

sas macros

Juan Shishido, D-Lab, UC Berkeley

Month Day, 2016

introduction

The SAS macro facility enables the generation of text in SAS programs

It has two components:

- macro language—is text based and defines the syntax needed for writing macro statements
- macro processor—resolves macro statements into standard SAS statements

macro variables and macros

The macro processor is activated when either `&` or `%` characters are encountered

These refer to macro variables and macros, respectively

macro variables

Macro variables are used to represent text and:

- can be defined and used anywhere in SAS programs
- do not depend on SAS data sets
- correspond to a single value (until explicitly changed)

They are used to substitute text in SAS programs

Also enable the substitution of text in SAS programs, but can include:

- macro variables
- DATA and PROC steps
- other macro statements

SAS programs can include any number of macro variables or macros

Macro variables defined outside of macros are considered global, which means they can be referenced anywhere in the program

When they are defined inside of macros, however, they are local and can only be referenced within the macro

Macro variables and macros are useful in situations where

- code is repetitive
- values must be dynamic
- execution of code depends on other values

syntax

defining macro variables

The easiest way to define a macro variable is by using the %let macro statement

For example

```
%let workshop = SAS;
```

Notice that quotes are *not* needed

In a program, we would reference this using &workshop

using macro variables

Suppose that the D-Lab needed to print details for a given set of workshops

The print procedure might look like

```
proc print data = SAS;  
    var = description;  
    title 'D-Lab Workshop on SAS';  
run;
```

For a particular SAS workshop

using macro variables

Suppose the D-Lab also wanted needed to print descriptions for its Python and R workshops

Rather than repeating the print statements, we could use the `&workshop` variable as follows

```
%let workshop = Python;  
proc print data = &workshop;  
    var = description;  
    title "D-Lab Workshop on &workshop";  
run;
```

using macro variables

And for R

```
%let workshop = R;  
proc print data = &workshop;  
    var = description;  
    title "D-Lab Workshop on &workshop";  
run;
```

Notice that we only have to replace the value of &workshop to get the descriptions for the other variables

Also, when calling macro variables that are part of strings, we must use double quotes

periods

double ampersands

creating macro variables from data steps

other macro statements

references

- SAS Macro Programming for Beginners
- SAS 9.3 Macro Language Reference
- Get Started Writing SAS Macros
- Get Started with Macro
- Resolving and Using `&&var&i` Macro Variables