



GCE SOLUTIONS

Siddharth Kumar, Principal Programmer

Add Derived Parameters using Multi-Dimensional Arrays



PARCAT1	PARAMCD	PARAM	PARAMN	PARCAT4	PARAMTYP	DTYPE
CHEMISTRY	ALP2U	ALP > 2 x ULN		ALPSI	DERIVED	COPY
CHEMISTRY	ALP3U	ALP > 3 x ULN		ALPSI	DERIVED	COPY
CHEMISTRY	ALP5U	ALP > 5 x ULN		ALPSI	DERIVED	COPY
CHEMISTRY	ALT10U	ALT > 10 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT20U	ALT > 20 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT3U	ALT > 3 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT5U	ALT > 5 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT8U	ALT > 8 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	AST10U	AST > 10 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST20U	AST > 20 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST3U	AST > 3 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST5U	AST > 5 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST8U	AST > 8 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	TBL1_5U	TBL > 1.5 x ULN		BILISI	DERIVED	COPY
CHEMISTRY	TBL2U	TBL > 2 x ULN		BILISI	DERIVED	COPY
CHEMISTRY	TBL3U	TBL > 3 x ULN		BILISI	DERIVED	COPY



Agenda

- ☐ The Syntax
- ☐ The Process
- ☐ Multidimensional array
- ☐ Situation – Solution
 - ☐ Laboratory Analysis Dataset
- ☐ Functions
- ☐ Conclusion



Introduction

- ❑ When do we use Arrays:
 - ✓ Perform same actions on multiple variables
 - ✓ Array allows to group a bunch of variables for the same process
 - ✓ The huge block of the repetitious statements and redundant calculation codes can be reduced to just a few lines
 - ✓ Code can be simplified with the use of arrays

- ❑ Derive / Update Treatment Variables in Analysis Dataset
- ❑ Derive / Update Analysis Flag Variables (anl01fl,..) in Analysis Dataset
- ❑ Concatenate and Apply Format to all the Treatment Variables for presentation in Output



The Syntax

array array-name {n} <\$> <length> array-elements <(initial values)>;

- ☐ array-name – Any valid SAS name that identifies the group of variables
- ☐ n – Number of elements within the array
- ☐ \$ - Indicates the elements within the array are character type variables
- ☐ Length – assigns length for the array elements
- ☐ Elements – List of SAS variables to be part of the array
- ☐ Initial values – Provides the initial values for each of the array elements.



Array Reference

array-name {subscript}

- ☐ Where *array-name* - is the name of an array that was previously defined with ARRAY statement in the same DATA step.
- ☐ *Subscript* - specifies the subscript, which can be a numeric constant, the name of a variable whose value is the number, a SAS numeric expression, or an asterisk (*).
- ☐ An array must be defined within the data step prior to being referenced.
- ☐ Array exists only for the duration of the data step in which they are defined



Multi Dimensional Array or Nested Array

- ☐ Multidimensional arrays are used when you want to group data or put values in a 'table like' format (i.e., rows and columns).
- ☐ The dimensions of arrays works like the following:
 - ✓ One-dimensional array: array x(cols)
 - ✓ Two-dimensional array: array y(rows, cols)
 - ✓ Three-dimensional array: array z(levels, rows, cols).
- ☐ The number of elements are placed in each dimension after the array name in the form {n, ..}.
- ☐ From right to left, the rightmost dimension represents columns; the next dimension represents rows. Each position farther left represents a higher dimension.



Multi Dimensional Array or Nested Array

`myArray {4,6} lab1-lab6 hem1-hem6 hist1-hist6 chem1-chem6;`

Lab1	Lab2	Lab3	Lab4	Lab5	Lab6
Hem1	Hem2	Hem3	Hem4	Hem5	Hem6
Hist1	Hist2	Hist3	Hist4	Hist5	Hist6
Chem1	Chem2	Chem3	Chem4	Chem5	Chem6

Variable	Array Reference
Hist3	<code>myArray{3,3}</code>
hem3	<code>myArray{2,3}</code>



Example of dataset when One-Dimensional & Two-Dimensional arrays are applied

```
data One;  
  array trt (3) tr01sdtm tr02sdtm tr03sdtm ;  
  array apd (3) ap01edtm ap02edtm ap03edtm;  
run;
```

tr01sdtm	tr02sdtm	tr03sdtm	ap01edtm	ap02edtm1	ap03edtm
.	.	.	.	.	.

```
data Two;  
  array x (3,2) tr01sdtm ap01edtm  
               tr02sdtm ap02edtm  
               tr03sdtm ap03edtm;  
run;
```

tr01sdtm	ap01edtm	tr02sdtm	ap02edtm	tr03sdtm	ap03edtm
.	.	.	.	.	.



Do Loop

```
Do i = 1 to 4; * row;  
    do j = 1 to 6; * column
```

```
    if myArray[i , j] > 80 then  
        myArray[i , j] = . ;  
    end;  
end;
```

- Multidimensional arrays are usually processed inside nested Do loops.

- A do loop is needed for each dimension
- one for the rows (which is represented by i and set from 1 to 4) . This Do loop processes the inner Do loop four times.
- one for the columns (represented by j and set from 1 to 6). This Do loop applies the derivation to all the variables in one row.
- Note, if you make i reference the rows (1 to 4), that i is put in the first position in the array reference.
- An array reference can use two or more index variables as the subscript to refer to two or more dimensions of an array.



Programming Specifications

PARCAT1	PARAMCD	PARAM	PARAMN	PARCAT4	PARAMTYP	DTYPE
CHEMISTRY	ALP2U	ALP > 2 x ULN		ALPSI	DERIVED	COPY
CHEMISTRY	ALP3U	ALP > 3 x ULN		ALPSI	DERIVED	COPY
CHEMISTRY	ALP5U	ALP > 5 x ULN		ALPSI	DERIVED	COPY
CHEMISTRY	ALT10U	ALT > 10 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT20U	ALT > 20 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT3U	ALT > 3 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT5U	ALT > 5 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	ALT8U	ALT > 8 x ULN		ALTSI	DERIVED	COPY
CHEMISTRY	AST10U	AST > 10 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST20U	AST > 20 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST3U	AST > 3 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST5U	AST > 5 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	AST8U	AST > 8 x ULN		ASTSI	DERIVED	COPY
CHEMISTRY	TBL1_5U	TBL > 1.5 x ULN		BILISI	DERIVED	COPY
CHEMISTRY	TBL2U	TBL > 2 x ULN		BILISI	DERIVED	COPY
CHEMISTRY	TBL3U	TBL > 3 x ULN		BILISI	DERIVED	COPY



Given Data

	SUBJID	AVISITN	PARAM	PARAMCD	PARAMTYP	PARCAT1	AVAL	AVALC	TRTA	R2ANRHI	R2ANRLO	
1	4001299	4	Albumin (g/L)	ALBSI		CHEMISTRY	45	45	Treatment	0.8653846154	1.2857142857	V
2	4001299	4	Alkaline phosphatase (U/L)	ALPSI		CHEMISTRY	60	60	Treatment	0.4651162791	1.5	V
3	4001299	4	Alanine aminotransferase (U/L)	ALTSI		CHEMISTRY	21	21	Treatment	0.512195122		V
4	4001299	4	Aspartate aminotransferase (U/L)	ASTSI		CHEMISTRY	18	18	Treatment	0.4864864865		V
5	4001299	4	Bilirubin (umol/L)	BILSI		CHEMISTRY	9	9	Treatment	0.4166666667	2.5	V
6	4001299	4	Creatinine (umol/L)	CREATSI		CHEMISTRY	97	97	Treatment	0.9166666667	1.5714285714	V
7	4001299	4	Phosphate (mmol/L)	PHOSSI		CHEMISTRY	1.07	1.07	Treatment	0.7333333333	1.2222222222	V
8	4001299	8	Albumin (g/L)	ALBSI		CHEMISTRY	41	41	Treatment	0.7884615385	1.1714285714	V
9	4001299	8	Alkaline phosphatase (U/L)	ALPSI		CHEMISTRY	66	66	Treatment	0.511627907	1.65	V
10	4001299	8	Alanine aminotransferase (U/L)	ALTSI		CHEMISTRY	18	18	Treatment	0.4390243902		V
11	4001299	8	Aspartate aminotransferase (U/L)	ASTSI		CHEMISTRY	21	21	Treatment	0.5675675676		V
12	4001299	8	Bilirubin (umol/L)	BILSI		CHEMISTRY	9	9	Treatment	0.4166666667	2.5	V
13	4001299	8	Creatinine (umol/L)	CREATSI		CHEMISTRY	120	120	Treatment	1.1333333333	1.9428571429	V
14	4001299	8	Phosphate (mmol/L)	PHOSSI		CHEMISTRY	1	1	Treatment	0.6888888889	1.1481481481	V
15	4001299	12	Albumin (g/L)	ALBSI		CHEMISTRY	41	41	Treatment	0.7884615385	1.1714285714	V
16	4001299	12	Alkaline phosphatase (U/L)	ALPSI		CHEMISTRY	61	61	Treatment	0.4728682171	1.525	V
17	4001299	12	Alanine aminotransferase (U/L)	ALTSI		CHEMISTRY	15	15	Treatment	0.3658536585		V
18	4001299	12	Aspartate aminotransferase (U/L)	ASTSI		CHEMISTRY	17	17	Treatment	0.4594594595		V
19	4001299	12	Bilirubin (umol/L)	BILSI		CHEMISTRY	9	9	Treatment	0.4166666667	2.5	V
20	4001299	12	Creatinine (umol/L)	CREATSI		CHEMISTRY	107	107	Treatment	1.0083333333	1.7285714286	V
21	4001299	12	Phosphate (mmol/L)	PHOSSI		CHEMISTRY	1.19	1.19	Treatment	0.8222222222	1.3703703704	V
22	4001299	16	Albumin (g/L)	ALBSI		CHEMISTRY	44	44	Treatment	0.8461538462	1.2571428571	V
23	4001299	16	Alkaline phosphatase (U/L)	ALPSI		CHEMISTRY	65	65	Treatment	0.503875969	1.625	V
24	4001299	16	Alanine aminotransferase (U/L)	ALTSI		CHEMISTRY	16	16	Treatment	0.3902439024		V
25	4001299	16	Aspartate aminotransferase (U/L)	ASTSI		CHEMISTRY	17	17	Treatment	0.4594594595		V
26	4001299	16	Bilirubin (umol/L)	BILSI		CHEMISTRY	5	5	Treatment	0.25	1.5	V
27	4001299	16	Creatinine (umol/L)	CREATSI		CHEMISTRY	99	99	Treatment	0.9333333333	1.6	V
28	4001299	16	Phosphate (mmol/L)	PHOSSI		CHEMISTRY	0.94	0.94	Treatment	0.6444444444	1.0740740741	V





The Code

```
data dlb2;
  set dlb1;
  array _param(4,5) $200. ('ALT > 3 x ULN'      'ALT > 5 x ULN' 'ALT > 8 x ULN' 'ALT > 10 x ULN' 'ALT > 20 x ULN'
                          'AST > 3 x ULN'      'AST > 5 x ULN' 'AST > 8 x ULN' 'AST > 10 x ULN' 'AST > 20 x ULN'
                          'TBL > 1.5 x ULN'     'TBL > 2 x ULN' 'TBL > 3 x ULN' ''                ''
                          'ALP > 2 x ULN'      'ALP > 3 x ULN' 'ALP > 5 x ULN' ''                '' );
  array _paramcd(4,5) $8. ('ALT3U'   'ALT5U' 'ALT8U' 'ALT10U' 'ALT20U'
                          'AST3U'   'AST5U' 'AST8U' 'AST10U' 'AST20U'
                          'TBL1_5U' 'TBL2U' 'TBL3U' ''        ''
                          'ALP2U'   'ALP3U' 'ALP5U' ''        '' );
  array _ratio_limit(4,5) (3      5 8 10 20
                          3      5 8 10 20
                          1.5  2 3  .  .
                          2      3 5  .  . );
  array _parcat4 (4) $20. ('ALTSI' 'ASTSI' 'BILISI' 'ALPSI');
  output;
  AVAL=.;
  PARAMTYP='DERIVED'; DTYPE='COPY';

  do j=1 to 4;
    if paramcd=_parcat4(j) then do;

      do i=1 to 5;
        if _param(j,i) ne '' then do;
          param=_param(j,i);
          paramcd=_paramcd(j,i);
          if R2ANRHI>_ratio_limit(j,i)>. then do; avalc = 'Y'; aval = 1; end;
          else if .<R2ANRHI<=_ratio_limit(j,i) then do; avalc = 'N'; aval = 0; end;
          output;
        end;
      end;
    end;
  end;

run;
```

Do loop for the rows (which is represented by j and set from 1 to 4)

Do loop for the column (which is represented by i and set from 1 to 5)



The Output

Added Data

SUBJID	AVISITN	PARAM	PARAMCD	PARAMTYP	PARCAT1	AVAL	AVALC	R2ANRHI	R2ANRLO	TRTA	DTYPE
4001299	4	Albumin (g/L)	ALBSI		CHEMISTRY	45	45	0.8653846154	1.2857142857	Treatment	
4001299	4	Alkaline phosphatase (U/L)	ALPSI		CHEMISTRY	60	60	0.4651162791	1.5	Treatment	
4001299	4	ALP > 2x ULN	ALP2U	DERIVED	CHEMISTRY	0	N	0.4651162791	1.5	Treatment	COPY
4001299	4	ALP > 3x ULN	ALP3U	DERIVED	CHEMISTRY	0	N	0.4651162791	1.5	Treatment	COPY
4001299	4	ALP > 5x ULN	ALP5U	DERIVED	CHEMISTRY	0	N	0.4651162791	1.5	Treatment	COPY
4001299	4	Alanine aminotransferase (U/L)	ALTSI		CHEMISTRY	21	21	0.512195122	.	Treatment	
4001299	4	ALT > 3x ULN	ALT3U	DERIVED	CHEMISTRY	0	N	0.512195122	.	Treatment	COPY
4001299	4	ALT > 5x ULN	ALT5U	DERIVED	CHEMISTRY	0	N	0.512195122	.	Treatment	COPY
4001299	4	ALT > 8x ULN	ALT8U	DERIVED	CHEMISTRY	0	N	0.512195122	.	Treatment	COPY
4001299	4	ALT > 10x ULN	ALT10U	DERIVED	CHEMISTRY	0	N	0.512195122	.	Treatment	COPY
4001299	4	ALT > 20x ULN	ALT20U	DERIVED	CHEMISTRY	0	N	0.512195122	.	Treatment	COPY
4001299	4	Aspartate aminotransferase (U/L)	ASTSI		CHEMISTRY	18	18	0.4864864865	.	Treatment	
4001299	4	AST > 3x ULN	AST3U	DERIVED	CHEMISTRY	0	N	0.4864864865	.	Treatment	COPY
4001299	4	AST > 5x ULN	AST5U	DERIVED	CHEMISTRY	0	N	0.4864864865	.	Treatment	COPY
4001299	4	AST > 8x ULN	AST8U	DERIVED	CHEMISTRY	0	N	0.4864864865	.	Treatment	COPY
4001299	4	AST > 10x ULN	AST10U	DERIVED	CHEMISTRY	0	N	0.4864864865	.	Treatment	COPY
4001299	4	AST > 20x ULN	AST20U	DERIVED	CHEMISTRY	0	N	0.4864864865	.	Treatment	COPY
4001299	4	Bilirubin (umol/L)	BILSI		CHEMISTRY	9	9	0.4166666667	2.5	Treatment	
4001299	4	TBL > 1.5x ULN	TBL1_5U	DERIVED	CHEMISTRY	0	N	0.4166666667	2.5	Treatment	COPY
4001299	4	TBL > 2x ULN	TBL2U	DERIVED	CHEMISTRY	0	N	0.4166666667	2.5	Treatment	COPY
4001299	4	TBL > 3x ULN	TBL3U	DERIVED	CHEMISTRY	0	N	0.4166666667	2.5	Treatment	COPY
4001299	4	Creatinine (umol/L)	CREATSI		CHEMISTRY	97	97	0.9166666667	1.5714285714	Treatment	
4001299	4	Phosphate (mmol/L)	PHOSSI		CHEMISTRY	1.07	1.07	0.7333333333	1.2222222222	Treatment	
4001299	8	Albumin (g/L)	ALBSI		CHEMISTRY	41	41	0.7884615385	1.1714285714	Treatment	
4001299	8	Alkaline phosphatase (U/L)	ALPSI		CHEMISTRY	66	66	0.511627907	1.65	Treatment	
4001299	8	ALP > 2x ULN	ALP2U	DERIVED	CHEMISTRY	0	N	0.511627907	1.65	Treatment	COPY
4001299	8	ALP > 3x ULN	ALP3U	DERIVED	CHEMISTRY	0	N	0.511627907	1.65	Treatment	COPY
4001299	8	ALP > 5x ULN	ALP5U	DERIVED	CHEMISTRY	0	N	0.511627907	1.65	Treatment	COPY





Functions: Dim, Lbound, Hbound

- Determining the Number of Elements in an Array Efficiently
 - DIM returns the number of elements in an array dimension.
 - HBOUND returns the value of the upper bound of an array dimension.
 - LBOUND returns the value of the lower bound of an array dimension.
- form of the DIM function is: **DIM***n*(array-name)
- LBOUND function: **LBOUND***n*(array-name)
- HBOUND functions :**HBOUND***n*(array-name)
- where *n* is the specified dimension that has a default value of 1.

Example: array mult{2:6,4:13,2} mult1-mult100;

Syntax	Alternative Syntax	Value
HBOUND(MULT)	HBOUND(MULT,1)	6
HBOUND2(MULT)	HBOUND(MULT,2)	13
HBOUND3(MULT)	HBOUND(MULT,3)	2



Use of Functions DIM, LBOUND & HBOUND IN MULTI-DIMENSIONAL ARRAY

```
data dlb2_func;
set adlb;
array _param(4,5) $200. ('ALT > 3 x ULN'      'ALT > 5 x ULN' 'ALT > 8 x ULN' 'ALT > 10 x ULN' 'ALT > 20 x ULN'
                        'AST > 3 x ULN'      'AST > 5 x ULN' 'AST > 8 x ULN' 'AST > 10 x ULN' 'AST > 20 x ULN'
                        'TBL > 1.5 x ULN'     'TBL > 2 x ULN' 'TBL > 3 x ULN' ''
                        'ALP > 2 x ULN'      'ALP > 3 x ULN' 'ALP > 5 x ULN' ''
                        );
array _paramcd(4,5) $8. ('ALT3U'  'ALT5U' 'ALT8U' 'ALT10U' 'ALT20U'
                       'AST3U'  'AST5U' 'AST8U' 'AST10U' 'AST20U'
                       'TBL1_5U' 'TBL2U' 'TBL3U' ''
                       'ALP2U'  'ALP3U' 'ALP5U' ''
                       );
array _ratio_limit(4, 5) (3  5 8 10 20
                          3  5 8 10 20
                          1.5 2 3 . .
                          2  3 5 . .);
array _parcat4 (4) $20. ('ALTSI' 'ASTSI' 'BILISI' 'ALPSI');
output;
AVAL=.;
PARAMTYP='DERIVED'; DTYPE='COPY';

do j=1 to dim(_paramcd, 1);
  if paramcd=_parcat4(j) then do;

    do i= lbound(_param,2) to hbound(_param,2) ;
      if _param(j,i) ne '' then do;
        param=_param(j,i);
        paramcd=_paramcd(j,i);
        if R2ANRHI>_ratio_limit(j,i)>. then do; avalc = 'Y'; aval = 1; end;
        else if .<R2ANRHI<=_ratio_limit(j,i) then do; avalc = 'N'; aval = 0; end;
        output;
      end;
    end;
  end;
end;

run;
```

Use of function DIM

Use of function LBOUND & HBOUND



Conclusion

❑ INNOVATION

❑ AUTOMATION

- ✓ Easier to maintain and update
- ✓ Easier to add new criteria

❑ MOTIVATION

- Minimize the Code, Save Time and Efforts
- Effective use of Multi-Dimensional Arrays or Nested Arrays can increase EFFICIENCY of program.



THANK YOU

Any QUESTIONS