APPENDIX 41: THE MATURE AND YOUNG WOMEN 1999

This appendix to the National Longitudinal Surveys of Mature Women and Young Women provides information specific to the 1999 surveys of these cohorts. It briefly describes aspects of the data that are new for 1999, and provides the SAS code used to edit or create variables for this release.

In 1999, 2,467 Mature Women respondents and 2,900 Young Women respondents completed interviews. Most surveys were conducted by personal interview, although respondents can request a phone interview and some chose this option (38.2%). The Mature Women and Young Women data sets each include over 11,000 variables that are new for 1999. For each cohort, about 2,500 of these are part a new section for 1999, Intra-Family Transfers. Another 350 variables are either new created variables that span multiple interviews, or variables that provide more detail about the process of determining the non-interview status. These latter variables apply not only to respondents with completed interviews in 1999, but also to respondents who may not have completed interviews for 1999. These sections are described in greater detail below.

There are also some changes to documentation for 1999. First, the Census Bureau has always assigned topcodes to asset, income, and expenditure amounts to protect the anonymity of individual respondents. Information about these topcodes is now included on the documentation records. Second, documentation for "mark all that apply" questions is different in 1999. Each possible answer category, in a separate record, receives a mark if the respondent selected that answer—i.e., '1' is a mark indicating a positive response for the first mark-all response, '2' is a mark for the second mark-all response, and so forth. On the final record of a mark-all series, a respondent with any or all choices marked has '-8' indicating done selecting or no more, a reply of don't know to the series has '-2' and a refusal to answer the series has '-1'. For some variables, the final record of a mark-all series may have additional options; and for some mark-all groups, done selecting is indicated by '0.'

THE INTRA-FAMILY TRANSFERS SECTION

This section looks at transfers of time and money to the respondent/husband by certain children and to these children by the respondent/husband. During the interview, a child roster is created for respondent's biological, adopted, and step children, both inside and outside the household. Whether transfer information is collected for a child depends on the child's age, whether he/she has been married, and whether he/she has children. How the transfer questions are asked—whether for each child individually, or for a group of children together—depends on the number of eligible children inside and outside the household. The content of transfer questions includes the purpose of the transfer (loan or gift) and the amount of money or help given or received. Questions are asked separately for transfers in the past 12 months and for larger transfers ever made by the respondent/husband or their children. Other questions contained in this section include respondent support of children away at college, and transfers between respondents in the Young Women cohort and their mothers in the Mature Women cohort. More information is available in the 1999 editions of the Mature and Young Women user's guides.

There are some problems in this section, noted on the documentation records. IFT-CK-G contains values of "2" only because OWNRENT_FLG is not set, so for household children age 19 or over, the skip is always to IFT-14. Data quality in this set of questions may be compromised for cases where the respondent lives with the child because the question text is incorrect. Those few cases in IFT-18 through IFT-21 incorrectly default from D/R responses to time unit in IFT-17. Second, because YNGCHD_FLG incorrectly flags too many respondents as having children age 13 or less, too many respondents flow through IFT-CK-M, IFT-CK-M1, and IFT-CK-M2 to items that ask whether the child helps the respondent care for her (young) children. Specifically, there are about 300 respondents with children age 13 or less, and about 2800 that have the YNGCHD flag. A second problem for the CK-M items is that in IFT-CK-M, option "2" was stored as "1" and in IFT-CK-M2, "3" was stored as "1". The skips, however, are maintained.

NEW CREATED VARIABLES

Two new sets of created variables are provided in this release. One set provides labor force status (working/not working) by calendar week from 1994 through date of interview in 1999. The other set of variables provides ordered marriage start and end dates for the entire survey period: 1967–99 for the Mature Women cohort and 1968–99 for the Young Women cohort.

Calendar Week Labor Force Status
Variables in this set include:
> NCV-WORK-94-01 to NCV-WORK-94-52 (working/not each week of 1994)
> NCV-WORK94 (total weeks working in 1994)
> NCV-WORK-95-01 to NCV-WORK-95-52 (working/not each week of 1995)
> NCV-WORK95 (total weeks working in 1995)
> NCV-WORK-96-01 to NCV-WORK-96-52 (working/not each week of 1996)
> NCV-WORK96 (total weeks working in 1996)
> NCV-WORK-97-01 to NCV-WORK-97-52 (working/not each week of 1997)
> NCV-WORK97 (total weeks working in 1997)
> NCV-WORK-98-01 to NCV-WORK-98-52 (working/not each week of 1998)
> NCV-WORK98 (total weeks working in 1998)
> NCV-WORK-99-01 to NCV-WORK-99-39 (working/not each week through the respondent's interview date in 1999)
> NCV-WORK99 (total weeks working to date of interview in 1999)

Data for these variables come from employer start and stop dates reported in completed interviews for 1995, 1997, and 1999. A value of "1" for a calendar week labor force status variable indicates that the respondent was working during that week; a value of "0" indicates that the respondent was not working during that week. A respondent who is not interviewed is missing (-4) the calendar week variables. A respondent is also missing (-4) the calendar weeks after the date of her most recent interview. If the job start or end year is provided, an unknown or missing day is set to 15; an unknown or missing month is set to 1 (January). If days provided are inconsistent for a given month (e.g., April 31), the day is reset to the closest consistent day (e.g., April 30). Total weeks worked is the sum of weeks worked during the calendar year; it is non-missing for any respondent with at least one known weekly status during the year.

Ordered Marriage Start and End Dates
Variables in this set include:
> STDAT01-STDAT05 (marriage start dates YYMM)
> ENDAT01-ENDAT05 (marriage end dates YYMM)

Data for these variables come from marriage dates provided from 1967 to 1999 for Mature Women and from 1968 to 1999 for Young Women. All ordered marriage start and end dates are created numeric variables using the form YYMM. For example, a woman first married in October of 1965 would have STDAT01 = 6510. Consistent with the NLSY79, missing codes for these created variables indicate never married (-999), first marriage never ended (-998), first marriage ended and no second marriage is reported (-997), and so forth.

In some cases described in more detail below, a woman may report her marital status as married but not provide a marriage date, or her first reported marriage date might be an end date. In the former case, the missing code applied to the created variable is unreported (0) for all the marriage dates; in the latter case, the unreported code (0) is applied to the first marriage start date and all subsequent dates follow the usual pattern.

A number of rules were used in order to construct this system of dates.

- The first marriage (STDAT01, ENDAT01) is defined as the marriage that is current at the time of the first interview, or the first marriage reported after the first interview. For example, at the time of their first interview Mature Women (ages 30-44) were asked for the dates of their first marriage and for the dates of their latest marriage. Because we do not know the entire marital history, we consider the first

marriage to be the marriage current at the time of the first interview. It is the first marriage of the interview period.

- The first reported start and end date(s) of a marriage are preferred. Because the same dates are not always reported in succeeding interviews for the same marriage, the first report is preferred, as it is closer in time to the event and more likely to be accurate.

- A marriage end is defined as a widowed or divorced status. Dates for separation are not included in these ordered marriage dates as end dates because they are not collected consistently over the interview period.

- A valid start date is either the first reported marriage or requires a prior end date. If a woman reports two marriage start dates in succession and has no marriage end date between them, the second marriage start date is not considered valid and is not included in the ordered set. This can happen if more than one marital status change occurs between two interviews and only one change is recorded (for example, see Young Women 1969). More often, a second start date is an alternate report of the first marriage start date (see 2 above).

- The first available marriage date may be an end date. A woman may report divorced or widowed as her first marital status and provide a date for that without providing/being asked for a start date for that marriage. In this case her first marriage date is an end date, and any marriage that succeeds this is her second marriage.

- Not all women who report being married have a marriage date. A question about marriage dates is not included in every survey, and a woman may not have been interviewed at the time the question was included. This is particularly relevant for Young Women (ages 14-24 at the time of their first interview), who were not asked for marriage dates until 1978. About 300 of these women did report being married before 1978, but were deceased prior to that year or were not interviewed in 1978 or a later survey; no marriage dates are available for them.

- If the marriage-start or marriage-end year is known and the month is unknown, the month is set to June (06).

SAS Code for 1999 Created Variables

This section contains the SAS programs used to generate the 1999 created variables. As discussed above, the 1999 variables include those based on 1999 interview data as well as new variables spanning multiple survey years. The programs are presented in the following order:

1. Create Reason for Noninterview for all Respondents
2. Create Monthly Labor Recode for the Respondent
3. Create Monthly Labor Recode for the Husband/Partner
4. Edits to the Transfers Section to Correct Instrument Errors
5. Create Number of Household and Number of Family Members
6. Take Most Recent Highest Grade Completed and Update with Data from 1999
7. Create Total Net Family Assets. Set to Missing if Any Amount Is Missing
8. Create Summation and Net Family Income
9. Recode the 1990 Industry Codes to Create the 1980 Industry Codes
10. Recode the 1990 Occupation Codes to Create the 1980 Occupation Codes
11. Create Hourly Rate of Pay for Each Job Active Since DOLI
12. Edit the Husband Employer Sort Section
13. Edit the Respondent Employer Sort Section
14. Create the Weeks Working, Unemployed, and out of the Labor Force for the Respondent
15. New Created Variables: Marriage Start and End Dates
16. New Created Variables: Calendar Week Labor Force Status

/*1. Create Reason for Noninterview for all Respondents */

```
if R3498100>0 then R4267000=R3498100;
if 201 le R4266900 le 205 then R4267000=0;
if R4266900=213 or R4266900=219 then R4267000=11;
if R4266900=214 then R4267000=5;
if R4266900=216 then R4267000=1;
if R4266900=217 then R4267000=6;
if R4266900=218 then R4267000=9;
if R4266900=223 then R4267000=7;
if R4266900=234 then R4267000=8;
if R4266900=250 then R4267000=10;
if R4266900=251 then R4267000=13;
if 261 le R4266900 le 271 then R4267000=2;
if R4266900=260 then R4267000=4;

/*create or edit as follows for completed interviews only*/
if 201 le R4266900 le 205 then do;
```

/*2. Create Monthly Labor Recode for the Respondent */

```
if (R4332000 ge 0 & R4332200 ge 0) then HRUSLT=R4332000+R4332200;
  else if (R4332000 ge 0 and R4332200 LT 0) then HRUSLT=R4332000;
  else if (R4332000 LT 0 & R4332200 ge 0) then HRUSLT=R4332200;

  if (R4333500 ge 0 & R4334000 ge 0) then HRACTT=R4333500+R4334000;
  else if (R4333500 ge 0 & R4334000 LT 0) then HRACTT=R4333500;
  else if (R4333500 LT 0 & R4334000 ge 0) then HRACTT=R4334000;

  if (HRACTT=0 & (R4329500=-1 | R4329500=-2 | R4329500=0)) then HRCK6=1;
```

```
else if HRACTT=0 then HRCK6=2;
else HRCK6=3;


if ((R4329500=0 | R4329500=-1 | R4329500=-2) & ((0 le R4333500 LT 15) | R4333500=-2)) then HRCK7=1;
else if (R4329500=0 | R4329500=-1 | R4329500=-2) & R4333500 ge 15 then HRCK7=2;
else if (HRUSLT ge 35 | R4332400=1) & (HRACTT LT 35) & ((R4333500 ge 0) | (R4334000 ge 0)) then
      HRCK7=3;
else if (R4332700=1 & HRACTT LT 35) & (1 le R4332800 le 3 ) then HRCK7=4;
else HRCK7=5;


if (R4329200=1 & HRCK6=3) | (R4329400=1 & HRCK7 ge 2) then R4344300=1;
else if (R4331600=1 | R4331600=-1 | R4331600=-2 | R4331600=0) then R4344300=2;
else if R4334800=1 | R4334800=-1 | R4334800=-2 | R4334900=1 then R4344300=3;
else if R4340700=1 | R4340700=-1 | R4340700=-2 | R4340800=1 | R4340800=2 then R4344300=4;
else if ((R4329200=3 & R4276700 ge 50) & R4330100=1 & R4335800=1 & R4340700=0 & (R4340800=3 |
      R4340800=4 | R4340800=-2 | R4340800=-1)) then R4344300=5;
else if ((R4330100=0 | R4344100=1 | (R4335800=3 & R4276700 ge 50)) | (R4342200=3 & R4276700 ge 50) |
      R4344200=5 | (R4276700 ge 50 & (R4355400=4 | R4357400=4 | R4359200=4 | R4361000=4 |
      R4362800=4 | R4364000=4))) then R4344300=5;
else if R4329200=3 & R4276700 ge 50 & R4329700=1 then R4344300=5;
else if (R4330200=1 | R4330300=1 | R4329800=1 | R4344200=1) then R4344300=6;
else R4344300=7;
```

/*3. Create Monthly Labor Recode for the Husband/Partner*/

```
/* edit: code with asterisk omitted due to 1999 instrument error on HWH4 and HWH5*/

if 1 le R4317700 le 4 then do;
   if (R4738300 ge 0 & R4738500 ge 0) then HRUSLTH=R4738300+R4738500;
   else if (R4738300 ge 0 and R4738500 LT 0) then HRUSLTH=R4738300;
   else if (R4738300 LT 0 & R4738500 ge 0) then HRUSLTH=R4738500;


   if (R4739700 ge 0 & R4740200 ge 0) then HRACTTH=R4739700+R4740200;
   else if (R4739700 ge 0 & R4740200 LT 0) then HRACTTH=R4739700;
   else if (R4739700 LT 0 & R4740200 ge 0) then HRACTTH=R4740200;


   if ((HRACTTH=0) & (R4735800=-1 | R4735800=-2 | R4735800=0)) then HRCK6H=1;
   else if HRACTTH=0 then HRCK6H=2;
   else HRCK6H=3;


   if ((R4735800=0 | R4735800=-1 | R4735800=-2) & (0 le R4739700 LT 15 | R4739700=-2)) then HRCK7H=1;
   else if ((R4735800=0 | R4735800=-1 | R4735800=-2) & R4739700 ge 15) then HRCK7H=2;
   else if ((HRUSLTH ge 35 | R4738700=1) & (HRACTTH LT 35) & (R4739700 ge 0 | R4740200 ge 0)) then
         HRCK7H=3;
   else if ((R4739000=1 & HRACTTH LT 35) & (1 le R4739100 le 3 )) then HRCK7H=4;
   else HRCK7H=5;


   if (R4735500=1 & HRCK6H=3) | (R4735700=1 & HRCK7H ge 2) then R4750500=1;
   else if (R4737900=1 | R4737900=-1 | R4737900=-2 | R4737900=0) then R4750500=2;
   else if R4741000=1 | R4741000=-1 | R4741000=-2 | R4741100=1 then R4750500=3;
   else if R4746900=1 | R4746900=-1 |R4746900=-2 | R4747000=1 | R4747000=2 then R4750500=4;
   else if ((R4735500=3 & R4735300 ge 50) & R4736400=1 & R4742000=1 & R4746900=2 & (R4747000=3
         |R4747000=4 | R4747000=-2 | R4747000=-1)) then R4750500=5;
   else if (R4736400=0 | R4750300=1 | (R4742000=3 & R4735300 ge 50) | (R4748400=3 & R4735300 ge 50) |
         R4750400=5 |(R4735300 ge 50 & (R4764200=4 | R4766200=4 | R4768000=4 | R4769700=4 |
         R4773100=4))) then R4750500=5;
```

```
*else if R4735500=3 & H00700=1 & R4735300 ge 50 then R4750500=5;
*else if (R4736500=1 | R4736600=1 | H00800=1 | R4750400=1) then R4750500=6;
else if (R4736500=1 | R4736600=1 | R4750400=1) then R4750500=6;
else R4750500=7;
end;
```

/*4. Edits to the Transfers Section to Correct Instrument Errors*/

/* First find all children on person roster based on 1999 relationship to respondent. Set relationship and date of birth for these on child roster (posn shown by CNUM). Then fill date of birth for internal and get date of birth for external from IFT-6. Correct the IFT relationship for internal children*/

```
array RLTN        R4293200 R4295300 R4297400 R4299400 R4301400 R4303500 R4305600 R4307400
                  R4309100 R4310800 R4312500 R4314000;
array R1018        R4293100 R4295200 R4297300 R4299300 R4301300 R4303400 R4305500 R4307300
                  R4309000 R4310700 R4312400 R4313800;
array PST          R4318800 R4319400 R4320000 R4320600 R4321200 R4321800 R4322400 R4323000
                  R4323600 R4324200 R4324800 R4325400;
array BYY          R4293600 R4295700 R4297800 R4299800 R4301800 R4303900 R4306000 R4307800
                  R4309500 R4311200 R4312900 R4314400;
array BMM          R4293400 R4295500 R4297600 R4299600 R4301600 R4303700 R4305800 R4307600
                  R4309300 R4311000 R4312700 R4314200;
array BDD          R4293500 R4295600 R4297700 R4299700 R4301700 R4303800 R4305900 R4307700
                  R4309400 R4311100 R4312800 R4314300;
array CFL         CFL02-CFL13;
array CREL        CREL02-CREL13;
array CMDY        CMDY02-CMDY13;
CNUM=0;
   do over RLTN; if PST=1 then do;
      if RLTN=. and R1018>0 then RLTN=R1018;
      if (2 le RLTN le 3 or 17 le RLTN le 20) then do; CFL=1; end;
      if CFL=1 then do; CREL=RLTN;
         if BMM>0 and BDD>0 and BYY>0 then CMDY=BMM*1000000+BDD*10000+BYY;
         if BYY=-2 then CMDY=-2; if BYY=-1 then CMDY=-1;
         CNUM=CNUM+1;
         if CNUM=1 and R5176100 =. then do; R5152300=CREL; R5154200=CMDY; end;
         if CNUM=2 and R5176200 =. then do; R5152400=CREL; R5154300=CMDY; end;
         if CNUM=3 and R5176300 =. then do; R5152500=CREL; R5154400=CMDY; end;
         if CNUM=4 and R5176400 =. then do; R5152600=CREL; R5154500=CMDY; end;
         if CNUM=5 and R5176500 =. then do; R5152700=CREL; R5154600=CMDY; end;
         if CNUM=6 and R5176600 =. then do; R5152800=CREL; R5154700=CMDY; end;
         if CNUM=7 and R5176700 =. then do; R5152900=CREL; R5154800=CMDY; end;
         if CNUM=8 and R5176800 =. then do; R5153000=CREL; R5154900=CMDY; end;
         if CNUM=9 and R5176900 =. then do; R5153100=CREL; R5155000=CMDY; end;
      end;
   end;
end;

array IFT5         R5185300 R5189700 R5193900 R5198100 R5201700 R5205300 R5208900 R5212500
                  R5216000 R5219000 R5221900 R5224800 R5227700 R5230600 R5233500 R5236100
                  R5238200 R5240300 R5242300 R5244300 R5245900 R5247900 R5249400;
array CHDREL       R5152300 R5152400 R5152500 R5152600 R5152700 R5152800 R5152900 R5153000
                  R5153100 CHDREL10-CHDREL23;
array IFTREL       R5166000 R5166100 R5166200 R5166300 R5166400 R5166500 R5166600 R5166700
                  R5166800 IFTREL10-IFTREL23;
```

```
array CYY          R5185600 R5190000 R5194200 R5198400 R5202000 R5205600 R5209200 R5212800
                   R5216300 R5219300 R5222200 R5225100 R5228000 R5230900 R5233800 R5236400
                   R5238500 R5240600 R5242600 R5244600 R5246200 R5248200 R5249700;
array CMM          R5185400 R5189800 R5194000 R5198200 R5201800 R5205400 R5209000 R5212600
                   R5216100 R5219100 R5222000 R5224900 R5227800 R5230700 R5233600 R5236200
                   R5238300 R5240400 R5242400 R5244400 R5246000 R5248000 R5249500;
array CDD          R5185500 R5189900 R5194100 R5198300 R5201900 R5205500 R5209100 R5212700
                   R5216200 R5219200 R5222100 R5225000 R5227900 R5230800 R5233700 R5236300
                   R5238400 R5240500 R5242500 R5244500 R5246100 R5248100 R5249600;
array CHMDY        CHMDY01-CHMDY23;
array CDOB         R5154200 R5154300 R5154400 R5154500 R5154600 R5154700 R5154800 R5154900
                   R5155000 CDOB10-CDOB23;
array CHDDOB       R5168300 R5168400 R5168500 R5168600 R5168700 R5168800 R5168900 R5169000
                   R5169100 R5169200 R5169300 R5169400 R5169500 R5169600 R5169700 R5169800
                   R5169900 R5170000 R5170100 R5170200 R5170300 R5170400 R5170500;
    do over CHDDOB; CHDDOB=CDOB;
       if CMM>0 and CDD>0 and CYY>0 then CHMDY=CMM*1000000+CDD*10000+CYY;
       if CYY=-2 then CHMDY=-2; if CYY=-1 then CHMDY=-1;
       if CDOB=. and CHMDY ge -2 then CHDDOB=CHMDY;
       if 2 le CHDREL le 3 then IFTREL=1; if 17 le CHDREL le 18 then IFTREL=2;
       if 19 le CHDREL le 20 then IFTREL=3; if 1 le IFT5 le 3 then IFTREL=IFT5;
    end;
```

/*5. Create Number of Household and Number of Family Members */

```
/* note that specs are not the same as in prior years*/

R4317300=0; R4317200=0;
array HHM    R4318200 R4318800 R4319400 R4320000 R4320600 R4321200 R4321800 R4322400 R4323000
             R4323600 R4324200 R4324800 R4325400;
array REL    R4317900 R4293200 R4295300 R4297400 R4299400 R4301400 R4303500 R4305600 R4307400
             R4309100 R4310800 R4312500 R4314000;
array R10    R101801 R4293100 R4295200 R4297300 R4299300 R4301300 R4303400 R4305500 R4307300
             R4309000 R4310700 R4312400 R4313800;
    do over HHM;
       if HHM=1 then R4317200=R4317200+1;
       if R101>0 then REL=R101;
       if 2 le REL le 22 and HHM=1 then R4317300=R4317300+1;
    end;
```

/*6. Take Most Recent Highest Grade Completed and Update with Data from 1999*/

```
HGC=R4192800;
if R4254100=1995 then HGC=R3476600;
if R4254100=1993 then HGC=R1520410;
if R4254100=1991 then HGC=R1346410;
if R4254100=1988 then HGC=R1215110;
if R4254100=1987 then HGC=R1097410;
if R4254100=1985 then HGC=R1051610;
if R4254100=1983 then HGC=R0929510;
if R4254100=1982 then HGC=R0797110;
if R4254100=1992 then HGC=R0989700;
if R4254100=1989 then HGC=R0989700;
if COHORT='MW' & HGC=. then HGC=R0381500;
if COHORT='MW' & HGC=. then HGC=R0079000;
if HGC=-7 then HGC=95;
```

```
if HGC=-6 then HGC=12;
if HGC=-5 then HGC=94;
if HGC=-4 then HGC=93;
if HGC=-3 then HGC=98;
if HGC=-2 then HGC=97;
if HGC=-1 then HGC=96;
if 1 le R5142000 le 10 then R5141500=R5142000+7;
if R5142000=11 then R5141500=95;
if 1 le R5142200 le 3 then R5141500=12;
if 3 le R5142500 le 4 then R5141500=18;
if R5142500=2 then R5141500=16;
if R5142500=1 then R5141500=14;
if HGC>R5141500 then R5141500=HGC;
```

/*7. Create Total Net Family Assets. Set to Missing if Any Amount Is Missing*/

```
C=0;
  if R5104500=0 & (R5104600<0 | R5104600=.)&(R5104700<0 | R5104700=.) then do;
    HOUSE=0; PROPDE=0; end;
  if R5104700>=0 & R5105200>=0 then PROPDE = R5104700 + R5105200;
  if R5104700<0 & R5105200>=0 then PROPDE = R5105200;
  if R5104700>=0 & R5105200<0 then PROPDE = R5104700;
  if R5104600>=0 & PROPDE>=0 then HOUSE=R5104600-PROPDE;
  if R5109300=1 & 0 le R5104600 le R5109200 &  0 le R5104700 le R5109400 then HOUSE=0;
  if HOUSE^=. then ASSET=HOUSE;    else C=C+1;
  if (R5105500=0 | R5120200=0) & (R5105600<0 | R5105600=.) then SAVE=0;
  else if R5105600>=0 then SAVE=R5105600;
  if R5105700=1 & R5105800=1 then SAVE=40000;
  if R5105700=1 & R5105800=0 then SAVE=25000;
  if R5105700=0 & R5105900=1 then SAVE=5500;
  if R5105700=0 & R5105900=0 then SAVE=1000;
  if R5105700=-1 | R5105800=-1 | R5105900=-1 | R5105700=-2 | R5105800=-2 | R5105900=-2  then SAVE=.;
  if SAVE^=. then ASSET=ASSET+SAVE; else C=C+1;
  if (R5106000=0 | R5120200=0) & (R5106100<0 | R5106100=.) then BOND=0;
  else if R5106100>=0 then BOND=R5106100;
   if R5106200=1 & R5106300=1 then BOND=5000;
   if R5106200=1 & R5106300=0 then BOND=3000;
   if R5106200=0 & R5106400=1 then BOND=750;
   if R5106200=0 & R5106400=0 then BOND=500;
   if R5106200=-1 | R5106300=-1 | R5106400=-1 | R5106200=-2 | R5106300=-2 | R5106400=-2 then BOND=.;
  if BOND^=. then ASSET=ASSET+BOND; else C=C+1;
  if (R5106500=0 | R5120200=0) & (R5106600<0 | R5106600=.) then STOCK=0;
  else if R5106600>=0 then STOCK=R5106600;
   if R5106700=1 & R5106800=1 then STOCK=40000;
   if R5106700=1 & R5106800=0 then STOCK=27500;
   if R5106700=0 & R5106900=1 then STOCK=10000;
   if R5106700=0 & R5106900=0 then STOCK=5000;
   if R5106700=-1 | R5106800=-1 | R5106900=-1 | R5106700=-2 | R5106800=-2 | R5106900=-2 then STOCK=.;
  if STOCK^=. then ASSET=ASSET+STOCK; else C=C+1;
   if R5107000=0  & (R5107100<0 | R5107100=.) then IRA=0;
   else if R5107100 ge 0 then IRA=R5107100;
   if R5107200=1 & R5107300=1 then IRA=30000;
   if R5107200=1 & R5107300=0 then IRA=22500;
   if R5107200=0 & R5107400=1 then IRA=10000;
   if R5107200=0 & R5107400=0 then IRA=5000;
   if R5107200=-1 | R5107300=-1 | R5107400=-1 | R5107200=-2 | R5107300=-2 | R5107400=-2 then IRA=.;
```

```
if IRA^=. then ASSET=ASSET+IRA;  else C=C+1;
 if R5107500=0 & (R5107600<0 | R5107600=.) then LOAN=0;
 else if R5107600>=0 then LOAN=R5107600;
 if LOAN^=. then ASSET=ASSET+LOAN; else C=C+1;
 if (R5107700=0 | R5107900=1) & (R5108000<0 | R5108000=.) then INSU=0;
 else if R5108000 ge 0 then INSU=R5108000;
 if INSU^=. then ASSET=ASSET+INSU;  else C=C+1;
 if R5108300=0 & (R5108400<0 | R5108400=.) then TRUS=0;
 else if R5108400 >= 0 then TRUS = R5108400;
 if TRUS^=. then ASSET=ASSET+TRUS;  else C=C+1;
 if R5108800=0 & (R5108900<0 | R5108900=.) then SETTLE=0;
 else if R5108800=1 & R5108900<0 then SETTLE=0;
 else if R5108900>=0 then SETTLE=R5108900;
  if SETTLE^=. then ASSET=ASSET+SETTLE;  else C=C+1;
 if (R5109100=0 | R5119600=0) & (R5109200<0 | R5109200=.) & (R5109400<0 | R5109400=.) then
    IFARM=0;
 else if R5109200>=0 & R5109400>=0 then IFARM=R5109200-R5109400;
 if IFARM^=. then ASSET=ASSET+IFARM;  else C=C+1;
 if (R5109500=0 |(R5111800=0 & R5115800=0)) & (R5109600<0 | R5109600=.) & (R5109700<0 |
    R5109700=.) then IBUS=0;
 else if R5109600>=0 & R5109700>=0 then IBUS=R5109600-R5109700;
 if IBUS^=. then ASSET=ASSET+IBUS;  else C=C+1;
 if (R5109800=0 | R5119900=0) & (R5109900<0 | R5109900=.) & (R5110000<0 | R5110000=.) &
    (R5110100<0 | R5110100=.) then IREAL=0;
 else if R5109900>=0 & R5110000>=0 & R5110100>=0
 then IREAL=R5109900-R5110000-R5110100;
 if IREAL^=. then ASSET=ASSET+IREAL; else C=C+1;
 if R5110200=0 & (R5110300<0 | R5110300=.) then DEBT=0;
 else if R5110300>=0 then DEBT=R5110300;
 if DEBT^=. then ASSET=ASSET-DEBT;  else C=C+1;

 if C=0 then do;
  if ASSET<=-99999 then R5131100=-99999;
  if ASSET>=4999999 then R5131100=4999999;
  if ASSET>-99999 & ASSET<4999999 then R5131100=ASSET;
 end;
 if C>0 then R5131100=.;
```

/*8. Create Summation and Net Family Income */

/* Sum all income amounts to create summation. If all are missing then summation is missing*/
/* Set net family income equal summation. If any amount is missing, set net family income to missing*/

```
 NI=0;  CC=0;
 array ALLINC    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     ALIR    ALIP
                 OTHER;
 array ANY       R5111600 R5111800 R5112100 R5112400 R5112700 R5113100 R5113300 R5113500
                 R5113700 R5114000 R5114200 R5114400 R5114600 R5114800 R5115000 R5115200
                 R5115600 R5115800 R5116100 R5116400 R5116700 R5117100 R5117300 R5117500
                 R5117700 R5118100 R5118300 R5118500 R5118700 R5118900 R5119100 R5119300
                 R5119600 R5119900 R5120200 R5120400 R5120700 R5121000 R5121300 R5121500
                 R5128100;
```

```
array AMT          R5111700 R5111900 AMT03   AMT04    AMT05    R5113200 R5113400 R5113600
                   R5113800 R5114100 R5114300 R5114500 R5114700 R5114900 R5115100 R5115400
                   R5115700 R5115900 AMT19   AMT20   AMT21    R5117200 R5117400 R5117600
                   R5117900 R5118200 R5118400 R5118600 R5118800 R5119000 R5119200 R5119500
                   R5119700 R5120000 R5120300 AMT36-AMT38 R5121400 AMT40   R5128200;
array LOSS         LOSS01 R5112000 LOSS03-LOSS17 R5116000 LOSS19 LOSS20-LOSS32 R5119800
                   R5120100 LOSS35-LOSS39 R5121600 LOSS41;
array PER          PER01 PER02 R5112200 R5112500 R5112800 PER06- PER18 R5116200 R5116500
                   R5116800 PER22-PER35 R5120500 R5120800 R5121100 PER39-PER41;
array UNIT         UNIT01 UNIT02 R5112300 R5112600 R5112900 UNIT06-UNIT18 R5116300 R5116600
                   R5116900 UNIT22-UNIT35 R5120600 R5120900 R5121200 UNIT39-UNIT41;
if R5113900=0 then do; REPRI=0; REMIL=0; REFED=0; RESTE=0; REUNI=0; REIRA=0; REOTH=0; end;
if R5115500 ne 1 then do; SWAGE=0; SBUS=0; SUNEM=0; SSUB=0; SSOC=0; SVET=0; SCOM=0;
SSSD=0; SDIS=0; end;
if R5115500 ne 1 or R5118000=0 then do;
   SREPRI=0; SREMIL=0; SREFED=0; SRESTE=0; SREUNI=0; SREIRA=0; SREOTH=0; end;
if R5121500=1 then R5121500=3;
do over ALLINC;
if PER ge 0 and UNIT ge 0 then AMT=PER*UNIT;
if (ANY=0 or ANY=4) and (AMT<0 or AMT=.) then ALLINC=0;
if ANY=1 and AMT ge 0 then ALLINC=AMT;
if ANY=3 and LOSS ge 0 then ALLINC=0-LOSS;
if ALLINC=. then CC=CC+1;
if ALLINC ne . then do; NI=NI+1;
if NI=1 then R5131200=ALLINC;
if NI>1 then R5131200=R5131200+ALLINC; end; end;
  if R5121700=0 and R5121800=0 then CHDSUP=0;
   if R5122800=2 & R5125700>=0 then CHDSUP=R5125700;
   if R5122800=1 & R5126500>=0 then CHDSUP=R5126500;
   if R5122800=3 & R5125700>=0 & R5126500>=0 then CHDSUP=R5125700+R5126500;
   if R5121900>=0 then CHDSUP=R5121900;
  if R5127000=0 then RCHD=0;
   if R5127100>=0 then RCHD=52*R5127100;
   if R5127200>=0 then RCHD=12*R5127200;
   if R5127300>=0 then RCHD=R5127300;
  if (R5115500 ne 1 | R5127600=0) then HPCHD=0;
   else if R5127700>=0 then HPCHD=52*R5127700;
   else if R5127800>=0 then HPCHD=12*R5127800;
   else if R5127900>=0 then HPCHD=R5127900;
if CHDSUP>=0 & RCHD>=0 & HPCHD>=0 then CHILD=CHDSUP-RCHD-HPCHD;
if CHILD=. then CC=CC+1;
if CHILD ne . then do; NI=NI+1; if NI=1 then R5131200=CHILD; if NI>1 then
        R5131200=R5131200+CHILD; end;
   if R5128400 ne 2 then FAM=0;
   if R5128500=1 then FAM=  2000;
   else if R5128500=2 then FAM=  5000;
   else if R5128500=3 then FAM=  6750;
   else if R5128500=4 then FAM=  8250;
   else if R5128500=5 then FAM= 12500;
   else if R5128500=6 then FAM= 16250;
   else if R5128500=7 then FAM= 18750;
   else if R5128500=8 then FAM= 22500;
   else if R5128500=9 then FAM= 30000;
   else if R5128500=10 then FAM= 42500;
   else if R5128500=11 then FAM= 62500;
   else if R5128500=12 then FAM= 87500;
```

```
else if R5128500=13 then FAM=100000;
else if R5128500=14 then FAM=0;
if FAM ne . then do; NI=NI+1; if NI=1 then R5131200=FAM; if NI>1 then R5131200=R5131200+FAM;
    end;
if R5131200 ne . and R5131200 LT -99999 then R5131200=-99999;
if R5131200>349999 then R5131200=349999;
R5131300=R5131200; if R5131300=. then R5131300=0;
if CC>0 then R5131200=.;
```

/*9. Recode the 1990 Industry Codes to Create the 1980 Industry Codes*/

/* for 1999 jobs active since DOLI for the respondent and husband/partner */

```
array X      R4442200 R4464200 R4484200 R4504200 R4523400 R4542300 R4560100 R4577300 R4593000
             R4606900 R4618800 R4629400 R4637300 R4856500 R4869900 R4881200 R4892100 R4902700
             R4913100 R4922700 R4930900 R4938800 R4945100 R4951300 R4957100 R4963100 R4968700
             R4974600 R4979700 R4985000;
array Y      R4442300 R4464300 R4484300 R4504300 R4523500 R4542400 R4560200 R4577400 R4593100
             R4607000 R4618900 R4629500 R4637400 R4856600 R4870000 R4881300 R4892200 R4902800
             R4913200 R4922800 R4931000 R4938900 R4945200 R4951400 R4957200 R4963200 R4968800
             R4974700 R4979800 R4985100;
array bEg    R4439800 R4462100 R4482100 R4502100 R4521300 R4540200 R4558100 R4575300 R4591000
             R4604900 R4617000 R4627800 R4635900 R4854100 R4867800 R4879200 R4890100 R4900700
             R4911200 R4920800 R4929000 R4936800 R4943200 R4949900 R4955200 R4961200 R4967300
             R4973000 R4978000 R4983600;
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;
```

/*10. Recode the 1990 Occupation Codes to Create the 1980 Occupation Codes */

/* for 1999 jobs active since DOLI for the respondent and husband/partner */

```
array XX      R4442700 R4464700 R4484700 R4504700 R4523900 R4542800 R4560600 R4577700 R4593500
              R4607400 R4619300 R4629900 R4637600 R4857100 R4870400 R4881800 R4892700 R4903300
              R4913700 R4923300 R4931500 R4939300 R4945700 R4951800 R4957700 R4963700 R4969100
              R4975000 R4980100 R4985400;
array YY      R4442800 R4464800 R4484800 R4504800 R4524000 R4542900 R4560700 R4577800 R4593600
              R4607500 R4619400 R4630000 R4637700 R4857200 R4870500 R4881900 R4892800 R4903400
```

R4913800 R4923400 R4931600 R4939400 R4945800 R4951900 R4957800 R4963800 R4969200
R4975100 R4980200 R4985500;

array bg   R4439800 R4462100 R4482100 R4502100 R4521300 R4540200 R4558100 R4575300 R4591000
R4604900 R4617000 R4627800 R4635900 R4854100 R4867800 R4879200 R4890100 R4900700
R4911200 R4920800 R4929000 R4936800 R4943200 R4949900 R4955200 R4961200 R4967300
R4973000 R4978000 R4983600;

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;	

/*11. Create Hourly Rate of Pay for Each Job Active Since DOLI */

/* for respondent and husband/partner */

array HROP   R4447500 R4468900 R4488900 R4508900 R4528100 R4546800 R4564200 R4580100
R4596400 R4610000 R4621400 R4632600 R4858800 R4872000 R4883400 R4894300
R4904900 R4915300 R4924800 R4932700 R4940300 R4947100 R4952600 R4959100
HROP25 R4969900 HROP27 R4980900 HROP29;

array WROP   R4451700 R4472200 R4492100 R4511500 R4530500 R4549200 R4566100 R4581900
WROP9-WROP12 R4863100 R4874600 R4885500 R4896300 R4907300 R4917000
WROP19-WROP24 R4964700 WROP26-WROP29;

array MROP   R4452200 R4472700 R4492600 R4511900 R4531000 R4549600 R4566500 R4582200
R4597900 MROP10 R4622200 MROP12 R4863600 R4875100 R4886000 R4896700
R4907700 R4917400 MROP19-MROP26 R4976100 MROP28 MROP29;

array AROP   R4452700 R4473100 R4493100 R4512300 R4531500 R4550100 R4566900 R4582500
R4598300 AROP10 R4622500 AROP12 R4864100 R4875500 R4886500 R4897200
R4908100 R4917800 AROP19-AROP28 R4986200;

array BROP   R4453200 R4473600 R4493600 R4512800 R4531900 BROP6 R4567400 R4582900 BROP9-
BROP12 R4864600 R4876000 R4887000 R4897700 BROP17-BROP29;

array OROP   R4453900 R4474200 R4494200 R4513400 R4532500 R4550800 R4568000 OROP8-OROP12
R4865300 R4876700 R4887700 R4898400 R4908800 R4918500 OROP19-OROP29;

array SWHRP   R4454300 R4474500 R4494600 R4513800 R4532900 R4551100 R4568300 R4583700
R4599000 R4611000 R4623100 R4633600 R4865700 R4877100 R4888100 R4898700
R4909200 R4918800 R4927600 R4934900 R4941900 R4948600 R4954000 R4960000
R4965900 R4971700 R4976700 R4982300 R4986700;

array USHD   R4441300 R4463400 R4483400 R4503400 R4522600 R4541500 R4559300 R4576500
R4592200 R4606100 R4618200 R4628800 R4855600 R4869100 R4880500 R4891400
R4902000 R4912400 R4922000 R4930200 R4938000 R4944400 R4950800 R4956400
R4962400 R4968200 R4973900 R4979000 R4984300;

array USHW   R4441800 R4463900 R4483900 R4503900 R4523100 R4542000 R4559800 R4577000
R4592700 R4606600 R4618500 R4629100 R4856100 R4869600 R4881000 R4891900

```
                    R4902500 R4912900 R4922500 R4930700 R4938500 R4944900 R4951100 R4956900
                    R4962900 R4968500 R4974400 R4979500 R4984800;
array TURP          R4447000 R4468400 R4488400 R4508400 R4527600 R4546300 R4563700 R4579700
                    R4595900 R4609500 R4620900 R4632100 R4858300 R4871500 R4882900 R4893800
                    R4904400 R4914800 R4924400 R4932300 R4939800 R4946700 R4952200 R4958700
                    R4964200 R4969500 R4975400 R4980500 R4985800;
array USWY          R4453500 R4473800 R4493800 R4513000 R4532100 R4550400 R4567600 R4583100
                    R4598600 R4610700 R4622700 R4633300 R4864900 R4876300 R4887300 R4898000
                    R4908400 R4918100 R4926900 R4934200 R4941200 R4948300 R4953700 R4959700
                    R4965200 R4971300 R4976300 R4981900 R4986400;
   do over SWHRP;
    if TURP=1 then SWHRP=HROP;
    if USHW LT USHD then USHW=-3;
    if USHW>0  then do;
     if (TURP=2 | TURP=3 | TURP=7) then SWHRP=(100*WROP)/USHW;
     if TURP=4 then SWHRP=(100*BROP)/(USHW*2);
     if (TURP=5 | TURP=8) then SWHRP=(100*MROP)/(USHW*4.33);
     if (TURP=6 & USWY ge 1) then SWHRP=(100*AROP)/(USHW*USWY);
     if OROP>0 then SWHRP=OROP;
     SWHRP = FLOOR(SWHRP + .5);
    end;  end;
```

/*12. Edit the Husband Employer Sort Section */

/* Edit impossible day of month so the date function will work and set deleted(=0) to missing. Use date function to find most recent endate then compare with DOLI and DOI. Call most recent job after DOLI HCPS and count current jobs and noncurrent jobs after DOLI. If current gt 1 or no current and noncurrent gt 1 unmark as HCPS any sort line gt 1. CK-HES-C, CK-HES-D, CK-HES-E, and CK-HES-F are incorrect in instrument so create these check items */

```
DOIDAT=MDY(R4260000,R4260100,R4260200);
DOLIDAT=MDY(R4253800,R4253900,R4254000);
if R4840100=1 then do;
HHIDAT=0;
array HSORTY   R4841600 R4842300 R4843000 R4843700 R4844400 R4845100 R4845800 R4846500
               R4847200 R4847900 R4848600 R4849300 R4850000 R4850700 R4851400 R4852000
               R4852600;
array HSORTM   R4841700 R4842400 R4843100 R4843800 R4844500 R4845200 R4845900 R4846600
               R4847300 R4848000 R4848700 R4849400 R4850100 R4850800 R4851500 R4852100
               R4852700;
array HSORTD   R4841800 R4842500 R4843200 R4843900 R4844600 R4845300 R4846000 R4846700
               R4847400 R4848100 R4848800 R4849500 R4850200 R4850900 R4851600 R4852200
               R4852800;
array HENDT    HNDT01-HNDT17;
do over HENDT;
if HSORTM=2 and HSORTD>28 then HSORTD=28;
if (HSORTM=4 or HSORTM=6 or HSORTM=9 or HSORTM=11) and HSORTD=31 then HSORTD=30;
if HSORTM=0 then HSORTM=.;
if HSORTD=0 then HSORTD=.;
if HSORTY>1900 then HSORTY=HSORTY-1900;
HENDT=MDY(HSORTM,HSORTD,HSORTY);
if HENDT>HHIDAT then HHIDAT=HENDT;
end;

HCUR=0; HNCUR=0;
   if HNDT01 ge DOLIDAT and HNDT01=HHIDAT then do;
```

```
    R4842000=1; if HNDT01=DOIDAT then HCUR=HCUR+1;
    else if HNDT01<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT02 ge DOLIDAT and HNDT02=HHIDAT then do;
    R4842700=1; if HNDT02=DOIDAT then HCUR=HCUR+1;
    else if HNDT02<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT03 ge DOLIDAT and HNDT03=HHIDAT then do;
    R4843400=1; if HNDT03=DOIDAT then HCUR=HCUR+1;
    else if HNDT03<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT04 ge DOLIDAT and HNDT04=HHIDAT then do;
    R4844100=1; if HNDT04=DOIDAT then HCUR=HCUR+1;
    else if HNDT04<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT05 ge DOLIDAT and HNDT05=HHIDAT then do;
    R4844800=1; if HNDT05=DOIDAT then HCUR=HCUR+1;
    else if HNDT05<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT06 ge DOLIDAT and HNDT06=HHIDAT then do;
    R4845500=1; if HNDT06=DOIDAT then HCUR=HCUR+1;
    else if HNDT06<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT07 ge DOLIDAT and HNDT07=HHIDAT then do;
    R4846200=1; if HNDT07=DOIDAT then HCUR=HCUR+1;
    else if HNDT07<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT08 ge DOLIDAT and HNDT08=HHIDAT then do;
    R4846900=1; if HNDT08=DOIDAT then HCUR=HCUR+1;
    else if HNDT08<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT09 ge DOLIDAT and HNDT09=HHIDAT then do;
    R4847600=1; if HNDT09=DOIDAT then HCUR=HCUR+1;
    else if HNDT09<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT10 ge DOLIDAT and HNDT10=HHIDAT then do;
    R4848300=1; if HNDT10=DOIDAT then HCUR=HCUR+1;
    else if HNDT10<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT11 ge DOLIDAT and HNDT11=HHIDAT then do;
    R4849000=1; if HNDT11=DOIDAT then HCUR=HCUR+1;
    else if HNDT11<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT12 ge DOLIDAT and HNDT12=HHIDAT then do;
    HCPS12=1; if HNDT12=DOIDAT then HCUR=HCUR+1;
    else if HNDT12<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT13 ge DOLIDAT and HNDT13=HHIDAT then do;
    HCPS13=1; if HNDT13=DOIDAT then HCUR=HCUR+1;
    else if HNDT13<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT14 ge DOLIDAT and HNDT14=HHIDAT then do;
    R4851100=1; if HNDT14=DOIDAT then HCUR=HCUR+1;
    else if HNDT14<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT15 ge DOLIDAT and HNDT15=HHIDAT then do;
```

```
    HCPS15=1; if HNDT15=DOIDAT then HCUR=HCUR+1;
    else if HNDT15<DOIDAT then HNCUR=HNCUR+1; end;

  if HNDT16 ge DOLIDAT and HNDT16=HHIDAT then do;
    HCPS16=1; if HNDT16=DOIDAT then HCUR=HCUR+1;
    else if HNDT16<DOIDAT then HNCUR=HNCUR+1; end;

  if HNDT17 ge DOLIDAT and HNDT17=HHIDAT then do;
    R4853000=1; if HNDT17=DOIDAT then HCUR=HCUR+1;
    else if HNDT17<DOIDAT then HNCUR=HNCUR+1; end;

if (HCUR>1 or (HCUR=0 and HNCUR>1)) then do;
array HCP    R4842000 R4842700 R4843400 R4844100 R4844800 R4845500 R4846200 R4846900 R4847600
             R4848300 R4849000 HCPS12 HCPS13 R4851100 HCPS15-HCPS16 R4853000;
array HLN    R4841900 R4842600 R4843300 R4844000 R4844700 R4845400 R4846100 R4846800 R4847500
             R4848200 R4848900 R4849600 R4850300 R4851000 R4851700 R4852300 R4852900;
    do over HCP; if HLN ne 1 then HCP=.; end;
end;

if HCUR ge 1 then R4840300=1; else if HCUR=0 and HNCUR ge 1 then R4840300=2; else R4840300=3;
if HCUR=1 then R4840400=1; else if HCUR>1 then R4840400=2;
if HCUR=0 and HNCUR=1 then R4840500=1; else if HCUR=0 and HNCUR>1 then R4840500=2;
if R4840300=1 or R4840300=2 then R4840700=1; else R4840700=0;
end;
```

/*13. Edit the Respondent Employer Sort Section */

```
/* Fill in the unknown start day with 15 and month with jan to use later in weeks vars.  Correct a bad end date
using the sort date.  Edit impossible day of month so date function will work and set deleted jobs to missing.
Use date function to find most recent job endate then compare with DOLI and DOI.  Call most recent job after
DOLI CPS and count current jobs and noncurrent jobs after DOLI.  If current gt 1 or no current and noncurrent
gt 1 unmark as CPS any sort line gt 1.  CK-RES-C, CK-RES-D, CK-RES-E, and CK-RES-F are incorrect in
instrument so create these check items */

HIDAT=0;
array SORTY    R4429700 R4430400 R4431100 R4431800 R4432500 R4433200 R4433900 R4434600
               R4435300 R4436000 R4436700 R4437400 R4438100;
array SORTM    R4429800 R4430500 R4431200 R4431900 R4432600 R4433300 R4434000 R4434700
               R4435400 R4436100 R4436800 R4437500 R4438200;
array SORTD    R4429900 R4430600 R4431300 R4432000 R4432700 R4433400 R4434100 R4434800
               R4435500 R4436200 R4436900 R4437600 R4438300;
array ENDT     ENDT01-ENDT13;
array REF4     R4408500 R4409600 R4410700 R4411800 R4412900 R4414000 R4415100 R4416200
               R4417300 R4418400 R4419500 R4420500 R4421500;
array STK      R4409000 R4410100 R4411200 R4412300 R4413400 R4414500 R4415600 R4416700
               R4417800 R4418900 R4419900 R4420900 R4421900;
array RFM      R4408700 R4409800 R4410900 R4412000 R4413100 R4414200 R4415300 R4416400
               R4417500 R4418600 R4419600 R4420600 R4421600;
array RFD      R4408800 R4409900 R4411000 R4412100 R4413200 R4414300 R4415400 R4416500
               R4417600 R4418700 R4419700 R4420700 R4421700;

do over RFD;
   if STK=2 and RFD=. then RFD=15;
   if STK>0 and RFM=. then RFM=01;
   if REF4<1970 and SORTY>1970 then REF4=SORTY;
   if SORTM=2 and SORTD>28 then SORTD=28;
```

```
    if (SORTM=4 or SORTM=6 or SORTM=9 or SORTM=11) and SORTD=31 then SORTD=30;
    if SORTM=0 then SORTM=.;
    if SORTD=0 then SORTD=.;
    if SORTY>1900 then SORTY=SORTY-1900;
    ENDT=MDY(SORTM,SORTD,SORTY);
    if ENDT>HIDAT then HIDAT=ENDT;
 end;

   CUR=0; NCUR=0;
   if ENDT01 ge DOLIDAT and ENDT01=HIDAT then do;
      R4430100=1; if ENDT01=DOIDAT then CUR=CUR+1;
      else if ENDT01<DOIDAT then NCUR=NCUR+1; end;

   if ENDT02 ge DOLIDAT and ENDT02=HIDAT then do;
      R4430800=1; if ENDT02=DOIDAT then CUR=CUR+1;
      else if ENDT02<DOIDAT then NCUR=NCUR+1; end;

   if ENDT03 ge DOLIDAT and ENDT03=HIDAT then do;
      R4431500=1; if ENDT03=DOIDAT then CUR=CUR+1;
      else if ENDT03<DOIDAT then NCUR=NCUR+1; end;

   if ENDT04 ge DOLIDAT and ENDT04=HIDAT then do;
      R4432200=1; if ENDT04=DOIDAT then CUR=CUR+1;
      else if ENDT04<DOIDAT then NCUR=NCUR+1; end;

   if ENDT05 ge DOLIDAT and ENDT05=HIDAT then do;
      R4432900=1; if ENDT05=DOIDAT then CUR=CUR+1;
      else if ENDT05<DOIDAT then NCUR=NCUR+1; end;

   if ENDT06 ge DOLIDAT and ENDT06=HIDAT then do;
      R4433600=1; if ENDT06=DOIDAT then CUR=CUR+1;
      else if ENDT06<DOIDAT then NCUR=NCUR+1; end;

   if ENDT07 ge DOLIDAT and ENDT07=HIDAT then do;
      R4434300=1; if ENDT07=DOIDAT then CUR=CUR+1;
      else if ENDT07<DOIDAT then NCUR=NCUR+1; end;

   if ENDT08 ge DOLIDAT and ENDT08=HIDAT then do;
      R4435000=1; if ENDT08=DOIDAT then CUR=CUR+1;
      else if ENDT08<DOIDAT then NCUR=NCUR+1; end;

   if ENDT09 ge DOLIDAT and ENDT09=HIDAT then do;
      R4435700=1; if ENDT09=DOIDAT then CUR=CUR+1;
      else if ENDT09<DOIDAT then NCUR=NCUR+1; end;

   if ENDT10 ge DOLIDAT and ENDT10=HIDAT then do;
      R4436400=1; if ENDT10=DOIDAT then CUR=CUR+1;
      else if ENDT10<DOIDAT then NCUR=NCUR+1; end;

   if ENDT11 ge DOLIDAT and ENDT11=HIDAT then do;
      R4437100=1; if ENDT11=DOIDAT then CUR=CUR+1;
      else if ENDT11<DOIDAT then NCUR=NCUR+1; end;

   if ENDT12 ge DOLIDAT and ENDT12=HIDAT then do;
      R4437800=1; if ENDT12=DOIDAT then CUR=CUR+1;
      else if ENDT12<DOIDAT then NCUR=NCUR+1; end;
```

```
if ENDT13 ge DOLIDAT and ENDT13=HIDAT then do;
   CPS13=1; if ENDT13=DOIDAT then CUR=CUR+1;
   else if ENDT13<DOIDAT then NCUR=NCUR+1; end;

if (CUR>1 or (CUR=0 and NCUR>1)) then do;
array CP    R4430100 R4430800 R4431500 R4432200 R4432900 R4433600 R4434300 R4435000 R4435700
            R4436400 R4437100 R4437800 CPS13;
array LN    R4430000 R4430700 R4431400 R4432100 R4432800 R4433500 R4434200 R4434900 R4435600
            R4436300 R4437000 R4437700 R4438400;
   do over CP; if LN ne 1 then CP=.;
   end;
end;


if CUR ge 1 then R4428100=1; else if CUR=0 and NCUR ge 1 then R4428100=2; else R4428100=3;
if CUR=1 then R4428200=1; else if CUR>1 then R4428200=2;
if CUR=0 and NCUR=1 then R4428300=1; else if CUR=0 and NCUR>1 then R4428300=2;
if R4428100=1 or R4428100=2 then R4428500=1; else R4428500=0;
```

/*14. Create the Weeks Working, Unemployed, and out of the Labor Force for the Respondent*/

/* Note. technique allows repeats of job dates without double counting. Working takes precedence, then
        looking, then out of labor force.*/
/* Layout wks from earliest DOLI(1979) to last DOI(1999) then find DOI and DOLI each R on timeline. Any
        wk greater or equal DOLI and less or equal DOI set to zero. Count these wks to find wks between
        interview dates. Use date function find start wk and end wk for each job active since DOLI, then count
        wks from beginning to find place of start and end wks on time line. Set to one any wk greater or equal
        DOLI and less or equal DOI and greater or equal job start wk and less or equal job end wk. Initialize
        total wks working, unemployed and out of labor force to zero. From weeks layout count each wk=1 to
        get weeks working. If WORK less than WBID look for wks looking or on layoff. Check each period
        from DOLI to first job start, then from first job end to second job start etc. */
/* first do gaps where all weeks are looking or on layoff. Find DOLI to first start then add to WUMP all wks this
        gap if dont know or refuse set to D or R. Then do end of position 1 and start of position 2 job etc using
        the rostered item wks looking. Then if WORK+WUMP lt WBID, look at some wks looking and add
        number looking to WUMP. Do this for the lastgap and for each rostered gap if D or R set WUMP to D
        or R. Set a ceiling for WUMP at WBID less WORK. If WORK and WUMP less than WBID set
        remaining weeks to WOLF. If WUMP set to D or R, set WOLF to same*/

```
CSTDAT=MDY(01,01,1979);
DLICW=INTCK('WEEK',CSTDAT,DOLIDAT);
DOICW=INTCK('WEEK',CSTDAT,DOIDAT);
R4267200=0;
array W     (l)  W0001- W1092;
do l=1 to 1092;
if DLICW le L le DOICW then do; W=0; R4267200=R4267200+1; End;
end;


array YS    (k) R4408900 R4410000 R4411100 R4412200 R4413300 R4414400 R4415500 R4416600
            R4417700 R4418800 R4419800 R4420800 R4421800;
array DS    (k) R4408800 R4409900 R4411000 R4412100 R4413200 R4414300 R4415400 R4416500
            R4417600 R4418700 R4419700 R4420700 R4421700;
array MS    (k) R4408700 R4409800 R4410900 R4412000 R4413100 R4414200 R4415300 R4416400
            R4417500 R4418600 R4419600 R4420600 R4421600;
array YE    (k) R4429700 R4430400 R4431100 R4431800 R4432500 R4433200 R4433900 R4434600
            R4435300 R4436000 R4436700 R4437400 R4438100;
```

```
array DE      (k) R4429900 R4430600 R4431300 R4432000 R4432700 R4433400 R4434100 R4434800
              R4435500 R4436200 R4436900 R4437600 R4438300;
array ME      (k) R4429800 R4430500 R4431200 R4431900 R4432600 R4433300 R4434000 R4434700
              R4435400 R4436100 R4436800 R4437500 R4438200;
array ESTI    (k) R4408200 R4409300 R4410400 R4411500 R4412600 R4413700 R4414800 R4415900
              R4417000 R4418100 R4419200 R4420200 R4421200;
array STDAT (k) STDAT01-STDAT13;
array ENDAT(k) ENDAT01-ENDAT13;
array STCW  (k) STCW01-STCW13;
array ENCW  (k) ENCW01-ENCW13;

do k=1 to 13;
  if ESTI=1 then do;
    YS=YS-1900;
    STDAT=MDY(MS,DS,YS);
    ENDAT=MDY(ME,DE,YE);
    STCW=INTCK('WEEK',CSTDAT,STDAT);
    ENCW=INTCK('WEEK',CSTDAT,ENDAT);

    C=0;
    Do L=1 to 1092; C=C+1;
        if DLICW le L le DOICW then do;
            if stcw>0 and STCW le C le ENCW then do;
                if 1 le k le 13 then do; w=1; end;
            end;
        end;
    end;
  end;
end;

R4438500=0; R4438600=0; R4438700=0;
do L=1 to 1092;
    if 001 le L le 1092 then do;
        if w=1 then R4438500=R4438500+1;
end;
end;

if R4438500<R4267200 then do;
    do L=1 to 1092;
    if 001 le L le 1092 then do;
        if W=0 then do;
        if DLICW le L le STCW01 then do;
            if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
            else if -2 le R4734400 le -1 then R4438600=R4734400; end;
        if DLICW le L le STCW02 and STCW01=. then do;
            if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
            else if -2 le R4734400 le -1 then R4438600=R4734400; end;
        if DLICW le L le STCW03 and STCW01=. and STCW02=. then do;
            if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
            else if -2 le R4734400 le -1 then R4438600=R4734400; end;
        if DLICW le L le STCW04 and STCW01=. and STCW02=. and STCW03=. then do;
            if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
            else if -2 le R4734400 le -1 then R4438600=R4734400; end;
        if DLICW le L le STCW05 and STCW01=. and STCW02=. and STCW03=. and STCW04=. then do;
            if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
            else if -2 le R4734400 le -1 then R4438600=R4734400; end;
```

```
if DLICW le L le STCW06 and STCW01=. and STCW02=. and STCW03=. and STCW04=. and
    STCW05=. then do;
    if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4734400 le -1 then R4438600=R4734400; end;
if DLICW le L le STCW07 and STCW01=. and STCW02=. and STCW03=. and STCW04=. and
    STCW05=. and STCW06=. then do;
    if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4734400 le -1 then R4438600=R4734400; end;
if DLICW le L le STCW08 and STCW01=. and STCW02=. and STCW03=. and STCW04=. and
    STCW05=. and STCW06=. and STCW07=. then do;
    if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4734400 le -1 then R4438600=R4734400; end;
if DLICW le L le STCW09 and STCW01=. and STCW02=. and STCW03=. and STCW04=. and
    STCW05=. and STCW06=. and STCW07=. and STCW08=. then do;
    if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4734400 le -1 then R4438600=R4734400; end;
if DLICW le L le  DOICW and STCW01=. and STCW02=. and STCW03=. and STCW04=. and
    STCW05=. and STCW06=. and STCW07=. and STCW08=. and STCW09=. then do;
    if R4734400=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4734400 le -1 then R4438600=R4734400; end;


if ENCW01 le L le STCW02 then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW03 and STCW02=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW04 and STCW02=. and STCW03=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW05 and STCW02=. and STCW03=. and STCW04=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW06 and STCW02=. and STCW03=. and STCW04=. and STCW05=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW07 and STCW02=. and STCW03=. and STCW04=. and STCW05=. and
    STCW06=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW08 and STCW02=. and STCW03=. and STCW04=. and STCW05=. and
    STCW06=. and STCW07=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le STCW09 and STCW02=. and STCW03=. and STCW04=. and STCW05=. and
    STCW06=. and STCW07=. and STCW08=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;
if ENCW01 le L le  DOICW and STCW02=. and STCW03=. and STCW04=. and STCW05=. and
    STCW06=. and STCW07=. and STCW08=. and STCW09=. then do;
    if R4729800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4729800 le -1 then R4438600=R4729800; end;


if ENCW02 le L le STCW03 then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
```

if ENCW02 le L le STCW04 and STCW03=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
if ENCW02 le L le STCW05 and STCW03=. and STCW04=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
if ENCW02 le L le STCW06 and STCW03=. and STCW04=. and STCW05=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
if ENCW02 le L le STCW07 and STCW03=. and STCW04=. and STCW05=. and STCW06=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
if ENCW02 le L le STCW08 and STCW03=. and STCW04=. and STCW05=. and STCW06=. and
    STCW07=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
if ENCW02 le L le STCW09 and STCW03=. and STCW04=. and STCW05=. and STCW06=. and
    STCW07=. and STCW08=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;
if ENCW02 le L le DOICW and STCW03=. and STCW04=. and STCW05=. and STCW06=. and
    STCW07=. and STCW08=. and STCW09=. then do;
    if R4730300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730300 le -1 then R4438600=R4730300; end;

if ENCW03 le L le STCW04 then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;
if ENCW03 le L le STCW05 and STCW04=. then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;
if ENCW03 le L le STCW06 and STCW04=. and STCW05=. then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;
if ENCW03 le L le STCW07 and STCW04=. and STCW05=. and STCW06=. then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;
if ENCW03 le L le STCW08 and STCW04=. and STCW05=. and STCW06=. and STCW07=. then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;
if ENCW03 le L le STCW09 and STCW04=. and STCW05=. and STCW06=. and STCW07=. and
    STCW08=. then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;
if ENCW03 le L le DOICW and STCW04=. and STCW05=. and STCW06=. and STCW07=. and
    STCW08=. and STCW09=. then do;
    if R4730800=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4730800 le -1 then R4438600=R4730800; end;

if ENCW04 le L le STCW05 then do;
    if R4731300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4731300 le -1 then R4438600=R4731300; end;
if ENCW04 le L le STCW06 and STCW05=. then do;
    if R4731300=1 and R4438600 ge 0 then R4438600=R4438600+1;
    else if -2 le R4731300 le -1 then R4438600=R4731300; end;

```
if ENCW04 le L le STCW07 and STCW05=. and STCW06=. then do;
   if R4731300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731300 le -1 then R4438600=R4731300; end;
if ENCW04 le L le STCW08 and STCW05=. and STCW06=. and STCW07=. then do;
   if R4731300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731300 le -1 then R4438600=R4731300; end;
if ENCW04 le L le STCW09 and STCW05=. and STCW06=. and STCW07=. and STCW08=. then do;
   if R4731300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731300 le -1 then R4438600=R4731300; end;
if ENCW04 le L le  DOICW and STCW05=. and STCW06=. and STCW07=. and STCW08=. and
     STCW09=. then do;
   if R4731300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731300 le -1 then R4438600=R4731300; end;

if ENCW05 le L le STCW06 then do;
   if R4731800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731800 le -1 then R4438600=R4731800; end;
if ENCW05 le L le STCW07 and STCW06=. then do;
   if R4731800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731800 le -1 then R4438600=R4731800; end;
if ENCW05 le L le STCW08 and STCW06=. and STCW07=. then do;
   if R4731800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731800 le -1 then R4438600=R4731800; end;
if ENCW05 le L le STCW09 and STCW06=. and STCW07=. and STCW08=. then do;
   if R4731800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731800 le -1 then R4438600=R4731800; end;
if ENCW05 le L le  DOICW and STCW06=. and STCW07=. and STCW08=. and STCW09=. then do;
   if R4731800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4731800 le -1 then R4438600=R4731800; end;

if ENCW06 le L le STCW07 then do;
   if R4732300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732300 le -1 then R4438600=R4732300; end;
if ENCW06 le L le STCW08 and STCW07=. then do;
   if R4732300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732300 le -1 then R4438600=R4732300; end;
if ENCW06 le L le STCW09 and STCW07=. and STCW08=. then do;
   if R4732300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732300 le -1 then R4438600=R4732300; end;
if ENCW06 le L le  DOICW and STCW07=. and STCW08=. and STCW09=. then do;
   if R4732300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732300 le -1 then R4438600=R4732300; end;

if ENCW07 le L le STCW08 then do;
   if R4732800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732800 le -1 then R4438600=R4732800; end;
if ENCW07 le L le STCW09 and STCW08=. then do;
   if R4732800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732800 le -1 then R4438600=R4732800; end;
if ENCW07 le L le  DOICW and STCW08=. and STCW09=. then do;
   if R4732800=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4732800 le -1 then R4438600=R4732800; end;

if ENCW08 le L le STCW09 and R4733300=1 then do;
   if R4733300=1 and R4438600 ge 0 then R4438600=R4438600+1;
   else if -2 le R4733300 le -1 then R4438600=R4733300; end;
```

```
        if ENCW08 le L le  DOICW and R4733300=1 and STCW09=. then do;
            if R4733300=1 and R4438600 ge 0 then R4438600=R4438600+1;
            else if -2 le R4733300 le -1 then R4438600=R4733300; end;
end;
end;
end;
end;


if R4438600 ge 0 and R4438500+R4438600<R4267200 then do;
array SNA         R4734400 R4729800 R4730300 R4730800 R4731300 R4731800 R4732300 R4732800
R4733300 R4733600;
array WK          R4734500 R4729900 R4730400 R4730900 R4731400 R4731900 R4732400 R4732900;
    do over SNA;
      if SNA=2 and WK>0 and R4438600 ge 0 then R4438600=R4438600+WK;
      if SNA=2 and -2 le WK le -1 then R4438600=WK;
    end; end;


if R4438600>R4267200-R4438500 then R4438600=R4267200-R4438500;
if R4438500 ge 0 and R4438600 ge 0 then R4438700=R4267200-(R4438500+R4438600);
if R4438600<0 then R4438700=R4438600;
end;
```

/* 15. New Created Variables: Marriage Start and End Dates */

```
/*eg from young women cohort*/
/*Look at all surveys from beginning, make century consistent in dates.  Some surveys report year of last
        marriage, next to last, year last marriage end.  Others report any change to marital status and date of
        change.  In latter case assign end date if status is married/separated to widowed or divorced and assign
        start date if status is single, widowed, or divorced to married.  Assign no date if reported change is
        inconsistent with prior status.  If month of reported status change is unknown assign 06 (June).  Once
        start and end dates determined assign missing codes*/

array yr R1609700 R1609900 R1610100 R3505500 R3505900 R3506300 R3506700 R4326600 R4327300;
        Do over yr; if yr>1900 then yr=yr-1900; end;
    STSTAT01=R0086200;
     if R0086200=. and R0003300>0 then STSTAT01=R0003300;
     if R0086200=R0146100 then STSTAT01=R0003300;
    array STSTAT    STSTAT01 ststat08-ststat34;
    array stat      R0146100 stat08-stat34;
    array YC        R0195300 R0959000 R0959400 R0959800 R0960200 R1078700 R1079100 R1079500
                    R1204000 R1204400 R1204800 R1205200 R1336600 R1337000 R1337400 R1337800
                    R1435800 R1436200 R1436600 R1609700 R1609900 R1610100 R3505500 R3505900
                    R3506300 R3506700 R4326600 R4327300;
    array MC        R0195200 R0958900 R0959300 R0959700 R0960100 R1078600 R1079000 R1079400
                    R1203900 R1204300 R1204700 R1205100 R1336700 R1337100 R1337500 R1337900
                    R1435700 R1436100 R1436500 R1609600 R1609800 R1610000 R3505400 R3505800
                    R3506200 R3506600 R4326500 R4327200;
    array typc      TYPC01 R0958800 R0959200 R0959600 R0960000 R1078500 R1078900 R1079300
                    R1203800 R1204200 R1204600 R1205000 R1336500 R1336900 R1337300 R1337700
                    R1435600 R1436000 R1436400 R1609200 R1609300 R1609400 R3505300 R3505700
                    R3506100 R3506500 R4326400 R4327100;
    array ye        YLME01 ylme08-ylme34;
    array me        MLME01 mlme08-mlme34;
    array ym        YLM01 ylm08  -ylm34;
    array mm        MLM01 mlm08  -mlm34;
do over yc; if YC>0 then do;
```

```
if mc=. or mc<0 then mc=06;
if (STSTAT=1 or STSTAT=2 or STSTAT=5 and 3 le STAT le 4) then do; YE=YC; ME=MC; end;
if ((3 le STSTAT le 4 or STSTAT=6) and (1 le STAT le 2)) then do; YM=YC; MM=MC; end;
if typc=1 or typc=5 or typc=6 then do; Ym = YC; Mm = MC; end;
if 2 le typc le 3 then do; Ye = YC; Me = MC; end; end; End;
array Ylm        (k) YLM01 R0678900 R0677900 R0676700 R0912500 R0911400 R0910000 ylm08-ylm34;
array Mlm        (k) MLM01 R0678800 R0677800 R0676600 R0912400 R0911300 R0909900 mlm08-mlm34;
array Ylme       (k) YLME01 R0679500 R0678500 R0677400 ylme05 R0912100 R0910900 ylme08-ylme34;
array Mlme       (k) MLME01 R0679400 R0678400 R0677300 mlme05 R0912000 R0910800 mlme08-
mlme34;
R5438300=.; R5438400=.; R5438500=.; R5438600=.; R5438700=.; R5438800=.; R5438900=.; R5439000=.;
R5439100=.; R5439200=.;
do k=1 to 34;
    if YLM>0 and MLM=. then MLM=06; if YLME>0 and MLME=. then MLME=06;
    if Ylm>0 and Mlm>0 then do; STDAT=Ylm*100+MLM;
        if R5438400=. and R5438300=. then R5438300=stdat;
        if (R5438400 ne . and R5438500=. and STDAT>R5438400) then R5438500=stdat;
        if (R5438600 ne . and R5438700=. and stdat>R5438600) then R5438700=STDAT;
        if (R5438800 ne . and R5438900=. and stdat>R5438800) then R5438900=stdat;
        if (R5439000 ne . and R5439100=. and stdat>R5439000) then R5439100=stdat;
    end;
    if YlmE>0 and MlmE>0 then do; ENDAT=YlmE*100+MLME;
        If R5438400=. and ENDAT>R5438300 then R5438400=endat;
        If R5438500 ne . and R5438600=. And endat >R5438500 then R5438600=endat;
        If R5438700 ne . and R5438800=. and endat>R5438700 then R5438800=endat;
        If R5438900 ne . and R5439000=. and endat>R5438900 then R5439000=endat;
        If R5439100 ne . and R5439200=. and endat>R5439100 then R5439200=endat;
    end; end;
Single=1;
array mar    R0003300 R0086200 R0146100 R0253300 R0334400 R0417200 R0519600 R0581000 R0587500
             R0710000 R0756500 R0940700 R0947400 R1063000 R1221000 R1352700 R1569900 R1664710
             R3507200 R4278200;
do over mar; if 1 le mar le 5 then single=0; end;
if R5439100>0 and R5439200=. then do;
    if R5438300=. then R5438300=0; R5439200=-990; end;
if R5439000>0 and R5439100=. then do;
    if R5438300=. then R5438300=0; R5439100=-991; R5439200=-991; end;
if R5438900>0 and R5439000=. then do;
    if R5438300=. then R5438300=0; R5439000=-992; R5439100=-992; R5439200=-992; end;
if R5438800>0 and R5438900=. then do;
    if R5438300=. then R5438300=0; R5438900=-993; R5439000=-993; R5439100=-993; R5439200=-993;
        end;
if R5438700>0 and R5438800=. then do;
    if R5438300=. then R5438300=0; R5438800=-994; R5438900=-994; R5439000=-994; R5439100=-994;
        R5439200=-994; end;
if R5438600>0 and R5438700=. then do;
    if R5438300=. then R5438300=0; R5438700=-995; R5438800=-995; R5438900=-995; R5439000=-995;
        R5439100=-995; R5439200=-995; end;
if R5438500>0 and R5438600=. then do;
    if R5438300=. then R5438300=0; R5438600=-996; R5438700=-996; R5438800=-996; R5438900=-996;
        R5439000=-996; R5439100=-996; R5439200=-996; end;
if R5438400>0 and R5438500=. then do;
    if R5438300=. then R5438300=0; R5438500=-997; R5438600=-997; R5438700=-997; R5438800=-997;
        R5438900=-997; R5439000=-997; R5439100=-997; R5439200=-997; end;
if R5438300>0 and R5438400=. then do;
```

```
        R5438400=-998; R5438500=-998; R5438600=-998; R5438700=-998; R5438800=-998; R5438900=-998;
            R5439000=-998; R5439100=-998; R5439200=-998; end;
    if R5438300=. and R5438400=. and single=1 then do;
        R5438300=-999; R5438400=-999; R5438500=-999; R5438600=-999; R5438700=-999; R5438800=-999;
            R5438900=-999; R5439000=-999; R5439100=-999; R5439200=-999; end;
    if R5438300=. and R5438400=. and single=0 then do;
        R5438300=0; R5438400=0; R5438500=0; R5438600=0; R5438700=0; R5438800=0; R5438900=0;
            R5439000=0; R5439100=0; R5439200=0; end;
```

/*16. New Created Variables: Calendar Week Labor Force Status */

```
/*e.g. using data from 1999 survey */
/*Fill in the unknown days and months per specifications.  Set up a time line of weeks from January 1994 to
        present.  Initialize WORK to "0" for weeks covered by interview period.  Set weeks between job start
        and end dates to "1."  Count weeks worked by calendar year for 94, 95, 96, 97, 98, 99. */

if 201 le R4266900 le 205 then do;
    DOLIDAT3=MDY(R4253800,R4253900,R4254000);
    DOIDAT3=MDY(R4260000,R4260100,99);
end;
CSTDAT=MDY(01,01,94);
DLICW=INTCK('WEEK',CSTDAT,DOLIDAT3);
DOICW=INTCK('WEEK',CSTDAT,DOIDAT3);
array STK    R4409000 R4410100 R4411200 R4412300 R4413400 R4414500 R4415600 R4416700 R4417800
            R4418900 R4419900 R4420900 R4421900;
array RFM    R4408700 R4409800 R4410900 R4412000 R4413100 R4414200 R4415300 R4416400 R4417500
            R4418600 R4419600 R4420600 R4421600;
array RFD    R4408800 R4409900 R4411000 R4412100 R4413200 R4414300 R4415400 R4416500 R4417600
            R4418700 R4419700 R4420700 R4421700;
do over RFD;
    if STK=2 and RFD=. then RFD=15;
    if STK>0 and RFM=. then RFM=01;
end;
array W      (l) R54077-R54128 R54130-R54181 R54183-R54234 R54236-R54287 R54289-R54340
            R54342-R54381;
do L=1 to 300;
if DLICW ne . and DLICW le L le DOICW then do; W=0; end;
end;
array YS     (j) R4408900 R4410000 R4411100 R4412200 R4413300 R4414400 R4415500 R4416600
            R4417700 R4418800 R4419800 R4420800 R4421800;
array DS     (j) R4408800 R4409900 R4411000 R4412100 R4413200 R4414300 R4415400 R4416500
            R4417600 R4418700 R4419700 R4420700 R4421700;
array MS     (j) R4408700 R4409800 R4410900 R4412000 R4413100 R4414200 R4415300 R4416400
            R4417500 R4418600 R4419600 R4420600 R4421600;
array YE     (j) R4429700 R4430400 R4431100 R4431800 R4432500 R4433200 R4433900 R4434600
            R4435300 R4436000 R4436700 R4437400 R4438100;
array DE     (j) R4429900 R4430600 R4431300 R4432000 R4432700 R4433400 R4434100 R4434800
            R4435500 R4436200 R4436900 R4437600 R4438300;
array ME     (j) R4429800 R4430500 R4431200 R4431900 R4432600 R4433300 R4434000 R4434700
            R4435400 R4436100 R4436800 R4437500 R4438200;
array ESTI   (j) R4408200 R4409300 R4410400 R4411500 R4412600 R4413700 R4414800 R4415900
            R4417000 R4418100 R4419200 R4420200 R4421200;
array STDAT (j) STDAT01-STDAT13;
array ENDAT (j) ENDAT01-ENDAT13;
```

```
array STCW   (j)  STCW01-STCW13;
array ENCW   (j)  ENCW01-ENCW13;
    do J=01 TO 13;
    if ye=99 then ye=.; if me>12 then me=01; if de>31 then de=15;
    if ys=99 then ys=.;
        if (-2 le ms le -1 or ms>12) then ms=01;
        if (-2 le ds le -1 or ds>31) then ds=15;
    if ME=2 and DE>28 then DE=28;
    if (ME=4 or ME=6 or ME=9 or ME=11) and DE=31 then DE=30;
    if ye=0 then ye=.;
    if ME=0 then ME=.;
    if DE=0 then DE=.;
    if MS=2 and DS>28 then DS=28;
    if (MS=4 or MS=6 or MS=9 or MS=11) and DS=31 then DS=30;
    if YS>1900 then YS=YS-1900;
    if ye>1900 then ye=ye-1900;
    if ESTI=1 then do;
      STDAT=MDY(MS,DS,YS);
      ENDAT=MDY(ME,DE,YE);
      STCW=INTCK('WEEK',CSTDAT,STDAT);
      ENCW=INTCK('WEEK',CSTDAT,ENDAT);
    C=0;
    Do L=1 to 300; C=C+1;
        if DLICW le L le DOICW then do;
            if stcw ne . and STCW le C le ENCW then do; w=1; end;
        end;
      end;
    end;
   end;
N99=0; N98=0; N97=0; N96=0; N95=0; N94=0;
do L=1 to 300;
if 1 le l le 52 then do;
if W ne . then do; N94=N94+1;
if N94=1 then WRK94=W;
if N94>1 then WRK94=WRK94+W; end;
end;
if 53 le l le 104 then do;
if W ne . then do; N95=N95+1;
if N95=1 then WRK95=W;
if N95>1 then WRK95=WRK95+W; end;
end;
if 105 le L le 156 then do;
if W ne . then do; N96=N96+1;
if N96=1 then WRK96=W;
if N96>1 then WRK96=WRK96+W; end;
end;
if 157 le L le 208 then do;
if W ne . then do; N97=N97+1;
if N97=1 then WRK97=W;
if N97>1 then WRK97=WRK97+W; end;
end;
if 209 le L le 260 then do;
if W ne . then do; N98=N98+1;
if N98=1 then R5434100=W;
if N98>1 then R5434100=R5434100+W; end;
end;
```

```
if 261 le L le 300 then do;
if W ne . then do; N99=N99+1;
if N99=1 then R5438200=W;
if N99>1 then R5438200=R5438200+W; end;
end;
end;
```

/*average weeks worked per year for those completing all surveys and number of active employers reported 95, 97, and 99 */
/*eg for mature women cohort*/
/*Count completed interviews and add weeks worked for completed interviews. For avg weeks worked divide
        total weeks worked by number of surveys. Set avg to missing for all except those with all surveys
        completed round to whole, then count different ecodes among active employers for 95, 97, and 99. Eg.,
        for ecodes up to 9 , self employer counts as a single employer */

```
nsurveys=0;  cumww=0;
        array totalm      R0000200 R0085610 R0088410 R0133810 R0205310 R0288310 R0308410 R0329510
                          R0456510 R0491210 R0528410 R0666420 R0721520 R0783320 R0887820 R1009320
                          R1601400 R3498500 R4267100;
        array wworkm      R0017100 R0086500 R0095000 R0199200 R0283200 R0292000 R0312300 R0452200
                          R0465000 R0496500 R0663500 R0674400 R0733300 R0885100 R1006500 R1301600
                          R2550200 R3636800 R4438500;
        do over totalm;
        if totalm>0 then nsurveys=nsurveys+1;
        if totalm>0 and wworkm<0 then wworkm=0;
        if 0 lt wworkm lt 1 then wworkm=1;
        if wworkm>52 then wworkm=52;
        if totalm>0 and wworkm ge 0 then cumww=cumww+wworkm;
        R5439300=ROUND (cumww/nsurveys,1);
        end;
        if nsurveys<19 then R5439300=-5;

        array RSW       R1601400 R3498500 R4267100;
        array RCW       R1819900 R3638000 R4440400;
        array Rew       R1815300 R3633500 R4429600;
        array RCWB      R1820000 R3658700 R4462700;
        array Rewb      R1815400 R3633700 R4430300;
        array Rcwc      R1820100 R3678200 R4482700;
        array Rewc      R1815500 R3633900 R4431000;
        array Rcwd      R1820200 R3698200 R4502700;
        array Rewd      R1815600 R3634100 R4431700;
        array Rcwe      R1820300 R3717500 R4521900;
        array Rewe      R1815700 R3634300 R4432400;
        array Rcwf      R1820400 R3736100 R4540800;
        array Rewf      R1815800 R3634500 R4433100;
        array Rcwg      R1820500 R3752600 R4558700;
        array Rewg      R1815900 R3634700 R4433800;
        array Rcwh      R1820600 R3768800 R4575900;
        array REWH      R1816000 R3634900 R4434500;
        array Rcwi      R1820700 R3784800 R4591600;
        array REWI      R1816100 R3635100 R4435200;
        array Rcwj      cwj1       R3795500 R4605500;
        array REWJ      EWJ1       R3635300 R4435900;
        array Rcwk      cwk1       R3803500 R4617600;
        array REWK      EWK1       R3635500 R4436600;
        array Rcwl      cwl1       cwl2       R4628400;
```

```
array Rewl        ewl1    ewl2    R4437300;
array Rcwm        cwm1    cwm2    R4636400;
array Rewm        ewm1    ewm2    R4438000;

RNCAPI=0;
 do over RCW;
  if RSW>0 then RNCAPI=RNCAPI+1;
  if RCW ge 0 and REW=0 then EMP0=1;
  if RCW ge 0 and Rew=1 then EMP1=1;
  if RCW ge 0 and Rew=2 then EMP2=1;
  if RCW ge 0 and Rew=3 then EMP3=1;
  if RCW ge 0 and Rew=4 then EMP4=1;
  if RCW ge 0 and Rew=5 then EMP5=1;
  if RCW ge 0 and Rew=6 then EMP6=1;
  if RCW ge 0 and Rew=7 then EMP7=1;
  if RCW ge 0 and Rew=8 then EMP8=1;
  if RCW ge 0 and Rew=9 then EMP9=1;
  if RCWB ge 0 and REWB=0 then EMP0=1;
  if RCWB ge 0 and Rewb=1 then EMP1=1;
  if RCWB ge 0 and Rewb=2 then EMP2=1;
  if RCWB ge 0 and Rewb=3 then EMP3=1;
  if RCWB ge 0 and Rewb=4 then EMP4=1;
  if RCWB ge 0 and Rewb=5 then EMP5=1;
  if RCWB ge 0 and Rewb=6 then EMP6=1;
  if RCWB ge 0 and Rewb=7 then EMP7=1;
  if RCWB ge 0 and Rewb=8 then EMP8=1;
  if RCWB ge 0 and Rewb=9 then EMP9=1;
  if RCWC ge 0 and REWc=0 then EMP0=1;
  if RCWC ge 0 and Rewc=1 then EMP1=1;
  if RCWC ge 0 and Rewc=2 then EMP2=1;
  if RCWC ge 0 and Rewc=3 then EMP3=1;
  if RCWC ge 0 and Rewc=4 then EMP4=1;
  if RCWC ge 0 and Rewc=5 then EMP5=1;
  if RCWC ge 0 and Rewc=6 then EMP6=1;
  if RCWC ge 0 and Rewc=7 then EMP7=1;
  if RCWC ge 0 and Rewc=8 then EMP8=1;
  if RCWC ge 0 and Rewc=9 then EMP9=1;
  if RCWD ge 0 and Rewd=0 then emp0=1;
  if RCWD ge 0 and RewD=1 then EMP1=1;
  if RCWD ge 0 and RewD=2 then EMP2=1;
  if RCWD ge 0 and RewD=3 then EMP3=1;
  if RCWD ge 0 and RewD=4 then EMP4=1;
  if RCWD ge 0 and RewD=5 then EMP5=1;
  if RCWD ge 0 and RewD=6 then EMP6=1;
  if RCWD ge 0 and RewD=7 then EMP7=1;
  if RCWD ge 0 and RewD=8 then EMP8=1;
  if RCWD ge 0 and RewD=9 then EMP9=1;
  if Rcwe ge 0 and Rewe=0 then emp0=1;
  if RCWE ge 0 and RewE=1 then EMP1=1;
  if RCWE ge 0 and RewE=2 then EMP2=1;
  if RCWE ge 0 and RewE=3 then EMP3=1;
  if RCWE ge 0 and RewE=4 then EMP4=1;
  if RCWE ge 0 and RewE=5 then EMP5=1;
  if RCWE ge 0 and RewE=6 then EMP6=1;
  if RCWE ge 0 and RewE=7 then EMP7=1;
  if RCWE ge 0 and RewE=8 then EMP8=1;
```

```
if RCWE ge 0 and RewE=9 then EMP9=1;
if Rcwf ge 0 and Rewf=0 then emp0=1;
if RCWF ge 0 and RewF=1 then EMP1=1;
if RCWF ge 0 and RewF=2 then EMP2=1;
if RCWF ge 0 and RewF=3 then EMP3=1;
if RCWF ge 0 and RewF=4 then EMP4=1;
if RCWF ge 0 and RewF=5 then EMP5=1;
if RCWF ge 0 and RewF=6 then EMP6=1;
if RCWF ge 0 and RewF=7 then EMP7=1;
if RCWF ge 0 and RewF=8 then EMP8=1;
if RCWF ge 0 and RewF=9 then EMP9=1;
if Rcwg ge 0 and Rewg=0 then emp0=1;
if RCWG ge 0 and RewG=1 then EMP1=1;
if RCWG ge 0 and RewG=2 then EMP2=1;
if RCWG ge 0 and RewG=3 then EMP3=1;
if RCWG ge 0 and RewG=4 then EMP4=1;
if RCWG ge 0 and RewG=5 then EMP5=1;
if RCWG ge 0 and RewG=6 then EMP6=1;
if RCWG ge 0 and RewG=7 then EMP7=1;
if RCWG ge 0 and RewG=8 then EMP8=1;
if RCWG ge 0 and RewG=9 then EMP9=1;
if Rcwh ge 0 and Rewh=0 then emp0=1;
if RCWh ge 0 and Rewh=1 then EMP1=1;
if RCWh ge 0 and Rewh=2 then EMP2=1;
if RCWh ge 0 and Rewh=3 then EMP3=1;
if RCWh ge 0 and Rewh=4 then EMP4=1;
if RCWh ge 0 and Rewh=5 then EMP5=1;
if RCWh ge 0 and Rewh=6 then EMP6=1;
if RCWh ge 0 and Rewh=7 then EMP7=1;
if RCWh ge 0 and Rewh=8 then EMP8=1;
if RCWh ge 0 and Rewh=9 then EMP9=1;
if Rcwi ge 0 and Rewi=0 then emp0=1;
if RCWi ge 0 and Rewi=1 then EMP1=1;
if RCWi ge 0 and Rewi=2 then EMP2=1;
if RCWi ge 0 and Rewi=3 then EMP3=1;
if RCWi ge 0 and Rewi=4 then EMP4=1;
if RCWi ge 0 and Rewi=5 then EMP5=1;
if RCWi ge 0 and Rewi=6 then EMP6=1;
if RCWi ge 0 and Rewi=7 then EMP7=1;
if RCWi ge 0 and Rewi=8 then EMP8=1;
if RCWi ge 0 and Rewi=9 then EMP9=1;
if Rcwj ge 0 and Rewj=0 then emp0=1;
if RCWj ge 0 and Rewj=1 then EMP1=1;
if RCWj ge 0 and Rewj=2 then EMP2=1;
if RCWj ge 0 and Rewj=3 then EMP3=1;
if RCWj ge 0 and Rewj=4 then EMP4=1;
if RCWj ge 0 and Rewj=5 then EMP5=1;
if RCWj ge 0 and Rewj=6 then EMP6=1;
if RCWj ge 0 and Rewj=7 then EMP7=1;
if RCWj ge 0 and Rewj=8 then EMP8=1;
if RCWJ ge 0 and Rewj=9 then EMP9=1;
if RcwK ge 0 and RewK=0 then emp0=1;
if RCWK ge 0 and RewK=1 then EMP1=1;
if RCWK ge 0 and RewK=2 then EMP2=1;
if RCWK ge 0 and RewK=3 then EMP3=1;
if RCWK ge 0 and RewK=4 then EMP4=1;
```

```
        if RCWK ge 0 and RewK=5 then EMP5=1;
        if RCWK ge 0 and RewK=6 then EMP6=1;
        if RCWK ge 0 and RewK=7 then EMP7=1;
        if RCWK ge 0 and RewK=8 then EMP8=1;
        if RCWK ge 0 and RewK=9 then EMP9=1;
        if Rcwl ge 0 and Rewl=0 then emp0=1;
        if RCWl ge 0 and Rewl=1 then EMP1=1;
        if RCWl ge 0 and Rewl=2 then EMP2=1;
        if RCWl ge 0 and Rewl=3 then EMP3=1;
        if RCWl ge 0 and Rewl=4 then EMP4=1;
        if RCWl ge 0 and Rewl=5 then EMP5=1;
        if RCWl ge 0 and Rewl=6 then EMP6=1;
        if RCWl ge 0 and Rewl=7 then EMP7=1;
        if RCWl ge 0 and Rewl=8 then EMP8=1;
        if RCWl ge 0 and Rewl=9 then EMP9=1;
        if Rcwm ge 0 and Rewm=0 then emp0=1;
    end;

if RNCAPI=3 then R5439400=0;
array ALLEMP EMP0-EMP9;
do over ALLEMP; if ALLEMP=1 then R5439400=R5439400+1; end;
if R5439400=. then R5439400=-5;
```