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.

<table>
<thead>
<tr>
<th>PERSON ID</th>
<th>AGE_1</th>
<th>REGION_1</th>
<th>AGE_2</th>
<th>REGION_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Study time period
- **Respondents enter the survey at different time points**: must identify correct wave for each respondent.
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; \(\leftarrow\) Range matches the number of waves in the survey
  IF NOT MISSING(B_INDX) THEN LEAVE; \(\leftarrow\) Once assigned, leave the loop.
  IF 58 \leq\) RELAGE{K} \leq 60 AND
     STATUS{K} = 1 THEN B_INDX = K;
END;

Do not want a later wave assigned if condition is still met.
CREATE A VARIABLE FOR THE ARRAY INDEX

<table>
<thead>
<tr>
<th>Year/wave</th>
<th>B_INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992/1</td>
<td>1</td>
</tr>
<tr>
<td>1994/2</td>
<td>2</td>
</tr>
<tr>
<td>1996/3</td>
<td>3</td>
</tr>
<tr>
<td>1998/4</td>
<td>4</td>
</tr>
<tr>
<td>2000/5</td>
<td>5</td>
</tr>
<tr>
<td>2002/6</td>
<td>6</td>
</tr>
<tr>
<td>2004/7</td>
<td>7</td>
</tr>
</tbody>
</table>
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:);
### BASICS TO CREATE THE ANALYTIC VARIABLES

#### Data illustration

**Raw variable**

<table>
<thead>
<tr>
<th>B_INDEX</th>
<th>R_RMSNAT_1</th>
<th>R_RMSNAT_2</th>
<th>R_RMSNAT_3</th>
<th>R_RMSNAT_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

**Analysis variable**

<table>
<thead>
<tr>
<th>B_INDEX</th>
<th>B_MSTATMAR</th>
<th>B_MSTATDIV</th>
<th>B_MSTATDwid</th>
<th>B_MSTATNEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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:

\[
\text{IF } B\_\text{INDX NE 11 THEN } B\_\text{STATNXT} = \text{STATUS}\{B\_\text{INDX} + 1\};
\]
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_INDEX NE 1 THEN DO;
   _ADLABCHK = ADLA{B_INDEX}; ← First process data for wave 2 and greater.
   B_ADLA = ADLA{B_INDEX} > 0;
   IF MISSING (ADLA{B_INDEX}) THEN B_ADLA = .;
END;

ELSE IF B_INDEX EQ 1 THEN DO; ← Then process data for wave 1.
   _ADLAB1CHK = ADLA{2};
   B_ADLA = ADLA{2} > 0;
   IF MISSING (ADLA{2}) THEN B_ADLA = .;
END;
<table>
<thead>
<tr>
<th>B_INDX</th>
<th>R_RADLA_2</th>
<th>R_RADLA_3</th>
<th>R_RADLA_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_INDX</th>
<th>_ADLABCHK</th>
<th>_ADLAB1CHK</th>
<th>B_ADLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>.</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Out of bounds Data illustration
INTEGRATING ARRAYS SUBSCRIPTS WITH INDEXES AND VARIABLES

```plaintext

DO I = 1997 TO 2012;

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

END;
```

**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.
INTEGRATING ARRAYS SUBSCRIPTS WITH INDEXES AND VARIABLES

### Example Table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1998</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1999</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Cross tab illustration
HANDLING ARRAYS OF DIFFERENT LENGTHS

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

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

```
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;
```
### Handling Arrays of Different Lengths

#### Data illustration

<table>
<thead>
<tr>
<th>B_INDEX</th>
<th>A_INDEX</th>
<th>R_HHITOT_3</th>
<th>R_HHITOT_4</th>
<th>R_HHITOT_5</th>
<th>R_HHITOT_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
<td>9,500</td>
<td>10,000</td>
<td>15,000</td>
<td>20,000</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>400,000</td>
<td>450,000</td>
<td>500,000</td>
<td>500,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_INDEX</th>
<th>A_INDEX</th>
<th>HHINC_SERIES1</th>
<th>HHINC_SERIES2</th>
<th>HHINC_SERIES3</th>
<th>HHINC_SERIES4-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
<td>10,000</td>
<td>15,000</td>
<td>20,000</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>400,000</td>
<td>450,000</td>
<td>500,000</td>
<td>.</td>
</tr>
</tbody>
</table>
### Administrative data

<table>
<thead>
<tr>
<th>ID</th>
<th>YEAR</th>
<th>PAY1</th>
<th>PAY2</th>
<th>PAY3</th>
<th>PAY4</th>
<th>PAY5</th>
<th>PAY6</th>
<th>PAY7</th>
<th>PAY8</th>
<th>PAY9</th>
<th>PAY10</th>
<th>PAY11</th>
<th>PAY12</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>1996</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>111</td>
<td>1997</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
</tr>
<tr>
<td>111</td>
<td>1998</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
</tr>
</tbody>
</table>

### Survey data

<table>
<thead>
<tr>
<th>ID</th>
<th>B_WLTH</th>
<th>B_INCOME</th>
<th>B_EMPLOY</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>200,000</td>
<td>50,000</td>
<td>0</td>
</tr>
</tbody>
</table>

“LOOKING AHEAD AND ACROSS” Data illustration
### Merged data

<table>
<thead>
<tr>
<th>ID</th>
<th>YEAR</th>
<th>PAY8</th>
<th>PAY9</th>
<th>PAY10</th>
<th>PAY11</th>
<th>PAY12</th>
<th>B_WLTH</th>
<th>B_INCOME</th>
<th>B_EMPLOY</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>1996</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>200,000</td>
<td>50,000</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>1997</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>650</td>
<td>200,000</td>
<td>50,000</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>1998</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>200,000</td>
<td>50,000</td>
<td>0</td>
</tr>
</tbody>
</table>
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**  \( \text{ANALYSIS_MO} \neq 12 \)  **AND**  \( \text{YEAR} = \text{ANALYSIS_YR} \) **THEN**

\[
\text{BENEPAID} = \text{PAY}\{\text{ANALYSIS_MO} + 1}\}
\]

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

**IF**  \( \text{ANALYSIS_MO} = 12 \)  **AND**  \( \text{YEAR} = \text{ANALYSIS_YR} + 1 \) **THEN**

\[
\text{BENEPAID} = \text{PAY}\{1\};
\]

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

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

How to create an index

How to use an index
- A variable (e.g. B_INDEX)
- A calculation (e.g. B_INDEX + 1)
- A DO LOOP index (e.g., K = B_INDEX to A_INDEX)
- 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