How ***Big*** Does It Have To Be?!

Using Macros To Set Array Size At Run-Time.

John Fleming

Edmonton SAS User Group
October 26, 2011

Defining The Problem

```
proc transpose data=ds1 (keep = uli
                                start_dt
                                end_dt
                                mis
                                diag)
                out =ds2 (drop = _name_)
                prefix=diag_;
  by uli start_dt mis end_dt;
  var diag;

run;
```

Defining The Problem (2)

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
5	diag_1	Char	7			
6	diag_2	Char	7			
7	diag_3	Char	7			
8	diag_4	Char	7			
9	diag_5	Char	7			
10	diag_6	Char	7			
11	diag_7	Char	7			
12	diag_8	Char	7			
13	diag_9	Char	7			
14	diag_10	Char	7			
15	diag_11	Char	7			

How Many Diag_x Variables Are There?

Get SAS To Work This Out At Run-Time . . .

SAS Can Do It!!

(It Can Do It Better And Faster Than We Can Too.)

. . . With An Added Twist!

- Output data set must contain at least ten diagnosis variables
- If more than ten diagnosis variables, must be able to dynamically adjust array sizes in our code so processing includes all variables.

The Macro

```
* Define macro function with four parameters:

  var_count = macro variable name to store array size
  ds        = data set name
  var       = common part of array variable names
  z         = minimum array size parameter;

%macro array_size (var_count=, ds=, var=, z=);

[Part 1 - Declare Global Macro Variable]
[Part 2 - Number of Variables]
[Part 3 - Add Variables, If Needed]

%mend array_size;
```

Part 1 – Declare Global Macro Variable

```
%macro array_size (var_count=, ds=, var=, z=);

  %global &var_count;

  [Code For Part 2 And Part 3 Here]

%mend array_size;
```

Part 2 – How Many Variables In The Array?

```
%let i = 0;

%let dsid = %sysfunc(open(&ds,i));

  %do %until (&x = 0);

    %let i = %eval(&i + 1);
    %let x = %sysfunc(varnum(&dsid,&var.&i));

  %end;

%let rc=%sysfunc(close(&dsid));
```

Part 3 – Do We Need To Add New Variables?

```
* Do this part if we need to add new variables.;

%if %eval(&i - 1) < &z %then %do;

  %let &var_count = &z;

  [Code For Adding And Initializing New Variables]

%end;

* Do this part if we do not.;

%if %eval(&i - 1) >= &z %then %do;

  %let &var_count = %eval(&i - 1);

%end;
```

Part 3 – Code For Adding New Variables

```
* Determine length of new variable to add.;

%let dsid  = %sysfunc(open(&ds,i));
%let vlong = %sysfunc(varlen(&dsid,%eval(&i-1));

* Add and initialize new variables.;

   %do j = &i %to &z;

      %if (%sysfunc(vartype(&dsid,&i))=N) %then %do;
        length &var.&j &vlong;  &var.&j=.;
      %end;

      %else %do;
        length &var.&j $ &vlong; &var.&j=' ';
      %end;

    %end;

%let rc=%sysfunc(close(&dsid));
```

Invoking The Macro

```
data old_ds;
  set old_ds;

  %array_size (var_count=diag_var_count,
               ds=old_ds,
               var=diag_,
               z=10);

run;
```

Later On – Adding Variables And Defining Arrays

```
data new_ds;
  set old_ds;

  length colon_cat_diag_1 -
         colon_cat_diag_&diag_var_count $15;

  array olddiag{&diag_var_count} $
       diag_1 - diag_&diag_var_count;

  array colon_cat_diag{&diag_var_count} $
        colon_cat_diag_1 - colon_cat_diag_&diag_var_count;

  [More Code Here]

run;
```

Later On – Controlling Program Execution

```
do i = 1 to &diag_var_count;

   [More Code Here]

end;
```

Alberta Health Services

Questions?

John Fleming

Community Oncology
Cancer Care, Alberta Health Services
1500 Sun Life Building, 10123 99 Street NW
Edmonton, AB T5J 3H1

john.fleming@albertahealthservices.ca
(780) 643 - 4341