AN ARRAY OF POSSIBILITIES: USING ARRAYS TO MANIPULATE LONGITUDINAL SURVEY DATA

Lakhpreet "Preeti" Gill
Senior Programmer Analyst
Mathematica Policy Research

OVERVIEW

- Review the data challenges and research needs
- Create an index based on respondent specific baseline and analysis waves
- Foundation for using the index to create variables
- Manipulate the index to select information for later waves
- Troubleshoot "out of bounds" cases
- Integrate the subscript with the index to populate time series variables
  - Pegged to an existing variable value
  - Arrays of different element numbers
- Merge wide and long data to "look ahead" and "look across"

Material covered assumes some basic knowledge of using arrays.

WHY ARRAYS FOR LONGITUDINAL SURVEY DATA?

Data structure

- **Wide**: each set of interviews is represented by variables *added* to a record. A new record is not created.

Study time period

- **Respondents enter the survey at different time points**: must identify correct wave for each respondent

PERSON ID	AGE_1	REGION_1	AGE_2	REGION_2
1				
2				
3				

HEALTH AND RETIREMENT STUDY

Data example

- Complex sample survey design
- Multiple waves on each person record
- Interviews occur every two years
- Linking to related data sets with person/year data structures

Research example

- Need to create variables that draw from different waves at baseline and analysis for each respondent
- Baseline years from 1992 to 2004
  - Waves 1 through 7
- Analysis years from 1998 to 2012
  - Wave 4 through 11

CREATE A VARIABLE FOR THE ARRAY INDEX

```
ARRAY RELAGE(11) R_RELAGE_1   - R_RELAGE_11;
ARRAY STATUS(11) R_RIWSTAT_1 - R_RIWSTAT_11;


DO K = 1 TO 11;
      IF NOT MISSING(B_INDX) THEN LEAVE;
       IF 58 <= RELAGE{K}<= 60 AND
      STATUS{K} = 1 THEN B_INDX= K;
END;
```

← *Range matches the number of waves in the survey*

← *Once assigned, leave the loop. Do not want a later wave assigned if condition is still met.*

Year/wave	B_INDX
1992/1	1
1994/2	2
1996/3	3
1998/4	4
2000/5	5
2002/6	6
2004/7	7

CREATE A VARIABLE FOR THE ARRAY INDEX

Cross-tab illustration

BASICS TO CREATE ANALYTIC VARIABLES

```
ARRAY MSTAT   (11) R_RMSTAT_1  - R_RMSTAT_11;

_MSTATCHK = MSTAT{B_INDX};
B_MSTATMAR = MSTAT{B_INDX}  IN (1,2,3);
B_MSTATDIV = MSTAT{B_INDX} IN (4,5,6);
B_MSTATWID = MSTAT{B_INDX} = 7;
B_MSTATNEV = MSTAT{B_INDX} = 8;
IF MISSING (MSTAT{B_INDX}) THEN CALL MISSING(OF
B_MSTAT:);
```

Raw variable

B_INDX	R_RMSTAT_1	R_RMSTAT_2	R_RMSTAT_3	R_RMSTAT_4
3	.	.	1	1
4	.	2	2	5

Analysis variable

B_INDX	B_MSTATMAR	B_MSTATDIV	B_MSTATDWID	B_MSTATNEV
3	1	0	0	0
4	0	1	0	0

BASICS TO CREATE THE ANALYTIC VARIABLES

Data illustration

INCREMENT THE ARRAY INDEX VARIABLE

```
ARRAY STATUS     (11) R_RIWSTAT_1 - R_RIWSTAT_11;


B_STAT           = STATUS{B_INDX};
B_STATNXT        = STATUS{B_INDX + 1};
```

← *Use a calculation to get the variables for the wave after baseline*

Manipulating the array index can take the array out of bounds. This usually means the index is too large for the number of elements specified with the array.

Correct by conditioning out those cases:
IF B_INDX NE 11 THEN B_STATNXT = STATUS{B_INDX + 1};

CAUTIONARY! OUT OF BOUNDS #1

Data challenge

OUT OF BOUNDS #2

```
ARRAY ADLA (2:11) R_RADLA_2 - R_RADLA_11 ;
```

← *Survey variable starts in wave 2, but some respondents reached baseline in wave 1.*

```
IF B_INDX NE 1 THEN DO;
        _ADLABCHK = ADLA{B_INDX};
        B_ADLA = ADLA{B_INDX} > 0;
        IF MISSING (ADLA{B_INDX}) THEN B_ADLA = .;
END;

ELSE IF B_INDX EQ 1 THEN DO;
        _ADLAB1CHK = ADLA{2};
        B_ADLA = ADLA{2} > 0;
        IF MISSING (ADLA{2}) THEN B_ADLA = .;
END;
```

← *First process data for wave 2 and greater.*

← *Then process data for wave 1.*

Raw variable

B_INDX	R_RADLA_2	R_RADLA_3	R_RADLA_4
3	0	2	2
1	0	0	0

Analysis variable

B_INDX	_ADLABCHK	_ADLAB1CHK	B_ADLA
3	2	.	1
1	.	0	0

OUT OF BOUNDS

Data illustration

INTEGRATING ARRAYS SUBSCRIPTS WITH INDEXES AND VARIABLES

```
ARRAY CY (1997:2012) CY_1997 - CY_2012;

DO I = 1997 TO 2012;

  IF ANALYSIS_YR = I THEN CY{I} = 1;
    ELSE CY{I} = 0;

END;
```

Subscript, index and variable have the same values.

Array subscript: Defines the range of elements in the array.

Array index: Used in the array reference to call an element from the array. Must equal the subscript to avoid an "out of bounds" error.

Variable: Values are compared to the array index.

ANALYSIS_YR	CY_1997	CY_1998	CY_1999	CY_2000
1997	1	0	0	0
1998	0	1	0	0
1999	0	0	1	0
2000	0	0	0	1

INTEGRATING ARRAYS SUBSCRIPTS WITH INDEXES AND VARIABLES

Cross tab illustration

HANDLING ARRAYS OF DIFFERENT LENGTHS

Creates 6 variables with array name prefix and array element suffix.

```
ARRAY HITOT (11)R_HHITOT_1 - R_HHITOT_11;
ARRAY HHINC_SERIES (6);

J = 1;

DO I = B_INDX TO A_INDX;
      HHINC_SERIES {J} = HITOT {I};
      J + 1;
END;
```

Two array indexes that increment differently

I = baseline year to analysis year for raw survey variables

J = Range of time between baseline and analysis for new variable series

Raw variable

B_INDX	A_INDX	R_HHITOT_3	R_HHITOT_4	R_HHITOT_5	R_HHITOT_6
4	6	9,500	10,000	15,000	20,0000
3	5	400,000	450,000	500,000	500,000

Analysis variable

B_INDX	A_INDX	HHINC_SERIES1	HHINC_SERIES2	HHINC_SERIES3	HHINC_SERIES4-HHINC_SERIES6
4	6	10,000	15,000	20,0000	.
3	5	400,000	450,000	500,000	.

HANDLING ARRAYS OF DIFFERENT LENGTHS

Data illustration

Administrative data

ID	YEAR	PAY1	PAY2	PAY3	PAY4	PAY5	PAY6	PAY7	PAY8	PAY9	PAY10	PAY11	PAY12
111	1996	.	.	.	.	.	.	.	600	600	600	600	600
111	1997	650	650	650	650	650	650	650	650	650	650	650	650
111	1998	700	700	700	700	700	700	700	700	700	700	700	700

Survey data

ID	B_WLTH	B_INCOME	B_EMPLOY
111	200,000	50,000	0

"LOOKING AHEAD AND ACROSS"

Data illustration

Merged data

ID	YEAR	PAY8	PAY9	PAY10	PAY11	PAY12	B_WLTH	B_INCOME	B_EMPLOY
111	1996	600	600	600	600	600	200,000	50,000	0
111	1997	650	650	650	650	650	200,000	50,000	0
111	1998	700	700	700	700	700	200,000	50,000	0

"LOOKING AHEAD AND ACROSS"

Data illustration

"LOOKING AHEAD AND ACROSS" – SURVEY DATA (WIDE) MERGED WITH ADMINISTRATIVE DATA (LONG)

Using dates as an index variable

ANALYSIS_MO = 1 – 12:
Month variable for when a respondent reaches the end of the analysis

ANALYSIS_YR = 1998-2006
Year variable for when a respondent reaches the end of the analysis

```
IF ANALYSIS_MO NE 12 AND YEAR = ANALYSIS_YR THEN
      BENEPAID = PAY{ANALYSIS_MO + 1};
```

If the index month is January through November, take the benefit paid from the same index year but the month after.

```
IF ANALYSIS_MO = 12 AND YEAR = ANALYSIS_YR + 1 THEN
      BENEPAID = PAY{1};
```

If the index month is December, then look to the next year and pick the benefits paid from January.

SUMMARY

How to create an index

How to use an index

- A variable (e.g. B_INDX)
- A calculation (e.g. B_INDX + 1)
- A DO LOOP index (e.g., K = B_INDX to A_INDX)
- A hardcoded value (e.g., January = 1)
- Set to the values of an array subscript and a variable (e.g., 1997:2012)

Troubleshoot out of bounds

- Conditional logic for the array index
- Change the range for the array subscript

Pairing data step logic and arrays to “look ahead and across”

THANK YOU!

Lakhpreet “Preeti” Gill

pgill@mathematica-mpr.com