INTRODUCTION TO SAS
STAT 512

Statistical analyses, in practice, are always carried out by computer software. In this class, I will focus on the use of SAS to perform these analyses, specifically SAS for Windows Version 9.3. Before getting started with an overview, I want to make it clear that you are more than welcome to use alternative software for the homework (e.g., Minitab, JMP, SPSS, or Matlab). I (and my TA) just won't be providing step-by-step procedures or source code for these other software nor discussing the output format and where to find various statistical summaries.

Getting Started

1. Version 9.3 of SAS is available on the PCs in ITaP labs on campus. Version 9.3 is also available to anyone affiliated with Purdue via ITaP's software remote. Both of these options require you to login via your Purdue career account.
2. Purdue's SAS license also allows any student or staff member to install a copy of PC-SAS Version 9.3 on their own computer. You can check out the necessary disks by visiting the reception area in Young Hall 5th floor.

Using the 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. Many labs are scheduled for classes during the day. Logging in, 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. Once logged in, you can access the course materials from the Stat 512 web page and save material on your career account. After saving your work, log out from the PC by clicking in the "logout" box at the bottom of the screen.

The STAT 512 Web Page

The STAT 512 web page is located at http://www.stat.purdue.edu/mlevins/Stat512_13/Notes/Stat512.html. This web page contains links to all the files (e.g., data, sas files, homework assignments) as well as announcements and important dates during the semester. Bookmark this site if you use your own machine—you will visit it often. The web page is broken up into several subdirectories like Class Notes, Homework, Exam Material, Data Sets, and SAS Files. Click on one to see a list of the files available.

- **Lecture Notes** contains .ppt files of the lecture notes. If you are on a ITaP machine, clicking on a .ppt file will launch Powerpoint, from which you can read and print the notes.
- **Homework Assignments and solution keys** contains .pdf files of the homework assignments as well (posted at an appropriate time) solution keys. If you are on a ITaP machine, clicking on a .pdf file will launch Acrobat reader, from which you can read and print the homework. If you are using your own or departmental computer and do not have Acrobat reader, it can be downloaded for free off the Web.
- **SAS Files** contains .sas files used in the class and can serve as templates for your homework.
- **Data sets** contain .dat or .txt 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 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 put the files on a flash drive. You can, of course, always get another copy from the Web page.

Using SAS for Windows

A sample SAS program

The file pisa.sas is the SAS code for the example found in Topic2. This problem deals with the relationship between the lean in the tower of Pisa and time measured in years. Here is the program:

```
title1 'Leaning Tower of Pisa Example';

data a1;
 input year lean @@;
cards;
75 642 76 644 77 656 78 667 79 673 80 688 81 696 82 698
83 713 84 717 85 725 86 742 87 757 102 .
;

data a1p; set a1; if lean ne .;
run;

ods html close;
ods rtf file=''H:\pisa.rtf'';

proc print data=a1;
run;
```

```
symbol1 v=circle i=sm70;
proc gplot data=a1p;
 plot lean*year;
run;

symbol1 v=circle i=rl;
proc gplot data=a1p;
   plot lean*year;
run;

proc reg data=a1;
 model lean=year/clb p r;
 output out=a2 p=pred r=resid;
 id year;

proc gplot data=a2;
 plot resid*year/ vref=0; where lean ne .;
run;
```

To run this SAS program, first double-click the file: the .sas extension links to the SAS program that 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 pisa.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.
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, by default, will have HTML output from your program. Because I typed ods html close; nothing is put in the output window. If you use ods listing;, there will be text output in the output window. Since I specified a file, all output is being sent to that file. You may want to play around a bit with the different output options to see which you prefer. 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 menus. Then do **Edit −Clear All**. Then highlight the

Editor window and submit the program again. If you are using HTML or listing output, you can 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 either to 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 pisa.sas program.

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 **a1** containing two variables.

infile and input: There are two ways to incorporate data into a SAS program. One is to 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 two variables *year* and *lean*. The other method of inputting data is what we use here. The datalines statement allows us to include the data directly in the 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 three procedures: *proc print, proc gplot*, and *proc reg*.

The first procedure in this program is *proc print*, which just prints the data to verify that they were read correctly. The *data=a1* is necessary because the data set *a1p* 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 *symbol 1* command sets the shape of the symbol to be used in the plot. The i= option draws a “smoothed” curve. This procedure 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 512. The model statement has the form

response variable = list of predictor and nuisance variables

The equal sign can be interpreted ”is explained by”. The / tells SAS that some options are being requested. We will discuss these later. The *output statement* enables you to save results for further analysis. This creates a new file named **a2** that contains all the original data plus additional variables. Here the new variables are the predicted (p=**pred**) and residual (r=**resid)** values. These are then used to generate a residual plot. We will talk about this plot in Chapter 2.

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.