

SAS for Windows

These notes give a brief guide to getting started with **SAS for Windows, version 8**.

SAS is a powerful statistics package covering a very wide range of statistical methods. However, it is quite demanding in terms of resources; even the simplest program will require around 400,000 bytes of free space on the M:/ directory.

- **To start up SAS**

From the **WINDOWS XP DESKTOP**, choose

Start/Programs/Statistics, then *SAS 8.2*

Be patient while SAS loads.

The opening display shows **LOG** and program **EDITOR** windows.

- EDITOR window - allows you to write, edit and submit SAS programs.
- LOG window - displays messages (including error messages) about the execution of the program.

Once you run some procedures, SAS will open up the OUTPUT and RESULTS windows:

- OUTPUT window - displays the output from your program.
- RESULTS window - summarises the information in the OUTPUT window, assisting in navigation.

There is also an EXPLORER window - another navigation tool used for managing certain advanced tasks.

To move between the different windows

Select the window and click to make it active

or from the menu, click

WINDOW

and select your chosen window from the list.

Menu Options

Some of the menu options available depend upon the active window. These are the most frequently used.

- FILE - allows programs to be read from or saved to a file, and printed. Also contains the command to exit SAS.
- EDIT - usual options for cutting, pasting and copying text and those for finding and replacing text. Clear All option empties the window.
- VIEW - options for opening each of the main windows if any have been closed.
- RUN - options specific to the program editor window. The program currently held in the program editor window is run by choosing the Submit option. Also here is Recall last submit.

LOG and OUTPUT windows do not permit their contents to be edited. When either of these windows is active, the FILE and EDIT menus have several options disabled.

- WINDOW - allows user to move between open windows. Shows only the windows that are open.
- HELP - extensive help system with Online training and Sample programs.

A Simple SAS Program

SAS programs consist of a DATA step followed by one or more procedure (PROC) steps. They are created in the EDITOR window.

```
e.g. DATA A1;
      INPUT X Y;
      CARDS;
      15 95
      26 71
      10 83
      9 71
      15 102
      20 87
      ;
      PROC PRINT;
      PROC REG;
      MODEL Y = X;
      RUN;
```

Note: Each statement (except data lines) ends with a semicolon.

DATA step - Data are read and a SAS data set created. The SAS data set may be named, e.g. DATA A1.

The INPUT line supplies names for the *variables* (can be up to 8 characters long and must not start with a number). Variables can be numeric or character.

Character variables require a dollar sign after the variable name on the INPUT line (but nowhere else in the SAS program).

e.g. INPUT SEX \$ AGE;
where SEX takes values M or F and AGE is numeric.

PROC steps - give the statements requesting particular statistical analyses.

A final RUN statement must be included to process execution of the program.

To Run the SAS Program

From the menus choose

RUN/SUBMIT

or click on the “RUN” button.

Choose WINDOW/EDITOR (or simply close the Output window) to return to the Editor screen.

The RESULTS window on the left lists the various sections of output from the different procedures.

After Submitting

In the LOG window (not automatically displayed):

- a list of SAS statements contained in the program
- messages generated by SAS including notes, warnings and error messages.

In the OUTPUT window:

- output produced by the SAS procedures.

Output can be saved to a listings file * .LST by choosing from the menu

FILE/SAVE AS

Tidying Up

With the OUTPUT window active

EDIT/CLEAR ALL

clears out the OUTPUT window.

Output is added to the OUTPUT window each time a program is submitted.

Deleting Unwanted Sections of Output

Highlight a procedure within the RESULTS window and right click and choose DELETE (or press Delete key).

Printing the Output File

Under Windows XP, the printer is selected within SAS.

With the output window active, choose from the menu

`FILE/PRINT PREVIEW` to check how many pages you are about to print,
then `PRINT`

A selection of printers will be displayed: choose a convenient one.

Since the contents of the `OUTPUT` window isn't directly editable, it may be more convenient to print from another application:

Copy the `SAS` selected text to the clipboard by choosing

`EDIT/COPY`

In your chosen application, select `EDIT/PASTE`.

More About the `DATA` Step

Instead of the data step including lines giving the actual data values, data may be contained in an external file.

```
e.g. DATA A1;
      INFILE 'a:\ex1.dat'; - assumes data are in file ex1.dat on the A: drive.
      INPUT X Y;
      PROC PRINT;
      PROC REG;
      MODEL Y = X;
      RUN;
```

Data may often be supplied in a library file e.g. `INFILE 'n:\userlib\stalib3\sas\ ex1.dat'`;

In order to use a library file from the N: drive, the library must first be activated.

Double-click the "Departmental Software" icon on the Windows XP desktop, then select Statistics, then stalib3. You can then close both the window listing the files in stalib3 and the Departmental Software box.

More About the `PROC` Step

By default, a `PROC` statement such as `PROC PRINT` or `PROC REG` operates on the most recently created SAS data set. When more than one SAS data set is used in a program, it is useful to specify the data set to be processed

```
e.g. PROC PRINT DATA = A1;
```

With many SAS procedures it is possible to specify the variables to be analysed in a VAR statement

e.g. PROC MEANS; gives summary statistics such as mean, standard deviation etc.
VAR variables; The information is given for each variable listed in the VAR statement.

The DATA Step - further details

1. More than one set of data values per line

Normally SAS reads one set of data values (e.g. X,Y pair of values) at a time, and expects the next set to be found on the next line.

To read the same data line in more than one execution of the data step, type @@ on the input line.

e.g. INPUT GROUP \$ WTGAIN @@;

would expect to find data on two variables, the first being a character variable, with more than one pair of values per line, such as

```
A 20 A 16 B 13 B 19 ...
```

2. Program statements within the DATA step

Data transformations are often performed by placing "PROGRAM STATEMENTS" into the data step, immediately following the INPUT statement.

e.g. INPUT DRUG PLACEBO;
DIFF = DRUG - PLACEBO; the program statement creates
a new variable DIFF

Other examples:

```
INPUT X Y; INPUT X Y;  
Z = LOG(Y); Z = SQRT(Y);
```

3. Analysing a subset of the data

A subsetting IF statement is used

e.g. DATA ALL;
INFILE BODY;
INPUT SEX \$ HEIGHT WEIGHT; (SEX is a character
variable taking values M or F)

DATA FEM;
SET ALL;
IF SEX = 'F';

The statement IF SEX = 'F' results in a data set containing only observations on the females.

Note: The value of a *character* variable has to be enclosed in single quotes.

With many procedures separate analyses for each subset of the data (e.g. both males and females) may be obtained by using the BY statement.

When using a BY statement, the data must first be sorted in the order of the grouping variable, e.g.

```
PROC SORT;
      BY GROUP;           Provides information on the
PROC MEANS;              current data set, separately
      BY GROUP;          for each GROUP.
```

4. Keeping only certain variables

```
KEEP variable(s);
```

within the data step specifies which variables are to be included in any SAS data set being created.

Alternatively

```
DROP variable(s);
```

can be used to drop unwanted variables.

5. Conditional IF THEN/ELSE statements

A conditional IF statement with an associated THEN clause provides a way of getting SAS to execute a statement only for observations that meet a specified condition

e.g. IF $Y \geq 20$ THEN DELETE;

This tells SAS not to add the observation to the data set being created if $Y \geq 20$.

IF conditions may involve several comparisons joined by AND's and OR's.

New variables may also be created within an IF statement. An optional ELSE statement may also be present to specify an alternative action when the condition of the previous IF statement is false.

e.g. To define groups for a frequency distribution the following statements may follow the INPUT statement:

```
IF  $Y \leq 5$  THEN GROUP = 1;
ELSE IF  $5 < Y \leq 15$  THEN GROUP = 2;
ELSE IF  $Y > 15$  THEN GROUP = 3;
```

Graphics

SAS includes a series of procedures for producing high resolution graphics.

e.g.

```
PROC GPLOT;  
      PLOT Y * X;
```

produces a high resolution scatter plot of variable Y vs variable X . By default, the plotting symbol is $+$. This can, however, be altered using a `SYMBOL` statement

e.g.

```
SYMBOL INTERPOL = RL VALUE = DOT;  
PROC GPLOT;  
      PLOT Y * X;
```

requests a dot for the plotting symbol, and also requests display of the fitted linear regression line.

Note: Graphs are displayed in a separate Graphics window.

Printing Graphics

- Directly from SAS

From the menu options choose
`FILE/PRINT`

- From other applications

- (i) Copy the SAS graph to the clipboard by choosing
`EDIT/COPY`

In your chosen application select
`EDIT/PASTE SPECIAL...`

Paste as Device Independent Bitmap.

- (ii) Export the SAS graph using
`FILE/EXPORT....`

In your chosen application select
`INSERT/PICTURE...`