

EXST 700x

Lab #2: Basic Data Organization and PROC Statements

Tips

1. Previous lab assignments are posted online if you need to refer to them.
2. It is a good idea to save your SAS code often while you are working to avoid accidental loss. You can save it to your desktop or the temp directory, but the lab machines have too many users for us to guarantee the integrity of anything saved on the local hard drive, and you may not be able to get the same machine next week. It would be safest to bring a flash drive with you to save your work or save your work to TigerBytesII
3. You only need to save your SAS program and data where applicable. It is not necessary to save output or log since you can get them every time you run SAS program.
4. Read the details discussed in the lab assignments. It will help you to understand SAS.
5. You can use computers in either lab room 11 or lab room 44 if the lab is not occupied by a class. Room 44 has fewer scheduled classes and is usually open. If you want to install SAS onto your own computer, see the staff in room 161.

SAS input and output

For this and future assignments it is recommended that you create a directory specifically for the assignment. I will create a directory called “EXST700X Assignment 02” on a flash drive. If you do not have a flash drive you can create a subdirectory in your “My Documents” directory which is probably called “C:\Users\YourLogonID\Documents”. If you save a file without specifying a directory it will go into default initial folder used by SAS, a directory called “C:\Users\YourLogonID\Documents\My SAS Files\9.3”.

Most of the data sets that we will use for the EXST 7005 lab are small enough so that they can be placed directly into the SAS program in the program editor, as we did last week. However, many real world practical uses of SAS require using datasets much too large to include in the program. In this assignment access some external data sets.

Today’s example needs an external file called “grades.txt” that contains exactly the same data as was originally in the program. To use the external data set we simply remove the data from the program and use an “INFILE” statement to indicate to the program where it may find the data. Give the name of the data file with the full path in quotes;

```
INFILE 'F:\EXST700X Assignment 02\GRADE.txt';
```

The CURRENT FOLDER: SAS keeps track of what is called the current folder. When you initially open the current folder is usually “C:\Users\YourUserID\Documents\My SAS Files\9.3”. If you double click on a SAS file to open it, the directory where that file is stored becomes the current folder. If files to be input and files to output are to be placed in the same current folder, it is not necessary to provide the full path. The SAS code given as an example in this assignment is available online with the data included as “example 02 example.TXT”. The current folder is shown on the right side of the lower border of the SAS window. The current folder can be changed by clicking on the name of the current folder and using

windows to redirect the folder. If you have created a folder for today's exercise then direct SAS to this directory and make sure the data is stored in that folder.

ODS output files: The Output Delivery System (ODS) can create some output files of various types. We have already seen the ODS control output in our usual housekeeping statements.

```
dm 'log;clear;output;clear';
options nodate nocenter pageno=1 ls=78 ps=55;
OPTIONS FORMCHAR="|----|+|----+=|-/\\<>*" ;
ODS listing; ods graphics off; *ods noresults;

title1 "EXST 700X Example 02";
title2 "James P Geaghan, SECTION 0";

***EXST 700X*****;
*** Example 02      ***;
*** James P Geaghan ***;
*** SECTION 0      ***;
*****;
```

Notice that I have embellished the statements above with a box made of comment statements, and that I have deactivated the `ods noresults;` statement by turning it into a comment. Otherwise it would suppress the “RESULTS VIEWER” output of the windows program.

Other statements that may be included in this section that control program behavior and input or output. Three that you may find useful are the ODS statements;

```
ODS HTML style=minimal body='EXAMPLE02.html';
ODS RTF style=minimal body='EXAMPLE02.rtf';
ODS PDF style=minimal body='EXAMPLE02.PDF';
```

These statements will create an HTML output file, RTF output file and a PDF output file, respectively. The `style=minimal` simplifies the HTML output by suppressing font changes and background shading. These files will be written to the “current folder” unless full directory path information is provided within the quotes of the “body” option.

The HTML output is the same as the output SAS writes to the “RESULTS VIEWER”. However, the output included in the file starts at whatever point in the program the statements are placed. That is, only output created after the insertion point of the ODS statement is included in the output file. The output for each ODS ceases with the statements;

```
ods html close;
ods rtf close;
ods pdf close;
```

Permanent SAS datasets: When a data set is created in SAS with a simple data step such as “DATA grades;” it goes into a temporary SAS library called the “WORK” library. When the program ends the WORK library is cleared and lost to future use. These dataset can be referred to by single data set name statements like “CLASS” or as two-level name statements where the two levels are separate by a period, such as “WORK.SAS”.

It is also possible to create a permanent SAS LIBRARY that will persist when SAS is exited, and can be used for future analysis. To do this, a SASLIB statement must be placed early in the program before the DATA step. For example, the following statement,

```
LibName SASLib 'F:\EXST700X Assignment 02';
```

would create a SAS library called **SASLib** in the directory “F:\EXST700X Assignment 02”. To create a permanent dataset called **GRADE** in a DATA step, a “two level” dataset name would be used.

```
data SASLib.GRADE;
```

After running this statement you will find a SAS dataset called “**grade.sas7bdat**” in the directory specified in the SAS **libname** statement.

There is also a default directory for permanent storage called **SASUSER**, so **SASUSER** could be substituted for **TEMPDIR** without a prior **SASLIB** statement. An alternative would be to use the statement;

```
data sasuser.GRADE;
```

PROC SORT: The SORT procedure is used to alter a dataset such that all the observations are ordered according to values of one or more variables.

It is possible to leave the original data set unaltered and to sort the data into a newly created data set. To do this the SORT statement will be followed by an OUT option that gives the name of the new SAS dataset. The OUT option must part of the PROC SORT statement, before the semicolon that ends the SORT statement (though it does not have to be on the same line).

If you don’t use an OUT option the original dataset will be replaced with the sorted version. You can make the new dataset either a temporary one or a permanent one by choosing the library’s name before the dataset. By default, “WORK” is a temporary SAS library, and “SASUSERS” is a permanent SAS library.

A BY statement always follows a SORT statement. It specifies the variable by which the dataset is to be sorted. The default sorting order is ascending. In this case, we are going to sort data according to sections.

```
proc sort data=SASLib.GRADE; by SECTION;
run;
proc print data=SASLib.GRADE; by SECTION;
  Title3 'Data listing from PROC PRINT';
  Title4 'With a by statement';
run;
```