

Introduction to SAS

Philip Dixon, 12 August 2022

SAS versions SAS is a huge program with many components. We will use a small fraction of its capabilities in this class. It is available in three versions. SAS Enterprise Guide and SAS Studio are graphic project organizers that interface to “ordinary” SAS. I don’t use either (for various reasons) so I won’t discuss them, but I encourage you to explore what they can do if you like project organizers. On many installations, the ‘.sas’ type of file is linked to EG. If you want to recreate what I do, you have to start “ordinary” SAS then open the file of SAS commands.

Accessing SAS The easiest way to use SAS is to work on a computer that has SAS installed. This includes all the computers in Stat computer labs, the computers in most public computer labs, the computers in many departmental computing rooms, and many individual “office” or “lab” computers. If you are at one of those computers, just double click the SAS icon on the desktop and SAS will start.

You can also install SAS on your own or ISU-owned computer. See <https://www.stat.iastate.edu/statistical-software-sas>, about half-way down the page, to request a license.

If you are using a computer without SAS installed, you can run SAS remotely using Remote Desktop. This uses your computer as a display, keyboard, and mouse for a remote computer that has SAS installed. See

<https://www.stat.iastate.edu/statistical-software-sas>, Terminal Server SAS (top of the page) for instructions. This runs SAS on one of the Stat department terminal servers.

Some additional advice on installing access to terminal server SAS on a Windows machine (the only type I can use to check the instructions).

Open <https://www.stat.iastate.edu/statistical-software-sas>, and look for the Terminal Server SAS, “If you are using a Windows 7 or higher” text. Click the “here” link. That takes you to a CyBox folder called RemoteApp. You will need your netid, password, and 2nd factor authentication to get access to that CyBox folder. Click on the “SAS Server.rdp” name and select Download in the popup window. This should highlight the file. Then click the download arrow (top right of the window, looks like a downward pointing arrow. Your downloads window will pop up and the file should be in your downloads folder.

If you just click the download arrow before selecting the SAS Server.rdp file, you will download a zipped file called RemoteApp.zip with both files. Just unzip the contents, click RemoteApp a couple of times to drill down through the file structure until you get to SAS Server.rdp. Move that file to some place more accessible (desktop, a stat587 folder).

All of this download stuff only needs to be done once. To start SAS, just double click on the SAS Server.rdp file.

There is also SAS On Demand for Academics. This is like Remote Desktop in that your computer handles the display, keyboard and mouse. In this case, it is through SAS Studio on your computer. SAS itself runs on a server at SAS headquarters. See <https://www.stat.iastate.edu/statistical-software-sas>, at the bottom of the page for more information.

SAS Principles

“Ordinary” SAS follows a ‘write a program / run a program’ model. To get data analyzed using SAS, you must:

1. Write commands that tell SAS how to read the data
2. Write commands that tell SAS what sort of analyses you want
3. Tell SAS to execute those commands
4. Look at the LOG window to see whether SAS understood your commands
5. If you made a mistake, you need to:
 - (a) edit your program
 - (b) resubmit all or part of it
6. Look at the output and interpret it.

This year we will be using SAS version 9.4. This has a couple of major changes from earlier versions. If you use SAS on one of the Statistics servers (what we will do in lab), you will be running 9.4. If you use SAS in a departmental computing room or office, you may be running an earlier version. If so, I strongly suggest you ask your IT folks to upgrade.

After you start SAS, you will see a screen full of windows. Your commands go in the program editor window. After you run your commands, errors and any notes about the analysis are in the log window. The output appears in a Results Viewer window.

Default directory location: I find it very helpful to organize my work in folders/directories. I store data, SAS programs, and output in that directory. Right after I start SAS, I change the default directory to that for the current project. The working directory is shown in the lower right part of the frame of the SAS window. For many computers, that will be something like C:\users\netid. This is one level above your Documents folder. To change that, double left click on the directory name, navigate to the appropriate directory, then click OK.

SAS programs

SAS programs are organized into:

- DATA steps. These convert your data file(s) into a form that is usable by SAS.
- PROC steps. These ask SAS to run a particular analysis.

You may have more than one DATA step and more than one PROC step. The typical 587/402/500 program has one DATA step and multiple PROC steps.

SAS is a very powerful database manager. We will use only the simplest parts of its data management capability.

Data structure: In this course, we will assume that a data file has 'flat-file by rows' format, i. e.:

- Each row contains one observation.
- Data values are separated by spaces (one or more).
- There may be a header line giving the variable names. This can be omitted.

Here is an example:

```
group yield
a 29.9
a 11.4
b 26.6
b 23.7
a 25.3
b 28.5
b 14.2
b 17.9
a 16.5
a 21.1
b 24.3
```

There are 11 observations (one per row). Each observation has 2 variables. The first identifies a treatment group; the second is the yield.

Data sets: SAS does not work directly on the original data file. The analysis parts of SAS (the proc steps) use data that are stored in a SAS data set. So, the first part of a typical SAS program creates a SAS data set from the original data. The minimum DATA step has a DATA line, an INFILE (or a CARDS line or a DATALINES line) and an INPUT line.

```
data tomato;
```

```
infile 'tomato.txt';
input group $ yield;
run;
```

This creates a SAS data set called tomato. The data come from the file 'tomato.txt' in the default directory. If you don't change the default directory (procedure described above), you need to include the full path name to the file, e.g. f:\tomato.txt or C:\Users\pdixon\Documents\Stat587\tomato.txt.

Each row has two values. The first value will be stored with the variable name GROUP; the second will be called YIELD.

By default, SAS treats all variables as numeric. That is fine for YIELD, but the values of GROUP are not numbers. You tell SAS to read a variable as a character variable using the \$ **after** the variable name on the input line. If you omit the \$, you get lots of errors (value not numeric) in the log window.

PROC steps: Each proc step tells SAS to perform a particular analysis. Each proc step includes commands that describe the specific analysis. Some procs produce new SAS data sets. I will introduce appropriate commands as needed throughout the semester.

```
proc ttest;
  class group;
  var yield;
  title 'T-test of tomato yield';
run;
```

```
proc boxplot;
  plot yield*group;
  title 'High resolution boxplots for each group';
run;
```

Each command to SAS **MUST** end with a ; One command can span multiple lines, or one line can have multiple commands. SAS only cares that each command ends with a ;

HINT: If SAS gives you a bunch of errors, and the commands look correct, check to see if you left out the ; This is a common problem, and omitting the ; really confuses SAS.

While not always necessary, it is good practice to end each PROC or DATA step with a run; command.

HINT: If you submit some SAS commands but don't get any output. Or, you don't get any output from the last proc; step. You forgot the run; command. You don't need to resubmit everything. Just type run; in your program editor window (probably at the end), highlight it, then click submit. You should get the output from your last proc; step.

There is almost always more than one way to do something in SAS. I will describe the approach that works for me. If you see a different PROC used or a different sort of DATA step, it may just be a different way to accomplish the same thing.

Working with SAS

To start SAS: Click on the SAS icon.

The windows:

Four windows will appear on your screen.

- **Results/Explorer:** Results provides a point-and-click way to navigate directly to a specific result. I find it usually easier to page up and page down through the output window. Explorer provides a point-and-click interface to SAS data sets. This can be handy when you want quickly glance at a data set.
- **Editor:** You enter your SAS commands here. When you give the SUBMIT command, SAS will execute the commands that are in this window. You can execute a subset of the commands here by highlighting the desired commands before using SUBMIT.
- **LOG:** This window contains SAS's responses to your commands. Any errors will appear here. Various informative messages will appear here.
- **OUTPUT:** This window contains the output from your SAS commands.
- Starting in version 9.3, something called ODS HTML is turned on by default. As a result, the Results Viewer window will be added to your screen and your output will appear there instead. This is also saved as an HTML (.htm) file.

editing files The SAS editor window works just like a word processor. Type your commands in and use the enter key to get a new line.

If you use Word or other standard word processor to edit your file(s), you need to save the files in text format (.txt) using SaveAs. Or, cut and paste from the Word document to the program editor window. SAS can not read files saved as Word documents (.doc or .docx files).

Loading a class program The class web site will have examples of SAS programs. You can download them using your choice of browser, save them, then load them into SAS using FILE/OPEN.

The default extension for a sas program is .sas. Your browser may not allow you to save the file as a .sas file. You may have to save the file as a text file (.txt). Then, rename the file (highlight the file name, right click, rename) and replace the '_sas.txt' with '.sas'. This is a windows/Internet Explorer problem.

Running SAS code Typing code into the editor window doesn't run the code. To run it, make the window with your code active by clicking somewhere in the window. Or, right click on the name of that window in the list at the bottom of the screen. Then click the running person icon on the

main menu bar (submit), or select Run / Submit from the main menu. If you click the running person icon and a new program editor window opens up, your code window wasn't active.

To run only certain lines of code: highlight the lines you want by shift-clicking or mousing over code while holding down the left mouse button. Then submit (the running person icon).

When you submit code, the code is copied to the log window along with any feedback from SAS. If the code is good, you'll get information about what SAS did. If it wasn't, you'll get errors. The results will appear in the results viewer window. If you don't get any results, check the log for errors.

Saving a file To save the contents of any window to a file, under the *file* menu in the desired window, click on *save as...*, type in (or click on) where you want to save it, name the file, and click *OK*.

Reminder: always check the pathname to make sure SAS will save the file where you want to save it. If you haven't changed the default directory, your work will be saved in the default directory, which may not be easy to find.

Exiting SAS: File / Exit from the main menu.

Exercise:

- Download the creativity.csv file from the class web site and make sure it is in your SAS working directory.
- Type the following into the program editor:

```
proc import datafile='creativity.csv' out=creativity; run;
```

Then execute that program.
- Look at what is in the log window. You should see that SAS created a data set called creativity. You will not see anything in the Results window because you haven't yet done anything that produces output.
- Type `proc print; run;` into the program editor below the code that's already there. They run the code. This will run everything there, which includes the proc import. That will generate an error because the creativity data set already exists. Make sure you recognize what an error looks like. In this case, that's not a problem because we already have the data set. You should not have an error with the proc print. If it ran, the contents of the creativity data set should appear in the Results window.
- If you want to try out more things, read the data file using the File / Import Data menu.