Biostatistics, Epidemiology & Research Design (BERD)

Howard Cabral, PhD, MPH
Christine Chaisson, MPH

Seminar Series: CTSI Presents…

November 20, 2014

BOSTON UNIVERSITY

Demystifying SAS Macros

BUSPH Data Coordinating Center
Leah Forman, MPH
Clara Chen, MHS
Michael Winter, MPH

November 20, 2014

BOSTON UNIVERSITY

Macro Basics: an introduction

- Leah Forman
- Statistical Data Analyst
  - Data Coordinating Center

- SAS Macro language:
  - References previously written SAS code and commands it to run

- Why use it?
  - To repeat code, substitute text, pass values . .

Macro Structure – defining the macro

```
%macro examplemacro(one=, two=); /*Defines the Macro*/
     /*This macro is called "examplemacro"*/

data &one; /*Code that defines what happens when the macro is used*/
       set &two;
       if two = 3 then four = 5;
run;

%mend examplemacro; /*Ends the macro*/
```

Macro structure – executing the macro

```
data &one;
        set &two;
        if two = 3 then four = 5;
run;
```

- `%examplemacro(one=dset, two=var2); /*This calls the macro and tells it to execute*/`

- When the macro runs, it's as if you are running this code:

```
data dset;
     set var2;
     if two = 3 then four = 5;
 run;
```

Keyword parameters

- Enter values for keyword parameters **in any order**

```
%macro keywords (dsname=, varlist=);
      proc freq data=&dsname;
      tables &varlist;
run;
%mend keywords;

%keywords(dsname=one, varlist=studyid gender age)
 %keywords(varlist=studyid gender age, dsname=one)
```

Keyword Parameters cont'd

- Can assign default values:

```
%macro keywords (dsname=one, varlist=);

proc freq data=&dsname;
      tables &varlist;
run;

%mend keywords;

%keywords(varlist=studyid gender age)
%keywords(dsname=two,varlist=studyid gender age)
```

Passing values with “callsymputx”

- Assign a calculated variable to a macro variable
- Creates a macro variable in the datastep

```
proc means data=sug.shoe_vendors noprint;
        var Mfg_Suggested_Retail_Price;
        output out = outdat mean=MeanPrice;
run;
```

The output dataset, “outdat”:

Obs	_TYPE_	_FREQ_	MeanPrice
1	0	361	$155.32

```
data _null_;
       set outdat;
       call symputx ('MeanPriceMac', MeanPrice);
run;


data lowmean;
       set sug.shoe_vendors;
       where Mfg_Suggested_Retail_Price<&MeanPriceMac;

run;
```

Debugging: %put

- Lets you see the value of your macro variable.
- Recall this example:

```
proc means data=sug.shoe_vendors noprint;
        var Mfg_Suggested_Retail_Price;
        output out = outdat mean=MeanPrice;
run;

data _null_;
        set outdat;
        call symputx ('MeanPriceMac', MeanPrice);
run;
```

- **%put mean of Price is &MeanPriceMac**

%put mean of Price is &MeanPriceMac;
mean of Price is 155.31855956 ← what appears in the log

Debugging: mprint, symbolgen

```
13  %macro meansy (varname=);
14      proc means data=microbiology;
15      var &varname;
16  run;
17  %mend meansy;
18
19  %meansy(varname=TBSpecNum)
```

```
NOTE: Writing HTML Body file: sashtml.htm
NOTE: There were 2745 observations read from the data set
WORK.MICROBIOLOGY.
NOTE: PROCEDURE MEANS used (Total process time):
      real time          3.48 seconds
```

Debugging: mprint, symbolgen

```
28   options mprint symbolgen;
29   %macro meansy (varname=);
30       proc means data=microbiology;
31       var &varname;
32   run;
33   %mend meansy;
34%meansy(varname=TBSpecNum)
MPRINT(MEANSY):   proc means data=microbiology;
SYMBOLGEN:  Macro variable VARNAME resolves to TBSpecNum
MPRINT(MEANSY):   var TBSpecNum;
MPRINT(MEANSY):   run;

NOTE: There were 2745 observations read from the data set WORK.MICROBIOLOGY.
NOTE: PROCEDURE MEANS used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```

Global vs. Local Macro Variables

- Global:
  - Exists for the rest of the SAS session
  - %LET statement (outside of a macro definition)
  - %GLOBAL statement
  - CALL SYMPUTX
- Local:
  - Exists only during execution of the macro in which it is created
  - %LET statement (inside a macro definition)
  - %LOCAL statement (inside a macro definition)
  - Macro parameters

Global vs. local: exercise

- Which Macros are global, and which are local?

```
%let presenter=Leah;
%macro groupexercise (type= );
       %let day=Thanksgiving;
       title “Turkey eaten on &day";
       title2 "presentation by &presenter";
       proc means data=turkeyday;
       var  &type;
       %global type2;
       %let type2=&type;
run;
%mend groupexercise;
```

- %***groupexercise***(type=reeses hersheys)

Which are local, and which are global?

- To find out, you can use %put and they will print out in the log:

- %put presenter = &presenter type = &type day=&day type2=&type2 ;

Log - (Untitled)

```
39   %put presenter = &presenter type = &type day=&day type2=&type2
SYMBOLGEN:  Macro variable PRESENTER resolves to Leah
WARNING: Apparent symbolic reference TYPE not resolved.
WARNING: Apparent symbolic reference DAY not resolved.
SYMBOLGEN:  Macro variable TYPE2 resolves to reeses hersheys
presenter = Leah type = &type day=&day type2=reeses hersheys
```

Building a Macro: A Real World Example

- Clara Chen
- Assistant Director of Operations, Data Coordinating Center

Constructing a macro: output N(%)

Step 1: Use example and examine ODS tables to identify locations of needed information

```
proc sort data=BWeight;by Married;run;
ods trace on; /*Writes ODS Table Info To Log*/
proc freq data=BWeight ;
where  ;
by Married ;
     tables black /missing;
run;
ods trace off;
```

SAS log

```
29    ods trace on; /*Writes ODS Table Info To Log*/
30    proc freq data=BWeight ;
31    where   ;
32    by Married ;
33          tables black /missing;
34    run;

NOTE: Writing HTML Body file: sashtml.htm

Output Added:
-------------
Name:         OneWayFreqs
Label:        One-Way Frequencies
Template:     Base.Freq.OneWayFreqs
Path:         Freq.ByGroup1.Table1.OneWayFreqs
-------------
NOTE: The above message was for the following BY group:
      Married=[0]No

Output Added:
-------------
Name:         OneWayFreqs
Label:        One-Way Frequencies
Template:     Base.Freq.OneWayFreqs
Path:         Freq.ByGroup2.Table1.OneWayFreqs
-------------
NOTE: PROCEDURE FREQ used (Total process time):
      real time           3.51 seconds
      cpu time            0.20 seconds

NOTE: The above message was for the following BY group:
      Married=[1]Yes
NOTE: There were 50000 observations read from the data set WORK.BWEIGHT.

35    ods trace off;
```

Save output using ODS OUTPUT

Step 2: Write ods tables to a temporary set for further manipulation via ods output statement

```
proc freq data=BWeight ;
where  ;
by Married ;
      tables black /missing;
      ods output onewayfreqs=blackF;
run;
```

blackF dataset looks like this:

	Married	Table	black	Black Race	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	[0]No	Table black	[0]No	[0]No	9053	63.00	9053	63.00
2	[0]No	Table black	[1]Yes	[1]Yes	5316	37.00	14369	100.00
3	[1]Yes	Table black	[0]No	[0]No	32805	92.07	32805	92.07
4	[1]Yes	Table black	[1]Yes	[1]Yes	2826	7.93	35631	100.00

Create summary variables

Step 3: Create summary variables to use with PROC REPORT

```
data blackMarried ;
    length Stat $100 Variable $32 Label $100 Level $50;
    set blackF;
    Stat = compress(Frequency)|| " (" ||compress(round(Percent, .01 ))|| ")" ;
    Variable = "Black" ;
    Level = F_black ;
    Label = vlabel(black);
    keep Married Level Label variable Stat;
run;
```

blackMarried dataset looks like:

	Stat	Variable	Label	Level	Married
1	9053 (63)	Black	Black Race	[0]No	[0]No
2	5316 (37)	Black	Black Race	[1]Yes	[0]No
3	32805 (92.07)	Black	Black Race	[0]No	[1]Yes
4	2826 (7.93)	Black	Black Race	[1]Yes	[1]Yes

Generalize data steps for a macro

Step 4: Put data steps together in a macro generalizing set, where, by variable and variable

```
%Macro NPct(StatData= , WhereStmnt= , ByVar= , Var= )
proc sort; data=&StatData.; by &ByVar.; run;
proc freq data=&StatData. ;
where &WhereStmnt. ;
by &ByVar. ;
    tables &Var. /missing;
    ods output onewayfreqs=&Var.F;
run;
data &Var.&ByVar. ;
   length Stat $100 Variable $32 Label $100 Level $50;
   set &Var.F;
   Stat = compress(Frequency)|| " (" ||compress(round(Percent, .01 ))|| ")" ;
   Variable = "&Var." ;
   Level = F_&Var. ;
   Label = vlabel(&Var.);
   keep &ByVar. Level Label variable Stat;
run;
%Mend NPct;
```

Call macro

```
%NPct(StatData=BWeight , WhereStmnt=, ByVar=Married, Var=black);
%NPct(StatData=BWeight , WhereStmnt=, ByVar=Married, Var=boy);
%NPct(StatData=BWeight , WhereStmnt=, ByVar=Married, Var=smoke);
```

Create one table with all output

```
data Table1;
        set blackMarried boyMarried smokeMarried;
run;
```

Table1 dataset looks like this:

	Stat	Variable	Label	Level	Married
1	9053 (63)	Black	Black Race	[0]No	[0]No
2	5316 (37)	Black	Black Race	[1]Yes	[0]No
3	32805 (92.07)	Black	Black Race	[0]No	[1]Yes
4	2826 (7.93)	Black	Black Race	[1]Yes	[1]Yes
5	7027 (48.9)	boy	Boy	[0]No	[0]No
6	7342 (51.1)	boy	Boy	[1]Yes	[0]No
7	17181 (48.22)	boy	Boy	[0]No	[1]Yes
8	18450 (51.78)	boy	Boy	[1]Yes	[1]Yes
9	11149 (77.59)	smoke	Smoker	[0]No	[0]No
10	3220 (22.41)	smoke	Smoker	[1]Yes	[0]No
11	32318 (90.7)	smoke	Smoker	[0]No	[1]Yes
12	3313 (9.3)	smoke	Smoker	[1]Yes	[1]Yes

Which you can turn into a report like this:

Table 1: Descriptives by Marital Status

		Married	
		[0]No	**[1]Yes**
Black Race	[0]No	9053 (63)	32805 (92.07)
	[1]Yes	5316 (37)	2826 (7.93)
Boy	[0]No	7027 (48.9)	17181 (48.22)
	[1]Yes	7342 (51.1)	18450 (51.78)
Smoker	[0]No	11149 (77.59)	32318 (90.7)
	[1]Yes	3220 (22.41)	3313 (9.3)

Categorical variables expressed as: N (%)

Macro Libraries

- Michael Winter
- Associate Director of Statistical Programming, Data Coordinating Center

Macro Libraries

- Macro libraries are a way to control and manage macros
- Especially useful when a group of programmers work together
- And they aren’t very complicated!

Storing Compiled Macros

Need to include a LIBNAME statement identifying where to store the macros, and some options:

```
Libname DCCMac "\\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSUG\DCC
Macros";
Options Mstored SASmStore = DCCMac McompileNote = all;
```

SASmStore: *Identifies the libname where macros are stored.*

Mstored: *Indicates that stored macros are available in the location specified by SASmStore.*

McomplileNote = all: *Writes a note to the log when compile is successful.*

Example – Storing Compiled Macro

Using the macro Clara just defined, we just need to add some code to store it in the macro library:

```
%Macro NPct(StatData= , WhereStmnt= , ByVar= , Var= )/store des = "Create Dataset
Containing N Pct for 'Var'"; ← /store stores the macro,  des= adds a description
proc freq data=&StatData. ;
where &WhereStmnt. ;
by &ByVar. ;
     tables &Var. /missing;
     ods output onewayfreqs=&Var.F;
run;
data &Var.&ByVar. ;
     length Stat $100 Variable $32 Label $100 Level $50;
     set &Var.F;
     Stat = compress(Frequency)|| " (" ||compress(round(Percent, .01 ))|| ")" ;
     Variable = "&Var." ;
     Level = F_&Var. ;
       Label = vlabel(&Var.);
     keep &ByVar. Level Label variable Stat;
run;
%Mend NPct;
```

SAS Log

```
73
74   Libname DCCMac "\\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSUG\DCC Macros";
NOTE: Libref DCCMAC was successfully assigned as follows:
      Engine:        V9
      Physical Name: \\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSUG\DCC Macros
75   Options Mstored SASmStore = DCCMac McompileNote = all Mautolocdisplay;
76
77   /*MACRO TO CREATE A DATASET WITH THE FOLLOWING VARIABLES:*/
78   /*"ByVar": Variable named for the variable specified in the proc means by statement.*/
79   /*Level: Specifies the level of the variable specified in the tables statement of proc freq.*/
80   /*Variable: Specifies the variable name for the corresponding Statistic.*/
81   /*Stat: The Mean ± SD for the Variable specified in the proc means Var statemen.*/
82   /* Label: The label for the variable specified*/
83   %Macro NPct(StatData= , WhereStmnt= , ByVar= , Var= )/store des = "Create Dataset Containing N
83 !  Pct for 'Var'";
84   /*N%*/
85   proc freq data=&StatData. ;
86   where &WhereStmnt. ;
87   by &ByVar. ;
88         tables &Var. /missing;
89         ods output onewayfreqs=&Var.F;
90   run;
91
92   data &Var.&ByVar. ;
93         length Stat $100 Variable $32 Label $100 Level $50;
94         set &Var.F;
95
96         Stat = compress(Frequency)|| " (" ||compress(round(Percent, .01 ))|| ")" ;
97         Variable = "&Var." ;
98         Level = F_&Var. ;
99           Label = vlabel(&Var.);
100        keep &ByVar. Level Label variable Stat;
101  run;
102
103  %Mend NPct;
NOTE: The macro NPCT completed compilation without errors.
      15 instructions 780 bytes.
```

Storing Macros

Be sure to save your source code (SAS program) with the macro definition – you cannot re-create it from the compiled macro!

Document the code well!

Viewing Stored Macros

You can view a list of stored macros in a macro catalog using the following code:

```
Libname DCCMac "\\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSUG\DCC Macros";
Options Mstored SASmStore = DCCMac McompileNote = all;

proc catalog catalog=DCCMac.sasmacr;
  contents;
run;
quit;
```

List of Stored Macros

The SAS System

Contents of Catalog DCCMAC.SASMACR

#	Name	Type	Create Date	Modified Date	Description
1	GENMOD	MACRO	06Aug14:14:21:35	06Aug14:14:21:35	
2	GENMOD2	MACRO	06Aug14:14:44:43	06Aug14:14:44:43	
3	MEANSD	MACRO	01Aug14:16:35:50	01Aug14:16:35:50	Create Dataset Containing Continuous Descriptives for 'Var'
4	MEANSD_T	MACRO	07Nov14:13:04:49	07Nov14:13:04:49	Create Dataset Containing Continuous Descriptives for 'Var'
5	MEANSD_T2	MACRO	09Sep14:16:58:31	09Sep14:16:58:31	Create Dataset Containing Continuous Descriptives for 'Var'
6	NPCT	MACRO	16Nov14:14:11:48	16Nov14:14:11:48	Create Dataset Containing N Pct for 'Var'
7	NPCT_T	MACRO	07Nov14:13:04:49	07Nov14:13:04:49	Create Dataset Containing N Pct for 'Var'
8	PHREG	MACRO	06Aug14:16:24:52	06Aug14:16:24:52	
9	PHREGTABLE	MACRO	07Aug14:10:31:33	07Aug14:10:31:33	
10	PHREGTV1	MACRO	06Aug14:16:24:52	06Aug14:16:24:52	
11	PHREGTV2	MACRO	06Aug14:16:24:52	06Aug14:16:24:52	

\\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSUG\Meeting_20140807\Constructing Macros.sas

Calling a Stored Macro

```
Libname DCCMac "\\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSUG\DCC Macros";
Options Mstored SASmStore = DCCMac McompileNote = all;

%NPct(StatData=BWeight , WhereStmnt=, ByVar=Married, Var=black);
%NPct(StatData=BWeight , WhereStmnt=, ByVar=Married, Var=boy);
%NPct(StatData=BWeight , WhereStmnt=, ByVar=Married, Var=smoke);
```