Preliminary Lab Instructions for PubH5450

You should refer to the syllabus and Course Info handout, that was distributed in class and is also downloadable as a link from www.biostat.umn.edu/~sudiptob/5450.html

Here are the instructions to get into the biostat server called saturn and use the SAS software in a preliminary example.

You must be working on a machine that has telnet capabilities. If you are working on Windows 98 or later, and you have internet connection, the telnet facility should be there in your machine. Whatever your environment, you need to know how to open the command prompt (also called console window or MS-DOS prompt) and it usually appears as a screen with a dark background.

1 Logging into saturn the first time:

The following sequence of commands applies to you ONLY if you are logging in the first time.

Step 1: telnet to the biostat server: Type the following into the command line:

```
telnet saturn.biostat.umn.edu
```

Step 2: You will be prompted for your login id and password. For the **sake of illustration**, let us **suppose** that your *user name* is `u201` and your *password* is `CrkEFhnx` - This information appears in the computer user sheets that were distributed in class, the lower half of which you filled in and returned to me. Now you type in these *exactly as they appear*. **Make sure to put in caps where there are caps! Also, please actually use the login id and password that appears in your sheet - the names I am using are purely for illustration.** Thus you enter:

```
login:  u201
password:  CrkEFhnx
```

Step 3: The system will automatically ask you to change your password. The procedure begins by asking you to retype your login password, which is the password you just typed in Step 2. After doing that, it will prompt you to change your password, where you should type in a password 6-8 characters long with at least one number and one letter. Please be sure to **REMEMBER YOUR PASSWORD.** Then you will be asked to retype this new password. That is it! Your password has been set for the rest of the semester. Suppose, again for illustration, I decide to use the password stupido1 for my future use. So, here is what I will have to do:

```
Enter login password:  CrkEFhnx

Enter new password:  stupido1

Re-enter new password:  stupido1

Your password has been changed.
```

Step 4: Write down your password somewhere and be sure to remember. For all future logins to the saturn server could, as per the illustration, use `stupido1` as your password, instead of `CrkEFhnx`. Thus,

```
login:  u201

password:  stupido1
```

should log you in smoothly.

2 Using SAS software on the saturn

There are two steps to the complete formation of a SAS program. We will briefly go over these two steps now. The first step involves writing your SAS code and data in a text editor. You may want to enter your data and the code together in one file, or you may want to create two separate files - one containing the data and the other containing the code. The second step is that of compiling and running the SAS program that you wrote. We will now see the details of these two steps in the following two subsections.

2.0.1 The pico editor

A text editor is a software that allows you to write stuff in it - without adding fancy formatting features. Thus it is not a word processor. For example, in the Windows environment, notepad is a text editor, while MS Word or Wordpad are not. Of course several word processors, like the previous two, can save files as unformatted text and therefore pretend that it is also a text editor. For our purposes, we will be working only with text editors. The saturn biostat server (fortunately or unfortunately) is not a Microsoft operating system. It is a UNIX system and therefore may not be as familiar to many of you as Windows is. UNIX however is a very stable operating system and is the OS of choice for scientific computing and demanding jobs. You may find more information about UNIX and its text editor pico, in the Syllabus and Course info file. Let us now see how we use the pico editor to create and implement a simple SAS program in UNIX.

Suppose we want to create a text file called example1.sas. You first type the following in the command line:

```
pico example1.sas
```

Assuming you did not already have a file by the above name (you are creating a *new* file!) pico will open a blank sheet for you. In that file you want to type in some data and then write a simple SAS program to obtain some simple descriptive measures for the data. Assume that you are given the following data:

Gender	Height	Weight
M	68.5	155
F	61.2	99
F	63.0	115
M	70.0	205
M	68.6	170
F	65.1	125
M	72.4	220

In order to write the above data in SAS and carry out a simple descriptive statistics exercise, we type in the following code in example1.sas:

```
options ls=80 nodate;

  data htwt; input gender $ height weight; datalines; M 68.5 155 F 61.2 99 F 63.0
  115 M 70.0 205 M 68.6 170 F 65.1 125 M 72.4 220 ; run; proc means data = htwt;
  run;
```

Having written the above in the pico editor, you may press `Ctrl+X` which prompts you to exit, but will ask you whether you want to save the file or not. Say yes to this prompt by typing y - otherwise, you'll lose all your code! Alternatively, you may save your file by typing `Ctrl+O` which writes the material in the buffer to a file. Play with it - to get yourself acquainted with this.

Let us analyse the above code. We look closely at three different blocks of the code. The first line, `options ls=80 nodate;` sets the page width (to 80 columns) and suppresses the printing of dates and times - you may not want to let your instructor know when you did your homework! The second block is, what is called in SAS, a data step. This is the step where you actually inform SAS about your data. You introduce variables gender, height and weight. Using the command `datalines`, you tell SAS that what follows will be the data. SAS will read this in the order of the variables. Thus the first column will be gender (the $ sign indicates that gender is a character variable), followed by height and weight. **Do not miss the semicolon at the end of datalines**, telling SAS that all the data have been entered. In the third and final block we invoke a procedure called `proc means` that gives a little summary of the data.

2.0.2 Running SAS

Having saved example1.sas in a text editor, we now proceed to the compilation and running of the program.

Step 1: To compile we type, sas `example1.sas` at the command prompt.

Step 2: To see what we have, first type `ls` at the command prompt.

Step 3: You should always have a `.log` file in your directory. You may also have a `.lst` file - which means that your program ran successfully. The `example1.log` file contains the compilation report. In case SAS was unable to run the program, you should look into this .log file to see if there are any errors. If no errors are detected, `example1.lst` would be your output file.

Try the above code and then replace `proc means` with `proc univariate`.