

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)
COMPUTE 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)
```

Recode NUM_GAP (0=2) (1 thru hi = 1) into R2550600
*FOR YW data only: If R0000100 = 5031 R2550600 = 3

HWH (Husband/Partner Work History) [R25578.]

Do if (R1664500 le 4) /** if HUSB or PARTNER
Compute NUM_HWH = Nvalid(R2558000, R2558100, R2559300, R2571900)
End if
Recode NUM_HWH (0=2) (1 thru hi = 1) into R2557800

HOJ (Husband/Partner On Jobs) [R25724]

Do if (R1664500 le 4) /** if HUSB or PARTNER
Compute R2572400 = 3 /* default value of '3', incomplete, irregular
Count NUM_HOJ = R2572500 R2572600 R2637400 (LO thru HI)
End if
Recode NUM_HOJ (0=2) (1 thru 4 = 1) into R2572400

HES (Husband/Partner Employer Sort) [R26841.]

Do if (R1664500 le 4) /** if HUSB or PARTNER
Count NUM_HES = R2684100 to R2685300 R2688700 (LO thru HI)
COMPUTE R2684100 = 1
If (NUM_HES = 0) R2684100 = 2
End if

HEA (Health) [R33513.]

COUNT ST_HEA = R3351700 R3363400 (LO thru HI)
COUNT END_HEA = R3370800 R3379400 (LO thru HI)

IF (ST_HEA = 2) & (END_HEA = 2) R3351300 = 1 /* section completed
IF (ST_HEA = 0) R3351300 = 2 /* prior breakoff
IF (ST_HEA ge 1) & (END_HEA = 0) R3351300 = 3 /* incomplete data

OFB (Other Family Background) [R34070.]

COUNT OFB_ST = R3450800 R3475500 (LO THRU HI)
COUNT OFB_END = R3475600 R3475800 (LO THRU HI)
IF (OFB_ST = 0) & (OFB_END = 0) R3407000 = 2 /** Prior break off
IF (OFB_ST ge 1) & (OFB_END = 0) R3407000 = 3 /* irregular or incomplete
IF (OFB_ST = 0) & (OFB_END ge 1) R3407000 = 3 /* V-Stet, irregular
IF (OFB_ST ge 1) & (OFB_END ge 1) R3407000 = 1 /* Section completed
IF ANY(R0000100, 0675, 2224) R3407000 = 3 /* YW only

EAT (Education and Training) [R34765.]

COMPUTE VAL_EAT = NVALID(R3476700, R3478600)
RECODE VAL_EAT (0=2) (2 = 1) (1 = 3) INTO R3476500

MOB (Mobility) [R34811.]

COUNT ST_MOB = R3481200 (LO THRU HI) /* 1st universe Q
COUNT END_MOB = R3483800 (LO THRU HI) /* Last universe Q
IF (ST_MOB = 1) & (END_MOB = 1) R3481100 = 1
IF (ST_MOB = 0) R3481100 = 2
IF (ST_MOB = 1) & (END_MOB = 0) R3481100 = 3

ACP (Attitudes and Contact Persons) [R34856.]

COUNT ST_ACP = R3485700 (LO THRU HI) /* 1st universe Q
COUNT END_ACP = R3488300 (LO THRU HI) /* Last universe Q
IF (ST_ACP = 1) & (END_ACP = 1) R3485600 = 1
IF (ST_ACP = 0) R3485600 = 2
IF (ST_ACP = 1) & (END_ACP = 0) R3485600 = 3

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 time 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-81I "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 time unit topcode methods described above.

RNI95-CROSS-SECTION [R16011.00]: CROSS-SECTIONAL REASON FOR NONINTERVIEW

This code creates the reason for non-interview variable that is consistent with the cross-sectional RNI variable found in previous releases of the Mature Women and Young Women data. It is "cross-sectional" because respondents who were out-of-scope in 1995 are assigned a value of -5 rather than a substantive RNI value (e.g., bad address, refusal). Thus, the universe of cases with a value greater than or equal to -4 are those where contact was attempted and an interview was either obtained (RNI = -4) or not obtained (RNI = 1 through 14).

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-interview and assigned a value of 300 in the variable called OUTCOME [R34919.]. However, in the RNI-95 and ISR-95 variables these cases were assigned to the "other" category.

Mature Women:

Variable Labels

R0000100 "ID_CODE 67"

R3491900 "INT_RMRKS - OUTCOME OF INT, 95 CENSUS"

```
***** Identifies the cases that are 1995 Congressional Refusals. *****
***** The list of case Ids came from the Census bureau. *****
Do if ANY(R0000100, 1227, 1798, 2222, 3606, 3740, 4552, 1991)
Compute R1601100 = 14
End if
```

Young Women:

Variable Labels

R0000100 "ID_CODE 68"

R3491900 "INT_RMRKS - OUTCOME OF INT, 95 CENSUS"

```
***** Identifies the cases that are 1995 Congressional Refusals. *****
***** The list of case Ids came from the Census bureau. *****
Do if ANY(R0000100, 0822, 1036, 2206)
Compute R1601100 = 14
End if
```

For both cohorts:

```
Recode R3491900 (214=5) (216=1) (217=6) (218=9) (219=11) (223=7)
              (234=8) (250=10) (251=13) (260 = 4) (261, 270=2)
              (300=11) into R1601100
```

```
VAR LAB R1601100 'Recoded 1995 RNI, X-sectn--matches past yrs'
VAL LAB R1601100
```

- 1 'Unable to contact-Reason unspecified'
- 2 'Mover-good address, interview impossible'
- 3 'Mover-good address, unable to IW after repeated attempts'
- 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'
- 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 IW obtained'
- 5 'Not in 95 sample'

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'
```

ISR-95 [R16013.]: INTERVIEW STATUS RECORD

This variable is a summary of the interview status of each respondent in the cohort. It combines information collected in 1995 and with data from previous surveys for the cases that were out-of-scope in 1995.

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 R1601300 = R1293400    /* previous ISR from 1992 MW
ELSE
RECODE  R1601200  (-4 = 1)   (10 = 2)   (9 = 3)   (12 = 4)   (7 = 5)
          (1,2,3,4,5,6,8,11,13 = 6)   (14 = 8)   INTO    R1601300
END IF
IF R3491900 = 300   R1601300 = 6          /* put them into the 'Other' category
```

Young Women:

```
DO IF (R3491900 < 0)          /* selects cases NOT in the 1995 sample
COMPUTE R1601300 = R1571800    /* previous ISR, 1993 YW
ELSE
RECODE  R1601200  (-4 = 1)   (10 = 2)   (9 = 3)   (12 = 4)   (7 = 5)
          (1,2,3,4,5,6,8,11,13 = 6)   (14 = 8)   INTO    R1601300
END IF
IF R3491900 = 300   R1601300 = 6          /* put them into the 'Other' category

VAL LAB R1601300
1  '1995 IW obtained'
2  'Non-IW: Deceased'
3  'Non-IW: Refused'
4  'Non-IW: Dropped after 2 consecutive years as Non_interview'
5  'Non-IW: Armed Forces'
6  'Non-IW: Other + training cases'
7  'Code not used in 95'
8  'Non-IW: Congressional refusal'
```

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*.

NUMFAM [R16646.]: NUMBER OF FAMILY MEMBERS IN HOUSEHOLD

A family member is a person related to the respondent who lives in the respondent's household. The total number of family members in the household is determined by counting the number of persons listed in the household person roster (R1CNTR02 to R1CNTR17) who meet two conditions: he or she must live in the household (HRC-6 = 1) and must either be related to the respondent (HRC-8 = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, or 13) or must be living with the respondent as though married (defined as HRC-14=1 and HRC-15 = 1). Thus, this count of the number of family members in the household includes husbands or partners, biological children, parents, in-laws, step children, and adopted children.

The SPSS code used to create NUMFAM (R16646.) for the Mature Women and Young Women cohorts is shown in parts A and B, respectively.

Part A. SPSS code for Mature Women

```
Miss val ALL (-4, -5)
Compute WEIGHT = R1601400

Do if WEIGHT > 0          /* selects cases with a 1995 interview
DO REPEAT
  X =  R1634800 R1634900 R1635000 R1635100 R1635200 R1635300
      R1635400 R1635500 R1635600 R1635700 R1635800 /*hrc-8
  / INHH = R1631400 R1631500 R1631600 R1631700 R1631800 R1631900
      R1632000 R1632100 R1632200 R1632300 R1632400 /*hrc-6
  / REL = REL2 to REL12
  / counter = 2 to 12

COMPUTE REL = 0          /** INITIALIZE TO 0

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 R1612300 = 1 REL8 = 1 /* hrc-4b.08

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) + 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)
```

Part B. SPSS code for Young Women

```
Miss val all (-4, -5)
Compute WEIGHT = R1601400

Do if WEIGHT > 0          /* selects cases with a 1995 interview
DO REPEAT
  X =  R1634800 R1634900 R1635000 R1635100 R1635200 R1635300
      R1635400 R1635500 R1635600 R1635700 R1635800 R1635900
      R1636000 R1636100 R1636200 R1636300 /*hrc-8
  / INHH = R1631400 R1631500 R1631600 R1631700 R1631800 R1631900
      R1632000 R1632100 R1632200 R1632300 R1632400 R1632500
      R1632600 R1632700 R1632800 R1632900 /*hrc-6
  / REL = REL2 to REL17
  / counter = 2 to 17

COMPUTE REL = 0          /** INITIALIZE TO 0
```



```

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)

```

NUMHH [R16647.]: NUMBER OF HOUSEHOLD MEMBERS

This created variable counts the number of household members residing in the respondent's household. Household members do not have to be related to the respondent, so boarders and other non-relatives are included in the count. The variable is created using information contained in questions HRC-4b (does respondent's husband live in the household), HRC-6 (does this person live in the household), and HRC-14 and HRC-15 (does respondent's partner live in the household).

Part A. SPSS code for Mature Women

```
Do if WEIGHT > 0
DO REPEAT
  X = R1631400 R1631500 R1631600 R1631700 R1631800 R1631900
      R1632000 R1632100 R1632200 R1632300 R1632400 /*hrc-6
  / HHMEM = HH2 to HH12
  / COUNTER = 2 to 12

COMPUTE      HHMEM = 0          /* initialize to 0 for summing
IF   X = 1      HHMEM = 1
IF   R1611700 = 1   HH2 = 1 /* hrc-4b.02
IF   R1612300 = 1   HH8 = 1 /* hrc-4b.08

DO IF COUNTER = R1664000 & R1663900 = 1 /* hrc-16 & hrc-15
COMPUTE HHMEM = 1
END IF
END REPEAT
End if

COMPUTE NUM_HH = SUM(HH2, HH3, HH4, HH5, HH6, HH7, HH8, HH9,
                    HH10, HH11, HH12) + 1

VAR LAB NUM_HH 'NUMHH: No. of HH mems in Rs house'
Rename vars (NUM_FAM = R1664700)
```

Part B. SPSS code for Young Women

```
Do if WEIGHT > 0 /* selects cases with a 1995 interview
DO REPEAT
  X = R1631400 R1631500 R1631600 R1631700 R1631800 R1631900
      R1632000 R1632100 R1632200 R1632300 R1632400 R1632500
      R1632600 R1632700 R1632800 R1632900 /*hrc-6
  / HHMEM = HH2 to HH17
  / COUNTER = 2 to 17

Compute      HHMEM = 0          /* initialize to 0 for summing
IF   X = 1      HHMEM = 1
IF   R1611700 = 1   HH2 = 1 /* hrc-4b.02
IF   R1611800 = 1   HH3 = 1 /* hrc-4b.03
IF   R1611900 = 1   HH4 = 1 /* hrc-4b.04
IF   R1612000 = 1   HH5 = 1 /* hrc-4b.05

DO IF COUNTER = R1664000 & R1663900 = 1 /* hrc-16 & hrc-15
Compute HHMEM = 1
End if
END REPEAT
End if

COMPUTE NUM_HH = SUM(HH2, HH3, HH4, HH5, HH6, HH7, HH8, HH9,
                    HH10, HH11, HH12, HH13, HH14, HH15, HH16,
                    HH17) + 1

Rename vars (NUM_FAM = R1664600)
```

CK-OJS-Q [R17560.-R17567.]

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.

EMPLOYER COMPARISON VARIABLES [R18153.-R18161.]

The revised employer comparison items involve a comparison of the current, and all previous, survey week employers. Each unique employer encountered in surveys subsequent to the initial year was assigned an identifying code. Therefore the values in these variables show moves to new employers, to self-employment or returns to previously encountered employers. They do not indicate a simple tally of the changes of employment from survey to survey. For example, the series of employer comparison variables in 1971 for a respondent who changed employers in 1968 and 1969 and returned to her 1966 employer would be coded 1 (for 1966), 2 (1968), 3 (1969), and 1 (1971). The variables are constructed by Census Bureau to preserve the confidentiality of respondents' employer names. See Appendix 36 for more details on the construction and interpretation of the employer comparison variables.

1980 INDUSTRY [R19005–R19013] AND OCCUPATION CODES [R19041.–R19049.]

```
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 = 740 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 =019;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
```

```
IF XX = 804 THEN YY =805;ELSE
IF XX = 864 THEN YY =863;ELSE
IF XX = 865 THEN YY =864;ELSE
IF XX = 866 THEN YY =865;ELSE
IF XX = 867 THEN YY =866;ELSE
IF XX = 868 THEN YY =867;ELSE
IF XX = 874 THEN YY =873;ELSE
IF XX = 903 THEN YY = -4;ELSE
IF XX = 904 THEN YY = -4;ELSE
IF XX = 905 THEN YY = -4;ELSE
IF XX = 909 THEN YY = -4;
ELSE YY = XX; END;
```

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(DOI - DOLI)) / 7)
VAR LAB WBID '# weeks btwn intvs, rounded off '
```

WORK, WUMP, WOLF [R25502., R25503., R25504.]: WEEKS VARIABLES

Collectively, these variables are referred to as the “weeks” variables because they are summary measures of the number of weeks working (WORK), unemployed (WUMP) and out of the labor force (WOLF).

The weeks working variable (WORK) is the number of weeks in the time between the date of the last interview and the date of 1995 interview during which the respondent was working at least 1 hour per week for at 1 or more employers. Partial weeks are rounded to the nearest whole number, so that a work spell of 50 days is computed as 7 weeks, while 55 days is rounded to 8 weeks of work. Working supersedes unemployed and out-of-labor-force status, so that a week in which the respondent was working contributes a “1” to the total number of weeks working, even if the respondent was laid off or looking for work from a second job during the same time period.

The number of weeks unemployed (WUMP) is defined as the number of weeks the respondent was laid off from work or looking for work during a spell when she was not working.

The number of weeks the respondent is out of the labor force (WOLF) is defined as the number of weeks not working and not looking for work, or not working and not on layoff. In other words, if the respondent is not actively seeking employment, and is not currently working at some other job, then she is classified as out of the labor force (OLF).

NOTES:

1. The SPSS code shown here was written to create the weeks variables from a data set that contained respondents from both cohorts. Thus, variables missing in one cohort are included here because they existed for the other cohort.
2. The calculation of the weeks working, weeks unemployed, and weeks out of the labor force did not include any gaps that may have existed *within* a job.
3. The number of weeks working, unemployed, or out of labor force should sum to the total number of weeks between interview dates. However, due to rounding of the number of weeks between interview dates and the number of weeks between start and stop dates with employers, the total number of weeks between the interview dates (WBID) does not always equal the sum of three weeks variables (WORK, WUMP and WOLF.)
4. The start- and stop-date variables referred to in the program are not included on the public release of the 1995 data. However, the variables containing employer dates are found throughout the OJS section, such as in OJS-5A, OJS-8, etc.

```
Compute SERIAL = R0000100
```

```
*****
**** Combines the separate DOI variables into a single DOI variable ****
****
**** R1602300 is the Date of Interview, day (2-digit) ****
**** R1602400 is the Date of Interview, month (2-digit) ****
**** R1602500 is the Date of Interview, year (4-digit) ****
**** (One case from MW (ID=1655) cohort has its DOI changed ****
**** from 1/1/80 to -3/-3/95.) ****
*****
*** HAND EDIT for case 11655 ****
DO IF (R0000100 = 1655)
  COMPUTE R1602300= -3
  COMPUTE R1602400= -3
  COMPUTE R1602500 = 1995
END IF

***** DOI of everyone else *****
IF R1602500 ne 1995 R1602500 = 1995
IF (R1602300< 1) or (R1602300> 31) R1602300= -3
IF (R1602400< 1) or (R1602400> 12) R1602400= -3

MISS VAL R1602300 R1602400 (-3)

COMPUTE DOI = DATE.DMY (R1602300,R1602400,R1602500)
```



```
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 OJS-1 = "yes". This will
** fill in missing DOLI and the START date when R agrees
** that she worked for the Ename 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
```

```
    OJS_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) & (OJS_1 = 1) /* R had job activity & Start=Doli
DO IF SYSMIS(START) & NVAL( 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 #-WKS-BTWN-INTVW-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(R2551000 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. **
*****
```

```

COMPUTE HIT_1B = NVALID(R2557300) /* 'Did R go thru GAP-1B?

*=====
*           Find Rs who missed or hit both the Gaps questions
**          This relies on the data as received from Census
*=====

COMPUTE HIT_GAP = 0
IF (HIT_1 = 0) & (HIT_1B = 0)      HIT_GAP = 0
IF (HIT_1 = 1) or (HIT_1B = 1)     HIT_GAP = 1
VAR LAB HIT_GAP 'Did R hit 1 gap question at least?'
VAL LAB HIT_GAP 0 'No gap Q hit' 1 '1 Gap Q hit, at least'

***=====
***          Create ALLWRK1
*** Identify Rs who worked for the entire spell between interviews ==
***=====

** Calculate DAYS Between Interview Dates DBID.D
COMPUTE DBID = (CTIME.DAYS(DOI - DOLI)) /* #DAYS btwn interviews
COMPUTE DOI.D = (CTIME.DAYS(DOI - DOLI)) + 1 /* DOI.D is = #DBID + 1
COMPUTE DOLI.D = 1 /* DOLI.D is always = DAY1 for every R

IF SYSMIS(DOLI) DOLI.D = -3
IF SYSMIS(DOLI) DBID = -3
IF SYSMIS(DOLI) DOI.D = -3
MISS VAL DOLI.D DBID (-3)

** Create and Initialize DAYS array variables ***
DO IF (DOLIMISS = 0) /* only if DBID can be computed
VECTOR DAY(5675) /* creates 5675 variables, DAY1 to DAY5675
LOOP I# = 1 to DBID /* execute the DO LOOP as many times as there are DBID
COMPUTE DAY(I#) = 0 /* assign 0 to DAY array vars between DOLI and DOI
END LOOP
END IF
EXECUTE

***=====
*** This will create a vector of dummy variables, 1 for each day ==
*** from the DOLI to the DOI. This "DAY array" will be used ==
*** to identify Rs who were working every day between interviews. ==
*** If R was working the DAYn variable = 1. DAY variables ==
*** falling between the DOLI and DOI are assigned a 0 by default. ==
*** If the number of "1s" in the DAY array is within 3.49 days ==
*** of number of days between interview dates then the R was ==
*** defined as working the entire spell, and ALLWRK1 = 1. ==

***** fill in 1 for days working
VECTOR DAYS = DAY1 to DAY5675 /* assigns 1=Working to each DAY variable
DO REPEAT
START = START1 START2 START3 START4 START5 START6
START7 START8 START9
/ END = END1 END2 END3 END4 END5 END6
END7 END8 END9
/ St.DAY = ST1.D ST2.D ST3.D ST4.D ST5.D ST6.D ST7.D
ST8.D ST9.D
/ End.DAY = END1.D END2.D END3.D END4.D END5.D END6.D
END7.D END8.D END9.D
/ DWORK = DWORK1 DWORK2 DWORK3 DWORK4 DWORK5 DWORK6 DWORK7
DWORK8 DWORK9

*** Convert employer dates to # Days after DOLI
COMPUTE ST.DAY = (CTIME.DAYS(START - DOLI)) + 1
COMPUTE END.DAY = (CTIME.DAYS(END - DOLI)) + 1

**** Compute #Days working for each Ename
COMPUTE DWORK = (END.DAY - ST.DAY)

```

```

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 DAY5675 (1) /* #days working, from DAYS array
COMPUTE WORK.rnd = rnd(WORKD / 7) /* Rounded: #weeks worked, from DAYS array
COMPUTE NOWORKD = DBID - WORKD
COMPUTE NOWORKW = RND(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(DOI - DOLI)) / 7) /* #weeks btwn intvs, not rounded
COMPUTE WBID.RND = RND((CTIME.DAYS(DOI - DOLI)) / 7) /* WBID, rounded off
COMPUTE DOI.W = ((CTIME.DAYS(DOI - 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.RND = -3

MISS VAL DOLI.W WBID.RND (-3)

*=====
*****
***** BUILD UNIVERSE *****
***** Classifies according to work & gap status *****
*****
*** 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-1B 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

```

```

0 'DOLI missing'
1 'No Gap/Workd all DBID'
2 'No work, GAP-1B only'
3 'GAP-1 only, activity'
4 'GAP-1B only, activity'
5 'Both gaps, activity'
-9 'NA, not yet coded'

```

```

*** Convert START and END dates to the # of weeks after the DOLI
*=====
***
*** This is done by counting the number of weeks the date
*** is from DOLI and adding 1. For example, if the DOLI is WEEK1,
*** and the Start-Date for Ename 1 is 23 days after the DOLI, then
*** then ST1.W = 4.39 (3.39 + 1), which indicates a Start-date that
*** is a little of 3 weeks from the DOLI.
***
*** Next, compute the # of weeks worked at each Employer. This is done
*** by subtracting the Start-week number computed in step 1 from
*** the End-week number for each employer. Thus, if ST1.1 = 1 and
*** END1.W = 6.71 the number of weeks worked is 5.71, which will
*** be rounded to 6 by the time the actual WEEKS variables are
*** computed.
**=====

```

```

DO REPEAT
  START = START1 START2 START3 START4 START5 START6
        START7 START8 START9
  / END = END1 END2 END3 END4 END5 END6
        END7 END8 END9
  / st.Wk# = ST1.W ST2.W ST3.W ST4.W ST5.W ST6.W ST7.W
            ST8.W ST9.W
  / end.Wk# = END1.W END2.W END3.W END4.W END5.W END6.W
            END7.W END8.W END9.W
  / WORK.WK = WORK1 WORK2 WORK3 WORK4 WORK5 WORK6 WORK7 WORK8 WORK9

```

```

*** Convert employer dates to # weeks after DOLI
COMPUTE ST.Wk# = ((CTIME.DAYS(START - DOLI)) / 7) + 1
COMPUTE END.Wk# = ((CTIME.DAYS(END - DOLI)) / 7) + 1

```

```

**** Build # weeks working variable for each Employer
IF NVAL(START, END) = 1 WORK.WK = -3 /* default if one date is missing
IF NVAL(START, END) = 2 WORK.WK = (END.WK# - ST.WK#) /* # wks working
END REPEAT

```

```

COUNT MISSWRK = WORK1, WORK2, WORK3, WORK4, WORK5, WORK6, WORK7,
              WORK8, WORK9 (-3)
RECODE MISSWRK (1 thru HI = 1) (0=0) /* 1= 1 date missng 0= No dates missing

```

```

VAR LAB ST1.W '# wks from DOLI to Start for Ename 1'
  / ST2.W '# wks from DOLI to Start for Ename 2'
  / ST3.W '# wks from DOLI to Start for Ename 3'
  / END1.W '# weeks from DOLI to End for Ename 1'
  / END2.W '# weeks from DOLI to End for Ename 2'
  / WORK1 '# wks working, END.WK# - ST.WK#'
  / MISSWRK '1 work date missing, weeks working cant be computed'
VAL LAB WORK1 - 3 '1 date missing' 1 '1 week working'

```

```

*=====
** Convert negative gap to 0 (INTER-CK = -2 changed to 0) **
**
** There are YW 4 cases with a negative value in INTER-CK (R25534.). **
** This doesn't make sense; The Employer dates and Gap data for 3 **
** cases are listed below so that the appropriate corrections are **
** made when computing the number of weeks OLF and UMP. **
*
* Serial R2551000 R2553400 Start 1 Stop 1 Start 2 Stop 2 *

```

```

* -----
* 2106      3      -2    11-OCT-93  15-AUG-94  01-AUG-94  14-JUL-95
* 0232      3      -2    06-OCT-93  15-JAN-94  01-JAN-94  24-JUL-95
*
* Serial R2551000 R2554100 Start 8 Stop 8 Start 9 Stop 9
* -----
* 4010      3      -2    01-JAN-94  15-JUN-95  01-JUN-95  20-JUL-95
*
*
** 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.
*
* Serial R2551100 R2553500 Start 2 Stop 2 Start 3 Stop 3
* -----
* 3381      3      -61    11-OCT-92  15-JAN-94  12-DEC-94  14-JUN-95
*
**
*=====

```

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
*** 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) ****
***
*** The meaning of EMPLIST values:
*** This variable is not really necessary to compute the gaps
*** nor the Weeks variables but it helps in checking the
*** data and fixing bugs in the code. The gap length following each
*** Employer name (Ename) is based on number of days between that
*** job's End-Date and the Start-date of the next most recent job,
*** or the End-date to the Date-of-interview (DOI) if there are no
*** more Enames after this one. Because of blank columns in
***

```

```

*** the Emp roster it was not possible to automatically compute every ***
*** End-Gap using the same algorithm. ***
*** ***
*=====

```

```

Numeric  GAP1_WK  GAP2_WK  GAP3_WK  GAP4_WK  GAP5_WK  GAP6_WK
         GAP7_WK  GAP8_WK  GAP9_WK

```

```

DO IF (HIT_1 = 1) /** if at least 1 END-GAP was found by CASES

```

```

**** 1 ename in 1st column only
DO IF NVAL(R1819900) = 1 & NVAL(R1820000 to R1820700) = 0
COMPUTE EMPLIST = 1
COMPUTE GAP_ST1 = END1.W + 1
COMPUTE GAP_END1 = DOI.W
COMPUTE GAP1_WK = RND(GAP_END1 - GAP_ST1) + 1
END IF

```

```

**** 1 Ename in column 2 only
DO IF NVAL(R1820000)= 1 & NVAL(R1819900, R1820100 to R1820700) = 0
COMPUTE EMPLIST = 2
COMPUTE GAP_ST2 = END2.W + 1
COMPUTE GAP_END2 = DOI.W
COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
END IF

```

```

**** 1 Ename in column 3 only
DO IF NVAL(R1820100)= 1 & NVAL(R1819900, R1820000, R1820200 to R1820700) = 0
COMPUTE EMPLIST = 3
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = DOI.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
END IF

```

```

**** 1 Ename in column 4 only
DO IF NVAL(R1820200)= 1 & NVAL(R1819900 to R1820100, R1820300 to R1820700) = 0
COMPUTE EMPLIST = 4
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = DOI.W
COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 1
END IF

```

```

**** 1 Ename in column 5 only
DO IF NVALID(R1820300)= 1 & NVALID(R1819900 to R1820200, R1820400 to R1820700) = 0
COMPUTE EMPLIST = 5
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = DOI.W
COMPUTE GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

```

```

**** 2 Enames in columns 1,2 only ** emplist = 6
DO IF NVALID(R1819900, R1820000) = 2 AND NVALID(R1820100 to R1820700) = 0
COMPUTE EMPLIST = 6
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 = DOI.W
COMPUTE GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
END IF

```

```

***** 2 Enames: Columns 1 & 3 only
DO IF NVALID(R1819900, R1820100)= 2 & NVALID(R1820000, R1820200 to R1820700) = 0
COMPUTE EMPLIST = 7
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 = DOI.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
END IF

***** 2 Enames: Columns 1 & 4 only
DO IF NVALID(R1819900, R1820200)= 2 & NVAL(R1820000, R1820100,
                                           R1820300 to R1820700) = 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(R1819900, R1820300)= 2 & NVAL(R1820000 to R1820200,
                                           R1820400 to R1820700) = 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(R1819900, R1820400)= 2 & NVAL(R1820000 to R1820300,
                                           R1820500 to R1820700) = 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(R1819900, R1820500)= 2 & NVAL(R1820000 to R1820400,
                                           R1820600, R1820700) = 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(R1820000, R1820100)= 2 & NVALID(R1819900, R1820200 to R1820700) = 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

```

```
DO IF NVALID(R1820000, R1820200)= 2 & NVALID(R1819900, R1820100,
                                                R1820300 to R1820700) = 0
```

```
COMPUTE EMLIST = 13
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
```

```
**** 2 Enames in columns 3 & 4 only
```

```
DO IF NVALID(R1820100, R1820200)= 2 & NVAL(R1819900, R1820000,
                                                R1820300 to R1820700) = 0
```

```
COMPUTE EMLIST = 14
COMPUTE GAP_ST3 = END3.W + 1
COMPUTE GAP_END3 = ST4.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 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 in columns 4 & 5 only
```

```
DO IF NVALID(R1820200, R1820300)= 2 & NVALID(R1819900 to R1820100,
                                                R1820400 to R1820700) = 0
```

```
COMPUTE EMLIST = 15
COMPUTE GAP_ST4 = END4.W + 1
COMPUTE GAP_END4 = ST5.W
COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 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 in columns 5 & 6 only
```

```
DO IF NVALID(R1820300, R1820400)= 2 & NVALID(R1819900 to R1820200,
                                                R1820500 to R1820700) = 0
```

```
COMPUTE EMLIST = 16
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 = DOI.W
COMPUTE GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
END IF
```

```
***** 3 Enames in column 1, 2, 3 only
```

```
DO IF NVALID(R1819900 to R1820100)= 3 & NVALID(R1820200 to R1820700) = 0
```

```
COMPUTE EMLIST = 17
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 = DOI.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
END IF
```

```
***** 3 Enames in column 1, 2, 4 only
```

```
DO IF NVAL(R1819900,R1820000,R1820200)= 3 & NVAL(R1820100,R1820300 to R1820700)=0
COMPUTE EMLIST = 18
```



```

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_END3 = ST5.W
COMPUTE GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
*
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = DOI.W
COMPUTE GAP5WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

***** 3 Enames in column 1, 4, 5 only
DO IF NVAL(R1819900,R1820200,R1820300)= 3 & NVAL(R1820000, R1820100,
R1820400 to R1820700)=0
COMPUTE EMPLIST = 23
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 = ST5.W
COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 1
*
COMPUTE GAP_ST5 = END5.W + 1
COMPUTE GAP_END5 = DOI.W
COMPUTE GAP5WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

***** 3 Enames in column 1, 5, 6 only
DO IF NVAL(R1819900,R1820300,R1820400)= 3 & NVAL(R1820000 to R1820200,
R1820500 to R1820700)=0
COMPUTE EMPLIST = 24
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 = ST6.W
COMPUTE GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
*
COMPUTE GAP_ST6 = END6.W + 1
COMPUTE GAP_END6 = DOI.W
COMPUTE GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
END IF

***** 3 Enames in column 4, 5, 6 only
DO IF NVAL(R1820200,R1820300,R1820400)= 3 & NVAL(R1819900 to R1820100,
R1820500 to R1820700)=0
COMPUTE EMPLIST = 25
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 = DOI.W
COMPUTE GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
END IF

***** 3 Enames in column 5, 6, 7 only
DO IF NVAL(R1820300,R1820400,R1820500)= 3 & NVAL(R1819900 to R1820200,
R1820600, R1820700)=0
COMPUTE EMPLIST = 26
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 = DOI.W
COMPUTE GAP7_WK = RND(GAP_END7 - GAP_ST7) + 1
END IF

```

```

***** 4 Enames in column 1, 2, 3, 4 only
DO IF NVALID(R1819900 to R1820200)= 4 & NVALID(R1820300 to R1820700) = 0
COMPUTE EMLIST = 27
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 = DOI.W
COMPUTE GAP4_WK = RND(GAP_END4 - GAP_ST4) + 1
END IF

```

```

***** 4 Enames in column 1, 2, 4, 5 only
DO IF NVAL(R1819900, R1820000, R1820200, R1820300)= 4
& NVAL(R1820100, R1820400 to R1820700)= 0
COMPUTE EMLIST = 28
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 = 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

```

```

***** 4 Enames in column 1, 3, 4, 5 only
DO IF NVAL(R1819900, R1820100 to R1820300)= 4 & NVAL(R1820000,
R1820400 to R1820700) = 0
COMPUTE EMLIST = 29
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 = 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

```

```

***** 4 Enames in column 2, 3, 4, 5 only
DO IF NVAL(R1820000 to R1820300)= 4 & NVAL(R1819900, R1820400 to R1820700) = 0

```

```

COMPUTE  EMPLIST = 30
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 = DOI.W
COMPUTE  GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

```

```

***** 4 Enames, in column 3, 4, 5, 6 only
DO IF  NVAL(R1820100 to R1820400)= 4 & NVAL(R1819900, R1820000,
                                           R1820500 to R1820700) = 0

```

```

COMPUTE  EMPLIST = 31
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 = DOI.W
COMPUTE  GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
END IF

```

```

***** 4 Enames, in column 1, 4, 5, 6 only
DO IF  NVAL(R1819900, R1820200 to R1820400)= 4 & NVAL(R1820000, R1820100,
                                           R1820500 to R1820700) = 0

```

```

COMPUTE  EMPLIST = 32
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 = 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 = DOI.W
COMPUTE  GAP6_WK = RND(GAP_END6 - GAP_ST6) + 1
END IF

```

```

***** 4 Enames, in column 2, 5, 6, & 7 only
DO IF  NVAL(R1820000, R1820300 to R1820500)= 4 & NVAL(R1819900, R1820100, R1820200,
                                           R1820600, R1820700) = 0

```

```

COMPUTE  EMPLIST = 33
COMPUTE  GAP_ST2 = END2.W + 1
COMPUTE  GAP_END2 = ST3.W
COMPUTE  GAP2_WK = RND(GAP_END2 - GAP_ST2) + 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 = DOI.W
COMPUTE GAP7_WK = RND(GAP_END7 - GAP_ST7) + 1
END IF

***** 4 Enames in column 1, 2, 8, 9 only
DO IF NVAL(R1819900, R1820000, R1820600, R1820700)= 4
& NVAL(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 = DOI.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 = DOI.W
COMPUTE GAP5_WK = RND(GAP_END5 - GAP_ST5) + 1
END IF

***** 5 Enames in column 1, 2, 3, 7, 8 only
DO IF NVAL(R1819900, R1820000, R1820100, R1820500, R1820600)= 5
& NVAL(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  GAP3_WK =  RND(GAP_END3 - GAP_ST3)  + 1
*_-
COMPUTE  GAP_ST7 =  END7.W  + 1
COMPUTE  GAP_END7 =  ST8.W
COMPUTE  GAP7_WK =  RND(GAP_END7 - GAP_ST7)  + 1
*_-
COMPUTE  GAP_ST8 =  END8.W  + 1
COMPUTE  GAP_END8 =  DOI.W
COMPUTE  GAP8_WK =  RND(GAP_END8 - GAP_ST8)  + 1
END IF

```

```

***** 5 Enames in column 2, 3, 4, 5, & 6 only
DO IF NVALID(R1820000 to R1820400)= 5 & NVAL(R1819900, R1820500, R1820700) = 0
COMPUTE  EMPLIST = 37
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 =  DOI.W
COMPUTE  GAP6_WK =  RND(GAP_END6 - GAP_ST6)  + 1
END IF

```

```

***** 5 Enames in column 3, 4, 5, 6, & 7 only
DO IF NVALID(R1820100 to R1820500)= 5 & NVAL(R1819900, R1820000,
R1820600, R1820700) = 0
COMPUTE  EMPLIST = 38
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 =  DOI.W
COMPUTE  GAP7_WK =  RND(GAP_END7 - GAP_ST7)  + 1
END IF

```

```

***** 6 Enames in columns 1, 2, 3, 4, 5, & 6 only
DO IF NVAL(R1819900 to R1820400) = 6 & NVAL(R1820500 to R1820700) = 0
COMPUTE  EMPLIST = 39
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 = DOI.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 = DOI.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)

```

```

COMPUTE  GAP_ST3 = END3.W + 1
COMPUTE  GAP_END3 = ST2.W
COMPUTE  GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
END IF

```

YW respondent

```

DO IF (SERIAL = 2834)
COMPUTE  GAP_ST3 = END3.W + 1
COMPUTE  GAP_END3 = ST2.W
COMPUTE  GAP3_WK = RND(GAP_END3 - GAP_ST3) + 1
COMPUTE  GAP_ST2 = END2.W + 1
COMPUTE  GAP_END2 = DOI.W
COMPUTE  GAP2_WK = RND(GAP_END2 - GAP_ST2) + 1
END IF

```

```

**=====
**      If GAP-END < GAP-START then set to 0
DO REPEAT
    ST = GAP_ST1      GAP_ST2      GAP_ST3      GAP_ST4      GAP_ST5
      GAP_ST6      GAP_ST7      GAP_ST8
  / END = GAP_END1    GAP_END2    GAP_END3    GAP_END4    GAP_END5
    GAP_END6    GAP_END7    GAP_END8
  / GAPWEEK = GAP1_WK    GAP2_WK    GAP3_WK    GAP4_WK    GAP5_WK
      GAP6_WK      GAP7_WK      GAP8_WK

```

```

IF rnd(END) - rnd(ST) < 0  GAPWEEK = 0
END REPEAT

```

```

*=====
*****      Blank GAP-3.1 for SERIAL=12768
DO IF SERIAL = 12768
RECODE R390401 (LO thru HI=SYSMIS)
END IF

```

```

*=====
***      Create End-gap length and UMPA and OLFA      ***
***      ***
**      1. Reconstruct total length of each End-gap, the gap following      ***
**      each Stop-date. The gap-length variables are called GAP1_N1,      ***
**      GAP1_N2....GAP1_N9      ***
**      ***
**      This variable reconstructs the CASES variable called GAP#n, which is      ***
**      the length (in weeks) of each gap between the End-Date and Start of      ***
**      the next job, or the End-Dt and the DOI. GAP-#n was computed by      ***
**      CASES prior to the GAP-1 sequence of questions. This variable was      ***
**      not written out by CASES, so it was not on the data tape CHRR received      ***
**      from Census. However, for most respondents it was possible to      ***
**      re-construct the value of the length of each End-gap. This      ***
**      "total length" variable was created by summing 2 variables that      ***
**      DID come in from Census: GAP-2 and INTER-CK.      ***
**      ***
**      TOTAL END-GAP = GAP-2 + INTER-CK      ***
**      ***
**      2. Compute complete and accurate measures of wks UMP and wks OLF      ***
**      for each gap following the end of a job. (The suffix of "A"      ***
**      is used because this section deals only with the End-Gap      ***
**      starting after the Stop-Date and ending with the Start-date      ***
**      of the next job. Questions GAP-1 to INTER-CK)      ***
**      ***
**      UMPA = GAP-2      ***
**      OLFA = INTER-CK = TOTAL GAP - GAP-2      ***
**      ***
**      Note that INTER-CK = TOTAL - GAP-2. Thus, if GAP-2 is DK (-2)      ***
**      then INTER-CK can't be computed. Therefore, the # weeks      ***
**      looking for work is unknown and the weeks OLF is unknown, so UMP= -3      ***
**      and OLF= -3.      ***
**      ***
**      ***

```



```
*=====
```

```
NUMERIC R2551600 RR2552600 RR2552800 RR2552900 INTER07
Numeric UMPA.1 UMPA.2 UMPA.3 UMPA.4 UMPA.5 UMPA.6 UMPA.7 UMPA.8
numeric OLFA.1 OLFA.2 OLFA.3 OLFA.4 OLFA.5 OLFA.6 OLFA.7 OLFA.8
```

```
DO REPEAT
```

```
    GAP_1 = R2551000 R2551100 R2551200 R2551300 R2551400 R2551500 R2551700
  / GAP_2 = RR2552200 RR2552300 RR2552400 RR2552500 RR2552600 RR2552700 RR2552900
  / 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 = amount in INTER-CK
```

```
  IF INTER < 0 GAP1_N = CHRRGAP /** INTER-CK = -2, so OLF = CHRRGAP
```

```
  IF SYSMIS(INTER) GAP1_N = CHRRGAP /* if INTER-CK is missng, gap_1 = CHRRGAP
```

```
  COMPUTE OLFA = GAP1_N /** entire gap-length is OLF
```

```
If sysmis(CHRRgap) & (gap1_n > 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 OLF=miss
```

```
IF (GAP_2 ge 0) & (INTER ge 0) GAP1_N = GAP_2 + INTER
```

```
IF (GAP_2 ge 0) & (SYSMIS(INTER)) GAP1_N = CHRRGAP
```

```

IF GAP_2 ge 0                                UMPA = GAP_2 /* R tells #weeks looking
IF (GAP_2 ge 0) & (INTER ge 0)                OLFA = INTER
IF (GAP_2 ge 0) & (SYSMIS(INTER))             OLFA = CHRRGAP - GAP_2
END IF
END REPEAT

MISS VAL   UMPA.1   UMPA.2   UMPA.3   UMPA.4   UMPA.5   UMPA.6   UMPA.8
           OLFA.1   OLFA.2   OLFA.3   OLFA.4   OLFA.5   OLFA.6   OLFA.8   (-3)

VAR LAB   GAP1_N1   'Length of END-GAP for Ename 1'

*****
**** Edit MW case 3381 *****
*****
** INTER-CK-01 was -61, but should be 47 wks
DO IF SERIAL = 3381
COMPUTE OLFA.2 = GAP2_WK
END IF

**=====
***** DOLI to Start-DT gap (GAP-1B) *****
**=====
*** The DOLI_GAP is the number of weeks between the DOLI and the ***
*** Start date of the job that was closest to, but not equal to, the ***
*** Date of Last Interview. Respondents with at least 1 job ***
*** that began on the DOLI would have no COLI-Gap, and so would ***
*** have skipped the questions pertaining to the DOLI-Gap (Q's 1B, 2B, ***
*** INTER-CHK, and 3B). ***
*** ***
*** When Gap-1B = 2 or 3 ("some" or "none") the DOLI-GAP length ***
*** could be computed using the data in questions 2B and INTER-CHK ***
*** (R2557400 and R2557500). DOLI_CHR = R2557400 + R2557500 ***
*** ***
*** However, when GAP-1B = 1 ("All") there was no data in INTER-CHK, ***
*** so the length of the total gap could not be computed using the ***
*** previous method. Instead the DOLI and the Start-Dates for ***
*** every employer listed were used to determine the number of weeks ***
*** between the DOLI and the earliest Start date that came after ***
*** the DOLI. ***
*** ***
*** When Gap-1B = 1: ***
*** DOLI-GAP = min(Start1 - DOLI; Start2 - DOLI,...Start9 - DOLI) ***
*** ***
***=====

DO REPEAT
  START = START1 START2 START3 START4 START5 START6
          START7 START8 START9 DOI
  / END = END1 END2 END3 END4 END5 END6
          END7 END8 END9 DOLI
  / DOLI_WK = DOLI.W1 to DOLI.W9 DOLI.W10

*** If Start is missing then that particular DOLI-Gap = -3
COMPUTE DOLI_WK = RND(CTIME.DAYS(START - DOLI) / 7)
IF NVAL(END) = 1 & SYSMIS(START) DOLI_WK = -3
END REPEAT

** If the DOLI-Start Gap for any Employer is 0 (zero), then
** it doesn't matter if the DOLI-Start gap for any other
** Employer is undefined (= -3). But if there is no 0 and
** at least one of the DOLI-Gaps is -3 then the length of
** the DOLI-GAP can't be determined.
*****
COUNT DOLIGAP0 = DOLI.W1, DOLI.W2, DOLI.W3, DOLI.W4, DOLI.W5,
                  DOLI.W6, DOLI.W7, DOLI.W8, DOLI.W9, DOLI.W10 (0)

```

```

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, DOLI.W10) = -3      DOLI_CHR = -3

IF MIN(DOLI.W1, DOLI.W2, DOLI.W3, DOLI.W4, DOLI.W5, DOLI.W6, DOLI.W7,
      DOLI.W8, DOLI.W9, DOLI.W10) > -1
      DOLI_CHR = MIN(DOLI.W1, DOLI.W2, DOLI.W3, DOLI.W4, DOLI.W5, DOLI.W6,
                    DOLI.W7, DOLI.W8, DOLI.W9, DOLI.W10)
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 INTER-CHK 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 (LO 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)

```

```

COMPUTE  OLFB = DOLI_CEN
END IF

```

```

YW
DO IF (SERIAL = 4217)
COMPUTE  OLFB = DOLI_CEN
END IF

```

```

**=====
**      Create and Initialize the WEEKS array variables      ***
**=====
**
**  There are total of 812 variables in the WEEKS array, called WEEK1, WEEK2,
**  WEEK3 up to WEEK812.  WEEK1 is the week containing the DOLI, WEEK2
**  is 1 week (7 days) after the DOLI, and so on.  The WEEKS variables
**  between the DOLI and the DOI are initialized with a value of 0,
**  while the WEEKS VARIABLES that come after the DOI are blank.  Thus,
**  for example, if a respondent's DOI is 91 weeks
**  after her DOLI (and so WBID.W=91) then WEEK1 thru WEEK91 = 0,
**  while WEEK92 thru WEEK812 are blank.
**
**  EXPLANATION OF COMMANDS:
**      The VECTOR command creates 812 variables, naming them
**      WEEK1 up through WEEK812.  They are initialized as system missing.
**
**      LOOPI#  executes the commands within the LOOP structure 812 times,
**      once for each WEEK variable.

```

```

DO IF (DOLIMISS = 0)          /** only if WBID can be computed
VECTOR  WEEK(812)             /** maximum number of WBID.W among 1995 respondents
LOOP I# = 1 to WBID.RND       /**LOOP I# = 1 to the number of wks btwn Ivw
COMPUTE WEEK(I#) = 0
END LOOP
END IF
EXECUTE

```

```

COMPUTE NUM_WEEK = NVALID(WEEK1 to WEEK812)
VAR LAB  NUM_WEEK  'Number of weeks vars w/ valid value'

```

```

**=====
**      FILL IN WEEKS-ARRAY WITH Work history info      ***
**=====
**
**      WEEKn = 1      WORK:  employed that week          ***
**      WEEKn = 2      WUMP:  looking for work, or laid off ***
**      WEEKn = 3      WOLF:  not looking for work or not laid off ***
**      WEEKn = .      week falls outside of gap between DOLI & DOI ***
**=====
VECTOR WEEKS = WEEK1 to WEEK812      /* assigns 1=Working to each WEEK variable

```

```

***** fill in 1 for weeks working  WORK
DO IF (DOLIMISS = 0)
DO REPEAT
    st.W = ST1.W ST2.W ST3.W ST4.W ST5.W ST6.W ST7.W
           ST8.W ST9.W
    / end.W = END1.W END2.W END3.W END4.W END5.W END6.W
           END7.W END8.W END9.W
    / WORK = WORK1 WORK2 WORK3 WORK4 WORK5 WORK6 WORK7 WORK8 WORK9

DO IF WORK ge 1          /** only if the # weeks working could be computed
LOOP I# = RND(ST.W) to RND(ST.W + (WORK - 1))
COMPUTE WEEKS(I#) = 1
END LOOP
END IF
END REPEAT
END IF
EXECUTE

```

```

COUNT WORKA = WEEK1 to WEEK812 (1)
IF ABS(WORKA - WBID.RND) LE 1  ALLWRK2 = 1

```

```
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)'
```

```

*=====
**
**          CREATE  THE FINAL WEEKS VARIABLES !!!!
**
**
*=====

```

```

*****      UNIVERSE = 0    DOLI is blank
DO IF SYSMIS(DOLI)
  COMPUTE  WUMP = -3      /* without the DOLI these weeks vars can't be created
  COMPUTE  WOLF = -3
  COMPUTE  WORK = -3
  IF (HIT_1 = 0 & HIT_1B = 0) & (NUM_JOB = 0)      WORK = 0
  IF NVALID(R1819900 to R1820700) = 0      WORK = 0      /* no Emps listed, so no work
  DO IF (HIT_1 = 0 & HIT_1B = 1)
  IF R2557300 = 1      WOLF = 0      /** R looked during all weeks in gap, so OLF= 0
  IF R2557300 = 3      WUMP = 0      /** R looked for 0 weeks in gap, so #wks UMP = 0
  END IF
  END IF

```

```

*****      UNIVERSE = 1, 1.5 -- Worked the whole time from DOLI to DOI

```

```

DO IF (UNIVERSE = 1)
  COMPUTE  WORK = WBID.RND
  COMPUTE  WUMP = 0
  COMPUTE  WOLF = 0

```

```

****  fix YW case 5031:  has no start or stop dates; R2557300 = blank
DO IF SERIAL = 5031
  COMPUTE  WUMP = -3
  COMPUTE  WOLF = -3
  COMPUTE  WORK = -3
  END IF

```

```

***  fix YW 0367:  R2557300 = blank, but DOLI-GAP = 26
DO IF SERIAL = 0367
  COMPUTE  WUMP = -3
  COMPUTE  WOLF = -3
  COMPUTE  WORK = WBID.RND - DOLI_CHR
  END IF
  END IF

```

```

*****      UNIVERSE = 2    No job activity since DOLI, Only a DOLI-Gap

DO IF UNIVERSE = 2
  COMPUTE  WORK = 0      /** must be 0 since there was no employr activity
  IF (UMPB ge 0)      WUMP = UMPB
  IF (OLFB ge 0)      WOLF = OLFB
  IF (UMPB < 0)      WUMP = -3
  IF (OLFB < 0)      WOLF = -3
  IF R2557300 = 3      WUMP = 0
  END IF

```

```

*****      UNIVERSE = 3    End gap only

```

```

DO IF UNIVERSE = 3
  DO IF (WORKA ge 0)      /* if separate weeks of work could be summed
  COUNT  WORK = WEEK1 to WEEK812 (1)
  COUNT  WUMP = WEEK1 to WEEK812 (2)
  COUNT  WOLF = WEEK1 to WEEK812 (3)
  END IF

  IF WORKA = -3      WORK = -3
  IF WORKA = -3      WUMP = -3
  IF WORKA = -3      WOLF = -3
  END IF

```

***** 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
```

```

*****
**** HAND Edit YW case 5031 *****
** Case 5031 had no job activity since the DOLI but also
** has no data in the gaps section, so WORK = 0,
** but WEMP and WUMP can't be defined.
DO IF SERIAL = 5031
COMPUTE WORK = 0
COMPUTE WUMP = -3
COMPUTE WOLF = -3
END IF

```

```

*****
**** HAND Edit MW case 1655 *****
** The DOI was set to 1980. Thus, the WBID can't be calculated.
** But she didn't work at all since DOLI, so WORK=0, but WUMP
** & WOLF = -3
DO IF SERIAL = 1655
COMPUTE WBID = -3
COMPUTE WORK = 0
COMPUTE WUMP = -3
COMPUTE WOLF = -3
END IF

```

```

*****
**** HAND Edit MW case 4156 *****
** Visual inspection of the case showed that the jobs
** were not sorted by date properly. The correct values
** for the weeks variables were assigned.
DO IF SERIAL = 4156
COMPUTE WORK = 6
COMPUTE WUMP = 22
COMPUTE WOLF = 62
END IF

```

```

*****
**** HAND Edit YW cases 3628 & 4041 *****
** Start and End-dates for 2 Enames are identical, yet the
** question Gap-1 was asked only once. This may be ok since
** the gap dates are the same so if the R was looking
** for work the whole time after Ename 1 then she was also
** looking the whole time for Ename 2 (since the dates are
** the same this single Gap-1 info covers the same
** period of time on the calendar). Thus, the gap lengths and
** WUMP and WOLF values were computed by hand.

```

```

DO IF ANY(SERIAL, 3628, 4041)
IF SERIAL = 4041 WORK = 67
IF SERIAL = 4041 WUMP = 19
IF SERIAL = 4041 WOLF = 0

IF SERIAL = 3628 WORK = 44
IF SERIAL = 3628 WUMP = 45
IF SERIAL = 3628 WOLF = 0
END IF

```

```

*****
**** HAND Edit MW cases *****
**** Duplicate job dates resulted in 1 End-gap being computed, which
**** threw off the SPSS code because columns for gap length and GAP-1
**** questions didn't line up. But the dates and gap lengths are ok.
IF SERIAL = 2126 WUMP = 25
IF SERIAL = 2126 WOLF = 0

IF SERIAL = 3911 WUMP = 59
IF SERIAL = 3911 WOLF = 0

```

Finish

NET FAMILY ASSETS AND FAMILY INCOME

1. 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.
2. 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.
3. 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).
4. 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.
5. 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.
6. 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.
7. 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.
8. Derivations are revised to reflect the current missing value treatment.
9. Only those respondents with outcomes (R34919.) of 201 through 205 (complete or partially complete interviews) are included in derivations.
10. 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.
11. 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.

NET FAMILY ASSETS [R34066.]

```
ASSET95=.; HOUSE95=.; PROPDE95=.; SAVE95=.; BOND95=.; STOCK95=.; LOAN95=.; IFARM95=.; IBUS95=.;  
IREAL95=.; ira95=.; insu95=.; trus95=.; DEBT95=.; ASEXCA95=.; C95=0; if R3381200=2 then  
R3381200=0;  
IF R3381200=0 & (R3381300<0 | R3381300=.) & (R3381400<0 | R3381400=.) THEN HOUSE95=0;  
IF R3381400>=0 & R3382000>=0 THEN PROPDE95= R3381400 + R3382000;  
IF R3381400=0 & R3382000>=0 THEN PROPDE95= R3382000;  
IF R3381400>=0 & R3382000=0 THEN PROPDE95= R3381400;  
IF R3381400<0 | R3382000<0 THEN PROPDE95=.;  
ELSE IF R3381300>=0 & PROPDE95>=0 THEN HOUSE95=R3381300-PROPDE95;  
IF R3386200=1 & R3381300<=R3386100 &  
R3381400<=R3386300 & R3386100^=. & R3386300^=. THEN HOUSE95=0;  
IF HOUSE95^=. THEN ASSET95=HOUSE95; ELSE C95=1;  
IF (R3382300=0 | R3396300=0) & (R3382400<0 | R3382400=.) THEN SAVE95=0;  
ELSE IF R3382400>=0 THEN SAVE95=R3382400;  
if R3382500=1 & R3382600=1 then save95=40000;  
if R3382500=1 & R3382600=0 then save95=25000;  
if R3382500=0 & R3382700=1 then save95=5500;  
if R3382500=0 & R3382700=0 then save95=1000;  
if R3382500=-1 | R3382600=-1 | R3382700=-1 |  
R3382500=-2 | R3382600=-2 | R3382700=-2 then save95=.;  
IF SAVE95^=. THEN ASSET95=ASSET95+SAVE95; ELSE C95=C95+1;  
IF (R3382800=0 | R3396300=0) & (R3382900<0 | R3382900=.) THEN BOND95=0;  
ELSE IF R3382900>=0 THEN BOND95=R3382900;  
if R3383000=1 & R3383100=1 then bond95=5000;  
if R3383000=1 & R3383100=0 then bond95=3000;  
if R3383000=0 & R3383200=1 then bond95=750;  
if R3383000=0 & R3383200=0 then bond95=500;  
if R3383000=-1 | R3383100=-1 | R3383200=-1 |  
R3383000=-2 | R3383100=-2 | R3383200=-2 then bond95=.;  
IF BOND95^=. THEN ASSET95=ASSET95+BOND95; ELSE C95=C95+1;  
IF (R3383300=0 | R3396300=0) & (R3383400<0 | R3383400=.) THEN STOCK95=0;  
ELSE IF R3383400>=0 THEN STOCK95=R3383400;  
if R3383500=1 & R3383600=1 then stock95=40000;  
if R3383500=1 & R3383600=0 then stock95=27500;  
if R3383500=0 & R3383700=1 then stock95=10000;  
if R3383500=0 & R3383700=0 then stock95=5000;  
if R3383500=-1 | R3383600=-1 | R3383700=-1 |  
R3383500=-2 | R3383600=-2 | R3383700=-2 then stock95=.;  
IF STOCK95^=. THEN ASSET95=ASSET95+STOCK95; ELSE C95=C95+1;  
if R3383800=0 & (R3383900<0 | R3383900=.) then ira95=0;  
else if R3383900 ge 0 then ira95=R3383900;  
if R3384000=1 & R3384100=1 then ira95=30000;  
if R3384000=1 & R3384100=0 then ira95=22500;  
if R3384000=0 & R3384200=1 then ira95=10000;  
if R3384000=0 & R3384200=0 then ira95=5000;  
if R3384000=-1 | R3384100=-1 | R3384200=-1 |  
R3384000=-2 | R3384100=-2 | R3384200=-2 then ira95=.;  
if ira95^=. then asset95=asset95+ira95; else c95=c95+1;  
IF R3384300=0 & (R3384400<0 | R3384400=.) THEN LOAN95=0;  
ELSE IF R3384400>=0 THEN LOAN95=R3384400;  
IF LOAN95^=. THEN ASSET95=ASSET95+LOAN95; ELSE C95=C95+1;  
if (R3384500=0 | R3384700=1) & (R3384800<0 | R3384800=.) then insu95=0;  
else if R3384800 ge 0 then insu95=R3384800;  
if insu95^=. then asset95=asset95+insu95; else c95=c95+1;  
if R3385100=0 & (R3385200<0 | R3385200=.) then trus95=0;  
else IF R3385200 >= 0 THEN TRus95 = R3385200;  
IF TRus95^=. THEN ASSET95=ASSET95+TRus95; ELSE C95=C95+1;  
IF R3385600=0 & (R3385700<0 | R3385700=.) then settle=0;  
else if R3385600=1 & R3385700<0 then settle=0;  
else if R3385700>=0 then settle=R3385700;  
if settle^=. then asset95=asset95+settle; else c95=c95+1;  
IF (R3386000=0 | R3395900=0) & (R3386100<0 | R3386100=.)  
& (R3386300<0 | R3386300=.) THEN IFARM95=0;  
ELSE IF R3386100>=0 & R3386300>=0 THEN IFARM95=R3386100-R3386300;  
IF IFARM95^=. THEN ASSET95=ASSET95+IFARM95; ELSE C95=C95+1;  
IF (R3386400=0 | (R3386600=0 & R3392400=0)) & (R3386500<0 | R3386500=.)  
& (R3386600<0 | R3386600=.) THEN IBUS95=0;  
ELSE IF R3386500>=0 & R3386600>=0 THEN IBUS95=R3386500-R3386600;  
IF IBUS95^=. THEN ASSET95=ASSET95+IBUS95; ELSE C95=C95+1;
```

```
IF (R3386700=0 | R3396100=0) & (R3386800<0 | R3386800=.)  
& (R3386900<0 | R3386900=.) & (R3387000<0 | R3387000=.) THEN IREAL95=0;  
ELSE IF R3386800>=0 & R3386900>=0 & R3387000>=0  
THEN IREAL95=R3386800-R3386900-R3387000;  
IF IREAL95^=. THEN ASSET95=ASSET95+IREAL95; ELSE C95=C95+1;  
IF R3387100=0 & (R3387200<0 | R3387200=.) THEN DEBT95=0;  
ELSE IF R3387200>=0 THEN DEBT95=R3387200;  
IF DEBT95^=. THEN ASSET95=ASSET95-DEBT95; ELSE C95=C95+1;  
IF C95>0 & ASSET95>-99999 & ASSET95<999999 THEN ASEXCA95=.;  
IF C95=0 THEN DO; IF asset95 ^=. & ASSET95<=-99999 THEN ASEXCA95=-99999;  
IF ASSET95>=999999 THEN ASEXCA95=999999;  
IF ASSET95>-99999 & ASSET95<999999 THEN ASEXCA95=ASSET95;  
END;
```

FAMILY INCOME [R34067.]

```
HUSBAND=0; RWAGE=.; RBUS=.; RUNEM=.; RSUB=.; RSOC=.; RVET=.; RCOM=.;
RSSD=.; RDIS=.; REPRI=.; REMIL=.; REFED=.; RESTE=.; REUNI=.; REIRA=.;
REOTH=.; SWAGE=.; SBUS=.; SUNEM=.; SSUB=.; SSOC=.; SVET=.; SCOM=.;
SSSD=.; SDIS=.; SREPRI=.; SREMIL=.; SREFED=.; SRESTE=.; SREUNI=.;
SREIRA=.; SREOTH=.; FARM=.; RENT=.; INTT=.; FOODS=.; AFDC=.; SSI=.;
ALI=.; CHILD=.; OTHER=.; FAM=.; FAMINC=.;
IF R3392100=1 THEN HUSBAND=1;
if f00200=2 & (R3388500<0 | R3388500=.) then rwage=0;
IF R3388500>=0 THEN RWAGE=R3388500;
if (R3388600=0 | R3388600=4)
& (R3388700=-1 | R3388700=-2 | R3388700=.) then rbus=0;
IF R3388600=1 & R3388700>=0 then rbus=R3388700;
if R3388600=3 & R3388700>=0 then rbus=0-R3388700;
if R3388800=0 & (R3388900<0 | R3388900=.) & (R3389000<0 | R3389000=.) then runem=0;
IF R3388900>=0 & R3389000>=0 THEN RUNEM=R3388900*R3389000;
if (runem=0 | R3389100=0) & (R3389200<0 | R3389200=.)
& (R3389300<0 | R3389300=.) then rsub=0;
IF R3389200>=0 & R3389300>=0 THEN RSUB=R3389200*R3389300;
if R3389400=0 & (R3389500<0 | R3389500=.) & (R3389600<0 | R3389600=.) then rsoc=0;
IF R3389500>=0 & R3389600>=0 THEN RSOC=R3389500*R3389600;
if R3389800=0 & (R3389900<0 | R3389900=.) then rvet=0;
IF R3389900>=0 THEN RVET=R3389900;
if R3390000=0 & (R3390100<0 | R3390100=.) then rcom=0;
IF R3390100>=0 THEN RCOM=R3390100;
if R3390200=0 & (R3390300<0 | R3390300=.) then rssid=0;
IF R3390300>=0 THEN RSSD=R3390300;
if R3390400=0 & (R3390500<0 | R3390500=.) then rdis=0;
IF R3390500>=0 THEN RDIS=R3390500;
if R3390600=0 then do; repri=0; remil=0; refed=0; reste=0; reuni=0;
reira=0; reoth=0; end;
if R3390700=0 & (R3390800<0 | R3390800=.) then repri=0;
IF R3390800>=0 THEN REPRI=R3390800;
if R3390900=0 & (R3391000<0 | R3391000=.) then remil=0;
IF R3391000>=0 THEN REMIL=R3391000;
if R3391100=0 & (R3391200<0 | R3391200=.) then refed=0;
IF R3391200>=0 THEN REFED=R3391200;
if R3391300=0 & (R3391400<0 | R3391400=.) then reste=0;
IF R3391400>=0 THEN RESTE=R3391400;
if R3391500=0 & (R3391600<0 | R3391600=.) then reuni=0;
IF R3391600>=0 THEN REUNI=R3391600;
if R3391700=0 & (R3391800<0 | R3391800=.) then reira=0;
IF R3391800>=0 THEN REIRA=R3391800;
if R3391900=0 & (R3392000<0 | R3392000=.) then reoth=0;
IF R3392000>=0 THEN REOTH=R3392000;
if (husband^=1 | R3392200=0) & (R3392300<0 | R3392300=.) then swage=0;
IF R3392300>=0 THEN SWAGE=R3392300;
if (husband^=1 | R3392400=0 | R3392400=4)
& (R3392500<0 | R3392500=.) then sbus=0;
if R3392400=1 & R3392500>=0 then sbus=R3392500;
if R3392400=3 & R3392500>=0 then sbus=0-R3392500;
if (husband^=1 | R3392600=0) & (R3392700<0 | R3392700=.)
& (R3392800<0 | R3392800=.) then sunem=0;
IF R3392700>=0 & R3392800>=0 THEN SUNEM=R3392700*R3392800;
if (husband^=1 | R3392900=0 | sunem=0) & (R3393000<0 | R3393000=.)
& (R3393100<0 | R3393100=.) then ssb=0;
IF R3393000>=0 & R3393100>=0 THEN SSUB=R3393000*R3393100;
if (husband^=1 | R3393200=0) & (R3393300<0 | R3393300=.)
& (R3393400<0 | R3393400=.) then ssoc=0;
IF R3393300>=0 & R3393400>=0 THEN SSOC=R3393300*R3393400;
if (husband^=1 | R3393600=0) & (R3393700<0 | R3393700=.) then svet=0;
IF R3393700>=0 THEN SVET=R3393700;
if (husband^=1 | R3393800=0) & (R3393900<0 | R3393900=.) then scom=0;
IF R3393900>=0 THEN SCOM=R3393900;
if (husband^=1 | R3394000=0) & (R3394100<0 | R3394100=.) then sssd=0;
IF R3394100>=0 THEN SSSD=R3394100;
if (husband^=1 | R3394200=0) & (R3394300<0 | R3394300=.) then sdis=0;
IF R3394300>=0 THEN SDIS=R3394300;
if (husband^=1 | R3394400=0) then do; srepri=0; sremil=0; srefed=0;
sreste=0; sreuni=0; sreira=0; sreoth=0; end;
```

```

if R3394500=0 & (R3394600<0 | R3394600=.) then srepri=0;
IF R3394600>=0 THEN SREPRI=R3394600;
if R3394700=0 & (R3394800<0 | R3394800=.) then sremil=0;
IF R3394800>=0 THEN SREMIL=R3394800;
if R3394900=0 & (R3395000<0 | R3395000=.) then srefed=0;
IF R3395000>=0 THEN SREFED=R3395000;
if R3395100=0 & (R3395200<0 | R3395200=.) then sreste=0;
IF R3395200>=0 THEN SRESTE=R3395200;
if R3395300=0 & (R3395400<0 | R3395400=.) then sreuni=0;
IF R3395400>=0 THEN SREUNI=R3395400;
if R3395500=0 & (R3395600<0 | R3395600=.) then sreira=0;
IF R3395600>=0 THEN SREIRA=R3395600;
if R3395700=0 & (R3395800<0 | R3395800=.) then sreoth=0;
IF R3395800>=0 THEN SREOTH=R3395800;
if (R3395900=0 | R3395900=4) & (R3396000=. | R3396000=-1 | R3396000=-2)
then FARM=0;
if R3395900=1 & R3396000>=0 then farm=R3396000;
if R3395900=3 & R3396000>=0 then farm=0-R3396000;
if (R3396100=0 | R3396200=4) & (R3396200=. | R3396200=-1 | R3396200=-2)
then rent=0;
if R3396100=1 & R3396200>=0 THEN RENT=R3396200;
if R3396100=3 & R3396200>=0 then rent=0-R3396200;
if R3396300=0 & (R3396400<0 | R3396400=.) then intt=0;
IF R3396400>=0 THEN INTT=R3396400;
if R3396500=0 & (R3396600<0 | R3396600=.) & (R3396700<0 | R3396700=.) then foods=0;
IF R3396600>=0 & R3396700>=0 THEN FOODS=R3396600*R3396700;
if R3396800=0 & (R3396900<0 | R3396900=.) & (R3397000<0 | R3397000=.) then afdc=0;
IF R3396900>=0 & R3397000>=0 THEN AFDC=R3396900*R3397000;
if R3397100=0 & (R3397200<0 | R3397200=.) & (R3397300<0 | R3397300=.) then ssi=0;
IF R3397200>=0 & R3397300>=0 THEN SSI=R3397200*R3397300;
if (R3397400=0 & R3397600=0) & (R3397500<0 | R3397500=.) & (R3397700<0 | R3397700=.)
then ali=0; IF R3397500>=0 & R3397700>=0 THEN ALI=R3397500-R3397700;
if R3397500>=0 & R3397700=. then ali=R3397500;
if R3397500=. & R3397700>=0 then ali=R3397700;
if R3397900=0 & (R3398000<0 | R3398000=.) then chdsup=0;
else if R3398000>=0 then chdsup=R3398000;
else if R3398700=2 & R3401600>=0 then chdsup=R3401600;
else if R3398700=1 & R3402400>=0 then chdsup=R3402400;
else if R3398700=3 & R3401600>=0 & R3402400>=0 then chdsup=R3401600+R3402400;
if R3402900=0 & ((R3403200<0 | R3403200=.) & (R3403100<0 | R3403100=.) &
(R3403300<0 | R3403300=.) then rchd=0;
else if R3403200>=0 then rchd=52*R3403200;
else if R3403100>=0 then rchd=12*R3403100;
else if R3403300>=0 then rchd=R3403300;
if (husband^=1 | R3403500=0) & ((R3403700<0 | R3403700=.) & (R3403600<0 | R3403600=.) &
(R3403800<0 | R3403800=.) then hpchd=0;
else if R3403700>=0 then hpchd=52*R3403700;
else if R3403600>=0 then hpchd=12*R3403600;
else if R3403800>=0 then hpchd=R3403800;
if chdsup>=0 & rchd>=0 & hpchd>=0 then child=chdsup-rchd-hpchd;
if R3404000=0 & (R3404100<0 | R3404100=.) then other=0;
IF R3404100>=0 THEN OTHER=R3404100;
if R3404300=2 & (R3404400<0 | R3404400=.) then fam=0; else IF R3404300=1 THEN DO;
IF R3404400=1 THEN FAM= 2000; ELSE IF R3404400=2 THEN FAM= 5000;
ELSE IF R3404400=3 THEN FAM= 6750; ELSE IF R3404400=4 THEN FAM= 8250;
ELSE IF R3404400=5 THEN FAM= 12500; ELSE IF R3404400=6 THEN FAM= 16250;
ELSE IF R3404400=7 THEN FAM= 18750; ELSE IF R3404400=8 THEN FAM= 22500;
ELSE IF R3404400=9 THEN FAM= 30000; ELSE IF R3404400=10 THEN FAM= 42500;
ELSE IF R3404400=11 THEN FAM= 62500; ELSE IF R3404400=12 THEN FAM= 87500;
ELSE IF R3404400=13 THEN FAM=100000;
ELSE IF (R3404400=0 | R3404400=14) then fam=0; END;
IF RWAGE^=. & RBUS^=. & RUNEM^=. & RSUB^=. & RSOC^=. & RVET^=. & RCOM^=.
& RSSD^=. & RDIS^=. & REPRI^=. & REMIL^=. & REFED^=. & RESTE^=. & REUNI^=.
& REIRA^=. & REOTH^=. & SWAGE^=. & SBUS^=. & SUNEM^=. & SSUB^=. & SSOC^=.
& SVET^=. & SCOM^=. & SSSD^=. & SDIS^=. & SREPRI^=. & SREMIL^=. & SREFED^=.
& SRESTE^=. & SREUNI^=. & SREIRA^=. & SREOTH^=. & FARM^=. & RENT^=. &
INTT^=. & FOODS^=. & AFDC^=. & SSI^=. & ALI^=. & CHILD^=. & OTHER^=. & FAM^=.
THEN FAMINC=RWAGE+RBUS+RUNEM+RSUB+RSOC+RVET+RCOM+RSSD+RDIS+REPRI
+REMIL+REFED+RESTE+REUNI+REIRA+REOTH+SWAGE+SBUS+SUNEM+SSUB+SSOC+SVET
+SCOM+SSSD+SDIS+SREPRI+SREMIL+SREFED+SRESTE+SREUNI+SREIRA+SREOTH+FARM
+RENT+INTT+FOODS+AFDC+SSI+ALI+CHILD+OTHER+FAM;

```

```
IF FAMINC^=. & FAMINC<=-99999 THEN FAMINC=-99999;  
IF FAMINC>=999999 THEN FAMINC=999999;
```

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 RSOC ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=RSOC+SUMMATN; ELSE SUMMATN=RSOC; 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 SBUS ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=SBUS+SUMMATN; ELSE SUMMATN=SBUS; END;
  IF SUNEM ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=SUNEM+SUMMATN; ELSE SUMMATN=SUNEM; END;
  IF SSUB ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=SSUB+SUMMATN; ELSE SUMMATN=SSUB; END;
  IF SSOC ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=SSOC+SUMMATN; ELSE SUMMATN=SSOC; 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 FARM ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=FARM+SUMMATN; ELSE SUMMATN=FARM; 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 FOODS ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=FOODS+SUMMATN; ELSE SUMMATN=FOODS; END;
  IF AFDC ^= . THEN DO;
    IF SUMMATN^= . THEN SUMMATN=AFDC+SUMMATN; ELSE SUMMATN=AFDC; END;
  IF SSI^= . THEN DO; IF SUMMATN^= . THEN SUMMATN=SSI+SUMMATN; ELSE SUMMATN=SSI; END;
  IF ALI^= . THEN DO; IF SUMMATN^= . THEN SUMMATN=ALI+SUMMATN; ELSE SUMMATN=ALI; 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>=1 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^= . & SUMMATN<-99999 THEN SUMMATN=-99999;
  IF SUMMATN>999999 THEN SUMMATN=999999;
```

PAYRATE [R19306.-R19314.] AND HOURLY RATE OF PAY [R19954.-R19962.]

```
array p      p1-p9;
array hrop   R1928200-R1929000;
array ehrop  R1929400- R1929600 ehrop4 R1929800 ehrop6 R1930000 ehrop8-ehrop9;
array wrop   R1975000- R1975700 wrop9;
array ewrop  R1976200-R1976700 ewrop7-ewrop9;
array mrop   R1979800-R1980300 mrop7-mrop9;
array emrop  R1981000 R1981100 emrop3-emrop9;
array arop   R1984600- R1985200 arop8 arop9;
array earop  R1985800- R1985900 earop3-earop9;
array brop   R1989400-R1990000 brop8 brop9;
array ebrop  R1990600-R1990700 ebrop3-ebrop9;
do over p;
  if p=. & hrop ge -1 then p=hrop;
  if hrop=-2 & ehrop ge -2 then p=ehrop;
  if p=. & wrop ge -1 then p=wrop;
  if wrop=-2 & ewrop ge -2 then p=ewrop;
  if p=. & mrop ge -1 then p=mrop;
  if mrop=-2 & emrop ge -2 then p=emrop;
  if p=. & arop ge -1 then p=arop;
  if arop=-2 & earop ge -2 then p=earop;
  if p=. & brop ge -1 then p=brop;
  if brop=-2 & ebrop ge -2 then p=ebrop;      end;
array swhrp  swhrp1-swhrp9;
array cow    R1816300-R1817100;
array ushw   R1868100-R1868900;
array turp   R1924600-R1925400;
array rop    p1-p9;
array uswy   R1994200-R1994800 uswy8-uswy9;
do over SWHRP;
  IF COW GE 1 & rop ge 0 THEN DO;
  if turp=1 then swhrp=rop; else
  if (2 le turp le 3 & ushw ge 1) then swhrp=rop/ushw; else
  if (turp=4 & ushw ge 1) then swhrp=rop/(ushw*2); else
  if (turp=5 & ushw ge 1) then swhrp=rop/(ushw*4.33); else
  if (turp=6 & ushw ge 1 & uswy ge 1) then swhrp=rop/(ushw*uswy);
  swHRP = FLOOR(swHRP + .5); IF swHRP < 1 | swHRP >99999 THEN swHRP = .;      END; END;
```


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_ATTENDING, 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-COMPLETED: 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'

*****
```

```

**          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
COMPUTE HGC_95 = CURR_ATT - 1
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 recoded 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'
VAL LAB CURR_ATT DONE_ATT
-1 'Refused' -2 'Dont know' 9 '9th grade' 10 '10th grade'
11 '11th grade' 12 '12th grade' 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/Diploma program'

DO IF (DONE_ATT>-1 & DONE_ATT<19) /* Finished attending & not missing
COMPUTE HGC_95 = DONE_ATT
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 'Ref' -2 'Dont know' 0 'none'
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 = R3476600)

```

Part B. SPSS code for Young Women

```

** Use this codeblock if using the YW data to create the
** previous measure of HGC

Numeric HGC_OLD
Do if WEIGHT > 0 /* selects the YW with 1995 interviews
IF SYSMIS(HGC_OLD) HGC_OLD = R1520410 /*HGC93_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R1346410 /*HGC91_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R1215110 /*HGC88_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R1097410 /*HGC87_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R1051610 /*HGC85_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R0929510 /*HGC83_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R0797110 /*HGC82_YW
IF SYSMIS(HGC_OLD) HGC_OLD = R0749910 /*HGC80_YW
End if

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-COMPLETED: 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'

**=====
**      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
COMPUTE HGC_95 = CURR_ATT - 1
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 recoded 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
COMPUTE HGC_95 = DONE_ATT
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 'Ref' -2 'Dont know' 0 'none'
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 = R3476600)

```