAS Help

You have now gone through SAS basics using one template program. SAS itself can give you a more detailed tour. In SAS, do Help menu→Getting Started with SAS Software. SAS also has detailed help on each procedure. You may find this too terse to be useful. Unless you are insatiably curious, wait a few weeks before trying this. In SAS, do Help menu→SAS System Help. In the list, click Help on SAS Software Products. Most statistical procedures are in SAS/STAT and clicking on a statistical procedure gives details of the structure and options. There is also an item called Sample SAS Programs and Applications. This contains other template files from which you might borrow a set of commands.

The STAT 514 Web Page

The STAT 514 web page is located at <http://www.stat.purdue.edu/~jennings/stat514>. This web page contains (or will contain) links to all the files (e.g., data, homework assignments) as well as announcements and important dates during the semester.

To download a `.sas` program or `.dat` file, just click the file name. Then click Save this file to disk and navigate to the directory where you keep your SAS work. On a ITaP lab PC, you should see in the

Windows Explorer list a location with your login ID as its name. This is your home directory (H: drive) in which you can save files permanently. Other spaces on these machines are cleaned regularly. If you plan to use ITaP and your own computer, it may be easier to put the files on a floppy disk. You can, of course, always get another copy from the Web page.