Class Exercise 9

This exercise is based upon Chapter 10 of the SAS Advanced Certification Prep Guide. We will review execution of macro variables during the DATA step. We try to set up a more natural context, in which we work with a grouped data set, and use \_N\_ to index our macro variables.

Begin again by importing the simulated Medicare data set from Excel into WORK.meddb. Run the following commands to refamiliarize yourself with the data.

proc print data=meddb (obs=20);

format dos date7. SSN SSN11. claim dollar14.2;

run;

Let’s create some grouped output first.

proc sql;

create table tosclaim as select tos, count(\*) as nct, sum(claim) as totalclaim

from meddb

group by tos;

quit;

We will next use CALL SYMPUT to print a title for total claims for Type of Service A and B. Do either of the macro variables in the title need to be adjusted for leading or trailing blanks?

data aandbtoal;

set tosclaim;

if tos='A and B' then do;

call symput('tos',tos);

call symput('nct',nct);

end;

proc print noobs label;

title "Total Medicare &tos claims (n=&nct)";

label totalclaim="Total Claims";

var totalclaim;

format totalclaim dollar18.2;

where tos='A and B';

run;

We have used PUT for date formats and seen it used for numeric formats. We can try it with a SSN and see how it works. Fix any formatting problems with the output.

data meddb1rec; set meddb;

if \_n\_=25;

call symput('record',\_n\_);

call symput('SSN',put(SSN,ssn11.));

run;

proc print data=meddb1rec noobs;

title "Medicare Claim for Entry &record (SSN=&SSN)";

var tos dos provider claim;

format dos worddate21. claim dollar15.2;

run;

Let’s create a series of macro variables that stores each of the 4 types of service. What does the output from %PUT \_USER\_ print in the LOG window?

**data** \_null\_;

set tosclaim;

call symput('tos'||put(\_n\_,**1.**),tos);

**run**;

%put \_user\_;

We can now print claims by assigning the name of one of our new macro variables to another macro variable. Note that we had to use the &&& in two different places for the macro variables to resolve correctly. Review the LOG and comment. Graduate students should use a single & in the WHERE clause and comment on the LOG.

options symbolgen;

%let tostype=tos1;

proc print data=tosclaim label noobs;

var nct totalclaim;

label nct="Claims Count" totalclaim="Total Claims";

format totalclaim dollar15.2;

where tos="&&&tostype";

title "Type of Service: &&&tostype";

run;

Replace the WHERE clause above with a WHERE clause that uses a direct SYMGET call of tos1. The use of SYMGET here is superfluous, since the values of tos are readily available in tosclaim and do not need to be recovered from our list of macro variables.