

# Introduction to SAS on unix

Biostatistics Consulting Center  
University of Massachusetts School of Public Health

This handout is a brief introduction to using SAS on the unix operating system. Although it specifically focuses on sas on OIT's `stats` server, almost everything applies to Linux as well. The exceptions are discussions of connecting to the server, XWindows, and printing (or lack thereof).

For working on `stats` server, you will need:

- a username and password for `stats` server. If you don't have one, see <http://www.umass.edu/statdata/software/unix/>
- SecureCRT (for Windows) or JellyfiSSH (for Mac) or equivalent software for connecting to `stats` server – You can download these from <http://www.oit.umass.edu/software/download/>
- XWindows software for interactive use. For information on XWindows software options, see <http://www.umass.edu/statdata/software/handouts/xwindows.html>

Using off-campus connections (modem, DSL or Cable), XWindows is generally too slow to be practicable. Use SAS in batch mode.

## Useful references

For connecting to `stats` server from a PC with SecureCRT or from a Mac with MacSSH:

SecureCRT	<a href="http://www.oit.umass.edu/ssh/config/win_securecrt.html">http://www.oit.umass.edu/ssh/config/win_securecrt.html</a>
JellyfiSSH (Mac OS 10)	<a href="http://www.oit.umass.edu/ssh/config/mac_jellyfish.html">http://www.oit.umass.edu/ssh/config/mac_jellyfish.html</a>
MacSSH (Mac OS 9)	<a href="http://www.oit.umass.edu/ssh/config/mac_macssh.html">http://www.oit.umass.edu/ssh/config/mac_macssh.html</a>

General information on using unix:

<http://www.umass.edu/statdata/software/handouts/unix.html>

If you need to transfer files between `stats` server and your PC or Macintosh:

WinSCP for Windows	<a href="http://www.oit.umass.edu/sftp/config/win_winscp.html">http://www.oit.umass.edu/sftp/config/win_winscp.html</a>
Fugu for Macs	<a href="http://www.oit.umass.edu/sftp/config/mac_fugu.html">http://www.oit.umass.edu/sftp/config/mac_fugu.html</a>
Fetch for Macs	<a href="http://www.oit.umass.edu/software/details/mac_fetch.html">http://www.oit.umass.edu/software/details/mac_fetch.html</a>

If you plan to use SAS in batch mode, instructions on using one of the available text editors:

pico	<a href="http://www.umass.edu/statdata/software/handouts/pico.pdf">http://www.umass.edu/statdata/software/handouts/pico.pdf</a>
emacs	no local documentation available

---

I.	introduction.....	3
	documentation .....	3
	interactive vs. batch .....	3
II.	sas syntax.....	4
	general rules.....	4
	data step.....	4
	libnames.....	5
	procs .....	5
III.	SAS batch mode .....	6
	input and output files .....	6
	graphics output .....	6
	background mode .....	8
IV.	interactive mode - basics .....	9
	managing SAS windows.....	9
	using XLiveCD.....	9
	using Mac X11 .....	9
	Online Help .....	10
V.	SAS Interactive Environment.....	10
	SAS environment commands .....	11
	menus .....	11
	SAS command bar and toolbar.....	11
VI.	sample SAS session – interactive mode .....	12
	enter data using Viewtable.....	12
	save your data .....	13
	temporary .....	13
	permanent .....	13
	procs .....	14
	editing in the Program Editor .....	15
	point and click .....	15
	selecting text.....	15
	ending text selection.....	15
	line editing commands.....	16
	submit a SAS program.....	17
	entire program .....	17
	selected lines.....	17
	recall a SAS program.....	17
VII.	more about working with SAS .....	18
	save the contents of a window .....	18
	preview and save graphics .....	19
	save graphics output in a file .....	20
VIII.	printing sas output.....	21
IX.	moving SAS files between computers.....	21

---

## I. introduction

The following SAS System products are available on `stats` server:

Base SAS	
SAS/AF	
SAS/ASSIST	SAS/LAB
SAS/CONNECT	SAS/OR
SAS/EIS	SAS/QC
SAS/FSP	SAS/SHARE
SAS/GRAPH	SAS/STAT
SAS/IML	
SAS/INSIGHT	

### documentation

SAS documentation is available online at:

<http://support.sas.com/onlinedoc/913/>

You can order printed manuals at <http://support.sas.com/publishing>. If you've never used SAS, we recommend **Getting Started with The SAS System, Version 8**. This is useful for either Version 8 or Version 9, which are very similar.

### interactive vs. batch

Interactive SAS uses a point-and-click graphical user interface. Using this interface from a PC (Windows) or Mac requires XWindows emulation software. If you don't already have such software, see

<http://www.umass.edu/statdata/software/handouts/xwindows.html> for information on getting and using XWindows software.

If you plan to use `sas` on `stats` server from off-campus, you will find the XWindows interface much too slow to be useable. Use batch mode instead.

Without XWindows emulation you can use SAS in batch mode. See Section III for instructions on using batch mode. Using batch mode you use any text editor to prepare your SAS code and to look at your output and log. Two text editors available on `stats` server are `pico` and `emacs`. `pico` is easier, `emacs` more powerful. In addition to not requiring XWindows, using batch mode relieves you of the need to navigate the `sas` interactive interface, which can be cumbersome.

The sections on interactive `sas` assume you are using XLiveCD (for Windows PCs) or X11 (for Macs) for XWindows. Other XWindows software may look and behave somewhat differently.

Working on Linux, the necessary X software is already available.

## II. sas syntax

Whether you use interactive or batch mode, you will need to type commands to tell sas what to do. First, your data must be in a form that SAS can read. For this example, assume you have collected information on students' exercise habits. The information includes their name, height, weight, and whether or not they jog or play tennis. Fruther, let's assume you have a text file called `exercise.dat`, with the following data:

```
Robert 68 150 yes yes
George 67 180 no no
Agatha 63 110 no no
Sandy 60 125 yes yes
Bill 65 160 yes no
```

**The data file must be on the same physical machine as sas.** If necessary, transfer your data to `stats` server. See the references on page 1 for information on file transfer between `stats` server and your PC or Macintosh.

Assume the file is stored in the subdirectory `saswork` of your home directory. The following sas program can be used to read this file into sas:

```
data exercise;
  infile '$HOME/saswork/exercise.dat' missover;
  input name $ height weight jog $ tennis $;
run;
```

### general rules

- every SAS command begins with a keyword and ends with a semi-colon.
- line breaks, extra spaces, indentations, etc. are for user readability only. SAS pays no attention to them.
- filenames are enclosed in apostrophes.
- upper/lower case is irrelevant in sas commands. They matter in filenames, data values stored in character (text) variables, and anything within quotes.

### data step

- the **DATA** statement begins a data step. It tells SAS to create a *new* SAS dataset and name it **exercise**
- the **INFILE** statement tells SAS to read the data from the file `$HOME/saswork/exercise.dat`
- the **MISSOEVER** option tells SAS not to try to fill in missing information by reading data from more than one data line.
- the **INPUT** statement names the variables in the data file, in the order in which they are to be read. A **\$** following a variable name indicates that the data for that variable is character – not a number.
- the **RUN** statement ends the DATA step.

## libnames

The above program creates a *temporary* sas dataset. It can be used for the duration of the sas session, but is discarded when sas stops. Of course if you save the program, you can easily recreate the temporary dataset whenever you need to work with it, by running the program again. This may be perfectly sufficient for small data files, or ones infrequently used.

For large data files, you may wish to save the data as a *permanent* sas dataset. Sas works much more efficiently when it can use a sas dataset directly, without first having to read an external file.

To create a permanent sas dataset, you need a `libname` command that tells sas where to store the data, AND you must use the defined library as a prefix for the newly created dataset. Here is a version of the above program, which saves the sas data in a permanent sas dataset called `exercise`, in the subdirectory `$HOME/saswork`:

```
libname ref '$HOME/saswork';
data ref.exercise;
  infile '$HOME/saswork/exercise.dat' missover;
  input name $ height weight jog $ tennis $;
run;
```

- the **LIBNAME** statement defines a sas library called "ref", that points sas to the `$HOME/saswork/` subdirectory. Note that the library is a subdirectory, not a specific file. A sas library can contain many files.
- The **DATA** statement now uses the above defined library, "ref", as the prefix for the sas filename. This tells sas to store the newly created file, `exercise`, in the subdirectory defined by "ref" - `$HOME/saswork/`
- The rest of the program is unchanged.

## procs

The above programs create a sas dataset, but do not do anything with it. `Procs` (short for Procedures) process sas datasets to produce output. Here are a few examples, using the permanent sas dataset created above. To use these with a temporary dataset, remove the prefix on the file specification. Without a prefix that refers to it, the `LIBNAME` statement is irrelevant.

Remember that if you are using a temporary sas dataset, the data step to create the data must be run in the same sas session as the proc that uses the data. When using a permanent sas dataset, you do not need a data step at all in your program. All you need is the `libname` that tells sas where to find the data, and the prefix on the filename:

```
libname ref '$HOME/saswork';
proc freq data=ref.exercise;
  tables jog tennis;
run;
proc means data=ref.exercise;
  var height weight;
run;
```

### III. SAS batch mode

Using SAS in batch mode can be faster than interactive, especially if you're connecting to `stats` server from off-campus. On Linux, use a Command window to type the commands to start `sas`.

#### input and output files

You will need to use a text editor, such as `pico` or `emacs`, to prepare a file with your SAS code. Save this file with the `.sas` extension. To run the previous example, which reads the data from a text file into a temporary `sas` dataset and runs `PROC FREQ`, use your favorite text editor to make a file with these commands:

```
data exercise;
  infile '$HOME/saswork/exercise.dat' missover;
  input name $ height weight jog $ tennis $;
run;
proc freq data=exercise; tables jog; run
```

This assumes that you have saved the data file shown at the beginning of the "SAS Syntax" section in file `$HOME/saswork/exercise.dat`. Save this syntax file with the name `exercise.sas`. At the unix prompt (`stats%`) or in your command window, type:

```
sas exercise
```

and press the **[Enter]** key

This command runs SAS using the statements in file `exercise.sas`. The `.sas` extension for the SAS syntax file is assumed, so we do not need to specify it. Upon completion of the job, SAS stores results in two files in the current directory

- `exercise.log` contains the SAS log output
- `exercise.lst` contains text output produced by procedures.

Use `pico` or `emacs` to view the files, or type

```
more filename [Enter]
```

to list a file on screen.

#### graphics output

The `.lst` output file is only for text output. If you run a procedure that produces high-resolution graphics you must provide a separate file to store the graphical output.

##### hi resolution plots

The `plot` subcommand of `PROC REG` creates a high-resolution plot. Here we use the `filename` command and the `GOPTIONS` command with the `gsfname` option to save postscript graphics output in file `'sasgraph.ps'`:

```
filename mygraph 'sasgraph.ps';
goptions reset=all device=ps gsfname=mygraph
          gsfmode=replace colors=(black);
proc reg;
  model weight=height;
  plot weight*height;
run;
```

The **FILENAME** statement defines the file where the graph is to be stored ('sasgraph.ps'), and an "alias", mygraph, that points to that file. In the **GOPTIONS** statement, the **DEVICE=ps** options requests the output formatted as postscript; **GSFNAME** specifies the alias of the file where the graph is to be stored - it must match the "alias" defined on the **FILENAME** statement. **GSFMODE=replace** means the output produced by *subsequent* graphics procedures will replace the contents of the file.

Postscript is a black and white device. Therefore we also specify colors=(black), to get a graph suitable for the device. For color postscript, use **DEVICE=pscolor**.

The **GOPTIONS** command affects all subsequent graphs, unless it is changed. Therefore, if you are creating more than one graph in SAS session, add the statement

```
GOPTIONS GSFMODE=append;
```

before the second graph so future graphs in the same session will be appended to the output file, rather than replacing each other. The **RESET=all** option clears any previously set graphics options. We do not use it here because we only want to change the **GSFMODE** option, leaving everything else as is.

For graphics that are to be inserted into a Microsoft Word document, use **DEVICE=CGMOF97P**.

To see a list of all available devices, run the following syntax:

```
PROC GDEVICE; RUN;
```

Select a device appropriate for your software/hardware.

### lineprinter plots

Some procedures let you choose a low-resolution line plot that can be written to the .lst output file. PROC REG has a `lineprinter` option to do this.

```
proc reg data=ref.exercise lineprinter;
  model weight=height;
  plot weight*height;
run;
```

If you request high-resolution graphics output and don't provide a file to store the graph, sas will prompt for an X device to display the graph. If one is not available, sas skips the graph.

## background mode

If you have a SAS job that takes a long time to complete, you can run it as a batch job in background mode. If you do this you can use your terminal to do other work (or log off) while SAS is running.

As with ordinary batch mode, you need to prepare a file (in this example the file is `exercise.sas`) with your SAS code. At the UNIX system prompt (`stats%`), or in the Command window, type:

```
nice sas exercise & [Enter]
```

- **nice** runs sas at a lower priority. It is optional, and only suggested for large jobs requiring a lot of computer resources.
- the **&** at the end puts the job in background mode.

When you submit a job in background mode, the only response is a number:

```
[1] 19682
```

This number (in this example 19682) is the `pid` (process id) of the job. You can do other work or even log off while the job is running. If you remain logged on, you will get a message on screen when the job is done. You can then look at the `.log` and `.lst` files to see the results of the analysis.

If you log off before the background job has finished, you will not get a message. To find out whether the job is still running log on to `stats server` and type:

```
ps -ux [Enter]
```

The `ps -ux` command lists all your currently active jobs. In the column labeled `pid`, look for the `pid` that was associated with your job when you ran it. If it is not on the list, the job is finished.

## IV. interactive mode - basics

To start an interactive SAS session, open an XWindows window (Command window on Linux) and type:

```
sas
```

A lot of SAS windows will open on your screen (SAS Toolbox, SAS Results, SAS Explorer, SAS Output, SAS Log and SAS Program Editor). Some windows may be obscured by others in front of them.

See <http://www.umass.edu/statdata/software/handouts/xwindows.html> for instructions on getting started with XWindow software on stats server.

To end your session, select **Exit** from the Program Manager's **File** menu. The **SAS:Session Management** window can be used to minimize, restore, interrupt and terminate your SAS session. It looks like this:



Using XLiveCD, you can select any of the SAS windows by selecting from the X icon on the Windows program bar at the bottom of your screen.

Using Mac X11, the SAS Session Management window is minimized. Some other windows are hidden behind by ones in front of them. You can move windows around to make the ones in the back visible.

## managing SAS windows

### using XLiveCD

The usual three Windows symbols are on the top right corner of every window: dash, box, and x. Use the dash to **minimize**, the box to **maximize** or restore, and the x to **close** a window. To **restore** a minimized window, select it from the list of X Windows at the bottom of the screen. **Resize** a window by moving the cursor to the edge of the window where it makes a double arrow. Drag to change the size. To **move** a window, drag it by its title bar. **Activate** a window and bring it to the front by clicking anywhere in it, or by selecting it from the list of X Windows at the bottom of the screen. If you accidentally close a window, **re-open** it by selecting it from the View menu of another window.

### using Mac X11

Use the red, yellow and green "lights" at the top left to close, minimize or maximize any window. To **restore** a minimized window, click its icon in the tray on the bottom of the screen. To **resize** a window, drag the lower left corner. To **move** a window, drag it by its title bar. **Activate** a window and bring it to the front by

clicking anywhere in it. If you accidentally close a window, **re-open** it by selecting it from the View menu of another window.

## Online Help

**SAS Help and Documentation** does not work from within SAS on `stats` server. You can get the identical online help by going to <http://support.sas.com/onlinedoc/913/> using your browser.

**Learning SAS Programming** is not available on `stats` server.

On **Linux**, SAS Help and Documentation should work from within SAS, provided you have loaded the appropriate browser, as instructed in your installation documentation. Learning SAS Programming starts SAS Tutor, which requires separate installation. The CDs for SAS Tutor are available from OIT's Administrative Desk, room A119 LGRC.

## V. SAS Interactive Environment

Interactive SAS provides windows for all basic tasks. When you first start SAS software, many SAS windows open: **Explorer**, **Results**, **Program Editor**, **Log**, and **Output**.



- In the Explorer window, you can view and manage your SAS files and create shortcuts to non-SAS files. Use this window to create new libraries and SAS

files; open SAS files; and do most file management tasks such as moving, copying, and deleting files.

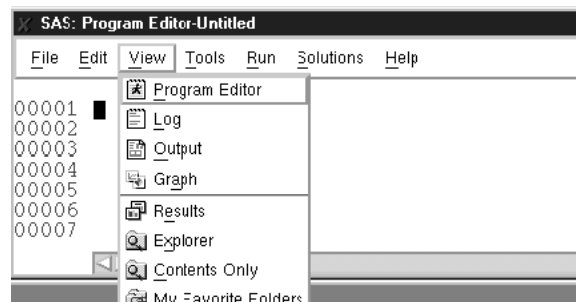
- In the `Program Editor` window, you enter, edit, and submit SAS programs. To open a saved SAS program file, click the `File` menu of the `Program Editor` and choose `Open`. Select the file to open.
- The `Log` window displays messages about your SAS session and any SAS programs you submit.
- In the `Output` window, you can browse output from SAS programs you submit. The `Output` window may be hidden behind the `Program Editor` and `Log` windows. You will need to move the other windows out of the way to see it.
- The `Results` window helps you navigate and manage output from SAS programs you submit. You can view or save individual items of output. The `Results` window may be hidden behind the `Explorer` window. You will need to move the `Explorer` window to see it.

## SAS environment commands

menus

SAS environment commands can be issued in one of three ways: using menus, the SAS command bar, or the `ToolBox`.

Pull down menus are located at the top of every window. The choices in the menus change according to which window is active. This is the `View` menu on the `Program Editor Window`:



SAS command bar and toolbar

This is the SAS `ToolBox`. It contains a command bar and the toolbar.



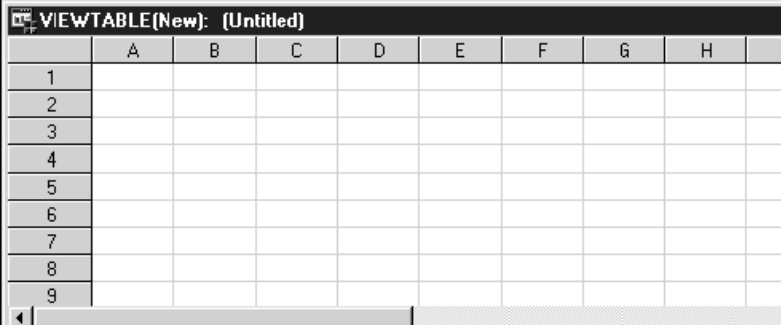
You can type SAS environment commands in the command bar on the left side of the `ToolBox`. For example, typing **output** in the command bar brings the `Output` window to the foreground. Many of the commands that you can type in the command bar are also available on the pull-down menus or the toolbar. The `ToolBox` is the set of icons on the right side of the `ToolBox`.

## VI. sample SAS session – interactive mode

We will use the same exercise data as before, but this time type it directly into sas.

### enter data using Viewtable

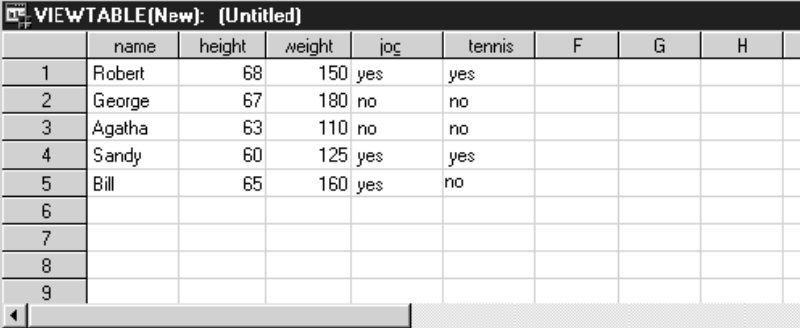
In addition to the five windows shown above, there are many other SAS windows available for various tasks. The `Viewtable` window is an easy way to create new datasets or browse and edit existing datasets. The `Viewtable` window displays SAS datasets in a tabular format. To open the `Viewtable` window, select **Table Editor** from the **Tools** menu.



	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									

The `Viewtable` is NOT a spreadsheet. The rows represent observations and are labeled with numbers; the columns represent variables, labeled with letters.

We want the variables to have more meaningful names than A,B,C... Click on the column heading A and type the word name. Tab to column heading B and type height. Continue until you have named the first five columns as shown below. Then type the data into the table. SAS will assign each variable as numeric or character based on the first row of data.



	name	height	weight	jog	tennis	F	G	H	I
1	Robert	68	150	yes	yes				
2	George	67	180	no	no				
3	Agatha	63	110	no	no				
4	Sandy	60	125	yes	yes				
5	Bill	65	160	yes	no				
6									
7									
8									
9									

## save your data

### temporary

When you have typed in your data and are satisfied with its accuracy, you are ready to save your data. From the `Viewtable`'s **File** menu, select **Save As**. From the left panel of the `Save As` dialog box, select the **Work** library. Type a name for your data into the `member name` box and click **Save**.

Libraries are directories (folders). Members are files. There are always at least four libraries created and controlled by SAS: `Maps`, `Sashelp`, `Sasuser` and `Work`. There may also be a `Gismaps` library. **Work** is the only one of these you should use for your datasets. **Work** is a temporary library or workspace that exists for the duration of your SAS session but is discarded at the end of each session.

### permanent

Because the `Work` library is discarded at the end of each session, any data you save there is temporary. If you wish to create a permanent SAS data set, you need to first define a library to store it.

#### define a new library

Libraries can be defined using the `LIBNAME` command introduced under the "SAS Syntax" section. Alternatively, they can be defined using the SAS interactive interface.

In the `Save As` dialog, click on the `Create New Library` icon (it looks like a file drawer). In the `New Library` dialog box, type a name for the new library. Library names may be no more than 8 characters, must begin with a letter, and may not contain any special characters – to be safe, use letters and numbers only. Use the `Browse` button to select the subdirectory (folder) where you want to save your data. If you prefer, you can type the subdirectory name directly into the `Path` box. If you type your path you can use relative or absolute paths, as well as unix "shortcuts", e.g. `$HOME` and `~`. Click **OK**.

For our example, define a library named **ref** with the path `$HOME/saswork` (where **saswork** is a directory that already exists in your root directory).

#### save your data set

When you have defined your SAS library location, you can use it to save or retrieve SAS datafiles.

1. In the `Save As` dialog box list of libraries on the left panel, select the name you used for your new library. In this example, select library **ref**.
2. In the `Member Name` box: enter **exercise**
3. Click **Save**.

This saves the SAS dataset in the Viewtable window to a file named **exercise.sas7bdat** in the directory **saswork** under your main directory. SAS assigns the extension **.sas7bdat** to all SAS permanent datasets. Although the UNIX name for the SAS dataset is **exercise.sas7bdat**, within SAS programs you must refer to datasets using the library name with only the first part of the filename, for example, **ref.exercise**.

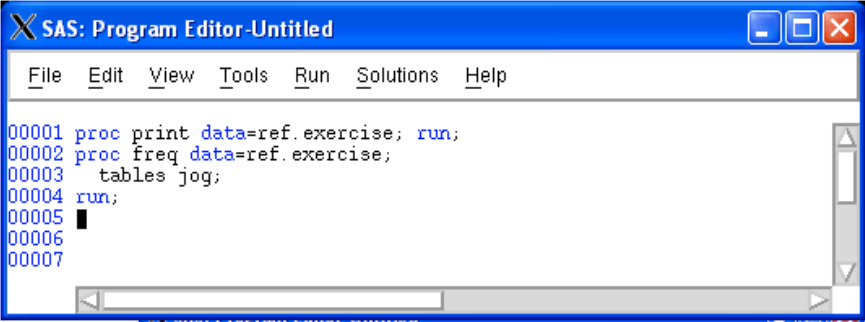
*Notes:*

- the **library** is the name of the **directory** where you store your permanent SAS datasets. It is **NOT** a specific file. The **member** name specifies a particular file.
- The extension **.sas7bdat** is always understood, and must not be stated explicitly.

## procs

Once you have a SAS dataset, you can use procedures (PROCS) to work with the data. PROC PRINT is used to display the data on screen, and PROC FREQ to get frequency count of selected variables. In the example below, we show how to use a permanent SAS dataset:

- The library 'ref' was defined earlier in the sas session.
- The procs specify the data to use by its two-part name, **ref.exercise**, where **ref** is the sas library and **exercise** is the sas dataset name.
- The file extension **.sas7bdat** is not stated.



```
SAS: Program Editor-Untitled
File Edit View Tools Run Solutions Help
00001 proc print data=ref.exercise; run;
00002 proc freq data=ref.exercise;
00003     tables jog;
00004 run;
00005 █
00006
00007
```

## editing in the Program Editor

### point and click

When you are entering SAS programming statements into the Program Editor Window, you can:

- type over or insert text in the editor window. A block shaped cursor indicates that the editor is in 'replace' mode – any text you type replaces existing text. To toggle between 'replace' and 'insert' modes, press **ctrl-x**. The cursor changes to an underline. (With XLiveCD or on Linux you can use the Insert key to toggle between 'replace' and 'insert'.)
- copy, cut and paste selected text in the Program Editor Window.

### selecting text

1. click the pointer at the beginning of the section you wish to select;
2. from the Edit menu click **Select**;
3. click the pointer at the end of the section to be selected. The section is now selected.

Choose Cut or Copy and Paste on the Edit menu to manipulate the selected text, or Submit to run the selected lines of code. See section "Submit a SAS Program" for more about submitting code.

### ending text selection

When you select text, selection mode stays in effect as long as any text is highlighted. You may need to end text selection mode before you can use other commands, or start a new selection. On the Edit menu click **Deselect**. All highlighting should turn off.

## line editing commands

Line editing commands can also be used to insert, delete, copy, or move lines or blocks of lines. Line editing commands are typed over the line number(s) of the line(s) you want to apply the command to, and must be followed by pressing **[Enter]**

**D[n]** Delete one or more lines. Type **D** over the line number of the line to be deleted, and press **[Enter]**. To delete more than one line, follow **D** with the number of lines to delete - e.g. **d5** deletes 5 lines.

**DD** Delete a block of lines. Type **DD** over the line numbers of the first and last lines to be deleted, and press **[Enter]**.

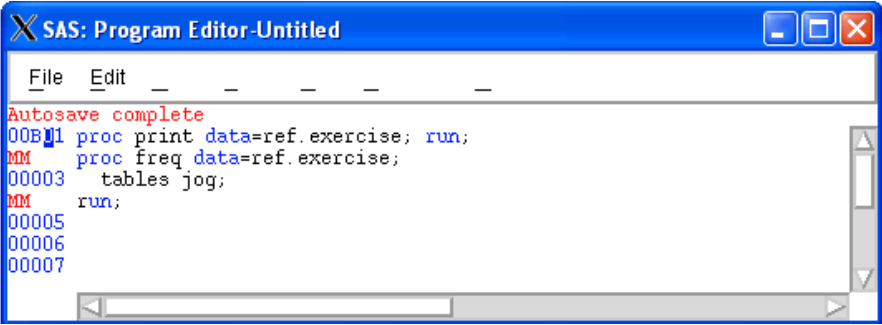
**I[n]** Insert one or more new lines after the one on which you type **I**. To insert more than one line, type **I** followed by the number of lines to insert – e.g. **i3** - and press **[Enter]**.

**IB[n]** Insert one or more lines before the line on which you type the **IB** (insert before). To insert more than one line, type **IB** followed by the number of lines to insert – e.g. **ib3** - and press **[Enter]**.

**MM** Move a block of lines. Type **MM** over the line numbers of the first and last lines to be moved. Type **A** (for **A**fter) or **B** (for **B**efore) over the number of the line where the block is to be moved. Press **[Enter]**.

**CC** Copy a block of lines. Type **CC** over the line numbers of the first and last lines to be copied. Type **A** (for **A**fter) or **B** (for **B**efore) over the number of the line where the block is to be pasted. Press **[Enter]**.

To move the PROC FREQ before the PROC PRINT, here is how the Program Editor would look just before you press **[Enter]**.



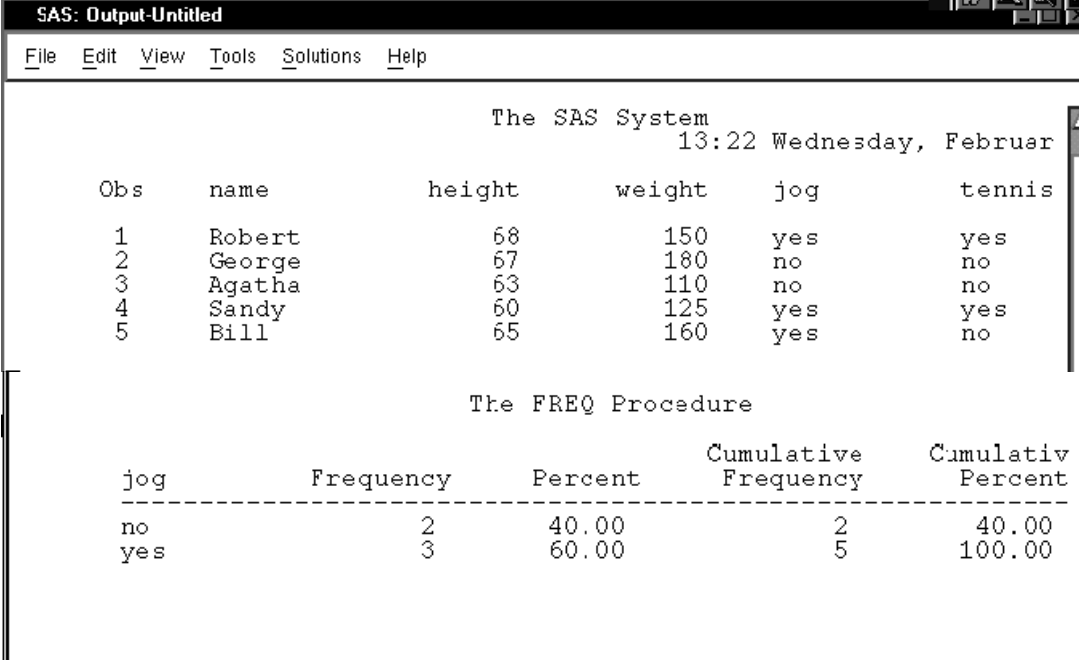
```
SAS: Program Editor-Untitled
File Edit _ _ _ _ _
Autosave complete
00001 proc print data=ref.exercise; run;
MM   proc freq data=ref.exercise;
00003   tables jog;
MM   run;
00005
00006
00007
```

## submit a SAS program

entire program

To process the code in the Program Editor, from the Program Editor's **Run** menu, select **Submit**.

If there are no errors in the code, the results appear in the Output window:



The SAS System  
13:22 Wednesday, February

Obs	name	height	weight	jog	tennis
1	Robert	58	150	yes	yes
2	George	57	180	no	no
3	Agatha	53	110	no	no
4	Sandy	60	125	yes	yes
5	Bill	55	160	yes	no

The FREQ Procedure

jog	Frequency	Percent	Cumulative Frequency	Cumulative Percent
no	2	40.00	2	40.00
yes	3	60.00	5	100.00

selected lines

If you only need to process a portion of the code in the Program Editor, select the text to be processed, then choose **Submit**. See section "editing in the Program Editor" for how to select and deselect text.

## recall a SAS program

When you submit the entire program, it is no longer in the Program Editor window. To retrieve it, select **Run** → **Recall Last Submit**.

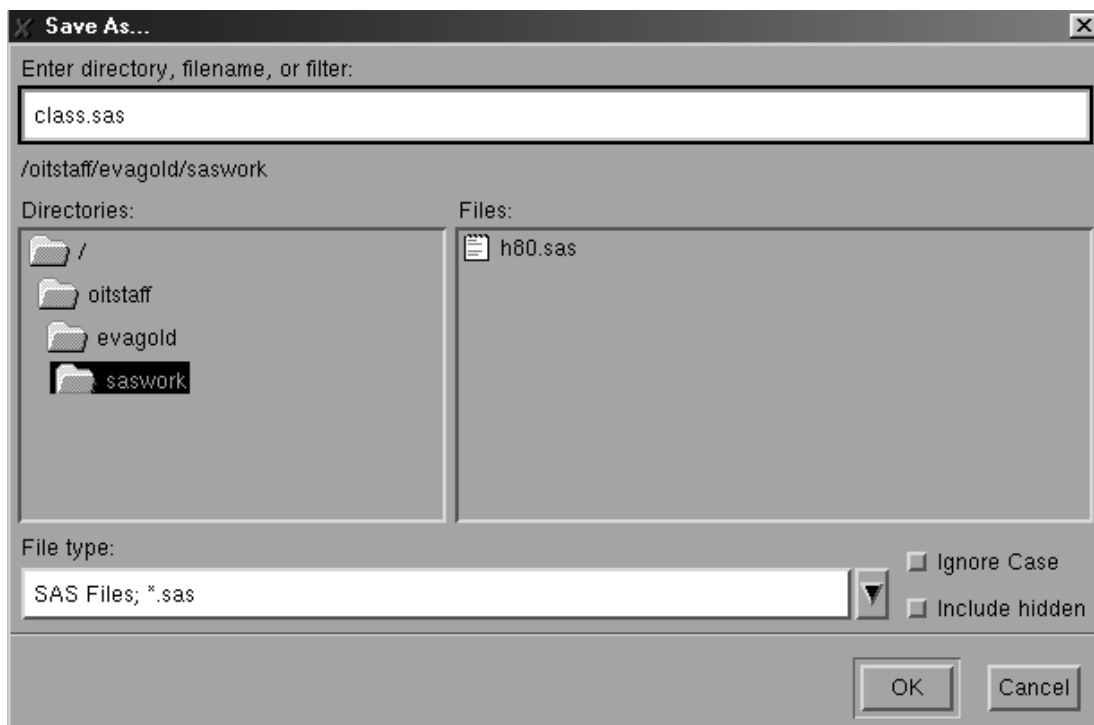
Submitting a selected portion of the program does not clear the Program Editor window.

## VII. more about working with SAS

### save the contents of a window

You can save the contents of a window as a text (ascii) file. Use this if you need to print a file from *stats* server. After saving the contents of a window, you can download it for printing from your local machine.

1. from the **File** menu of the window you wish to save, select **Save As**.
2. in the Save As dialog box, select the directory where you want the file saved.
3. Type a name for the file in the text box at the top of the Save As dialog.



Here we save the contents of the Program Editor Window as file **class.sas** in the directory: **/oitstaff/evagold/saswork/**

Note that the extension **.sas** is used for Program Editor files; **.log** for Log window files; **.lst** for Output window files. Although it is possible to use other extensions, things will go more smoothly if you honor these conventions.

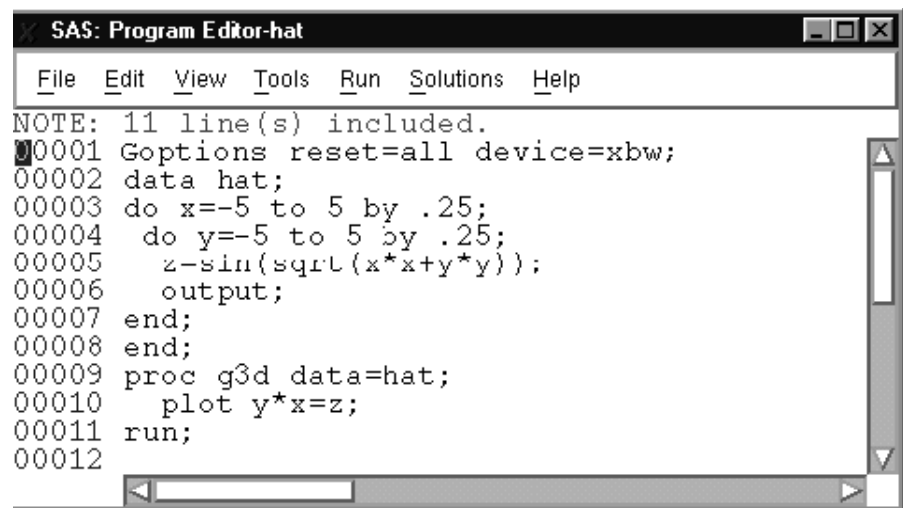
## preview and save graphics

Using SAS interactively, you can preview graphics output on your screen and save the results for viewing or printing later. Graphics generated by any of the graphics procedures appears in a separate window.

The GOPTIONS command preceding any graphics procedure controls many features of the graph. The DEVICE and TARGET options control the destination of the graph (screen or file), and its resolution. The default is color screen graphics. If you print this on a black and white or grey-scale printer, the colors on the printed copy will likely be indistinguishable. To generate black and white screen graphics more suitable for printing, use the command:

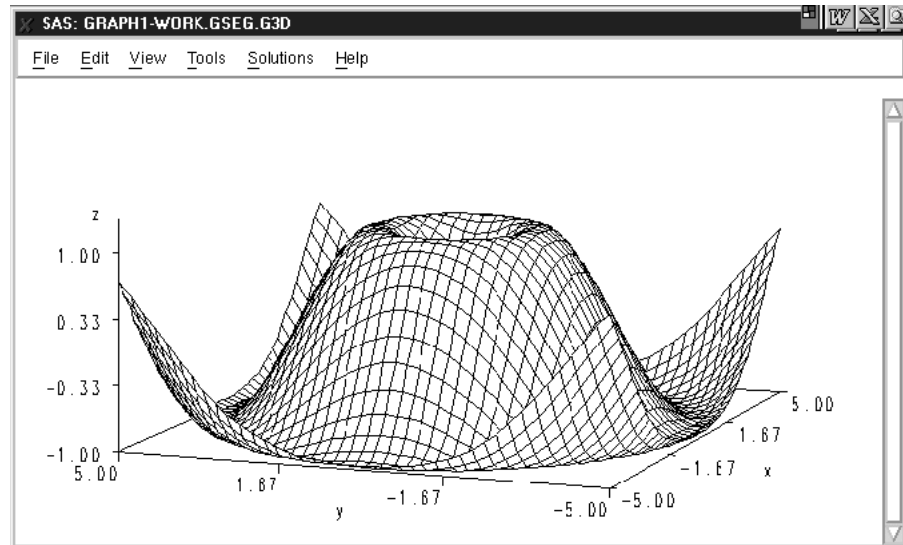
```
options reset=all colors=(black);
```

This SAS program uses proc g3d to make a black and white cowboy hat graphic. The xbw device is black and white screen output.



```
SAS: Program Editor-hat
File Edit View Tools Run Solutions Help
NOTE: 11 line(s) included.
00001 Goptions reset=all device=xbw;
00002 data hat;
00003 do x=-5 to 5 by .25;
00004   do y=-5 to 5 by .25;
00005     z=sin(sqrt(x*x+y*y));
00006     output;
00007   end;
00008 end;
00009 proc g3d data=hat;
00010   plot y*x=z;
00011 run;
00012
```

Here is the screen output from this program:



Screen graphics such as this do not produce high quality printed copy. To get graphics suitable for printing and still be able to preview your graph on screen, use the **TARGET** option on the **GOPTIONS** statement:

```
options reset=all colors=(black) target=ps;
```

This produces a graph formatted for printing on a postscript (ps) printer and displays it on screen. To save the image, from the **File** menu of the graphics window select **Export Image**. Select the image format (e.g. .gif, ps, ..) and enter a filename and location. If you have multiple images in the Graphics window, only the one visible when you do this is Exported.

## save graphics output in a file

To save graphics output directly to a file, use a **FILENAME** statement to specify the file to use, and the **GSFNAME** option on the **GOPTIONS** statement to write to that file:

```
FILENAME gref 'test.ps';  
GOPTIONS RESET=all DEVICE=ps GSFMODE=replace  
GSFNAME=gref COLORS=(black);
```

The various options and how they interact are explained under "**SAS batch mode**".

## VIII. printing sas output

There is no printer available on `stats` server. Any files to be printed must be saved and downloaded to your local machine for printing.

### sas output (\*.lst) files

Save your sas output files from the output window, as instructed under "save the contents of a window". If you use sas in batch mode, your output is saved automatically. Using WinSCP, Fugu, or Fetch, download the file to your local machine. For Windows, be sure to select "text" file transfer, rather than "binary" (or "automatic"). This will properly convert it to the format that Windows can use. With Mac OS X, automatic file transfer should work. (If not, Fetch has features to control the type of transfer. Fugu does not.) SAS output is an ordinary text file, which you can open and print with any text editor on your local machine.

### sas graphics

Save your sas graphs as shown under "SAS batch mode, graphics output" section. Download it to your PC or Mac using binary file transfer. What you do with it there will depend on what DEVICE you selected when you created the image. Files created using `DEVICE=ps`, can be opened with Adobe Distiller, which will convert it to pdf format, to be printed using Acrobat. Files created using `DEVICE=CGMOF97P` can be inserted in Microsoft Word documents.

## IX. moving SAS files between computers

The way SAS stores data files (permanent datasets and catalogs) depends on both the version of SAS used to create the dataset/catalog and the operating system on which it is running. In general, a SAS dataset or catalog created on one operating system cannot necessarily be read directly on another operating system.

If you need to move SAS data between computers, be sure to find out what version of SAS and what operating system was used to create the SAS data and what version and operating system will be used to read it. SAS files created using SAS Version 8 or 9 on Windows and Version 8 on OITUNIX (now defunct) can be moved between a Windows PC and `stats` server without special processing. Use binary FTP to move the files.

See "[Transporting SAS Libraries](http://www.umass.edu/statdata/software/)" at <http://www.umass.edu/statdata/software/> for information on moving SAS files between versions and operating systems.