SAS 1. Getting started with SAS

Note 1: SAS is available to be freely downloaded to your computer, please see "SAS installation" note for instructions.

Note 2: Alternatively, as shown in today's lectures, you can use SAS directly from the virtual sinc site: https://it.stonybrook.edu/services/virtual-sinc-site

1. The SAS language

2.

- Open SAS: The screen is divided in to 3 windows, the "Explorer", the "Log (untitled)", and the "Editor (untitled)". Across the top, there are several pull down menus including: File, Edit, View, Tools, Solutions, Window, and Help. Below these, there is the tool bar. Most important is the ***little running person*** who stands for "run the program"!
- Note that the cursor is in the "Editor" and as a result the title bar of the "Program Editor" is highlighted, signifying it is the active window.
  1. The following is a sample program. Please type it in the "Editor". Then click on the ***little running person***.

```
Data test;
input gender $ exam1;
datalines;
M 80
F 90
M 86
F 78
;
proc print data=test;
run;
```

  2. Save the SAS program: File → Save as (for example, **a:orange**). You can also save by clicking on the diskette like icon in the tool bar.
  3. You can also save the SAS Output and the SAS Log. Just be sure that the cursor is in the window you intend to save. Use the square on the left top corner to enlarge/shrink the window. Please read the SAS Log for error messages (in red) and warnings (in green).

Errors People who are just learning a programming language often get frustrated because their programs do not work correctly the first time they write them. To make matters worse, SAS errors often come up in bright red letters, and for the poor person whose results turn out more red than black, this can be a very humbling experience. You should expect errors.

SAS uses statements to write a series of instructions called a SAS program.
SAS programs A SAS program is a sequence of statements executed in order. A statement gives information or instructions to SAS and **must be appropriately placed** in the program.

SAS statements As with any language, there are a few rules to follow when writing SAS programs. The most important rule is

Every SAS statement ends with a semicolon.

Some other tips for SAS includes

--- SAS statements can be in upper- or lowercase.
--- Statements can continue on the next line (as long as you don't split words in two).
--- Statements can be on the same line as other statements.
--- Statements can start in any column.

Comments To make your programs more understandable, you can insert comments into your programs. There are two styles of comments you can use: one starts with an asterisk (*) and ends with a semicolon (;). The other style starts with a slash asterisk (/*) and ends with an asterisk slash (*/). The following SAS program shows the use of both of these style comments:

```
* Read books' page from file;
DATA books;
INFILE ' D:\Users\mshu\Desktop\Books.dat';
INPUT pages fs;
Run;
PROC PRINT DATA = books; /* Print the results */
RUN;
```

```
* creat a file test then print it
  my name is min shu
  by the way, today is cold;
Data test;
input gender /*you know*/$ /*m*/ exam1/*first exam*/;  *there are two
variables;
*data part almost finish;
datalines;
M 80
F 90
M 86
F 78
;
*
*above is my data
*;
proc print data=test; /* Print the test data */
title "test3";
/*how are you balllllllllllllllllllllllllllllllllllll,now,
let's run it*/
run;
/************************************************
proc corr data=test;
run;
************************************************/
```

2. SAS data sets

Before you run an analysis, SAS must be able to read your data. The data must be in a special form called a SAS data set.

Variables and observations In traditional SAS terminology the data consist of variables and observations. Adopting the terminology of relational databases, **SAS data sets are also called tables**, observations are also called rows, and variables are also called columns. Below you see a rectangular table containing a small data set

Variables (Also Called Columns)

Observations (Also Called Rows)	Id	Name	Height	Weight
1	53	Susie	42	41
2	54	Charlie	46	55
3	55	Calvin	40	35
4	56	Lucy	46	52
5	57	Dennis	44	.
6	58		43	50

Data types In SAS there are just two data types: numeric and character. Numeric fields are, well, numbers. They can be added and subtracted, can have any number of decimal places, and can be positive or negative. In addition to numerals, numeric fields can contain plus signs (+), minus signs (-), decimal points (.), or E for scientific notation. Character data are everything else.

If a variable contains letters or special characters, it must be character data. However, if it contains only numbers, then it may be numeric or character.

Missing data Sometimes despite your best efforts, your data may be incomplete. The value of a particular variable may be missing for some observations. In those cases, missing character data are represented by blanks, and missing numeric data are represented by a single period (.). In the preceding data set, the value of Weight for observation 5 is missing, and its place is marked by a period. The value of Name for observation 6 is missing and is just left blank.

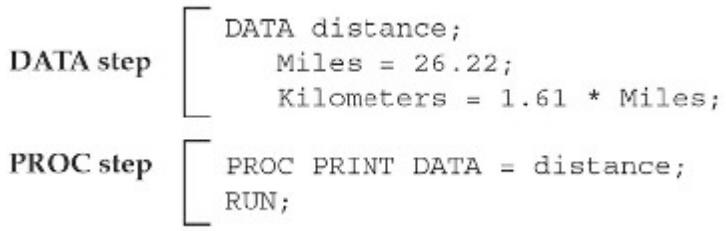
Rules for SAS names Follow these simple rules when making up names for variables and data set members:

--- Names must be 32 characters or fewer in length.
--- Names must start with a letter or an underscore (_).
--- Names can contain only letters, numerals, or underscores (_). No %$!*&#@.
--- Names can contain upper- and lowercase letters.

This last point is an important one. SAS is insensitive to case so you can use uppercase, lowercase or mixed case --- whichever looks best to you.

3. The Two Parts of a SAS Program

SAS programs are constructed from two basic building blocks: DATA steps and PROC steps. A typical program starts with a DATA step to create a SAS data set and then passes the data to a PROC step for processing. For example



DATA and PROC steps are made up of statements. A step may have as few as one or as many as hundreds of statements. Most statements work in only one type of step_in DATA steps but not PROC steps, or vice versa.

DATA steps start with the DATA statement, which starts, not surprisingly, with the word DATA. Procedures, on the other hand, start with a PROC statement in which the keyword PROC is followed by the name of the procedure (PRINT, SORT, or MEANS, for example). The DATA step has ended when it reaches the PROC statement. The PROC step ends with a RUN statement. The table below outlines the basic differences between DATA and PROC steps:

DATA steps	PROC steps
▶ begin with DATA statements	▶ begin with PROC statements
▶ read and modify data	▶ perform specific analysis or function
▶ create a SAS data set	▶ produce results or report

4. The DATA Step's Built-in Loop

DATA steps read and modify data, and they do it in a way that is flexible, giving you lots of control over what happens to your data. However, DATA steps also have an underlying structure, an implicit, built-in loop. You don't tell SAS to execute this loop: SAS does it automatically. Keep in mind the following:

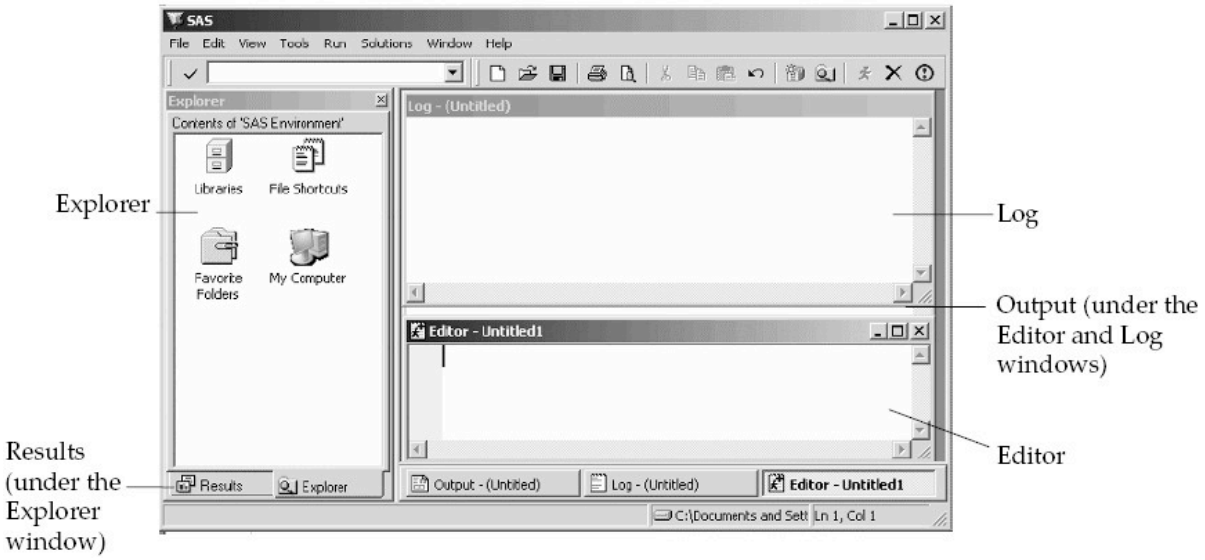
DATA steps execute line by line and observation by observation.

5. Windows and Commands in the SAS Windowing Environment

There are five basic SAS windows: the Results and Explorer windows, and three programming windows: Editor, Log, and Output.

Editor This window is a text editor. You can use it to type in, edit, and submit SAS programs as well as edit other text files such as raw data files.

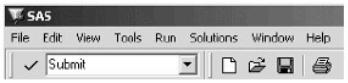
Log The Log window contains notes about your SAS session, and after you submit a SAS program, any notes, errors, or warnings associated with your program as well as the program statements themselves will appear in the Log window.
Output If your program generates any printable results, then they will appear in the Output window.
Results The Results window is like a table of contents for your Output window; the results tree lists each part of your results in an outline form.
Explorer The Explorer window gives you easy access to your SAS files and libraries.



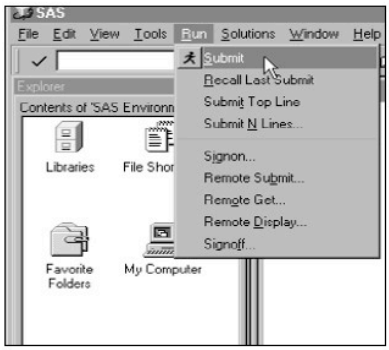
6. Submitting a Program in the SAS

Getting your program into the editor The first thing you need to do is get your program into the Editor window. You can either type your program into the editor, or you can bring the program into the Editor window from a file.

Submitting your program Once your program appears in the editor, you execute it using the SUBMIT command. Depending on your operating environment, you have a few choices on how to execute the SUBMIT command.

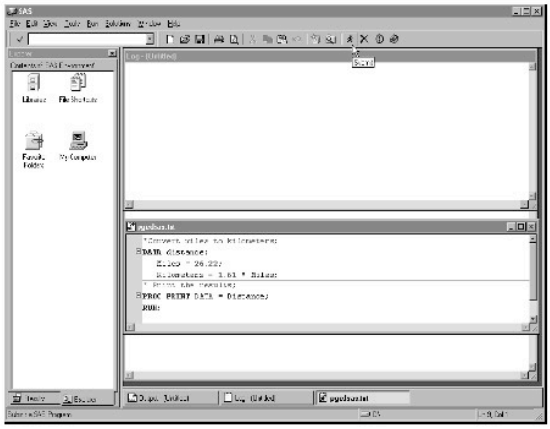


Make the Editor window active and enter SUBMIT in the command line area of your SAS session.



Make the Editor window active and select `Submit` from the Run pull-down menu.

The figure to the right shows a program in the Enhanced Editor in the Windows operating environment ready to be submitted using the `Submit` icon on the toolbar.



You can also read your SAS log in the Log Window and view your results in the Output Window.