Unpacking Multiple Entries

Responses to multiple entry questions found in early years of the surveys of the four Original Cohorts were coded in a geometric progression format to conserve space on the tape. Variables such as 'Method of Seeking Employment,' 'Method of Finding Current or Last Job,' 'Type of Financial Aid Received,' 'Type of Child Care Arrangement,' and numerous health-related questions have been formatted in this way since the surveys began. Multiple entry items are identified by an asterisk under the source code box in the questionnaire and by a special detailed codeblock in the documentation. These responses need to be "unpacked" before they can be used in analysis. The example below pertains specifically to the Mature Women's cohort; it is applicable to all Original Cohort multiple entry variables, including the Young Women.

In later survey rounds, choose-all-that-apply items are coded as a series of yes-no variables in the data set. For example, although the respondent would see a list of possible fringe benefits and select all that were available to her, in the data this question is represented as a series of questions like "Did your employer make medical benefits available," "Did your employer make paid vacation available," and so on (see R20086.00-R20190.00 in 1995). Staff plan to return to earlier rounds and unpack the original variables in a similar manner, but these variables are not yet available to the public.

Example

Codes for the Mature Women's variable R03380., 'Fringe Benefits at Current Job 77,' range from 1 (the respondent reported only one such benefit, "medical insurance") to 259 (the respondent reported "medical insurance," "life insurance," and "paid sick leave") to 1023 (the respondent reported that she had access to all of the benefits listed). Although there are several different ways to sort out which respondent has positive answers on which components, this appendix provides one example in SAS and one example in SPSS.

Program 1: Unpacking Fringe Benefits Data in SAS

This SAS program unpacks fringe benefits from the variable "fringe." It creates 10 (dichotomous) dummy variables indicating the presence or absence of each of the 10 benefits. Each dummy is set to missing if fringe is missing (coded at -998 or -999). Note that the variables are created in reverse order from the codeblock, i.e., MEDICAL is code 1 on the tape and FRINGE10 in the program. The program statements listed below can be modified by the user to include the expanded set of fringe benefits available in later survey years as well as to unpack other multiple entry variables by extending the dummy, the counter, and the number of variables to agree with the total number of responses listed in the codeblock in the documentation.

```
data benefits;
infile 'D:\documents\requests\unpack.dat' lrecl=4;
input
 R0338000 4.;
 if R0338000 = -998 then R0338000 = .;
 if R0338000 = -999 then R0338000 = .;
label R0338000 = "FRINGE BNFTS CUR_JOB_77";
array fringe fringe01-fringe10;
do over fringe; if R0338000 ne . then fringe=0; end;
all=R0338000;

if all ge 512 then do; fringe10=1; all=all-512; end;
if all ge 256 then do; fringe09=1; all=all-256; end;
if all ge 128 then do; fringe08=1; all=all-128; end;
if all ge  64 then do; fringe07=1; all=all- 64; end;
if all ge  32 then do; fringe06=1; all=all- 32; end;
if all ge  16 then do; fringe05=1; all=all- 16; end;
if all ge   8 then do; fringe04=1; all=all-  8; end;
if all ge   4 then do; fringe03=1; all=all-  4; end;
if all ge   2 then do; fringe02=1; all=all-  2; end;
if all ge   1 then do; fringe01=1; all=all-  1; end;

label fringe01='medical,surgi';
label fringe02='life insuranc';
label fringe03='a retirement ';
label fringe04='training/educ';
label fringe05='profit sharin';
label fringe06='stock options';
label fringe07='freeâ€¦.meals';
label fringe08='freeâ€¦..mdse';
label fringe09='paid sick lea';
label fringe10='paid vacation';
run;
```

Program 2: Unpacking Fringe Benefits Data in SPSS

The SPSS program works in the same way as the SAS program. Users of this alternative package can follow this template.

```
/* UNPACKING 1981 YOUNG MEN FRINGE BENEFITS: SPSS/

compute FB1=0
variable labels FB1 '81 NONE'
compute FB2=0
variable labels FB2 '81 FLEX HRS'
compute FB3=0
variable labels FB3 '81 PAID VACATION'
compute FB4=0
variable labels FB4 '81 PD SICK'
compute FB5=0
variable labels FB5 '81 FR MERCH'
compute FB6=0
```

```
variable labels FB6 '81 FR MEALS'
compute FB7=0
variable labels FB7 '81 STOCK'
compute FB8=0
variable labels FB8 '81 PROFT'
compute FB9=0
variable labels FB9 '81 TRED'
compute FB10=0
variable labels FB10 '81 RETR'
compute FB11=0
variable labels FB11= '81 LIFE'
compute FB12=0
variable labels FB12 '81 HLTH'
compute FB81a=FB81
variable labels FB81a 'VARIABLE FOR NONE'

do if (2048 le FB81)
compute FB1=1
compute FB81=FB81-2048
else
compute FB1=-4
end if

do if (1024 le FB81)
compute FB2=1
compute FB81=FB81-1024
else
compute FB2=-4
end if

do if (512 le FB81)
compute FB3=1
compute FB81=FB81-512
else
compute FB=-4
end if

do if (256 le FB81)
compute FB4=1
compute FB81=FB81-256
else
compute FB4=-4
endif

do if (128 le FB81)
compute FB5=1
compute FB81=FB81-128
else
compute FB5=-4
end if

do if (64 le FB81)
compute FB6=1
compute FB81=FB81-64
else
compute FB6=-4
end if

do if (32 le FB81)
compute FB7=1
compute FB81=FB81-32
else compute FB7=-4
end if

do if (16 le FB81)
compute FB8=1
compute FB81=FB81-16
else
compute FB8=-4
end if

do if (8 le FB81)
compute FB9=1
compute FB81=FB81-8
else
compute FB9=-4
end if

do if (4 le FB81)
compute FB10=1
compute FB81=FB81-4
else
compute FB10=-4
end if

do if (2 le FB81)
compute FB11=1
compute FB81=FB81-2
else
compute FB11=-4
end if
```

```
do if (1 le FB81)
compute FB12=1
compute FB81=FB81-1
else
compute FB12=-4
end if
```