Appendix 38: Derivations for 1995 **KEY**
and Other Created Variables

CK-DATA ITEMS
These check items were intended primarily to help users keep track of the number of respondents who complete each section. Since respondents had to finish the first five sections of the survey to be defined as a completed interview (i.e., R34919 equals 201, 203, 204, or 205), the check items in sections one through five (FRONT through GAP) cannot take on a value of 2 ("no data in section")

Front section
No ck-data item was created for this section

HRC (Household Record Card) [R16063]
compute r1606300 = 1 /* default value, data looks OK
if sysmis(r1606400) r1606400 = -3
if sysmis(r1606500) r1606500 = -3
if sysmis(r1606600) r1606600 = -3
if sysmis(r1606700) r1606700 = -3

do repeat
    x = r1606400 r1606500 r1606600 r1606700
if x = -3 r1606300= 3
end repeat
if (r1611500 = 1) & nvalid(r1611700 to r1612000)=0 r1606300= 3

RWH (Respondent's work history) [R16649]
compute r1664900 = 1 /* default value, no problem
** Do for YW respondents only
if r0000100 = 4801 r1674700 = -3
if r0000100 = 4893 r1674700 = -3
if r0000100 = 0096 r1674700 = -3

if r1674700 = -3 r1664900 = 3

OJS (Respondent's On Jobs) [R16815]
compute r1681500 = 1 /* default value
do if (r1681800 = 1) & nvalid(r1682100 to r1682500) = 0
    compute r1681500 = 3
end if

RES (Respondent's Employer Sort) [R18015]
compute r1801500 = 1
compute val_res1 = nvalid(r1801800 to r1802600)
calculate val_r3em = nvalid(r1807700 to r1808500)
if (val_res1 = 0) & (val_r3em ge 1) r1801500 = 3

RSP (Respondent's Employer Supplement)
No check item created

GAP (Respondent's Gaps) [R25506]
compute num_gap = nvalid(r2550700, r2551000, r2551100, r2551200,
r2551300, r2551400, r2551500, r2551600, r2551700,
r2557300, r2557400)
TOPCODES FOR INCOME AND ASSETS

Certain income and asset items in this survey are topcoded at the Census Bureau to protect the confidentiality of respondent information. The topcoding procedure combines several of the highest values into a single category in order to prevent a respondent from being identified by an unusually high value.

A. Topcodes for Assets

In general, the topcode for each asset item is equal to the mean of its three highest values. Asset topcodes are found in the income section (e.g., INC-5A “AMOUNT OF U.S. SAVINGS BONDS”) and in the respondent’s and husband’s employer supplement (RSP, HSP) sections (e.g., RSP-110 “AMOUNT PRESENTLY IN ACCOUNT”)

B. Topcodes for Income

Income topcodes differ from asset topcodes in that there is a different topcode for each unit amount. For example, hourly pay items have a topcode of $48.07 and yearly pay items have a topcode of $100,001. Income topcodes are found in the income section (e.g., INC_22B “INCOME FROM WAGES/SALARY PAST 12 MONTHS”) and in the respondent’s and husband’s employer supplement (RSP, HSP) sections (e.g., RSP-811 “HOURLY RATE OF PAY”)

C. Topcodes for Other Pay Items

Other pay items such as expenditures may also be topcoded to protect the identity of respondents. These other items follow either the mean of the three highest values or the unit topcode methods described above.
RNI95-CUMULATIVE [R16012.]: CUMULATIVE REASON FOR NONINTERVIEW

This variable assigns a reason for non-interview for all respondents in the cohort, including those respondents who were out-of-scope in 1995. This variable is created using RNI information from 1995 and from previous surveys. For example, respondents who were coded as ‘deceased’ in a previous survey were out of scope in 1995, but they are assigned a value of 10 (deceased) in the 1995 cumulative RNI variable. This is also true of respondents previously identified as "congressional refusals" (RNI=14) and "dropped from survey" (RNI=12). These out-of-scope respondents are included in this cumulative 1995 RNI, unlike the cross-sectional RNI where they were assigned values of -5. The logic of the following SPSS code is that the cumulative RNI for 1995 is equal to the cross-sectional (XS) RNI for the cases that were in-scope for 1995 and equal to the previous RNI for cases that were out-of-scope.

The “Training Cases” problem: In 1995, thirty-one respondents were inadvertently interviewed using the incorrect version of the CAPI instrument. As a result, the information collected in these interviews was incomplete and irregular. These cases were declared to be non-interviews and assigned a value of 300 in the variable called OUTCOME [R34919]. However, in the RNI-95-CROSS-SECTION and ISR-95 variables these cases were assigned to the “other” category.

Mature Women:
DO IF (R3491900 < 0)    /* selects cases NOT in the 1995 sample
COMPUTE R1601200 = R1009310   /* copies cumul. RNI from 92MW into cumul. RNI 95
ELSE
COMPUTE R1601200 = R1601100
END IF

Young Women:
DO IF (R3491900 < 0)    /* selects cases NOT in the 1995 sample
COMPUTE R1601200 = R1365200   /* copies cumul. RNI from 93YW into cumul. RNI 95
ELSE
COMPUTE R1601200 = R1601100
END IF

VAR LAB R1601200 '1995 Cumulative RNI'
VAL LAB R1601200
1 'Unable to contact - Reason unspecified'
2 'Mover-good address, interview impossible'
3 'mover-good address, unable to interview'
4 'mover-no good address given'
5 'Nonmover-unable to interview after repeated attempts'
6 'Temporarily absent'
7 'Armed Forces'
8 'Institutionalized'
9 'Refused'
10 'Deceased (92 & 93 RNI + 95 info)'
11 'Other + training cases'
12 'No interview for 2 yrs, dropped from sample'
13 'Moved outside US, not in Armed Forces'
14 'Congressional refusal'
-4 '95 interview obtained'
SAMPLE-WEIGHT [R16014.]: 1995 SAMPLING WEIGHT

Prior to 1995, respondents who had no sampling weight due to their non-interview status were assigned to the universal missing code (-128) Starting with the 1995 survey the sampling weight of non-interview cases was assigned a value of zero The interview status of each respondent is determined by the cumulative reason for noninterview variable (RNI-CUMULATIVE, [R16012]). For additional information on the sampling weights for 1995 please consult the most recent versions of the NLS Mature Women User's Guide and the NLS Young Women User's Guide
IF ANY(X, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13) & (INHH = 1) REL = 1
IF R1611700 = 1 REL2 = 1 /* hrc-4b.02
IF R1611800 = 1 REL3 = 1 /* hrc-4b.03
IF R1611900 = 1 REL4 = 1 /* hrc-4b.04
IF R1612000 = 1 REL5 = 1 /* hrc-4b.05
DO IF COUNTER = R1664000 & R1663900 = 1 /* hrc-16 & hrc-15
COMPUTE REL = 1
END IF
END REPEAT
End if

COMPUTE NUM_FAM = SUM(REL2, REL3, REL4, REL5, REL6, REL7, REL8, REL9,
REL10, REL11, REL12, REL13, REL14, REL15, REL16,
REL17) + 1
VAR LAB NUM_FAM 'NUMFAM: No. of family members in HH'
VAL LAB NUM_FAM 1 'Respondent' 2 '2' 3 '3' 4 '4' 5 '5' 6 '6'

Rename vars (NUM_FAM = R1664600)
Due to a CAPI programming error the data for these rostered check items were not written out to the data file. However, these check items functioned properly during the course of the survey. The employer information obtained in these questions was used to finalize the employer roster and ultimately contributed to the creation of the CK-RSP-I-ARR variables found near the beginning of the RSP section.

array X R1899300 -R1900100,
array Y IND80R01 -IND80R09;
array beg R1811700-R1812500.

Do over Y; IF BEG=. THEN X=.;
IF X = 012 THEN Y =020; ELSE
IF X = 020 THEN Y =021; ELSE
IF X = 030 THEN Y =020; ELSE
IF X = 031 THEN Y =030; ELSE
IF X = 032 THEN Y =031; ELSE
IF X = 450 THEN Y =460; ELSE
IF X = 451 THEN Y =461; ELSE
IF X = 452 THEN Y =462; ELSE
IF X = 623 THEN Y =630; ELSE
IF X = 630 THEN Y =631; ELSE
IF X = 631 THEN Y =632; ELSE
IF X = 632 THEN Y =640; ELSE
IF X = 633 THEN Y =640; ELSE
IF X = 640 THEN Y =682; ELSE
IF X = 661 THEN Y =682; ELSE
IF X = 662 THEN Y =661; ELSE
IF X = 663 THEN Y =662; ELSE
IF X = 891 THEN Y =730; ELSE
IF X = 732 THEN Y =740; ELSE
IF X = 474 THEN Y =741; ELSE
IF X = 741 THEN Y =742; ELSE
IF X = 742 THEN Y =742; ELSE
IF X = 801 THEN Y =682; ELSE
IF X = 802 THEN Y =801; ELSE
IF X = 810 THEN Y =802; ELSE
IF X = 863 THEN Y =892; ELSE
IF X = 873 THEN Y =892; ELSE
IF X = 893 THEN Y =892; ELSE
IF X = 940 THEN Y =990; ELSE
IF X = 941 THEN Y =990; ELSE
IF X = 942 THEN Y =990; ELSE
IF X = 950 THEN Y =990; ELSE
IF X = 951 THEN Y =990; ELSE
IF X = 952 THEN Y =990; ELSE
IF X = 960 THEN Y =990; ELSE
IF X = 991 THEN Y =990; ELSE
IF X = 992 THEN Y =990;
ELSE Y =X;   END;

ARRAY XX R1902900 - R1903700;
array YY OCC80R01 - OCC80R09;
array bg R1811700-R1812500;

do over YY; if bg= then XX=.
IF XX = 016 THEN YY =017;ELSE
IF XX = 017 THEN YY =018;ELSE
IF XX = 018 THEN YY =016;ELSE
IF XX = 019 THEN YY =018;ELSE
IF XX = 021 THEN YY =019;ELSE
IF XX = 022 THEN YY =019;ELSE
IF XX = 353 THEN YY =349;ELSE
IF XX = 368 THEN YY =369;ELSE
IF XX = 436 THEN YY =437;ELSE
IF XX = 461 THEN YY =463;ELSE
IF XX = 462 THEN YY =464;ELSE
IF XX = 463 THEN YY =465;ELSE
IF XX = 464 THEN YY =466;ELSE
IF XX = 465 THEN YY =467;ELSE
IF XX = 466 THEN YY =468;ELSE
IF XX = 467 THEN YY =468;ELSE
IF XX = 628 THEN YY =633;ELSE
IF XX = 674 THEN YY =673;ELSE
IF XX = 795 THEN YY =794;ELSE

WBID [R25501.]: NUMBER OF WEEKS BETWEEN INTERVIEW DATES

This variable was constructed to count the number of weeks between the date of the 1995 interview and the date of the last interview. The SPSS code used to compute the weeks between interview dates is shown below. The logic of the code is to count the number of days between interview dates, divide by 7, and then round.

```
COMPUTE WBID = Rnd((CTIME.DAYS(DO1 - DOL)) / 7)
VAR LAB  WBID  '# weeks btwn intvs, rounded off'
```
PRINT FORMAT DOI (DATE9)
VAR LAB DOI 'Date of interview'

************ Impute DOLI & Start-Date **************
** NOTE: **
** Rs with missing Day in R1602600 will have a "15" imputed to **
** complete the date, but only if QJS-1 = "yes". This will **
** fill in missing DOLI and the START date when R agrees **
** that she worked for the Enname on the DOLI. The imputation **
** means that the true DOLI and the true Start-Date could **
** be as much as 2 weeks earlier or 2 weeks later. Thus, the **
** Weeks Between Interview Date (WBID) and therefore the **
** #weeks worked (WORK) could be off by as much as 2 weeks. **

COMPUTE DOLI_DD = R1602600
COMPUTE DOLI_MM = R1602700
COMPUTE DOLI_YY = R1602800
IF (R1602600 < 1) or (R1602600 > 31) DOLI_DD = 15
COMPUTE DOLI = DATE.DMY (DOLI_DD,DOLI_MM,DOLI_YY)
PRINT FORMAT DOLI (DATE9)

DO REPEAT
   QJS_1 = R1682100  R1682200  R1682300  R1682400  R1682500
   / START = START1 START2 START3 START4 START5
   / END = END1 END2 END3 END4 END5
   / CK_RSP_I = R1819100 to R1820700
   DO IF (CK_RSP_I ge 0) & (QJS_1 = 1) /* R had job activity & Start=Doli
   DO IF SYMISSTART & NYVAL(END)=1
   COMPUTE START = DOLI
   END IF
   END IF
PRINT FORMAT START(DATE9)
END REPEAT
EXECUTE
PRINT FORMAT DOLI (DATE9)

Compute DOLIMISS = Sysmis(DOLI)

*======= FINDS NUMBER OF Enames with activity since DOLI =======*
** A valid value means activity since DOLI. This includes **
** jobs that the R currently holds as well as ones where she **
** is no longer employed. The END-DATE must be greater than the **
** DOLI or must be equal to the DOI. **
** Rs with no activity since the DOLI have a "between-job" **
** gap equal to the #-WBID-BTWN-INTW-DATES (WBID). Thus, by **
** by definition WORK = 0 for these Rs. The WUMP and WOLF totals **
** for these Rs is determined in Part B of the GAPS section **
** in questions GAP-1B, GAP-2B, and INTER-CHK. **

**********************************************************************
COUNT NUM_JOB = R1819900 to R1820700 (1,2) /* # Enames w/activity since DOLI
Recode Num_job (0=0) (1 thru HI=1) into Activity /* R has job activity

*======= # R's with data in END-Gap section (GAP-1. ) =======*
** This universe includes Rs with a gap after the end date **
** of any Ename. A gap is computed for every Ename with **
** an END-DT prior to the DOI or prior to the Ename with **
** the next most recent Start-date. **

******************************************************************************
COMPUTE NUM_1 = NVALID(R2551L000 to R2551700)
RECODE NUM_1 (0=0) (1 thru HI=1) INTO HIT_1 /* Did R go thru GAP-1B?

*======= # R's with data in DOLI-Gap section (GAP-1B. ) =======*
** This universe includes Rs with no job activity since **
** the DOLI as well as Rs whose earliest START-DT is after **
** the DOLI **

******************************************************************************
DO IF DWORK ge 1  /* only if the # days working could be computed
LOOP I# = ST DAY to (ST.DAY + (DWORK - 1))
COMPUTE DAYS(I#) = 1
END LOOP
END IF
END REPEAT
Execute

COUNT WORKD = DAY1 to DAYS675 (1) /* #days working, from DAYS array
COMPUTE WORK.IND = RN( WORKD / 7 ) /* Rounded: #weeks worked, from DAYS array
COMPUTE NOWORKD = DBID - WORKD
COMPUTE NOWORKD = RN( NOWORKD / 7 ) /* Rounded: #weeks not working
COMPUTE ALLWRK1 = 0  /* default value
IF ABS(WORKD - DBID) < 3.5  ALLWRK1 = 1
VAR LAB ALLWRK1 'DAYS array shows R worked all weeks in WBID'

Calculate Weeks Between Interview Dates WBID.W

COMPUTE WBID = ((ctime.DAYS(DO1 - DOLI)) / 7) /* #weeks between interviews, not rounded
COMPUTE WBID.IND = RN( (ctime.DAYS(DO1 - DOLI)) / 7 ) /* WBID, rounded off
COMPUTE DOLI.W = ((ctime.DAYS(DO1 - DOLI)) / 7) + 1
DO IF NVAL(DOLI) ge 1  /* if there's a valid date for the DOLI
COMPUTE DOLI.W = 1  /* the DOLI is always week 1 for every R
END IF

IF SYSMIS(DOLI)  DOLI.W = -3
IF SYSMIS(DOLI)  WBID.IND = -3
MISS VAL DOLI.W WBID.IND (-3)

Build Universe

Initial value of universe, should be recoded 1-5
in subsequent commands

COMPUTE UNIVERSE = -9

*** Identify Rs with missing DOLI UNIVERSE = 0
IF DOLIMISS = 1  UNIVERSE = 0

*** Identify Rs with valid DOLI, no work, and HIT_1B UNIVERSE = 2
IF HIT_1B = 1 and DOLIMISS = 0  UNIVERSE = 2

*** Identify Rs hitting GAP-1 only, + valid DOLI UNIVERSE = 3
IF (HIT_1 = 1) & (HIT_1B = 0) & (DOLIMISS = 0)  UNIVERSE = 3

*** Rs with activity, hitting GAP-1 only, + valid DOLI UNIVERSE = 4
DO IF (ACTIVITY = 1) & (HIT_1 = 0) & (HIT_1B = 1) & (DOLIMISS = 0)
COMPUTE UNIVERSE = 4
END IF

*** Rs with activity, hitting both gaps, + valid DOLI UNIVERSE = 5
IF (HIT_1 = 1) & (HIT_1B=1) & (DOLIMISS = 0)  UNIVERSE = 5

*** Identify Rs with valid DOLI, WORKED ALL weeks UNIVERSE = 1
IF (HIT_GAP = 0) & (DOLIMISS = 0)  UNIVERSE = 1

VAL LAB UNIVERSE
The source of the problem is that the CASES program obtained a negative value when subtracting the End-date from the Start date of the following Employer. For example, for Serial=2106, the Stop-Date for Enamel is 15-AUG-94 and the Start-Date is 01-AUG-94. The CASES program calculated a gap of -2 weeks between these two jobs, when in reality there should have been no gap since the jobs overlapped. Thus, for these 3 cases INTER-CK is recoded to 0.

It is unclear how CASES obtained a value of -61 for this MW case.

If any(SERIAL, 2106, 0232, 4010) R2553400 = 0

CHRR's END-GAP Compute total length of gap in GAP-1

This section calculates the length of the gap following the End-Dates of every job listed (called the END-GAP).

This gap-length is created from the Start and Stop dates and the DOI date. This gap-length is expressed as the number of weeks, and it should match the GAP1_Nn length reconstructed from data in questions GAP-2 and INTER_CK. (Recall that END-GAP length = GAP_2n + INTER_CK.

You'll notice that the number of END-GAPs created by CHRR does not match the number found in the raw data. This is because the CASES program sometimes missed End-gaps that really existed and "found" END-GAPs that didn't exist. Consequently, there are cases where CHRR calculated the length of an End-Gap (CHRRgap) that has no corresponding data in GAP-1, GAP-2, or INTER-CK. This is because these questions weren't asked if no gap existed. Thus, the number of weeks looking for work or on layoff (GAP-2) is not available, so that neither the # weeks unemployed (WUMP) nor out of the labor force (OLF) can be computed. But the #weeks working (WORK) CAN be calculated by looking at the Start and Stop dates. Therefore, there will be respondents who have a value in WORK but have missing data in WUMP and WOLF.

HERE'S ANOTHER WAY TO COMPUTE THE LENGTH OF THE GAP, MEASURED IN # WEEKS.

COMPUTE GAPLength = RND((CTIME.DAYS(DOI - END1)) / 7)
*-
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = DOI.W
COMPUTE GAP3_WK = RND((GAP_END3 - GAP_ST3) + 1
END IF

***** 2 Enames. Columns 1 & 4 only
DO IF NVALID(R18199000, R18202000) = 2 & NVAL(R18200000, R18201000, R18203000 to R18207000) = 0
COMPUTE EMPLIST = 8
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST4.W
COMPUTE GAP1_WK = RND((GAP_END1 - GAP_ST1) + 1
*-
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = DOI.W
COMPUTE GAP4_WK = RND((GAP_END4 - GAP_ST4) + 1
END IF

***** 2 Enames: Columns 1 & 5 only
DO IF NVALID(R18199000, R18203000) = 2 & NVAL(R18200000 to R18202000, R18204000 to R18207000) = 0
COMPUTE EMPLIST = 9
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST5.W
COMPUTE GAP1_WK = RND((GAP_END1 - GAP_ST1) + 1
*-
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = DOI.W
COMPUTE GAP5_WK = RND((GAP_END5 - GAP_ST5) + 1
END IF

***** 2 Enames: Columns 1 & 6 only
DO IF NVALID(R18199000, R18204000) = 2 & NVAL(R18200000 to R18203000, R18205000 to R18207000) = 0
COMPUTE EMPLIST = 10
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST6.W
COMPUTE GAP1_WK = RND((GAP_END1 - GAP_ST1) + 1
*-
COMPUTE GAP_ST6 = END6.W + 1
COMPUTE GAP_END6 = DOI.W
COMPUTE GAP6_WK = RND((GAP_END6 - GAP_ST6) + 1
END IF

***** 2 Enames: Columns 1 & 7 only
DO IF NVALID(R18199000, R18205000) = 2 & NVAL(R18200000 to R18204000, R18206000, R18207000) = 0
COMPUTE EMPLIST = 11
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST7.W
COMPUTE GAP1_WK = RND((GAP_END1 - GAP_ST1) + 1
*-
COMPUTE GAP_ST7 = END7.W + 1
COMPUTE GAP_END7 = DOI.W
COMPUTE GAP7_WK = RND((GAP_END7 - GAP_ST7) + 1
END IF

**** 2 Enames in columns 2 & 3 only
DO IF NVALID(R18200000, R18201000) = 2 & NVALID(R18199000, R18202000 to R18207000) = 0
COMPUTE EMPLIST = 12
COMPUTE GAP_ST2 = END2.W + 1
COMPUTE GAP_END2 = ST3.W
COMPUTE GAP2_WK = RND((GAP_END2 - GAP_ST2) + 1
*-
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = DOI.W
COMPUTE GAP3_WK = RND((GAP_END3 - GAP_ST3) + 1
END IF

**** 2 Enames in columns 2 & 4 only
**COMPUTE GAP_ST1 = END1.W + 1**
**COMPUTE GAP_END1 = ST2.W**
**COMPUTE GAP1.WK = RND(GAP_END1 - GAP_ST1) + 1**
**-.**
**COMPUTE GAP_ST2 = END2.W + 1**
**COMPUTE GAP_END2 = ST4.W**
**COMPUTE GAP2.WK = RND(GAP_END2 - GAP_ST2) + 1**
**-.**
**COMPUTE GAP_ST4 = END4.W + 1**
**COMPUTE GAP_END4 = DOI.W**
**COMPUTE GAP4.WK = RND(GAP_END4 - GAP_ST4) + 1**
END IF

***** 3 Enames in column 1, 3, 4 only
**DO IF NVAL(R1819900, R1820100, R1820200) = 3 & NVAL(R1820000, R1820300 to R1820700) = 0**
**COMPUTE EMPLIST = 19**
**COMPUTE GAP_ST1 = END1.W + 1**
**COMPUTE GAP_END1 = ST3.W**
**COMPUTE GAP1.WK = RND(GAP_END1 - GAP_ST1) + 1**
**-.**
**COMPUTE GAP_ST3 = END3.W + 1**
**COMPUTE GAP_END3 = ST4.W**
**COMPUTE GAP3.WK = RND(GAP_END3 - GAP_ST3) + 1**
**-.**
**COMPUTE GAP_ST4 = END4.W + 1**
**COMPUTE GAP_END4 = DOI.W**
**COMPUTE GAP4.WK = RND(GAP_END4 - GAP_ST4) + 1**
END IF

***** 3 Enames in column 2, 3, 4 only
**DO IF NVAL(R1820000, R1820100, R1820200) = 3 & NVAL(R1819900, R1820300 to R1820700) = 0**
**COMPUTE EMPLIST = 20**
**COMPUTE GAP_ST2 = END2.W + 1**
**COMPUTE GAP_END2 = ST3.W**
**COMPUTE GAP2.WK = RND(GAP_END2 - GAP_ST2) + 1**
**-.**
**COMPUTE GAP_ST3 = END3.W + 1**
**COMPUTE GAP_END3 = ST4.W**
**COMPUTE GAP3.WK = RND(GAP_END3 - GAP_ST3) + 1**
**-.**
**COMPUTE GAP_ST4 = END4.W + 1**
**COMPUTE GAP_END4 = DOI.W**
**COMPUTE GAP4.WK = RND(GAP_END4 - GAP_ST4) + 1**
END IF

***** 3 Enames in column 3, 4, 5 only
**DO IF NVAL(R1820100, R1820200, R1820300) = 3 & NVAL(R1819900, R1820000, R1820400 to R1820700) = 0**
**COMPUTE EMPLIST = 21**
**COMPUTE GAP_ST3 = END3.W + 1**
**COMPUTE GAP_END3 = ST4.W**
**COMPUTE GAP3.WK = RND(GAP_END3 - GAP_ST3) + 1**
**-.**
**COMPUTE GAP_ST4 = END4.W + 1**
**COMPUTE GAP_END4 = ST5.W**
**COMPUTE GAP4.WK = RND(GAP_END4 - GAP_ST4) + 1**
**-.**
**COMPUTE GAP_ST5 = END5.W + 1**
**COMPUTE GAP_END5 = DOI.W**
**COMPUTE GAP5.WK = RND(GAP_END5 - GAP_ST5) + 1**
END IF

***** 3 Enames in column 1, 3, 5 only
**DO IF NVAL(R1819900, R1820100, R1820300) = 3 & NVAL(R1820000, R1820200, R1820400 to R1820700) = 0**
**COMPUTE EMPLIST = 22**
**COMPUTE GAP_ST1 = END1.W + 1**
**COMPUTE GAP_END1 = ST3.W**
**COMPUTE GAP1.WK = RND(GAP_END1 - GAP_ST1) + 1**
**-.**
**COMPUTE GAP_ST3 = END3.W + 1**
COMPUTE GAP_ST7 = END7_W + 1
COMPUTE GAP_END7 = DO1.W
COMPUTE GAP7WK = RND(GAP_END7 - GAP_ST7) + 1
END IF

***** 4 Enames in column 1, 2, 3, 4 only
DO IF NVALID(R1819900 to R18202000) = 4 & NVALID(R1820300 to R1820700) = 0
COMPUTE EMPLIST = 27
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST2.W
COMPUTE GAP1WK = RND(GAP_END1 - GAP_ST1) + 1
**-
COMPUTE GAP_ST2 = END2.W + 1
COMPUTE GAP_END2 = ST3.W
COMPUTE GAP2WK = RND(GAP_END2 - GAP_ST2) + 1
**-
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = ST4.W
COMPUTE GAP3WK = RND(GAP_END3 - GAP_ST3) + 1
**-
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = DO1.W
COMPUTE GAP4WK = RND(GAP_END4 - GAP_ST4) + 1
END IF

***** 4 Enames in column 1, 2, 4, 5 only
DO IF NVALID(R1819900, R1820000, R1820200, R1820300) = 4 & NVALID(R1820100, R1820400 to R1820700) = 0
COMPUTE EMPLIST = 28
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST2.W
COMPUTE GAP1WK = RND(GAP_END1 - GAP_ST1) + 1
**-
COMPUTE GAP_ST2 = END2.W + 1
COMPUTE GAP_END2 = ST4.W
COMPUTE GAP2WK = RND(GAP_END2 - GAP_ST2) + 1
**-
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = ST5.W
COMPUTE GAP4WK = RND(GAP_END4 - GAP_ST4) + 1
**-
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = DO1.W
COMPUTE GAP5WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

***** 4 Enames in column 1, 3, 4, 5 only
DO IF NVALID(R1819900, R1820100 to R1820300) = 4 & NVALID(R1820000, R1820400 to R1820700) = 0
COMPUTE EMPLIST = 29
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST3.W
COMPUTE GAP1WK = RND(GAP_END1 - GAP_ST1) + 1
**-
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = ST4.W
COMPUTE GAP3WK = RND(GAP_END3 - GAP_ST3) + 1
**-
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = ST5.W
COMPUTE GAP4WK = RND(GAP_END4 - GAP_ST4) + 1
**-
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = DO1.W
COMPUTE GAP5WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

***** 4 Enames in column 2, 3, 4, 5 only
DO IF NVALID(R1820000 to R1820300) = 4 & NVALID(R1819900, R1820400 to R1820700) = 0
- COMPUTE GAP_ST6 = END6.W + 1
- COMPUTE GAP_END6 = ST7.W
- COMPUTE GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
- COMPUTE GAP_ST7 = END7.W + 1
- COMPUTE GAP_END7 = DO1.W
- COMPUTE GAP7_WK = RND(GAP_END7 - GAP_ST7) + 1
END IF

***** 4 Enames in column 1, 2, 8, 9 only
DO IF NVALID(R1819900, R1820000, R1820600, R1820700) = 4
& NVALID(R1820100 to R1820500) = 0
- COMPUTE EMPLIST = 34
- COMPUTE GAP_ST1 = END1.W + 1
- COMPUTE GAP_END1 = ST2.W
- COMPUTE GAP1_WK = RND(GAP_END1 - GAP_ST1) + 1
- COMPUTE GAP_ST2 = END2.W + 1
- COMPUTE GAP_END2 = ST8.W
- COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
- COMPUTE GAP_ST8 = END8.W + 1
- COMPUTE GAP_END8 = ST9.W
- COMPUTE GAP8_WK = RND(GAP_END8 - GAP_ST8) + 1
- COMPUTE GAP_ST9 = END9.W + 1
- COMPUTE GAP_END9 = DO1.W
- COMPUTE GAP9_WK = RND(GAP_END9 - GAP_ST9) + 1
END IF

***** 5 Enames in column 1, 2, 3, 4, & 5 only
DO IF NVALID(R1819900 to R1820300)= 5 & NVALID(R1820400 to R1820700) = 0
- COMPUTE EMPLIST = 35
- COMPUTE GAP_ST1 = END1.W + 1
- COMPUTE GAP_END1 = ST2.W
- COMPUTE GAP1_WK = RND(GAP_END1 - GAP_ST1) + 1
- COMPUTE GAP_ST2 = END2.W + 1
- COMPUTE GAP_END2 = ST3.W
- COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
- COMPUTE GAP_ST3 = END3.W + 1
- COMPUTE GAP_END3 = ST4.W
- COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
- COMPUTE GAP_ST4 = END4.W + 1
- COMPUTE GAP_END4 = ST5.W
- COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 1
- COMPUTE GAP_ST5 = END5.W + 1
- COMPUTE GAP_END5 = DO1.W
- COMPUTE GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

***** 5 Enames in column 1, 2, 3, 7, 8 only
DO IF NVALID(R1819900, R1820000, R1820100, R1820500, R1820600)= 5
& NVALID(R1820200 to R1820400, R1820700) = 0
- COMPUTE EMPLIST = 36
- COMPUTE GAP_ST1 = END1.W + 1
- COMPUTE GAP_END1 = ST2.W
- COMPUTE GAP1_WK = RND(GAP_END1 - GAP_ST1) + 1
- COMPUTE GAP_ST2 = END2.W + 1
- COMPUTE GAP_END2 = ST3.W
- COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
- COMPUTE GAP_ST3 = END3.W + 1
- COMPUTE GAP_END3 = ST7.W
COMPUTE GAP_END2 = ST1.W
COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
*-
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = ST4.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
*-
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = ST5.W
COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 1
*-
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = ST6.W
COMPUTE GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
*-
COMPUTE GAP_ST6 = END6.W + 1
COMPUTE GAP_END6 = DO1.W
COMPUTE GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
END IF

***** 7 Enames in columns 1, 2, 3, 4, 5, 6, & 7 only
DO IF NVAL(R1819900 to R1820500) = 7 & NVAL(R1820600, R1820700) = 0
COMPUTE EMPLIST = 40
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = ST2.W
COMPUTE GAP1_WK = RND(GAP_END1 - GAP_ST1) + 1
*-
COMPUTE GAP_ST2 = END2.W + 1
COMPUTE GAP_END2 = ST3.W
COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
*-
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = ST4.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
*-
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = ST5.W
COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 1
*-
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = ST6.W
COMPUTE GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
*-
COMPUTE GAP_ST6 = END6.W + 1
COMPUTE GAP_END6 = ST7.W
COMPUTE GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
*-
COMPUTE GAP_ST7 = END7.W + 1
COMPUTE GAP_END7 = DO1.W
COMPUTE GAP7_WK = RND(GAP_END7 - GAP_ST7) + 1
END IF
END IF

VAL LAB EMPLIST
-9 'Not yet assigned' 1 'Emp 1 only' 2 'Emp 2 only'
 3 'Emp 3 only' 4 'Emp 4 alone' 5 'Emp 5 alone'
 6 'Emp 1,2 only' 7 'Emp 1,3 only' 8 'Emp 1,4 only'
 9 'Emp 1,5 only' 10 'Emp 1,6 only' 11 'Emp 1,7 only'
12 'Emp 2,3 only' 13 'Emp 2,4 only' 14 'Emp 3,4 only'
15 'Emp 4,5 only' 16 'Emp 5,6 only' 17 'Emp 1,2,3 only'
18 'Emp 1,2,4 only' 19 'Emp 1,3,4 only' 20 'Emp 2,3,4 only'
21 'Emp 3,4,5 only' 22 'Emp 1,3,5 only' 23 'Emp 1,4,5 only'
24 'Emp 1,5,6 only' 25 'Emp 4,5,6 only' 26 'Emp 5,6,7 only'
27 'Emp 1,2,3,4' 28 'Emp 1,3,4,5 only' 29 'Emp 2,3,4,5'

************** HAND-EDITS FOR INDIVIDUAL CASES **************
MW respondent
DO IF (SERIAL = 2196)
DO REPEAT
  GAP_1 = R2551000 R2551100 R2551200 R2551300 R2551400 R2551500 R2551700
  GAP_2 = R2552200 R2552300 R2552400 R2552500 R2552600 R2552700 R2552900
  INTER = INTER01 INTER02 INTER03 INTER04 INTER05 INTER06 INTER08
  GAP1_N = GAP1_N1 GAP1_N2 GAP1_N3 GAP1_N4 GAP1_N5 GAP1_N6 GAP1_N8
  CHRRGAP = GAP1_WK GAP2_WK GAP3_WK GAP4_WK GAP5_WK GAP6_WK GAP8_WK
  UMPA = UMPA.1 UMPA.2 UMPA.3 UMPA.4 UMPA.5 UMPA.6 UMPA.8
  OLFA = OLFA.1 OLFA.2 OLFA.3 OLFA.4 OLFA.5 OLFA.6 OLFA.8
  if GAP_1 is "DK" (-2) then End-Gap = CHRRGAP and OLF, UMP = -3
  DO IF (GAP_1 < 0)
      COMPUTE GAP1_N = CHRRGAP
      COMPUTE OLFA = -3
      COMPUTE UMPA = -3
  END IF

  **** GAP_1 = 1 R looked for work or laid off for ALL weeks during gap
  ** If GAP_1 = 1 ("all") the CASES program skipped past GAP-2 and
  ** INTER-CK and so the total length of the gap could not be computed
  ** using the formula TOTAL END-GAP = GAP-2 + INTER-CK. Thus,
  ** the gap length calculated by CHRR using the Start- and Stop dates
  ** (GAPn_WK) was used instead.
  DO IF GAP_1 = 1
      COMPUTE GAP1_N = CHRRGAP  /** computed in previous section by CHRR
      COMPUTE UMPA = CHRRGAP  /** since R was looking she's considered UNEMP
      COMPUTE OLFA = 0  /* if looked "all" wks then wks NOT looking must be 0
  END IF

  ******** GAP_1 = 3 #wks looking or laid-off was "none"
  ** If R spent no (0) weeks looking for work, or on layoff, then
  ** the entire gap is classified as OLF (where INTER-CK = INTER = OLFA)
  DO IF GAP_1 = 3
      COMPUTE UMPA = 0  /* if looked for 0 wks in GAP_1 then GAP-2=0
      IF INTER ge 0  GAP1_N = INTER  /* Gap length is amount in INTER-CK
      IF INTER < 0  GAP1_N = CHRRGAP  /** INTER-CK = -2, so OLFA = CHRRGAP
      IF SYMSIS(INTER) GAP1_N = CHRRGAP  /* if INTER-CK is missing, gap_1 = CHRRGAP
      COMPUTE OLFA = GAP1_N  /* entire gap-length is OLF

      If sysmis(CHRRgap) & (gapn_1 > 0) CHRRgap = gap1_n
  END IF

  ******** GAP_1 = 2 looked for "some" weeks
  ** When GAP-2 is -2 (DK) it is impossible to determine the
  ** value of INTER-CK since INTER-CK = TOTAL - GAP-2 .
  ** Yet the CASES program computed INTER-CK anyway and
  ** obtained negative values. But it is still possible to
  ** compute the TOTAL gap length using the values in
  ** INTER-CK and GAP-2. For example, if GAP-2 = -2 and
  ** INTER-CK is 57, the actual length of the End-gap is 55 weeks.
  DO IF (GAP_1 = 2)  /** some weeks looking
  IF (GAP_2 < 0)  GAP1_N = CHRRGAP  /* gap length as computed by CHRR
  IF (GAP_2 < 0)  UMPA = -3  /* R doesn't know #wks looking, so UMPA = miss
  IF (GAP_2 < 0)  OLFA = -3  /* R doesn't know #wks looking, so OLFA=miss

  IF (GAP_2 ge 0) & (INTER ge 0)  GAP1_N = GAP_2 + INTER
  IF (GAP_2 ge 0) & (SYMSIS(INTER))  GAP1_N = CHRRGAP
IF DOLIGAP0 ge 1  DOLI_CHR = 0  /* length of at least 1 gap was 0 weeks

DO IF (DOLIGAP0 = 0)
IF MIN(DOLI.W1, DOLI.W2, DOLI.W3, DOLI.W4, DOLI.W5, DOLI.W6, DOLI.W7, DOLI.W8,
    DOLI.W9, DOLIW10) = -3  DOLI_CHR = -3

IF MIN(DOLI.W1, DOLI.W2, DOLI.W3, DOLI.W4, DOLI.W5, DOLI.W6, DOLI.W7,
    DOLI.W8, DOLI.W9, DOLIW10) > -1
    DOLI_CHR = MIN(DOLI.W1, DOLI.W2, DOLI.W3, DOLI.W4, DOLI.W5, DOLI.W6, DOLI.W7,
    DOLI.W8, DOLI.W9, DOLIW10)
END IF

IF SYSMIS(DOLI)  DOLI_CHR = -3
VAR LAB DOLI_CHR  '# wks btwn DOLI --> Start, as calculated by CHRR'

======== DOLI_CEN  =========================================================
** This is the length of the DOLI->Start gap as calculated from
** the data in the Census tape
IF (R2557300 = 2)  DOLI_CEN = R2557400 + R2557500
IF (R2557300 = 3)  DOLI_CEN = R2557500

============================================
****** Create UMPB OLFB (from INTERCHK or DOLI-GAP)  ******
*** These items measure the #weeks looking for work or laid-off ***
*** (UMPB) and #weeks not looking for work (OLFB) during the ***
*** gap from the DOLI to the earliest Start-date of any job. ***

============================================
****  ***  R doesn't know if she looked for work during DOLI-GAP (GAP=1B = -2)
IF R2557300 < 0  UMPB = -3
IF R2557300 < 0  OLFB = -3

****  ***  R looked for work for "All" weeks during the DOLI-GAP (GAP=1B = 1)
** therefore there were no weeks out-of-labor-force, all weeks in gap unemployd
IF R2557300 = 1  UMPB = DOLI_CHR
IF R2557300 = 1  OLFB = 0

****  ***  R looked for work or was laid off for "some" weeks (GAP=1B = 2)
IF R2557300 = 2 & R2557400 < 0  UMPB = -3
IF R2557300 = 2 & R2557400 < 0  OLFB = -3

IF R2557300 = 2 & R2557400 ge 0  UMPB = R2557400
IF R2557300 = 2 & R2557400 ge 0  OLFB = DOLI_CHR - R2557400

****  ***  R looked for work or was laid off for "None" (0) weeks (GAP=1B = 3)
IF R2557300 = 3  UMPB = 0
IF R2557300 = 3  OLFB = DOLI_CHR

RECODE UMPB OLFB (L0 thru -1 = -3)  /* convert other negatives into missing

***********  ****************************  ******************
**** HAND EDIT: START = END & no other work
*****************************************************
** At least 4 cases had no interval between the
** only START-STOP dates, meaning GAP =1B was the
** only gap CHRR's code (above) assigns the DOLI-GAP
** a value of 0 (since technically START - DOLI = 0)
** whereas the actually length of the gap is the WBID.
** Rather than adding additional loops to program
** these strange pairs of dates the cases will be
** hand edited

MW
DO IF ANY(SERIAL, 0140, 0395, 1873)
VAR LAB ALLWRK2 'WEEKS array shows R worked all weeks in WBID'

*======== Filling in WUMP, "looking, on layoff" = 2
VECTOR WEEKS = WEEK1 to WEEK812
DO IF (DOLIMISS = 0)
DO REPEAT
  GAP_ST = GAP_ST1 GAP_ST2 GAP_ST3 GAP_ST4 GAP_ST5 GAP_ST6
  GAP_ST8
  / UMPA = UMPA.1 UMPA.2 UMPA.3 UMPA.4 UMPA.5 UMPA.6
  UMPA 8
  / OLFA = OLFA.1 OLFA 2 OLFA.3 OLFA.4 OLFA.5 OLFA.6
  OLFA 8

  COMPUTE STARTGAP = GAP_ST - 1

  DO IF (UMPA GE 1) /* selects Rs with valid values of UMP
  LOOP I# = RND(STARTGAP) to RND(STARTGAP + UMPA) IF WEEKS(I#) ne 1
  COMPUTE WEEKS(I#) = 2 /**** WUMP
  END LOOP
  END IF
  END REPEAT
  END IF
  EXECUTE

****** PLUGGING IN OLF, "Not looking/ not on layoff" = 3
VECTOR WEEKS = WEEK1 to WEEK812
DO IF (DOLIMISS = 0)
DO REPEAT
  GAP_ST = GAP_ST1 GAP_ST2 GAP_ST3 GAP_ST4 GAP_ST5 GAP_ST6
  GAP_ST8
  / UMPA = UMPA.1 UMPA.2 UMPA.3 UMPA.4 UMPA.5 UMPA.6
  UMPA 8
  / OLFA = OLFA.1 OLFA 2 OLFA.3 OLFA.4 OLFA.5 OLFA.6
  OLFA 8

  COMPUTE STARTGAP = GAP_ST - 1

  DO IF (OLFA GE 1) /*** selects if valid values (i.e., >= 0)
  LOOP I# = RND(STARTGAP + UMPA) to RND(STARTGAP + UMPA) + (OLFA - 1)
  COMPUTE WEEKS(I#) = 3 /**** WOLF
  END LOOP
  END IF
  END REPEAT
  END IF
  EXECUTE

==================================================================
* Create the Preliminary Weeks-working variables by counting values
* in the WEEKS ARRAY. These total do not yet include the DOLI_Gap
* measured in GAP-1B, GAP-2B, and INTER-CHK
==================================================================

DO IF (DOLIMISS = 0)
  COUNT WORKA = WEEK1 to WEEK812 (1) /* if WEEKN = 1, R worked that week
  IF (MISSWRK ge 1) & (ALLWRK1 = 0) WORKA = -3

  COUNT UMPA = WEEK1 to WEEK812 (2) /* if WEEKN = 2, R was looking

  COUNT OLFA = WEEK1 to WEEK812 (3) /* out of labor force

  IF WORKA ge 0 SUM_WEEK = WORKA + UMPA + OLFA
  END IF

VAR LAB SUM_WEEK 'Sum of 3 weeks variables, END-Gap only'
VAR LAB WORKA '# WEEKS WORKING BTWN DOLI & DOI'
  / UMPA '## WEEKS not working but looking for work'
  / OLFA '## weeks not working & NOT looking (OLF)'

********** UNIVERSE = 4  Some job activity, but only a DOLI-Gap

DO IF UNIVERSE = 4
  COUNT  WORK = WEEK1 to WEEK812 (1)
  COMPUTE WUMP = UMPB
  COMPUTE WOLF = OLFB
  IF WORKA = -3  WORK = -3
  IF (UMPB < 0)  WUMP = -3
  IF (OLFB < 0)  WOLF = -3
  IF R2557300 = 3  WUMP = 0
END IF

********** UNIVERSE = 5  BOTH gaps
   Do If UNIVERSE = 5
   COUNT  WORK.A = WEEK1 to WEEK812(1)
   COUNT  WUMP.A = WEEK1 to WEEK812 (2)
   COUNT  WOLF.A = WEEK1 to WEEK812 (3)
   COMPUTE WORK = WORK.A
   COMPUTE WUMP = WUMP.A + UMPB
   COMPUTE WOLF = WOLF.A + OLFB
   IF WORKA = -3  WORK = -3
   IF (UMPB < 0)  WUMP = -3
   IF (OLFB < 0)  WOLF = -3
End If

*----------------------------------------

Patches

***** These are basically patches to fix cases that had more
***** complex patterns of missing data and so forth. A case-by-case
***** review revealed errors (sometimes minor ones) in the weeks
***** variables.
***** Patch values for some cases when a work spell is missing and R was not
***** working the whole time. The OLF can be assigned a 0 when R said
***** she was looking the whole time, even the exact number of
***** weeks looking is unknown. Conversely, if R didn't look at all
***** then WUMP is 0 and WOLF is unknown. Finally, if the work dates aren't
***** complete and the remaining dates did not account for the whole WBID
***** then the total number of weeks working can't be computed.

DO IF (MISSWRK = 1)  &  (ALLWRK1 = 0)
  COMPUTE WORK = -3
  Compute WORK RND = -3  /* the count isn't complete b/c of missing date
  COMPUTE WUMP = -3
  COMPUTE WOLF = -3
  IF (UMPA > -1)  &  (UMPB > -1)  WUMP = UMPA + UMPB
  IF (OLFA > -1)  &  (OLFB > -1)  WOLF = OLFA + OLFB
END IF

***** Blank (set to -3) WUMP and WOLF because the sum of
***** WORK, WUMP, WOLF is more than 2 away from WBID and
***** the # weeks worked is accurate. It may be possible to
***** determine the WUMP and WOLF on a case-by-case basis
***** given time. The number of cases affected by this patch
***** is about 60

DO IF  ABS((WORK + WUMP + WOLF) - (WBID.RND)) > 2
     &  ABS(WORK - WORK RND le 2)
  Compute WUMP = -3
  COMPUTE WOLF = -3
END IF
NET FAMILY ASSETS AND FAMILY INCOME

Although rights to estates/trusts data are available from R15057 and R15058 in the 1993 Young Women survey, it is not included in 1993 total net family assets. In the 1995 survey, rights to estates/trusts are included (see trus95), using the format provided by the 1990 Older Men's Appendix 28.

When the respondent reports 'yes' to investment in business, prior derivation assigns a value of 0 to "investment in business" when either the respondent or her husband or partner reports no business income. The code has been changed to assign a value of 0 only when both respondent and her husband/partner report no income, if either reports don't know/refuse this value is missing.

In reporting income from other family members, the respondent selects from 1-14 categories of income. In the 1993 Young Women and in the 1992 Mature Women surveys, a response of "nothing" corresponds to category 14, although this category is not included in the derivation for other family income. Category 14 may have been re-coded for the 1992 Mature Women, as no "14" appears in the code-block. For the 1993 Young Women, 157 respondents who report category 14 (see R15194) are not included in the derivation of key income. Because there are 40 cases with a value of 0 and 377 cases with a value of 14 in the 95 data, "14" is added to the derivation for this component of key income, and 0 has been allowed (outside the stated categories but assumed to reflect a dollar amount).

Asset amounts attributable to IRA/Keogh accounts, not available in the 1993 Young Women data, have been added. The general format for including this component of assets is taken from the 1990 Older Men's Appendix 28.

Stepladders have been added for estimating amounts of savings, savings bonds, stocks, and IRA accounts. When the respondent reports "yes" to having the asset but does not provide the amount, further yes/no responses to certain stated amounts provide boundaries for calculating a midpoint that estimates the amount of the asset. Given two "yes" responses, the upper bound estimates the asset; given two "no" responses, the lower bound estimates the asset. Only those who do not know or refuse to respond to the stepladder are missing on these assets.

Asset amounts attributable to life insurance policies are included using the format provided by the 1990 Older Men's Appendix 28. An additional question in 1995 and the associated skip require respondents to identify whether all their life insurance policies are term policies. Respondents who report no policies and those who report only term policies are given a value of 0 on this asset. Those who report at least one policy that is not a term policy are given the reported cash values of their policies or set to missing on this asset.

For business income of the respondent, business income of spouse, farm income, and rental income, two new categories in the "Did you have. " type questions are loss amount (3) and break-even (4). A response of 4 is assigned $0. For a response of 3, the dollar amount is included in the derivation as a negative amount.

Derivations are revised to reflect the current missing value treatment.

Only those respondents with outcomes (R34919) of 201 through 205 (complete or partially complete interviews) are included in derivations.

In calculating the income from the respondent's wages, if no employers are on the roster (from CK-RSP-A [R18103]) and the amount of wages is don't know/refuse/missing, "RWAGE" is given a value of 0.

Money from life insurance settlements has been included in prior years as other income (cf. 1990 Older Men). In the 1995 survey, life insurance settlements over $10,000 are included with net family assets because there is insufficient information to determine whether money reported has been received in the past 12 months. Any respondent who responds "no" to the lead-in or who responds "yes" to the lead-in but gives an amount less than $10,000 is assigned 0 for the derivation of this component of assets. Settlements of $10,000 or more contribute to net family assets in the stated amount.
IF (R3386700=0 | R3396100=0) & (R3386800<0 | R3386800=0) & (R3386900<0 | R3386900=0) & (R3387000<0 | R3387000=0) THEN IREAL95=0;
ELSE IF R3386800>=0 & R3386900>=0 & R3387000>=0
THEN IREAL95=R3386800-R3386900-R3387000;
IF IREAL95>=0, THEN ASSET95=ASSET95+IREAL95; ELSE C95=C95+1;
IF R3387100<0 & (R3387200<0 | R3387200=0) THEN DEBT95=0;
ELSE IF R3387200<0 THEN DEBT95=R3387200;
IF DEBT95>=0, THEN ASSET95=ASSET95-DEBT95; ELSE C95=C95+1;
IF C95<0 & ASSET95<99999 & ASSET95<99999 THEN ASEXCA95=-;
IF C95=0 THEN DO; IF asset95 = & ASSET95<99999 THEN ASEXCA95=-99999;
IF ASSET95=-99999 THEN ASEXCA95=99999;
IF ASSET95=-99999 & ASSET95<999999 THEN ASEXCA95=ASSET95;
END.
if R33945000 = 0 & (R33946000 | R33946000). then srepli = 0;
if R33946000 = 0 THEN SREP1 = R33946000;
if R33947000 = 0 & (R33948000 | R33948000). then sreml1 = 0;
if R33949000 = 0 & (R33949000 | R33949000). then srefed = 0;
if R33950000 = 0 THEN SREPED = R33950000;
if R33951000 = 0 & (R33952000 | R33952000). then sreste = 0;
if R33952000 = 0 THEN SRESTE = R33952000;
if R33953000 = 0 & (R33954000 | R33954000). then sreml2 = 0;
if R33954000 = 0 THEN SREUNI = R33954000;
if R33955000 = 0 & (R33956000 | R33956000). then sreira = 0;
if R33956000 = 0 THEN SREIRA = R33956000;
if R33957000 = 0 & (R33958000 | R33958000). then sreoth = 0;
if R33958000 = 0 THEN SREOTH = R33958000;
if (R33959000 | R33959000). & (R33960000 | R33960000) = 0 THEN SREOTH = R33958000;
if (R33959000 | R33959000). & (R33960000 | R33960000) = 0 then farm = R33960000;
if (R33960000 | R33960000) = 0 then farm = R33960000;
if (R33961000 = 0 | R33962000 = 0) & (R33962000 | R33962000) = 0 then rent = 0;
if (R33961000 = 1 | R33962000 = 0) THEN RENT = R33962000;
if (R33961000 = 1 | R33962000 = 0) & (R33962000 | R33962000) = 0 then intt = 0;
if (R33964000 = 0) THEN INTT = R33964000;
if (R33965000 = 0 & (R33966000 | R33966000).) & (R33967000 | R33967000).) then foods = 0;
if (R33966000 = 0 & (R33967000 | R33967000).) THEN FOODS = R33966000*R33967000;
if (R33968000 = 0 & (R33969000 | R33969000).) & (R33970000 | R33970000).) then afdc = 0;
if (R33969000 = 0 & (R33970000 | R33970000).) THEN AFDC = R33969000*R33970000;
if (R33971000 = 0 & (R33972000 | R33972000).) & (R33973000 | R33973000).) then ssi = 0;
if (R33972000 = 0 & (R33973000 | R33973000).) THEN SSI = R33972000*R33973000;
if (R33974000 = 0 & (R33976000 | R33976000).) & (R33977000 | R33977000).) THEN all1 = 0;
if (R33975000 = 0 & (R33977000 | R33977000).) THEN ALI = R33975000-R33977000;
if (R33975000 = 0 & (R33977000 | R33977000).) THEN all1 = R33975000;
if (R33976000 = 0 & (R33977000 | R33977000).) THEN all1 = R33976000;
if (R33980000 = 0 & (R33980000 | R33980000).) THEN chdsup = 0;
else if (R33980000 = 0 & (R33980000 | R33980000).) THEN chdsup = 0;
else if (R33987000 = 2 & R34016000 = 0) THEN chdsup = R34016000;
else if (R33987000 = 1 & R34024000 = 0) THEN chdsup = R34024000;
else if (R33987000 = 3 & R34016000 = 0 & R34024000 = 0) THEN chdsup = R34016000*R34024000;
if (R34029000 = 0 & (R34032000 | R34032000).) & (R34031000 = 0 | R34031000).) THEN chdsup = 0;
else if (R34031000 = 0 & R34033000 = 0) THEN chdsup = R34031000;
else if (R34033000 = 0) THEN chdsup = R34033000;
if (R34035000 = 0 & (R34037000 | R34037000).) & (R34036000 | R34036000).) & (R34036000 | R34036000).) THEN hpcde = 0;
else if (R34037000 = 0 & hpcde = 0) THEN hpcde = 0;
else if (R34036000 = 0 & hpcde = 0) THEN hpcde = 0;
else if (R34038000 = 0 & hpcde = 0) THEN hpcde = 0;
if chdsup = 0 & rchd = 0 & hpcde = 0 THEN child = chdsup * rchd * hpcde;
if (R34040000 = 0 & (R34040100 = 0)) THEN other = 0;
if (R34041000 = 0) THEN OTHER = R34041000;
if (R34043000 = 2 & (R34044000 | R34044000).) THEN fam = 0; else IF (R34043000 = 1) THEN DO,
IF R34044000 = 1 THEN FAM = 2000; ELSE IF R34044000 = 2 THEN FAM = 5000;
ELSE IF R34044000 = 3 THEN FAM = 6750; ELSE IF R34044000 = 4 THEN FAM = 8250;
ELSE IF R34044000 = 5 THEN FAM = 12500; ELSE IF R34044000 = 6 THEN FAM = 16250;
ELSE IF R34044000 = 7 THEN FAM = 18750; ELSE IF R34044000 = 8 THEN FAM = 22500;
ELSE IF R34044000 = 9 THEN FAM = 30000; ELSE IF R34044000 = 10 THEN FAM = 42500;
ELSE IF R34044000 = 11 THEN FAM = 62500; ELSE IF R34044000 = 12 THEN FAM = 87500;
ELSE IF R34044000 = 13 THEN FAM = 1000000;
else if (R34044000 = 141) THEN fam = 0; END;
SUMMATION [R34068.]

SUMMATN := IF RWAGE ^= THEN DO;
if summatn ^= then summatn = rwage + summatn; else SUMMATN = RWAGE; END;
IF RBUS ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RBUS + SUMMATN; ELSE SUMMATN = RBUS; END;
IF RUNEM ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RUNEM + SUMMATN; ELSE SUMMATN = RUNEM; END;
IF RSUB ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RSUB + SUMMATN; ELSE SUMMATN = RSUB; END;
IF RSO Java` THEN DO;
IF SUMMATN ^= THEN SUMMATN = RSO + SUMMATN; ELSE SUMMATN = RSO; END;
IF RVET ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RVET + SUMMATN; ELSE SUMMATN = RVET; END;
IF RCOM ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RCOM + SUMMATN; ELSE SUMMATN = RCOM; END;
IF RSSD ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RSSD + SUMMATN; ELSE SUMMATN = RSSD; END;
IF RDIS ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RDIS + SUMMATN; ELSE SUMMATN = RDIS; END;
IF SWAGE ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SWAGE + SUMMATN; ELSE SUMMATN = SWAGE; END;
IF SBS ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SBS + SUMMATN; ELSE SUMMATN = SBS; END;
IF SUNEM ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SUNEM + SUMMATN; ELSE SUMMATN = SUNEM; END;
IF SSSUB ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SSSUB + SUMMATN; ELSE SUMMATN = SSSUB; END;
IF SSO ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SSO + SUMMATN; ELSE SUMMATN = SSO; END;
IF SVET ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SVET + SUMMATN; ELSE SUMMATN = SVET; END;
IF SCOM ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SCOM + SUMMATN; ELSE SUMMATN = SCOM; END;
IF SSSD ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SSSD + SUMMATN; ELSE SUMMATN = SSSD; END;
IF SDIS ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SDIS + SUMMATN; ELSE SUMMATN = SDIS; END;
IF SFARM ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SFARM + SUMMATN; ELSE SUMMATN = SFARM; END;
IF RENT ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = RENT + SUMMATN; ELSE SUMMATN = RENT; END;
IF INTT ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = INTT + SUMMATN; ELSE SUMMATN = INTT; END;
IF FODS ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = FODS + SUMMATN; ELSE SUMMATN = FODS; END;
IF AFDC ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = AFDC + SUMMATN; ELSE SUMMATN = AFDC; END;
IF ALI ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = SSI + SUMMATN; ELSE SUMMATN = SSI; END;
IF CHILD ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = CHILD + SUMMATN; ELSE SUMMATN = CHILD; END;
IF OTHER ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = OTHER + SUMMATN; ELSE SUMMATN = OTHER; END;
IF FAM ^= THEN DO;
IF SUMMATN ^= THEN SUMMATN = FAM + SUMMATN; ELSE SUMMATN = FAM; END;
IF SUMMATN ^= & R3388300 ^= THEN DO; IF R3388300 = 1 THEN SUMMATN = 2000;
ELSE IF R3388300 = 2 THEN SUMMATN = 5000; ELSE IF R3388300 = 3 THEN SUMMATN = 6750;
ELSE IF R3388300 = 4 THEN SUMMATN = 8250; ELSE IF R3388300 = 5 THEN SUMMATN = 12500;
ELSE IF R3388300 = 6 THEN SUMMATN = 16250; ELSE IF R3388300 = 7 THEN SUMMATN = 18750;
ELSE IF R3388300 = 8 THEN SUMMATN = 22500; ELSE IF R3388300 = 9 THEN SUMMATN = 30000;
ELSE IF R3388300 = 10 THEN SUMMATN = 42500; ELSE IF R3388300 = 11 THEN SUMMATN = 62500;
ELSE IF R3388300 = 12 THEN SUMMATN = 87500; ELSE IF R3388300 = 13 THEN SUMMATN = 100000;
ELSE IF R3388300 = 14 THEN SUMMATN = 0; END;
IF SUMMATN ^= THEN SUMMATN = 99999; END;
IF SUMMATN ^= THEN SUMMATN = 99999;
HIGHEST-GRADE-COMP [R34766.]: UPDATE OF HIGHEST GRADE COMPLETED

The goal of this variable is to update the Highest Grade Completed for respondents who have obtained some additional education since the date of the last interview. Therefore, the highest grade completed for 1995 is the same as the highest grade completed in the previous survey year if the respondent has not obtained any additional schooling since the date of last interview.

Part A contains the SPSS code used to create the 1995 Highest Grade Completed for the Mature Women respondents. Part B contains the code for the Young Women cohort.

A. SPSS code for Mature Women

Variable labels:
R0000100 "ID_CODE 67"
R0079000 "HGC BY R, 67"
R0381500 "HGC BY R, 77"
R0989700 "HGC BY R, 89 (TO DT)"
R1601400 "SAMPLING WEIGHT, 95"
R3476600 "HGC, 95 REV"
R3477000 "GRADE_ATTNDING, 95"
R3477100 "HGC, 95"
R3477600 "TYPE_COLL_DEG RCV SNC L_INT, 95"

Miss val R0000100 (-998, -999)
Miss val R0079000 R0381500 R0989700 (-127, -128)
Miss val R1601400 R3476600 R3477000 R3477100 R3477600 (-4, -5)

COMPUTE WEIGHT = R1601400

**** Use this code to create the previous measure of
**** HGC if using the MW data
Numeric HGC_OLD
Do if WEIGHT > 0 / *** selects the MW with 1995 interviews
IF SYSMIS(HGC_OLD) HGC_OLD = R0989700 /*HGC89_MW
IF SYSMIS(HGC_OLD) HGC_OLD = R0381500 /*HGC77_MW
IF SYSMIS(HGC_OLD) HGC_OLD = R0079000 /*HGC67_MW
End if

*****************************************************************
**** Create REVISED HGC variable
**** RECODING NEGATIVE VALUES in HGC77_MW HGC89_MW
****
**** Old New Label
****
**** -1 96 Elementary, year unspecified
**** -2 97 High school, year unspecified
**** -3 98 College, year unspecified
**** -4 93 Kindergarten
**** -5 94 Pre-school
**** -7 95 Nonacademic degree or diploma
**** -6 12 GED

*****************************************************************

RECODE HGC_OLD (-1=96) (-2=97) (-3=98) (-4=93) (-5=94) (-6=12)
(-7=95) (ELSE = COPY)

*** Initializes the 1995 HGC variable to previous years' value
Compute HGC_95 = HGC_OLD
VAR LAB HGC_95 'HIGHEST-GRADE-COMPLETE: Updated using 95 data'

***** convert EAT-9 to numbers corresponding to years of school
***** (new variable used in HGC is called EAT_9x)
RECODE R3477600 (1=14) (2=16) (3,4=18) into EAT_9x
VAL LAB EAT_9x 14 'Assoc.' 16 'BA, 16 yrs' 18 'PhD or more'

*****************************************************************
***** convert EAT-9 to numbers corresponding to years of school
*** (new variable used in HGC is called EAT_9x)
RECODE R3477600 (1=14) (2=16) (3=18) into EAT_9x
VAL LAB EAT_9x 14 'Assoc.' 16 'BA, 16 yrs' 18 'PhD or more'

*====================================================================*
** BUILD 1995 Highest Grade Completed **
** New values of HGC are coded only if the 1995 grade is higher **
** than the previous HGC **
*====================================================================*
** ** Code if CURR_ATT **
RECODE R3477000 (1=9) (2=10) (3=11) (4=12) (5=13) (6=14) (7=15) (8=16)
          (9=17) (10=18) (11=95) (-1=-1) (-2=-2) INTO CURR_ATT
VAR LAB CURR_ATT 'Grade currently attending'

DO IF (CURR_ATT>-1 & CURR_ATT<19) /* excludes DK, Sysmis, Nonacademic
    IF (CURR_ATT le HGC_OLD) HGC_95 = HGC_OLD /* reset to OLD if CURR < OLD
END IF

** ** Code if R is DONE ATTENDING **
****** The OLD HGC is recorded only if DONE_ATT >= old HGC
RECODE R3477100 (1=9) (2=10) (3=11) (4=12) (5=13) (6=14) (7=15) (8=16)
          (9=17) (10=18) (11=95) (-1=-1) (-2=-2) INTO DONE_ATT
VAR LAB DONE_ATT 'Grade completed since DOLI'

DO IF (DONE_ATT>-1 & DONE_ATT<19) /* Finished attending & not missing
    IF (DONE_ATT lt HGC_OLD) HGC_95 = HGC_OLD
END IF

IF EAT_9x > HGC_OLD HGC_95 = EAT_9x

****** LABELS ******
VAL LAB HGC 95
  1 '1st grade' 2 '2nd grade' 3 '3rd grade'
  4 '4th grade' 5 '5th grade' 6 '6th grade'
  7 '7th grade' 8 '8th grade' 9 '9th grade'
 10 '10th grade' 11 '11th grade' 12 '12th grade or GED'
 13 '1st yr college' 14 '2nd yr college' 15 '3rd yr college'
 16 '4th yr college' 17 '5th yr college' 18 '6th+ yr college'
 95 'Non-academic degree or diploma'

Rename Vars (HGC_95 = R3477600)