Center for Family and Demographic Research

Advanced SAS Topics: SAS Arrays

Presented by the CFDR

Instructor: Meredith Porter

June 16, 2008

Center for Family and Demographic Research

What is an Array?

- Array

  array fun (5) var1 var2 var3 var4 var5;

Center for Family and Demographic Research

What is an Array?

- A SAS array
  - is a temporary grouping of SAS variables that are arranged in a particular order and identified by an **array-name**
  - exists only for the duration of the current DATA step
  - is not a variable

-(SAS online documentation)

What is an Array?

- Basic array processing involves the following steps:
  - grouping variables into arrays
  - selecting a current variable for an action
  - repeating an action

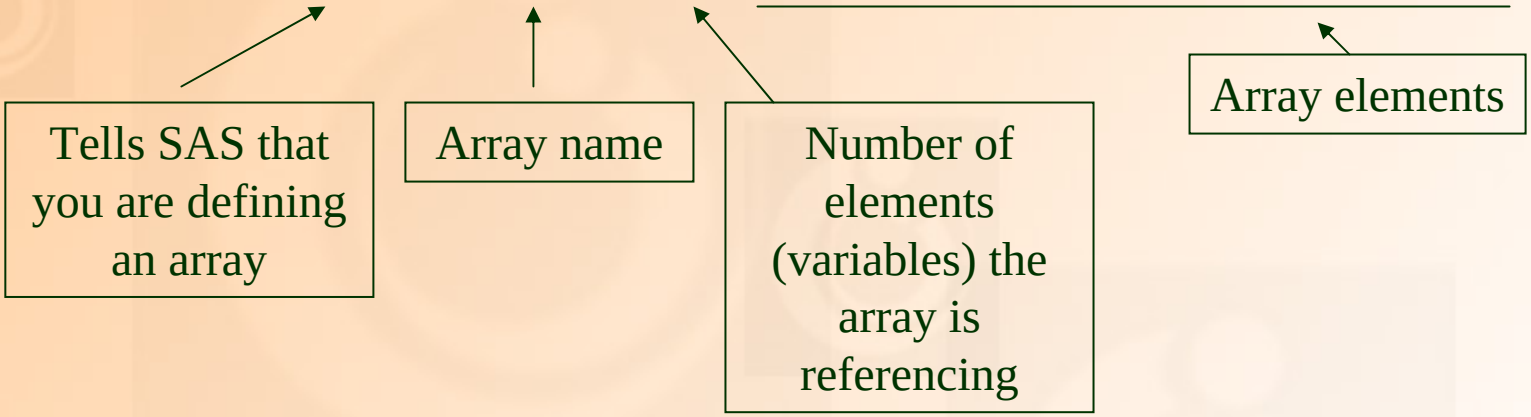
-(SAS online documentation)

Center for Family and Demographic Research

What is an Array?

- SAS arrays are defined by using an array statement…

```
array fun (5) var1 var2 var3 var4 var5;
```



What is an Array?

array fun (5) var1 var2 var3 var4 var5;

- After defining this array…
  - fun(1) refers to var1
  - fun(2) refers to var2
  - fun(3) refers to var3
  - fun(4) refers to var4
  - Can you guess what fun(5) refers to?

What is an Array?

- Arrays can be defined in different ways with the same results…

Center for Family and Demographic Research

What is an Array?

```
array fun (5) var1 var2 var3 var4 var5;
array fun [5] var1 var2 var3 var4 var5;
array fun {5} var1 var2 var3 var4 var5;

array fun (1-5) var1 var2 var3 var4 var5;

array fun (5) var1-var5;

array fun (*) var1 var2 var3 var4 var5;
```

Center for Family and Demographic Research

What is an Array?

- If you want an array that includes all numeric variables:
  - array a(*) _numeric_;
- If you want an array that includes all character variables:
  - array b(*) _character_;
- If you want an array that includes all variables:
  - array c(*) _all_;

Center for Family and Demographic Research

What is an Array?

- Elements of an array can be variables that already exist OR variables that you will create during the data step.

```
array fun (5) var1 var2 var3 var4 var5;
array funa (5) var1a var2a var3a var4a var5a;
```

What is an Array?

- So far, this doesn't look so bad, right?
- What about those i's, j's, do's…?
  - These will come when we actually make these array statements do some work for us!

…and why should we make arrays work for us?

Center for Family and Demographic Research

Why use SAS Arrays?

- Write less code!
  - Repeat an action/set of actions on multiple variables
- Restructure SAS data sets

Example1: Recode missing values

- Let's say you have ten variables that are coded [98 = n/a] [99 = refused]. You want to recode the 98's and 99's for these variables to (.)
- You could do it this way…

Example1: Recode missing values

```
if var1 = 98 or var1 = 99 then var1 = .;
if var2 = 98 or var2 = 99 then var2 = .;
if var3 = 98 or var3 = 99 then var3 = .;
if var4 = 98 or var4 = 99 then var4 = .;
if var5 = 98 or var5 = 99 then var5 = .;
if age = 98 or age = 99 then age = .;
if marstat = 98 or marstat = 99 then marstat =.;
…
```

Example1: Recode missing values

- You can be a super SAS coder…just use an array.

Center for Family and Demographic Research

Example1: Recode missing values

```
(1)  array v (10) var1-var5 age marstat race mom dad;
(2)  do i=1 to 10;
(3)      if v(i)=98 or v(i)=99 then v(i)= . ;
(4)                  end;
```

…but what does this all mean?

Center for Family and Demographic Research

Example1: Recode missing values

(1) **array v (10) var1-var5 age marstat race mom dad;**

- Tells SAS that there is an array called v. It references 10 variables: var1, var2, var3, var4, var5, age, marstat, race, mom, dad.

Example1: Recode missing values

(2) **do i=1 to 10;**

- Sets an index (i), so that SAS will run through any subsequent steps for each of the ten variables until it hits the end statement.

Center for Family and Demographic Research

Example1: Recode missing values

(3) **if v(i)=98 or v(i)=99 then v(i)= . ;**

- This is the same as saying:
  - If variable 1 of the array (var1) = 98 or variable 1 of the array (var1) = 99 then variable 1 of the array (var1) = .
  - If variable 2 of the array (var2) = 98 or variable 2 of the array (var2) = 99 then variable 2 of the array (var2) = .
  - …
  - If variable 10 of the array (dad) = 98 or variable 10 of the array (dad) = 99 then variable 10 of the array (dad) = .

(4) **end;**

- Ends the do statement.

Center for Family and Demographic Research

Example 2: Recode all numeric missing values

```
(1) array  vnum (*) _numeric_ ;
(2)     do i=1 to dim(vnum);
(3)             if vnum(i) in(98, 99) then vnum(i)=.;
(4)     end;
```

Example 2: Recode all numeric missing values

(1) **array vnum (*) _numeric_ ;**

- The array ‘vnum’ is referencing all the numeric variables in your dataset.
- Using the “*” means you don’t have to count the number of numeric variables…SAS will do it for you.

Example 2: Recode all numeric missing values

(2) **do i=1 to dim(vnum);**

- Tells SAS to do subsequent commands starting at the first numeric variable and going to the last one (dim is for dimension).

Example 2: Recode all numeric missing values

(3) **if vnum(i) in(98, 99) then vnum(i)=.;**

- Goes through each numeric variable (because of "do i = 1…"). If the variable equals 98 or 99, then that variable is recoded to "."

(4) **end;**

- Ends the do loop.

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

- Let's suppose that "Jess" is looking at currently cohabiting women. For each R, she wants to know whether each of her pregnancies (if any) occurred during the current cohabiting union.

Example 3: Create new variables based on specifications

- She has variables:
  - *prgbeg01-prgbeg19*: century month pregnancy began for each of 19 possible pregnancies (missing if no pregnancy)
  - *startco*: century month current cohabiting union began (some have missing values)

- She wants to create dummy variables cpreg01-cpreg19 to indicate whether that pregnancy occurred during a cohabiting union.

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

- She could it this way…

```
if prgbeg01 ge startco then cpreg01 = 1;
  else cpreg01 = 0;
if prgbeg02 ge startco then cpreg02 = 1;
  else cpreg02 = 0;
if prgbeg03 ge startco then cpreg03 = 1;
  else cpreg03 = 0;
if prgbeg04 ge startco then cpreg04 = 1;
  else cpreg04 = 0;
…
```

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

- …but Jess is so clever. She's going to use an array.

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

```
(1)data mywork.three; set mywork.two; where rmarital = 2;
(2)  array pregbeg (*) prgbeg01-prgbeg19; /* existing vars */
(3)  array cp (*) cpreg01-cpreg19; /* new variables */
(4)    retain prgbeg01-prgbeg19 cpreg01-cpreg19 i;
(5)           do i = 1 to 19;
(6)       if pregbeg(i) ge startco then cp(i) = 1; else cp(i)=0;
(7)       drop i;
(8)           end;
```

Example 3: Create new variables based on specifications

```
(1) data mywork.three; set mywork.two; where rmarital = 2;
```

- This is indicating the data step…tells SAS which data we are using, and under what conditions.

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

```
(2)array pregbeg (*) prgbeg01-prgbeg19; /* existing vars */
```

- We are creating an array called “pregbeg” and telling SAS which variables are part of the array (our existing variables indicating when pregnancy began).

```
(3) array cp (*) cpreg01-cpreg19; /* new variables */
```

- We are creating another array called “cp” and telling SAS which variables are part of this array (the dummy variables we want to create cpreg01-cpreg19).

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

```
(4) retain prgbeg01-prgbeg19 cpreg01-cpreg19 i;
```

- The retain statement is used to hold the values of variables across iterations of the data step. Normally, all variables in the data step are set to missing at the start of each iteration of the data step.

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

```
(5) do i = 1 to 19;
```

- This sets our index…SAS will do subsequent steps (until the end statement) 19 times.

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

```
(6) if pregbeg(i) ge startco then cp(i)= 1; else cp(i) = 0;
```

- SAS will go through this 19 times…
  - If pregbeg(1) ge startco . then cp(1) = 1; else cp(1) = 0;
  - If pregbeg(2) ge startco . then cp(2) = 1; else cp(2) = 0;
  - …to make sense of this, remember what the arrays are referencing!

Center for Family and Demographic Research

Example 3: Create new variables based on specifications

```
(7) drop i;
```

- Do this so that index variable (i) does not become a variable in data set.

```
(8) end;
```

- Ends do loop.

Example 4: Create a respondent (Woman) level file from a pregnancy file

- "Steve" has decided that family demography is where it's at. He wants to look at women's pregnancy outcomes and union status at pregnancy outcomes in the NSFG.

I'm sick of all this deviance stuff.

Center for Family and Demographic Research

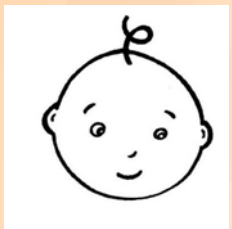
Example 4: Create a respondent (Woman) level file from a pregnancy file

- The Female Respondent file contains one record for each of the 7,643 women in the survey and includes most information from the questionnaires. It is structured like this example:

CASEID	VAR1	VAR2	VAR3
0001	1	5	0
0002	1	2	0
0003	0	3	1
0004	0	.	1
0005	1	1	0

Example 4: Create a respondent (Woman) level file from a pregnancy file

- The Female Interval (pregnancy) file contains one record for each of the 13,593 pregnancies (current and past), along with other related characteristics. It is structured like this example:



CASEID (Respondent ID Number)	PREGORDR (Number)	RMAROUT6 (Marital status)	VAR 4
0001	1	1	1
0001	2	2	1
0001	3	2	0
0004	1	1	1
0005	1	5	1
0005	2	6	0

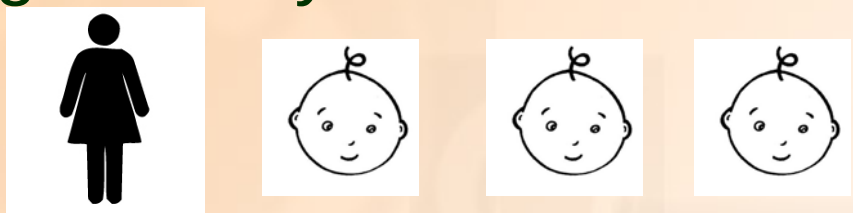
Example 4: Create a respondent (Woman) level file from a pregnancy file

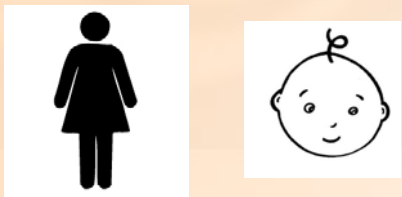
- What can Steve do?

Never mind...

Example 4: Create a respondent (Woman) level file from a pregnancy file

- One solution: Create a RESPONDENT (Woman) LEVEL file, with information for each pregnancy (where applicable) attached, using an array.

R1 

R4 

Center for Family and Demographic Research

Example 4: Create a respondent (Woman) level file from a pregnancy file

- Steve needs to create variables for each woman to indicate the outcome of each pregnancy (if applicable) and their marital status at the outcome of each pregnancy (if applicable).

Center for Family and Demographic Research

Example 4: Create a respondent (Woman) level file from a pregnancy file

- The respondents have varying numbers of pregnancies, so he has to create outcome variables for each potential pregnancy (pregnancy 1, pregnancy 2, pregnancy 3...through to the highest number of pregnancies in the data).

Example 4: Create a respondent (Woman) level file from a pregnancy file

- After the woman's last pregnancy, these variables will just take on a missing value. In this example, the greatest number of pregnancies for any woman is 19.

Center for Family and Demographic Research

```
(1)    libname NSFG6 'T:\Public\Data\NSFG\Wave
       6\SAS datasets';

(2)    libname mywork 'C:\My Documents\My SAS
       Files';
```

- Line 1 references the library “NSFG.” This is where we will get our data.
- Line 2 references the library “mywork.” This is where we will store our created data set.

```
(3)   data mywork.fpreg (keep=caseid pregordr
      prgoutcome01-prgoutcome19 rmarout01-
      rmarout19);
```

- Line 3 names the new data set “mywork.fpreg”. The “keep =” tells SAS the variables to keep in this data set, including variables that are already created and ones that will be created with the array.

```
(4)   set nsfg6.preg;
(5)   by caseid;
(6)   retain caseid pregordr prgoutcome01-
      prgoutcome19 rmarout01-rmarout19 i ;
```

- Line 4 sets the new dataset to equal the existing dataset NSFG6.preg.
- Line 5 indicates that observations should be ordered according to the variable CASEID.
- Line 6 holds the values over throughout the iterations of the data step.

```
(7)    array p(*) prgoutcome01-prgoutcome19 ;
(8)    array r(*) rmarout01-rmarout19;
```

- Line 7 creates an array called “p” which refers to a series of variables that we will be creating called prgoutcome01, prgoutcome02, etc.

- Line 8 does the same as above, but creates variables for the respondent’s marital status at the outcome of each pregnancy.

```
(9)   if first.caseid then do ;
(10)        do j= 1 to 19 ;
(11)        p(j)= . ;
(12)        r(j)= . ;
(13)        end ;
```

- Line 9 tells SAS to do the following as soon as it reaches a new and unique respondent ID.
- Line 10 tells SAS to do something 19 times, beginning with the number 1 and ending with the number 19.
- Lines 11 and 12 tell SAS to initialize each of the new variables (**`prgoutcome01-prgoutcome19`** & **`rmarout01-rmarout19`**) we are creating with the arrays to missing to start.
- Line 13 ends this initialization.

```
(14)  i= 1 ;
(15)  end ;
```

- Line 14 begins the index at 1.
- Line 15 tells SAS that you are finished with this step.

```
(16)  p(i)=prgoutcome;
(17)  r(i)=rmarout6;
```

There is a variable indicating preg outcome for each pregnancy

There is a variable indicating marital status of woman at each the end of each pregnancy

- Lines 16 tells SAS to set the first pregnancy to the value of the first array. Prgoutcome is the name of the variable indicating the outcome of each pregnancy in the pregnancy file.

- Line 17 tells SAS to set the first marital status to the value of the first array. Rmarout6 is the name of the variable indicating the marital status at the outcome of the pregnancy.

```
(18)  i=i+1 ;
(19)  if last.caseid then output ;
(20)  run ;
```

- Line 18 moves the index forward through all of the women's pregnancies.
- Line 19 tells SAS to output the variables once it reaches the last pregnancy.

- Remember, in the pregnancy file, each pregnancy makes up its own record.
- For each woman, we are going through and attaching her pregnancy information "horizontally" onto her record.

CASEID	VAR1	VAR2	VAR3
0001	1	5	0
0002	1	2	0
0003	0	3	1
0004	0	.	1
0005	1	1	0

So, when we say r(i)=rmarout6, we are telling SAS that rmarout01=rmarout6 for the first pregnancy record…

CASEID (Respondent ID Number)	PREGORDR (Number)	RMAROUT6 (Marital status)	prgoutcome VAR 4
0001	1	1	1
0001	2	2	1
0001	3	2	0
0004	1	1	1
0005	1	5	1
0005	2	6	0

Center for Family and Demographic Research

- Steve's data will be structured like the following example:

CASEID	PREGORDR	PRGOUTCOME01	PRGOUTCOME02	...	PRGOUTCOME19	RMAROUT01	RMAROUT02	...	RMAROUT19
1	3	1	1		.	1	2		.
4	1	1	.		.	1	.		.
5	2	1	1		.	5	6		.

- Use proc print or click on data set in Explorer window to see data in table format.

Center for Family and Demographic Research

- Still feeling some array anxiety?
  - It gets easier with practice!
  - Use examples as templates…
  - See the CFDR webpage for help

  http://www.bgsu.edu/organizations/cfdr/page36022.html