

INTRODUCTION TO SAS

STAT 502 – SPRING 2001

Statistical analyses, in practice, are always carried out by computer software. In this class, we will use SAS to perform these analyses. While versions of SAS are available for almost every type of computer made, this document focuses on using SAS installed on personal computers, specifically SAS for Windows (PC-SAS Version 8).

Getting Started

1. Version 8 of SAS is available on the PCs in PUEC labs. You must have a Purdue University Computing Center **career account** in order to use PUEC facilities. If you do not have a career account, take your Purdue ID to any PUEC PC lab and use the Account Summary (just swipe your ID). Or you can go to MATH 231. You will keep this account as long as you are at Purdue. You can find PUEC information at <http://labinfo.cc.purdue.edu>.

2. Purdue's SAS license allows any student or staff member to get a copy of PC-SAS Version 8. If your department has PCs or if you have a PC, you can check out a SAS CD and install SAS on your machine. See Carol Funkhouser in the PUEC office in MATH G175.

Installation hints. SAS complete occupies more than 500 megabytes of disk space. The parts of SAS used in STAT 502 occupy just over 300 megabytes of hard disk space. To keep SAS to "only" 300 meg: (1) Ignore the CDs 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 PUEC 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 PUEC 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." If it does not, choose another machine or ask the lab consultant for help. Type your career account ID in the "Login" field. Use the tab key or the mouse (point and click once) to move to the "Password" field. Type your career account password (it will not show on the screen) and hit the enter key.

Accessing the course account. Once logged in, you can access the course materials from your PUEC Career Account web page. There should be a desktop icon that you click on to link to this page. From that page, click on the [Access my course resources](#) link. A screen will appear containing the courses in which you are registered. Select Stat 502 and the disk icon to the right. The resources are then available on the K: drive through Windows Explorer. There are four subdirectories, notes, hws, datasets, and sasfiles.

Logging out. After saving your work either on disk or on the H: drive, logout 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.

The STAT 502 Web Page

An alternative method to access the course account (except homework answer keys) is to go to <http://www.stat.purdue.edu/~bacraig/stat502.html>. This web page contains links to all the files on the course account as well as announcements and important dates during the semester. Bookmark this URL if you use your own machine—you will visit it often. Similar to the course account subdirectories, the page has four links, Class Notes, Homework and Exam Material, Data Sets, and SAS Files. Click on one to see a list of the files available.

- **Class Notes** contains .pdf files of the lecture notes. If you are on a PUCG machine, clicking on a .pdf file will launch Acrobat reader, from which you can read and print the notes. If you are using your own or departmental computer and do not have Acrobat reader, it can be downloaded for free off the Web.
- **Homework and Exam Material** contains .pdf files of the homework assignments. In addition, a set of practice exam questions, are made available near an exam. Homework solutions will also be available in this directory but only through the PUCG course account.
- **SAS Files** contains .sas files used in the class and can serve as templates for your homework.
- **Data sets** contains .dat files which are written in plain text.

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 PUCG 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 PUCG and your own computer, it may be easier put the files on a floppy disk. You can, of course, always get another copy from the Web page.

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 PUCG lab PCs, you can launch SAS by **Start menu** → **standard software** → **statistical packages** → **The SAS System** → **SAS v8**. If you install SAS on your own machine, it will have a similar entry in your Start menu.

A sample SAS file

The file example5-1.dat is the data set on page 49 of the text (Table 5-1). This data set is based on studying the relationship between age and systolic blood pressure. Open the file example5-1.dat using WordPad or any word processor like WORD. It should appear as follows:

```
1 144 39
2 220 47
3 138 45
4 145 47
5 162 65
  :   :   :
```

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 5-1, you should see that the first variable is the individual number, the second is the systolic blood pressure, and the third is the age. Having a header/label row in a .dat file is usually avoided.

A sample SAS program

The SAS program file named example5-1.sas contains a SAS program that reads and analyzes the data in example5-1.dat. Here is the program:

```
options nocenter linesize=72;
goptions colors=(none);
title 'Data for Chapter 5 Example';

data table5;
  infile 'C:\TeX\st502\DATA\example5-1.dat';
  input indiv sbp age;

proc print data=table5;
symbol1 v=circle;
proc gplot;
  plot sbp*age /frame;

proc reg simple;
  model sbp=age /cli clm;
  output out=fit r=res p=pred;

proc gplot;
  plot res*pred / frame;
proc univariate noprint;
  var res;
  histogram res / normal kernel(l=2);
  qqplot res / normal(l=1 mu=est sigma=est);
run;
```

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 example5-1.sas is automatically entered into the Editor because you started SAS from the program file.

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 example5-1.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 each window, right-click to bring up a contextual menu. Then do **Edit** → **Clear All**. Then highlight the Editor window and submit the program again. 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 either cut and paste from the **Output** window into Word or save the entire output file as an .rtf file and then use Word to edit. 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 indentation 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.

Note also that *names in SAS should be no more than 8 characters long, should contain only letters and numbers, and should begin with a letter*. This applies to the names you assign to both variables and data sets. Now let's look at the commands in the *example5-1.sas* program.

options linesize=72 restricts the width of the output to 72 characters and the **nocenter** tells SAS to not center the output. The 72 characters were chosen so the output fits on the computer screen.

goptions specifies various options for the graphics. The **colors=(none)** option tells SAS to only use black and white in the graphics.

title 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.

data: 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 **table5** 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 *indiv*, *sbp*, and *age* from left to right in the data file. The other method of inputting data is to use the *cards* statement and include the data set in the .sas file.

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 in SAS is

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

This program uses six procedures: *proc print*, *proc gplot*, *proc reg*, 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=table5* is unnecessary because data set *table5* 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.

proc reg is the basic linear regression procedure. This combined with **proc glm** will dominate Stat 502. The *model* statement has the form

response variable = list of predictor and nuisance variables

The equal sign can be interpreted "is explained by". The *output* statement enable you to save results for further analysis. This creates a new file named **fit** that contains all the original data plus additional variables. Here the new variables are the predicted (p=**pred**) and residual (r=**res**) values.

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. The *noprint* option suppresses much of the output. 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 5.

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.