INTRODUCTION TO BASIC BUT VERY USEFUL MACROS

Susan Godwin, Family Health International

Macros are relatively simple and very useful for everyday applications. Using macros can be a complicated endeavor, but does not have to be. Macros reduce the amount of text that you must type to perform programming tasks, so they save you a tremendous amount of time. This poster assumes that you have a good basic knowledge of SAS and introduces macros and presents two examples of their uses which illustrate that the time taken to write these macros was well spent.

In this example one macro produces a contingency table based on two different pairs of variables.

```
1 proc format;
2  value $dsname 'majsign'= 'pe.ssd'
3   'mainsign'='pe.ssd'
4   'malegud'='fu.ssd'
5   'chbreak'='fr.ssd'
6   'displ8'='sp.ssd';
7 run;
8 %macro doit (passvar1, passvar2);
9   data _null_
10    call symput('footvar', put('&passvar1', $dsname.));
11   run;
12 proc sort data=seg.ever out=work.ever;
13   by &passvar1 &passvar2;
14   run;
15 proc freq data=work.ever order=data;
16   tables &passvar1 * &passvar2/measure=
17     footnote '&footvar';
18 run;
19 %mend doit;
20 %doit (majsign, dicondl);
21 %doit (mainsign, dicondl);
22 %doit (malegud, dicondl);
23 %doit (mainsign, dispemrl);
24 %doit (malegud, dispemrl);
25 run;
```

In line 9 %MACRO begins the macro. DOIT is the name of the macro and PASSVAR1 and PASSVAR2 are the macro variables being defined. The CALL SYMPUT statement in line 11 can only be used in a data step. This statement produces the macro variable FOOTVAR and assigns the appropriate DSNAME format. In lines 14-16 the SEG.EVER dataset is sorted by the macro variables PASSVAR1 and PASSVAR2. In line 18 &PASSVAR1 and &PASSVAR2 invoke the macro variables, which insert the values of PASSVAR1 and PASSVAR2 in place of the macro variables. In line 19 &FOOTVAR resolves to the value of the FOOTVAR macro variable. The value of FOOTVAR is read in as the footnote. In line 21 %MEND tells SAS that this is the end of the macro. In lines 23-29 %DOIT invokes the DOIT macro and gives the values of the macro variables PASSVAR1 and PASSVAR2.

This example produces plots on a chart when specific conditions are met.

```
1 data work.ever;
2 set seg.ever;
3 where sex='f';
4 run;
5 %macro plot(yvar, xvar);
6 proc plot;
7 plot &yvar*&xvar;
8 run;
9 %mend plot;
10 %plot (majsign, dicondl);
11 %plot (mainsign, dicondl);
12 %plot (malegud, dicondl);
13 %plot (mainsign, dispemrl);
14 %plot (malegud, dispemrl);
15 %plot (mainsign, dispemrl);
16 %plot (malegud, dispemrl);
17 run;
```

In line 5 %MACRO begins the macro. PLOT is the name of the macro and YVAR and XVAR are the macro variables being defined. In line 7 &YVAR and &XVAR invoke the YVAR and XVAR macro variables. In line 9 %MEND tells SAS that this is the end of the PLOT macro. In lines 11-16 %PLOT invokes the macro and gives the values of the macro variables YVAR and XVAR.

The output of these and other examples can be found in the handout accompanying this presentation.