Biostatistics, Epidemiology & Research Design (BERD)

Howard Cabral, PhD, MPH
Christine Chaisson, MPH

Seminar Series: CTSI Presents…

November 20, 2014
Macro Basics: an introduction

- Leah Forman
- Statistical Data Analyst
  - Data Coordinating Center
- SAS Macro language:
  - References previously written SAS code and commands 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:

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

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

```latex
%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

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

The output dataset, “outdat”:

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>TYPE</em></th>
<th><em>FREQ</em></th>
<th>MeanPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>361</td>
<td>$155.32</td>
</tr>
</tbody>
</table>
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:

```sas
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 ;
  ```
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

```plaintext
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;
```
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

```r
proc freq data=BWeight;
where ;
by Married ;
  tables black /missing;
  ods output onewayfreqs=blackF;
run;
```
blackF dataset looks like this:

<table>
<thead>
<tr>
<th>Married</th>
<th>Table</th>
<th>black</th>
<th>Black Race</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0]No</td>
<td>[0]No</td>
<td>[0]No</td>
<td>9053</td>
<td>63.00</td>
<td>9053</td>
<td>63.00</td>
</tr>
<tr>
<td>2</td>
<td>[0]No</td>
<td>[1]Yes</td>
<td>[1]Yes</td>
<td>5316</td>
<td>37.00</td>
<td>14369</td>
<td>100.00</td>
</tr>
<tr>
<td>3</td>
<td>[1]Yes</td>
<td>[0]No</td>
<td>[0]No</td>
<td>32805</td>
<td>92.07</td>
<td>32805</td>
<td>92.07</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Stat</th>
<th>Variable</th>
<th>Label</th>
<th>Level</th>
<th>Married</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9053 (63)</td>
<td>Black Race</td>
<td>[0] No</td>
<td>[0] No</td>
</tr>
<tr>
<td>2</td>
<td>5316 (37)</td>
<td>Black Race</td>
<td>[1] Yes</td>
<td>[0] No</td>
</tr>
<tr>
<td>3</td>
<td>32805 (92.07)</td>
<td>Black Race</td>
<td>[0] No</td>
<td>[1] Yes</td>
</tr>
</tbody>
</table>
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= , WhereStmt= , ByVar= , Var= )
proc sort; data=&StatData.; by &ByVar.; run;
proc freq data=&StatData. ;
where &WhereStmt. ;
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 , WhereStmt=, ByVar=Married, Var=black);
%NPct(StatData=BWeight , WhereStmt=, ByVar=Married, Var=boy);
%NPct(StatData=BWeight , WhereStmt=, ByVar=Married, Var=smoke);
Create one table with all output

data Table1;
    set blackMarried boyMarried smokeMarried;
run;
Table 1 dataset looks like this:

<table>
<thead>
<tr>
<th>Stat</th>
<th>Variable</th>
<th>Label</th>
<th>Level</th>
<th>Married</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9053 (63)</td>
<td>Black Race</td>
<td>[0]No</td>
<td>[0]No</td>
</tr>
<tr>
<td>2</td>
<td>5316 (37)</td>
<td>Black Race</td>
<td>[1]Yes</td>
<td>[0]No</td>
</tr>
<tr>
<td>3</td>
<td>32805 (92.07)</td>
<td>Black Race</td>
<td>[0]No</td>
<td>[1]Yes</td>
</tr>
<tr>
<td>5</td>
<td>7027 (48.9)</td>
<td>Boy</td>
<td>[0]No</td>
<td>[0]No</td>
</tr>
<tr>
<td>6</td>
<td>7342 (51.1)</td>
<td>Boy</td>
<td>[1]Yes</td>
<td>[0]No</td>
</tr>
<tr>
<td>7</td>
<td>17181 (42.22)</td>
<td>Boy</td>
<td>[0]No</td>
<td>[1]Yes</td>
</tr>
<tr>
<td>9</td>
<td>11149 (77.59)</td>
<td>Smoker</td>
<td>[0]No</td>
<td>[0]No</td>
</tr>
<tr>
<td>10</td>
<td>3220 (22.41)</td>
<td>Smoker</td>
<td>[1]Yes</td>
<td>[0]No</td>
</tr>
<tr>
<td>11</td>
<td>32318 (90.7)</td>
<td>Smoker</td>
<td>[0]No</td>
<td>[1]Yes</td>
</tr>
</tbody>
</table>
Which you can turn into a report like this:

Table 1: Descriptives by Marital Status

<table>
<thead>
<tr>
<th></th>
<th>Married</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[0]No</td>
<td>[1]Yes</td>
<td></td>
</tr>
<tr>
<td>Black Race</td>
<td>9053 (63)</td>
<td>32805 (92.07)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5316 (37)</td>
<td>2826 (7.93)</td>
<td></td>
</tr>
<tr>
<td>Boy</td>
<td>7027 (48.9)</td>
<td>17181 (48.22)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7342 (51.1)</td>
<td>18450 (51.78)</td>
<td></td>
</tr>
<tr>
<td>Smoker</td>
<td>11149 (77.59)</td>
<td>32318 (90.7)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3220 (22.41)</td>
<td>3313 (9.3)</td>
<td></td>
</tr>
</tbody>
</table>

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:

```sas
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:

```sas
%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

74 Libname DCCMac "\ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSSUG\DCC Macros";
75 NOTE: Libref DCCMAC was successfully assigned as follows:
76 Engine: V9
77 Physical Name: \ad.bu.edu\bumcfiles\SPH\DCC\Dept\DCCSSUG\DCC Macros
78 Options Mstored SASmStore = DCCMac McompileNote = all Mautolocdisplay;
79
80 /*MACRO TO CREATE A DATASET WITH THE FOLLOWING VARIABLES:*/
81 /*"ByVar": Variable named for the variable specified in the proc means by statement.*/
82 /*Level: Specifies the level of the variable specified in the tables statement of proc freq.*/
83 /*Stat: Specifies the variable name for the corresponding Statistic.*/
84 /*Label: The label for the variable specified*/
85 %Macro NPct(StatData=, WhereStmt=, ByVar=, Var= )/store des = "Create Dataset Containing N
86 Pct for 'Var'";
87 /*NX*/
88 proc freq data=&StatData. ;
89 where &WhereStmt. ;
90 by &ByVar. ;
91 tables &Var. /missing;
92 ods output onewayfreqs=&Var.F;
93 run;
94 data &Var.&ByVar. ;
95 length Stat $100 Variable $32 Label $100 Level $50;
96 set &Var.F;
97 Stat = compress(Frequency)!! "(" !compress(round(Percent, .01 ))!! ")" ;
98 Variable = "&Var." ;
99 Level = F.&Var. ;
100 Label = vlabel(&Var.);
101 keep &ByVar. Level Label variable Stat;
102 run;
103 %Mend NPct;
104
105 NOTE: The macro NPCT completed compilation without errors.
106 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:

```sas
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

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Create Date</th>
<th>Modified Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GENMOD2</td>
<td>MACRO</td>
<td>06Aug14:14:44:43</td>
<td>06Aug14:14:44:43</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MEANSD</td>
<td>MACRO</td>
<td>01Aug14:16:35:50</td>
<td>01Aug14:16:35:50</td>
<td>Create Dataset Containing Continuous Descriptives for 'Var'</td>
</tr>
<tr>
<td>4</td>
<td>MEANSD_T</td>
<td>MACRO</td>
<td>07Nov14:13:04:49</td>
<td>07Nov14:13:04:49</td>
<td>Create Dataset Containing Continuous Descriptives for 'Var'</td>
</tr>
<tr>
<td>5</td>
<td>MEANSD_T2</td>
<td>MACRO</td>
<td>09Sep14:16:58:31</td>
<td>09Sep14:16:58:31</td>
<td>Create Dataset Containing Continuous Descriptives for 'Var'</td>
</tr>
<tr>
<td>6</td>
<td>NPCT</td>
<td>MACRO</td>
<td>16Nov14:14:11:48</td>
<td>16Nov14:14:11:48</td>
<td>Create Dataset Containing N Pct for 'Var'</td>
</tr>
<tr>
<td>7</td>
<td>NPCT_T</td>
<td>MACRO</td>
<td>07Nov14:13:04:49</td>
<td>07Nov14:13:04:49</td>
<td>Create Dataset Containing N Pct for 'Var'</td>
</tr>
<tr>
<td>8</td>
<td>PHREG</td>
<td>MACRO</td>
<td>06Aug14:16:24:52</td>
<td>06Aug14:16:24:52</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>PHREGTV1</td>
<td>MACRO</td>
<td>06Aug14:16:24:52</td>
<td>06Aug14:16:24:52</td>
<td></td>
</tr>
</tbody>
</table>

`\ad.bu.edu\bumcfiles\SPH\DCCI\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 , WhereStmt=, ByVar=Married, Var=black);
%NPct(StatData=BWeight , WhereStmt=, ByVar=Married, Var=boy);
%NPct(StatData=BWeight , WhereStmt=, ByVar=Married, Var=smoke);