ARRAYs and DO OVER Loops[1]

ARRAYs and DO OVER loops are a way of programming more efficiently. Using them can save writing many lines of code, can reduce the risk of error, and can make error detection and correction easier.

```
* array1.sas ;
title1 'array1.sas' ;
options compress = yes nodate ;

data one;
input v1 - v3  ;
cards ;
.2 .3 .4
.5 .6 .321
.21 .3 .4
.15 .36 .13
;
data two; set one ;

** Create variables without an array and do over loop ;
** With 3 variables - it doesn't make much difference - ;
** but imagine transforming MANY variables.  You would ;
** have to type A LOT of lines - and could make many mistakes. ;
a1 = v1 * 100 ;
a2 = v2 * 100 ;
a3 = v3 * 100 ;

*** A simple example of ARRAYS and DO OVER *** ;

** Create new variables in an array from ;
** Existing variables - Also in an array. ;
**** When processing multiple arrays, it is IMPORTANT ;
**** to make check and make sure that you have the same ;
**** number of variables in the arrays - and that the variables;
**** correspond to each other as you expect think they do. ;
** Each variable in the array is called an "element". ;

** Create new variables that are the original variables multiplied by 100. ;

** The syntax is: ARRAY (array name) (list of variables included in array) ;

array orig  v1 - v3 ;
array perc  p1 - p3 ;

** The arrays can be processed with a DO OVER loop. ;
** Within the DO OVER LOOP, when an ARRAY NAME is referred to, ALL of the ;
** variables in the LIST OF VARIABLES INCLUDED IN ARRAY are processed;

do over orig ;
 perc = orig * 100 ;
end;
proc print ;
run ;
```

array1.sas

Obs	v1	v2	v3	a1	a2	a3	p1	p2	p3
1	0.20	0.30	0.400	20	30	40.0	20	30	40.0
2	0.50	0.60	0.321	50	60	32.1	50	60	32.1
3	0.21	0.30	0.400	21	30	40.0	21	30	40.0
4	0.15	0.36	0.130	15	36	13.0	15	36	13.0

---

[1]**Prepared by Patty Glynn, University of Washington. May 1, 2001, updated 6/15/02** C:\all\help\helpnew\array.wpd