To do or not do: A Quick Intro to SAS Arrays and Do Loops

Gopinath Narasimhan
Saskatchewan Cancer Agency

March 12, 2014

What's an Array?

- Most computer languages have some notation for repeating
- e.g. vector, matrix
- In SAS data step it's called an ARRAY
  - A group of variables defined in a data step.
  - All elements must be character or numeric.
  - Array elements don't need to be contiguous, the same length, or even related at all.

How do we use arrays?

- Use arrays to read and analyze repetitive data with a minimum of coding.
- An array and a loop can make the program smaller.
- Examples of Use
  - Applying the same computation to many variables simultaneously (e.g. Fahrenheit to Celsius)
  - Recoding variables (e.g. setting missing values to “99”)
  - Transposing data
  - Temporary Arrays

Applying same computation to multiple variables

- For each record (row) there are 24 variables (temp1-temp24) with the temperatures for each hour of the day.
- Temps are in Fahrenheit and need to convert them to Celsius.

```
data;
input etc.
celsius_temp1 = 5/9(temp1 - 32);
celsius_temp2 = 5/9(temp2 - 32);
. . .
celsius_temp24 = 5/9(temp24 - 32);
run;
```

Applying same computation to multiple variables

- Define arrays and use a do…end loop

```
data ;
input etc.;
array temperature_array {24} temp1-temp24;
array celsius_array {24} celsius_temp1-
  celsius_temp24;
do i = 1 to 24;
celsius_array{i} = 5/9(temperature_array{i}
  - 32);
end;
run;
```

Recoding Variables

- Missing coded as 99

```
data ...; set ...;
   array temperature_array {24} temp1-temp24;
   do i = 1 to 24;
   if temperature_array{i}=99 then
     temperature_array{i} = .;
   end;
   run;
```

Note…

- Variables do not need to be named consecutively.
- An array would work just as well with non-consecutive variable names.

```
array temperature_array {24} a b c d
  e... x;
```

- In this example, the variable “a” is equivalent to the first temperature, “b” to the second etc.

BASIC ARRAY CONCEPTS

- SAS arrays temporarily group and refer to SAS variables.
- Not a new data structure, the array name is not a variable, and arrays do not define additional variables.
- Array provides a different name to reference a group of variables.
- Since similar processing is generally completed on the array elements, references to the array are usually found within DO groups.

Transposing data

- Take existing data rotate so that information is arranged in different structure
- Alternative to PROC TRANSPOSE
- Can provide efficiency advantages when doing multiple transposes

A Simple Example

Want to change data from

LOCATION	PROD1	PROD2	PROD3	PROD4
123 My City NJ USA	319.43	89.99	149.55	17.99

To:

LOCATION	PRODUCT	PRICE
123 My City NJ USA	PROD1	319.43
123 My City NJ USA	PROD2	89.99
123 My City NJ USA	PROD3	149.55
123 My City NJ USA	PROD4	17.99

The long way:

```
data longRotate(keep=Location Product Price);
set CcurrentPrice;
    Length Product $ 12;

    Product='Television';
    Price= Prod1;
    output;

    Product='Radio';
    Price=Prod2;
    output;

    Product='MicroWave';
    Price=Prod3;
    output;

    Product='Toaster';
    Price=Prod4;
    output;
run;
```

An easier way…

```
data ShortRotate(keep=Location Product Price);
Set CurrentPrice;
    Array Prod {*} _numeric_;
    Array Product_Names {4} $ 12
        ('Television', 'Radio', 'Microwave', 'Toaster');
    do j= 1 to dim(Prod);
        Product=Product_Names{j};
        Price=Prod{j};
    Output;
end;
run;
```

Temporary arrays

- A temporary array is an array that only exists for the duration of the data step where it is defined.
- Useful for storing constant values (for calculations).
- No corresponding variables to identify the array elements.
- The elements are defined by the key word _TEMPORARY_.

Example: Temporary arrays

- Have a data set that has some blood measures in conventional units. We wish to convert to SI units.

Example: Temporary arrays

```
data converted ;
 set raw ;
     array rawvals [3] Glucose HDL   Bilirubin ;
     array conv    [3] _temporary_ ( 0.0555 0.0259 17.1 ) ;
     array newvals [3] GlucSI  HDLSI BiliSI ;

     do j = 1 to 3 ;
         newvals[j] = rawvals[j] * conv[j] ;
     end;
     drop j Glucose HDL Bilirubin ;
 run ;
```

References

- Steve First and Teresa Schudrowitz. Arrays Made Easy: An Introduction to Arrays and Array Processing. Paper 242-30. SUGI 30 http://www2.sas.com/proceedings/sugi30/242-30.pdf
- Stephen Keelan, Off and Running with Arrays in SAS, SUGI 27, Paper 66-27 http://www2.sas.com/proceedings/sugi27/p066-27.pdf
- Howard Cherniak, Some Tips and Tricks, Vancouver SAS Users Group, Oct 31, 2007 http://www.sas.com/offices/NA/canada/downloads/presentations/Vancouver_Fall2007/Tips_Tricks.pdf


QUESTIONS
YOU HAVE?
memegenerator.net