

NESTED PROCEDURE LOOPS: YES! THEY EXIST & FLOURISH

Linda C. Rowland, IRA, Richardson, Texas

ABSTRACT

Utilizing the power of nested do loops for variable assignments, has always been a recognized option. However, the real power, use, and benefit of a nested loop is generated when PROC statements are a part of the repetitive function. Placing PROCs within nested loops requires the use of macros. The minor inconvenience macro programming may create is insignificant when compared to the tremendous advantage of addressing repetitive programming within a loop. Applications of this type of coding are infinite and encompass all fields of discipline.

INTRODUCTION

Jumping Johosavat! There is a SAS nested procedure loop. For those of you who are unfamiliar with SAS macros, possibly even Mr. Spock, the combination of the words "PROC" and "loop" may seem illogical. There are some of us, myself included, that spent many an hour copying code and changing the parameter to ensure that all combinations of outputs were considered. Although there is insufficient information available at this time to evaluate the hypothesis, my previous coworkers and I are certain that all tedious time consuming tasks, such as copying lines of code into infinity, causes adverse health conditions. Little voices, being the most common of all detrimental side effects, may remind the reader of one of his/her own experiences.

Most frequent among these voices are the phrases: "This is an inefficient use of personnel, computer space and time.", "This creates a lot of quality issues!", "If my computer science professor could see me now, I would be in trouble!....Can they change your grade after you graduate?", "Why can't this be like any other language....All I would have to do is put it in a nested loop!", "Where was I?", "Did I miss a parameter?", "How many parameters did I miss?", "What am I doing?", and so oninto garbled patterns that somewhat assimilate brain waves. The happy ending to the story is there is a simple solution. Nested PROC loops are at one's disposal through the use of macros.

BENEFITS

A debate over the efficiency of a nested loop versus typing in all iterations in hard code is nonexistent. Common sense allows us the luxury of deducing the fact that a working computer can produce all possible iterations much faster than we, as people, can possibly type them all by hand. Quality, in the form of typing errors and omissions, also presents little argument. I have yet to hear of anyone in the programming arena, who has never made some type of typing mistake while keying large volumes of iterative code.

Improved efficiency and quality are definite benefits. The next section provides a step-by-step approach that may be adapted to any number of procedure statements.

METHODOLOGY

The methodology for the nested loop is fairly basic. An outer loop calls an inner loop. It is very similar to nested loops in other languages, with the exception that it is necessary to utilize macro utilities. As depicted in Exhibit 1, the outer loop is a macro. The parameters of the nested loop are passed to the outer loop macro when the outer loop macro is called. The "put" statements allow the parameters to be valid numeric do loop parameters. The do loop portion of the outer loop code is very analogous to nested loop code in any language, the inner loop macro being akin to a function call.

The inner loop bears the power of the PROC statement(s). Pertinent variables are passed from the outer loop to the inner loop as parameters. Although the variable name need not be altered within the code, the subscript needs a preceding ampersand. Exhibit 1 illustrates an example.

CONCLUSION

Macros are an efficient and effective approach to providing the results of nested loops. With a small amount of effort, personnel and computer resources may be utilized more efficiently and productively. The time invested in writing a few macros is minuscule when compared to hard coding all iterative lines of code. The power and application of this approach is limitless and encompasses all fields of discipline.

SAMPLE CODE EXHIBIT 1

DATA SAMPLE; SET INFORM;

```
*****THE INNER LOOP*****  
%MACRO PRHAVEN(VI,VJ);                *INPUT PARAMETERS;  
    PROC FREQ; TABLES V&I*V&J;  
    PROC "TO YOUR HEARTS CONTENT";  
  
    *MAKE SURE ANY REFERENCE TO PARAMETER INDICES ;  
    *ARE CODED WITH AN AMPERAND;  
%MEND PRHAVEN;
```

```
*****THE OUTER LOOP*****  
%MACRO FRTAB(&PARM1, &PARM2);  
    %PUT &PARM1;  
    %PUT &PARM2;  
    %DO K=1 %TO &PARM1;  
        %DO J=2 %TO &PARM2;  
            %PRHAVEN(V&K, V&J);        *INNER LOOP CALL;  
        %END;  
    %END;  
%MEND FRTAB(&PARM1,&PARM2);
```

%FRTAB(PARM1, PARM2);

*CALL OUTER MACRO TO INITIATE THE;
*NESTED LOOP;

ACKNOWLEDGEMENTS

IRA, An Integrated Database Marketing Service Firm, 701 N
Plano Road, Richardson, TX 75081

SAS Institute; special gratitude to Russ Tyndall for his
exemplary input.

SAS is a registered trademark or trademark of SAS
Institute, Inc in the USA and other countries.