SAS Macros:
An Introduction

Sheila Barron
University of Iowa, College of Education
Statistics Outreach Center

University of Iowa SAS User Group (UISUG)
December 1, 2010

Overview

I. What, when, and why

II. Macro variables

III. Macro programs

IV. Macro functions and statements

V. Debugging tips and tricks

VI. Discussion / Questions

The what, when, and why of SAS macros

- **What:** The SAS macro facility is a tool for text substitution.
- **When:** Consider using the macro facility for situations involving routine or repetitive tasks.
- **Why:** Efficiency – You can write a program once, and then with only very small changes produce new results.

Macro variables

- A macro variable is simply a reference to a value. The value is not part of any dataset, it is just held by SAS for when you need it.
- %LET tells SAS that you want to define a macro variable. It is followed by the name of the macro variable, an equal sign, and the value of the macro variable.

```
%LET SUBJ = Reading;
%LET DESC = Reading Comprehension;
%LET I = 10;
```

- Macro variables are typically used to repeatedly insert a piece of text throughout a program. A macro variable is referenced in a program using the ampersand (&).

Example 1 – First without a macro variable

- TITLE1 BOLD HEIGHT=14PT 'Descriptive Statistics for Reading';
- PROC FREQ;
- TABLES READING;
- PROC UNIVARIATE NOPRINT;
- HISTOGRAM READING / MIDPOINTS = 0 TO 10 BY 1;
- PROC MEANS MAXDEC=2 N MEAN STD MIN MAX;
- VAR READING;
- RUN;

Example 1 – Now with a macro variable called SUBJ

- **%LET SUBJ = Reading;**
- TITLE1 BOLD HEIGHT=14PT "Descriptive Statistics for &SUBJ";
- PROC FREQ;
- TABLES **&SUBJ**;
- PROC UNIVARIATE NOPRINT;
- HISTOGRAM **&SUBJ** / MIDPOINTS = 0 TO 10 BY 1;
- PROC MEANS MAXDEC=2 N MEAN STD MIN MAX;
- VAR **&SUBJ**;
- RUN;

Example 1 – I only need one change to run the same program for Math

- **%LET SUBJ = Math;**
- TITLE1 BOLD HEIGHT=14PT "Descriptive Statistics for &SUBJ";
- PROC FREQ;
- TABLES &SUBJ /NOCUM;
- PROC UNIVARIATE NOPRINT;
- HISTOGRAM &SUBJ / MIDPOINTS = 0 TO 10 BY 1;
- PROC MEANS MAXDEC=2 N MEAN STD MIN MAX;
- VAR &SUBJ;
- RUN;

Macro variables (cont.)

- There is another type of macro variable called an automatic macro variable.
- These macro variables are automatically created when you start a SAS session or when you execute certain statements.
  - The most common examples of these are SYSDATE (or SYSDATE9), SYSTIME, and SYSLAST

Example 1 – With some automatic macro variables added

- %LET SUBJ = Reading;

- TITLE1 BOLD HEIGHT=14PT "Descriptive Statistics for &SUBJ";
- FOOTNOTE1 JUSTIFY=LEFT "Report Date: &SYSDATE9 Dataset Used: &SYSLAST";

- PROC FREQ;
- TABLES &SUBJ /NOCUM;

- PROC UNIVARIATE NOPRINT;
- HISTOGRAM &SUBJ / MIDPOINTS = 0 TO 10 BY 1;

- PROC MEANS MAXDEC=2 N MEAN STD MIN MAX;
- VAR &SUBJ;

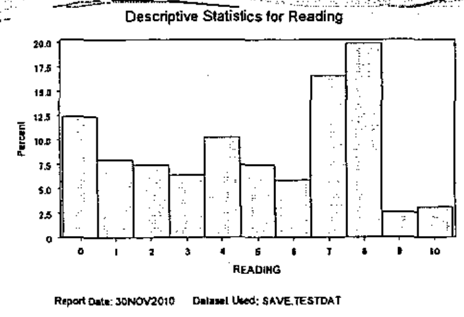
- RUN;

Output from Example 1

- *Descriptive Statistics for Reading*
- *The FREQ Procedure*

[illegible]	[illegible]	[illegible]
0	124	12.40
1	80	8.00
2	74	7.40
3	64	6.40
4	103	10.30
5	74	7.40
6	59	5.90
7	166	16.60
8	199	19.90
9	26	2.60
10	31	3.10

- *Report Date: 30NOV2010 Dataset Used: SAVE.TESTDAT*



- *Descriptive Statistics for Reading*
-
- *The MEANS Procedure*

N	Mean	Std Dev	Minimum	Maximum
1000	4.85	3.02	0.00	10.00

- *Report Date: 30NOV2010 Dataset Used: SAVE.TESTDAT*

Lots of Macro Variable Examples

- %LET EX1 = Can macro variables can have blanks?;
- %PUT &EX1;

- %LET EX2 = "What happens if there are quotation marks in my macro variable?";
- %PUT &EX2;

- %LET EX3 = 42 + 35;
- %PUT &EX3;

- %LET EX4a = Macro variables can;
- %LET EX4b = include macro variables;
- %LET EX4 = &EX4a &EX4b;
- %PUT &EX4;

- %LET EX5 =   What happens with leading blanks?;
- %PUT &EX5;

MACRO VAR5.SAS

Lots of Macro Variable Examples— A look at my log

- 1 %LET EX1 = Can macro variables have blanks?;
- 2 %PUT &EX1;
- **Can macro variables have blanks?**

- 3 %LET EX2 = "What happens if there are quotation marks in my macro variable?";
- 4 %PUT &EX2;
- **"What happens if there are quotation marks in my macro variable?"**

- 5 %LET EX3 = 42 + 35;
- 6 %PUT &EX3;
- 42 + 35

- 7 %LET EX4a = Macro variables can;
- 8 %LET EX4b = include macro variables;
- 9 %LET EX4 = &EX4a &EX4b;
- 10 %PUT &EX4;
- **Macro variables can include macro variables**

- 11 %LET EX5 =   What happens with leading blanks?;
- 12 %PUT &EX5;
- **What happens with leading blanks?**

Combining Macro Variables with Text

- If the text comes before the macro variable, then no problem go ahead and include the macro variable.

- 787 %LET GRD = 8;
- 788
- 789 DATA SAVE.GRADE&GRD; SET SAVE.TESTDAT;
- 790 IF GRADE = &GRD;
- 791
- 792 RUN;

- NOTE: There were 1000 observations read from the data set SAVE.TESTDAT.
- NOTE: The data set SAVE.GRADE8 has 368 observations and 34 variables.

Combining Macro Variables with Text (cont.)

- If the text comes after the macro variable, you have to be able to tell SAS that the macro variable name has ended.

- What would happen if I ran this code?

- DATA SAVE.&SUBJDAT;SET SAVE.TESTDAT;
- KEEP ID GRADE &SUBJ;

- I get this WARNING and then I get an error.

  - WARNING: Apparent symbolic reference SUBJDAT not resolved.

Combining Macro Variables with Text (cont.)

- If the text comes after the macro variable, put a period after the macro variable to tell SAS that it has reached the end of the macro variable name.

  - **DATA SAVE.&SUBJ.DATA;SET SAVE.TESTDAT;**
  - KEEP ID GRADE &SUBJ;

- Here is what the log looks like:

  - 11 DATA SAVE.&SUBJ.DAT;SET SAVE.TESTDAT;
  - 12 KEEP ID GRADE &SUBJ;

- NOTE: There were 1000 observations read from the data set SAVE.TESTDAT.
- NOTE: The data set SAVE.READINGDAT has 1000 observations and 3 variables.

Macro Programs

- A macro program is a chunk of code that has been defined as a macro.

- %MACRO tells SAS that you are beginning a macro.
  - Follow %MACRO with the name you want to give your macro.

- %MEND tells SAS that you are ending a macro.
  - You can also give the name in the %MEND, but it is optional.

- In between %MACRO and %MEND you can put any SAS statements -- DATA steps, PROCs, other macros.

- To actually run a macro, you need to call it. Give the name of the macro with a percent sign before it.

Macro Program Example

- %MACRO *DESCRIP*;
- TITLE1 BOLD HEIGHT=14PT 'Descriptive Statistics for Reading';

- DATA SAVE.GRADE8;SET SAVE.TESTDAT;
- IF GRADE = 8;
- KEEP ID GRADE READING;

- PROC FREQ;
- TABLES READING;

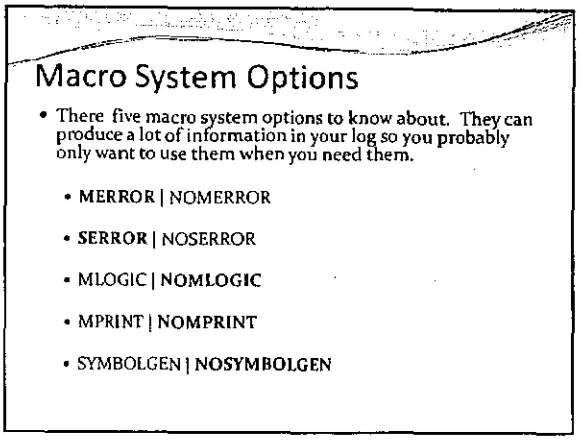
- PROC UNIVARIATE NOPRINT;
- HISTOGRAM READING / MIDPOINTS = 0 TO 10 BY 1;

- PROC MEANS MAXDEC=1 N MEAN STD MIN MAX;
- VAR READING;

- RUN;

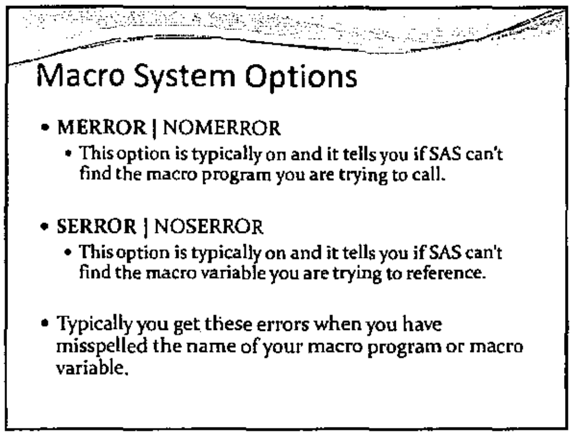
- %MEND DESCRIP;

- %*DESCRIP*



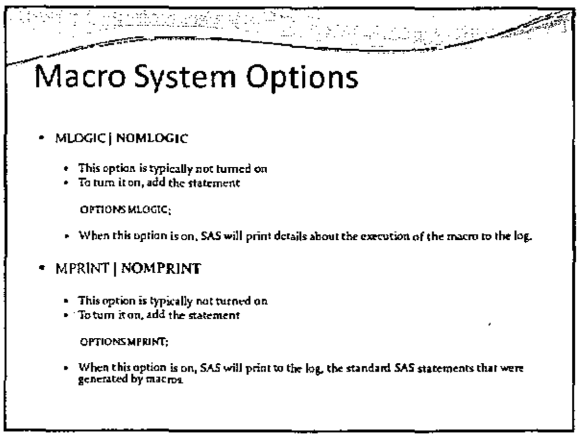
Macro System Options

- There five macro system options to know about. They can produce a lot of information in your log so you probably only want to use them when you need them.
  - MERROR | NOMERROR
  - SERROR | NOSERROR
  - MLOGIC | NOMLOGIC
  - MPRINT | NOMPRINT
  - SYMBOLGEN | NOSYMBOLGEN



Macro System Options

- MERROR | NOMERROR
  - This option is typically on and it tells you if SAS can't find the macro program you are trying to call.
- SERROR | NOSERROR
  - This option is typically on and it tells you if SAS can't find the macro variable you are trying to reference.
- Typically you get these errors when you have misspelled the name of your macro program or macro variable.

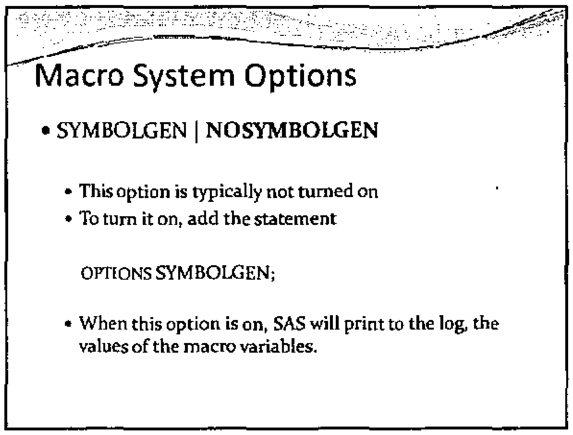


Macro System Options

- MLOGIC | NOMLOGIC
  - This option is typically not turned on
  - To turn it on, add the statement

    OPTIONS MLOGIC;
  - When this option is on, SAS will print details about the execution of the macro to the log.
- MPRINT | NOMPRINT
  - This option is typically not turned on
  - To turn it on, add the statement

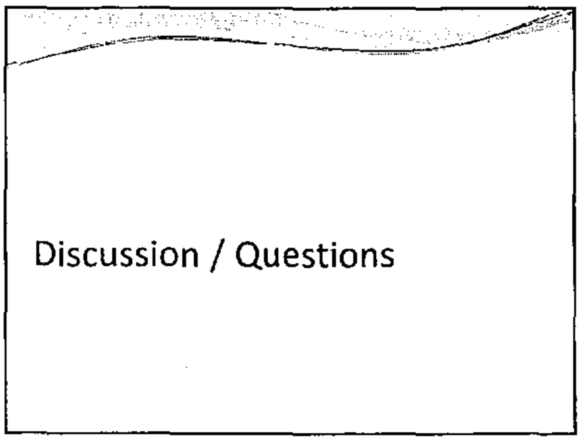
    OPTIONS MPRINT;
  - When this option is on, SAS will print to the log, the standard SAS statements that were generated by macros



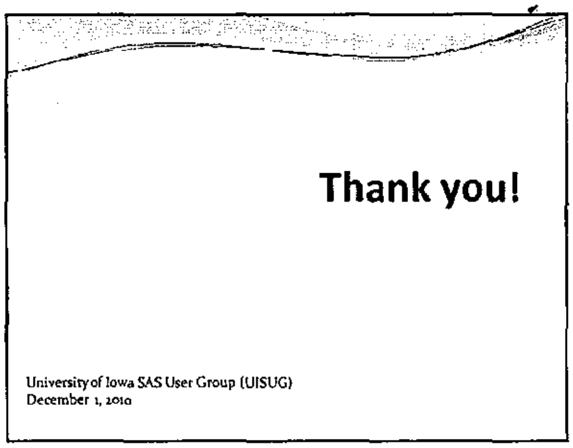
Macro System Options

- SYMBOLGEN | NOSYMBOLGEN
  - This option is typically not turned on
  - To turn it on, add the statement

    OPTIONS SYMBOLGEN;
  - When this option is on, SAS will print to the log, the values of the macro variables.



Discussion / Questions



Thank you!

University of Iowa SAS User Group (UISUG)
December 1, 2010