
**NLSY97 Appendix 2:
Employment Variable Creation**

INTRODUCTION

A number of the created employment variables use the same program(s) as input. The two programs in this section are referred to throughout the employment variables. Thus, for example, to create the “Weeks Worked at Employee Job #x during 19xx” variables, survey staff first run the program below titled “emp_begin.sas” and then run the program included in the weeks worked section of this appendix.

EMP_BEGIN.SAS

This program calculates total weeks worked at each job for each respondent. It converts start and stop dates for jobs and within-job gaps to continuous week numbers, subtracts within-job gaps, and finally counts the total weeks worked.

Name in Program	Question Name on CD	Name in Program	Question Name on CD
dli_d, dli_m, dli_y	YINF-900_D, _M, _Y	BGDY3_2-BGDY3_4	YEMP-102400.03.02-D-.03.04-D
int_d, int_m, int_y	YINTDATE-D, -M, -Y	BGMO3_2-BGMO3_4	YEMP-102400.03.02-M-.03.04-M
PUBID	PUBID	BGYR3_2-BGYR3_4	YEMP-102400.03.02-Y-.03.04-Y
stardy1-stardy9	YEMP_STARTDATE.01-D-.09-D	BGDY4_2, BGMO4_2, BGYR4_2	YEMP-102400.04.02-D, -M, -Y
starmo1-starmo9	YEMP_STARTDATE.01-M-.09-M	BGDY5_2, BGMO5_2, BGYR5_2	YEMP-102400.05.02-D, -M, -Y
staryr1-staryr9	YEMP_STARTDATE.01-Y-.09-Y	EGDY1_1-EGDY1_9	YEMP-102700.01.01-D-.01.09-D
stopdy1-stopdy9	YEMP_STOPDATE.01-D-.09-D	EGMO1_1-EGMO1_9	YEMP-102700.01.01-M-.01.09-M
stopmo1-stopmo9	YEMP_STOPDATE.01-M-.09-M	EGYR1_1-EGYR1_9	YEMP-102700.01.01-Y-.01.09-Y
stopyr1-stopyr9	YEMP_STOPDATE.01-Y-.09-Y	EGDY2_1-EGDY2_5	YEMP-102700.02.01-D-.02.05-D
UID1-UID9	YEMP_UID.01-.09	EGMO2_1-EGMO2_5	YEMP-102700.02.01-M-.02.05-M
BGDY1_1-BGDY6_1	YEMP-102200.01.01-D-.06.01-D	EGYR2_1-EGYR2_5	YEMP-102700.02.01-Y-.02.05-Y
BGMO1_1-BGMO6_1	YEMP-102200.01.01-M-.06.01-M	EGDY3_1-EGDY3_4	YEMP-102700.03.01-D-.03.04-D
BGYR1_1-BGYR6_1	YEMP-102200.01.01-Y-.06.01-Y	EGMO3_1-EGMO3_4	YEMP-102700.03.01-M-.03.04-M
BGDY1_2-BGDY1_9	YEMP-102400.01.02-D-.01.09-D	EGYR3_1-EGYR3_4	YEMP-102700.03.01-Y-.03.04-Y
BGMO1_2-BGMO1_9	YEMP-102400.01.02-M-.01.09-M	EGDY4_1, EGMO4_1, EGYR4_1	YEMP-102700.04.01-D, -M, -Y
BGYR1_2-BGYR1_9	YEMP-102400.01.02-Y-.01.09-Y	EGDY4_2, EGMO4_2, EGYR4_2	YEMP-102700.04.02-D, -M, -Y
BGDY2_2-BGDY2_5	YEMP-102400.02.02-D-.02.05-D	EGDY5_1, EGMO5_1, EGYR5_1	YEMP-102700.05.01-D, -M, -Y
BGMO2_2-BGMO2_5	YEMP-102400.02.02-M-.02.05-M	EGDY5_2, EGMO5_2, EGYR5_2	YEMP-102700.05.02-D, -M, -Y
BGYR2_2-BGYR2_5	YEMP-102400.02.02-Y-.02.05-Y	EGDY6_1, EGMO6_1, EGYR6_1	YEMP-102700.06.01-D, -M, -Y

******* SECTION 1: The program first converts interview dates into continuous week numbers.*******

```

/* The following are hand edit cases for round 2 int date */
if PUBID=1818 then do; INT_D=2; INT_M=1; INT_Y=1999; end;
if PUBID=5294 then do; INT_D=12; INT_M=12; INT_Y=1998; end;
/* The following are hand edit cases for round 1 int date */
if (pubid=2 or pubid=5615 or pubid=5902) then do; dli_y=1997; end;

/* Convert Interview Date to week number */
/*Convert interview month and day to total days (intdays)*/
if INT_M>0 and INT_D>0 then do;
  if INT_M=1 then intdays=INT_D;
  if INT_M=2 then intdays=INT_D+31;
  if INT_M=3 then intdays=INT_D+59;
  if INT_M=4 then intdays=INT_D+90;
  if INT_M=5 then intdays=INT_D+120;
  if INT_M=6 then intdays=INT_D+151;
  if INT_M=7 then intdays=INT_D+181;
  if INT_M=8 then intdays=INT_D+212;
  if INT_M=9 then intdays=INT_D+243;
  if INT_M=10 then intdays=INT_D+273;
  if INT_M=11 then intdays=INT_D+304;
  if INT_M=12 then intdays=INT_D+334;
end;

/* Convert days into week numbers*/
/* Basic Formula: weekno=endweek{specific year}+ceil[(totdays+{# of days remaining in DEC})/7] */

```

```

/* Default interview week = 9999 */
intwk=9999;
  if INT_Y>0 and intdays>0 then do;
    if INT_Y=1998 then do; INTWK=939+ceil((intdays+4)/7); end;
  end;
  if INT_Y>0 and intdays>0 then do;
    if INT_Y=1999 then do; INTWK=991+ceil((intdays+5)/7); end;
  end;

/* Convert Date of Last Interview to week number */
/* Convert interview month and day to total days (dlidays) */
  if DLI_M>0 and DLI_D>0 then do;
    if DLI_M=1 then dlidays=DLI_D;           if DLI_M=2 then dlidays=DLI_D+31;
    if DLI_M=3 then dlidays=DLI_D+59;       if DLI_M=4 then dlidays=DLI_D+90;
    if DLI_M=5 then dlidays=DLI_D+120;      if DLI_M=6 then dlidays=DLI_D+151;
    if DLI_M=7 then dlidays=DLI_D+181;      if DLI_M=8 then dlidays=DLI_D+212;
    if DLI_M=9 then dlidays=DLI_D+243;      if DLI_M=10 then dlidays=DLI_D+273;
    if DLI_M=11 then dlidays=DLI_D+304;     if DLI_M=12 then dlidays=DLI_D+334;
  end;

/*Convert days into week numbers*/
/* Basic Formula: weekno=endweek{specific year}+ceil[(totdays+{# of days remaining in DEC})/7] */

/* Default interview week = 9999 */
DLIWK=9999;
  if DLI_Y>0 and dlidays>0 then do;
    if DLI_Y=1997 then do; DLIWK=887+ceil((dlidays+3)/7); end;
  end;
  if DLI_Y>0 and dlidays>0 then do;
    if DLI_Y=1998 then do; DLIWK=939+ceil((dlidays+4)/7); end;
  end;

/* Hand edited cases */
if PUBID=471 or PUBID=476 then INTWK=990;
if PUBID=7315 then DLIWK=970;

/* Done to create a seamless flow between rounds */
intwk=intwk-1;

/* SECTION 2: This part converts employment start and stop dates into continuous week numbers.*/

/*Invalid start and stop data lead to imputed values that are marked by dummy variables for invalid start
dates and invalid end dates. The dummy variables uflag1-uflag9 denotes updating of start dates of jobs
worked in both rounds to the Round1 interview date. This is done to create a separate created variable for
Round1 and Round2 which will be added in the end. */

/* Hand edit for respondent with UID8=0. By inspection this is assumed to be mistaken for 9801. */
if pubid=1665 then UID8=9801;

/* ostartm represents the "old" start dates, used when start dates are updated to interview dates. */
array ostartm (i) ostartmo1-ostarmo9;   array ostartd (i) ostartdy1-ostardy9;
array ostarty (i) ostartyr1-ostaryr9;   array ostopm (i) ostopmo1-ostopmo9;

```

Appendix 2: Employment Variable Creation

```
array ostopd (i) ostopdy1-ostopdy9;      array ostopy (i) ostopyr1-ostopyr9;
array startm (i) starmo1-starmo9;      array startd (i) stardy1-stardy9;
array stary (i) staryr1-staryr9;      array stopm (i) stopmo1-stopmo9;
array stopd (i) stopdy1-stopdy9;      array stopy (i) stopyr1-stopyr9;
array UID (i) UID1-UID9;

array sttdays (i) sttday1-sttday9; /* total days in that year from startdate (to Jan 1) */
array stpdays (i) stpday1-stpday9; /* total days in that year from stopdate (to Jan 1) */
array startwk (i) starw1-starw9;
array stopwk (i) stopw1-stopw9;
array srflag (i) srflg1-srflg10;
array spflag (i) spflg1-spflg10;
array uflag (i) uflag1-uflag9; /* uflag=1 when job startdate is updated */
array smofl (i) smofl1-smofl9; /* dummy equals 1 when a start month is imputed */
array emofl (i) emofl1-emofl9; /* dummy equals 1 when a stop month is imputed */

/* Initialize smofl and emofl */
do i=1 to 9; smofl=0; emofl=0; end;

/* Two hand edit cases for respondents who reported jobs in Round1 that lasted up to Round1 interview
date. In Round2, respondents indicated that these jobs ended before Round1 interview date, and therefore
should not be on the Round2 job roster. Therefore, these jobs are erased from the Round2 roster. */
if pubid=2461 then do;
  stardy2=-4; starmo2=-4; staryr2=-4; uid2=-4; stopdy2=-4; stopmo2=-4; stopyr2=-4; end;
if pubid=5276 then do;
  stardy1=-4; starmo1=-4; staryr1=-4; uid1=-4; stopdy1=-4; stopmo1=-4; stopyr1=-4; end;

/* Define old start and stop dates */
do i=1 to 9;
  ostartm=startm;      ostartd=startd;
  ostary=stary;      ostopm=stopm;
  ostopd=stopd;      ostopy=stopy;
end;

/*Fill-in start/stop day for those missing*/
/* flag1 = impute start day (valid month)      flag2 = impute start month (valid day)
   flag3 = impute start day and month      flag4 = impute stop day (valid month)
   flag5 = impute stop month (valid day)      flag6 = impute stop day and month */

flag1=0;      flag2=0;      flag3=0;
do over stary;
  if stary>0 then do;
    if startm>0 and startd<=0 then do; startd=15; flag1=1; srflag=1; end;
    if startm<=0 and startd>0 then do; startm=6; flag2=1; srflag=1; smofl=1; end;
    if startm<=0 and startd<=0 then do; startm=6; startd=15; flag3=1; srflag=1; smofl=1; end;
    /* keep imputed values for start months and days from preceding the Round1 interview date. */
    if stary=dli_y and startm=dli_m and startd<dli_d and srflag=1 then do; startd=dli_d; end;
    if stary=dli_y and startm<dli_m and srflag=1 then do; startm=dli_m; startd=dli_d; end;
  end;
end;

flag4=0;      flag5=0;      flag6=0;
```

```
do over stopy;
  if stopy>0 then do;
    if stopm>0 and stopd<=0 then do; stopd=15; flag4=1; spflag=1; end;
    if stopm<=0 and stopd>0 then do; stopm=6; flag5=1; spflag=1; emofl=1; end;
    if stopm<=0 and stopd<=0 then do; stopm=6; stopd=15; flag6=1; spflag=1; emofl=1; end;
  /* keep imputed values for stop months and days from exceeding the Round2 interview date. */
    if stopy=int_y and stopm=int_m and stopd>int_d and spflag=1 then do; stopd=int_d; end;
    if stopy=int_y and stopm>int_m then do; stopm=int_m; stopd=int_d; end;
  end;
end;
```

/* The following lines of code account for the cases where the respondent has a start date for a job reported in Round 1 that comes before dli in the Round 2 roster. These cases will have the original startdate reported on the Round 2 roster that is used to count the number of weeks employed. The idea is to only count the weeks employed from Round1 interview date to today. Then, the activity from Round 1 will be added to the Round 2 activity to get the full event history. To achieve this, all jobs reported in Round 1 (UID's begin with 97) will have their startdates updated to the dli. The start week and stop week for these jobs will be counted the same. */

```
do over uflag; uflag=0; end; /* Initialize uflag */
do i=1 to 9;
  if 9700<UID<9800 then do;
    if starty=dli_y and startm=dli_m and startd<dli_d then do; startd=dli_d; uflag=1; end;
    if starty=dli_y and startm<dli_m then do; startm=dli_m; startd=dli_d; uflag=1; end;
    if starty<dli_y then do; starty=dli_y; startm=dli_m; startd=dli_d; uflag=1; end;
  end;
end;
```

/*Convert START month and day to total days*/

```
do over startm;
  if startm>0 and startd>0 then do;
    if startm=1 then sttdays=startd;
    if startm=3 then sttdays=startd+59;
    if startm=5 then sttdays=startd+120;
    if startm=7 then sttdays=startd+181;
    if startm=9 then sttdays=startd+243;
    if startm=11 then sttdays=startd+304;
    if startm=2 then sttdays=startd+31;
    if startm=4 then sttdays=startd+90;
    if startm=6 then sttdays=startd+151;
    if startm=8 then sttdays=startd+212;
    if startm=10 then sttdays=startd+273;
    if startm=12 then sttdays=startd+334;
  end;
end;
```

/*Account for leap years*/

```
do over starty;
  if starty=1980 or starty=1984 or starty=1988 or starty=1992 or starty=1996 then do;
    if startm>0 and startd>0 then do;
      if startm=1 then sttdays=startd;
      if startm=3 then sttdays=startd+60;
      if startm=5 then sttdays=startd+121;
      if startm=7 then sttdays=startd+182;
      if startm=9 then sttdays=startd+244;
      if startm=11 then sttdays=startd+305;
      if startm=2 then sttdays=startd+31;
      if startm=4 then sttdays=startd+91;
      if startm=6 then sttdays=startd+152;
      if startm=8 then sttdays=startd+213;
      if startm=10 then sttdays=startd+274;
      if startm=12 then sttdays=startd+335;
    end;
  end;
end;
```

Appendix 2: Employment Variable Creation

```
end;

/*Convert STOP month and day to total days*/
do over stopm;
  if stopm>0 and stopd>0 then do;
    if stopm=1 then stpdays=stopd;
    if stopm=2 then stpdays=stopd+31;
    if stopm=3 then stpdays=stopd+59;
    if stopm=4 then stpdays=stopd+90;
    if stopm=5 then stpdays=stopd+120;
    if stopm=6 then stpdays=stopd+151;
    if stopm=7 then stpdays=stopd+181;
    if stopm=8 then stpdays=stopd+212;
    if stopm=9 then stpdays=stopd+243;
    if stopm=10 then stpdays=stopd+273;
    if stopm=11 then stpdays=stopd+304;
    if stopm=12 then stpdays=stopd+334;
  end;
end;

/*Account for leap years*/
do over stopy;
  if stopy=1980 or stopy=1984 or stopy=1988 or stopy=1992 or stopy=1996 then do;
    if stopm>0 and stopd>0 then do;
      if stopm=1 then stpdays=stopd;
      if stopm=2 then stpdays=stopd+31;
      if stopm=3 then stpdays=stopd+60;
      if stopm=4 then stpdays=stopd+91;
      if stopm=5 then stpdays=stopd+121;
      if stopm=6 then stpdays=stopd+152;
      if stopm=7 then stpdays=stopd+182;
      if stopm=8 then stpdays=stopd+213;
      if stopm=9 then stpdays=stopd+244;
      if stopm=10 then stpdays=stopd+274;
      if stopm=11 then stpdays=stopd+305;
      if stopm=12 then stpdays=stopd+335;
    end;
  end;
end;

/*Convert days into week numbers*/
/** Basic Formula: weekno=endweek{specific year}+ceil[(totdays+{# of days remaining in DEC})/7] **/
do over starty;
  if starty>0 and sttdays>0 then do;
    if starty=1980 then do; startwk=ceil((sttdays+2)/7); end;
    if starty=1981 then do; startwk=52+ceil((sttdays+4)/7); end;
    if starty=1982 then do; startwk=104+ceil((sttdays+5)/7); end;
    if starty=1983 then do; startwk=156+ceil((sttdays+6)/7); end;
    if starty=1984 then do; startwk=209+ceil((sttdays)/7); end;
    if starty=1985 then do; startwk=261+ceil((sttdays+2)/7); end;
    if starty=1986 then do; startwk=313+ceil((sttdays+3)/7); end;
    if starty=1987 then do; startwk=365+ceil((sttdays+4)/7); end;
    if starty=1988 then do; startwk=417+ceil((sttdays+5)/7); end;
    if starty=1989 then do; startwk=470+ceil((sttdays)/7); end;
    if starty=1990 then do; startwk=522+ceil((sttdays+1)/7); end;
    if starty=1991 then do; startwk=574+ceil((sttdays+2)/7); end;
    if starty=1992 then do; startwk=626+ceil((sttdays+3)/7); end;
    if starty=1993 then do; startwk=678+ceil((sttdays+5)/7); end;
    if starty=1994 then do; startwk=730+ceil((sttdays+6)/7); end;
    if starty=1995 then do; startwk=783+ceil((sttdays)/7); end;
    if starty=1996 then do; startwk=835+ceil((sttdays+1)/7); end;
    if starty=1997 then do; startwk=887+ceil((sttdays+3)/7); end;
    if starty=1998 then do; startwk=939+ceil((sttdays+4)/7); end;
    if starty=1999 then do; startwk=991+ceil((sttdays+5)/7); end;
```

```

end;
if starty<0 and starty>-4 then do; startwk=-3; end;
end;

do over stopy;
if stopy>0 and stpdays>0 then do;
if stopy=1980 then do; stopwk=ceil((stpdays+2)/7); end;
if stopy=1981 then do; stopwk=52+ceil((stpdays+4)/7); end;
if stopy=1982 then do; stopwk=104+ceil((stpdays+5)/7); end;
if stopy=1983 then do; stopwk=156+ceil((stpdays+6)/7); end;
if stopy=1984 then do; stopwk=209+ceil((stpdays)/7); end;
if stopy=1985 then do; stopwk=261+ceil((stpdays+2)/7); end;
if stopy=1986 then do; stopwk=313+ceil((stpdays+3)/7); end;
if stopy=1987 then do; stopwk=365+ceil((stpdays+4)/7); end;
if stopy=1988 then do; stopwk=417+ceil((stpdays+5)/7); end;
if stopy=1989 then do; stopwk=470+ceil((stpdays)/7); end;
if stopy=1990 then do; stopwk=522+ceil((stpdays+1)/7); end;
if stopy=1991 then do; stopwk=574+ceil((stpdays+2)/7); end;
if stopy=1992 then do; stopwk=626+ceil((stpdays+3)/7); end;
if stopy=1993 then do; stopwk=678+ceil((stpdays+5)/7); end;
if stopy=1994 then do; stopwk=730+ceil((stpdays+6)/7); end;
if stopy=1995 then do; stopwk=783+ceil((stpdays)/7); end;
if stopy=1996 then do; stopwk=835+ceil((stpdays+1)/7); end;
if stopy=1997 then do; stopwk=887+ceil((stpdays+3)/7); end;
if stopy=1998 then do; stopwk=939+ceil((stpdays+4)/7); end;
if stopy=1999 then do; stopwk=991+ceil((stpdays+5)/7); end;
end;
if stopy<0 and stopy>-4 then do; stopwk=-3; end;
end;

```

/ The following code decreases the Round 2 interview date by one so that, in Round 3, the Round 2 int. week will not be counted twice for respondents with a job during that time. The actual Round 2 int. week will counted as part of the job tenure in Round 3. The same procedure was used in Round 1. */*

```

do over stopwk;
if stopwk>0 and UID>0 then do; if stopwk>intwk then do; stopwk=intwk; end; end;
end;

```

/ The following lines considers jobs that begin the same week as the Round2 interview date. Since we are updating the Round2 interview week by -1, we need to account for jobs that start in the same week or tenures of -1 will result. */*

```

do over startwk;
if startwk>0 and UID>0 then do; if startwk>intwk then do; startwk=intwk; end; end;
end;

```

/ Two hand edit cases result from stop dates being greater than Round2 interview dates. Stop dates updated back to interview dates for each respondent. These have been subtracted by one. */*

```

if pubid=2019 then stopw1=988;
if pubid=8995 then stopw1=984;

```

/ To account for respondents interviewed in Round1 but not Round2 */*

```

if starmo1=-5 then do;

```


Appendix 2: Employment Variable Creation

```
do over startwk; startwk=-5; stopwk=-5; end; end;

/* Correcting for imputed values that resulted in the start date being later than the stop date. In these
cases, the imputed date will be updated to the good date. */
do i=1 to 9;
  if startwk>stopwk and (flag1=1 or flag2=1 or flag3=1) then do; startwk=stopwk; end;
  if startwk>stopwk and (flag4=1 or flag5=1 or flag6=1) then do; stopwk=startwk; end;
end;

/* SECTION 3: This part converts within-job gap start & stop dates into continuous week numbers.*/

/* Reads in raw data on within-job gaps and converts the gap dates into NLSY97 week numbers. The flag
variables equal one for invalid gap data and zero otherwise. */

/* JOB 1 GAPS */
/* These variables are read as follows:      BGDY1_1 = Begin day of within-job gap 1 on job 1
      EGMO1_6 = End month of within-job gap 6 on job 1
      BGAP1_3 = Begin week of within-job gap 3 on job 1  [CREATED]  */
array bgdy BGDY1_1-BGDY1_9;      array bgmo BGMO1_1-BGMO1_9;
array bgyr BGYR1_1-BGYR1_9;      array egdy EGDY1_1-EGDY1_9;
array egmo EGMO1_1-EGMO1_9;      array egyr EGYR1_1-EGYR1_9;
array bdays bday1_1-bday1_9;    /* begin day of job1_gap# (internal calculation)*/
array edays eday1_1-eday1_9;    /* end day of job1_gap# (internal calculation)*/
array bweek bgap1_1-bgap1_9;    /* begin week of job1_gap# (created) */
array eweek egap1_1-egap1_9;    /* end week of job1_gap# (created) */
array bflag bflg1_1-bflg1_9;
array eflag eflg1_1-eflg1_9;
array bgfl bgfl1_1-bgfl1_9;
array egfl egfl1_1-egfl1_9;

/* only impute start/stop dates if day is missing */
/* Fill-in start day for those missing */
do over bgyr;
  if bgyr>0 then do;
    if bgmo>0 and bgdy<=0 then do; bgdy=15; bflag=1; end;
  /* Account for beginning gap dates before job */
  if bgyr=staryr1 and bgmo=starmom1 and bgdy<starday1 and bflag=1 then do; bgdy=starday1; end;
  end;
end;

/* Fill-in stop day for those missing */
do over egyr;
  if egyr>0 then do;
    if egmo>0 and egdy<=0 then do; egdy=15; eflag=1; end;
  /* Account for end gap dates after job end */
  if egyr=stopyr1 and egmo=stopmom1 and egdy>stopday1 and eflag=1 then do; egdy=stopday1; end;
  end;
end;

/*Set flag for gap exists but invalid data*/
do over bgyr;
  bgfl=-4;
```

Appendix 2: Employment Variable Creation

```
if (-4 < bgmo < 0) or (-4 < bgyr < 0) then do; bgfl=1; end;
end;

do over egyr;
  egfl=-4;
  if (-4 < egyr < 0) or (-4 < egmo < 0) then do; egfl=1; end;
end;

/**Identify within-job gaps on JOB 1; Convert gap dates to week numbers ***/
/*Convert START month and day to total days (BDAYS)*/
do over bgmo;
  if bgmo>0 and bgdy>0 then do;
    if bgmo=1 then bdays=bgdy;
    if bgmo=2 then bdays=bgdy+31;
    if bgmo=3 then bdays=bgdy+59;
    if bgmo=4 then bdays=bgdy+90;
    if bgmo=5 then bdays=bgdy+120;
    if bgmo=6 then bdays=bgdy+151;
    if bgmo=7 then bdays=bgdy+181;
    if bgmo=8 then bdays=bgdy+212;
    if bgmo=9 then bdays=bgdy+243;
    if bgmo=10 then bdays=bgdy+273;
    if bgmo=11 then bdays=bgdy+304;
    if bgmo=12 then bdays=bgdy+334;
  end;
end;

/**Account for leap years***/
do over bgyr;
  if bgyr=1980 or bgyr=1984 or bgyr=1988 or bgyr=1992 or bgyr=1996 then do;
    if bgmo>0 and bgdy>0 then do;
      if bgmo=1 then bdays=bgdy;
      if bgmo=2 then bdays=bgdy+31;
      if bgmo=3 then bdays=bgdy+60;
      if bgmo=4 then bdays=bgdy+91;
      if bgmo=5 then bdays=bgdy+121;
      if bgmo=6 then bdays=bgdy+152;
      if bgmo=7 then bdays=bgdy+182;
      if bgmo=8 then bdays=bgdy+213;
      if bgmo=9 then bdays=bgdy+244;
      if bgmo=10 then bdays=bgdy+274;
      if bgmo=11 then bdays=bgdy+305;
      if bgmo=12 then bdays=bgdy+335;
    end;
  end;
end;

/**Convert STOP month and day to total days (EDAYS)***/
do over egmo;
  if egmo>0 and egdy>0 then do;
    if egmo=1 then edays=egdy;
    if egmo=2 then edays=egdy+31;
    if egmo=3 then edays=egdy+59;
    if egmo=4 then edays=egdy+90;
    if egmo=5 then edays=egdy+120;
    if egmo=6 then edays=egdy+151;
    if egmo=7 then edays=egdy+181;
    if egmo=8 then edays=egdy+212;
    if egmo=9 then edays=egdy+243;
    if egmo=10 then edays=egdy+273;
    if egmo=11 then edays=egdy+304;
    if egmo=12 then edays=egdy+334;
  end;
end;

/**Account for leap years***/
do over egyr;
  if egyr=1980 or egyr=1984 or egyr=1988 or egyr=1992 or egyr=1996 then do;
    if egmo>0 and egdy>0 then do;
      if egmo=1 then edays=egdy;
      if egmo=2 then edays=egdy+31;
```

Appendix 2: Employment Variable Creation

```
if egmo=3 then edays=egdy+60;          if egmo=4 then edays=egdy+91;
if egmo=5 then edays=egdy+121;        if egmo=6 then edays=egdy+152;
if egmo=7 then edays=egdy+182;        if egmo=8 then edays=egdy+213;
if egmo=9 then edays=egdy+244;        if egmo=10 then edays=egdy+274;
if egmo=11 then edays=egdy+305;       if egmo=12 then edays=egdy+335;
end;
end;
end;

/**Convert days into week numbers**/
/** Basic Formula: weekno=endweek{specific year}+ceil[(totdays+{# of days remaining in DEC})/7] **/
/** Note: Use this program takes the week following the actual start of the gap as the measure of when the
non-working period begins. **/
do over bgyr;
  if bgyr>0 and bdays>0 then do;
    if bgyr=1980 then bweek=ceil((bdays+2)/7);          if bgyr=1981 then bweek=52+ceil((bdays+4)/7);
    if bgyr=1982 then bweek=104+ceil((bdays+5)/7);    if bgyr=1983 then bweek=156+ceil((bdays+6)/7);
    if bgyr=1984 then bweek=209+ceil((bdays)/7);      if bgyr=1985 then bweek=261+ceil((bdays+2)/7);
    if bgyr=1986 then bweek=313+ceil((bdays+3)/7);    if bgyr=1987 then bweek=365+ceil((bdays+4)/7);
    if bgyr=1988 then bweek=417+ceil((bdays+5)/7);    if bgyr=1989 then bweek=470+ceil((bdays)/7);
    if bgyr=1990 then bweek=522+ceil((bdays+1)/7);    if bgyr=1991 then bweek=574+ceil((bdays+2)/7);
    if bgyr=1992 then bweek=626+ceil((bdays+3)/7);    if bgyr=1993 then bweek=678+ceil((bdays+5)/7);
    if bgyr=1994 then bweek=730+ceil((bdays+6)/7);    if bgyr=1995 then bweek=783+ceil((bdays)/7);
    if bgyr=1996 then bweek=835+ceil((bdays+1)/7);    if bgyr=1997 then bweek=887+ceil((bdays+3)/7);
    if bgyr=1998 then bweek=939+ceil((bdays+4)/7);    if bgyr=1999 then bweek=991+ceil((bdays+5)/7);
    if bweek>0 then do; bweek=bweek+1; end;
  end;
end;

do over egyr;
  if egyr>0 and edays>0 then do;
    if egyr=1980 then eweek=ceil((edays+2)/7);          if egyr=1981 then eweek=52+ceil((edays+4)/7);
    if egyr=1982 then eweek=104+ceil((edays+5)/7);    if egyr=1983 then eweek=156+ceil((edays+6)/7);
    if egyr=1984 then eweek=209+ceil((edays)/7);      if egyr=1985 then eweek=261+ceil((edays+2)/7);
    if egyr=1986 then eweek=313+ceil((edays+3)/7);    if egyr=1987 then eweek=365+ceil((edays+4)/7);
    if egyr=1988 then eweek=417+ceil((edays+5)/7);    if egyr=1989 then eweek=470+ceil((edays)/7);
    if egyr=1990 then eweek=522+ceil((edays+1)/7);    if egyr=1991 then eweek=574+ceil((edays+2)/7);
    if egyr=1992 then eweek=626+ceil((edays+3)/7);    if egyr=1993 then eweek=678+ceil((edays+5)/7);
    if egyr=1994 then eweek=730+ceil((edays+6)/7);    if egyr=1995 then eweek=783+ceil((edays)/7);
    if egyr=1996 then eweek=835+ceil((edays+1)/7);    if egyr=1997 then eweek=887+ceil((edays+3)/7);
    if egyr=1998 then eweek=939+ceil((edays+4)/7);    if egyr=1999 then eweek=991+ceil((edays+5)/7);
    if eweek>0 then do; eweek=eweek-1; end;
  end;
end;

/* The following lines omit gap start and stop dates for gaps less than one work week (5 days) */
do over bdays;
  if edays-bdays<5 and bweek>eweek and bdays ne . and edays ne . then do; bweek=.; eweek=.; end;
end;

/* The following omits cases where bweek>eweek, which are caused when missing values are substituted
in. For example, when the day of a beginning gap is unknown, the program uses the 15th. This can cause
```

bweek>eweek, which will cause problems when writing programs for the created variables for the government. This situation will be fixed by making eweek and bweek the same. */

do over bweek; if bweek>eweek then bweek=eweek; end;

/* To correct for bad gap information */

do over bdays;

if eweek>stopw1 and eweek ne . then eweek=stopw1;

if bweek<starw1 and bweek ne . then eweek=starw1;

end;

/****** At this point in the program, this code is repeated for gaps within each of the respondent's jobs (job 2, job 3, etc.). Due to space constraints the complete program is not included here; researchers should contact NLS User Services if additional information is required.******/

/***** SECTION 4: The program finally counts weeks worked between 1980 and 1999. *****/

/* By creating a dummy variables for each week in from 1980 to 1999, this program counts the weeks worked by the respondent and removes the within job gaps. It will place a "1" into weeks where the respondent was employed and a "0" into weeks where the respondent was not employed. There are 9 different event histories, one for each possible job on the roster. Weeks range from the first week of 1980 to the last week of 1999, for a total of 1044 weeks.

array job1wks (i) wk1_1-wk1_1044; array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044; array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044; array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044; array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;
array starw (i) starw1-starw9; array stopw (i) stopw1-stopw9;

/* Default Settings*/

do i=1 to 1044;

job1wks=0; job2wks=0; job3wks=0;

job4wks=0; job5wks=0; job6wks=0;

job7wks=0; job8wks=0; job9wks=0;

end;

/* Define rd2wk as the maximum of dliwk and age14wk. This is used for bad start/stop weeks. */

if dliwk>age14wk then do; rd2wk=dliwk; end;

if age14wk=>dliwk then do; rd2wk=age14wk; end;

/*** total weeks worked on job 1 ***/

/* Set up starfl dummy for invalid dates used later */

starfl_1=0; stopfl_1=0;

if starw1=-3 and uid1 ne -5 then do; starw1=rd2wk; starfl_1=1; end;

if stopw1=-3 and uid1 ne -5 then do; stopw1=intwk; stopfl_1=1; end;

if smofl1=1 then do; starfl_1=1; end;

if emofl1=1 then do; stopfl_1=1; end;

if starw1>0 and stopw1>0 then do; /* [1] */

do i=(starw1) to (stopw1); job1wks=1; end;

Appendix 2: Employment Variable Creation

```
/* Remove gap 1 on job 1 */
  if bgap1_1>0 & egap1_1>0 then do; do i=(bgap1_1) to (egap1_1); job1wks=0; end; end;
/* Remove gap 2 on job 1 */
  if bgap1_2>0 & egap1_2>0 then do; do i=(bgap1_2) to (egap1_2); job1wks=0; end; end;
/* and so on through gap 9: variables bgap1_9 and egap1_9 */

/** end of normal gaps, begin missing gap information ***/
if bgap1_1=. then bgap1_1=10000;      if egap1_1=. then egap1_1=0;
/* and so on for gaps 2, 3, etc., through */
if bgap1_9=. then bgap1_9=10000;      if egap1_9=. then egap1_9=0;

/* Remove gap 1 on job 1 - beginning gap date bad */
  if bgfl1_1=1 & egap1_1>0 then do; do i=(starw1) to (egap1_1); job1wks=-3; gpfl1_1=1; end; end;
/* Remove gap 1 on job 1 - end gap date bad */
  if bgap1_1>0 & egfl1_1=1 then do; do i=(bgap1_1) to (stopw1); job1wks=-3; gpfl1_1=1; end; end;
/* Remove gap 1 on job 1 - both gap dates bad */
  if bgfl1_1=1 & egfl1_1=1 then do; do i=(starw1) to (stopw1); job1wks=-3; gpfl1_1=1; end; end;

/* At this point the program loops through the above three lines of code for each gap 2-9 (for example, the
gap 2 variables are bgap1_2, egap1_2, bgfl1_2, egfl1_2, and gpfl1_2). This code is deleted here due to
space considerations; users who need the complete program should contact NLS User Services. */

if (starfl_1=1) then do;
  do i=(starw1) to min(stopw1, bgap1_1-1, bgap1_2-1, bgap1_3-1, bgap1_4-1, bgap1_5-1, bgap1_6-1,
    bgap1_7, bgap1_8, bgap1_9);
    job1wks=-3;
  end;
end;

if (stopfl_1=1) then do;
  do i=max(starw1, egap1_1+1, egap1_2+1, egap1_3+1, egap1_4+1, egap1_5+1, egap1_6+1, egap1_7+1,
    egap1_8+1, egap1_9+1) to (stopw1);
    job1wks=-3;
  end;
end;

/***** At this point in the program, this code is repeated for to calculate weeks worked, excluding
gaps, for each of the respondent's jobs. Due to space constraints the complete program is not included
here; researchers should contact NLS User Services if additional information is required. *****/

endsas;
```

Appendix 2: Employment Variable Creation

BDATE1.SAS

This program changes the respondent's birthday and 14th birthday to a continuous week number. The variables used are the following:

Name in Program	Question Name on CD
birthdy	KEY!BDATE_D
birthmo	KEY!BDATE_M
birthyr	KEY!BDATE_Y
norcid	YNORCID

```
/***** Calculate Age 14 year *****/
```

```
AGE14YR=birthyr+14;
```

```
/** Convert Age 14 Birthdate to week number **/
```

```
/*change age14 month/day to total days(bbdays)*/
```

```
if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=2 then bbdays=birthdy+31;
  if birthmo=3 then bbdays=birthdy+59;
  if birthmo=4 then bbdays=birthdy+90;
  if birthmo=5 then bbdays=birthdy+120;
  if birthmo=6 then bbdays=birthdy+151;
  if birthmo=7 then bbdays=birthdy+181;
  if birthmo=8 then bbdays=birthdy+212;
  if birthmo=9 then bbdays=birthdy+243;
  if birthmo=10 then bbdays=birthdy+273;
  if birthmo=11 then bbdays=birthdy+304;
  if birthmo=12 then bbdays=birthdy+334;
end;
```

```
/*Account for leap years*/
```

```
if age14yr=1980 or age14yr=1984 or
  age14yr=1988 or age14yr=1992 or
  age14yr=1996 then do;
  if birthmo>0 and birthdy>0 then do;
    if birthmo=1 then bbdays=birthdy;
    if birthmo=2 then bbdays=birthdy+31;
    if birthmo=3 then bbdays=birthdy+60;
    if birthmo=4 then bbdays=birthdy+91;
    if birthmo=5 then bbdays=birthdy+121;
    if birthmo=6 then bbdays=birthdy+152;
    if birthmo=7 then bbdays=birthdy+182;
    if birthmo=8 then bbdays=birthdy+213;
    if birthmo=9 then bbdays=birthdy+244;
    if birthmo=10 then bbdays=birthdy+274;
    if birthmo=11 then bbdays=birthdy+305;
    if birthmo=12 then bbdays=birthdy+335;
  end;
end;
```

```
/* Convert days into week numbers */
```

```
/** Basic Formula: weekno=endweek{specific
year}+ceil[(totdays+{# of days remaining in
DEC})/7] **/
```

```
age14wk=9999; /* Default age 14 week = 9999 */
```

```
if age14yr>0 and bbdays>0 then do;
  if age14yr=1980 then do;
    age14wk=ceil((bbdays+2)/7); end;
  if age14yr=1981 then do;
    age14wk=52+ceil((bbdays+4)/7); end;
  if age14yr=1982 then do;
    age14wk=104+ceil((bbdays+5)/7); end;
  if age14yr=1983 then do;
    age14wk=156+ceil((bbdays+6)/7); end;
  if age14yr=1984 then do;
    age14wk=209+ceil((bbdays)/7); end;
  if age14yr=1985 then do;
    age14wk=261+ceil((bbdays+2)/7); end;
  if age14yr=1986 then do;
    age14wk=313+ceil((bbdays+3)/7); end;
  if age14yr=1987 then do;
    age14wk=365+ceil((bbdays+4)/7); end;
  if age14yr=1988 then do;
    age14wk=417+ceil((bbdays+5)/7); end;
  if age14yr=1989 then do;
    age14wk=470+ceil((bbdays)/7); end;
  if age14yr=1990 then do;
    age14wk=522+ceil((bbdays+1)/7); end;
  if age14yr=1991 then do;
    age14wk=574+ceil((bbdays+2)/7); end;
  if age14yr=1992 then do;
    age14wk=626+ceil((bbdays+3)/7); end;
  if age14yr=1993 then do;
    age14wk=678+ceil((bbdays+5)/7); end;
  if age14yr=1994 then do;
    age14wk=730+ceil((bbdays+6)/7); end;
  if age14yr=1995 then do;
    age14wk=783+ceil((bbdays)/7); end;
  if age14yr=1996 then do;
    age14wk=835+ceil((bbdays+1)/7); end;
  if age14yr=1997 then do;
    age14wk=887+ceil((bbdays+3)/7); end;
```

Appendix 2: Employment Variable Creation

```
if age14yr=1998 then do;
  age14wk=939+ceil((bbdays+4)/7); end;
if age14yr=1999 then do;
  age14wk=991+ceil((bbdays+5)/7); end;
end;

/** Convert Birthdate to week number **/
/*Default birthdate week=0 if bdate <12/30/79*/
birthwk=0;

if birthyr>0 and bbdays>0 then do;
  if birthyr=1980 then do;
    birthwk=ceil((bbdays+2)/7); end;
  if birthyr=1981 then do;
    birthwk=52+ceil((bbdays+4)/7); end;
  if birthyr=1982 then do;
    birthwk=104+ceil((bbdays+5)/7); end;
  if birthyr=1983 then do;
    birthwk=156+ceil((bbdays+6)/7); end;
  if birthyr=1984 then do;
    birthwk=209+ceil((bbdays)/7); end;
  if birthyr=1985 then do;
    birthwk=261+ceil((bbdays+2)/7); end;
  if birthyr=1986 then do;
    birthwk=313+ceil((bbdays+3)/7); end;
  if birthyr=1987 then do;
    birthwk=365+ceil((bbdays+4)/7); end;

  if birthyr=1988 then do;
    birthwk=417+ceil((bbdays+5)/7); end;
  if birthyr=1989 then do;
    birthwk=470+ceil((bbdays)/7); end;
  if birthyr=1990 then do;
    birthwk=522+ceil((bbdays+1)/7); end;
  if birthyr=1991 then do;
    birthwk=574+ceil((bbdays+2)/7); end;
  if birthyr=1992 then do;
    birthwk=626+ceil((bbdays+3)/7); end;
  if birthyr=1993 then do;
    birthwk=678+ceil((bbdays+5)/7); end;
  if birthyr=1994 then do;
    birthwk=730+ceil((bbdays+6)/7); end;
  if birthyr=1995 then do;
    birthwk=783+ceil((bbdays)/7); end;
  if birthyr=1996 then do;
    birthwk=835+ceil((bbdays+1)/7); end;
  if birthyr=1997 then do;
    birthwk=887+ceil((bbdays+3)/7); end;
  if birthyr=1998 then do;
    birthwk=939+ceil((bbdays+4)/7); end;
  if birthyr=1999 then do;
    birthwk=991+ceil((bbdays+5)/7); end;
end;
endsas;
```

Appendix 2: Employment Variable Creation

HOURLY RATE OF PAY, HOURLY MONETARY COMPENSATION AND JOB LENGTH < 13 WEEKS

Variables Created: CV_HRLY_PAY
CV_HRLY_COMPENSATION
CV_JOB<13_WKS

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
PUBID	PUBID	E2120011-E2120016	YEMP-21200.01-000001 --000006
E200	YEMP-200	E2120021-E2120026	YEMP-21200.02-000001 --000006
E6001-E60010	YEMP-600.01-.10	E2120031-E2120036	YEMP-21200.03-000001 --000006
E192001-E192009	YEMP-19200.01-.09	E2120041-E2120046	YEMP-21200.04-000001 --000006
E207001-E207009	YEMP-20700.01-.09	E2120051-E2120056	YEMP-21200.05-000001 --000006
E229001-E229006	YEMP-22900.01-.06	E2120061-E2120066	YEMP-21200.06-000001 --000006
E230001-E230009	YEMP-23000.01-.09	E2260411-E2260419, E2260410	YEMP-22604.01-000001 --000010
E232001-E232004	YEMP-23200.01-.04	E2260421-E2260429, E2260420	YEMP-22604.02-000001 --000010
E239001-E239004	YEMP-23900.01-.04	E2260431-E2260439, E2260430	YEMP-22604.03-000001 --000010
E239011-E239019	YEMP-23901.01-.09	E2260441-E2260449, E2260440	YEMP-22604.04-000001 --000010
E245011-E245016	YEMP-24501.01-.06	E1022511-E1022519, E1022510	YEMP-100225.01-000001 --000010
E245021-E245026	YEMP-24502.01-.06	E1022521-E1022529, E1022520	YEMP-100225.02-000001 --000010
E245141-E245146	YEMP-24514.01-.06	E1022531-E1022539, E1022530	YEMP-100225.03-000001 --000010
E24514B3	YEMP-24514B.03	E1022541-E1022549, E1022540	YEMP-100225.04-000001 --000010
E334001-E334007	YEMP-33400.01-.07	E1022551-E1022559, E1022550	YEMP-100225.05-000001 --000010
E335001-E335005	YEMP-33500.01-.05	E3840741, E3840744	YEMP-38407.04.01, .04
E336001-E336004	YEMP-33600.01-.04	E3840753	YEMP-38407.05.03
E344021-E344027	YEMP-34402.01-.07	E3841611-E3841615	YEMP-38416.01.01-.05
E34402B1-E34402B4	YEMP-34402AB.01-.04	E3841621-E3841625	YEMP-38416.02.01-.05
E344281-E344285	YEMP-34428.01-.05	E3841631-E3841634	YEMP-38416.03.01-.04
E344301	YEMP-34430.01	E3841641, E3841644	YEMP-38416.04.01, .04
E361001-E361005	YEMP-36100.01-.05	E3841653	YEMP-38416.05.03
E362001-E362007	YEMP-36200.01-.07	E831001-E831007	YEMP-83100.01-.07
E2160011-E2160015	YEMP-21600.01.02-.06	E868001-E868005	YEMP-86800.01-.05
E2160021-E2160025	YEMP-21600.02.02-.06	E869001-E869006	YEMP-86900.01-.06
E2160031-E2160034	YEMP-21600.03.02-.05	E871001-E871005	YEMP-87100.01-.05
E2160041-E2160043	YEMP-21600.04.02-.04	E885011-E885015	YEMP-88501.01-.05
E2160051, E2160052	YEMP-21600.05.02, .03	E885021-E885025	YEMP-88502.01-.05
E2160061, E2160062	YEMP-21600.06.02, .03	E885121-E885125	YEMP-88512.01-.05
E2250011-E2250015	YEMP-22500.01.02-.06	E973001-E973007	YEMP-97300.01-.07
E2250021-E2250025	YEMP-22500.02.02-.06	E974001-E974005	YEMP-97400.01-.05
E2250031-E2250034	YEMP-22500.03.02-.05	E975001-E975004	YEMP-97500.01-.04
E2250041-E2250043	YEMP-22500.04.02-.04	E983001-E983004	YEMP-98300.01-.04
E2250051, E2250052	YEMP-22500.05.02, .03	E984021-E984027	YEMP-98402.01-.07
E2250061, E2250062	YEMP-22500.06.02, .03	E98402D1-E98402D4	YEMP-98402D.01-.04
E37901B1-E37901B9	YEMP-37901B.01-.09	E984031-E984035	YEMP-98403.01-.05
E3800B1-E3800B9	YEMP-38000B.01-.09	E984041-E984045	YEMP-98404.01-.05
E3800F1-E3800F6	YEMP-38000F.01-.06	E984141-E984145	YEMP-98414.01-.05
E380131-E380139	YEMP-38013.01-.09	E98414B1	YEMP-98414B.01
E380141-E380146	YEMP-38014.01-.06	E984291-E984295	YEMP-98429.01-.05
E380231-E380236	YEMP-38023.01-.06	E995001-E995004	YEMP-99500.01-.04
E380271, E380272	YEMP-38027.01, .02	E1001001-E1001007	YEMP-100100.01-.07
E381011-E381016	YEMP-38101.01-.06	E1020511-E1020514	YEMP-100205.01.02-.05
E381021, E381022	YEMP-38102.01, .02	E1020521-E1020525	YEMP-100205.02.02-.06
E381031, E381032	YEMP-38103.01, .02	E1020531-E1020534	YEMP-100205.03.02-.05
E381041, E381042	YEMP-38104.01, .02	E1020541-E1020543	YEMP-100205.04.02-.04
E381051, E381052	YEMP-38105.01, .02	E1020551-E1020555	YEMP-100205.05.02-.06
E381061-E381066	YEMP-38106.01-.06	E1021411-E1021414	YEMP-100214.01.02-.05
E381071-E381076	YEMP-38107.01-.06	E1021421-E1021425	YEMP-100214.02.02-.06
E381161-E381166	YEMP-38116.01-.06	E1021431-E1021434	YEMP-100214.03.02-.05
E38116B1	YEMP-38116B.01	E1021441, E1021442	YEMP-100214.04.02, .03
E382011-E382016	YEMP-38201.01-.06	E1021432	YEMP-100214.04.04
E382021-E382024	YEMP-38202.01-.04	E1021451-E1021455	YEMP-100214.05.02-.06

Appendix 2: Employment Variable Creation

E382111-E382114	YEMP-38211.01-.04	E226091-E226094	YEMP-22609.01.01-.04.01
E599001-E599007	YEMP-59900.01-.07	E226101-E226103	YEMP-22610.01.02-.03.02
E344031-E344035	YEMP-34403.01-.05	E226111	YEMP-22611.01.03
E344041-E344045	YEMP-34404.01-.05	E226121, E226122	YEMP-22612.01.04, .02.04
E34413C1-E34413C5	YEMP-34413C.01-.05	E226131	YEMP-22613.01.05
E34413E1	YEMP-34413E.01	E226151, E226152	YEMP-22615.01.07, .02.07
E380011-E380019	YEMP-38001.01-.09	E226172	YEMP-22617.02.09
E380021-E380025	YEMP-38002.01-.05	E226261-E226264	YEMP-22626.01.01-.04.01
E380031-E380035	YEMP-38003.01-.05	E226271, E226272	YEMP-22627.01.04, .02.04
E380121-E380125	YEMP-38012.01-.05	E226281	YEMP-22628.01.05
E38012B1	YEMP-38012B.01	E226301, E226302	YEMP-22630.01.07, .02.07
E381011-E381016	YEMP-38101.01-.06	E226322	YEMP-22632.02.09
E381021, E381022	YEMP-38102.01, .02	E377011-E377019	YEMP-37701.01-.09
E381051, E381052	YEMP-38105.01, .02	E1002301-E1002304	YEMP-100230.01.01-.04.01
E382011-E382016	YEMP-38201.01-.06	E1002311, E1002313	YEMP-100231.01.02, .03.02
E382021-E382024	YEMP-38202.01-.04	E1002335	YEMP-100233.05.04
E382111-E382114	YEMP-38211.01-.04	E1002341	YEMP-100234.01.05
E383131-E383139	YEMP-38313.01-.09	E1002361	YEMP-100236.01.07
E383291-E383295	YEMP-38329.01-.05	E1002392	YEMP-100239.02.09
E38329B1-E38329B3	YEMP-38329B.01-.03	E1002481-E1002484	YEMP-100248.01.01-.04.01
E38329D1-E38329D3	YEMP-38329D.01-.03	E1002495	YEMP-100249.05.04
E383301-E383304	YEMP-38330.01-.04	E1002501	YEMP-100250.01.05
E3840711-E3840715	YEMP-38407.01.01-.05	E1002521	YEMP-100252.01.07
E3840721-E3840725	YEMP-38407.02.01-.05	E1002542	YEMP-100254.02.09
E3840731-E3840734	YEMP-38407.03.01-.04		

Codes for Created Variable

Note that hourly rate of pay is reported with two implied decimal places.

This program creates the hourly rate of pay for NLSY97 respondents. The hourly rate of pay is constructed from stop date information for respondents who have a job lasting more than 13 weeks. For all other respondents the start wage is used.

In addition, this program creates an hourly monetary compensation variable for NLSY97 respondents. This variable that includes information about all compensation received by the respondent, such as tips, bonuses, commissions, overtime, etc., in the calculation. Hourly monetary compensation differs from hourly rate of pay variable, which calculates only the base pay rate. This variable is constructed from stop date information for respondents with jobs longer than 13 weeks and start date information for other.

Finally, a variable indicating whether the jobs lasted more than 13 weeks is also created. There are up to 9 jobs reported, so each variable is created for 9 jobs.

/* DECLARING THE ARRAYS TO BE LATER ON USED IN THE PROGRAM */

array E19200 E192001-E192009;	array E37901B E37901B1-E37901B9;
array E59900 E599001-E599009;	array E22900 E229001-E229009;
array E23000 E230001-E230009;	array E23200 E232001-E232009;
array E23900 E239001-E239009;	array E33400 E334001-E334009;
array E34402 E344021-E344029;	array E34402B E34402B1-E34402B9;
array E33600 E336001-E336009;	array E33500 E335001-E335009;
array E34428 E344281-E344289;	array E34430 E344301-E344309;
array E36100 E361001-E361009;	array E36200 E362001-E362009;
array E20700 E207001-E207009;	array E24501 E245011-E245019;
array E24514 E245141-E245149;	array E24514B E24514B1-E24514B9;
array E38013 E380131-E380139;	array E38014 E380141-E380149;
array E38023 E380231-E380239;	array E38106 E381061-E381069;
array E38107 E381071-E381079;	array E38116 E381161-E381169;
array E3800B E3800B1-E3800B9;	array E38027 E380271-E380279;

Appendix 2: Employment Variable Creation

array E23901 E239011-E239019;
array E38101 E381011-E381019;
array E38102 E381021-E381029;
array E38201 E382011-E382019;
array E24502 E245021-E245029;
array E38211 E382111-E382119;
array E38211B E38211B1-E38211B9;
array E38313 E383131-E383139;
array E86800 E868001-E868009;
array E87100 E871001-E871009;
array E97300 E973001-E973009;
array E97500 E975001-E975009;
array E98402 E984021-E984029;
array E98429 E984291-E984299;
array E100000 E1000001-E1000009;
array E99500 E995001-E995009;
array E88502 E885021-E885029;
array E88512B E88512B1-E88512B9;
array E34404 E344041-E344049;
array E34413E E34413E1-E34413E9;
array E98404 E984041-E984049;
array E98414B E98414B1-E98414B9;
array E38002 E380021-E380029;
array E38012 E380121-E380129;
array E38329B E38329B1-E38329B9;
array E38329 E383291-E383299;
array E35600 E356001-E356009;
array E22610 E226101-E226109;
array E22612 E226121-E226129;
array E22614 E226141-E226149;
array E22616 E226161-E226269;
array E22626 E226261-E226269;
array E22628 E226281-E226289;
array E22630 E226301-E226309;
array E22632 E226321-E226329;
array E100231 E1002311-E1002319;
array E100233 E1002331-E1002339;
array E100235 E1002351-E1002359;
array E100237 E1002371-E1002379;
array E100248 E1002481-E1002489;
array E100250 E1002501-E1002509;
array E100252 E1002521-E1002529;
array E100254 E1002541-E1002549;
array E37701 E377011-E377019;
array E58201 E582011-E582019;
array E3800F E3800F1-E3800F9;
array E38116B E38116B1-E38116B9;
array E38103 E381031-E381039;
array E38105 E381051-E381059;
array E38202 E382021-E382029;
array E38103F E38103F1-E38103F9;
array E34400 E344001-E344009;
array E83100 E831001-E831009;
array E86900 E869001-E869009;
array E87800 E878001-E878009;
array E97400 E974001-E974009;
array E98300 E983001-E983009;
array E98402D E98402D1-E98402D9;
array E98429E E98429E1-E98429E9;
array E100100 E1001001-E1001009;
array E88501 E885011-E885019;
array E88512 E885121-E885129;
array E34403 E344031-E344039;
array E34413C E34413C1-E34413C9;
array E98403 E984031-E984039;
array E98414 E984141-E984149;
array E38001 E380011-E380019;
array E38003 E380031-E380039;
array E38012B E38012B1-E38012B9;
array E38329D E38329D1-E38329D9;
array E38330 E383301-E383309;
array E22609 E226091-E226099;
array E22611 E226111-E226119;
array E22613 E226131-E226139;
array E22615 E226151-E226159;
array E22617 E226171-E226179;
array E22627 E226271-E226279;
array E22629 E226291-E226299;
array E22631 E226311-E226319;
array E100230 E1002301-E1002309;
array E100232 E1002321-E1002329;
array E100234 E1002341-E1002349;
array E100236 E1002361-E1002369;
array E100239 E1002391-E1002399;
array E100249 E1002491-E1002499;
array E100251 E1002511-E1002519;
array E100253 E1002531-E1002539;
array E36802 E368021-E368029;
array E58401 E584011-E584019;
array E225001 E2250011 E2250021 E2250031 E2250041 E2250051 E2250061 E2250071 E2250081 E2250091;
array E225002 E2250012 E2250022 E2250032 E2250042 E2250052 E2250062 E2250072 E2250082 E2250092;
array E225003 E2250013 E2250023 E2250033 E2250043 E2250053 E2250063 E2250073 E2260083 E2250093;
array E225004 E2250014 E2250024 E2250034 E2250044 E2250054 E2250064 E2250074 E2250084 E2250094;
array E225005 E2250015 E2250025 E2250035 E2250045 E2250055 E2250065 E2250075 E2250085 E2250095;
array E216001 E2160011 E2160021 E2160031 E2160041 E2160051 E2160061 E2160071 E2160081 E2160091;
array E216002 E2160012 E2160022 E2160032 E2160042 E2160052 E2160062 E2160072 E2160082 E2160092;
array E216003 E2160013 E2160023 E2160033 E2160043 E2160053 E2160063 E2160073 E2160083 E2160093;
array E216004 E2160014 E2160024 E2160034 E2160044 E2160054 E2160064 E2160074 E2160084 E2160094;
array E216005 E2160015 E2160025 E2160035 E2160045 E2160055 E2160065 E2160075 E2160085 E2160095;
array E212001 E2120011 E2120021 E2120031 E2120041 E2120051 E2120061 E2120071 E2120081 E2120091;

Appendix 2: Employment Variable Creation

array E212002 E2120012 E2120022 E2120032 E2120042 E2120052 E2120062 E2120072 E2120082 E2120092;
array E212003 E2120013 E2120023 E2120033 E2120043 E2120053 E2120063 E2120073 E2120083 E2120093;
array E212004 E2120014 E2120024 E2120034 E2120044 E2120054 E2120064 E2120074 E2120084 E2120094;
array E212005 E2120015 E2120025 E2120035 E2120045 E2120055 E2120065 E2120075 E2120085 E2120095;
array E212006 E2120016 E2120026 E2120036 E2120046 E2120056 E2120066 E2120076 E2120086 E2120096;
array E384161 E3841611 E3841621 E3841631 E3841641 E3841651 E3841661 E3841671 E3841681 E3841691;
array E384162 E3841612 E3841622 E3841632 E3841642 E3841652 E3841662 E3841672 E3841682 E3841692;
array E384163 E3841613 E3841623 E3841633 E3841643 E3841653 E3841663 E3841673 E3841683 E3841693;
array E384164 E3841614 E3841624 E3841634 E3841644 E3841654 E3841664 E3841674 E3841684 E3841694;
array E384165 E3841615 E3841625 E3841635 E3841645 E3841655 E3841665 E3841675 E3841685 E3841695;
array E384071 E3840711 E3840721 E3840731 E3840741 E3840751 E3840761 E3840771 E3840781 E3840791;
array E384072 E3840712 E3840722 E3840732 E3840742 E3840752 E3840762 E3840772 E3840782 E3840792;
array E384073 E3840713 E3840723 E3840733 E3840743 E3840753 E3840763 E3840773 E3840783 E3840793;
array E384074 E3840714 E3840724 E3840734 E3840744 E3840754 E3840764 E3840774 E3840784 E3840794;
array E384075 E3840715 E3840725 E3840735 E3840745 E3840755 E3840765 E3840775 E3840785 E3840795;
array E102051 E1020511 E1020521 E1020531 E1020541 E1020551 E1020561 E1020571 E1020581 E1020591;
array E102052 E1020512 E1020522 E1020532 E1020542 E1020552 E1020562 E1020572 E1020582 E1020592;
array E102053 E1020513 E1020523 E1020533 E1020543 E1020553 E1020563 E1020573 E1020583 E1020593;
array E102054 E1020514 E1020524 E1020534 E1020544 E1020554 E1020564 E1020574 E1020584 E1020594;
array E102055 E1020515 E1020525 E1020535 E1020545 E1020555 E1020565 E1020575 E1020585 E1020595;
array E102141 E1021411 E1021421 E1021431 E1021441 E1021451 E1021461 E1021471 E1021481 E1021491;
array E102142 E1021412 E1021422 E1021432 E1021442 E1021452 E1021462 E1021472 E1021482 E1021492;
array E102143 E1021413 E1021423 E1021433 E1021443 E1021453 E1021463 E1021473 E1021483 E1021493;
array E102144 E1021414 E1021424 E1021434 E1021444 E1021454 E1021464 E1021474 E1021484 E1021494;
array E102145 E1021415 E1021425 E1021435 E1021445 E1021455 E1021465 E1021475 E1021485 E1021495;
array E226041 E2260411 E2260421 E2260431 E2260441 E2260451 E2260461 E2260471 E2260481 E2260491;
array E226042 E2260412 E2260422 E2260432 E2260442 E2260452 E2260462 E2260472 E2260482 E2260492;
array E226043 E2260413 E2260423 E2260433 E2260443 E2260453 E2260463 E2260473 E2260083 E2260493;
array E226044 E2260414 E2260424 E2260434 E2260444 E2260454 E2260464 E2260474 E2260484 E2260494;
array E226045 E2260415 E2260425 E2260435 E2260445 E2260455 E2260465 E2260475 E2260485 E2260495;
array E226046 E2260416 E2260426 E2260436 E2260446 E2260456 E2260466 E2260476 E2260486 E2260496;
array E226047 E2260417 E2260427 E2260437 E2260447 E2260457 E2260467 E2260477 E2260487 E2260497;
array E226048 E2260418 E2260428 E2260438 E2260448 E2260458 E2260468 E2260478 E2260088 E2260498;
array E226049 E2260419 E2260429 E2260439 E2260449 E2260459 E2260469 E2260479 E2260489 E2260499;
array E226040 E2260410 E2260420 E2260430 E2260440 E2260450 E2260460 E2260470 E2260480 E2260490;
array E102251 E1022511 E1022521 E1022531 E1022541 E1022551 E1022561 E1022571 E1022581 E1022591;
array E102252 E1022512 E1022522 E1022532 E1022542 E1022552 E1022562 E1022572 E1022582 E1022592;
array E102253 E1022513 E1022523 E1022533 E1022543 E1022553 E1022563 E1022573 E2260083 E1022593;
array E102254 E1022514 E1022524 E1022534 E1022544 E1022554 E1022564 E1022574 E1022584 E1022594;
array E102255 E1022515 E1022525 E1022535 E1022545 E1022555 E1022565 E1022575 E1022585 E1022595;
array E102256 E1022516 E1022526 E1022536 E1022546 E1022556 E1022566 E1022576 E1022586 E1022596;
array E102257 E1022517 E1022527 E1022537 E1022547 E1022557 E1022567 E1022577 E1022587 E1022597;
array E102258 E1022518 E1022528 E1022538 E1022548 E1022558 E1022568 E1022578 E2260088 E1022598;
array E102259 E1022519 E1022529 E1022539 E1022549 E1022559 E1022569 E1022579 E1022589 E1022599;
array E102250 E1022510 E1022520 E1022530 E1022540 E1022550 E1022560 E1022570 E1022580 E1022590;

/****** Section 1: START WAGES FOR THE YOUTH *****/

/*** Part 1: Start hourly rate of pay *****/

/* For each time unit, the information could either be given in the earlier part (from E19200) or in the later part (from E83100); it also depends on whether the respondent has other compensation. */

/* Respondents reporting an HOURLY wage*/

HRWAGE01=-4;	HRWAGE02=-4;	HRWAGE03=-4;
HRWAGE04=-4;	HRWAGE05=-4;	HRWAGE06=-4;
HRWAGE07=-4;	HRWAGE08=-4;	HRWAGE09=-4;

Appendix 2: Employment Variable Creation

```
array HRWAGE HRWAGE01 HRWAGE02 HRWAGE03 HRWAGE04 HRWAGE05 HRWAGE06
      HRWAGE07 HRWAGE08 HRWAGE09;

do I=1 to 9;
  if E19200[I]=1 then do;
    if (E22900[I]>=0) then HRWAGE[I]=E22900[I];
    if (E23000[I]>=0) then HRWAGE[I]=E23000[I];
    if (E22900[I]=-2 or E23000[I]=-2) then HRWAGE[I]=E23200[I];
    if (E22900[I]=-3 or E23000[I]=-3) then HRWAGE[I]=E23200[I];
    if (E22900[I]=-1 or E23000[I]=-1) then HRWAGE[I]=-1;
    if E23900[I]>=0 then HRWAGE[I]=E23900[I];
  end;

  if E83100[I]=1 then do;
    if (E86800[I]>=0) then HRWAGE[I]=E86800[I];
    if (E86900[I]>=0) then HRWAGE[I]=E86900[I];
    if (E86800[I]=-2 or E86900[I]=-2) then HRWAGE[I]=E87100[I];
    if (E86800[I]=-3 or E86900[I]=-3) then HRWAGE[I]=E87100[I];
    if (E86800[I]=-1 or E86900[I]=-1) then HRWAGE[I]=-1;
    if E87800[I]>=0 then HRWAGE[I]=E87800[I];
  end;
end;

/*Respondents reporting a DAILY wage*/
DAILY01=-4;      DAILY02=-4;      DAILY03=-4;
DAILY04=-4;      DAILY05=-4;      DAILY06=-4;
DAILY07=-4;      DAILY08=-4;      DAILY09=-4;

array DAILY DAILY01 DAILY02 DAILY03 DAILY04 DAILY05 DAILY06 DAILY07 DAILY08 DAILY09;

do I=1 to 9; /* daily start wage divided by the number of hours worked per week*/
  if E19200[I]=2 then do;

    /*no compensation*/
    if (E33400[I]>=0 and E34402[I]>0 and E34402B[I]>0) then DAILY[I]=(E33400[I]*E34402B[I]/E34402[I]);
    if (E33400[I]=-2 and E33600[I]>=0 and E34402[I]>0 and E34402B[I]>0) then
      DAILY[I]=(E33600[I]*E34402B[I]/E34402[I]);
    if E34400[I]>=0 and E34402[I]>0 and E34402B[I]>0 then DAILY[I]=E34400[I]*E34402B[I]/E34402[I];
    /* missing value*/ if -4<E34402[I]<0 then DAILY[I]=E34402[I];

    /*received compensation*/
    /*without overtime*/
    if (E33500[I]>=0 and E34402[I]>0 and E34402B[I]>0) then DAILY[I]=(E33500[I]*E34402B[I]/E34402[I]);
    if (E33500[I]=-2 and E33600[I]>=0 and E34402[I]>0 and E34402B[I]>0) then
      DAILY[I]=(E33600[I]*E34402B[I]/E34402[I]);
    if E34400[I]>=0 and E34402[I]>0 and E34402B[I]>0 then DAILY[I]=E34400[I]*E34402B[I]/E34402[I];
    /*missing values*/
    if -4<E34402[I]<0 then DAILY[I]=E34402[I];
    if (E33400[I]>=0 or E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34402[I]=0 then DAILY[I]=-3;
    if -4<E34402B[I]<0 then DAILY[I]=E34402B[I];
    if E34402B[I]=0 then DAILY[I]=-3;

    /*with overtime*/
    if (E33500[I]>=0 and E34428[I]>0 and E34430[I]>0) then DAILY[I]=(E33500[I]*E34430[I]/E34428[I]);
    if (E33500[I]=-2 and E33600[I]>=0 and E34428[I]>0 and E34430[I]>0) then
      DAILY[I]=(E33600[I]*E34430[I]/E34428[I]);
    if E34400[I]>=0 and E34428[I]>0 and E34430[I]>0 then DAILY[I]=E34400[I]*E34430[I]/E34428[I];
```

Appendix 2: Employment Variable Creation

```
/*missing values*/
if -4<E34428[I]<0 then DAILY[I]=E34428[I];
if (E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34428[I]=0 then DAILY[I]=-3;
if -4<E34430[I]<0 then DAILY[I]=E34430[I];
if E34430[I]=0 then DAILY[I]=-3;

/*if still paid hourly...*/
if E36200[I]>=0 then DAILY[I]=E36200[I];
if E36100[I]>=0 then DAILY[I]=E36100[I];
end;

if E83100[I]=2 then do;

/*no compensation*/
if (E97300[I]>=0 and E98402[I]>0 and E98402D[I]>0) then DAILY[I]=(E97300[I]*E98402D[I]/E98402[I]);
if (E97300[I]=-2 and E97500[I]>=0 and E98402[I]>0 and E98402D[I]>0) then
    DAILY[I]=(E97500[I]*E98402D[I]/E98402[I]);
if E98300[I]>=0 and E98402[I]>0 and E98402D[I]>0 then DAILY[I]=E98300[I]*E98402D[I]/E98402[I];
/* missing value*/ if -4<E98402[I]<0 then DAILY[I]=E98402[I];

/*received compensation*/
/*with overtime */
if (E97400[I]>=0 and E98429[I]>0 and E98429E[I]>0) then DAILY[I]=(E97400[I]*E98429E[I]/E98429[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98429[I]>0 and E98429E[I]>0) then
    DAILY[I]=(E97500[I]*E98429E[I]/E98429[I]);
if E98300[I]>=0 and E98429[I]>0 and E98429E[I]>0 then DAILY[I]=E98300[I]*E98429E[I]/E98429[I];
/*missing values*/
if -4<E98429[I]<0 then DAILY[I]=E98429[I];
if (E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98429[I]=0 then DAILY[I]=-3;
if -4<E98429E[I]<0 then DAILY[I]=E98429E[I];
if E98429E[I]=0 then DAILY[I]=-3;

/* without overtime */
if (E97400[I]>=0 and E98402[I]>0 and E98402D[I]>0) then DAILY[I]=(E97400[I]*E98402D[I]/E98402[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98402[I]>0 and E98402D[I]>0) then
    DAILY[I]=(E97500[I]*E98402D[I]/E98402[I]);
if E98300[I]>=0 and E98402[I]>0 and E98402D[I]>0 then DAILY[I]=E98300[I]*E98402D[I]/E98402[I];
/*missing values*/
if -4<E98402[I]<0 then DAILY[I]=E98402[I];
if (E97300[I]>=0 or E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98402[I]=0 then DAILY[I]=-3;
if -4<E98402D[I]<0 then DAILY[I]=E98402D[I];
if E98402D[I]=0 then DAILY[I]=-3;

/*if still paid hourly...*/
if E100100[I]>=0 then DAILY[I]=E100100[I];
if E100000[I]>=0 then DAILY[I]=E100000[I];
end;
end;

/*Respondents reporting a WEEKLY wage*/
WEEKLY01=-4;      WEEKLY02=-4;      WEEKLY03=-4;
WEEKLY04=-4;      WEEKLY05=-4;      WEEKLY06=-4;
WEEKLY07=-4;      WEEKLY08=-4;      WEEKLY09=-4;

array WEEKLY WEEKLY01 WEEKLY02 WEEKLY03 WEEKLY04 WEEKLY05 WEEKLY06
        WEEKLY07 WEEKLY08 WEEKLY09;
```

Appendix 2: Employment Variable Creation

```
do I=1 to 9; /* weekly start wage divided by the number of hours worked per week*/
if E19200[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) then do;

/*no compensation*/
if (E33400[I]>=0 and E34402[I]>0) then WEEKLY[I]=(E33400[I]/E34402[I]);
if (E33400[I]=-2 and E33600[I]>=0 and E34402[I]>0) then WEEKLY[I]=(E33600[I]/E34402[I]);
if E34400[I]>=0 and E34402[I]>0 then WEEKLY[I]=E34400[I]/E34402[I];
/*missing value*/ if -4<E34402[I]<0 then WEEKLY[I]=E34402[I];

/*received compensation*/
/*with overtime*/
if (E33500[I]>=0 and E34428[I]>0) then WEEKLY[I]=(E33500[I]/E34428[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34428[I]>0) then WEEKLY[I]=(E33600[I]/E34428[I]);
if E34400[I]>=0 and E34428[I]>0 then WEEKLY[I]=E34400[I]/E34428[I];
/*missing values*/
if -4<E34428[I]<0 then WEEKLY[I]=E34428[I];
if (E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34428[I]=0 then WEEKLY[I]=-3;

/*without overtime*/
if (E33500[I]>=0 and E34402[I]>0) then WEEKLY[I]=(E33500[I]/E34402[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34402[I]>0) then WEEKLY[I]=(E33600[I]/E34402[I]);
if E34400[I]>=0 and E34402[I]>0 then WEEKLY[I]=E34400[I]/E34402[I];
/*missing values*/
if -4<E34402[I]<0 then WEEKLY[I]=E34402[I];
if (E33400[I]>=0 or E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34402[I]=0 then WEEKLY[I]=-3;

/*if still paid hourly...*/
if E36200[I]>=0 then WEEKLY[I]=E36200[I];
if E36100[I]>=0 then WEEKLY[I]=E36100[I];
end;

if E83100[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) then do;

/*no compensation*/
if (E97300[I]>=0 and E98402[I]>0) then WEEKLY[I]=(E97300[I]/E98402[I]);
if (E97300[I]=-2 and E97500[I]>=0 and E98402[I]>0) then WEEKLY[I]=(E97500[I]/E98402[I]);
if E98300[I]>=0 and E98402[I]>0 then WEEKLY[I]=E98300[I]/E98402[I];
/*missing value*/ if -4<E98402[I]<0 then WEEKLY[I]=E98402[I];

/*received compensation*/
/*with overtime*/
if (E97400[I]>=0 and E98429[I]>0) then WEEKLY[I]=(E97400[I]/E98429[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98429[I]>0) then WEEKLY[I]=(E97500[I]/E98429[I]);
if E98300[I]>=0 and E98429[I]>0 then WEEKLY[I]=E98300[I]/E98429[I];
/*missing values*/
if -4<E98429[I]<0 then WEEKLY[I]=E98429[I];
if (E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98429[I]=0 then WEEKLY[I]=-3;

/*without overtime*/
if (E97400[I]>=0 and E98402[I]>0) then WEEKLY[I]=(E97400[I]/E98402[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98402[I]>0) then WEEKLY[I]=(E97500[I]/E98402[I]);
if E98300[I]>=0 AND E98402[I]>0 then WEEKLY[I]=E98300[I]/E98402[I];
/*missing values*/
if -4<E98402[I]<0 then WEEKLY[I]=E98402[I];
if (E97300[I]>=0 or E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98402[I]=0 then WEEKLY[I]=-3;

/*if still paid hourly...*/
```

Appendix 2: Employment Variable Creation

```
if E100100[I]>=0 then WEEKLY[I]=E100100[I];
if E100000[I]>=0 then WEEKLY[I]=E100000[I];
end;
end;

/*Respondents reporting a BIWEEKLY wage*/
BIWKLY01=-4;      BIWKLY02=-4;      BIWKLY03=-4;
BIWKLY04=-4;      BIWKLY05=-4;      BIWKLY06=-4;
BIWKLY07=-4;      BIWKLY08=-4;      BIWKLY09=-4;

array BIWKLY BIWKLY01 BIWKLY02 BIWKLY03 BIWKLY04 BIWKLY05 BIWKLY06 BIWKLY07
      BIWKLY08 BIWKLY09;

do I=1 to 9; /* biwkly start wage divided by the number of hours worked per week*/
if E19200[I]=4 then do;

/*no compensation*/
if (E33400[I]>=0 and E34402[I]>0) then BIWKLY[I]=E33400[I]/(2*E34402[I]);
if (E33400[I]=-2 and E33600[I]>=0 and E34402[I]>0) then BIWKLY[I]=E33600[I]/(2*E34402[I]);
if E34400[I]>=0 and E34402[I]>0 then BIWKLY[I]=E34400[I]/(2*E34402[I]);
/*missing value*/ if -4<E34402[I]<0 then BIWKLY[I]=E34402[I];

/*received compensation*/
/*with overtime*/
if (E33500[I]>=0 and E34428[I]>0) then BIWKLY[I]=E33500[I]/(2*E34428[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34428[I]>0) then BIWKLY[I]=E33600[I]/(2*E34428[I]);
if E34400[I]>=0 and E34428[I]>0 then BIWKLY[I]=E34400[I]/(2*E34428[I]);
/*missing values*/
if -4<E34428[I]<0 then BIWKLY[I]=E34428[I];
if (E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34428[I]=0 then BIWKLY[I]=-3;

/*without overtime*/
if (E33500[I]>=0 and E34402[I]>0) then BIWKLY[I]=E33500[I]/(2*E34402[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34402[I]>0) then BIWKLY[I]=E33600[I]/(2*E34402[I]);
if E34400[I]>=0 and E34402[I]>0 then BIWKLY[I]=E34400[I]/(2*E34402[I]);
/*missing values*/
if -4<E34402[I]<0 then BIWKLY[I]=E34402[I];
if (E33400[I]>=0 or E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34402[I]=0 then BIWKLY[I]=-3;

/*if still paid hourly...*/
if E36200[I]>=0 then BIWKLY[I]=E36200[I];
if E36100[I]>=0 then BIWKLY[I]=E36100[I];
end;

if E83100[I]=4 then do;

/*no compensation*/
if (E97300[I]>=0 and E98402[I]>0) then BIWKLY[I]=E97300[I]/(2*E98402[I]);
if (E97300[I]=-2 and E97500[I]>=0 and E98402[I]>0) then BIWKLY[I]=E97500[I]/(2*E98402[I]);
if E98300[I]>=0 and E98402[I]>0 then BIWKLY[I]=E98300[I]/(2*E98402[I]);
/*missing value*/ if -4<E98402[I]<0 then BIWKLY[I]=E98402[I];

/*received compensation*/
/*with overtime*/
if (E97400[I]>=0 and E98429[I]>0) then BIWKLY[I]=E97400[I]/(2*E98429[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98429[I]>0) THEN BIWKLY[I]=E97500[I]/(2*E98429[I]);
if E98300[I]>=0 and E98429[I]>0 then BIWKLY[I]=E98300[I]/(2*E98429[I]);
```

Appendix 2: Employment Variable Creation

```
/*missing values*/
if -4<E98429[I]<0 then BIWKLY[I]=E98429[I];
if (E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98429[I]=0 then BIWKLY[I]=-3;

/*without overtime*/
if (E97400[I]>=0 and E98402[I]>0) then BIWKLY[I]=E97400[I]/(2*E98402[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98402[I]>0) then BIWKLY[I]=E97500[I]/(2*E98402[I]);
if E98300[I]>=0 and E98402[I]>0 then BIWKLY[I]=E98300[I]/(2*E98402[I]);
/*missing values*/
if -4<E98402[I]<0 then BIWKLY[I]=E98402[I];
if (E97300[I]>=0 or E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98402[I]=0 then BIWKLY[I]=-3;

/*if still paid hourly...*/
if E100100[I]>=0 then BIWKLY[I]=E100100[I];
if E100000[I]>=0 then BIWKLY[I]=E100000[I];
end;
end;

/*Respondents reporting a MONTHLY wage*/
MONTH01=-4;      MONTH02=-4;      MONTH03=-4;
MONTH04=-4;      MONTH05=-4;      MONTH06=-4;
MONTH07=-4;      MONTH08=-4;      MONTH09=-4;

array MONTH MONTH01 MONTH02 MONTH03 MONTH04 MONTH05 MONTH06 MONTH07
      MONTH08 MONTH09;

do I=1 to 9; /* month start wage divided by the number of hours worked per week*/
if E19200[I]=5 then do;

/*no compensation*/
if (E33400[I]>=0 and E34402[I]>0) then MONTH[I]=E33400[I]/(4.3*E34402[I]);
if (E33400[I]=-2 and E33600[I]>=0 and E34402[I]>0) then MONTH[I]=E33600[I]/(4.3*E34402[I]);
if E34400[I]>=0 and E34402[I]>0 then MONTH[I]=E34400[I]/(4.3*E34402[I]);
/*missing value*/ if -4<E34402[I]<0 then MONTH[I]=E34402[I];

/*received compensation*/
/*with overtime*/
if (E33500[I]>=0 and E34428[I]>0) then MONTH[I]=E33500[I]/(4.3*E34428[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34428[I]>0) then MONTH[I]=E33600[I]/(4.3*E34428[I]);
if E34400[I]>=0 and E34428[I]>0 then MONTH[I]=E34400[I]/(4.3*E34428[I]);
/*missing values*/
if -4<E34428[I]<0 then MONTH[I]=E34428[I];
if (E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34428[I]=0 then MONTH[I]=-3;

/*without overtime*/
if (E33500[I]>=0 and E34402[I]>0) then MONTH[I]=E33500[I]/(4.3*E34402[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34402[I]>0) then MONTH[I]=E33600[I]/(4.3*E34402[I]);
if E34400[I]>=0 and E34402[I]>0 then MONTH[I]=E34400[I]/(4.3*E34402[I]);
/*missing values*/
if -4<E34402[I]<0 then MONTH[I]=E34402[I];
if (E33400[I]>=0 or E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34402[I]=0 then MONTH[I]=-3;

/*if still paid hourly...*/
if E36200[I]>=0 then MONTH[I]=E36200[I];
if E36100[I]>=0 then MONTH[I]=E36100[I];
end;
```


Appendix 2: Employment Variable Creation

```
if E83100[I]=5 then do;

/*no compensation*/
  if (E97300[I]>=0 and E98402[I]>0) then MONTH[I]=E97300[I]/(4.3*E98402[I]);
  if (E97300[I]=-2 and E97500[I]>=0 and E98402[I]>0) then MONTH[I]=E97500[I]/(4.3*E98402[I]);
  if E98300[I]>=0 and E98402[I]>0 then MONTH[I]=E98300[I]/(4.3*E98402[I]);
  /*missing value*/ if -4<E98402[I]<0 then MONTH[I]=E98402[I];

/*received compensation*/
/*with overtime*/
  if (E97400[I]>=0 and E98429[I]>0) then MONTH[I]=E97400[I]/(4.3*E98429[I]);
  if (E97400[I]=-2 and E97500[I]>=0 and E98429[I]>0) then MONTH[I]=E97500[I]/(4.3*E98429[I]);
  if E98300[I]>=0 and E98429[I]>0 then MONTH[I]=E98300[I]/(4.3*E98429[I]);
  /*missing values*/
  if -4<E98429[I]<0 then MONTH[I]=E98429[I];
  if (E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98429[I]=0 then MONTH[I]=-3;

/*without overtime*/
  if (E97400[I]>=0 and E98402[I]>0) then MONTH[I]=E97400[I]/(4.3*E98402[I]);
  if (E97400[I]=-2 and E97500[I]>=0 and E98402[I]>0) then MONTH[I]=E97500[I]/(4.3*E98402[I]);
  if E98300[I]>=0 and E98402[I]>0 then MONTH[I]=E98300[I]/(4.3*E98402[I]);
  /*missing values*/
  if -4<E98402[I]<0 then MONTH[I]=E98402[I];
  if (E97300[I]>=0 or E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98402[I]=0 then MONTH[I]=-3;

/*if still paid hourly...*/
  if E100100[I]>=0 then MONTH[I]=E100100[I];
  if E100000[I]>=0 then MONTH[I]=E100000[I];
end;
end;

/*Respondents reporting an ANNUAL wage*/
ANNUAL01=-4;      ANNUAL02=-4;      ANNUAL03=-4;
ANNUAL04=-4;      ANNUAL05=-4;      ANNUAL06=-4;
ANNUAL07=-4;      ANNUAL08=-4;      ANNUAL09=-4;

array ANNUAL ANNUAL01 ANNUAL02 ANNUAL03 ANNUAL04 ANNUAL05 ANNUAL06
      ANNUAL07 ANNUAL08 ANNUAL09;

do I=1 to 9; /* annual start wage divided by the number of hours worked per week*/
if E19200[I]=6 then do;

/*no compensation*/
  if (E33400[I]>=0 and E34402[I]>0 and E35600[I]>0) then ANNUAL[I]=E33400[I]/(E35600[I]*E34402[I]);
  if (E33400[I]=-2 and E33600[I]>=0 and E34402[I]>0 and E35600[I]>0) then
    ANNUAL[I]=E33600[I]/(E35600[I]*E34402[I]);
  if E34400[I]>=0 and E34402[I]>0 and E35600[I]>0 then ANNUAL[I]=E34400[I]/(E35600[I]*E34402[I]);
  /*missing value*/ if -4<E34402[I]<0 then ANNUAL[I]=E34402[I];

/*received compensation*/
/*with overtime*/
  if (E33500[I]>=0 and E34428[I]>0 and E35600[I]>0) then ANNUAL[I]=E33500[I]/(E35600[I]*E34428[I]);
  if (E33500[I]=-2 and E33600[I]>=0 and E34428[I]>0 and E35600[I]>0) then
    ANNUAL[I]=E33600[I]/(E35600[I]*E34428[I]);
  if E34400[I]>=0 and E34428[I]>0 and E35600[I]>0 then ANNUAL[I]=E34400[I]/(E35600[I]*E34428[I]);
  /*missing values*/
  if -4<E34428[I]<0 then ANNUAL[I]=E34428[I];
```

Appendix 2: Employment Variable Creation

```
if (E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34428[I]=0 then ANNUAL[I]=-3;

/*without overtime*/
if (E33500[I]>=0 and E34402[I]>0 and E35600[I]>0) then ANNUAL[I]=E33500[I]/(E35600[I]*E34402[I]);
if (E33500[I]=-2 and E33600[I]>=0 and E34402[I]>0 and E35600[I]>0) then
  ANNUAL[I]=E33600[I]/(E35600[I]*E34402[I]);
if E34400[I]>=0 and E34402[I]>0 and E35600[I]>0 then ANNUAL[I]=E34400[I]/(E35600[I]*E34402[I]);
/*missing values*/
if -4<E34402[I]<0 then ANNUAL[I]=E34402[I];
if (E33400[I]>=0 or E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34402[I]=0 then ANNUAL[I]=-3;
if -4<E35600[I]<0 then ANNUAL[I]=E35600[I];
if E35600[I]=0 then ANNUAL[I]=-3;

/*if still paid hourly...*/
if E36200[I]>=0 then ANNUAL[I]=E36200[I];
if E36100[I]>=0 then ANNUAL[I]=E36100[I];
end;

if E83100[I]=6 then do;

/*no compensation*/
if (E97300[I]>=0 and E98402[I]>0 and E99500[I]>0) then ANNUAL[I]=E97300[I]/(E99500[I]*E98402[I]);
if (E97300[I]=-2 and E97500[I]>=0 and E98402[I]>0 and E99500[I]>0) then
  ANNUAL[I]=E97500[I]/(E99500[I]*E98402[I]);
if E98300[I]>=0 and E98402[I]>0 and E99500[I]>0 then ANNUAL[I]=E98300[I]/(E99500[I]*E98402[I]);
/*missing value*/ if -4<E98402[I]<0 then ANNUAL[I]=E98402[I];

/*received compensation*/
/*with overtime*/
if (E97400[I]>=0 and E98429[I]>0 and E99500[I]>0) then ANNUAL[I]=E97400[I]/(E99500[I]*E98429[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98429[I]>0 and E99500[I]>0) then
  ANNUAL[I]=E97500[I]/(E99500[I]*E98429[I]);
if E98300[I]>=0 and E98429[I]>0 and E99500[I]>0 then ANNUAL[I]=E98300[I]/(E99500[I]*E98429[I]);
/*missing values*/
if -4<E99500[I]<0 then ANNUAL[I]=E99500[I];
if -4<E98429[I]<0 then ANNUAL[I]=E98429[I];
if (E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98429[I]=0 then ANNUAL[I]=-3;

/* without overtime*/
if (E97400[I]>=0 and E98402[I]>0 and E99500[I]>0) then ANNUAL[I]=E97400[I]/(E99500[I]*E98402[I]);
if (E97400[I]=-2 and E97500[I]>=0 and E98402[I]>0 and E99500[I]>0) then
  ANNUAL[I]=E97500[I]/(E99500[I]*E98402[I]);
if E98300[I]>=0 and E98402[I]>0 and E99500[I]>0 then ANNUAL[I]=E98300[I]/(E99500[I]*E98402[I]);
/*missing values*/
if -4<E99500[I]<0 then ANNUAL[I]=E99500[I];
if -4<E98402[I]<0 then ANNUAL[I]=E98402[I];
if (E97300[I]>=0 or E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98402[I]=0 then ANNUAL[I]=-3;
if -4<E99500[I]<0 then ANNUAL[I]=E99500[I];
if E99500[I]=0 then ANNUAL[I]=-3;

/*if still paid hourly...*/
if E100100[I]>=0 then ANNUAL[I]=E100100[I];
if E100000[I]>=0 then ANNUAL[I]=E100000[I];
end;
end;

/*Respondents reporting a SEMIMONTHLY wage*/
```

Appendix 2: Employment Variable Creation

```
SEMIM01=-4;          SEMIM02=-4;          SEMIM03=-4;
SEMIM04=-4;          SEMIM05=-4;          SEMIM06=-4;
SEMIM07=-4;          SEMIM08=-4;          SEMIM09=-4;

array SEMIM SEMIM01 SEMIM02 SEMIM03 SEMIM04 SEMIM05 SEMIM06 SEMIM07 SEMIM08
      SEMIM09;

do I=1 to 9; /* semim start wage divided by the number of hours worked per week*/
if E19200[I]=8 then do;

/*no compensation*/
  if (E33400[I]>=0 and E34402[I]>0) then SEMIM[I]=E33400[I]/(2.15*E34402[I]);
  if (E33400[I]=-2 and E33600[I]>=0 and E34402[I]>0) then SEMIM[I]=E33600[I]/(2.15*E34402[I]);
  if E34400[I]>=0 and E34402[I]>0 then SEMIM[I]=E34400[I]/(2.15*E34402[I]);
  /*missing value*/ if -4<E34402[I]<0 then SEMIM[I]=E34402[I];

/*received compensation*/
/*with overtime*/
  if (E33500[I]>=0 and E34428[I]>0) then SEMIM[I]=E33500[I]/(2.15*E34428[I]);
  if (E33500[I]=-2 and E33600[I]>=0 and E34428[I]>0) then SEMIM[I]=E33600[I]/(2.15*E34428[I]);
  if E34400[I]>=0 and E34428[I]>0 then SEMIM[I]=E34400[I]/(2.15*E34428[I]);
  /*missing values*/
  if -4<E34428[I]<0 then SEMIM[I]=E34428[I];
  if (E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34428[I]=0 then SEMIM[I]=-3;

/*without overtime*/
  if (E33500[I]>=0 and E34402[I]>0) then SEMIM[I]=E33500[I]/(2.15*E34402[I]);
  if (E33500[I]=-2 and E33600[I]>=0 and E34402[I]>0) then SEMIM[I]=E33600[I]/(2.15*E34402[I]);
  if E34400[I]>=0 and E34402[I]>0 then SEMIM[I]=E34400[I]/(2.15*E34402[I]);
  /*missing values*/
  if -4<E34402[I]<0 then SEMIM[I]=E34402[I];
  if (E33400[I]>=0 or E33500[I]>=0 or E33600[I]>=0 or E34400[I]>=0) and E34402[I]=0 then SEMIM[I]=-3;

/*if still paid hourly...*/
  if E36200[I]>=0 then SEMIM[I]=E36200[I];
  if E36100[I]>=0 then SEMIM[I]=E36100[I];
end;

if E83100[I]=8 then do;

/*no compensation*/
  if (E97300[I]>=0 and E98402[I]>0) then SEMIM[I]=E97300[I]/(2.15*E98402[I]);
  if (E97300[I]=-2 and E97500[I]>=0 and E98402[I]>0) then SEMIM[I]=E97500[I]/(2.15*E98402[I]);
  if E98300[I]>=0 and E98402[I]>0 then SEMIM[I]=E98300[I]/(2.15*E98402[I]);
  /*missing value*/ if -4<E98402[I]<0 then SEMIM[I]=E98402[I];

/*received compensation*/
/*with overtime*/
  if (E97400[I]>=0 and E98429[I]>0) then SEMIM[I]=E97400[I]/(2.15*E98429[I]);
  if (E97400[I]=-2 and E97500[I]>=0 and E98429[I]>0) then SEMIM[I]=E97500[I]/(2.15*E98429[I]);
  if E98300[I]>=0 and E98429[I]>0 then SEMIM[I]=E98300[I]/(2.15*E98429[I]);
  /*missing values*/
  if -4<E98429[I]<0 then SEMIM[I]=E98429[I];
  if (E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98429[I]=0 then SEMIM[I]=-3;

/*without overtime*/
  if (E97400[I]>=0 and E98402[I]>0) then SEMIM[I]=E97400[I]/(2.15*E98402[I]);
```

Appendix 2: Employment Variable Creation

```
if (E97400[I]=-2 and E97500[I]>=0 and E98402[I]>0) then SEMIM[I]=E97500[I]/(2.15*E98402[I]);
if E98300[I]>=0 and E98402[I]>0 then SEMIM[I]=E98300[I]/(2.15*E98402[I]);
/*missing values*/
if -4<E98402[I]<0 then SEMIM[I]=E98402[I];
if (E97300[I]>=0 or E97400[I]>=0 or E97500[I]>=0 or E98300[I]>=0) and E98402[I]=0 then SEMIM[I]=-3;

/*if still paid hourly...*/
if E100100[I]>=0 then SEMIM[I]=E100100[I];
if E100000[I]>=0 then SEMIM[I]=E100000[I];
end;
end;

/*Respondents reporting an OTHER wage*/
OTHERF1=0;          OTHERF2=0;          OTHERF3=0;
OTHERF4=0;          OTHERF5=0;          OTHERF6=0;
OTHERF7=0;          OTHERF8=0;          OTHERF9=0;

array OTHERF OTHERF1-OTHERF9;
do I=1 to 9;

if E19200[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) or E83100[I] in
(0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHERF[I]=OTHERF[I]+1;
end;

**** Part 2: Create the Hourly Rate of Pay based on the start wage ****/

HRWG01=-4;          HRWG02=-4;          HRWG03=-4;
HRWG04=-4;          HRWG05=-4;          HRWG06=-4;
HRWG07=-4;          HRWG08=-4;          HRWG09=-4;

array HRWG HRWG01 HRWG02 HRWG03 HRWG04 HRWG05 HRWG06 HRWG07 HRWG08 HRWG09;

do I=1 to 9; /* Report hourly wage to be 0 for the family business*/
if E19200[I] in (9,14) or E83100[I] in (9,14) then HRWG[I]=0;
end;

do I=1 to 9; /* report hourly rate -1 or -2 if amount is -1 or -2*/
if E19200[I] IN (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,25,28,-1,-2) then do;
if E33400[I]=-1 or E33500[I]=-1 then HRWG[I]=-1;
if (E33400[I]=-2 and E33600[I]=-2) or (E33500[I]=-2 and E33600[I]=-2) then HRWG[I]=-2;
if E33400[I]=-3 or E33500[I]=-3 then HRWG[I]=-3;
end;

if E83100[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,25,28,-1,-2) then do;
if E97300[I]=-1 or E97400[I]=-1 then HRWG[I]=-1;
if (E97300[I]=-2 and E97500[I]=-2) or (E97400[I]=-2 and E97500[I]=-2) then HRWG[I]=-2;
if E97300[I]=-3 or E97400[I]=-3 then HRWG[I]=-3;
end;
end;

do I=1 to 9; /* report hourly rate -3 if no hours reported*/
if E212001[I]=1 and E34428[I]=-4 then HRWG[I]=-3;
end;

do I=1 to 9;

if ANNUAL[I] ge 0 then HRWG[I]=ANNUAL[I];    f MONTH[I] ge 0 then HRWG[I]=MONTH[I];
```

Appendix 2: Employment Variable Creation

```
if BIWKLY[I] ge 0 then HRWG[I]=BIWKLY[I];      if WEEKLY[I] ge 0 then HRWG[I]=WEEKLY[I];
if DAILY[I] ge 0 then HRWG[I]=DAILY[I];      if HRWAGE[I] ge 0 then HRWG[I]=HRWAGE[I];
if SEMIM[I] ge 0 then HRWG[I]=SEMIM[I];
if HRWAGE[I] eq -1 or DAILY[I]=-1 or WEEKLY[I] eq -1 or BIWKLY[I] eq -1 or MONTH[I] eq -1 or
  ANNUAL[I] eq -1 or SEMIM[I]=-1 then HRWG[I]=-1;
if HRWAGE[I] eq -2 or DAILY[I]=-2 or WEEKLY[I] eq -2 or BIWKLY[I] eq -2 or MONTH[I] eq -2 or
  ANNUAL[I] eq -2 or SEMIM[I]=-2 then HRWG[I]=-2;
if HRWAGE[I] eq -3 or DAILY[I]=-3 or WEEKLY[I] eq -3 or BIWKLY[I] eq -3 or MONTH[I] eq -3 or
  ANNUAL[I] eq -3 or SEMIM[I]=-3 then HRWG[I]=-3;
end;
```

******Part 3: Report the corrected wage if the correction is made ****/**

```
do I=1 to 9;
if E226041[I]=1 and E226042[I]=0 then do; /* rate incorrect but hours correct*/
  /* no overtime, use E34402 for hours */
  if E22609[I]=1 and E22626[I]>=0 then HRWG[I]=E22626[I];
  if E22609[I]=2 and E34402[I]>0 and E34402B[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]*E34402B[I]/E34402[I];
  if E22609[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E34402[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]/E34402[I];
  if E22609[I]=4 and E34402[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(2*E34402[I]);
  if E22609[I]=5 and E34402[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(4.3*E34402[I]);
  if E22609[I]=6 and E34402[I]>0 and E35600[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]/(E35600[I]*E34402[I]);
  if E22609[I]=8 and E34402[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(2.15*E34402[I]);
  if E22609[I] in (9,14) then HRWG[I]=0;
  if E22609[I]=2 and E34402B[I] le 0 then HRWG[I]=-3;
  if E22609[I]=2 and -4<E34402B[I]<0 then HRWG[I]=E34402B[I];
  if E22609[I]=6 and E35600[I] le 0 then HRWG[I]=-3;
  if E22609[I]=6 and -4<E35600[I]<0 then HRWG[I]=E35600[I];
  if E22609[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,223,24,25,25,28,-1,-2) and -4<E22626[I]<0 then
    HRWG[I]=E22626[I];
  if E22609[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,223,24,25,25,28,-2) and E34402[I]=0 then
    HRWG[I]=-3;
  if E22609[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,223,24,25,25,28,-2) and -4<E34402[I]<0 then
    HRWG[I]=E34402[I];

  /* overtime, use E34428 for hours*/
  if E22609[I]=1 and E22626[I]>=0 then HRWG[I]=E22626[I];
  if E22609[I]=2 and E34428[I]>0 and E34430[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]*E34430[I]/E34428[I];
  if E22609[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E34428[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]/E34428[I];
  if E22609[I]=4 and E34428[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(2*E34428[I]);
  if E22609[I]=5 and E34428[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(4.3*E34428[I]);
  if E22609[I]=6 and E34428[I]>0 and E35600[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]/(E35600[I]*E34428[I]);
  if E22609[I]=8 and E34428[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(2.15*E34428[I]);
  if E22609[I] in (9,14) then HRWG[I]=0;
  if E22609[I]=2 and E34430[I] le 0 then HRWG[I]=-3;
  if E22609[I]=2 and -4<E34430[I]<0 then HRWG[I]=E34430[I];
  if E22609[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,223,24,25,25,28,-1,-2) and -4<E22626[I]<0 then
    HRWG[I]=E22626[I];
  if E22609[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,223,24,25,25,28,-2) and E34428[I]=0 then
    HRWG[I]=-3;
```

Appendix 2: Employment Variable Creation

```
if E22609[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,223,24,25,25,28,-2) and -4<E34428[I]<0 then
  HRWG[I]=E34428[I];
end;

if E226041[I]=0 and E226042[I]=1 then do; /*rate correct but hours incorrect*/
  if E19200[I] ne 1 and E22610[I]>0 and E34402[I]>0 and HRWG[I]>=0 then
    HRWG[I]=HRWG[I]*E34402[I]/E22610[I];
  if E19200[I] ne 1 and E22610[I]>0 and E34428[I]>0 and HRWG[I]>=0 then
    HRWG[I]=HRWG[I]*E34428[I]/E22610[I];
  if E19200[I] ne 1 and E22610[I]=0 then HRWG[I]=-3;
  if E19200[I] ne 1 and -4<E22610[I]<0 then HRWG[I]=E22610[I];
end;

if E226041[I]=1 and E226042[I]=1 then do; /* neither rate nor hours is correct*/
  if E22609[I]=1 and E22626[I]>=0 then HRWG[I]=E22626[I];
  if E22609[I]=2 and E22610[I]>0 and E34402B[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]*E34402B[I]/E22610[I];
  if E22609[I]=2 and E22610[I]>0 and E34430[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]*E34430[I]/E22610[I];
  if E22609[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22610[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]/E22610[I];
  if E22609[I]=4 and E22610[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(2*E22610[I]);
  if E22609[I]=5 and E22610[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(4.3*E22610[I]);
  if E22609[I]=6 and E22610[I]>0 and E35600[I]>0 and E22626[I]>=0 then
    HRWG[I]=E22626[I]/(E35600[I]*E22610[I]);
  if E22609[I]=8 and E22610[I]>0 and E22626[I]>=0 then HRWG[I]=E22626[I]/(2.15*E22610[I]);
  if E22609[I] in (9,14) then HRWG[I]=0;
  if E22609[I]=2 and E34430[I] le 0 then HRWG[I]=-3;
  if E22609[I]=2 and E34402B[I] le 0 then HRWG[I]=-3;
  if E22609[I]=2 and -4<E34430[I]<0 then HRWG[I]=E34430[I];
  if E22609[I]=2 and -4<E34402B[I]<0 then HRWG[I]=E34402B[I];
  if E22609[I]=6 and E35600[I] le 0 then HRWG[I]=-3;
  if E22609[I]=6 and -4<E35600[I]<0 then HRWG[I]=E35600[I];
  if E22609[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-1,-2) and -4<E22626[I]<0 then
    HRWG[I]=E22626[I];
  if E22609[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22610[I]=0 then HRWG[I]=-3;
  if E22609[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and -4<E22610[I]<0 then
    HRWG[I]=E22610[I];
end;
end;

do I=1 to 9;
  if E102251[I]=1 and E102252[I]=0 then do; /* rate incorrect but hours correct*/
    /* no overtime, use E98402 for hours */
    if E100230[I]=1 and E100248[I]>=0 then HRWG[I]=E100248[I];
    if E100230[I]=2 and E98402[I]>0 and E98402D[I]>0 and E100248[I]>=0 then
      HRWG[I]=E100248[I]*E98402D[I]/E98402[I];
    if E100230[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E98402[I]>0 and E100248[I]>=0 then
      HRWG[I]=E100248[I]/E98402[I];
    if E100230[I]=4 and E98402[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(2*E98402[I]);
    if E100230[I]=5 and E98402[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(4.3*E98402[I]);
    if E100230[I]=6 and E98402[I]>0 and E99500[I]>0 and E100248[I]>=0 then
      HRWG[I]=E100248[I]/(E99500[I]*E98402[I]);
    if E100230[I]=8 and E98402[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(2.15*E98402[I]);
    if E100230[I] in (9,14) then HRWG[I]=0;
    if E100230[I]=2 and E98402D[I] le 0 then HRWG[I]=-3;
  end;
end;
```

Appendix 2: Employment Variable Creation

```
if E100230[I]=2 and -4<E98402D[I]<0 then HRWG[I]=E98402D[I];
if E100230[I]=6 and E99500[I] le 0 then HRWG[I]=-3;
if E100230[I]=6 and -4<E99500[I]<0 then HRWG[I]=E99500[I];
if E100230[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-1,-2) and -4<E100248[I]<0 then
  HRWG[I]=E100248[I];
if E100230[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E98402[I]=0 then
  HRWG[I]=-3;
if E100230[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and -4<E98402[I]<0 then
  HRWG[I]=E98402[I];

/* overtime, use E98429 for hours*/
if E100230[I]=1 and E100248[I]>=0 then HRWG[I]=E100248[I];
if E100230[I]=2 and E98429[I]>0 and E98429E[I]>0 and E100248[I]>=0 then
  HRWG[I]=E100248[I]*E98429E[I]/E98429[I];
if E100230[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E98429[I]>0 and E100248[I]>=0 then
  HRWG[I]=E100248[I]/E98429[I];
if E100230[I]=4 and E98429[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(2*E98429[I]);
if E100230[I]=5 and E98429[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(4.3*E98429[I]);
if E100230[I]=6 and E98429[I]>0 and E99500[I]>0 and E100248[I]>=0 then
  HRWG[I]=E100248[I]/(E99500[I]*E98429[I]);
if E100230[I]=8 and E98429[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(2.15*E98429[I]);
if E100230[I] in (9,14) then HRWG[I]=0;
if E100230[I]=2 and E98429E[I] le 0 then HRWG[I]=-3;
if E100230[I]=2 and -4<E98429E[I]<0 then HRWG[I]=E98429E[I];
if E100230[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-1,-2) and -4<E100248[I]<0 then
  HRWG[I]=E100248[I];
if E100230[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E98429[I]=0 then
  HRWG[I]=-3;
if E100230[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and -4<E98429[I]<0 then
  HRWG[I]=E98429[I];
end;

if E102251[I]=0 and E102252[I]=1 then do; /*rate correct but hours incorrect*/
  if E83100[I] ne 1 and E100231[I]>0 and E98402[I]>0 and HRWG[I]>=0 then
    HRWG[I]=HRWG[I]*E98402[I]/E100231[I];
  if E83100[I] ne 1 and E100231[I]>0 and E98429[I]>0 and HRWG[I]>=0 then
    HRWG[I]=HRWG[I]*E98429[I]/E100231[I];
  if E83100[I] ne 1 and E100231[I]=0 then HRWG[I]=-3;
  if E83100[I] ne 1 and -4<E100231[I]<0 then HRWG[I]=E100231[I];
end;

if E102251[I]=1 and E102252[I]=1 then do; /* neither rate nor hours is correct*/
  if E100230[I]=1 and E100248[I]>=0 then HRWG[I]=E100248[I];
  if E100230[I]=2 and E100231[I]>0 and E98402D[I]>0 and E100248[I]>=0 then
    HRWG[I]=E100248[I]*E98402D[I]/E100231[I];
  if E100230[I]=2 and E100231[I]>0 and E98429E[I]>0 and E100248[I]>=0 then
    HRWG[I]=E100248[I]*E98429E[I]/E100231[I];
  if E100230[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100231[I]>0 and E100248[I]>=0 then
    HRWG[I]=E100248[I]/E100231[I];
  if E100230[I]=4 and E100231[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(2*E100231[I]);
  if E100230[I]=5 and E100231[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(4.3*E100231[I]);
  if E100230[I]=6 and E100231[I]>0 and E99500[I]>0 and E100248[I]>=0 then
    HRWG[I]=E100248[I]/(E99500[I]*E100231[I]);
  if E100230[I]=8 and E100231[I]>0 and E100248[I]>=0 then HRWG[I]=E100248[I]/(2.15*E100231[I]);
  if E100230[I] in (9,14) then HRWG[I]=0;
  if E100230[I]=2 and E98429E[I] le 0 then HRWG[I]=-3;
  if E100230[I]=2 and E98402D[I] le 0 then HRWG[I]=-3;
```

Appendix 2: Employment Variable Creation

```

if E100230[I]=2 and -4<E98429E[I]<0 then HRWG[I]=E98429E[I];
if E100230[I]=2 and -4<E98402D[I]<0 then HRWG[I]=E98402D[I];
if E100230[I]=6 and E99500[I] le 0 then HRWG[I]=-3;
if E100230[I]=6 and -4<E99500[I]<0 then HRWG[I]=E99500[I];
if E100230[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-1,-2) and -4<E100248[I]<0 then
    HRWG[I]=E100248[I];
if E100230[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100231[I]=0 then
    HRWG[I]=-3;
if E100230[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and -4<E100231[I]<0 then
    HRWG[I]=E100231[I];
end;
end;

```

******Part 4: Add the start compensation ****/**

```

OT1=-4;      OT2=-4;      OT3=-4;      OT4=-4;      OT5=-4;
OT6=-4;      OT7=-4;      OT8=-4;      OT9=-4;

OTF1=0;      OTF2=0;      OTF3=0;      OTF4=0;      OTF5=0;
OTF6=0;      OTF7=0;      OTF8=0;      OTF9=0;

```

```

array OT OT1-OT9;
array OTF OTF1-OTF9;

```

do I=1 to 9; /* if report HOURLY rate of pay */

```

if E24501[I]>0 and E24502[I]=1 and E24514[I]>=0 then OT[I]=E24514[I];
if E24501[I]>0 and E24502[I]=2 and E24514B[I]>0 and E24514[I]>=0 then
    OT[I]=E24514[I]*E24514B[I]/E24501[I];
if E24501[I]>0 and E24502[I] in (3,7,12,13,15,16,17,21,22,23,24,25,26,28,-2) and E24514[I]>=0 then
    OT[I]=E24514[I]/E24501[I];
if E24501[I]>0 and E24502[I]=4 and E24514[I]>=0 then OT[I]=E24514[I]/(2*E24501[I]);
if E24501[I]>0 and E24502[I]=5 and E24514[I]>=0 then OT[I]=E24514[I]/(4.3*E24501[I]);
if E24501[I]>0 and E24502[I]=6 and E24514[I]>=0 then OT[I]=-3; /* Since the weeks per year is unknown.*/
if E24501[I]>0 and E24502[I]=8 and E24514[I]>=0 then OT[I]=E24514[I]/(2.15*E24501[I]);
if E24501[I]>0 and E24502[I] in (0,7,12,13,14,15,16,17,21,22,23,24,25,26,28,-2) then OTF[I]=OTF[I]+1;
if E24501[I]>0 and E24502[I]=10 and HRWG[I]>=0 then OT[I]=HRWG[I]*1.5;
if E24501[I]>0 and E24502[I]=11 and HRWG[I]>=0 then OT[I]=HRWG[I]*2;
if E24502[I]=14 then OT[I]=0;
/*MISSING VALUE*/;
if -4<E24501[I]<0 then OT[I]=E24501[I];
if -4<E24514[I]<0 then OT[I]=E24514[I];
if E24514[I]>=0 and E24501[I]=0 then OT[I]=-3;
if E24502[I]=2 and E24514B[I]=0 then OT[I]=-3;
if E24502[I]=2 and -4<E24514B[I]<0 then OT[I]=E24514B[I];

```

end;

do I=1 to 9; /*if report payment in OTHER units*/

```

if E34403[I]>0 and E34404[I]=1 and E34413C[I]>=0 then OT[I]=E34413C[I];
if E34403[I]>0 and E34404[I]=2 and E34413E[I]>0 and E34413C[I]>=0 then
    OT[I]=E34413C[I]*E34413E[I]/E34403[I];
if E34403[I]>0 and E34404[I] in (3,7,12,13,15,16,17,21,22,23,24,25,26,28,-2) and E34413C[I]>=0 then
    OT[I]=E34413C[I]/E34403[I];
if E34403[I]>0 and E34404[I]=4 and E34413C[I]>=0 then OT[I]=E34413C[I]/(2*E34403[I]);
if E34403[I]>0 and E34404[I]=5 and E34413C[I]>=0 then OT[I]=E34413C[I]/(4.3*E34403[I]);

```


Appendix 2: Employment Variable Creation

```
if E34403[I]>0 and E34404[I]=6 and E34413C[I]>=0 then OT[I]=-3; /* Since there is no weeks per year. */
if E34403[I]>0 and E34404[I]=8 and E34413C[I]>=0 then OT[I]=E34413C[I]/(2.15*E34403[I]);
if E34403[I]>0 and E34404[I] in (0,7,12,13,14,15,16,17,21,22,23,24,25,26,28,-2) then OTF[I]=OTF[I]+1;
if E34403[I]>0 and E34404[I]=10 and HRWG[I]>=0 then OT[I]=HRWG[I]*1.5;
if E34403[I]>0 and E34404[I]=11 and HRWG[I]>=0 then OT[I]=HRWG[I]*2;
if E34404[I]=14 then OT[I]=0;
/*missing value*/;
if -4<E34403[I]<0 then OT[I]=E34403[I];
if -4<E34413C[I]<0 then OT[I]=E34413C[I];
if E34413C[I]>=0 and E34403[I]=0 then OT[I]=-3;
if E34404[I]=2 and E34413E[I]=0 then OT[I]=-3;
if E34404[I]=2 and -4<E34413E[I]<0 then OT[I]=E34413E[I];
```

end;

do I=1 to 9; /*if report overtime pay later*/

```
if E88501[I]>0 and E88502[I]=1 and E88512[I]>=0 then OT[I]=E88512[I];
if E88501[I]>0 and E88502[I]=2 and E88512B[I]>0 and E88512[I]>=0 then
    OT[I]=E88512[I]*E88512B[I]/E88501[I];
if E88501[I]>0 and E88502[I] in (3,7,12,13,15,16,17,21,22,23,24,25,26,28,-2) and E88512[I]>=0 then
    OT[I]=E88512[I]/E88501[I];
if E88501[I]>0 and E88502[I]=4 and E88512[I]>=0 then OT[I]=E88512[I]/(2*E88501[I]);
if E88501[I]>0 and E88502[I]=5 and E88512[I]>=0 then OT[I]=E88512[I]/(4.3*E88501[I]);
if E88501[I]>0 and E88502[I]=6 and E88512[I]>=0 then OT[I]=-3; /* Since there is no weeks per year. */
if E88501[I]>0 and E88502[I]=8 and E88512[I]>=0 then OT[I]=E88512[I]/(2.15*E88501[I]);
if E88501[I]>0 and E88502[I] in (0,7,12,13,14,15,16,17,21,22,23,24,25,26,28,-2) then OTF[I]=OTF[I]+1;
if E88501[I]>0 and E88502[I]=10 and HRWG[I]>=0 then OT[I]=HRWG[I]*1.5;
if E88501[I]>0 and E88502[I]=11 and HRWG[I]>=0 then OT[I]=HRWG[I]*2;
if E88502[I]=14 then OT[I]=0;
/*missing value*/;
if -4<E88501[I]<0 then OT[I]=E88501[I];
if -4<E88512[I]<0 then OT[I]=E88512[I];
if E88512[I]>=0 and E88501[I]=0 then OT[I]=-3;
```

end;

do I=1 to 9;

```
if E98403[I]>0 and E98404[I]=1 and E98414[I]>=0 then OT[I]=E98414[I];
if E98403[I]>0 and E98404[I]=2 and E98414B[I]>0 and E98414[I]>=0 then
    OT[I]=E98414[I]*E98414B[I]/E98403[I];
if E98403[I]>0 and E98404[I] in (3,7,12,13,15,16,17,21,22,23,24,25,26,28,-2) and E98414[I]>=0 then
    OT[I]=E98414[I]/E98403[I];
if E98403[I]>0 and E98404[I]=4 and E98414[I]>=0 then OT[I]=E98414[I]/(2*E98403[I]);
if E98403[I]>0 and E98404[I]=5 and E98414[I]>=0 then OT[I]=E98414[I]/(4.3*E98403[I]);
if E98403[I]>0 and E98404[I]=6 and E98414[I]>=0 then OT[I]=-3; /* Since there is no weeks per year. */
if E98403[I]>0 and E98404[I]=8 and E98414[I]>=0 then OT[I]=E98414[I]/(2.15*E98403[I]);
if E98403[I]>0 and E98404[I] in (0,7,12,13,14,15,16,17,21,22,23,24,25,26,28,-2) then OTF[I]=OTF[I]+1;
if E98403[I]>0 and E98404[I]=10 and HRWG[I]>=0 then OT[I]=HRWG[I]*1.5;
if E98403[I]>0 and E98404[I]=11 and HRWG[I]>=0 then OT[I]=HRWG[I]*2;
if E98404[I]=14 then OT[I]=0;
/*MISSING VALUE*/;
if -4<E98403[I]<0 then OT[I]=E98403[I];
if -4<E98414[I]<0 then OT[I]=E98414[I];
if E98414[I]>=0 and E98403[I]=0 then OT[I]=-3;
```

Appendix 2: Employment Variable Creation

end;

do I=1 to 9; /* report the corrected overtime payment if the correction is made */

if E226044[I]=1 and E226043[I]=0 then do; /*rate incorrect but hours correct*/
if E19200[I]=1 then do; /* report hourly payrate*/
if E22612[I]=1 and E22627[I]>=0 then OT[I]=E22627[I];
if E22612[I]=2 and E22627[I]>=0 and E24514B[I]>0 and E24501[I]>0 then
OT[I]=E22627[I]*E24514B[I]/E24501[I];
if E22612[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22627[I]>=0 and E24501[I]>0 then
OT[I]=E22627[I]/E24501[I];
if E22612[I]=4 and E22627[I]>=0 and E24501[I]>0 then OT[I]=E22627[I]/(2*E24501[I]);
if E22612[I]=5 and E22627[I]>=0 and E24501[I]>0 then OT[I]=E22627[I]/(4.3*E24501[I]);
if E22612[I]=6 and E22627[I]>=0 and E24501[I]>0 then OT[I]=-3; /* Since there is no weeks per year. */
if E22612[I]=8 and E22627[I]>=0 and E24501[I]>0 then OT[I]=E22627[I]/(2.15*E24501[I]);
if E22612[I] in (9,14) then OT[I]=0;
if E24501[I]=0 then OT[I]=-3;
if -4<E24501[I]<0 then OT[I]=E24501[I];
if -4<E22627[I]<0 then OT[I]=E22627[I];
if E22612[I]=2 and E24514B[I] le 0 then OT[I]=-3;
if E22612[I]=2 and -4<E24514B[I]<0 then OT[I]=E24514B[I];

end;

if E19200[I] ne 1 then do; /*report non-hourly payrate*/

if E22612[I]=1 and E22627[I]>=0 then OT[I]=E22627[I];
if E22612[I]=2 and E22627[I]>=0 and E34413E[I]>0 and E34403[I]>0 then
OT[I]=E22627[I]*E34413E[I]/E34403[I];
if E22612[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22627[I]>=0 and E34403[I]>0 then
OT[I]=E22627[I]/E34403[I];
if E22612[I]=4 and E22627[I]>=0 and E34403[I]>0 then OT[I]=E22627[I]/(2*E34403[I]);
if E22612[I]=5 and E22627[I]>=0 and E34403[I]>0 then OT[I]=E22627[I]/(4.3*E34403[I]);
if E22612[I]=6 and E22627[I]>=0 and E34403[I]>0 then OT[I]=-3; /* Since there is no weeks per year. */
if E22612[I]=8 and E22627[I]>=0 and E34403[I]>0 then OT[I]=E22627[I]/(2.15*E34403[I]);
if E22612[I] in (9,14) then OT[I]=0;
if E34403[I]=0 then OT[I]=-3;
if -4<E34403[I]<0 then OT[I]=E34403[I];
if -4<E22627[I]<0 then OT[I]=E22627[I];
if E22612[I]=2 and E34413E[I] le 0 then OT[I]=-3;
if E22612[I]=2 and -4<E34413E[I]<0 then OT[I]=E34413E[I];

end;

end;

if E226044[I]=0 and E226043[I]=1 then do; /*rate correct but hours incorrect*/
if E24502[I] ne 1 and OT[I]>=0 and E24501[I]>0 and E22611[I]>0 then OT[I]=OT[I]*E24501[I]/E22611[I];
if E34404[I] ne 1 and OT[I]>=0 and E34403[I]>0 and E22611[I]>0 then OT[I]=OT[I]*E34403[I]/E22611[I];
if E24502[I] ne 1 and E34404[I] ne 1 and E22611[I]=0 then OT[I]=-3;
if E24502[I] ne 1 and E34404[I] ne 1 and -4<E22611[I]<0 then OT[I]=E22611[I];

end;

if E226044[I]=1 and E226043[I]=1 then do; /* neither rate nor the hours is correct*/
if E22612[I]=1 and E22627[I]>=0 then OT[I]=E22627[I];
if E22612[I]=2 and E22627[I]>=0 and E24514B[I]>0 and E22611[I]>0 then
OT[I]=E22627[I]*E24514B[I]/E22611[I];
if E22612[I]=2 and E22627[I]>=0 and E34413E[I]>0 and E22611[I]>0 then
OT[I]=E22627[I]*E34413E[I]/E22611[I];
if E22612[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22627[I]>=0 and E22611[I]>0 then
OT[I]=E22627[I]/E22611[I];
if E22612[I]=4 and E22627[I]>=0 and E22611[I]>0 then OT[I]=E22627[I]/(2*E22611[I]);

Appendix 2: Employment Variable Creation

```
if E22612[I]=5 and E22627[I]>=0 and E22611[I]>0 then OT[I]=E22627[I]/(4.3*E22611[I]);
if E22612[I]=6 and E22627[I]>=0 and E22611[I]>0 then OT[I]=-3; /* Since there is no weeks per year. */
if E22612[I]=8 and E22627[I]>=0 and E22611[I]>0 then OT[I]=E22627[I]/(2.15*E22611[I]);
if E22612[I] in (9,14) then OT[I]=0;
if E22612[I]=2 and E24514B[I] le 0 then OT[I]=-3;
if E22612[I]=2 and E34413E[I] le 0 then OT[I]=-3;
if E22612[I]=2 and -4<E24514B[I]<0 then OT[I]=E24514B[I];
if E22612[I]=2 and -4<E34413E[I]<0 then OT[I]=E34413E[I];
if E22612[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22611[I]=0 then OT[I]=-3;
if E22612[I] in (2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and -4<E22611[I]<0 then
    OT[I]=E22611[I];
if E22612[I] in (1,2,3,4,5,6,7,8,0,12,13,15,16,17,99,21,22,23,24,25,26,28,-1,-2) and -4<E22627[I]<0 then
    OT[I]=E22627[I];
```

```
end;
end;
```

****** Part 5: Amount paid in tips, commissions, bonuses, incentive pay or other ******

```
array OTHPAY1 OTHPAY11 OTHPAY21 OTHPAY31 OTHPAY41 OTHPAY51 OTHPAY61 OTHPAY71
    OTHPAY81 OTHPAY91;
array OTHPAY2 OTHPAY12 OTHPAY22 OTHPAY32 OTHPAY42 OTHPAY52 OTHPAY62 OTHPAY72
    OTHPAY82 OTHPAY92;
array OTHPAY3 OTHPAY13 OTHPAY23 OTHPAY33 OTHPAY43 OTHPAY53 OTHPAY63 OTHPAY73
    OTHPAY83 OTHPAY93;
array OTHPAY4 OTHPAY14 OTHPAY24 OTHPAY34 OTHPAY44 OTHPAY54 OTHPAY64 OTHPAY74
    OTHPAY84 OTHPAY94;
array OTHPAY5 OTHPAY15 OTHPAY25 OTHPAY35 OTHPAY45 OTHPAY55 OTHPAY65 OTHPAY75
    OTHPAY85 OTHPAY95;
array OTHPF1 OTHPF11 OTHPF21 OTHPF31 OTHPF41 OTHPF51 OTHPF61 OTHPF71 OTHPF81
    OTHPF91;
array OTHPF2 OTHPF12 OTHPF22 OTHPF32 OTHPF42 OTHPF52 OTHPF62 OTHPF72 OTHPF82
    OTHPF92;
array OTHPF3 OTHPF13 OTHPF23 OTHPF33 OTHPF43 OTHPF53 OTHPF63 OTHPF73 OTHPF83
    OTHPF93;
array OTHPF4 OTHPF14 OTHPF24 OTHPF34 OTHPF44 OTHPF54 OTHPF64 OTHPF74 OTHPF84
    OTHPF94;
array OTHPF5 OTHPF15 OTHPF25 OTHPF35 OTHPF45 OTHPF55 OTHPF65 OTHPF75 OTHPF85
    OTHPF95;
```

```
do I=1 to 9;
    OTHPAY1[I]=-4;          OTHPAY2[I]=-4;          OTHPAY3[I]=-4;
    OTHPAY4[I]=-4;          OTHPAY5[I]=-4;
    OTHPF1[I]=0;           OTHPF2[I]=0;           OTHPF3[I]=0;
    OTHPF4[I]=0;           OTHPF5[I]=0;
end;
```

/ do it for non-overtime payment*/*

******* FOR TIPS *******

do I=1 to 9; */*with overtime*/*

```
if E34428[I]>0 and E216001[I]=1 and E225001[I]>=0 then OTHPAY1[I]=E225001[I];
if E34428[I]>0 and E216001[I]=2 and E34430[I]>0 and E225001[I]>=0 then
    OTHPAY1[I]=(E225001[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E216001[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225001[I]>=0 then
    OTHPAY1[I]=E225001[I]/E34428[I];
if E34428[I]>0 and E216001[I]=4 and E225001[I]>=0 then OTHPAY1[I]=E225001[I]/(2*E34428[I]);
if E34428[I]>0 and E216001[I]=5 and E225001[I]>=0 then OTHPAY1[I]=E225001[I]/(4.3*E34428[I]);
```

Appendix 2: Employment Variable Creation

```
if E34428[I]>0 and E216001[I]=6 and E225001[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E225001[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E216001[I]=8 and E225001[I]>=0 then OTHPAY1[I]=E225001[I]/(2.15*E34428[I]);
/*missing value*/;
if E225001[I]>=0 and -4<E34428[I]<0 then OTHPAY1[I]=E34428[I];
if -4<E225001[I]<0 then OTHPAY1[I]=E225001[I];
if E225001[I]>=0 and E34428[I]=0 then OTHPAY1[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E34402[I]>0 and E216001[I]=1 and E225001[I]>=0 then OTHPAY1[I]=E225001[I];
if E34402[I]>0 and E216001[I]=2 and E34402B[I]>0 and E225001[I]>=0 then
  OTHPAY1[I]=(E225001[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E216001[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225001[I]>=0 then
  OTHPAY1[I]=E225001[I]/E34402[I];
if E34402[I]>0 and E216001[I]=4 and E225001[I]>=0 then OTHPAY1[I]=E225001[I]/(2*E34402[I]);
if E34402[I]>0 and E216001[I]=5 and E225001[I]>=0 then OTHPAY1[I]=E225001[I]/(4.3*E34402[I]);
if E34402[I]>0 and E216001[I]=6 and E225001[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E225001[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E216001[I]=8 and E225001[I]>=0 then OTHPAY1[I]=E225001[I]/(2.15*E34402[I]);
if E216001[I] in (9,14) then OTHPAY1[I]=0;
if E216001[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY1[I]=OTHPAY1[I]+1;
/*missing value*/;
if E225001[I]>=0 and -4<E34402[I]<0 then OTHPAY1[I]=E34402[I];
if E225001[I]>=0 and E34402[I]=0 then OTHPAY1[I]=-3;
if E216001[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY1[I]=-3;
if E216001[I]=2 and -4<E34402B[I]<0 then OTHPAY1[I]=E34402B[I];
if E216001[I]=2 and -4<E34430[I]<0 then OTHPAY1[I]=E34430[I];
if E216001[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E216001[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
end;

/* if tips is corrected in the later part */
do I=1 to 9;
if E226045[I]=1 then do;
if E22613[I]=1 and E22628[I]>=0 then OTHPAY1[I]=E22628[I]; /* with overtime*/
if E22613[I]=2 and E22628[I]>=0 and E34430[I]>0 and E34428[I]>0 then
  OTHPAY1[I]=E22628[I]*E34430[I]/E34428[I];
if E22613[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22628[I]>=0 and E34428[I]>0 then
  OTHPAY1[I]=E22628[I]/E34428[I];
if E22613[I]=4 and E22628[I]>=0 and E34428[I]>0 then OTHPAY1[I]=E22628[I]/(2*E34428[I]);
if E22613[I]=5 and E22628[I]>=0 and E34428[I]>0 then OTHPAY1[I]=E22628[I]/(4.3*E34428[I]);
if E22613[I]=6 and E22628[I]>=0 and E34428[I]>0 and E35600[I]>0 then
  OTHPAY1[I]=E22628[I]/(E35600[I]*E34428[I]);
if E22613[I]=8 and E22628[I]>=0 and E34428[I]>0 then OTHPAY1[I]=E22628[I]/(2.15*E34428[I]);
if E22613[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/
if -4<E34428[I]<0 then OTHPAY1[I]=E34428[I];
if -4<E22628[I]<0 then OTHPAY1[I]=E22628[I];
if E34428[I]=0 then OTHPAY1[I]=-3;

if E22613[I]=1 and E22628[I]>=0 then OTHPAY1[I]=E22628[I]; /* without overtime*/
if E22613[I]=2 and E22628[I]>=0 and E34402B[I]>0 and E34402[I]>0 then
  OTHPAY1[I]=E22628[I]*E34402B[I]/E34402[I];
if E22613[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22628[I]>=0 and E34402[I]>0 then
  OTHPAY1[I]=E22628[I]/E34402[I];
if E22613[I]=4 and E22628[I]>=0 and E34402[I]>0 then OTHPAY1[I]=E22628[I]/(2*E34402[I]);
```

Appendix 2: Employment Variable Creation

```
if E22613[I]=5 and E22628[I]>=0 and E34402[I]>0 then OTHPAY1[I]=E22628[I]/(4.3*E34402[I]);
if E22613[I]=6 and E22628[I]>=0 and E34402[I]>0 and E35600[I]>0 then
    OTHPAY1[I]=E22628[I]/(E35600[I]*E34402[I]);
if E22613[I]=8 and E22628[I]>=0 and E34402[I]>0 then OTHPAY1[I]=E22628[I]/(2.15*E34402[I]);
if E22613[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/
if -4<E34402[I]<0 then OTHPAY1[I]=E34402[I];
if E34402[I]=0 then OTHPAY1[I]=-3;
if E22613[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY1[I]=-3;
if E22613[I]=2 and -4<E34402B[I]<0 then OTHPAY1[I]=E34402B[I];
if E22613[I]=2 and -4<E34430[I]<0 then OTHPAY1[I]=E34430[I];
if E22613[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E22613[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
end;
end;

/***** FOR COMMISSIONS *****/
do I=1 to 9; /*with overtime*/
if E34428[I]>0 and E216002[I]=1 and E225002[I]>=0 then OTHPAY2[I]=E225002[I];
if E34428[I]>0 and E216002[I]=2 and E34430[I]>0 and E225002[I]>=0 then
    OTHPAY2[I]=(E225002[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E216002[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225002[I]>=0 then
    OTHPAY2[I]=E225002[I]/E34428[I];
if E34428[I]>0 and E216002[I]=4 and E225002[I]>=0 then OTHPAY2[I]=E225002[I]/(2*E34428[I]);
if E34428[I]>0 and E216002[I]=5 and E225002[I]>=0 then OTHPAY2[I]=E225002[I]/(4.3*E34428[I]);
if E34428[I]>0 and E216002[I]=6 and E225002[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E225002[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E216002[I]=8 and E225002[I]>=0 then OTHPAY2[I]=E225002[I]/(2.15*E34428[I]);
*missing value*/;
if E225002[I]>=0 and -4<E34428[I]<0 then OTHPAY2[I]=E34428[I];
if -4<E225002[I]<0 then OTHPAY2[I]=E225002[I];
if E225002[I]>=0 and E34428[I]=0 then OTHPAY2[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E34402[I]>0 and E216002[I]=1 and E225002[I]>=0 then OTHPAY2[I]=E225002[I];
if E34402[I]>0 and E216002[I]=2 and E34402B[I]>0 and E225002[I]>=0 then
    OTHPAY2[I]=(E225002[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E216002[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225002[I]>=0 then
    OTHPAY2[I]=E225002[I]/E34402[I];
if E34402[I]>0 and E216002[I]=4 and E225002[I]>=0 then OTHPAY2[I]=E225002[I]/(2*E34402[I]);
if E34402[I]>0 and E216002[I]=5 and E225002[I]>=0 then OTHPAY2[I]=E225002[I]/(4.3*E34402[I]);
if E34402[I]>0 and E216002[I]=6 and E225002[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E225002[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E216002[I]=8 and E225002[I]>=0 then OTHPAY2[I]=E225002[I]/(2.15*E34402[I]);
if E216002[I] in (9,14) then OTHPAY2[I]=0;
if E216002[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY2[I]=OTHPAY2[I]+1;
/*missing value*/
if E225002[I]>=0 and -4<E34402[I]<0 then OTHPAY2[I]=E34402[I];
if E225002[I]>=0 and E34402[I]=0 then OTHPAY2[I]=-3;
if E216002[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY2[I]=-3;
if E216002[I]=2 and -4<E34402B[I]<0 then OTHPAY2[I]=E34402B[I];
if E216002[I]=2 and -4<E34430[I]<0 then OTHPAY2[I]=E34430[I];
if E216002[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E216002[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];
end;
```

Appendix 2: Employment Variable Creation

```
/* if commissions is corrected in the later part */
do I=1 to 9;
if E226046[I]=1 then do;

if E22614[I]=1 and E22629[I]>=0 then OTHPAY2[I]=E22629[I]; /* with overtime*/
if E22614[I]=2 and E22629[I]>=0 and E34430[I]>0 and E34428[I]>0 then
    OTHPAY2[I]=E22629[I]*E34430[I]/E34428[I];
if E22614[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22629[I]>=0 and E34428[I]>0 then
    OTHPAY2[I]=E22629[I]/E34428[I];
if E22614[I]=4 and E22629[I]>=0 and E34428[I]>0 then OTHPAY2[I]=E22629[I]/(2*E34428[I]);
if E22614[I]=5 and E22629[I]>=0 and E34428[I]>0 then OTHPAY2[I]=E22629[I]/(4.3*E34428[I]);
if E22614[I]=6 and E22629[I]>=0 and E34428[I]>0 and E35600[I]>0 then
    OTHPAY2[I]=E22629[I]/(E35600[I]*E34428[I]);
if E22614[I]=8 and E22629[I]>=0 and E34428[I]>0 then OTHPAY2[I]=E22629[I]/(2.15*E34428[I]);
if E22614[I] in (9,14) then OTHPAY2[I]=0;
/*missing value*/
if -4<E34428[I]<0 then OTHPAY2[I]=E34428[I];
if -4<E22629[I]<0 then OTHPAY2[I]=E22629[I];
if E34428[I]=0 then OTHPAY2[I]=-3;

if E22614[I]=1 and E22629[I]>=0 then OTHPAY2[I]=E22629[I]; /* without overtime*/
if E22614[I]=2 and E22629[I]>=0 and E34402B[I]>0 and E34402[I]>0 then
    OTHPAY2[I]=E22629[I]*E34402B[I]/E34402[I];
if E22614[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22629[I]>=0 and E34402[I]>0 then
    OTHPAY2[I]=E22629[I]/E34402[I];
if E22614[I]=4 and E22629[I]>=0 and E34402[I]>0 then OTHPAY2[I]=E22629[I]/(2*E34402[I]);
if E22614[I]=5 and E22629[I]>=0 and E34402[I]>0 then OTHPAY2[I]=E22629[I]/(4.3*E34402[I]);
if E22614[I]=6 and E22629[I]>=0 and E34402[I]>0 and E35600[I]>0 then
    OTHPAY2[I]=E22629[I]/(E35600[I]*E34402[I]);
if E22614[I]=8 and E22629[I]>=0 and E34402[I]>0 then OTHPAY2[I]=E22629[I]/(2.15*E34402[I]);
if E22614[I] in (9,14) then OTHPAY2[I]=0;
/*missing value*/
if -4<E34402[I]<0 then OTHPAY2[I]=E34402[I];
if E34402[I]=0 then OTHPAY2[I]=-3;
if E22614[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY2[I]=-3;
if E22614[I]=2 and -4<E34402B[I]<0 then OTHPAY2[I]=E34402B[I];
if E22614[I]=2 and -4<E34430[I]<0 then OTHPAY2[I]=E34430[I];
if E22614[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E22614[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];

end;
end;

/***** FOR BONUSES *****/
do I=1 to 9; /*with overtime*/
if E34428[I]>0 and E216003[I]=1 in (1,21) and E225003[I]>=0 then OTHPAY3[I]=E225003[I];
if E34428[I]>0 and E216003[I]=2 in (2,22) and E34430[I]>0 and E225003[I]>=0 then
    OTHPAY3[I]=(E225003[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E216003[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225003[I]>=0 then
    OTHPAY3[I]=E225003[I]/E34428[I];
if E34428[I]>0 and E216003[I]=4 and E225003[I]>=0 then OTHPAY3[I]=E225003[I]/(2*E34428[I]);
if E34428[I]>0 and E216003[I]=5 and E225003[I]>=0 then OTHPAY3[I]=E225003[I]/(4.3*E34428[I]);
if E34428[I]>0 and E216003[I]=6 and E225003[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E225003[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E216003[I]=8 and E225003[I]>=0 then OTHPAY3[I]=E225003[I]/(2.15*E34428[I]);
/*MISSING VALUE*/;
if E225003[I]>=0 and -4<E34428[I]<0 then OTHPAY3[I]=E34428[I];
```

Appendix 2: Employment Variable Creation

```

if -4<E225003[I]<0 then OTHPAY3[I]=E225003[I];
if E225003[I]>=0 and E34428[I]=0 then OTHPAY3[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E34402[I]>0 and E216003[I]=1 and E225003[I]>=0 then OTHPAY3[I]=E225003[I];
if E34402[I]>0 and E216003[I]=2 and E34402B[I]>0 and E225003[I]>=0 then
    OTHPAY3[I]=(E225003[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E216003[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225003[I]>=0 then
    OTHPAY3[I]=E225003[I]/E34402[I];
if E34402[I]>0 and E216003[I]=4 and E225003[I]>=0 then OTHPAY3[I]=E225003[I]/(2*E34402[I]);
if E34402[I]>0 and E216003[I]=5 and E225003[I]>=0 then OTHPAY3[I]=E225003[I]/(4.3*E34402[I]);
if E34402[I]>0 and E216003[I]=6 and E225003[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E225003[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E216003[I]=8 and E225003[I]>=0 then OTHPAY3[I]=E225003[I]/(2.15*E34402[I]);
if E216003[I] in (9,14) then OTHPAY3[I]=0;
if E216003[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPF3[I]=OTHPF3[I]+1;
*MISSING VALUE*;
if E225003[I]>=0 and -4<E34402[I]<0 then OTHPAY3[I]=E34402[I];
if E225003[I]>=0 and E34402[I]=0 then OTHPAY3[I]=-3;
if E216003[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY3[I]=-3;
if E216003[I]=2 and -4<E34402B[I]<0 then OTHPAY3[I]=E34402B[I];
if E216003[I]=2 and -4<E34430[I]<0 then OTHPAY3[I]=E34430[I];
if E216003[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E216003[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;

/* if bonus is corrected in the later part */
do I=1 to 9;
if E226047[I]=1 then do;
if E22615[I]=1 and E22630[I]>=0 then OTHPAY3[I]=E22630[I]; /* with overtime*/
if E22615[I]=2 and E22630[I]>=0 and E34430[I]>0 and E34428[I]>0 then
    OTHPAY3[I]=E22630[I]*E34430[I]/E34428[I];
if E22615[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22630[I]>=0 and E34428[I]>0 then
    OTHPAY3[I]=E22630[I]/E34428[I];
if E22615[I]=4 and E22630[I]>=0 and E34428[I]>0 then OTHPAY3[I]=E22630[I]/(2*E34428[I]);
if E22615[I]=5 and E22630[I]>=0 and E34428[I]>0 then OTHPAY3[I]=E22630[I]/(4.3*E34428[I]);
if E22615[I]=6 and E22630[I]>=0 and E34428[I]>0 and E35600[I]>0 then
    OTHPAY3[I]=E22630[I]/(E35600[I]*E34428[I]);
if E22615[I]=8 and E22630[I]>=0 and E34428[I]>0 then OTHPAY3[I]=E22630[I]/(2.15*E34428[I]);
if E22615[I] in (9,14) then OTHPAY3[I]=0;
/*missing value*/
if -4<E34428[I]<0 then OTHPAY3[I]=E34428[I];
if -4<E22630[I]<0 then OTHPAY3[I]=E22630[I];
if E34428[I]=0 then OTHPAY3[I]=-3;

if E22615[I]=1 and E22630[I]>=0 then OTHPAY3[I]=E22630[I]; /* without overtime*/
if E22615[I]=2 and E22630[I]>=0 and E34402B[I]>0 and E34402[I]>0 then
    OTHPAY3[I]=E22630[I]*E34402B[I]/E34402[I];
if E22615[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22630[I]>=0 and E34402[I]>0 then
    OTHPAY3[I]=E22630[I]/E34402[I];
if E22615[I]=4 and E22630[I]>=0 and E34402[I]>0 then OTHPAY3[I]=E22630[I]/(2*E34402[I]);
if E22615[I]=5 and E22630[I]>=0 and E34402[I]>0 then OTHPAY3[I]=E22630[I]/(4.3*E34402[I]);
if E22615[I]=6 and E22630[I]>=0 and E34402[I]>0 and E35600[I]>0 then
    OTHPAY3[I]=E22630[I]/(E35600[I]*E34402[I]);
if E22615[I]=8 and E22630[I]>=0 and E34402[I]>0 then OTHPAY3[I]=E22630[I]/(2.15*E34402[I]);
if E22615[I] in (9,14) then OTHPAY3[I]=0;

```

Appendix 2: Employment Variable Creation

```
/*missing value*/
if -4<E34402[I]<0 then OTHPAY3[I]=E34402[I];
if E34402[I]=0 then OTHPAY3[I]=-3;
if E22615[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY3[I]=-3;
if E22615[I]=2 and -4<E34402B[I]<0 then OTHPAY3[I]=E34402B[I];
if E22615[I]=2 and -4<E34430[I]<0 then OTHPAY3[I]=E34430[I];
if E22615[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E22615[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;
end;

/***** FOR INCENTIVE PAY *****/
do I=1 to 9; /*with overtime*/
if E34428[I]>0 and E216004[I]=1 and E225004[I]>=0 then OTHPAY4[I]=E225004[I];
if E34428[I]>0 and E216004[I]=2 and E34430[I]>0 and E225004[I]>=0 then
    OTHPAY4[I]=(E225004[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E216004[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225004[I]>=0 then
    OTHPAY4[I]=E225004[I]/E34428[I];
if E34428[I]>0 and E216004[I]=4 and E225004[I]>=0 then OTHPAY4[I]=E225004[I]/(2*E34428[I]);
if E34428[I]>0 and E216004[I]=5 and E225004[I]>=0 then OTHPAY4[I]=E225004[I]/(4.3*E34428[I]);
if E34428[I]>0 and E216004[I]=6 and E225004[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E225004[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E216004[I]=8 and E225004[I]>=0 then OTHPAY4[I]=E225004[I]/(2.15*E34428[I]);
/*missing value*/;
if E225004[I]>=0 and -4<E34428[I]<0 then OTHPAY4[I]=E34428[I];
if -4<E225004[I]<0 then OTHPAY4[I]=E225004[I];
if E225004[I]>=0 and E34428[I]=0 then OTHPAY4[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E34402[I]>0 and E216004[I]=1 and E225004[I]>=0 then OTHPAY4[I]=E225004[I];
if E34402[I]>0 and E216004[I]=2 and E34402B[I]>0 and E225004[I]>=0 then
    OTHPAY4[I]=(E225004[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E216004[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225004[I]>=0 then
    OTHPAY4[I]=E225004[I]/E34402[I];
if E34402[I]>0 and E216004[I]=4 and E225004[I]>=0 then OTHPAY4[I]=E225004[I]/(2*E34402[I]);
if E34402[I]>0 and E216004[I]=5 and E225004[I]>=0 then OTHPAY4[I]=E225004[I]/(4.3*E34402[I]);
if E34402[I]>0 and E216004[I]=6 and E225004[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E225004[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E216004[I]=8 and E225004[I]>=0 then OTHPAY4[I]=E225004[I]/(2.15*E34402[I]);
if E216004[I] in (9,14) then OTHPAY4[I]=0;
if E216004[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY4[I]=OTHPAY4[I]+1;
/*missing value*/
if E225004[I]>=0 and -4<E34402[I]<0 then OTHPAY4[I]=E34402[I];
if E225004[I]>=0 and E34402[I]=0 then OTHPAY4[I]=-3;
if E216004[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY4[I]=-3;
if E216004[I]=2 and -4<E34402B[I]<0 then OTHPAY4[I]=E34402B[I];
if E216004[I]=2 and -4<E34430[I]<0 then OTHPAY4[I]=E34430[I];
if E216004[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E216004[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
end;

/* if incentive pay is corrected in the later part */
do I=1 to 9;
if E226048[I]=1 then do;
if E22616[I]=1 and E22631[I]>=0 then OTHPAY4[I]=E22631[I]; /* with overtime*/
```


Appendix 2: Employment Variable Creation

```
if E22616[I]=2 and E22631[I]>=0 and E34430[I]>0 and E34428[I]>0 then
  OTHPAY4[I]=E22631[I]*E34430[I]/E34428[I];
if E22616[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22631[I]>=0 and E34428[I]>0 then
  OTHPAY4[I]=E22631[I]/E34428[I];
if E22616[I]=4 and E22631[I]>=0 and E34428[I]>0 then OTHPAY4[I]=E22631[I]/(2*E34428[I]);
if E22616[I]=5 and E22631[I]>=0 and E34428[I]>0 then OTHPAY4[I]=E22631[I]/(4.3*E34428[I]);
if E22616[I]=6 and E22631[I]>=0 and E34428[I]>0 and E35600[I]>0 then
  OTHPAY4[I]=E22631[I]/(E35600[I]*E34428[I]);
if E22616[I]=8 and E22631[I]>=0 and E34428[I]>0 then OTHPAY4[I]=E22631[I]/(2.15*E34428[I]);
if E22616[I] in (9,14) then OTHPAY4[I]=0;
/*missing value*/
if -4<E34428[I]<0 then OTHPAY4[I]=E34428[I];
if -4<E22631[I]<0 then OTHPAY4[I]=E22631[I];
if E34428[I]=0 then OTHPAY4[I]=-3;

if E22616[I]=1 and E22631[I]>=0 then OTHPAY4[I]=E22631[I]; /* without overtime*/
if E22616[I]=2 and E22631[I]>=0 and E34402B[I]>0 and E34402[I]>0 then
  OTHPAY4[I]=E22631[I]*E34402B[I]/E34402[I];
if E22616[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22631[I]>=0 and E34402[I]>0 then
  OTHPAY4[I]=E22631[I]/E34402[I];
if E22616[I]=4 and E22631[I]>=0 and E34402[I]>0 then OTHPAY4[I]=E22631[I]/(2*E34402[I]);
if E22616[I]=5 and E22631[I]>=0 and E34402[I]>0 then OTHPAY4[I]=E22631[I]/(4.3*E34402[I]);
if E22616[I]=6 and E22631[I]>=0 and E34402[I]>0 and E35600[I]>0 then
  OTHPAY4[I]=E22631[I]/(E35600[I]*E34402[I]);
if E22616[I]=8 and E22631[I]>=0 and E34402[I]>0 then OTHPAY4[I]=E22631[I]/(2.15*E34402[I]);
if E22616[I] in (9,14) then OTHPAY4[I]=0;
/*missing value*/
if -4<E34402[I]<0 then OTHPAY4[I]=E34402[I];
if E34402[I]=0 then OTHPAY4[I]=-3;
if E22616[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY4[I]=-3;
if E22616[I]=2 and -4<E34402B[I]<0 then OTHPAY4[I]=E34402B[I];
if E22616[I]=2 and -4<E34430[I]<0 then OTHPAY4[I]=E34430[I];
if E22616[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E22616[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
end;
end;

/***** FOR OTHERS *****/
do I=1 to 9; /*with overtime*/
  if E34428[I]>0 and E216005[I]=1 and E225005[I]>=0 then OTHPAY5[I]=E225005[I];
  if E34428[I]>0 and E216005[I]=2 and E34430[I]>0 and E225005[I]>=0 then
    OTHPAY5[I]=(E225005[I]*E34430[I])/E34428[I];
  if E34428[I]>0 and E216005[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225005[I]>=0 then
    OTHPAY5[I]=E225005[I]/E34428[I];
  if E34428[I]>0 and E216005[I]=4 and E225005[I]>=0 then OTHPAY5[I]=E225005[I]/(2*E34428[I]);
  if E34428[I]>0 and E216005[I]=5 and E225005[I]>=0 then OTHPAY5[I]=E225005[I]/(4.3*E34428[I]);
  if E34428[I]>0 and E216005[I]=6 and E225005[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E225005[I]/(E35600[I]*E34428[I]);
  if E34428[I]>0 and E216005[I]=8 and E225005[I]>=0 then OTHPAY5[I]=E225005[I]/(2.15*E34428[I]);
  /*missing value*/
  if E225005[I]>=0 and -4<E34428[I]<0 then OTHPAY5[I]=E34428[I];
  if -4<E225005[I]<0 then OTHPAY5[I]=E225005[I];
  if E225005[I]>=0 and E34428[I]=0 then OTHPAY5[I]=-3;
end;

do I=1 to 9; /*without overtime*/
```

Appendix 2: Employment Variable Creation

```

if E34402[I]>0 and E216005[I]=1 and E225005[I]>=0 then OTHPAY5[I]=E225005[I];
if E34402[I]>0 and E216005[I]=2 and E34402B[I]>0 and E225005[I]>=0 then
    OTHPAY5[I]=(E225005[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E216005[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E225005[I]>=0 then
    OTHPAY5[I]=E225005[I]/E34402[I];
if E34402[I]>0 and E216005[I]=4 and E225005[I]>=0 then OTHPAY5[I]=E225005[I]/(2*E34402[I]);
if E34402[I]>0 and E216005[I]=5 and E225005[I]>=0 then OTHPAY5[I]=E225005[I]/(4.3*E34402[I]);
if E34402[I]>0 and E216005[I]=6 and E225005[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E225005[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E216005[I]=8 and E225005[I]>=0 then OTHPAY5[I]=E225005[I]/(2.15*E34402[I]);
if E216005[I] in (9,14) then OTHPAY5[I]=0;
if E216005[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPF5[I]=OTHPF5[I]+1;
/*missing value*/
if E225005[I]>=0 and -4<E34402[I]<0 then OTHPAY5[I]=E34402[I];
if E225005[I]>=0 and E34402[I]=0 then OTHPAY5[I]=-3;
if E216005[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY5[I]=-3;
if E216005[I]=2 and -4<E34402B[I]<0 then OTHPAY5[I]=E34402B[I];
if E216005[I]=2 and -4<E34430[I]<0 then OTHPAY5[I]=E34430[I];
if E216005[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E216005[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;

/* if other compensation is corrected in the later part */
do I=1 to 9;
if E226049[I]=1 then do;
if E22617[I]=1 and E22632[I]>=0 then OTHPAY5[I]=E22632[I]; /* with overtime*/
if E22617[I]=2 and E22632[I]>=0 and E34430[I]>0 and E34428[I]>0 then
    OTHPAY5[I]=E22632[I]*E34430[I]/E34428[I];
if E22617[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22632[I]>=0 and E34428[I]>0 then
    OTHPAY5[I]=E22632[I]/E34428[I];
if E22617[I]=4 and E22632[I]>=0 and E34428[I]>0 then OTHPAY5[I]=E22632[I]/(2*E34428[I]);
if E22617[I]=5 and E22632[I]>=0 and E34428[I]>0 then OTHPAY5[I]=E22632[I]/(4.3*E34428[I]);
if E22617[I]=6 and E22632[I]>=0 and E34428[I]>0 and E35600[I]>0 then
    OTHPAY5[I]=E22632[I]/(E35600[I]*E34428[I]);
if E22617[I]=8 and E22632[I]>=0 and E34428[I]>0 then OTHPAY5[I]=E22632[I]/(2.15*E34428[I]);
if E22617[I] in (9,14) then OTHPAY5[I]=0;
/*missing value*/
if -4<E34428[I]<0 then OTHPAY5[I]=E34428[I];
if -4<E22632[I]<0 then OTHPAY5[I]=E22632[I];
if E34428[I]=0 then OTHPAY5[I]=-3;

if E22617[I]=1 and E22632[I]>=0 then OTHPAY5[I]=E22632[I]; /* without overtime*/
if E22617[I]=2 and E22632[I]>=0 and E34402B[I]>0 and E34402[I]>0 then
    OTHPAY5[I]=E22632[I]*E34402B[I]/E34402[I];
if E22617[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E22632[I]>=0 and E34402[I]>0 then
    OTHPAY5[I]=E22632[I]/E34402[I];
if E22617[I]=4 and E22632[I]>=0 and E34402[I]>0 then OTHPAY5[I]=E22632[I]/(2*E34402[I]);
if E22617[I]=5 and E22632[I]>=0 and E34402[I]>0 then OTHPAY5[I]=E22632[I]/(4.3*E34402[I]);
if E22617[I]=6 and E22632[I]>=0 and E34402[I]>0 and E35600[I]>0 then
    OTHPAY5[I]=E22632[I]/(E35600[I]*E34402[I]);
if E22617[I]=8 and E22632[I]>=0 and E34402[I]>0 then OTHPAY5[I]=E22632[I]/(2.15*E34402[I]);
if E22617[I] in (9,14) then OTHPAY5[I]=0;
/*missing value*/
if -4<E34402[I]<0 then OTHPAY5[I]=E34402[I];
if E34402[I]=0 then OTHPAY5[I]=-3;
if E22617[I]=2 and E34402B[I] le 0 and E34430[I] le 0 then OTHPAY5[I]=-3;
if E22617[I]=2 and -4<E34402B[I]<0 then OTHPAY5[I]=E34402B[I];

```

Appendix 2: Employment Variable Creation

```
if E22617[I]=2 and -4<E34430[I]<0 then OTHPAY5[I]=E34430[I];
if E22617[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E22617[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;
end;

/***** do it for non-overtime payment if answered in the later part *****/

/***** FOR TIPS *****/
do I=1 to 9; /*with overtime*/
if E98429[I]>0 and E102051[I]=1 and E102141[I]>=0 then OTHPAY1[I]=E102141[I];
if E98429[I]>0 and E102051[I]=2 and E98429E[I]>0 and E102141[I]>=0 then
    OTHPAY1[I]=(E102141[I]*E98429E[I])/E98429[I];
if E98429[I]>0 and E102051[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102141[I]>=0 then
    OTHPAY1[I]=E102141[I]/E98429[I];
if E98429[I]>0 and E102051[I]=4 and E102141[I]>=0 then OTHPAY1[I]=E102141[I]/(2*E98429[I]);
if E98429[I]>0 and E102051[I]=5 and E102141[I]>=0 then OTHPAY1[I]=E102141[I]/(4.3*E98429[I]);
if E98429[I]>0 and E102051[I]=6 and E102141[I]>=0 and E99500[I]>0 then
    OTHPAY1[I]=E102141[I]/(E99500[I]*E98429[I]);
if E98429[I]>0 and E102051[I]=8 and E102141[I]>=0 then OTHPAY1[I]=E102141[I]/(2.15*E98429[I]);
/*missing value*/
if E102141[I]>=0 and -4<E98429[I]<0 then OTHPAY1[I]=E98429[I];
if -4<E102141[I]<0 then OTHPAY1[I]=E102141[I];
if E102141[I]>=0 and E98429[I]=0 then OTHPAY1[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E98402[I]>0 and E102051[I]=1 and E102141[I]>=0 then OTHPAY1[I]=E102141[I];
if E98402[I]>0 and E102051[I]=2 and E98402D[I]>0 and E102141[I]>=0 then
    OTHPAY1[I]=(E102141[I]*E98402D[I])/E98402[I];
if E98402[I]>0 and E102051[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102141[I]>=0 then
    OTHPAY1[I]=E102141[I]/E98402[I];
if E98402[I]>0 and E102051[I]=4 and E102141[I]>=0 then OTHPAY1[I]=E102141[I]/(2*E98402[I]);
if E98402[I]>0 and E102051[I]=5 and E102141[I]>=0 then OTHPAY1[I]=E102141[I]/(4.3*E98402[I]);
if E98402[I]>0 and E102051[I]=6 and E102141[I]>=0 and E99500[I]>0 then
    OTHPAY1[I]=E102141[I]/(52*E98402[I]);
if E98402[I]>0 and E102051[I]=8 and E102141[I]>=0 then OTHPAY1[I]=E102141[I]/(2.15*E98402[I]);
if E102051[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY1[I]=OTHPAY1[I]+1;
if E102051[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/
if E102141[I]>=0 and -4<E98402[I]<0 then OTHPAY1[I]=E98402[I];
if E102141[I]>=0 and E98402[I]=0 then OTHPAY1[I]=-3;
if E102051[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY1[I]=-3;
if E102051[I]=2 and -4<E98402D[I]<0 then OTHPAY1[I]=E98402D[I];
if E102051[I]=2 and -4<E98429E[I]<0 then OTHPAY1[I]=E98429E[I];
if E102051[I]=6 and E99500[I] le 0 then OTHPAY1[I]=-3;
if E102051[I]=6 and -4<E99500[I]<0 then OTHPAY1[I]=E99500[I];
end;

/* if tips is corrected in the later part */
do I=1 to 9;
if E102255[I]=1 then do;
if E100234[I]=1 and E100250[I]>=0 then OTHPAY1[I]=E100250[I]; /* with overtime*/
if E100234[I]=2 and E100250[I]>=0 and E98429E[I]>0 and E98429[I]>0 then
    OTHPAY1[I]=E100250[I]*E98429E[I]/E98429[I];
if E100234[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100250[I]>=0 and E98429[I]>0 then
    OTHPAY1[I]=E100250[I]/E98429[I];
```

Appendix 2: Employment Variable Creation

```
if E100234[I]=4 and E100250[I]>=0 and E98429[I]>0 then OTHPAY1[I]=E100250[I]/(2*E98429[I]);
if E100234[I]=5 and E100250[I]>=0 and E98429[I]>0 then OTHPAY1[I]=E100250[I]/(4.3*E98429[I]);
if E100234[I]=6 and E100250[I]>=0 and E98429[I]>0 and E99500[I]>0 then
    OTHPAY1[I]=E100250[I]/(E99500[I]*E98429[I]);
if E100234[I]=8 and E100250[I]>=0 and E98429[I]>0 then OTHPAY1[I]=E100250[I]/(2.15*E98429[I]);
if E100234[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/
if -4<E98429[I]<0 then OTHPAY1[I]=E98429[I];
if -4<E100250[I]<0 then OTHPAY1[I]=E100250[I];
if E98429[I]=0 then OTHPAY1[I]=-3;

if E100234[I]=1 and E100250[I]>=0 then OTHPAY1[I]=E100250[I]; /* without overtime*/
if E100234[I]=2 and E100250[I]>=0 and E98402D[I]>0 and E98402[I]>0 then
    OTHPAY1[I]=E100250[I]*E98402D[I]/E98402[I];
if E100234[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100250[I]>=0 and E98402[I]>0 then
    OTHPAY1[I]=E100250[I]/E98402[I];
if E100234[I]=4 and E100250[I]>=0 and E98402[I]>0 then OTHPAY1[I]=E100250[I]/(2*E98402[I]);
if E100234[I]=5 and E100250[I]>=0 and E98402[I]>0 then OTHPAY1[I]=E100250[I]/(4.3*E98402[I]);
if E100234[I]=6 and E100250[I]>=0 and E98402[I]>0 and E99500[I]>0 then
    OTHPAY1[I]=E100250[I]/(E99500[I]*E98402[I]);
if E100234[I]=8 and E100250[I]>=0 and E98402[I]>0 then OTHPAY1[I]=E100250[I]/(2.15*E98402[I]);
if E100234[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/
if -4<E98402[I]<0 then OTHPAY1[I]=E98402[I];
if E98402[I]=0 then OTHPAY1[I]=-3;
if E100234[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY1[I]=-3;
if E100234[I]=2 and -4<E98402D[I]<0 then OTHPAY1[I]=E98402D[I];
if E100234[I]=2 and -4<E98429E[I]<0 then OTHPAY1[I]=E98429E[I];
if E100234[I]=6 and E99500[I] le 0 then OTHPAY1[I]=-3;
if E100234[I]=6 and -4<E99500[I]<0 then OTHPAY1[I]=E99500[I];
end;
end;

/***** FOR COMMISSIONS *****/
do I=1 to 9; /*with overtime*/
if E98429[I]>0 and E102052[I]=1 and E102142[I]>=0 then OTHPAY2[I]=E102142[I];
if E98429[I]>0 and E102052[I]=2 and E98429E[I]>0 and E102142[I]>=0 then
    OTHPAY2[I]=(E102142[I]*E98429E[I])/E98429[I];
if E98429[I]>0 and E102052[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102142[I]>=0 then
    OTHPAY2[I]=E102142[I]/E98429[I];
if E98429[I]>0 and E102052[I]=4 and E102142[I]>=0 then OTHPAY2[I]=E102142[I]/(2*E98429[I]);
if E98429[I]>0 and E102052[I]=5 and E102142[I]>=0 then OTHPAY2[I]=E102142[I]/(4.3*E98429[I]);
if E98429[I]>0 and E102052[I]=6 and E102142[I]>=0 and E99500[I]>0 then
    OTHPAY2[I]=E102142[I]/(E99500[I]*E98429[I]);
if E98429[I]>0 and E102052[I]=8 and E102142[I]>=0 then OTHPAY2[I]=E102142[I]/(2.15*E98429[I]);
/*missing value*/
if E102142[I]>=0 and -4<E98429[I]<0 then OTHPAY2[I]=E98429[I];
if -4<E102142[I]<0 then OTHPAY2[I]=E102142[I];
if E102142[I]>=0 and E98429[I]=0 then OTHPAY2[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E98402[I]>0 and E102052[I]=1 and E102142[I]>=0 then OTHPAY2[I]=E102142[I];
if E98402[I]>0 and E102052[I]=2 and E98402D[I]>0 and E102142[I]>=0 then
    OTHPAY2[I]=(E102142[I]*E98402D[I])/E98402[I];
if E98402[I]>0 and E102052[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102142[I]>=0 then
    OTHPAY2[I]=E102142[I]/E98402[I];
```

Appendix 2: Employment Variable Creation

```
if E98402[I]>0 and E102052[I]=4 and E102142[I]>=0 then OTHPAY2[I]=E102142[I]/(2*E98402[I]);
if E98402[I]>0 and E102052[I]=5 and E102142[I]>=0 then OTHPAY2[I]=E102142[I]/(4.3*E98402[I]);
if E98402[I]>0 and E102052[I]=6 and E102142[I]>=0 and E99500[I]>0 then
    OTHPAY2[I]=E102142[I]/(E99500[I]*E98402[I]);
if E98402[I]>0 and E102052[I]=8 and E102142[I]>=0 then OTHPAY2[I]=E102142[I]/(2.15*E98402[I]);
if E102052[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY2[I]=OTHPAY2[I]+1;
if E102052[I] in (9,14) then OTHPAY2[I]=0;
/*missing value*/
if E102142[I]>=0 and -4<E98402[I]<0 then OTHPAY2[I]=E98402[I];
if E102142[I]>=0 and E98402[I]=0 then OTHPAY2[I]=-3;
if E102052[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY2[I]=-3;
if E102052[I]=2 and -4<E98402D[I]<0 then OTHPAY2[I]=E98402D[I];
if E102052[I]=2 and -4<E98429E[I]<0 then OTHPAY2[I]=E98429E[I];
if E102052[I]=6 and E99500[I] le 0 then OTHPAY2[I]=-3;
if E102052[I]=6 and -4<E99500[I]<0 then OTHPAY2[I]=E99500[I];
end;

/* if commissions is corrected in the later part */
do I=1 to 9;
if E102256[I]=1 then do;
if E100235[I]=1 and E100251[I]>=0 then OTHPAY2[I]=E100251[I]; /* with overtime*/
if E100235[I]=2 and E100251[I]>=0 and E98429E[I]>0 and E98429[I]>0 then
    OTHPAY2[I]=E100251[I]*E98429E[I]/E98429[I];
if E100235[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100251[I]>=0 and E98429[I]>0 then
    OTHPAY2[I]=E100251[I]/E98429[I];
if E100235[I]=4 and E100251[I]>=0 and E98429[I]>0 then OTHPAY2[I]=E100251[I]/(2*E98429[I]);
if E100235[I]=5 and E100251[I]>=0 and E98429[I]>0 then OTHPAY2[I]=E100251[I]/(4.3*E98429[I]);
if E100235[I]=6 and E100251[I]>=0 and E98429[I]>0 and E99500[I]>0 then
    OTHPAY2[I]=E100251[I]/(E99500[I]*E98429[I]);
if E100235[I]=8 and E100251[I]>=0 and E98429[I]>0 then OTHPAY2[I]=E100251[I]/(2.15*E98429[I]);
if E100235[I] in (9,14) then OTHPAY2[I]=0;
/*missing value*/
if -4<E98429[I]<0 then OTHPAY2[I]=E98429[I];
if -4<E100251[I]<0 then OTHPAY2[I]=E100251[I];
if E98429[I]=0 then OTHPAY2[I]=-3;

if E100235[I]=1 and E100251[I]>=0 then OTHPAY2[I]=E100251[I]; /* without overtime*/
if E100235[I]=2 and E100251[I]>=0 and E98402D[I]>0 and E98402[I]>0 then
    OTHPAY2[I]=E100251[I]*E98402D[I]/E98402[I];
if E100235[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100251[I]>=0 and E98402[I]>0 then
    OTHPAY2[I]=E100251[I]/E98402[I];
if E100235[I]=4 and E100251[I]>=0 and E98402[I]>0 then OTHPAY2[I]=E100251[I]/(2*E98402[I]);
if E100235[I]=5 and E100251[I]>=0 and E98402[I]>0 then OTHPAY2[I]=E100251[I]/(4.3*E98402[I]);
if E100235[I]=6 and E100251[I]>=0 and E98402[I]>0 and E99500[I]>0 then
    OTHPAY2[I]=E100251[I]/(E99500[I]*E98402[I]);
if E100235[I]=8 and E100251[I]>=0 and E98402[I]>0 then OTHPAY2[I]=E100251[I]/(2.15*E98402[I]);
if E100235[I] in (9,14) then OTHPAY2[I]=0;
/*missing value*/
if -4<E98402[I]<0 then OTHPAY2[I]=E98402[I];
if E98402[I]=0 then OTHPAY2[I]=-3;
if E100235[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY2[I]=-3;
if E100235[I]=2 and -4<E98402D[I]<0 then OTHPAY2[I]=E98402D[I];
if E100235[I]=2 and -4<E98429E[I]<0 then OTHPAY2[I]=E98429E[I];
if E100235[I]=6 and E99500[I] le 0 then OTHPAY2[I]=-3;
if E100235[I]=6 and -4<E99500[I]<0 then OTHPAY2[I]=E99500[I];
end;
end;
```

Appendix 2: Employment Variable Creation

```
/****** FOR BONUSES *****/
do I=1 to 9; /* with overtime*/
if E98429[I]>0 and E102053[I]=1 and E102143[I]>=0 then OTHPAY3[I]=E102143[I];
if E98429[I]>0 and E102053[I]=2 and E98429E[I]>0 and E102143[I]>=0 then
    OTHPAY3[I]=(E102143[I]*E98429E[I])/E98429[I];
if E98429[I]>0 and E102053[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102143[I]>=0 then
    OTHPAY3[I]=E102143[I]/E98429[I];
if E98429[I]>0 and E102053[I]=4 and E102143[I]>=0 then OTHPAY3[I]=E102143[I]/(2*E98429[I]);
if E98429[I]>0 and E102053[I]=5 and E102143[I]>=0 then OTHPAY3[I]=E102143[I]/(4.3*E98429[I]);
if E98429[I]>0 and E102053[I]=6 and E102143[I]>=0 and E99500[I]>0 then
    OTHPAY3[I]=E102143[I]/(E99500[I]*E98429[I]);
if E98429[I]>0 and E102053[I]=8 and E102143[I]>=0 then OTHPAY3[I]=E102143[I]/(2.15*E98429[I]);
/*missing value*/
if E102143[I]>=0 and -4<E98429[I]<0 then OTHPAY3[I]=E98429[I];
if -4<E102143[I]<0 then OTHPAY3[I]=E102143[I];
if E102143[I]>=0 and E98429[I]=0 then OTHPAY3[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E98402[I]>0 and E102053[I]=1 and E102143[I]>=0 then OTHPAY3[I]=E102143[I];
if E98402[I]>0 and E102053[I]=2 and E98402D[I]>0 and E102143[I]>=0 then
    OTHPAY3[I]=(E102143[I]*E98402D[I])/E98402[I];
if E98402[I]>0 and E102053[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102143[I]>=0 then
    OTHPAY3[I]=E102143[I]/E98402[I];
if E98402[I]>0 and E102053[I]=4 and E102143[I]>=0 then OTHPAY3[I]=E102143[I]/(2*E98402[I]);
if E98402[I]>0 and E102053[I]=5 and E102143[I]>=0 then OTHPAY3[I]=E102143[I]/(4.3*E98402[I]);
if E98402[I]>0 and E102053[I]=6 and E102143[I]>=0 and E99500[I]>0 then
    OTHPAY3[I]=E102143[I]/(E99500[I]*E98402[I]);
if E98402[I]>0 and E102053[I]=8 and E102143[I]>=0 then OTHPAY3[I]=E102143[I]/(2.15*E98402[I]);
if E102053[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPF3[I]=OTHPF3[I]+1;
if E102053[I] in (9,14) then OTHPAY3[I]=0;
/*missing value*/
if E102143[I]>=0 and -4<E98402[I]<0 then OTHPAY3[I]=E98402[I];
if E102143[I]>=0 and E98402[I]=0 then OTHPAY3[I]=-3;
if E102053[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY3[I]=-3;
if E102053[I]=2 and -4<E98402D[I]<0 then OTHPAY3[I]=E98402D[I];
if E102053[I]=2 and -4<E98429E[I]<0 then OTHPAY3[I]=E98429E[I];
if E102053[I]=6 and E99500[I] le 0 then OTHPAY3[I]=-3;
if E102053[I]=6 and -4<E99500[I]<0 then OTHPAY3[I]=E99500[I];
end;

/* if bonus is corrected in the later part */
do I=1 to 9;
if E102257[I]=1 then do;
if E100236[I]=1 and E100252[I]>=0 then OTHPAY3[I]=E100252[I]; /* with overtime*/
if E100236[I]=2 and E100252[I]>=0 and E98429E[I]>0 and E98429[I]>0 then
    OTHPAY3[I]=E100252[I]*E98429E[I]/E98429[I];
if E100236[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100252[I]>=0 and E98429[I]>0 then
    OTHPAY3[I]=E100252[I]/E98429[I];
if E100236[I]=4 and E100252[I]>=0 and E98429[I]>0 then OTHPAY3[I]=E100252[I]/(2*E98429[I]);
if E100236[I]=5 and E100252[I]>=0 and E98429[I]>0 then OTHPAY3[I]=E100252[I]/(4.3*E98429[I]);
if E100236[I]=6 and E100252[I]>=0 and E98429[I]>0 and E99500[I]>0 then
    OTHPAY3[I]=E100252[I]/(E99500[I]*E98429[I]);
if E100236[I]=8 and E100252[I]>=0 and E98429[I]>0 then OTHPAY3[I]=E100252[I]/(2.15*E98429[I]);
if E100236[I] in (9,14) then OTHPAY3[I]=0;
/*missing value*/
```

Appendix 2: Employment Variable Creation

```
if -4<E98429[I]<0 then OTHPAY3[I]=E98429[I];
if -4<E100252[I]<0 then OTHPAY3[I]=E100252[I];
if E98429[I]=0 then OTHPAY3[I]=-3;

if E100236[I]=1 and E100252[I]>=0 then OTHPAY3[I]=E100252[I]; /* without overtime*/
if E100236[I]=2 and E100252[I]>=0 and E98402D[I]>0 and E98402[I]>0 then
    OTHPAY3[I]=E100252[I]*E98402D[I]/E98402[I];
if E100236[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100252[I]>=0 and E98402[I]>0 then
    OTHPAY3[I]=E100252[I]/E98402[I];
if E100236[I]=4 and E100252[I]>=0 and E98402[I]>0 then OTHPAY3[I]=E100252[I]/(2*E98402[I]);
if E100236[I]=5 and E100252[I]>=0 and E98402[I]>0 then OTHPAY3[I]=E100252[I]/(4.3*E98402[I]);
if E100236[I]=6 and E100252[I]>=0 and E98402[I]>0 and E99500[I]>0 then
    OTHPAY3[I]=E100252[I]/(E99500[I]*E98402[I]);
if E100236[I]=8 and E100252[I]>=0 and E98402[I]>0 then OTHPAY3[I]=E100252[I]/(2.15*E98402[I]);
if E100236[I] in (9,14) then OTHPAY3[I]=0;
/*missing value*/
if -4<E98402[I]<0 then OTHPAY3[I]=E98402[I];
if E98402[I]=0 then OTHPAY3[I]=-3;
if E100236[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY3[I]=-3;
if E100236[I]=2 and -4<E98402D[I]<0 then OTHPAY3[I]=E98402D[I];
if E100236[I]=2 and -4<E98429E[I]<0 then OTHPAY3[I]=E98429E[I];
if E100236[I]=6 and E99500[I] le 0 then OTHPAY3[I]=-3;
if E100236[I]=6 and -4<E99500[I]<0 then OTHPAY3[I]=E99500[I];
end;
end;

/***** FOR INCENTIVE PAY *****/
do I=1 to 9; /* with overtime*/
if E98429[I]>0 and E102054[I]=1 and E102144[I]>=0 then OTHPAY4[I]=E102144[I];
if E98429[I]>0 and E102054[I]=2 and E98429E[I]>0 and E102144[I]>=0 then
    OTHPAY4[I]=(E102144[I]*E98429E[I])/E98429[I];
if E98429[I]>0 and E102054[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102144[I]>=0 then
    OTHPAY4[I]=E102144[I]/E98429[I];
if E98429[I]>0 and E102054[I]=4 and E102144[I]>=0 then OTHPAY4[I]=E102144[I]/(2*E98429[I]);
if E98429[I]>0 and E102054[I]=5 and E102144[I]>=0 then OTHPAY4[I]=E102144[I]/(4.3*E98429[I]);
if E98429[I]>0 and E102054[I]=6 and E102144[I]>=0 and E99500[I]>0 then
    OTHPAY4[I]=E102144[I]/(E99500[I]*E98429[I]);
if E98429[I]>0 and E102054[I]=8 and E102144[I]>=0 then OTHPAY4[I]=E102144[I]/(2.15*E98429[I]);
/*missing value*/
if E102144[I]>=0 and -4<E98429[I]<0 then OTHPAY4[I]=E98429[I];
if -4<E102144[I]<0 then OTHPAY4[I]=E102144[I];
if E102144[I]>=0 and E98429[I]=0 then OTHPAY4[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E98402[I]>0 and E102054[I]=1 and E102144[I]>=0 then OTHPAY4[I]=E102144[I];
if E98402[I]>0 and E102054[I]=2 and E98402D[I]>0 and E102144[I]>=0 then
    OTHPAY4[I]=(E102144[I]*E98402D[I])/E98402[I];
if E98402[I]>0 and E102054[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102144[I]>=0 then
    OTHPAY4[I]=E102144[I]/E98402[I];
if E98402[I]>0 and E102054[I]=4 and E102144[I]>=0 then OTHPAY4[I]=E102144[I]/(2*E98402[I]);
if E98402[I]>0 and E102054[I]=5 and E102144[I]>=0 then OTHPAY4[I]=E102144[I]/(4.3*E98402[I]);
if E98402[I]>0 and E102054[I]=6 and E102144[I]>=0 and E99500[I]>0 then
    OTHPAY4[I]=E102144[I]/(E99500[I]*E98402[I]);
if E98402[I]>0 and E102054[I]=8 and E102144[I]>=0 then OTHPAY4[I]=E102144[I]/(2.15*E98402[I]);
if E102054[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY4[I]=OTHPAY4[I]+1;
if E102054[I] in (9,14) then OTHPAY4[I]=0;
```

Appendix 2: Employment Variable Creation

```
*missing value*/;
if E102144[I]>=0 and -4<E98402[I]<0 then OTHPAY4[I]=E98402[I];
if E102144[I]>=0 and E98402[I]=0 then OTHPAY4[I]=-3;
if E102054[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY4[I]=-3;
if E102054[I]=2 and -4<E98402D[I]<0 then OTHPAY4[I]=E98402D[I];
if E102054[I]=2 and -4<E98429E[I]<0 then OTHPAY4[I]=E98429E[I];
if E102054[I]=6 and E99500[I] le 0 then OTHPAY4[I]=-3;
if E102054[I]=6 and -4<E99500[I]<0 then OTHPAY4[I]=E99500[I];
end;

/* if incentive pay is corrected in the later part */
do I=1 to 9;
if E102258[I]=1 then do;
if E100237[I]=1 and E100253[I]>=0 then OTHPAY4[I]=E100253[I]; /* with overtime*/
if E100237[I]=2 and E100253[I]>=0 and E98429E[I]>0 and E98429[I]>0 then
    OTHPAY4[I]=E100253[I]*E98429E[I]/E98429[I];
if E100237[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100253[I]>=0 and E98429[I]>0 then
    OTHPAY4[I]=E100253[I]/E98429[I];
if E100237[I]=4 and E100253[I]>=0 and E98429[I]>0 then OTHPAY4[I]=E100253[I]/(2*E98429[I]);
if E100237[I]=5 and E100253[I]>=0 and E98429[I]>0 then OTHPAY4[I]=E100253[I]/(4.3*E98429[I]);
if E100237[I]=6 and E100253[I]>=0 and E98429[I]>0 and E99500[I]>0 then
    OTHPAY4[I]=E100253[I]/(E99500[I]*E98429[I]);
if E100237[I]=8 and E100253[I]>=0 and E98429[I]>0 then OTHPAY4[I]=E100253[I]/(2.15*E98429[I]);
if E100237[I] in (9,14) then OTHPAY4[I]=0;
/*missing value*/
if -4<E98429[I]<0 then OTHPAY4[I]=E98429[I];
if -4<E100253[I]<0 then OTHPAY4[I]=E100253[I];
if E98429[I]=0 then OTHPAY4[I]=-3;

if E100237[I]=1 and E100253[I]>=0 then OTHPAY4[I]=E100253[I]; /* without overtime*/
if E100237[I]=2 and E100253[I]>=0 and E98402D[I]>0 and E98402[I]>0 then
    OTHPAY4[I]=E100253[I]*E98402D[I]/E98402[I];
if E100237[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100253[I]>=0 and E98402[I]>0 then
    OTHPAY4[I]=E100253[I]/E98402[I];
if E100237[I]=4 and E100253[I]>=0 and E98402[I]>0 then OTHPAY4[I]=E100253[I]/(2*E98402[I]);
if E100237[I]=5 and E100253[I]>=0 and E98402[I]>0 then OTHPAY4[I]=E100253[I]/(4.3*E98402[I]);
if E100237[I]=6 and E100253[I]>=0 and E98402[I]>0 and E99500[I]>0 then
    OTHPAY4[I]=E100253[I]/(E99500[I]*E98402[I]);
if E100237[I]=8 and E100253[I]>=0 and E98402[I]>0 then OTHPAY4[I]=E100253[I]/(2.15*E98402[I]);
if E100237[I] in (9,14) then OTHPAY4[I]=0;
/*missing value*/
if -4<E98402[I]<0 then OTHPAY4[I]=E98402[I];
if E98402[I]=0 then OTHPAY4[I]=-3;
if E100237[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY4[I]=-3;
if E100237[I]=2 and -4<E98402D[I]<0 then OTHPAY4[I]=E98402D[I];
if E100237[I]=2 and -4<E98429E[I]<0 then OTHPAY4[I]=E98429E[I];
if E100237[I]=6 and E99500[I] le 0 then OTHPAY4[I]=-3;
if E100237[I]=6 and -4<E99500[I]<0 then OTHPAY4[I]=E99500[I];
end;
end;

/****** FOR OTHERS *****/
do I=1 to 9; /*with overtime*/
if E98429[I]>0 and E102055[I]=1 and E102145[I]>=0 then OTHPAY5[I]=E102145[I];
if E98429[I]>0 and E102055[I]=2 and E98429E[I]>0 and E102145[I]>=0 then
    OTHPAY5[I]=(E102145[I]*E98429E[I])/E98429[I];
```


Appendix 2: Employment Variable Creation

```

if E98429[I]>0 and E102055[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102145[I]>=0 then
  OTHPAY5[I]=E102145[I]/E98429[I];
if E98429[I]>0 and E102055[I]=4 and E102145[I]>=0 then OTHPAY5[I]=E102145[I]/(2*E98429[I]);
if E98429[I]>0 and E102055[I]=5 and E102145[I]>=0 then OTHPAY5[I]=E102145[I]/(4.3*E98429[I]);
if E98429[I]>0 and E102055[I]=6 and E102145[I]>=0 and E99500[I]>0 then
  OTHPAY5[I]=E102145[I]/(E99500[I]*E98429[I]);
if E98429[I]>0 and E102055[I]=8 and E102145[I]>=0 then OTHPAY5[I]=E102145[I]/(2.15*E98429[I]);
/*missing value*/
if E102145[I]>=0 and -4<E98429[I]<0 then OTHPAY5[I]=E98429[I];
if -4<E102145[I]<0 then OTHPAY5[I]=E102145[I];
if E102145[I]>=0 and E98429[I]=0 then OTHPAY5[I]=-3;
end;

do I=1 to 9; /*without overtime*/
if E98402[I]>0 and E102055[I]=1 and E102145[I]>=0 then OTHPAY5[I]=E102145[I];
if E98402[I]>0 and E102055[I]=2 and E98402D[I]>0 and E102145[I]>=0 then
  OTHPAY5[I]=(E102145[I]*E98402D[I])/E98402[I];
if E98402[I]>0 and E102055[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E102145[I]>=0 then
  OTHPAY5[I]=E102145[I]/E98402[I];
if E98402[I]>0 and E102055[I]=4 and E102145[I]>=0 then OTHPAY5[I]=E102145[I]/(2*E98402[I]);
if E98402[I]>0 and E102055[I]=5 and E102145[I]>=0 then OTHPAY5[I]=E102145[I]/(4.3*E98402[I]);
if E98402[I]>0 and E102055[I]=6 and E102145[I]>=0 and E99500[I]>0 then
  OTHPAY5[I]=E102145[I]/(E99500[I]*E98402[I]);
if E98402[I]>0 and E102055[I]=8 and E102145[I]>=0 then OTHPAY5[I]=E102145[I]/(2.15*E98402[I]);
if E102055[I] in (0,7,12,13,15,16,17,14,99,21,22,23,24,25,26,28) then OTHPAY5[I]=OTHPAY5[I]+1;
if E102055[I] in (9,14) then OTHPAY5[I]=0;
/*missing value*/;
if E102145[I]>=0 and -4<E98402[I]<0 then OTHPAY5[I]=E98402[I];
if E102145[I]>=0 and E98402[I]=0 then OTHPAY5[I]=-3;
if E102055[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY5[I]=-3;
if E102055[I]=2 and -4<E98402D[I]<0 then OTHPAY5[I]=E98402D[I];
if E102055[I]=2 and -4<E98429E[I]<0 then OTHPAY5[I]=E98429E[I];
if E102055[I]=6 and E99500[I] le 0 then OTHPAY5[I]=-3;
if E102055[I]=6 and -4<E99500[I]<0 then OTHPAY5[I]=E99500[I];
end;

/* if other compensation is corrected in the later part */
do I=1 to 9;
if E102259[I]=1 then do;
if E100239[I]=1 and E100254[I]>=0 then OTHPAY5[I]=E100254[I]; /* with overtime*/
if E100239[I]=2 and E100254[I]>=0 and E98429E[I]>0 and E98429[I]>0 then
  OTHPAY5[I]=E100254[I]*E98429E[I]/E98429[I];
if E100239[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100254[I]>=0 and E98429[I]>0 then
  OTHPAY5[I]=E100254[I]/E98429[I];
if E100239[I]=4 and E100254[I]>=0 and E98429[I]>0 then OTHPAY5[I]=E100254[I]/(2*E98429[I]);
if E100239[I]=5 and E100254[I]>=0 and E98429[I]>0 then OTHPAY5[I]=E100254[I]/(4.3*E98429[I]);
if E100239[I]=6 and E100254[I]>=0 and E98429[I]>0 and E99500[I]>0 then
  OTHPAY5[I]=E100254[I]/(E99500[I]*E98429[I]);
if E100239[I]=8 and E100254[I]>=0 and E98429[I]>0 then OTHPAY5[I]=E100254[I]/(2.15*E98429[I]);
if E100239[I] in (9,14) then OTHPAY5[I]=0;
/*missing value*/
if -4<E98429[I]<0 then OTHPAY5[I]=E98429[I];
if -4<E100254[I]<0 then OTHPAY5[I]=E100254[I];
if E98429[I]=0 then OTHPAY5[I]=-3;

if E100239[I]=1 and E100254[I]>=0 then OTHPAY5[I]=E100254[I]; /* without overtime*/

```

Appendix 2: Employment Variable Creation

```
if E100239[I]=2 and E100254[I]>=0 and E98402D[I]>0 and E98402[I]>0 then
  OTHPAY5[I]=E100254[I]*E98402D[I]/E98402[I];
if E100239[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28,-2) and E100254[I]>=0 and E98402[I]>0 then
  OTHPAY5[I]=E100254[I]/E98402[I];
if E100239[I]=4 and E100254[I]>=0 and E98402[I]>0 then OTHPAY5[I]=E100254[I]/(2*E98402[I]);
if E100239[I]=5 and E100254[I]>=0 and E98402[I]>0 then OTHPAY5[I]=E100254[I]/(4.3*E98402[I]);
if E100239[I]=6 and E100254[I]>=0 and E98402[I]>0 and E99500[I]>0 then
  OTHPAY5[I]=E100254[I]/(E99500[I]*E98402[I]);
if E100239[I]=8 and E100254[I]>=0 and E98402[I]>0 then OTHPAY5[I]=E100254[I]/(2.15*E98402[I]);
if E100239[I] in (9,14) then OTHPAY5[I]=0;
/*missing value*/
if -4<E98402[I]<0 then OTHPAY5[I]=E98402[I];
if E98402[I]=0 then OTHPAY5[I]=-3;
if E100239[I]=2 and E98402D[I] le 0 and E98429E[I] le 0 then OTHPAY5[I]=-3;
if E100239[I]=2 and -4<E98402D[I]<0 then OTHPAY5[I]=E98402D[I];
if E100239[I]=2 and -4<E98429E[I]<0 then OTHPAY5[I]=E98429E[I];
if E100239[I]=6 and E99500[I] le 0 then OTHPAY5[I]=-3;
if E100239[I]=6 and -4<E99500[I]<0 then OTHPAY5[I]=E99500[I];
end;
end;

/**** Part 6: OVERALL START HOURLY COMPENSATION *****/

array HRCOMP HRCOMP01 HRCOMP02 HRCOMP03 HRCOMP04 HRCOMP05 HRCOMP06
HRCOMP07 HRCOMP08 HRCOMP09;
do I=1 to 9;
  HRCOMP[I]=0;
  if HRWG[I] ge 0 then HRCOMP[I]=HRCOMP[I]+HRWG[I];
  if OT[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OT[I];
  if OTHPAY1[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY1[I];
  if OTHPAY2[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY2[I];
  if OTHPAY3[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY3[I];
  if OTHPAY4[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY4[I];
  if OTHPAY5[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY5[I];
  if -4<HRWG[I]<0 or -4<OT[I]<0 or -4<OTHPAY1[I]<0 or -4<OTHPAY2[I]<0 or -4<OTHPAY3[I]<0 or -
    4<OTHPAY4[I]<0 or -4<OTHPAY5[I]<0 then HRCOMP[I]=-3;
  if HRWG[I]=-4 then HRCOMP[I]=-4;
end;

/***** Section 2: END WAGES FOR YOUTH *****/

/**** Part 1: End hourly rate of pay ****/

/*Respondents reporting an HOURLY wage*/
do I=1 to 9;
if (E37901B[I]=1 or E59900[I]=1) then do;
  if (E38013[I]=1 and E38014[I]=1) then do; /* without overtime at the beginning*/
    if E38023[I]>=0 then HRWAGE[I]=E38023[I];
    /*missing value*/
    if -4<E38023[I]<0 then HRWAGE[I]=E38023[I];
  end;
  if (E38106[I]=1 and E38107[I]=1) then do; /* with overtime at the beginning */
    if E38116[I]>=0 then HRWAGE[I]=E38116[I];
    /*missing value*/ if -4<E38106[I]<0 then HRWAGE[I]=E38106[I];
  end;
end;
end;
```

Appendix 2: Employment Variable Creation

end;

*/*Respondents reporting a DAILY wage*/*

do I=1 to 9;

if (E37901B[I]=1 or E59900[I]=1) then do;

if (E38013[I]=1 and E38014[I]=2) then do; */*without overtime at the beginning*/*

if E38023[I]>=0 and E3800B[I]=1 and E38027[I]>0 and E34402[I]>0 then

DAILY[I]=E38023[I]*E38027[I]/E34402[I];

if E38023[I]>=0 and E3800F[I]>0 and E38027[I]>0 then DAILY[I]=E38023[I]*E38027[I]/E3800F[I];

*/*missing value*/*

if E38023[I]>=0 and -4<E38027[I]<0 then DAILY[I]=E38027[I];

if E38023[I]>=0 and E3800B[I]=1 and -4<E34402[I]<0 then DAILY[I]=E34402[I];

if E38023[I]>=0 and -4<E3800F[I]<0 then DAILY[I]=E3800F[I];

if E38023[I]>=0 and (E34402[I]=0 or E3800F[I]=0) then DAILY[I]=-3;

end;

if (E38106[I]=1 and E38107[I]=2) then do; */*with overtime at the beginning*/*

if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and E38116B[I]>0 and E34428[I]>0 then

DAILY[I]=E38116B[I]*E38116[I]/E34428[I];

if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and E38103[I]>0 and E38116B[I]>0 then

DAILY[I]=E38116B[I]*E38116[I]/E38103[I];

*/*missing value*/*

if E38116[I]>=0 and -4<E38116B[I]<0 then DAILY[I]=E38116B[I];

if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and -4<E34428[I]<0 then DAILY[I]=E34428[I];

if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and -4<E38103[I]<0 then DAILY[I]=E38103[I];

if E38116[I]>=0 and (E34428[I]=0 or E38103[I]=0) then DAILY[I]=-3;

end;

end;

end;

*/*Respondents reporting a WEEKLY wage*/*

do I=1 to 9;

if (E37901B[I]=1 or E59900[I]=1) then do;

*/*without overtime at the beginning */*

if (E38013[I]=1 and E38014[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28)) then do;

if E38023[I]>=0 and E3800B[I]=1 and E34402[I]>0 then WEEKLY[I]=E38023[I]/E34402[I];

if E38023[I]>=0 and E3800F[I]>0 then WEEKLY[I]=E38023[I]/E3800F[I];

*/*missing value*/*

if E38023[I]>=0 and E3800B[I]=1 and -4<E34402[I]<0 then WEEKLY[I]=E34402[I];

if E38023[I]>=0 and -4<E3800F[I]<0 then WEEKLY[I]=E3800F[I];

if E38023[I]>=0 and (E34402[I]=0 or E3800F[I]=0) then WEEKLY[I]=-3;

end;

/ with overtime at the beginning */*

if (E38106[I]=1 and E38107[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28)) then do;

if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and E34428[I]>0 then

WEEKLY[I]=E38116[I]/E34428[I];

if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and E38103[I]>0 then WEEKLY[I]=E38116[I]/E38103[I];

*/*missing value*/*

if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and -4<E34428[I]<0 then WEEKLY[I]=E34428[I];

if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and -4<E38103[I]<0 then WEEKLY[I]=E38103[I];

if E38116[I]>=0 and (E34428[I]=0 or E38103[I]=0) then WEEKLY[I]=-3;

end;

end;

end;

*/*Respondents reporting a BIWEEKLY wage*/*

do I=1 to 9;

if (E37901B[I]=1 or E59900[I]=1) then do;

Appendix 2: Employment Variable Creation

```
if (E38013[I]=1 and E38014[I]=4) then do; /* without overtime at the beginning */
  if E38023[I]>=0 and E3800B[I]=1 and E34402[I]>0 then BIWKLY[I]=E38023[I]/(2*E34402[I]);
  if E38023[I]>=0 and E3800F[I]>0 then BIWKLY[I]=E38023[I]/(2*E3800F[I]);
  /*missing value*/
  if E38023[I]>=0 and E3800B[I]=1 and -4<E34402[I]<0 then BIWKLY[I]=E34402[I];
  if E38023[I]>=0 and -4<E3800F[I]<0 then BIWKLY[I]=E3800F[I];
  if E38023[I]>=0 and (E34402[I]=0 or E3800F[I]=0) then BIWKLY[I]=-3;
end;
if (E38106[I]=1 and E38107[I]=4) then do; /* with overtime at the beginning*/
  if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and E34428[I]>0 then
    BIWKLY[I]=E38116[I]/(2*E34428[I]);
  if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and E38103[I]>0 then
    BIWKLY[I]=E38116[I]/(2*E38103[I]);
  /*missing value*/
  if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and -4<E34428[I]<0 then BIWKLY[I]=E34428[I];
  if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and -4<E38103[I]<0 then BIWKLY[I]=E38103[I];
  if E38116[I]>=0 and (E34428[I]=0 or E38103[I]=0) then BIWKLY[I]=-3;
end;
end;
end;

/*Respondents reporting a MONTHLY wage*/
do I=1 to 9;
if (E37901B[I]=1 or E59900[I]=1) then do;
  if (E38013[I]=1 and E38014[I]=5) then do; /* without overtime at the beginning */
    if E38023[I]>=0 and E3800B[I]=1 and E34402[I]>0 then MONTH[I]=E38023[I]/(4.3*E34402[I]);
    if E38023[I]>=0 and E3800F[I]>0 then MONTH[I]=E38023[I]/(4.3*E3800F[I]);
    /*missing value*/
    if E38023[I]>=0 and E3800B[I]=1 and -4<E34402[I]<0 then MONTH[I]=E34402[I];
    if E38023[I]>=0 and -4<E3800F[I]<0 then MONTH[I]=E3800F[I];
    if E38023[I]>=0 and (E34402[I]=0 or E3800F[I]=0) then MONTH[I]=-3;
  end;
  if (E38106[I]=1 and E38107[I]=5) then do; /* with overtime at the beginning */
    if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and E34428[I]>0 then
      MONTH[I]=E38116[I]/(4.3*E34428[I]);
    if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and E38103[I]>0 then
      MONTH[I]=E38116[I]/(4.3*E38103[I]);
    /*missing value*/
    if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and -4<E34428[I]<0 then MONTH[I]=E34428[I];
    if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and -4<E38103[I]<0 then MONTH[I]=E38103[I];
    if E38116[I]>=0 and (E34428[I]=0 or E38103[I]=0) then MONTH[I]=-3;
  end;
end;
end;

/*Respondents reporting an ANNUAL wage*/
do I=1 to 9;
if (E37901B[I]=1 or E59900[I]=1) then do;
  if (E38013[I]=1 and E38014[I]=6) then do; /* without overtime at the beginning */
    if E38023[I]>=0 and E3800B[I]=1 and E34402[I]>0 and E35600[I]>0 then
      ANNUAL[I]=E38023[I]/(E35600[I]*E34402[I]);
    if E38023[I]>=0 and E3800F[I]>0 and E35600[I]>0 then ANNUAL[I]=E38023[I]/(E35600[I]*E3800F[I]);
    /*missing value*/
    if E38023[I]>=0 and E3800B[I]=1 and -4<E34402[I]<0 then ANNUAL[I]=E34402[I];
    if E38023[I]>=0 and -4<E3800F[I]<0 then ANNUAL[I]=E3800F[I];
    if E38023[I]>=0 and (E34402[I]=0 or E3800F[I]=0) then ANNUAL[I]=-3;
    if E35600[I] le 0 then ANNUAL[I]=-3;
  end;
end;
end;
```

Appendix 2: Employment Variable Creation

```
if -4<E35600[I]<0 then ANNUAL[I]=E35600[I];
end;
if (E38106[I]=1 and E38107[I]=6) then do; /* with overtime at the beginning */
if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and E34428[I]>0 and E35600[I]>0 then
  ANNUAL[I]=E38116[I]/(E35600[I]*E34428[I]);
if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and E38103[I]>0 and E35600[I]>0 then
  ANNUAL[I]=E38116[I]/(E35600[I]*E38103[I]);
/*missing value*/
if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and -4<E34428[I]<0 then ANNUAL[I]=E34428[I];
if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and -4<E38103[I]<0 then ANNUAL[I]=E38103[I];
if E38116[I]>=0 and (E34428[I]=0 or E38103[I]=0) then ANNUAL[I]=-3;
if E35600[I] le 0 then ANNUAL[I]=-3;
if -4<E35600[I]<0 then ANNUAL[I]=E35600[I];
end;
end;
end;

/*Respondents reporting a SEMIMONTHLY wage*/
do I=1 to 9;
if (E37901B[I]=1 or E59900[I]=1) then do;
if (E38013[I]=1 and E38014[I]=8) then do; /* without overtime at the beginning*/
if E38023[I]>=0 and E3800B[I]=1 and E34402[I]>0 then SEMIM[I]=E38023[I]/(2.15*E34402[I]);
if E38023[I]>=0 and E3800F[I]>0 then SEMIM[I]=E38023[I]/(2.15*E3800F[I]);
/*missing value*/
if E38023[I]>=0 and E3800B[I]=1 and -4<E34402[I]<0 then SEMIM[I]=E34402[I];
if E38023[I]>=0 and -4<E3800F[I]<0 then SEMIM[I]=E3800F[I];
if E38023[I]>=0 and (E34402[I]=0 or E3800F[I]=0) then SEMIM[I]=-3;
end;
if (E38106[I]=1 and E38107[I]=8) then do; /* with overtime at the beginning */
if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and E34428[I]>0 then
  SEMIM[I]=E38116[I]/(2.15*E34428[I]);
if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and E38103[I]>0 then
  SEMIM[I]=E38116[I]/(2.15*E38103[I]);
/*missing value*/
if E38116[I]>=0 and (E38102[I] ne 1 and E38102[I] ne 3) and -4<E34428[I]<0 then SEMIM[I]=E34428[I];
if E38116[I]>=0 and (E38102[I]=1 or E38102[I]=3) and -4<E38103[I]<0 then SEMIM[I]=E38103[I];
if E38116[I]>=0 and (E34428[I]=0 or E38103[I]=0) then SEMIM[I]=-3;
end;
end;
end;

**** Part 2: Create Hourly Rate of Pay based on the from end wage ****/

do I=1 to 9;
if E37901B[I]=1 or E59900[I]=1 then do;
if ANNUAL[I] ge 0 then HRWG[I]=ANNUAL[I];
if MONTH[I] ge 0 then HRWG[I]=MONTH[I];
if BIWKLY[I] ge 0 then HRWG[I]=BIWKLY[I];
if WEEKLY[I] ge 0 then HRWG[I]=WEEKLY[I];
if DAILY[I] ge 0 then HRWG[I]=DAILY[I];
if HRWAGE[I] ge 0 then HRWG[I]=HRWAGE[I];
if SEMIM[I] ge 0 then HRWG[I]=SEMIM[I];
if HRWAGE[I] eq -1 or DAILY[I]=-1 or WEEKLY[I] eq -1 or BIWKLY[I] eq -1 or MONTH[I] eq -1 or
  ANNUAL[I] eq -1 or SEMIM[I]=-1 then HRWG[I]=-1;
if HRWAGE[I] eq -2 or DAILY[I]=-2 or WEEKLY[I] eq -2 or BIWKLY[I] eq -2 or MONTH[I] eq -2 or
  ANNUAL[I] eq -2 or SEMIM[I]=-2 then HRWG[I]=-2;
```

Appendix 2: Employment Variable Creation

```
if HRWAGE[I] eq -3 or DAILY[I]=-3 or WEEKLY[I] eq -3 or BIWKLY[I] eq -3 or MONTH[I] eq -3 or
  ANNUAL[I] eq -3 or SEMIM[I]=-3 then HRWG[I]=-3;
end;
end;

/*set up the hourly wage for youths who report their wage in other manners */
do I=1 to 9;
if (E37901B[I]=1 or E59900[I]=1) then do;

/* if job was not offering any comp at the end */
  if (E38013[I]=1 and E38014[I] in (7,0,12,13,14,15,16,17,99,21,22,23,24,25,26,28)) then
    OTHERF[I]=OTHERF[I]+1;
  if (E38013[I]=1 and E38014[I] in (9,14)) then HRWG[I]=0;
  if E38013[I]=1 and -4<E38023[I]<0 then HRWG[I]=E38023[I];
/* if job was offering a comp at the end */
  if (E38106[I]=1 and E38107[I] in (7,0,12,13,14,15,16,17,99,21,22,23,24,25,26,28)) then
    OTHERF[I]=OTHERF[I]+1;
  if (E38106[I]=1 and E38107[I] in (9,14)) then HRWG[I]=0;
  if E38106[I]=1 and -4<E38116[I]<0 then HRWG[I]=E38116[I];
end;
end;

/* REPORT -1,-2 or -3 if THE AMOUNT REPORTED IS -1, -2 or -3*/
do I=1 to 9;
  if E37901B[I]=1 or E59900[I]=1 then do;
    if -4<E38023[I]<0 then HRWG[I]=E38023[I];
    if -4<E38116[I]<0 then HRWG[I]=E38116[I];
  end;
end;

/* The end wage if job lasts for >=13 weeks and report the same amount but diff. hours. NOTE: For this case, we
  only change hourly rate HRWG[I] without changing HRWAGE[I], DAILY[I], WEEKLW[I] BIWKLY[I],
  MONTH[I], ANNUAL[I] or SEMIM[I]*/
do I=1 to 9;
  if ((E37901B[I]=1 or E59900[I]=1) and E38013[I]=0 and E3800B[I]=0) and HRWG[I]>=0 and E34402[I]>0 and
    E3800F[I]>0 then HRWG[I]=HRWG[I]*E34402[I]/E3800F[I];
  if ((E37901B[I]=1 or E59900[I]=1) and E38013[I]=0 and E3800B[I]=0) and E3800F[I]=0 then HRWG[I]=-3;
  if ((E37901B[I]=1 or E59900[I]=1) and E38013[I]=0 and E3800B[I]=0) and -4<E3800F[I]<0 then
    HRWG[I]=E3800F[I];
  if ((E37901B[I]=1 or E59900[I]=1) and E38106[I]=0 and (E38102[I]=1 or E38102[I]=3)) and HRWG[I]>=0 and
    E34428[I]>0 and E38103[I]>0 then HRWG[I]=HRWG[I]*E34428[I]/E38103[I];
  if ((E37901B[I]=1 or E59900[I]=1) and E38106[I]=0 and (E38102[I]=1 or E38102[I]=3)) and HRWG[I]>=0 and
    E34402[I]>0 and E38103[I]>0 then HRWG[I]=HRWG[I]*E34402[I]/E38103[I];
  if ((E37901B[I]=1 or E59900[I]=1) and E38106[I]=0 and (E38102[I]=1 or E38102[I]=3)) and E38103[I]=0 then
    HRWG[I]=-3;
  if ((E37901B[I]=1 or E59900[I]=1) and E38106[I]=0 and (E38102[I]=1 or E38102[I]=3)) and -4<E38103[I]<0
    then HRWG[I]=E38103[I];
end;

**** Part 3: Add the end compensation ****/

**** With paid overtime ****/
do I=1 to 9;
if (E37901B[I]=1 or E59900[I]=1) then do;

if E38001[I]=0 then OT[I]=0; **** without compensation at the beginning ****/
```

Appendix 2: Employment Variable Creation

```
else do;
if E38002[I]>0 and E38003[I]=1 and E38012[I]>=0 then OT[I]=E38012[I];
if E38002[I]>0 and E38003[I]=2 and E38012[I]>=0 and E38012B[I]>0 then
    OT[I]=E38012[I]*E38012B[I]/E38002[I];
if E38002[I]>0 and E38003[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38012[I]>=0 then
    OT[I]=E38012[I]/E38002[I];
if E38002[I]>0 and E38003[I]=4 and E38012[I]>=0 then OT[I]=E38012[I]/(2*E38002[I]);
if E38002[I]>0 and E38003[I]=5 and E38012[I]>=0 then OT[I]=E38012[I]/(4.3*E38002[I]);
if E38002[I]>0 and E38003[I]=6 and E38012[I]>=0 then OT[I]=-3; /* Since no weeks per year available. */
if E38002[I]>0 and E38003[I]=8 and E38012[I]>=0 then OT[I]=E38012[I]/(2.15*E38002[I]);
if E38003[I] in (9,14) then OT[I]=0;
if E38002[I]>0 and E38003[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then OTF[I]=OTF[I]+1;
/*missing value*/
if -4<E38002[I]<0 then OT[I]=E38002[I];
if E38012[I]>=0 and E38002[I]=0 then OT[I]=-3;
if -4<E38012[I]<0 then OT[I]=E38012[I];
end;

if E38201[I]=1 then do; /*** with compensation at the beginning *****/
if E38101[I]=1 then do; /*same no. of hours as at the beginning */
if E19200[I]=1 then do; /* report hourly rate of pay at the beginning*/
if E24501[I]>0 and E38202[I]=1 and E38211[I]>=0 then OT[I]=E38211[I];
if E24501[I]>0 and E38202[I]=2 and E38211B[I]>0 and E38211[I]>=0 then
    OT[I]=E38211[I]*E38211B[I]/E24501[I];
if E24501[I]>0 and E38202[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38211[I]>=0 then
    OT[I]=E38211[I]/E24501[I];
if E24501[I]>0 and E38202[I]=4 and E38211[I]>=0 then OT[I]=E38211[I]/(2*E24501[I]);
if E24501[I]>0 and E38202[I]=5 and E38211[I]>=0 then OT[I]=E38211[I]/(4.3*E24501[I]);
if E24501[I]>0 and E38202[I]=6 and E38211[I]>=0 then OT[I]=-3; /* Since no weeks per year available. */
if E24501[I]>0 and E38202[I]=8 and E38211[I]>=0 then OT[I]=E38211[I]/(2.15*E24501[I]);
if E38202[I] in (9,14) then OT[I]=0;
if E24501[I]>0 and E38202[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then OTF[I]=OTF[I]+1;
/*missing value*/;
if -4<E24501[I]<0 then OT[I]=E24501[I];
if E38211[I]>=0 and E24501[I]=0 then OT[I]=-3;
if -4<E38211[I]<0 then OT[I]=E38211[I];
end;
end;

if E19200[I] ne 1 then do; /*report payment in other units at the beginning*/
if E34403[I]>0 and E38202[I]=1 and E38211[I]>=0 then OT[I]=E38211[I];
if E34403[I]>0 and E38202[I]=2 and E38211B[I]>0 and E38211[I]>=0 then
    OT[I]=E38211[I]*E38211B[I]/E34403[I];
if E34403[I]>0 and E38202[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38211[I]>=0 then
    OT[I]=E38211[I]/E34403[I];
if E34403[I]>0 and E38202[I]=4 and E38211[I]>=0 then OT[I]=E38211[I]/(2*E34403[I]);
if E34403[I]>0 and E38202[I]=5 and E38211[I]>=0 then OT[I]=E38211[I]/(4.3*E34403[I]);
if E34403[I]>0 and E38202[I]=6 and E38211[I]>=0 then OT[I]=-3; /* Since no weeks per year available. */
if E34403[I]>0 and E38202[I]=8 and E38211[I]>=0 then OT[I]=E38211[I]/(2.15*E34403[I]);
if E38202[I] in (9,14) then OT[I]=0;
if E34403[I]>0 and E38202[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then OTF[I]=OTF[I]+1;
/*missing value*/;
if -4<E34403[I]<0 then OT[I]=E34403[I];
if E38211[I]>=0 and E34403[I]=0 then OT[I]=-3;
if -4<E38211[I]<0 then OT[I]=E38211[I];
end;
end;
```

Appendix 2: Employment Variable Creation

```

else do; /* different no. of hours from the beginning*/
  if E38105[I]>0 and E38202[I]=1 and E38211[I]>=0 then OT[I]=E38211[I];
  if E38105[I]>0 and E38202[I]=2 and E38211B[I]>0 and E38211[I]>=0 then
    OT[I]=E38211[I]*E38211B[I]/E38105[I];
  if E38105[I]>0 and E38202[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38211[I]>=0 then
    OT[I]=E38211[I]/E38105[I];
  if E38105[I]>0 and E38202[I]=4 and E38211[I]>=0 then OT[I]=E38211[I]/(2*E38105[I]);
  if E38105[I]>0 and E38202[I]=5 and E38211[I]>=0 then OT[I]=E38211[I]/(4.3*E38105[I]);
  if E38105[I]>0 and E38202[I]=6 and E38211[I]>=0 then OT[I]=-3; /* Snice no weeks per year available. */
  if E38105[I]>0 and E38202[I]=8 and E38211[I]>=0 then OT[I]=E38211[I]/(2.15*E38105[I]);
  if E38202[I] in (9,14) then OT[I]=0;
  if E38105[I]>0 and E38202[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then OTF[I]=OTF[I]+1;
  /*missing value*/;
  if -4<E38105[I]<0 then OT[I]=E38105[I];
  if E38211[I]>=0 and E38105[I]=0 then OT[I]=-3;
  if -4<E38211[I]<0 then OT[I]=E38211[I];
end;
end;
end;
end;

/**** non-overtime payment ****/
do I=1 to 9;
  if (E37901B[I]=1 or E59900[I]=1) then do;

  /** case i. without overtime at the beginning, same no. of hours. ***/
  if E38313[I]=1 and E3800B[I]=1 and E212001[I] ne 1 then do;

  /* report hourly wage at the beginning*/
  /* for tips*/
  if E23901[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
  if E23901[I]>0 and E384071[I]=2 and E34402B[I]>0 and E384161[I]>=0 then
    OTHPAY1[I]=(E384161[I]*E34402B[I])/E23901[I];
  if E23901[I]>0 and E384071[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
    OTHPAY1[I]=E384161[I]/E23901[I];
  if E23901[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E23901[I]);
  if E23901[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E23901[I]);
  if E23901[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E384161[I]/(E35600[I]*E23901[I]);
  if E23901[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E23901[I]);
  if E23901[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF1[I]=OTHPF1[I]+1;
  if E384071[I] in (9,14) then OTHPAY1[I]=0;
  /*missing value*/;
  if E384161[I]>=0 and -4<E23901[I]<0 then OTHPAY1[I]=E23901[I];
  if E384161[I]>=0 and E23901[I]=0 then OTHPAY1[I]=-3;
  if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];
  if E384071[I]=2 and E34402B[I] le 0 then OTHPAY1[I]=-3;
  if E384071[I]=2 and -4<E34402B[I]<0 then OTHPAY1[I]=E34402B[I];
  if E384071[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
  if E384071[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];

  /*for commissions*/
  if E23901[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
  if E23901[I]>0 and E384072[I]=2 and E34402B[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E34402B[I])/E23901[I];

```


Appendix 2: Employment Variable Creation

```
if E23901[I]>0 and E384072[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
  OTHPAY2[I]=E384162[I]/E23901[I];
if E23901[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E23901[I]);
if E23901[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
  OTHPAY2[I]=E384162[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF2[I]=OTHPF2[I]+1;
if E384072[I] in (9,14) then OTHPAY2[I]=0;
*missing value*/;
if E384162[I]>=0 and -4<E23901[I]<0 then OTHPAY2[I]=E23901[I];
if E384162[I]>=0 and E23901[I]=0 then OTHPAY2[I]=-3;
if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];
if E384072[I]=2 and E34402B[I] le 0 then OTHPAY2[I]=-3;
if E384072[I]=2 and -4<E34402B[I]<0 then OTHPAY2[I]=E34402B[I];
if E384072[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E384072[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];

/*for bonuses*/
if E23901[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
if E23901[I]>0 and E384073[I]=2 and E34402B[I]>0 and E384163[I]>=0 then
  OTHPAY3[I]=(E384163[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E384073[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
  OTHPAY3[I]=E384163[I]/E23901[I];
if E23901[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E23901[I]);
if E23901[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E384163[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
if E384073[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E384163[I]>=0 and -4<E23901[I]<0 then OTHPAY3[I]=E23901[I];
if E384163[I]>=0 and E23901[I]=0 then OTHPAY3[I]=-3;
if -4<E384163[I]<0 then OTHPAY3[I]=E384163[I];
if E384073[I]=2 and E34402B[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=2 and -4<E34402B[I]<0 then OTHPAY3[I]=E34402B[I];
if E384073[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];

/*for incentive pay*/
if E23901[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
if E23901[I]>0 and E384074[I]=2 and E34402B[I]>0 and E384164[I]>=0 then
  OTHPAY4[I]=(E384164[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E384074[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
  OTHPAY4[I]=E384164[I]/E23901[I];
if E23901[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E23901[I]);
if E23901[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
  OTHPAY4[I]=E384164[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF4[I]=OTHPF4[I]+1;
if E384074[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
```

Appendix 2: Employment Variable Creation

```
if E384164[I]>=0 and -4<E23901[I]<0 then OTHPAY4[I]=E23901[I];
if E384164[I]>=0 and E23901[I]=0 then OTHPAY4[I]=-3;
if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];
if E384074[I]=2 and E34402B[I] le 0 then OTHPAY4[I]=-3;
if E384074[I]=2 and -4<E34402B[I]<0 then OTHPAY4[I]=E34402B[I];
if E384074[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E384074[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
```

*/*for others*/*

```
if E23901[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
if E23901[I]>0 and E384075[I]=2 and E34402B[I]>0 and E384165[I]>=0 then
    OTHPAY5[I]=(E384165[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E384075[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
    OTHPAY5[I]=E384165[I]/E23901[I];
if E23901[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E23901[I]);
if E23901[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E384165[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,00,21,22,23,24,25,26,28) then
    OTHPAY5[I]=OTHPAY5[I]+1;
if E384075[I] in (9,14) then OTHPAY5[I]=0;
/*missing value*/;
if E384165[I]>=0 and -4<E23901[I]<0 then OTHPAY5[I]=E23901[I];
if E384165[I]>=0 and E23901[I]=0 then OTHPAY5[I]=-3;
if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];
if E384075[I]=2 and E34402B[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=2 and -4<E34402B[I]<0 then OTHPAY5[I]=E34402B[I];
if E384075[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
```

/ report nonhourly wage at the beginning*/*

/ for tips*/*

```
if E34402[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
if E34402[I]>0 and E384071[I]=2 and E34402B[I]>0 and E384161[I]>=0 then
    OTHPAY1[I]=(E384161[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384071[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
    OTHPAY1[I]=E384161[I]/E34402[I];
if E34402[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E34402[I]);
if E34402[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E384161[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPAY1[I]=OTHPAY1[I]+1;
if E384071[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/;
if E384161[I]>=0 and -4<E34402[I]<0 then OTHPAY1[I]=E34402[I];
if E384161[I]>=0 and E34402[I]=0 then OTHPAY1[I]=-3;
if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];
```

*/*for commissions*/*

```
if E34402[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
if E34402[I]>0 and E384072[I]=2 and E34402B[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384072[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
    OTHPAY2[I]=E384162[I]/E34402[I];
```

Appendix 2: Employment Variable Creation

```
if E34402[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E34402[I]);
if E34402[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E384162[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
if E384072[I] in (9,14) then OTHPAY2[I]=0;
*missing value*/;
if E384162[I]>=0 and -4<E34402[I]<0 then OTHPAY2[I]=E34402[I];
if E384162[I]>=0 and E34402[I]=0 then OTHPAY2[I]=-3;
if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];

/*for bonuses*/
if E34402[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
if E34402[I]>0 and E384073[I]=2 and E34402B[I]>0 and E384163[I]>=0 then
    OTHPAY3[I]=(E384163[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384073[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
    OTHPAY3[I]=E384163[I]/E34402[I];
if E34402[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E34402[I]);
if E34402[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E384163[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF3[I]=OTHPF3[I]+1;
if E384073[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E384163[I]>=0 and -4<E34402[I]<0 then OTHPAY3[I]=E34402[I];
if E384163[I]>=0 and E34402[I]=0 then OTHPAY3[I]=-3;
if -4<E384163[I]<0 then OTHPAY3[I]=E384163[I];

/*for incentive pay*/
if E34402[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
if E34402[I]>0 and E384074[I]=2 and E34402B[I]>0 and E384164[I]>=0 then
    OTHPAY4[I]=(E384164[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384074[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
    OTHPAY4[I]=E384164[I]/E34402[I];
if E34402[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E34402[I]);
if E34402[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E384164[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF4[I]=OTHPF4[I]+1;
if E384074[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E384164[I]>=0 and -4<E34402[I]<0 then OTHPAY4[I]=E34402[I];
if E384164[I]>=0 and E34402[I]=0 then OTHPAY4[I]=-3;
if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];

/*for others*/
if E34402[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
if E34402[I]>0 and E384075[I]=2 and E34402B[I]>0 and E384165[I]>=0 then
    OTHPAY5[I]=(E384165[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384075[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
    OTHPAY5[I]=E384165[I]/E34402[I];
```

Appendix 2: Employment Variable Creation

```
if E34402[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E34402[I]);
if E34402[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E384165[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,00,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
if E384075[I] in (9,14) then OTHPAY5[I]=0;
*missing value*/;
if E384165[I]>=0 and -4<E34402[I]<0 then OTHPAY5[I]=E34402[I];
if E384165[I]>=0 and E34402[I]=0 then OTHPAY5[I]=-3;
if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];
end;

/** case ii. without overtime at the beginning, diff no. of hours. */
if E38313[I]=1 and E212001[I] ne 1 then do;

/* for tips*/
if E3800F[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
if E3800F[I]>0 and E384071[I]=2 and E38027[I]>0 and E384161[I]>=0 then
    OTHPAY1[I]=(E384161[I]*E38027[I])/E3800F[I];
if E3800F[I]>0 and E384071[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
    OTHPAY1[I]=E384161[I]/E3800F[I];
if E3800F[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E3800F[I]);
if E3800F[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E3800F[I]);
if E3800F[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E384161[I]/(E35600[I]*E3800F[I]);
if E3800F[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E3800F[I]);
if E3800F[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF1[I]=OTHPF1[I]+1;
if E384071[I] in (9,14) then OTHPAY1[I]=0;
*missing value*/;
if E384161[I]>=0 and -4<E3800F[I]<0 then OTHPAY1[I]=E3800F[I];
if E384161[I]>=0 and E3800F[I]=0 then OTHPAY1[I]=-3;
if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];
if E384071[I]=2 and E38027[I]=0 then OTHPAY1[I]=-3;
if E384071[I]=2 and -4<E38027[I]<0 then OTHPAY1[I]=E38027[I];
if E384071[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E384071[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];

/*for commissions*/
if E3800F[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
if E3800F[I]>0 and E384072[I]=2 and E38027[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E38027[I])/E3800F[I];
if E3800F[I]>0 and E384072[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
    OTHPAY2[I]=E384162[I]/E3800F[I];
if E3800F[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E3800F[I]);
if E3800F[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E3800F[I]);
if E3800F[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E384162[I]/(E35600[I]*E3800F[I]);
if E3800F[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E3800F[I]);
if E3800F[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
if E384072[I] in (9,14) then OTHPAY2[I]=0;
*missing value*/;
if E384162[I]>=0 and -4<E3800F[I]<0 then OTHPAY2[I]=E3800F[I];
if E384162[I]>=0 and E3800F[I]=0 then OTHPAY2[I]=-3;
```

Appendix 2: Employment Variable Creation

```
if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];
if E384072[I]=2 and E38027[I]=0 then OTHPAY2[I]=-3;
if E384072[I]=2 and -4<E38027[I]<0 then OTHPAY2[I]=E38027[I];
if E384072[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E384072[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];

/*for bonuses*/
if E3800F[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
if E3800F[I]>0 and E384073[I]=2 and E38027[I]>0 and E384163[I]>=0 then
    OTHPAY3[I]=(E384163[I]*E38027[I])/E3800F[I];
if E3800F[I]>0 and E384073[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
    OTHPAY3[I]=E384163[I]/E3800F[I];
if E3800F[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E3800F[I]);
if E3800F[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E3800F[I]);
if E3800F[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E384163[I]/(E35600[I]*E3800F[I]);
if E3800F[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E3800F[I]);
if E3800F[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF3[I]=OTHPF3[I]+1;
if E384073[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E384163[I]>=0 and -4<E3800F[I]<0 then OTHPAY3[I]=E3800F[I];
if E384163[I]>=0 and E3800F[I]=0 then OTHPAY3[I]=-3;
if -4<E384163[I]<0 then OTHPAY1[I]=E384163[I];
if E384073[I]=2 and E38027[I]=0 then OTHPAY3[I]=-3;
if E384073[I]=2 and -4<E38027[I]<0 then OTHPAY3[I]=E38027[I];
if E384073[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];

/*for incentive pay*/
if E3800F[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
if E3800F[I]>0 and E384074[I]=2 and E38027[I]>0 and E384164[I]>=0 then
    OTHPAY4[I]=(E384164[I]*E38027[I])/E3800F[I];
if E3800F[I]>0 and E384074[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
    OTHPAY4[I]=E384164[I]/E3800F[I];
if E3800F[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E3800F[I]);
if E3800F[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E3800F[I]);
if E3800F[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E384164[I]/(E35600[I]*E3800F[I]);
if E3800F[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E3800F[I]);
if E3800F[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF4[I]=OTHPF4[I]+1;
if E384074[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E384164[I]>=0 and -4<E3800F[I]<0 then OTHPAY4[I]=E3800F[I];
if E384164[I]>=0 and E3800F[I]=0 then OTHPAY4[I]=-3;
if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];
if E384074[I]=2 and E38027[I]=0 then OTHPAY4[I]=-3;
if E384074[I]=2 and -4<E38027[I]<0 then OTHPAY4[I]=E38027[I];
if E384074[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E384074[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];

/*for others*/
if E3800F[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
if E3800F[I]>0 and E384075[I]=2 and E38027[I]>0 and E384165[I]>=0 then
    OTHPAY5[I]=(E384165[I]*E38027[I])/E3800F[I];
```

Appendix 2: Employment Variable Creation

```
if E3800F[I]>0 and E384075[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
  OTHPAY5[I]=E384165[I]/E3800F[I];
if E3800F[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E3800F[I]);
if E3800F[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E3800F[I]);
if E3800F[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
  OTHPAY5[I]=E384165[I]/(E35600[I]*E3800F[I]);
if E3800F[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E3800F[I]);
if E3800F[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF5[I]=OTHPF5[I]+1;
if E384075[I] in (9,14) then OTHPAY5[I]=0;
*missing value*/;
if E384165[I]>=0 and -4<E3800F[I]<0 then OTHPAY5[I]=E3800F[I];
if E384165[I]>=0 and E3800F[I]=0 then OTHPAY5[I]=-3;
if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];
if E384075[I]=2 and E38027[I]=0 then OTHPAY5[I]=-3;
if E384075[I]=2 and -4<E38027[I]<0 then OTHPAY5[I]=E38027[I];
if E384075[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;

/** case iii. only one compensation at the beginning,same no. of hours. **/
if E38329[I]=1 and E20700[I]=1 then do;

/* for tips*/
if E212002[I]=1 then do;
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
/*report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
  OTHPAY1[I]=(E38329D[I]*E34430[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY1[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY1[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY1[I]=-3;
if -4<E38329D[I]<0 then OTHPAY1[I]=E38329D[I];

/*report nonhourly wage*/
if E34428[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I];
if E34428[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
  OTHPAY1[I]=(E38329D[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY1[I]=E38329D[I]/E34428[I];
if E34428[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2*E34428[I]);
if E34428[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(4.3*E34428[I]);
if E34428[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E38329D[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2.15*E34428[I]);
if E34428[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
```

Appendix 2: Employment Variable Creation

```
/*missing value*/;
if E38329D[I]>=0 and -4<E34428[I]<0 then OTHPAY1[I]=E34428[I];
if E38329D[I]>=0 and E34428[I]=0 then OTHPAY1[I]=-3;
if -4<E38329D[I]<0 then OTHPAY1[I]=E38329D[I];
if E38329B[I]=2 and E34430[I] le 0 then OTHPAY1[I]=-3;
if E38329B[I]=2 and -4<E34430[I]<0 then OTHPAY1[I]=E34430[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/*report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
    OTHPAY1[I]=(E38329D[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY1[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY1[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY1[I]=-3;
if -4<E38329D[I]<0 then OTHPAY1[I]=E38329D[I];

/*report non-hourly wage at the beginning*/
if E34402[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I];
if E34402[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
    OTHPAY1[I]=(E38329D[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY1[I]=E38329D[I]/E34402[I];
if E34402[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2*E34402[I]);
if E34402[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(4.3*E34402[I]);
if E34402[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E38329D[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2.15*E34402[I]);
if E34402[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF1[I]=OTHPF1[I]+1;
if E38329B[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/;
if E38329D[I]>=0 and -4<E34402[I]<0 then OTHPAY1[I]=E34402[I];
if E38329D[I]>=0 and E34402[I]=0 then OTHPAY1[I]=-3;
if E38329B[I]=2 and E34402B[I] le 0 then OTHPAY1[I]=-3;
if E38329B[I]=2 and -4<E34402B[I]<0 then OTHPAY1[I]=E34402B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
end;
end;

/*for commissions*/
if E212003[I]=1 then do;
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
/*report hourly wage at the beginning*/
```

Appendix 2: Employment Variable Creation

```
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
  OTHPAY2[I]=(E38329D[I]*E34430[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY2[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY2[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY2[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY2[I]=-3;
if -4<E38329D[I]<0 then OTHPAY2[I]=E38329D[I];

/*report nonhourly wage*/
if E34428[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I];
if E34428[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
  OTHPAY2[I]=(E38329D[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY2[I]=E38329D[I]/E34428[I];
if E34428[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2*E34428[I]);
if E34428[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(4.3*E34428[I]);
if E34428[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY2[I]=E38329D[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2.15*E34428[I]);
if E34428[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF2[I]=OTHPF2[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E34428[I]<0 then OTHPAY2[I]=E34428[I];
if E38329D[I]>=0 and E34428[I]=0 then OTHPAY2[I]=-3;
if -4<E38329D[I]<0 then OTHPAY2[I]=E38329D[I];
if E38329B[I]=2 and E34430[I] le 0 then OTHPAY2[I]=-3;
if E38329B[I]=2 and -4<E34430[I]<0 then OTHPAY2[I]=E34430[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/*report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
  OTHPAY2[I]=(E38329D[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY2[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY2[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY2[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY2[I]=-3;
```


Appendix 2: Employment Variable Creation

```
if -4<E38329D[I]<0 then OTHPAY2[I]=E38329D[I];

/*report nonhourly wage*/
if E34402[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I];
if E34402[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
    OTHPAY2[I]=(E38329D[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY2[I]=E38329D[I]/E34402[I];
if E34402[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2*E34402[I]);
if E34402[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(4.3*E34402[I]);
if E34402[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E38329D[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2.15*E34402[I]);
if E34402[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
if E38329B[I] in (9,14) then OTHPAY2[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E34402[I]<0 then OTHPAY2[I]=E34402[I];
if E38329D[I]>=0 and E34402[I]=0 then OTHPAY2[I]=-3;
if E38329B[I]=2 and E34402B[I] le 0 then OTHPAY2[I]=-3;
if E38329B[I]=2 and -4<E34402B[I]<0 then OTHPAY2[I]=E34402B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];
end;
end;

/*for bonuses*/
if E212004[I]=1 then do;
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
/* report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
    OTHPAY3[I]=(E38329D[I]*E34430[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY3[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF3[I]=OTHPF3[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY3[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY3[I]=-3;
if -4<E38329D[I]<0 then OTHPAY3[I]=E38329D[I];

/*report non-hourly wage*/
if E34428[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I];
if E34428[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
    OTHPAY3[I]=(E38329D[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY3[I]=E38329D[I]/E34428[I];
if E34428[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2*E34428[I]);
if E34428[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(4.3*E34428[I]);
if E34428[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E38329D[I]/(E35600[I]*E34428[I]);
```

Appendix 2: Employment Variable Creation

```
if E34428[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2.15*E34428[I]);
if E34428[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E34428[I]<0 then OTHPAY3[I]=E34428[I];
if E38329D[I]>=0 and E34428[I]=0 then OTHPAY3[I]=-3;
if -4<E38329D[I]<0 then OTHPAY3[I]=E38329D[I];
if E38329B[I]=2 and E34430[I] le 0 then OTHPAY3[I]=-3;
if E38329B[I]=2 and -4<E34430[I]<0 then OTHPAY3[I]=E34430[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/* report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
  OTHPAY3[I]=(E38329D[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY3[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY3[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY3[I]=-3;
if -4<E38329D[I]<0 then OTHPAY3[I]=E38329D[I];

/*report non-hourly wage*/
if E34402[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I];
if E34402[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
  OTHPAY3[I]=(E38329D[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY3[I]=E38329D[I]/E34402[I];
if E34402[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2*E34402[I]);
if E34402[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(4.3*E34402[I]);
if E34402[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E38329D[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2.15*E34402[I]);
if E34402[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
if E38329B[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E34402[I]<0 then OTHPAY3[I]=E34402[I];
if E38329D[I]>=0 and E34402[I]=0 then OTHPAY3[I]=-3;
if E38329B[I]=2 and E34402B[I] le 0 then OTHPAY3[I]=-3;
if E38329B[I]=2 and -4<E34402B[I]<0 then OTHPAY3[I]=E34402B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;
end;

/*for incentive pay*/
```

Appendix 2: Employment Variable Creation

```
if E212005[I]=1 then do;
  if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
    /* report hourly wage at the beginning*/
    if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I];
    if E23901[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
      OTHPAY4[I]=(E38329D[I]*E34430[I])/E23901[I];
    if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
      OTHPAY4[I]=E38329D[I]/E23901[I];
    if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2*E23901[I]);
    if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(4.3*E23901[I]);
    if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
      OTHPAY4[I]=E38329D[I]/(E35600[I]*E23901[I]);
    if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2.15*E23901[I]);
    if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
      OTHPF4[I]=OTHPF4[I]+1;
    *missing value*/;
    if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY4[I]=E23901[I];
    if E38329D[I]>=0 and E23901[I]=0 then OTHPAY4[I]=-3;
    if -4<E38329D[I]<0 then OTHPAY4[I]=E38329D[I];

    /* report nonhourly wage*/
    if E34428[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I];
    if E34428[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
      OTHPAY4[I]=(E38329D[I]*E34430[I])/E34428[I];
    if E34428[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
      OTHPAY4[I]=E38329D[I]/E34428[I];
    if E34428[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2*E34428[I]);
    if E34428[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(4.3*E34428[I]);
    if E34428[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
      OTHPAY4[I]=E38329D[I]/(E35600[I]*E34428[I]);
    if E34428[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2.15*E34428[I]);
    if E34428[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
      OTHPF4[I]=OTHPF4[I]+1;
    *missing value*/;
    if E38329D[I]>=0 and -4<E34428[I]<0 then OTHPAY4[I]=E34428[I];
    if E38329D[I]>=0 and E34428[I]=0 then OTHPAY4[I]=-3;
    if -4<E38329D[I]<0 then OTHPAY4[I]=E38329D[I];
    if E38329B[I]=2 and E34430[I] le 0 then OTHPAY4[I]=-3;
    if E38329B[I]=2 and -4<E34430[I]<0 then OTHPAY4[I]=E34430[I];
    if E38329B[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
    if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
  end;

  if E3800B[I]=1 then do; /*without overtime at the beginning*/
    /* report hourly wage at the beginning*/
    if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I];
    if E23901[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
      OTHPAY4[I]=(E38329D[I]*E34402B[I])/E23901[I];
    if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
      OTHPAY4[I]=E38329D[I]/E23901[I];
    if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2*E23901[I]);
    if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(4.3*E23901[I]);
    if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
      OTHPAY4[I]=E38329D[I]/(E35600[I]*E23901[I]);
    if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2.15*E23901[I]);
    if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
      OTHPF4[I]=OTHPF4[I]+1;
```

Appendix 2: Employment Variable Creation

```
*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY4[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY4[I]=-3;
if -4<E38329D[I]<0 then OTHPAY4[I]=E38329D[I];

/* report nonhourly wage*/
if E34402[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I];
if E34402[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
    OTHPAY4[I]=(E38329D[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY4[I]=E38329D[I]/E34402[I];
if E34402[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2*E34402[I]);
if E34402[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(4.3*E34402[I]);
if E34402[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E38329D[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2.15*E34402[I]);
if E34402[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPAY4[I]=OTHPAY4[I]+1;
if E38329B[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E34402[I]<0 then OTHPAY4[I]=E34402[I];
if E38329D[I]>=0 and E34402[I]=0 then OTHPAY4[I]=-3;
if E38329B[I]=2 and E34402B[I] le 0 then OTHPAY4[I]=-3;
if E38329B[I]=2 and -4<E34402B[I]<0 then OTHPAY4[I]=E34402B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
end;
end;

/*for others*/
if E212006[I]=1 then do;
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
/* report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
    OTHPAY5[I]=(E38329D[I]*E34430[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY5[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPAY5[I]=OTHPAY5[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY5[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY5[I]=-3;
if -4<E38329D[I]<0 then OTHPAY5[I]=E38329D[I];

/*report non-hourly wage*/
if E34428[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I];
if E34428[I]>0 and E38329B[I]=2 and E34430[I]>0 and E38329D[I]>=0 then
    OTHPAY5[I]=(E38329D[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY5[I]=E38329D[I]/E34428[I];
if E34428[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2*E34428[I]);
```

Appendix 2: Employment Variable Creation

```

if E34428[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(4.3*E34428[I]);
if E34428[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E38329D[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2.15*E34428[I]);
if E34428[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E34428[I]<0 then OTHPAY5[I]=E34428[I];
if E38329D[I]>=0 and E34428[I]=0 then OTHPAY5[I]=-3;
if -4<E38329D[I]<0 then OTHPAY5[I]=E38329D[I];
if E38329B[I]=2 and E34430[I] le 0 then OTHPAY5[I]=-3;
if E38329B[I]=2 and -4<E34430[I]<0 then OTHPAY5[I]=E34430[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/* report hourly wage at the beginning*/
if E23901[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I];
if E23901[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
    OTHPAY5[I]=(E38329D[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY5[I]=E38329D[I]/E23901[I];
if E23901[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2*E23901[I]);
if E23901[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(4.3*E23901[I]);
if E23901[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E38329D[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2.15*E23901[I]);
if E23901[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
*missing value*/;
if E38329D[I]>=0 and -4<E23901[I]<0 then OTHPAY5[I]=E23901[I];
if E38329D[I]>=0 and E23901[I]=0 then OTHPAY5[I]=-3;
if -4<E38329D[I]<0 then OTHPAY5[I]=E38329D[I];

/*report non-hourly wage*/
if E34402[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I];
if E34402[I]>0 and E38329B[I]=2 and E34402B[I]>0 and E38329D[I]>=0 then
    OTHPAY5[I]=(E38329D[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY5[I]=E38329D[I]/E34402[I];
if E34402[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2*E34402[I]);
if E34402[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(4.3*E34402[I]);
if E34402[I]>0 and E38329B[I]=6 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(52*E34402[I]);
if E34402[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2.15*E34402[I]);
if E34402[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
if E38329B[I] in (9,14) then OTHPAY5[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E34402[I]<0 then OTHPAY5[I]=E34402[I];
if E38329D[I]>=0 and E34402[I]=0 then OTHPAY5[I]=-3;
if E38329B[I]=2 and E34402B[I] le 0 then OTHPAY5[I]=-3;
if E38329B[I]=2 and -4<E34402B[I]<0 then OTHPAY5[I]=E34402B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;
end;

```

Appendix 2: Employment Variable Creation

```
end;

/** case iv. one compensation at the beginning, diff no. of hours. ***/
if E38329[I]=1 and E20700[I]=1 then do;

/* for tips*/
if E212002[I]=1 then do;
  if E3800B[I]=0 then do; /* without overtime at the beginning*/
    if E3800F[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I];
    if E3800F[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
      OTHPAY1[I]=(E38329D[I]*E38116B[I])/E3800F[I];
    if E3800F[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
      OTHPAY1[I]=E38329D[I]/E3800F[I];
    if E3800F[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2*E3800F[I]);
    if E3800F[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(4.3*E3800F[I]);
    if E3800F[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
      OTHPAY1[I]=E38329D[I]/(E35600[I]*E3800F[I]);
    if E3800F[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2.15*E3800F[I]);
    if E3800F[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,18,99,21,22,23,24,25,26,28) then
      OTHPF1[I]=OTHPF1[I]+1;
    if E38329B[I] in (9,14) then OTHPAY1[I]=0;
    /*missing value*/;
    if E38329D[I]>=0 and -4<E3800F[I]<0 then OTHPAY1[I]=E3800F[I];
    if E38329D[I]>=0 and E3800F[I]=0 then OTHPAY1[I]=-3;
    if -4<E38329D[I]<0 then OTHPAY1[I]=E38329D[I];
    if E38329B[I]=2 and E38116B[I]=0 then OTHPAY1[I]=-3;
    if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY1[I]=E38116B[I];
    if E38329B[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
    if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
  end;

  if E38102[I]=1 or E38102[I]=3 then do; /* with overtime at the beginning*/
    if E38103[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I];
    if E38103[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
      OTHPAY1[I]=(E38329D[I]*E38116B[I])/E38103[I];
    if E38103[I]>0 and E38329B[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
      OTHPAY1[I]=E38329D[I]/E38103[I];
    if E38103[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2*E38103[I]);
    if E38103[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(4.3*E38103[I]);
    if E38103[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
      OTHPAY1[I]=E38329D[I]/(E35600[I]*E38103[I]);
    if E38103[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY1[I]=E38329D[I]/(2.15*E38103[I]);
    if E38103[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,18,99,21,22,23,24,25,26,28) then
      OTHPF1[I]=OTHPF1[I]+1;
    if E38329B[I] in (9,14) then OTHPAY1[I]=0;
    /*missing value*/;
    if E38329D[I]>=0 and -4<E38103[I]<0 then OTHPAY1[I]=E38103[I];
    if E38329D[I]>=0 and E38103[I]=0 then OTHPAY1[I]=-3;
    if -4<E38329D[I]<0 then OTHPAY1[I]=E38329D[I];
    if E38329B[I]=2 and E38116B[I]=0 then OTHPAY1[I]=-3;
    if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY1[I]=E38116B[I];
    if E38329B[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
    if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
  end;
end;

/*for commissions*/
```

Appendix 2: Employment Variable Creation

```
if E212003[I]=1 then do;
  /*without overtime at the beginning*/
  if E3800F[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I];
  if E3800F[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
    OTHPAY2[I]=(E38329D[I]*E38116B[I])/E3800F[I];
  if E3800F[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY2[I]=E38329D[I]/E3800F[I];
  if E3800F[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(4.3*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E38329D[I]/(E35600[I]*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2.15*E3800F[I]);
  if E3800F[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
  if E38329B[I] in (9,14) then OTHPAY2[I]=0;
  *missing value*/;
  if E38329D[I]>=0 and -4<E3800F[I]<0 then OTHPAY2[I]=E3800F[I];
  if E38329D[I]>=0 and E3800F[I]=0 then OTHPAY2[I]=-3;
  if -4<E38329D[I]<0 then OTHPAY2[I]=E38329D[I];
  if E38329B[I]=2 and E38116B[I]=0 then OTHPAY2[I]=-3;
  if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY2[I]=E38116B[I];
  if E38329B[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
  if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];

  /* with overtime at the beginning*/
  if E38102[I]=1 or E38102[I]=3 then do;
  if E38103[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I];
  if E38103[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
    OTHPAY2[I]=(E38329D[I]*E38116B[I])/E38103[I];
  if E38103[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY2[I]=E38329D[I]/E38103[I];
  if E38103[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2*E38103[I]);
  if E38103[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(4.3*E38103[I]);
  if E38103[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E38329D[I]/(E35600[I]*E38103[I]);
  if E38103[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY2[I]=E38329D[I]/(2.15*E38103[I]);
  if E38103[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
  if E38329B[I] in (9,14) then OTHPAY2[I]=0;
  *missing value*/;
  if E38329D[I]>=0 and -4<E38103[I]<0 then OTHPAY2[I]=E38103[I];
  if E38329D[I]>=0 and E38103[I]=0 then OTHPAY2[I]=-3;
  if -4<E38329D[I]<0 then OTHPAY2[I]=E38329D[I];
  if E38329B[I]=2 and E38116B[I]=0 then OTHPAY2[I]=-3;
  if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY2[I]=E38116B[I];
  if E38329B[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
  if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];
  end;
end;

/*for bonuses*/
if E212004[I]=1 then do;
  /* without overtime at the beginning*/
  if E3800F[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I];
  if E3800F[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
    OTHPAY3[I]=(E38329D[I]*E38116B[I])/E3800F[I];
```

Appendix 2: Employment Variable Creation

```

if E3800F[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY3[I]=E38329D[I]/E3800F[I];
if E3800F[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2*E3800F[I]);
if E3800F[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(4.3*E3800F[I]);
if E3800F[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E38329D[I]/(E35600[I]*E3800F[I]);
if E3800F[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2.15*E3800F[I]);
if E3800F[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
if E38329B[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E3800F[I]<0 then OTHPAY3[I]=E3800F[I];
if E38329D[I]>=0 and E3800F[I]=0 then OTHPAY3[I]=-3;
if -4<E38329D[I]<0 then OTHPAY3[I]=E38329D[I];
if E38329B[I]=2 and E38116B[I]=0 then OTHPAY3[I]=-3;
if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY3[I]=E38116B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];

if E38102[I]=1 or E38102[I]=3 then do; /*with overtime at the beginning*/
if E38103[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I];
if E38103[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
  OTHPAY3[I]=(E38329D[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY3[I]=E38329D[I]/E38103[I];
if E38103[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2*E38103[I]);
if E38103[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(4.3*E38103[I]);
if E38103[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E38329D[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY3[I]=E38329D[I]/(2.15*E38103[I]);
if E38103[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
if E38329B[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E38103[I]<0 then OTHPAY3[I]=E38103[I];
if E38329D[I]>=0 and E38103[I]=0 then OTHPAY3[I]=-3;
if -4<E38329D[I]<0 then OTHPAY3[I]=E38329D[I];
if E38329B[I]=2 and E38116B[I]=0 then OTHPAY3[I]=-3;
if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY3[I]=E38116B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;
end;

/*for incentive pay*/
if E212005[I]=1 then do;
  /*without overtime at the beginning*/
  if E3800F[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I];
  if E3800F[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
    OTHPAY4[I]=(E38329D[I]*E38116B[I])/E3800F[I];
  if E3800F[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY4[I]=E38329D[I]/E3800F[I];
  if E3800F[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(4.3*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E38329D[I]/(E35600[I]*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2.15*E3800F[I]);

```


Appendix 2: Employment Variable Creation

```
if E3800F[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF4[I]=OTHPF4[I]+1;
if E38329B[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E38329D[I]>0 and -4<E3800F[I]<0 then OTHPAY4[I]=E3800F[I];
if E38329D[I]>=0 and E3800F[I]=0 then OTHPAY4[I]=-3;
if -4<E38329D[I]<0 then OTHPAY4[I]=E38329D[I];
if E38329B[I]=2 and E38116B[I]=0 then OTHPAY4[I]=-3;
if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY4[I]=E38116B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];

if E38102[I]=1 or E38102[I]=3 then do; /* with overtime at the beginning*/
if E38103[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I];
if E38103[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
  OTHPAY4[I]=(E38329D[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
  OTHPAY4[I]=E38329D[I]/E38103[I];
if E38103[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2*E38103[I]);
if E38103[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(4.3*E38103[I]);
if E38103[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
  OTHPAY4[I]=E38329D[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY4[I]=E38329D[I]/(2.15*E38103[I]);
if E38103[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF4[I]=OTHPF4[I]+1;
if E38329B[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E38329D[I]>0 and -4<E38103[I]<0 then OTHPAY4[I]=E38103[I];
if E38329D[I]>=0 and E38103[I]=0 then OTHPAY4[I]=-3;
if -4<E38329D[I]<0 then OTHPAY4[I]=E38329D[I];
if E38329B[I]=2 and E38116B[I]=0 then OTHPAY4[I]=-3;
if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY4[I]=E38116B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
end;
end;

/*for others*/
if E212006[I]=1 then do;
  /* without overtime at the beginning*/
  if E3800F[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I];
  if E3800F[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
    OTHPAY5[I]=(E38329D[I]*E38116B[I])/E3800F[I];
  if E3800F[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY5[I]=E38329D[I]/E3800F[I];
  if E3800F[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(4.3*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E38329D[I]/(E35600[I]*E3800F[I]);
  if E3800F[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2.15*E3800F[I]);
  if E3800F[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
  if E38329B[I] in (9,14) then OTHPAY5[I]=0;
  *missing value*/;
  if E38329D[I]>=0 and -4<E3800F[I]<0 then OTHPAY5[I]=E3800F[I];
  if E38329D[I]>=0 and E3800F[I]=0 then OTHPAY5[I]=-3;
  if -4<E38329D[I]<0 then OTHPAY5[I]=E38329D[I];
```

Appendix 2: Employment Variable Creation

```
if E38329B[I]=2 and E38116B[I]=0 then OTHPAY5[I]=-3;
if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY5[I]=E38116B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];

if E38102[I]=1 or E38102[I]=3 then do; /* with overtime at the beginning*/
if E38103[I]>0 and E38329B[I]=1 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I];
if E38103[I]>0 and E38329B[I]=2 and E38116B[I]>0 and E38329D[I]>=0 then
    OTHPAY5[I]=(E38329D[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E38329B[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E38329D[I]>=0 then
    OTHPAY5[I]=E38329D[I]/E38103[I];
if E38103[I]>0 and E38329B[I]=4 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2*E38103[I]);
if E38103[I]>0 and E38329B[I]=5 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(4.3*E38103[I]);
if E38103[I]>0 and E38329B[I]=6 and E38329D[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E38329D[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E38329B[I]=8 and E38329D[I]>=0 then OTHPAY5[I]=E38329D[I]/(2.15*E38103[I]);
if E38103[I]>0 and E38329B[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
if E38329B[I] in (9,14) then OTHPAY5[I]=0;
*missing value*/;
if E38329D[I]>=0 and -4<E38103[I]<0 then OTHPAY5[I]=E38103[I];
if E38329D[I]>=0 and E38103[I]=0 then OTHPAY5[I]=-3;
if -4<E38329D[I]<0 then OTHPAY5[I]=E38329D[I];
if E38329B[I]=2 and E38116B[I]=0 then OTHPAY5[I]=-3;
if E38329B[I]=2 and -4<E38116B[I]<0 then OTHPAY5[I]=E38116B[I];
if E38329B[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E38329B[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;
end;

end;

/** case v. more then one compensation at the beginning, same no. of hours. ***/
if E38330[I]=1 and E20700[I]=1 then do;

/* for tips*/
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
/*report hourly wage at the beginning*/
if E23901[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
if E23901[I]>0 and E384071[I]=2 and E34430[I]>0 and E384161[I]>=0 then
    OTHPAY1[I]=(E384161[I]*E34430[I])/E23901[I];
if E23901[I]>0 and E384071[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
    OTHPAY1[I]=E384161[I]/E23901[I];
if E23901[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E23901[I]);
if E23901[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E384161[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E384161[I]>=0 and -4<E23901[I]<0 then OTHPAY1[I]=E23901[I];
if E384161[I]>=0 and E23901[I]=0 then OTHPAY1[I]=-3;
if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];

/*report non-hourly wage*/
if E34428[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
```

Appendix 2: Employment Variable Creation

```
if E34428[I]>0 and E384071[I]=2 and E34430[I]>0 and E384161[I]>=0 then
  OTHPAY1[I]=(E384161[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E384071[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
  OTHPAY1[I]=E384161[I]/E34428[I];
if E34428[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E34428[I]);
if E34428[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E34428[I]);
if E34428[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E384161[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E34428[I]);
if E34428[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E384161[I]>=0 and -4<E34428[I]<0 then OTHPAY1[I]=E34428[I];
if E384161[I]>=0 and E34428[I]=0 then OTHPAY1[I]=-3;
if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];
if E384071[I]=2 and E34430[I] le 0 then OTHPAY1[I]=-3;
if E384071[I]=2 and -4<E34430[I]<0 THNE OTHPAY1[I]=E34430[I];
if E384071[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E384071[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/*report hourly wage at the beginning*/
if E23901[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
if E23901[I]>0 and E384071[I]=2 and E34402B[I]>0 and E384161[I]>=0 then
  OTHPAY1[I]=(E384161[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E384071[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
  OTHPAY1[I]=E384161[I]/E23901[I];
if E23901[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E23901[I]);
if E23901[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E384161[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
/*missing value*/;
if E384161[I]>=0 and -4<E23901[I]<0 then OTHPAY1[I]=E23901[I];
if E384161[I]>=0 and E23901[I]=0 then OTHPAY1[I]=-3;
if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];

/*report non-hourly wage*/
if E34402[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
if E34402[I]>0 and E384071[I]=2 and E34402B[I]>0 and E384161[I]>=0 then
  OTHPAY1[I]=(E384161[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384071[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
  OTHPAY1[I]=E384161[I]/E34402[I];
if E34402[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E34402[I]);
if E34402[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
  OTHPAY1[I]=E384161[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF1[I]=OTHPF1[I]+1;
if E384071[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/;
if E384161[I]>=0 and -4<E34402[I]<0 then OTHPAY1[I]=E34402[I];
if E384161[I]>=0 and E34402[I]=0 then OTHPAY1[I]=-3;
```

Appendix 2: Employment Variable Creation

```
if E384071[I]=2 and E34402B[I] le 0 then OTHPAY1[I]=-3;
if E384071[I]=2 and -4<E34402B[I]<0 THNE OTHPAY1[I]=E34402B[I];
if E384071[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E384071[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];
end;

/*for commissions*/
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
  /* report hourly wage at the beginning*/
  if E23901[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
  if E23901[I]>0 and E384072[I]=2 and E34430[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E34430[I])/E23901[I];
  if E23901[I]>0 and E384072[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
    OTHPAY2[I]=E384162[I]/E23901[I];
  if E23901[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E23901[I]);
  if E23901[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E23901[I]);
  if E23901[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E384162[I]/(E35600[I]*E23901[I]);
  if E23901[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E23901[I]);
  if E23901[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
  *missing value*/;
  if E384162[I]>=0 and -4<E23901[I]<0 then OTHPAY2[I]=E23901[I];
  if E384162[I]>=0 and E23901[I]=0 then OTHPAY2[I]=-3;
  if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];

  /* report non-hourly wage*/
  if E34428[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
  if E34428[I]>0 and E384072[I]=2 and E34430[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E34430[I])/E34428[I];
  if E34428[I]>0 and E384072[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
    OTHPAY2[I]=E384162[I]/E34428[I];
  if E34428[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E34428[I]);
  if E34428[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E34428[I]);
  if E34428[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E384162[I]/(E35600[I]*E34428[I]);
  if E34428[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E34428[I]);
  if E34428[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
  *missing value*/;
  if E384162[I]>=0 and -4<E34428[I]<0 then OTHPAY2[I]=E34428[I];
  if E384162[I]>=0 and E34428[I]=0 then OTHPAY2[I]=-3;
  if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];
  if E384072[I]=2 and E34430[I] le 0 then OTHPAY2[I]=-3;
  if E384072[I]=2 and -4<E34430[I]<0 THNE OTHPAY2[I]=E34430[I];
  if E384072[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
  if E384072[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
  /* report hourly wage at the beginning*/
  if E23901[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
  if E23901[I]>0 and E384072[I]=2 and E34402B[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E34402B[I])/E23901[I];
  if E23901[I]>0 and E384072[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
    OTHPAY2[I]=E384162[I]/E23901[I];
  if E23901[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E23901[I]);
```

Appendix 2: Employment Variable Creation

```
if E23901[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
  OTHPAY2[I]=E384162[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF2[I]=OTHPF2[I]+1;
*missing value*/;
if E384162[I]>=0 and -4<E23901[I]<0 then OTHPAY2[I]=E23901[I];
if E384162[I]>=0 and E23901[I]=0 then OTHPAY2[I]=-3;
if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];

/* report non-hourly wage*/
if E34402[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
if E34402[I]>0 and E384072[I]=2 and E34402B[I]>0 and E384162[I]>=0 then
  OTHPAY2[I]=(E384162[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384072[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
  OTHPAY2[I]=E384162[I]/E34402[I];
if E34402[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E34402[I]);
if E34402[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
  OTHPAY2[I]=E384162[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF2[I]=OTHPF2[I]+1;
if E384072[I] in (9,14) then OTHPAY2[I]=0;
*missing value*/;
if E384162[I]>=0 and -4<E34402[I]<0 then OTHPAY2[I]=E34402[I];
if E384162[I]>=0 and E34402[I]=0 then OTHPAY2[I]=-3;
if E384072[I]=2 and E34402B[I] le 0 then OTHPAY2[I]=-3;
if E384072[I]=2 and -4<E34402B[I]<0 then OTHPAY2[I]=E34402B[I];
if E384072[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E384072[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];
end;

/*for bonuses*/
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
  /* report hourly wage at the beginning*/
  if E23901[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
  if E23901[I]>0 and E384073[I]=2 and E34430[I]>0 and E384163[I]>=0 then
    OTHPAY3[I]=(E384163[I]*E34430[I])/E23901[I];
  if E23901[I]>0 and E384073[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
    OTHPAY3[I]=E384163[I]/E23901[I];
  if E23901[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E23901[I]);
  if E23901[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E23901[I]);
  if E23901[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
    OTHPAY3[I]=E384163[I]/(E35600[I]*E23901[I]);
  if E23901[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E23901[I]);
  if E23901[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF3[I]=OTHPF3[I]+1;
  *missing value*/;
  if E384163[I]>=0 and -4<E23901[I]<0 then OTHPAY3[I]=E23901[I];
  if E384163[I]>=0 and E23901[I]=0 then OTHPAY3[I]=-3;
  if -4<E384163[I]<0 then OTHPAY3[I]=E384163[I];

  /*report non-hourly wage*/
  if E34428[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
```

Appendix 2: Employment Variable Creation

```
if E34428[I]>0 and E384073[I]=2 and E34430[I]>0 and E384163[I]>=0 then
  OTHPAY3[I]=(E384163[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E384073[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
  OTHPAY3[I]=E384163[I]/E34428[I];
if E34428[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E34428[I]);
if E34428[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E34428[I]);
if E34428[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E384163[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E34428[I]);
if E34428[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
*missing value*/;
if E384163[I]>=0 and -4<E34428[I]<0 then OTHPAY3[I]=E34428[I];
if E384163[I]>=0 and E34428[I]=0 then OTHPAY3[I]=-3;
if -4<E384163[I]<0 then OTHPAY3[I]=E384163[I];
if E384073[I]=2 and E34430[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=2 and -4<E34430[I]<0 THNE OTHPAY3[I]=E34430[I];
if E384073[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/* report hourly wage at the beginning*/
if E23901[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
if E23901[I]>0 and E384073[I]=2 and E34402B[I]>0 and E384163[I]>=0 then
  OTHPAY3[I]=(E384163[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E384073[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
  OTHPAY3[I]=E384163[I]/E23901[I];
if E23901[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E23901[I]);
if E23901[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E384163[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
*missing value*/;
if E384163[I]>=0 and -4<E23901[I]<0 then OTHPAY3[I]=E23901[I];
if E384163[I]>=0 and E23901[I]=0 then OTHPAY3[I]=-3;
if -4<E384163[I]<0 then OTHPAY3[I]=E384163[I];

/*report non-hourly wage*/
if E34402[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
if E34402[I]>0 and E384073[I]=2 and E34402B[I]>0 and E384163[I]>=0 then
  OTHPAY3[I]=(E384163[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384073[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
  OTHPAY3[I]=E384163[I]/E34402[I];
if E34402[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E34402[I]);
if E34402[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E384163[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
if E384073[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E384163[I]>=0 and -4<E34402[I]<0 then OTHPAY3[I]=E34402[I];
if E384163[I]>=0 and E34402[I]=0 then OTHPAY3[I]=-3;
```

Appendix 2: Employment Variable Creation

```
if E384073[I]=2 and E34402B[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=2 and -4<E34402B[I]<0 THNE OTHPAY3[I]=E34402B[I];
if E384073[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];
end;

/*for incentive pay*/
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
  /*report hourly wage at the beginning*/
  if E23901[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
  if E23901[I]>0 and E384074[I]=2 and E34430[I]>0 and E384164[I]>=0 then
    OTHPAY4[I]=(E384164[I]*E34430[I])/E23901[I];
  if E23901[I]>0 and E384074[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
    OTHPAY4[I]=E384164[I]/E23901[I];
  if E23901[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E23901[I]);
  if E23901[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E23901[I]);
  if E23901[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E384164[I]/(E35600[I]*E23901[I]);
  if E23901[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E23901[I]);
  if E23901[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF4[I]=OTHPF4[I]+1;
  *missing value*/;
  if E384164[I]>=0 and -4<E23901[I]<0 then OTHPAY4[I]=E23901[I];
  if E384164[I]>=0 and E23901[I]=0 then OTHPAY4[I]=-3;
  if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];

  /*report non-hourly wage*/
  if E34428[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
  if E34428[I]>0 and E384074[I]=2 and E34430[I]>0 and E384164[I]>=0 then
    OTHPAY4[I]=(E384164[I]*E34430[I])/E34428[I];
  if E34428[I]>0 and E384074[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
    OTHPAY4[I]=E384164[I]/E34428[I];
  if E34428[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E34428[I]);
  if E34428[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E34428[I]);
  if E34428[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
    OTHPAY4[I]=E384164[I]/(E35600[I]*E34428[I]);
  if E34428[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E34428[I]);
  if E34428[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF4[I]=OTHPF4[I]+1;
  *missing value*/;
  if E384164[I]>=0 and -4<E34428[I]<0 then OTHPAY4[I]=E34428[I];
  if E384164[I]>=0 and E34428[I]=0 then OTHPAY4[I]=-3;
  if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];
  if E384074[I]=2 and E34430[I] le 0 then OTHPAY4[I]=-3;
  if E384074[I]=2 and -4<E34430[I]<0 THNE OTHPAY4[I]=E34430[I];
  if E384074[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
  if E384074[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
  /*report hourly wage at the beginning*/
  if E23901[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
  if E23901[I]>0 and E384074[I]=2 and E34402B[I]>0 and E384164[I]>=0 then
    OTHPAY4[I]=(E384164[I]*E34402B[I])/E23901[I];
  if E23901[I]>0 and E384074[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
    OTHPAY4[I]=E384164[I]/E23901[I];
  if E23901[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E23901[I]);
```

Appendix 2: Employment Variable Creation

```
if E23901[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
  OTHPAY4[I]=E384164[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF4[I]=OTHPF4[I]+1;
*missing value*/;
if E384164[I]>=0 and -4<E23901[I]<0 then OTHPAY4[I]=E23901[I];
if E384164[I]>=0 and E23901[I]=0 then OTHPAY4[I]=-3;
if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];

/*report non-hourly wage*/
if E34402[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
if E34402[I]>0 and E384074[I]=2 and E34402B[I]>0 and E384164[I]>=0 then
  OTHPAY4[I]=(E384164[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384074[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
  OTHPAY4[I]=E384164[I]/E34402[I];
if E34402[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E34402[I]);
if E34402[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
  OTHPAY4[I]=E384164[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF4[I]=OTHPF4[I]+1;
if E384074[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E384164[I]>=0 and -4<E34402[I]<0 then OTHPAY4[I]=E34402[I];
if E384164[I]>=0 and E34402[I]=0 then OTHPAY4[I]=-3;
if E384074[I]=2 and E34402B[I] le 0 then OTHPAY4[I]=-3;
if E384074[I]=2 and -4<E34402B[I]<0 then OTHPAY4[I]=E34402B[I];
if E384074[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E384074[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];
end;

/*for others*/
if E38102[I] ne 1 and E38102[I] ne 3 then do; /*with overtime at the beginning*/
  /* report hourly wage at the beginning*/
  if E23901[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
  if E23901[I]>0 and E384075[I]=2 and E34430[I]>0 and E384165[I]>=0 then
    OTHPAY5[I]=(E384165[I]*E34430[I])/E23901[I];
  if E23901[I]>0 and E384075[I] in (3,0,7,12,13,15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
    OTHPAY5[I]=E384165[I]/E23901[I];
  if E23901[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E23901[I]);
  if E23901[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E23901[I]);
  if E23901[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
    OTHPAY5[I]=E384165[I]/(E35600[I]*E23901[I]);
  if E23901[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E23901[I]);
  if E23901[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF5[I]=OTHPF5[I]+1;
  *missing value*/;
  if E384165[I]>=0 and -4<E23901[I]<0 then OTHPAY5[I]=E23901[I];
  if E384165[I]>=0 and E23901[I]=0 then OTHPAY5[I]=-3;
  if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];

  /*report non-hourly wage*/
  if E34428[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
```


Appendix 2: Employment Variable Creation

```
if E34428[I]>0 and E384075[I]=2 and E34430[I]>0 and E384165[I]>=0 then
  OTHPAY5[I]=(E384165[I]*E34430[I])/E34428[I];
if E34428[I]>0 and E384075[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
  OTHPAY5[I]=E384165[I]/E34428[I];
if E34428[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E34428[I]);
if E34428[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E34428[I]);
if E34428[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
  OTHPAY5[I]=E384165[I]/(E35600[I]*E34428[I]);
if E34428[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E34428[I]);
if E34428[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF5[I]=OTHPF5[I]+1;
*missing value*/;
if E384165[I]>=0 and -4<E34428[I]<0 then OTHPAY5[I]=E34428[I];
if E384165[I]>=0 and E34428[I]=0 then OTHPAY5[I]=-3;
if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];
if E384075[I]=2 and E34430[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=2 and -4<E34430[I]<0 THNE OTHPAY5[I]=E34430[I];
if E384075[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;

if E3800B[I]=1 then do; /*without overtime at the beginning*/
/* report hourly wage at the beginning*/
if E23901[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
if E23901[I]>0 and E384075[I]=2 and E34402B[I]>0 and E384165[I]>=0 then
  OTHPAY5[I]=(E384165[I]*E34402B[I])/E23901[I];
if E23901[I]>0 and E384075[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
  OTHPAY5[I]=E384165[I]/E23901[I];
if E23901[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E23901[I]);
if E23901[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E23901[I]);
if E23901[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
  OTHPAY5[I]=E384165[I]/(E35600[I]*E23901[I]);
if E23901[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E23901[I]);
if E23901[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF5[I]=OTHPF5[I]+1;
*missing value*/;
if E384165[I]>=0 and -4<E23901[I]<0 then OTHPAY5[I]=E23901[I];
if E384165[I]>=0 and E23901[I]=0 then OTHPAY5[I]=-3;
if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];

/*report non-hourly wage*/
if E34402[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
if E34402[I]>0 and E384075[I]=2 and E34402B[I]>0 and E384165[I]>=0 then
  OTHPAY5[I]=(E384165[I]*E34402B[I])/E34402[I];
if E34402[I]>0 and E384075[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
  OTHPAY5[I]=E384165[I]/E34402[I];
if E34402[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E34402[I]);
if E34402[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E34402[I]);
if E34402[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
  OTHPAY5[I]=E384165[I]/(E35600[I]*E34402[I]);
if E34402[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E34402[I]);
if E34402[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF5[I]=OTHPF5[I]+1;
if E384075[I] in (9,14) then OTHPAY5[I]=0;
*missing value*/;
if E384165[I]>=0 and -4<E34402[I]<0 then OTHPAY5[I]=E34402[I];
if E384165[I]>=0 and E34402[I]=0 then OTHPAY5[I]=-3;
```

Appendix 2: Employment Variable Creation

```
if E384075[I]=2 and E34402B[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=2 and -4<E34402B[I]<0 THNE OTHPAY5[I]=E34402B[I];
if E384075[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;
end;

/** case vi. more then one compensation at the beginning, diff no. of hours. ***/
if E38330[I]=1 and E20700[I]=1 and (E38102[I]=1 or E38102[I]=3) then do;

/* for tips*/
if E38103[I]>0 and E384071[I]=1 and E384161[I]>=0 then OTHPAY1[I]=E384161[I];
if E38103[I]>0 and E384071[I]=2 and E38116B[I]>0 and E384161[I]>=0 then
    OTHPAY1[I]=(E384161[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E384071[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384161[I]>=0 then
    OTHPAY1[I]=E384161[I]/E38103[I];
if E38103[I]>0 and E384071[I]=4 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2*E38103[I]);
if E38103[I]>0 and E384071[I]=5 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(4.3*E38103[I]);
if E38103[I]>0 and E384071[I]=6 and E384161[I]>=0 and E35600[I]>0 then
    OTHPAY1[I]=E384161[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E384071[I]=8 and E384161[I]>=0 then OTHPAY1[I]=E384161[I]/(2.15*E38103[I]);
if E38103[I]>0 and E384071[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF1[I]=OTHPF1[I]+1;
if E384071[I] in (9,14) then OTHPAY1[I]=0;
/*missing value*/;
if E384161[I]>=0 and -4<E38103[I]<0 then OTHPAY1[I]=E38103[I];
if E384161[I]>=0 and E38103[I]=0 then OTHPAY1[I]=-3;
if -4<E384161[I]<0 then OTHPAY1[I]=E384161[I];
if E384071[I]=2 and E38116B[I]=0 then OTHPAY1[I]=-3;
if E384071[I]=2 and -4<E38116B[I]<0 then OTHPAY1[I]=E38116B[I];
if E384071[I]=6 and E35600[I] le 0 then OTHPAY1[I]=-3;
if E384071[I]=6 and -4<E35600[I]<0 then OTHPAY1[I]=E35600[I];

/*for commissions*/
if E38103[I]>0 and E384072[I]=1 and E384162[I]>=0 then OTHPAY2[I]=E384162[I];
if E38103[I]>0 and E384072[I]=2 and E38116B[I]>0 and E384162[I]>=0 then
    OTHPAY2[I]=(E384162[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E384072[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384162[I]>=0 then
    OTHPAY2[I]=E384162[I]/E38103[I];
if E38103[I]>0 and E384072[I]=4 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2*E38103[I]);
if E38103[I]>0 and E384072[I]=5 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(4.3*E38103[I]);
if E38103[I]>0 and E384072[I]=6 and E384162[I]>=0 and E35600[I]>0 then
    OTHPAY2[I]=E384162[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E384072[I]=8 and E384162[I]>=0 then OTHPAY2[I]=E384162[I]/(2.15*E38103[I]);
if E38103[I]>0 and E384072[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
    OTHPF2[I]=OTHPF2[I]+1;
if E384072[I] in (9,14) then OTHPAY2[I]=0;
/*missing value*/;
if E384162[I]>=0 and -4<E38103[I]<0 then OTHPAY2[I]=E38103[I];
if E384162[I]>=0 and E38103[I]=0 then OTHPAY2[I]=-3;
if -4<E384162[I]<0 then OTHPAY2[I]=E384162[I];
if E384072[I]=2 and E38116B[I]=0 then OTHPAY2[I]=-3;
if E384072[I]=2 and -4<E38116B[I]<0 then OTHPAY2[I]=E38116B[I];
if E384072[I]=6 and E35600[I] le 0 then OTHPAY2[I]=-3;
if E384072[I]=6 and -4<E35600[I]<0 then OTHPAY2[I]=E35600[I];

/*for bonuses*/
```

Appendix 2: Employment Variable Creation

```
if E38103[I]>0 and E384073[I]=1 and E384163[I]>=0 then OTHPAY3[I]=E384163[I];
if E38103[I]>0 and E384073[I]=2 and E38116B[I]>0 and E384163[I]>=0 then
  OTHPAY3[I]=(E384163[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E384073[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384163[I]>=0 then
  OTHPAY3[I]=E384163[I]/E38103[I];
if E38103[I]>0 and E384073[I]=4 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2*E38103[I]);
if E38103[I]>0 and E384073[I]=5 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(4.3*E38103[I]);
if E38103[I]>0 and E384073[I]=6 and E384163[I]>=0 and E35600[I]>0 then
  OTHPAY3[I]=E384163[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E384073[I]=8 and E384163[I]>=0 then OTHPAY3[I]=E384163[I]/(2.15*E38103[I]);
if E38103[I]>0 and E384073[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF3[I]=OTHPF3[I]+1;
if E384073[I] in (9,14) then OTHPAY3[I]=0;
*missing value*/;
if E384163[I]>=0 and -4<E38103[I]<0 then OTHPAY3[I]=E38103[I];
if E384163[I]>=0 and E38103[I]=0 then OTHPAY3[I]=-3;
if -4<E384163[I]<0 then OTHPAY3[I]=E384163[I];
if E384073[I]=2 and E38116B[I]=0 then OTHPAY3[I]=-3;
if E384073[I]=2 and -4<E38116B[I]<0 then OTHPAY3[I]=E38116B[I];
if E384073[I]=6 and E35600[I] le 0 then OTHPAY3[I]=-3;
if E384073[I]=6 and -4<E35600[I]<0 then OTHPAY3[I]=E35600[I];

/*for incentive pay*/
if E38103[I]>0 and E384074[I]=1 and E384164[I]>=0 then OTHPAY4[I]=E384164[I];
if E38103[I]>0 and E384074[I]=2 and E38116B[I]>0 and E384164[I]>=0 then
  OTHPAY4[I]=(E384164[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E384074[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384164[I]>=0 then
  OTHPAY4[I]=E384164[I]/E38103[I];
if E38103[I]>0 and E384074[I]=4 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2*E38103[I]);
if E38103[I]>0 and E384074[I]=5 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(4.3*E38103[I]);
if E38103[I]>0 and E384074[I]=6 and E384164[I]>=0 and E35600[I]>0 then
  OTHPAY4[I]=E384164[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E384074[I]=8 and E384164[I]>=0 then OTHPAY4[I]=E384164[I]/(2.15*E38103[I]);
if E38103[I]>0 and E384074[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF4[I]=OTHPF4[I]+1;
if E384074[I] in (9,14) then OTHPAY4[I]=0;
*missing value*/;
if E384164[I]>=0 and -4<E38103[I]<0 then OTHPAY4[I]=E38103[I];
if E384164[I]>=0 and E38103[I]=0 then OTHPAY4[I]=-3;
if -4<E384164[I]<0 then OTHPAY4[I]=E384164[I];
if E384074[I]=2 and E38116B[I]=0 then OTHPAY4[I]=-3;
if E384074[I]=2 and -4<E38116B[I]<0 then OTHPAY4[I]=E38116B[I];
if E384074[I]=6 and E35600[I] le 0 then OTHPAY4[I]=-3;
if E384074[I]=6 and -4<E35600[I]<0 then OTHPAY4[I]=E35600[I];

/*for others*/
if E38103[I]>0 and E384075[I]=1 and E384165[I]>=0 then OTHPAY5[I]=E384165[I];
if E38103[I]>0 and E384075[I]=2 and E38116B[I]>0 and E384165[I]>=0 then
  OTHPAY5[I]=(E384165[I]*E38116B[I])/E38103[I];
if E38103[I]>0 and E384075[I] in (3,0,7,12,13.15,16,17,99,21,22,23,24,25,26,28) and E384165[I]>=0 then
  OTHPAY5[I]=E384165[I]/E38103[I];
if E38103[I]>0 and E384075[I]=4 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2*E38103[I]);
if E38103[I]>0 and E384075[I]=5 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(4.3*E38103[I]);
if E38103[I]>0 and E384075[I]=6 and E384165[I]>=0 and E35600[I]>0 then
  OTHPAY5[I]=E384165[I]/(E35600[I]*E38103[I]);
if E38103[I]>0 and E384075[I]=8 and E384165[I]>=0 then OTHPAY5[I]=E384165[I]/(2.15*E38103[I]);
```

Appendix 2: Employment Variable Creation

```
if E38103[I]>0 and E384075[I] in (0,7,12,13,14,15,16,17,99,21,22,23,24,25,26,28) then
  OTHPF5[I]=OTHPF5[I]+1;
if E384075[I] in (9,14) then OTHPAY5[I]=0;
*missing value*/;
if E384165[I]>=0 and -4<E38103[I]<0 then OTHPAY5[I]=E38103[I];
if E384165[I]>=0 and E38103[I]=0 then OTHPAY5[I]=-3;
if -4<E384165[I]<0 then OTHPAY5[I]=E384165[I];
if E384075[I]=2 and E38116B[I]=0 then OTHPAY5[I]=-3;
if E384075[I]=2 and -4<E38116B[I]<0 then OTHPAY5[I]=E38116B[I];
if E384075[I]=6 and E35600[I] le 0 then OTHPAY5[I]=-3;
if E384075[I]=6 and -4<E35600[I]<0 then OTHPAY5[I]=E35600[I];
end;
end;
end;

/**** Part 4: Overall End Hourly Compensation ****/

do I=1 to 9;
if E37901B[I]=1 or E59900[I]=1 then do;
  HRCOMP[I]=0;
if HRWG[I] ge 0 then HRCOMP[I]=HRCOMP[I]+HRWG[I];
if OT[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OT[I];
if OTHPAY1[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY1[I];
if OTHPAY2[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY2[I];
if OTHPAY3[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY3[I];
if OTHPAY4[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY4[I];
if OTHPAY5[I] ge 0 then HRCOMP[I]=HRCOMP[I]+OTHPAY5[I];
if -4<HRWG[I]<0 or -4<OT[I]<0 or -4<OTHPAY1[I]<0 or -4<OTHPAY2[I]<0 or -4<OTHPAY3[I]<0 or -
  4<OTHPAY4[I]<0 or -4<OTHPAY5[I]<0 then HRCOMP[I]=-3;
if HRWG[I]=-4 then HRCOMP[I]=-4;
end;
end;

/***** Section 3: Job lasts 13 weeks or less *****/

JLENG01=-4;          JLENG02=-4;          JLENG03=-4;
JLENG04=-4;          JLENG05=-4;          JLENG06=-4;
JLENG07=-4;          JLENG08=-4;          JLENG09=-4;

array JLENG JLENG01 JLENG02 JLENG03 JLENG04 JLENG05 JLENG06 JLENG07 JLENG08 JLENG09
      JLENG10;

do I=1 to 9;
if E37701[I]=0 then JLENG[I]=0;
if E58201[I]=0 then JLENG[I]=0;
if E37901B[I]>-1 or E59900[I]>-1 or E58401[I]>-1 then do;
  if E37901B[I]=1 or E59900[I]=1 or E58401[I]=0 then JLENG[I]=1;
  else JLENG[I]=0;
end;
end;

do I=1 to 9;
if JLENG[I]>-4 and HRWG[I]=-4 then HRWG[I]=-3;
if JLENG[I]>-4 and HRCOMP[I]=-4 then HRCOMP[I]=-3;
end;
```

Appendix 2: Employment Variable Creation

```
array HRWGR HRWGR01 HRWGR02 HRWGR03 HRWGR04 HRWGR05 HRWGR06 HRWGR07  
      HRWGR08 HRWGR09 HRWGR10;
```

```
HRWGR01=-4;      HRWGR02=-4;      HRWGR03=-4;  
HRWGR04=-4;      HRWGR05=-4;      HRWGR06=-4;  
HRWGR07=-4;      HRWGR08=-4;      HRWGR09=-4;
```

```
do I=1 to 9;  
  if HRWG[I] ne -4 then do;  
    HRWGR[I]=ROUND(HRWG[I],1);  
  end;  
end;
```

```
array HRCOMR HRCOMR01 HRCOMR02 HRCOMR03 HRCOMR04 HRCOMR05 HRCOMR06  
      HRCOMR07 HRCOMR08 HRCOMR09;
```

```
HRCOMR01=-4;      HRCOMR02=-4;      HRCOMR03=-4;  
HRCOMR04=-4;      HRCOMR05=-4;      HRCOMR06=-4;  
HRCOMR07=-4;      HRCOMR08=-4;      HRCOMR09=-4;
```

```
array HRCOMP HRCOMP01 HRCOMP02 HRCOMP03 HRCOMP04 HRCOMP05 HRCOMP06  
      HRCOMP07 HRCOMP08 HRCOMP09;
```

```
do I=1 to 9;  
  if HRCOMP[I] ne -4 then do;  
    HRCOMR[I]=ROUND(HRCOMP[I],1);  
  end;  
end;
```

```
endsas;
```

NUMBER OF WEEKS WORKED DURING 19XX

Variables Created: CV_WKSWK_YR.80 – CV_WKSWK_YR.99

Programs Used

This program uses **bdate1.sas** and **emp_begin.sas** as input (see the first page of this appendix for details).

This program counts the number of weeks each respondent worked at any employee-type job for each year of potential work activity (1980-99). Respondents not working in a given year are given a default value of zero (0) weeks worked. Otherwise, the variable indicates the actual cumulative weeks worked on all jobs in that year.

```
/* Section 1: Create variables for # of weeks worked at
any job during a given year based on round 2 data. */
```

```
array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;
array alljobs (i) wks1-wks1044;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
array uid (i) uid1-uid9;
```

```
do over alljobs; alljobs=0; end;
```

```
/** Overlay multiple jobs over JOB 1 work weeks **/
/* "Alljobs" is the event history array that accounts for
all jobs worked over 1980-1999. */
```

```
do over job1wks; alljobs=job1wks; end;
```

```
do over alljobs;
  if job2wks=1 then do; alljobs=job2wks; end;
  if job2wks=-3 and alljobs=0 then do;
    alljobs=job2wks; end;
end;
```

```
/* The program repeats the "do over all jobs" command
for jobs 3-9. The loops are not shown here; contact
NLS User Services for more information. */
```

```
/* Calculate cumulative wks on all jobs for each year */
/* 1980 */
```

```
wksw80=0;
do i=1 to 52;
  if alljobs=1 then do; wksw80=wksw80+1; end;
end;
do i=1 to 52;
  if alljobs=-3 then do; wksw80=-3; end;
end;
```

```
/******At this point the program loops through the
same code used above for 1980 for each year 1981-98,
creating the variables wks81, wks82, wks83, and so on
through wks99. These loops are deleted due to space
considerations; users who need to see the entire code
should contact NLS User Services. The week numbers
for the "do i" statement for each year are as follows:
```

1981	53-104	1991	575-626
1982	105-156	1992	627-678
1983	157-209	1993	679-730
1984	210-261	1994	731-783
1985	262-313	1995	784-835
1986	314-365	1996	836-887
1987	366-417	1997	888-939
1988	418-470	1998	940-991
1989	471-522	1999	992-1044
1990	523-574		*****/

```
do i=1 to 7;
```

```
/* start date invalid */
```

```
if starw<0 and starw>=4 then do;
```

```
  if stopw>1 then do; wksw80=-3; end;
  if stopw>52 then do; wksw81=-3; end;
  if stopw>104 then do; wksw82=-3; end;
  if stopw>156 then do; wksw83=-3; end;
  if stopw>209 then do; wksw84=-3; end;
  if stopw>261 then do; wksw85=-3; end;
  if stopw>313 then do; wksw86=-3; end;
  if stopw>365 then do; wksw87=-3; end;
  if stopw>417 then do; wksw88=-3; end;
  if stopw>470 then do; wksw89=-3; end;
  if stopw>522 then do; wksw90=-3; end;
  if stopw>574 then do; wksw91=-3; end;
  if stopw>626 then do; wksw92=-3; end;
  if stopw>678 then do; wksw93=-3; end;
  if stopw>730 then do; wksw94=-3; end;
  if stopw>783 then do; wksw95=-3; end;
  if stopw>834 then do; wksw96=-3; end;
  if stopw>887 then do; wksw97=-3; end;
  if stopw>939 then do; wksw98=-3; end;
  if stopw>991 then do; wksw99=-3; end;
```

```
end;
```

```
/* stop date invalid */
```

```
if stopw<0 and stopw>=4 then do;
```

Appendix 2: Employment Variable Creation

```

if starw<53 then do; wksw80=-3; end;
if starw<105 then do; wksw81=-3; end;
if starw<157 then do; wksw82=-3; end;
if starw<210 then do; wksw83=-3; end;
if starw<262 then do; wksw84=-3; end;
if starw<314 then do; wksw85=-3; end;
if starw<366 then do; wksw86=-3; end;
if starw<418 then do; wksw87=-3; end;
if starw<471 then do; wksw88=-3; end;
if starw<523 then do; wksw89=-3; end;
if starw<575 then do; wksw90=-3; end;
if starw<627 then do; wksw91=-3; end;
if starw<679 then do; wksw92=-3; end;
if starw<731 then do; wksw93=-3; end;
if starw<784 then do; wksw94=-3; end;
if starw<836 then do; wksw95=-3; end;
if starw<888 then do; wksw96=-3; end;
if starw<940 then do; wksw97=-3; end;
if starw<992 then do; wksw98=-3; end;
if starw<1044 then do; wksw99=-3; end;
end;

/** Include valid skips ***/
if e200=0 then do;
  wksw80=-4; wksw81=-4; wksw82=-4; wksw83=-4;
  wksw84=-4; wksw85=-4; wksw86=-4; wksw87=-4;
  wksw88=-4; wksw89=-4; wksw90=-4; wksw91=-4;
  wksw92=-4; wksw93=-4; wksw94=-4; wksw95=-4;
  wksw96=-4; wksw97=-4; wksw98=-4; wksw99=-4;
end;
if e200=-3 then do;
  wksw80=-3; wksw81=-3; wksw82=-3; wksw83=-3;
  wksw84=-3; wksw85=-3; wksw86=-3; wksw87=-3;
  wksw88=-3; wksw89=-3; wksw90=-3; wksw91=-3;
  wksw92=-3; wksw93=-3; wksw94=-3; wksw95=-3;
  wksw96=-3; wksw97=-3; wksw98=-3; wksw99=-3;
end;
if e200=-5 then do;
  wksw80=-5; wksw81=-5; wksw82=-5; wksw83=-5;
  wksw84=-5; wksw85=-5; wksw86=-5; wksw87=-5;
  wksw88=-5; wksw89=-5; wksw90=-5; wksw91=-5;
  wksw92=-5; wksw93=-5; wksw94=-5; wksw95=-5;
  wksw96=-5; wksw97=-5; wksw98=-5; wksw99=-5;
end;

check=0;
do over starw;
  if starw>0 then do; if starw<dliwk then check=1; end;
end;

/* Section 2: Take the created variables for total weeks
worked in any job during a given year for Round 1 and
Round 2 and add them. One variable is created for
each year 1980-1999. The default is set to zero. */

/* Define r1ten99 as zero to match up array lengths */
r1ten99=0;

array r1ten (20) r1ten80-r1ten99; /* R1 created var. */
array wksw (20) wksw80-wksw99; /* R2 created var. */
array twks (20) twks80-twks99;

/* Initialize weeks worked in a given year to be zero */
do i=1 to 20; twks(i)=0; end;

/* Begin by splitting up periods where R1 and R2
exclusively collect weeks worked information. Define
the R1 interview year as the split. Any weeks worked
information collected before the R1 interview year
should be independent of data collected in R2. */
do i=1 to 20;
  if r1ten(i)>0 then do; twks(i)=r1ten(i); end;
  if wksw(i)>0 then do; twks(i)=wksw(i); end;
  if r1ten(i)>0 and wksw(i)>0 then do;
    twks(i)=r1ten(i)+wksw(i); end;
end;

/* If either created variable from R1 or R2 is a -1, -2 or
-3, then the overall created variable will be -1, -2 or -3,
respectively. */
do i=1 to 20;
  if -4<r1ten(i)<0 then do; twks(i)=r1ten(i); end;
  if wksw(i)<0 then do; twks(i)=wksw(i); end;
end;

/* Special hand edit case. This respondent reported was
interviewed in R1 and started a new job later that week.
The UID code begins with "98" but the start week
must be updated by one week. Otherwise, the program
has the respondent working 53 weeks in 1998. */
if pubid=525 then twks98=52;

oops=0;
do i=16 to 20; if twks(i)=>53 then do; oops=1; end; end;

/* The following hand edit cases result from
respondents reporting jobs during the R2 interview that
began before the R1 interview date. */
if pubid=32 then twks97=52;
if pubid=2398 then do; twks96=52; twks97=50; end;
if pubid=2698 then do; twks96=52; twks97=48; end;
if pubid=3222 then twks97=52;
if pubid=3499 then twks97=40;
if pubid=4505 then twks97=52;
if pubid=4921 then do; twks94=10; twks95=11;
  twks96=11; end;
if pubid=5178 then do; twks96=38; twks97=45; end;
if pubid=5497 then twks97=49;
if pubid=5658 then twks97=36;
if pubid=6324 then do; twks94=13; twks95=14;
  twks96=14; twks97=41; twks98=33; end;
if pubid=6949 then twks97=11;
if pubid=3222 then do; twks96=52; twks97=52; end;

endsas;

```

NUMBER OF WEEKS WORKED SINCE LAST INTERVIEW

Variables Created: CV_WKSWK_DLI

Programs Used

This program uses **bdate1.sas** and **emp_begin.sas** as input (see the first page of this appendix for details).

Created for each individual, this program counts the number of weeks the respondent worked at any employee-type job since the last interview. Respondents not working in a given year are given a default value of zero.

```

array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;
array alljobs (i) wks1-wks1044;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;

do over alljobs; alljobs=0; end;

/* We update the startdate to the R1 interview date
when the stardate is before the interview date and the
stop date is after the interview date. */
alert=0;
do over starw;
  if starw<dliwk and starw>0 and stopw>dliwk then do;
    starw=dliwk; alert=1; end;
end;

/* Jobs that begin and end before the Round1 interview
date will not be counted in this program. */
array switch (i) switch1-switch9;
do over switch; switch=0; end;
whoa=0;
do over starw;
  if starw<dliwk and stopw<dliwk and starw>0 and
    stopw>0 then do; switch=1; whoa=1; end; end;

do i=1 to 1044;
  if switch1=1 then do; job1wks=0; end;
  if switch2=1 then do; job2wks=0; end;
  if switch3=1 then do; job3wks=0; end;
  if switch4=1 then do; job4wks=0; end;
  if switch5=1 then do; job5wks=0; end;
  if switch6=1 then do; job6wks=0; end;
  if switch7=1 then do; job7wks=0; end;
  if switch8=1 then do; job8wks=0; end;
  if switch9=1 then do; job9wks=0; end;
end;

/** Overlay multiple jobs over JOB 1 work weeks **/

do over job1wks; alljobs=job1wks; end;

do over alljobs;
  if job2wks=1 then do;
    alljobs=job2wks; end;
  if job2wks=-3 and alljobs=0 then do;
    alljobs=job2wks; end;
end;
/* The program repeats the "do over all jobs" command
for jobs 3-9. The loops are not shown here; contact
NLS User Services for more information. */

/* Calculate cumulative weeks on all jobs since birth */
allwks=0;
if birthwks>0 then do;
  do i=birthwks to 1044;
    if alljobs=1 then do; allwks=allwks+1; end;
  end;
  do i=birthwks to 1044;
    if alljobs=-3 then do; allwks=-3; end;
  end;
  if starw1<0 and starw1>-4 then do; allwks=-3; end;
  if starw2<0 and starw2>-4 then do; allwks=-3; end;
  if starw3<0 and starw3>-4 then do; allwks=-3; end;
  if starw4<0 and starw4>-4 then do; allwks=-3; end;
  if starw5<0 and starw5>-4 then do; allwks=-3; end;
  if starw6<0 and starw6>-4 then do; allwks=-3; end;
  if starw7<0 and starw7>-4 then do; allwks=-3; end;
  if starw8<0 and starw8>-4 then do; allwks=-3; end;
  if starw9<0 and starw9>-4 then do; allwks=-3; end;
  if stopw1<0 and stopw1>-4 then do; allwks=-3; end;
  if stopw2<0 and stopw2>-4 then do; allwks=-3; end;
  if stopw3<0 and stopw3>-4 then do; allwks=-3; end;
  if stopw4<0 and stopw4>-4 then do; allwks=-3; end;
  if stopw5<0 and stopw5>-4 then do; allwks=-3; end;
  if stopw6<0 and stopw6>-4 then do; allwks=-3; end;
  if stopw7<0 and stopw7>-4 then do; allwks=-3; end;
  if stopw8<0 and stopw8>-4 then do; allwks=-3; end;
  if stopw9<0 and stopw9>-4 then do; allwks=-3; end;
end;

if e200=0 then do; allwks=-4; end;
if e200=-3 then do; allwks=-3; end;
if e200=-5 then do; allwks=-5; end;

endsas;

```


NUMBER OF WEEKS WORKED SINCE AGE 14

Variables Created: CV_WKSWK_EVER

Programs Used

This program uses `bdate1.sas` and `emp_begin.sas` as input (see the first page of this appendix for details).

For each respondent, this program creates a variable counting the number of weeks worked at any employee-type job since the age of 14. Respondents not working are given a default value of zero.

```

/* Section 1: Creates a variable for number of weeks
worked at any job since age 14 using R2 data. */

array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;
array alljobs (i) wks1-wks1044;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;

/** Overlay multiple jobs over JOB 1 work weeks **/

do over job1wks; alljobs=job1wks; end;

do over alljobs;
  if job2wks=1 then do;
    alljobs=job2wks; end;
  if job2wks=-3 and alljobs=0 then do;
    alljobs=job2wks; end;
end;

/* The program repeats the "do over all jobs" command
for jobs 3-9. The loops are not shown here; contact
NLS User Services for more information. */

/* Calculate cumulative wks on all jobs since age 14 */
wks14w=0;

if age14wk>0 then do;
  do i=age14wk to 1044;
    if alljobs=1 then do; wks14w=wks14w+1; end;
    if alljobs=-3 then do; wks14w=-3; end;
  end;
end;

/* Start date invalid */
do i=1 to 9;
if starw<0 and starw>-4 then do;
  if age14wk<53 and stopw>1 then do;
    wks14w=-3; end;
  if age14wk<105 and stopw>52 then do;
    wks14w=-3; end;
  if age14wk<157 and stopw>104 then do;
    wks14w=-3; end;
  if age14wk<210 and stopw>156 then do;
    wks14w=-3; end;
  if age14wk<262 and stopw>209 then do;
    wks14w=-3; end;
  if age14wk<314 and stopw>261 then do;
    wks14w=-3; end;
  if age14wk<366 and stopw>313 then do;
    wks14w=-3; end;
  if age14wk<418 and stopw>365 then do;
    wks14w=-3; end;
  if age14wk<471 and stopw>417 then do;
    wks14w=-3; end;
  if age14wk<523 and stopw>470 then do;
    wks14w=-3; end;
  if age14wk<575 and stopw>522 then do;
    wks14w=-3; end;
  if age14wk<627 and stopw>574 then do;
    wks14w=-3; end;
  if age14wk<679 and stopw>626 then do;
    wks14w=-3; end;
  if age14wk<731 and stopw>678 then do;
    wks14w=-3; end;
  if age14wk<784 and stopw>730 then do;
    wks14w=-3; end;
  if age14wk<836 and stopw>783 then do;
    wks14w=-3; end;
  if age14wk<888 and stopw>835 then do;
    wks14w=-3; end;
  if age14wk<940 and stopw>887 then do;
    wks14w=-3; end;
  if age14wk<991 and stopw>939 then do;
    wks14w=-3; end;
  if age14wk<1044 and stopw>991 then do;
    wks14w=-3; end;
end;

/* Stop date invalid */
if stopw<0 and stopw>-4 then do;
  if age14wk<53 and starw<53 then do;
    wks14w=-3; end;
  if age14wk<105 and starw<105 then do;
    wks14w=-3; end;

```

Appendix 2: Employment Variable Creation

```
if age14wk<157 and starw<157 then do;
  wks14w=-3; end;
if age14wk<210 and starw<210 then do;
  wks14w=-3; end;
if age14wk<262 and starw<262 then do;
  wks14w=-3; end;
if age14wk<314 and starw<314 then do;
  wks14w=-3; end;
if age14wk<366 and starw<366 then do;
  wks14w=-3; end;
if age14wk<418 and starw<418 then do;
  wks14w=-3; end;
if age14wk<471 and starw<471 then do;
  wks14w=-3; end;
if age14wk<523 and starw<523 then do;
  wks14w=-3; end;
if age14wk<575 and starw<575 then do;
  wks14w=-3; end;
if age14wk<627 and starw<627 then do;
  wks14w=-3; end;
if age14wk<679 and starw<679 then do;
  wks14w=-3; end;
if age14wk<731 and starw<731 then do;
  wks14w=-3; end;
if age14wk<784 and starw<784 then do;
  wks14w=-3; end;
if age14wk<836 and starw<836 then do;
  wks14w=-3; end;
if age14wk<888 and starw<888 then do;
  wks14w=-3; end;
if age14wk<940 and starw<940 then do;
  wks14w=-3; end;
if age14wk<991 and starw<991 then do;
  wks14w=-3; end;
if age14wk<1044 and starw<1044 then do;
  wks14w=-3; end;
end;
end;

/** Include valid skips */
if e200=0 then do; wks14w=-4; end;
if e200=-3 then do; wks14w=-3; end;
if e200=-5 then do; wks14w=-5; end;

/* Section 2: Calculate the total number of weeks
worked since age 14 by adding together the created
variables from R1 and R2.

/* Initialize created variable for both rounds to zero. */
allwks14=0;

/* If both the R1 and R2 variables are positive, we can
simply add them together. If one is positive and one is
not, then the negative value will be the total created
variable for both rounds. If neither is positive, then the
total created variable for both rounds will be zero. */

/* Both rounds positive*/
if r1wks14>0 then do; allwks14=r1wks14; end;
if wks14w>0 then do; allwks14=wks14w; end;
if r1wks14>0 and wks14w>0 then do;
  allwks14=r1wks14+wks14w;
end;

/* If one variable is negative. */
if (r1wks14=-2 or r1wks14=-3) then
  allwks14=r1wks14;
if (wks14w=-2 or wks14w=-3 or wks14w=-5) then
  allwks14=wks14w;

/* The following hand edit cases result from
respondents reporting jobs during the R2 interview that
began and ended before the R1 interview date. */
if pubid=32 then allwks14=187;
if pubid=5497 then allwks14=137;
if pubid=2398 then allwks14=95;
if pubid=5658 then allwks14=132;
if pubid=2615 then allwks14=78;
if pubid=6324 then allwks14=91;
if pubid=2698 then allwks14=174;
if pubid=6949 then allwks14=29;
if pubid=4921 then allwks14=39;
if pubid=3222 then allwks14=190;
if pubid=5178 then allwks14=85;
if pubid=3003 then allwks14=122;

endsas;
```

WEEKS WORKED AT EMPLOYEE JOB #X DURING 19XX

Variables Created: CV_WKSWK_JOB_YR.01.80 – CV_WKSWK_JOB_YR.01.99
CV_WKSWK_JOB_YR.02.80 – CV_WKSWK_JOB_YR.02.99 etc. through job #9

Programs Used

This program uses `emp_begin.sas` as input (see the first page of this appendix for details).

This program creates variables for each of the respondent's jobs counting the number of weeks worked in each calendar year. A variable is created for each potential job even if the respondent has worked no jobs in a given year with the default value set to zero (0). The most jobs held by any respondent as of round 2 was nine, so variables are created for nine jobs for each respondent.

```
/* Section 1: Create the Round2 variable for number
of weeks worked at a given job in a given year. */
```

```
array starw (i) starw1-starw9;
array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;
```

```
/** Calculate cumulative weeks on individual jobs for
each year **/
```

```
/* 1980 */
```

```
wksw801=0; wksw802=0; wksw803=0; wksw804=0;
wksw805=0; wksw806=0; wksw807=0; wksw808=0;
wksw809=0;
```

```
do i=1 to 52;
  if job1wks=1 then do; wksw801=wksw801+1; end;
  if job2wks=1 then do; wksw802=wksw802+1; end;
  if job3wks=1 then do; wksw803=wksw803+1; end;
  if job4wks=1 then do; wksw804=wksw804+1; end;
  if job5wks=1 then do; wksw805=wksw805+1; end;
  if job6wks=1 then do; wksw806=wksw806+1; end;
  if job7wks=1 then do; wksw807=wksw807+1; end;
  if job8wks=1 then do; wksw808=wksw808+1; end;
  if job9wks=1 then do; wksw809=wksw809+1; end;
end;
```

```
do i=1 to 52;
  if job1wks=-3 then do; wksw801=-3; end;
  if job2wks=-3 then do; wksw802=-3; end;
  if job3wks=-3 then do; wksw803=-3; end;
  if job4wks=-3 then do; wksw804=-3; end;
  if job5wks=-3 then do; wksw805=-3; end;
  if job6wks=-3 then do; wksw806=-3; end;
  if job7wks=-3 then do; wksw807=-3; end;
  if job8wks=-3 then do; wksw808=-3; end;
```

```
if job9wks=-3 then do; wksw809=-3; end;
end;
```

```
/** The same three sets of commands applied to the
1980 weeks variables are used for each subsequent year.
However, this code is not shown here due to space
considerations. Researchers needing the complete code
should contact NLS User Services. The variables and
"do i" statements for each year are as follows:
```

```
1981 wksw811 wksw819 do i=53 to 104
1982 wksw821 wksw829 do i=105 to 156
1983 wksw831 wksw839 do i=157 to 209
1984 wksw841 wksw849 do i=210 to 261
1985 wksw851 wksw859 do i=262 to 313
1986 wksw861 wksw869 do i=314 to 365
1987 wksw871 wksw879 do i=366 to 417
1988 wksw881 wksw889 do i=418 to 470
1989 wksw891 wksw899 do i=471 to 522
1990 wksw901 wksw909 do i=523 to 574
1991 wksw911 wksw919 do i=575 to 626
1992 wksw921 wksw929 do i=627 to 678
1993 wksw931 wksw939 do i=679 to 730
1994 wksw941 wksw949 do i=731 to 783
1995 wksw951 wksw959 do i=784 to 835
1996 wksw961 wksw969 do i=836 to 887
1997 wksw971 wksw979 do i=888 to 939
1998 wksw981 wksw989 do i=940 to 991
1999 wksw991 wksw999 do i=992 to 1044 **/
```

```
/** Insert valid skips **/
```

```
if e200=0 then do;
  wksw801=-4; /* and so on through wksw809=-4 */
  /* and so on through wksw991 through wksw999 */
end;
```

```
if e200=-3 then do;
  wksw801=-3; /* and so on through wksw809=-3 */
  /* and so on through wksw991 through wksw999 */
end;
```

```
if e200=-5 then do;
  wksw801=-5; /* and so on through wksw809=-5 */
  /* and so on through wksw991 through wksw999 */
end;
```

```

/* Start date invalid */
if starw1<0 and starw1>-4 then do;
  if stopw1>1 then do; wksw801=-3; end;
  if stopw1>52 then do; wksw811=-3; end;
  if stopw1>104 then do; wksw821=-3; end;
  if stopw1>156 then do; wksw831=-3; end;
  if stopw1>209 then do; wksw841=-3; end;
  if stopw1>261 then do; wksw851=-3; end;
  if stopw1>313 then do; wksw861=-3; end;
  if stopw1>365 then do; wksw871=-3; end;
  if stopw1>417 then do; wksw881=-3; end;
  if stopw1>470 then do; wksw891=-3; end;
  if stopw1>522 then do; wksw901=-3; end;
  if stopw1>574 then do; wksw911=-3; end;
  if stopw1>626 then do; wksw921=-3; end;
  if stopw1>678 then do; wksw931=-3; end;
  if stopw1>730 then do; wksw941=-3; end;
  if stopw1>783 then do; wksw951=-3; end;
  if stopw1>835 then do; wksw961=-3; end;
  if stopw1>887 then do; wksw971=-3; end;
  if stopw1>939 then do; wksw981=-3; end;
  if stopw1>991 then do; wksw991=-3; end;
end;

/* Stop date invalid */
if stopw1<0 and stopw1>-4 then do;
  if starw1<53 then do; wksw801=-3; end;
  if starw1<105 then do; wksw811=-3; end;
  if starw1<157 then do; wksw821=-3; end;
  if starw1<210 then do; wksw831=-3; end;
  if starw1<262 then do; wksw841=-3; end;
  if starw1<314 then do; wksw851=-3; end;
  if starw1<366 then do; wksw861=-3; end;
  if starw1<418 then do; wksw871=-3; end;
  if starw1<471 then do; wksw881=-3; end;
  if starw1<523 then do; wksw891=-3; end;
  if starw1<575 then do; wksw901=-3; end;
  if starw1<627 then do; wksw911=-3; end;
  if starw1<679 then do; wksw921=-3; end;
  if starw1<731 then do; wksw931=-3; end;
  if starw1<784 then do; wksw941=-3; end;
  if starw1<836 then do; wksw951=-3; end;
  if starw1<888 then do; wksw961=-3; end;
  if starw1<940 then do; wksw971=-3; end;
  if starw1<991 then do; wksw981=-3; end;
  if starw1<1044 then do; wksw991=-3; end;
end;

/* At this point the program loops through the same
code invalid start/stop date for each job (2-9), using the
same week numbers. Researchers who need to see the
complete code should contact NLS User Services. */

/* Section 2: Combine the Round 1 and Round 2
created variables for number of weeks worked at job x
during year. */

```

```

/* WKSJ945 is the Round1 created variable for 5th job
(i.e. 5th UID code) in 1994. WKS945 is the Round2
created variable of the same description. TWKS594 is
the variable that combines these two created variables
(of the same description). */

/* 7 employers for R1, 9 employers for R2 */
array r1uid (i) r1uid1-r1uid7;
array uid (i) uid1-uid9;

/* Arrange the jobs by year for each round */
array wksj80 (i) wks801-wks807; /*R1*/
array wksw80 (i) wksw801-wksw809; /*R2*/
/* and so on through wksj98 and wksw98 */
array wksw99 (i) wksw991-wksw999; /* R2 only */

/* Arrange the jobs by year for total created value */
array twks80 (i) twks801-twks809;
array twks81 (i) twks811-twks819;
/* and so on through: */
array twks99 (i) twks991-twks999;

/* Initialize total weeks count to zero */
do i=1 to 9;
  twks80=0; /* and so on through twks99=0 */
end;

/* Accounts for Round2 non-interview cases */
if e200=-5 then do;
  wksw801=-5; /* and so on through wksw809=-5 */
  /* and so on through wksw991 through wksw999 */
end;

/* Begin by updating the total created variable if the R2
variable is positive. We are only worried about R2 jobs
and R1 jobs that were worked in Round2. If a negative
value is given (don't know, refusal, or non-interview)
then total created variable will be that negative value. */

/* 1980 */
do i=1 to 9;
  if wksw80>0 then do; twks80=wksw80; end;
  if wksw80<0 then do; twks80=wksw80; end;
end;

/* At this point the program repeats the 1980 pattern
for each year through 1996 (wksw96 and twks96) */

/* 1997 */
/* Since the R1 interview occurred in 1997-98, need to
account for both R1 and R2 created variables. Begin
with negative conditions. */

do i=1 to 9;
  if wksw97<0 then do; twks97=wksw97; end;
  /* Update if Round2 created variable is positive */

```

Appendix 2: Employment Variable Creation

```
if wksw97>0 then do; twks97=wksw97; end;
end;

/* When matching jobs from R1 to R2, we only need
to worry about jobs in R2 with a UID beginning with
"97" since only those jobs could have been worked in
both rounds. */

array match1 (i) match11-match19;
/* and so on through match7 */

/* Define match14 (for example) as a dummy variable
that equals one when the first job on the Round1 UID
roster and the fourth job on the Round2 roster have the
same UID. */
do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
  match5=0; match6=0; match7=0;
end;

/* Determine if any UID from the first position in R1
matches with any UID in Round2 */
do over uid;
  if r1uid1>0 and uid>0 then do;
    if r1uid1=uid then do; match1=1; end;
  end;
end;

/* UID from position 2, 3, etc. in R1 matches R2 */
do over uid;
  if r1uid2>0 and uid>0 then do; /* repeat for r1uid3-
riuid7*/
    if r1uid2=uid then do; /* repeat for r1uid3-riuid7*/
      match2=1; /* repeat for match3-match7*/
    end;
  end;
end;

/* Now reassign total created variable if there is a job
was worked in both rounds. Using the match variable, a
job worked in both rounds will update the total created
variable. Need to add positivity condition on the weeks
worked values since some of the values equal -3. */

if dli_y=1997 then do;
  do over match1;
    if match1=1 then do;
      twks80=wks801;
      twks81=wks811;
      /* and so on through twks96 and wks961 */
    end;
  end;
end;

do over match1;
  if match1=1 and wksw97 ge 0 and wks971 ge 0
  then do; twks97=wks971+wksw97; end;
  if match1=1 and (wksw97<0 or wks971<0) then
  do; twks97=-3; end;
end;

do over match2; /*repeat for match3-match7*/
  /* At this point the program loops through the
above code using match2 and wksXX2, match3 and
wksXX3, and so on through match7 and wksXX7.
Users who need the complete code should contact
NLS User Services. */
end;

/* 1998 */
do i=1 to 9;
  if wksw98<0 then do; twks98=wksw98; end;
  /* Update if Round2 created variable is positive */
  if wksw98>0 then do; twks98=wksw98; end;
end;

if dli_y=1998 then do;
  do over match1;
    if match1=1 then do;
      twks80=wks801;
      twks81=wks811;
      /* and so on through twks97 and wks971 */
    end;
  end;
end;

do over match1;
  if match1=1 and wksw98=>0 and wks981=>0 then
  do; twks98=wks981+wksw98; end;
  if match1=1 and (wksw98<0 or wks981<0) then
  do; twks98=-3; end;
end;

do over match2; /*repeat for match3-match7*/
  /* At this point the program loops through the
above code using match2 and wksXX2, match3 and
wksXX3, and so on through match7 and wksXX7.
Users who need the complete code should contact
NLS User Services. */
end;

/* 1999: can only use Round2 created variables */
do i=1 to 9;
  if wksw99<0 then do; twks99=wksw99; end;
  if wksw99>0 then do; twks99=wksw99; end;
end;
```

TOTAL TENURE AT JOB #X AS OF THE SURVEY DATE

Variables Created: CV_WKSWK_JOB_DLI.01 – CV_WKSWK_JOB_DLI.09

Programs Used

This program uses **emp_begin.sas** as input (see the first page of this appendix for details).

This program creates a variable for each job calculating the total length of job tenure in weeks (excluding within-job gaps) since the respondent's 14th birthday. A variable is created for each potential job even if the respondent has no data for that job, with the default value set to zero (0). The most jobs held by any respondent as of round 2 was nine, so variables are created for nine jobs for each respondent.

```

/* Section 1: Create a variable for total tenure at each
job between the R1 and R2 interview dates. */

array starw (i) starw1-starw9;
array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;

/** Calculate cumulative weeks on individual jobs for
all years **/
tenure1=0;      tenure2=0;      tenure3=0;
tenure4=0;      tenure5=0;      tenure6=0;
tenure7=0;      tenure8=0;      tenure9=0;

do i=1 to 1044;
  if job1wks=1 then do; tenure1=tenure1+1; end;
  if job2wks=1 then do; tenure2=tenure2+1; end;
  if job3wks=1 then do; tenure3=tenure3+1; end;
  if job4wks=1 then do; tenure4=tenure4+1; end;
  if job5wks=1 then do; tenure5=tenure5+1; end;
  if job6wks=1 then do; tenure6=tenure6+1; end;
  if job7wks=1 then do; tenure7=tenure7+1; end;
  if job8wks=1 then do; tenure8=tenure8+1; end;
  if job9wks=1 then do; tenure9=tenure9+1; end;
end;

flag=0;

/* Invalid information for a certain week leads to
invalid created variables. */
do i=1 to 1044;
  if job1wks=-3 then do; tenure1=-3; flag=1; end;
  if job2wks=-3 then do; tenure2=-3; flag=2; end;
  if job3wks=-3 then do; tenure3=-3; flag=3; end;
  if job4wks=-3 then do; tenure4=-3; flag=4; end;
  if job5wks=-3 then do; tenure5=-3; flag=5; end;
  if job6wks=-3 then do; tenure6=-3; flag=6; end;
  if job7wks=-3 then do; tenure7=-3; flag=7; end;
  if job8wks=-3 then do; tenure8=-3; flag=8; end;
  if job9wks=-3 then do; tenure9=-3; flag=9; end;
end;

do i=1 to 1044;
  /* Start date invalid */
  if starw1<0 and starw1>-4 then do; tenure1=-3; end;
  if starw2<0 and starw2>-4 then do; tenure2=-3; end;
  if starw3<0 and starw3>-4 then do; tenure3=-3; end;
  if starw4<0 and starw4>-4 then do; tenure4=-3; end;
  if starw5<0 and starw5>-4 then do; tenure5=-3; end;
  if starw6<0 and starw6>-4 then do; tenure6=-3; end;
  if starw7<0 and starw7>-4 then do; tenure7=-3; end;
  if starw8<0 and starw8>-4 then do; tenure8=-3; end;
  if starw9<0 and starw9>-4 then do; tenure9=-3; end;
  /* Stop date invalid */
  if stopw1<0 and stopw1>-4 then do; tenure1=-3; end;
  if stopw2<0 and stopw2>-4 then do; tenure2=-3; end;
  if stopw3<0 and stopw3>-4 then do; tenure3=-3; end;
  if stopw4<0 and stopw4>-4 then do; tenure4=-3; end;
  if stopw5<0 and stopw5>-4 then do; tenure5=-3; end;
  if stopw6<0 and stopw6>-4 then do; tenure6=-3; end;
  if stopw7<0 and stopw7>-4 then do; tenure7=-3; end;
  if stopw8<0 and stopw8>-4 then do; tenure8=-3; end;
  if stopw9<0 and stopw9>-4 then do; tenure9=-3; end;
end;

if e200=0 then do;
  tenure1=-4;      tenure2=-4;      tenure3=-4;
  tenure4=-4;      tenure5=-4;      tenure6=-4;
  tenure7=-4;      tenure8=-4;      tenure9=-4;
end;

if e200=-3 then do;
  tenure1=-3;      tenure2=-3;      tenure3=-3;
  tenure4=-3;      tenure5=-3;      tenure6=-3;
  tenure7=-3;      tenure8=-3;      tenure9=-3;
end;

if e200=-5 then do;
  tenure1=-5;      tenure2=-5;      tenure3=-5;
  tenure4=-5;      tenure5=-5;      tenure6=-5;
  tenure7=-5;      tenure8=-5;      tenure9=-5;
end;

```

Appendix 2: Employment Variable Creation

```
/* Section 2: Calculate the total job tenure by
matching up the UID's from Round1 and Round2. */
```

```
/* When matching jobs from Round1 to Round2, we
only need to examine jobs in Round2 with a UID
beginning with "97" since only those jobs that could
have been worked in both rounds. For example,
match14 is a dummy variable that equals one when the
first job on the Round1 UID roster and the fourth job
on the Round2 roster have the same UID */
```

```
array r1uid (i) r1uid1-r1uid7;
array uid (i) uid1-uid9;
array match1 (i) match11-match19;
array match2 (i) match21-match29;
array match3 (i) match31-match39;
array match4 (i) match41-match49;
array match5 (i) match51-match59;
array match6 (i) match61-match69;
array match7 (i) match71-match79;
```

```
do i=1 to 9;
  match1=0; match2=0; match3=0; match4=0;
  match5=0; match6=0; match7=0;
end;
```

```
/* Determine if any UID from the first position in
Round1 matches with any UID in Round2 */
do over uid;
  if r1uid1>0 and uid>0 then do;
    if r1uid1=uid then do; match1=1; end;
  end;
end;
```

```
/* UID from position 2, 3, etc. in R1 matches R2 */
```

```
do over uid;
  if r1uid2>0 and uid>0 then do; /* repeat for r1uid3-
    r1uid7 */
    if r1uid2=uid then do; /*repeat for r1uid3-r1uid7*/
      match2=1; /* repeat for match3-match7 */
    end;
  end;
end;
```

```
/* Define totten as the total tenure of a job in weeks.
This includes jobs worked in both Round1 and
Round2 or jobs worked in only Round2. We are
considering jobs worked only in Round1 in this
program. */
array totten (i) totten1-totten9;
array tenure (i) tenure1-tenure9;
array r1ten (i) r1ten1-r1ten7;
```

```
do over totten; totten=0; end;
```

```
do over tenure; totten=tenure; end;
```

```
/* Now reassign total tenure if there is a job was worked
in both rounds. Using the match variable, a job worked
in both rounds will update the total tenure variable.
Need to add positivity condition on the tenure values
since some of the values equal -3. */
```

```
do over match1; /* repeat for match2-match7 */
  if match1=1 and tenure=>0 and r1ten1=>0 then do;
    totten=r1ten1+tenure; end;
  if match1=1 and -4<r1ten1<0 then do;
    totten=r1ten1; end;
end;
endsas;
```

TOTAL HOURS WORKED IN 19XX

Variables Created: CV_HOURS_WK_YR.80 – CV_HOURS_WK_YR.99

Programs Used

This program uses **bdate1.sas** and **emp_begin.sas** as input (see the first page of this appendix for details).

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
E200	YEMP-200	E38000F1-E38000F6	YEMP-38000F.01
E239011-E239019	YEMP-23901.01-.09	E381031, E381032	YEMP-38103.01, .02
E245011-E245016	YEMP-24501.01-.06	E381051, E381052	YEMP-38105.01, .02
E344021-E344027	YEMP-34402.01-.07	E599011-E599017	YEMP-59901.01-.07
E344031-E344035	YEMP-34403.01-.05	E880001-E880006	YEMP-88000.01-.06
E344281-E344285	YEMP-34428.01-.05	E885011-E885015	YEMP-88501.01-.05
E2261012-E2261032	YEMP-22610.01.02-.03.02	E984021-E984027	YEMP-98402.01-.07
E379021-E379028	YEMP-37902.01-.08	E984031-E984035	YEMP-98403.01-.05
E379041-E379047	YEMP-37904.01-.07	E984291-E984295	YEMP-98429.01-.05
E380001-E380009	YEMP-38000.01-.09		

This program calculates the number of hours worked by the respondent at all **employee-type** jobs in each calendar year. A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures.

```
/* Section 1: Creates a variable for total annual hours
worked at employee-type job during a given year using
Round 2 data. */
```

```
/* Organize hours for each job */
array starw (k) starw1-starw9;
array stopw (k) stopw1-stopw9;
```

```
/* shrs1 - starting hours   ehrl1 - ending hours */
array shrs (k) shrs1-shrs9;
array ehrl (k) ehrl1-ehrl9;
array hours (k) hours1-hours9;
array hrck (k) hrck1-hrck9;
```

```
/* To the right of the array statements are the true
ranges of the pulled variables. Values of variables
outside the ranges will be represented by dots. */
array e23901 (k) e239011-e239019; /* 1-9 */
array e24501 (k) e245011-e245019; /* 1-6 */
array e34402 (k) e344021-e344029; /* 1-7 */
array e34403 (k) e344031-e344039; /* 1,2,3,5 */
array e34428 (k) e344281-e344289; /* 1,2,3,5 */
array e37904 (k) e379041-e379049; /* 1-5,7 */
array e38000f (k) e38000f1-e38000f9; /* 1-6 */
array e38103 (k) e381031-e381039; /* 1,2 */
array e38105 (k) e381051-e381059; /* 1,2 */
array e59901 (k) e599011-e599019; /* 1-7 */
array e88000 (k) e880001-e880009; /* 1-6 */
array e88501 (k) e885011-e885019; /* 1-5 */
array e98402 (k) e984021-e984209; /* 1-7 */
array e98403 (k) e984031-e984039; /* 1,2,3,5 */
```

```
array e98429 (k) e984291-e984299; /* 1,2,3,5 */
array e100256 (k) e1002561-e1002569; /* 1,2,5 */
```

```
/* Define the number of hours per week worked at the
start date and the end date of a job. Question E59901
decides whether the job was listed on the Round 1
roster, for jobs under 13 weeks. Question E88000 also
decides whether the job was listed on the Round 1
roster, but only for jobs that last longer than 13 weeks.
The code below defines the starting hours and ending
hours per week at those jobs, including overtime. */
```

```
do k=1 to 9;
/* start hours jobs< 13 weeks */
if e23901>-4 then do; shrs=e23901; end;
if e23901>-4 and e24501>-4 then do;
shrs=e23901+e24501; end;
if e34402>-4 then do; shrs=e34402; end;
if e34403>-4 then do; shrs=e34403; end;
if e34403>-4 and e34428>-4 then do;
shrs=e34403+e34428; end;
/* end hours jobs< 13 weeks */
if e37904>-4 then do; ehrl=e37904; end;
/* end hours for jobs > 13 weeks */
if e38000f>-4 then do; ehrl=e38000f; end;
if e38103>-4 then do; ehrl=e38103; end;
if e38103>-4 and e38105>-4 then do;
ehrl=e38103+e38105; end;
/* jobs from dli that are less than 13 weeks long */
if e59901>-4 then do; shrs=e59901; end;
/* jobs from dli that are longer than 13 weeks */
```


Appendix 2: Employment Variable Creation

```

if e88000>-4 then do; shrs=e88000; end;
/* E88501 is an overtime question. */
if e88501>-4 and e88000>-4 then do;
  shrs=e88501+e88000; end;
if e98402>-4 then do; shrs=e98402; end;
if e98429>-4 then do; shrs=e98429; end;
if e98429>-4 and e98403>-4 then do;
  shrs=e98402+e98429; end;
if e100256>-4 then do; shrs=e100256; end;
end;

```

```

/* The code below decides which hours per week total
(starting or ending) will be used in determining hours
worked per year. We prefer using ending hours as the
measure, hours per week starts with a default number,
is updated if starting hours are available, and is updated
once more if ending hours are available. */

```

```

* Set default hours to zero;
do over hours; hours=0; hrck=0; end;

```

```

* Take starting hours if reported (eliminates -4's);
do over shrs;
  if shrs=>0 then do; hours=shrs; end;
/* hrck marks when invalid answers are given to the
hrs/wk questions */
  if -4<shrs<0 then do; hrck=shrs; end;
end;

```

```

* Write over if end hours reported;
do over ehrrs;
  if ehrrs=>0 then do; hours=ehrrs; end;
  if -4<ehrrs<0 then do; hrck=ehrrs; end;
end;

```

```

array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;

```

```

/** Calculate cumulative weeks on individual jobs for
each year **/

```

```

/* 1980 */ /* beginning here code loops for each year */
wks801=0; wks802=0; wks803=0; wks804=0;
  wks805=0; wks806=0; wks807=0; wks808=0;
  wks809=0;
ah801=0; ah802=0; ah803=0; ah804=0; ah805=0;
  ah806=0; ah807=0; ah808=0; ah809=0;

```

```

/* Update hours counter when respondent reports
employment in weeks 1 to 52. */

```

```

do i=1 to 52;
  if job1wks=1 then do; wks801=wks801+1; end;
  if job2wks=1 then do; wks802=wks802+1; end;
  if job3wks=1 then do; wks803=wks803+1; end;
  if job4wks=1 then do; wks804=wks804+1; end;
  if job5wks=1 then do; wks805=wks805+1; end;
  if job6wks=1 then do; wks806=wks806+1; end;
  if job7wks=1 then do; wks807=wks807+1; end;
  if job8wks=1 then do; wks808=wks808+1; end;
  if job9wks=1 then do; wks809=wks809+1; end;
end;

```

```

do i=1 to 52;
  if job1wks ne -3 and job2wks ne -3 and job3wks ne -
    3 and job4wks ne -3 and job5wks ne -3 and
    job6wks ne -3 and job7wks ne -3 and job8wks
    ne -3 and job9wks ne -3 then do;
    if hours1>0 then ah801=hours1*wks801;
    if hours2>0 then ah802=hours2*wks802;
    if hours3>0 then ah803=hours3*wks803;
    if hours4>0 then ah804=hours4*wks804;
    if hours5>0 then ah805=hours5*wks805;
    if hours6>0 then ah806=hours6*wks806;
    if hours7>0 then ah807=hours7*wks807;
    if hours8>0 then ah808=hours8*wks808;
    if hours9>0 then ah809=hours9*wks809;
  end;

```

```

  if job1wks=-3 or job2wks=-3 or job3wks=-3 or
    job4wks=-3 or job5wks=-3 or job6wks=-3 or
    job7wks=-3 or job8wks=-3 or job9wks=-3 then
    do;
    ah801=-3; ah802=-3; ah803=-3;
    ah804=-3; ah805=-3; ah806=-3;
    ah807=-3; ah808=-3; ah809=-3;
    goto exit1;
  end;
end; exit1;

```

```

if ah801=>0 and ah802=>0 and ah803=>0 and
  ah804=>0 and ah805=>0 and ah806=>0 and
  ah807=>0 and ah808=>0 and ah809=>0 then
  do;
  tothrs80=ah801+ah802+ah803+ah804+ah805
    +ah806+ah807+ah808+ah809;
end;

```

```

if ah801=-3 or ah802=-3 or ah803=-3 or ah804=-3 or
  ah805=-3 or ah806=-3 or ah807=-3 or ah808=-3
  or ah809=-3 then do;
  tothrs80=-3;
end;

```

```

if wks801>0 and hrck1<0 then tothrs80=hrck1;
/* and so on through wks809>0 and hrck9<0 */

```

```

/* At this point the program loops through the above
code for each year through 1999. This code is not

```

Appendix 2: Employment Variable Creation

shown here due to space constraints. Users who need the complete code should contact NLS User Services. */

```
/* Fill in -3's for cases where start/stop date is
   unknown/refused. */
```

```
do k=1 to 9;
/* start date invalid */
if starw<0 and starw>=-4 then do;
  if stopw>1 then do; tothrs80=-3; end;
  if stopw>52 then do; tothrs81=-3; end;
  if stopw>104 then do; tothrs82=-3; end;
  if stopw>156 then do; tothrs83=-3; end;
  if stopw>209 then do; tothrs84=-3; end;
  if stopw>261 then do; tothrs85=-3; end;
  if stopw>313 then do; tothrs86=-3; end;
  if stopw>365 then do; tothrs87=-3; end;
  if stopw>417 then do; tothrs88=-3; end;
  if stopw>470 then do; tothrs89=-3; end;
  if stopw>522 then do; tothrs90=-3; end;
  if stopw>574 then do; tothrs91=-3; end;
  if stopw>626 then do; tothrs92=-3; end;
  if stopw>678 then do; tothrs93=-3; end;
  if stopw>730 then do; tothrs94=-3; end;
  if stopw>783 then do; tothrs95=-3; end;
  if stopw>835 then do; tothrs96=-3; end;
  if stopw>887 then do; tothrs97=-3; end;
  if stopw>939 then do; tothrs98=-3; end;
  if stopw>991 then do; tothrs99=-3; end;
end;
```

```
/* stop date invalid */
if stopw<0 and stopw>=-4 then do;
  if starw<53 then do; tothrs80=-3; end;
  if starw<105 then do; tothrs81=-3; end;
  if starw<157 then do; tothrs82=-3; end;
  if starw<210 then do; tothrs83=-3; end;
  if starw<262 then do; tothrs84=-3; end;
  if starw<314 then do; tothrs85=-3; end;
  if starw<366 then do; tothrs86=-3; end;
  if starw<418 then do; tothrs87=-3; end;
  if starw<471 then do; tothrs88=-3; end;
  if starw<523 then do; tothrs89=-3; end;
  if starw<575 then do; tothrs90=-3; end;
  if starw<627 then do; tothrs91=-3; end;
  if starw<679 then do; tothrs92=-3; end;
  if starw<731 then do; tothrs93=-3; end;
  if starw<784 then do; tothrs94=-3; end;
  if starw<836 then do; tothrs95=-3; end;
  if starw<888 then do; tothrs96=-3; end;
  if starw<940 then do; tothrs97=-3; end;
  if starw<991 then do; tothrs98=-3; end;
  if starw<1044 then do; tothrs99=-3; end;
end;
end;
```

```
if e200=0 then do;
```

```
  tothrs80=-4; tothrs81=-4; tothrs82=-4; tothrs83=-4;
  tothrs84=-4; tothrs85=-4; tothrs86=-4; tothrs87=-4;
  tothrs88=-4; tothrs89=-4; tothrs90=-4; tothrs91=-4;
  tothrs92=-4; tothrs93=-4; tothrs94=-4; tothrs95=-4;
  tothrs96=-4; tothrs97=-4; tothrs98=-4; tothrs99=-4;
end;
```

```
if e200=-3 then do;
  tothrs80=-3; tothrs81=-3; tothrs82=-3; tothrs83=-3;
  tothrs84=-3; tothrs85=-3; tothrs86=-3; tothrs87=-3;
  tothrs88=-3; tothrs89=-3; tothrs90=-3; tothrs91=-3;
  tothrs92=-3; tothrs93=-3; tothrs94=-3; tothrs95=-3;
  tothrs96=-3; tothrs97=-3; tothrs98=-3; tothrs99=-3;
end;
```

```
if e200=-5 then do;
  tothrs80=-5; tothrs81=-5; tothrs82=-5; tothrs83=-5;
  tothrs84=-5; tothrs85=-5; tothrs86=-5; tothrs87=-5;
  tothrs88=-5; tothrs89=-5; tothrs90=-5; tothrs91=-5;
  tothrs92=-5; tothrs93=-5; tothrs94=-5; tothrs95=-5;
  tothrs96=-5; tothrs97=-5; tothrs98=-5; tothrs99=-5;
end;
```

/* Section 2: Sum the created variables for total hours worked in any job during a given year for R1 and R2 */

```
array tothrs (20) tothrs80-tothrs99; /* R2 variable */
array r1hrs (20) r1hrs80-r1hrs99; /* R1 variable */
array allhrs (20) allhrs80-allhrs99; /* Total from both
   rounds created variable */
```

```
/* Initialize the total created variable to be zero. */
do i=1 to 20; allhrs(i)=0; end;
```

/* Begin by splitting up periods where R1 and R2 exclusively collect hours worked information. Define the R1 interview year as the split. Any hours worked data collected before the R1 interview year should be independent of information collected in R2. */

```
do i=1 to 20;
  if r1hrs(i)>0 then do; allhrs(i)=r1hrs(i); end;
  if tothrs(i)>0 then do; allhrs(i)=tothrs(i); end;
  if r1hrs(i)>0 and tothrs(i)>0 then do;
    allhrs(i)=r1hrs(i)+tothrs(i); end;
end;
```

```
/* Define negative values for the total created var. */
do i=1 to 20;
  if -4<r1hrs(i)<0 then do; allhrs(i)=r1hrs(i); end;
  if tothrs(i)=-1 or tothrs(i)=-2 or tothrs(i)=-3 or
    tothrs(i)=-5 then do;
    allhrs(i)=tothrs(i); end;
end;
```

```
endsas;
```

TOTAL HOURS WORKED SINCE AGE 14

Variables Created: CV_HOURS_WK_EVER

Programs Used

This program uses `bdate1.sas` and `emp_begin.sas` as input (see the first page of this appendix for details).

Variables Used

This program uses the same variables as the previous program, "Total Hours Worked in 19xx."

This program calculates the number of hours worked by the respondent at all **employee-type** jobs since turning 14. A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures

```
/* Section 1: Create a variable for total hours worked at
employee-type jobs since age 14 using round 2 data. */
```

```
/* Organize hours for each job */
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
```

```
/* shrs1 - starting hours   ehrl - ending hours */
array shrs (i) shrs1-shrs9;
array ehrl (i) ehrl1-ehrl9;
array hours (i) hours1-hours9;
array hrck (i) hrck1-hrck9;
```

```
/* To the right of the array statements are the true
ranges of the pulled variables. Values of variables
outside the ranges will be represented by dots. */
array e23901 (i) e239011-e239019; /* 1-9 */
array e24501 (i) e245011-e245019; /* 1-6 */
array e34402 (i) e344021-e344029; /* 1-7 */
array e34403 (i) e344031-e344039; /* 1,2,3,5 */
array e34428 (i) e344281-e344289; /* 1,2,3,5 */
array e37904 (i) e379041-e379049; /* 1-5,7 */
array e38000f (i) e38000f1-e38000f9; /* 1-6 */
array e38103 (i) e381031-e381039; /* 1,2 */
array e38105 (i) e381051-e381059; /* 1,2 */
array e59901 (i) e599011-e599019; /* 1-7 */
array e88000 (i) e880001-e880009; /* 1-6 */
array e88501 (i) e885011-e885019; /* 1-5 */
array e98402 (i) e984021-e984029; /* 1-7 */
array e98403 (i) e984031-e984039; /* 1,2,3,5 */
array e98429 (i) e984291-e984299; /* 1,2,3,5 */
array e100256 (i) e1002561-e1002569; /* 1-3 */
```

```
/* Define the number of hours per week worked at the
start date and the end date of a job. Question E59901
decides whether the job was listed on the Round 1
roster, for jobs under 13 weeks. Question E88000 also
decides whether the job was listed on the Round 1
roster, but only for jobs that last longer than 13 weeks.
```

```
The code below defines the starting hours and ending
hours per week at those jobs, including overtime. */
```

```
do i=1 to 9;
/* starting hours for jobs shorter than 13 weeks */
if e23901>-4 then do; shrs=e23901; end;
if e23901>-4 and e24501>-4 then do;
shrs=e23901+e24501; end;
if e34402>-4 then do; shrs=e34402; end;
if e34403>-4 then do; shrs=e34403; end;
if e34403>-4 and e34428>-4 then do;
shrs=e34403+e34428; end;
/* ending hours for jobs less than 13 weeks long */
if e37904>-4 then do; ehrl=e37904; end;
/* ending hours for jobs longer than 13 weeks */
if e38000f>-4 then do; ehrl=e38000f; end;
if e38103>-4 then do; ehrl=e38103; end;
if e38103>-4 and e38105>-4 then do;
ehrl=e38103+e38105; end;
/* jobs from dli that are less than 13 weeks long */
if e59901>-4 then do; shrs=e59901; end;
/* jobs from dli that are longer than 13 weeks */
if e88000>-4 then do; shrs=e88000; end;
/* E88501 is an overtime question. */
if e88501>-4 and e88000>-4 then do;
shrs=e88501+e88000; end;
if e98402>-4 then do; shrs=e98402; end;
if e98429>-4 then do; shrs=e98429; end;
if e98429>-4 and e98403>-4 then do;
shrs=e98402+e98429; end;
if e100256>-4 then do; shrs=e100256; end;
end;
```

```
/* The code below decides which hours per week total
(starting or ending) will be used in determining hours
worked per year. We prefer using ending hours as the
measure, hours per week starts with a default number (-
16), is updating if starting hours are available, and is
updated once more if ending hours are available. */
```

Appendix 2: Employment Variable Creation

```

* Set default hours to zero (-16);
do over hours;
  hours=0; hrck=0;
end;

/* Take starting hours if reported (eliminates -4's); hrck
marks when invalid answers are given to the hrs/wk
questions */
do over shrs;
  if shrs=>0 then do; hours=shrs; end;
  if -4<shrs<0 then do; hrck=shrs; end;
end;

/* Write over if end hours reported */
do over ehrr;
  if ehrr=>0 then do; hours=ehrr; end;
  if -4<ehrr<0 then do; hrck=ehrr; end;
end;

array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;

/** Calculate cumulative weeks on individual jobs for
each year **/

wks1=0; wks2=0; wks3=0; wks4=0; wks5=0; wks6=0;
wks7=0; wks8=0; wks9=0;
thrs1=0; thrs2=0; thrs3=0; thrs4=0; thrs5=0; thrs6=0;
thrs7=0; thrs8=0; thrs9=0;

if 0<age14wk<1044 then do;
do i=age14wk to 1044;
  if job1wks=1 then do; wks1=wks1+1; end;
  if job2wks=1 then do; wks2=wks2+1; end;
  if job3wks=1 then do; wks3=wks3+1; end;
  if job4wks=1 then do; wks4=wks4+1; end;
  if job5wks=1 then do; wks5=wks5+1; end;
  if job6wks=1 then do; wks6=wks6+1; end;
  if job7wks=1 then do; wks7=wks7+1; end;
  if job8wks=1 then do; wks8=wks8+1; end;
  if job9wks=1 then do; wks9=wks9+1; end;
end;
end;

if 0<age14wk<1044 then do;
do i=age14wk to 1044;
  if job1wks ne -3 and job2wks ne -3 and job3wks ne -
3 and job4wks ne -3 and job5wks ne -3 and
job6wks ne -3 and job7wks ne -3 and
job8wks ne -3 and job9wks ne -3 then do;

thrs1=hours1*wks1; thrs2=hours2*wks2;
thrs3=hours3*wks3; thrs4=hours4*wks4;
thrs5=hours5*wks5; thrs6=hours6*wks6;
thrs7=hours7*wks7; thrs8=hours8*wks8;
thrs9=hours9*wks9;
end;
if job1wks=-3 or job2wks=-3 or job3wks=-3 or
job4wks=-3 or job5wks=-3 or job6wks=-3 or
job7wks=-3 or job8wks=-3 or
job9wks=-3 then do;
  thrs1=-3; r2hrs14=-3;
  goto exit1;
end;
end;
end; exit1:

if thrs1=>0 and thrs2=>0 and thrs3=>0 and thrs4=>0
and thrs5=>0 and thrs6=>0 and thrs7=>0 and
thrs8=>0 and
thrs9=>0 then do;
  r2hrs14=thrs1+thrs2+thrs3+thrs4+thrs5+thrs6+
thrs7+thrs8+thrs9;
end;

if thrs1=-3 or thrs2=-3 or thrs3=-3 or thrs4=-3 or
thrs5=-3 or thrs6=-3 or thrs7=-3 or thrs8=-3
or thrs9=-3 then do;
  r2hrs14=-3;
end;

/* hrck used when invalid hrs./wk answers are given */
if wks1>0 and hrck1<0 and r2hrs14>0 then
r2hrs14=hrck1;
/* and so on through wks9 and hrck9 */

/* Remove invalid start/stop dates */
do i=1 to 9;
if starw<0 and starw>-4 then do;
  if age14wk<53 and stopw>1 then r2hrs14=-3;
  if age14wk<105 and stopw>52 then r2hrs14=-3;
  if age14wk<157 and stopw>104 then r2hrs14=-3;
  if age14wk<210 and stopw>156 then r2hrs14=-3;
  if age14wk<262 and stopw>209 then r2hrs14=-3;
  if age14wk<314 and stopw>261 then r2hrs14=-3;
  if age14wk<366 and stopw>313 then r2hrs14=-3;
  if age14wk<418 and stopw>365 then r2hrs14=-3;
  if age14wk<471 and stopw>417 then r2hrs14=-3;
  if age14wk<523 and stopw>470 then r2hrs14=-3;
  if age14wk<575 and stopw>522 then r2hrs14=-3;
  if age14wk<627 and stopw>574 then r2hrs14=-3;
  if age14wk<679 and stopw>626 then r2hrs14=-3;
  if age14wk<731 and stopw>678 then r2hrs14=-3;
  if age14wk<784 and stopw>730 then r2hrs14=-3;
  if age14wk<836 and stopw>783 then r2hrs14=-3;
  if age14wk<888 and stopw>835 then r2hrs14=-3;
  if age14wk<940 and stopw>887 then r2hrs14=-3;
  if age14wk<991 and stopw>939 then r2hrs14=-3;

```

Appendix 2: Employment Variable Creation

```
if age14wk<1044 and stopw>991 then r2hrs14=-3;
end;

if stopw<0 and stopw>-4 then do;
  if age14wk<53 and starw<53 then r2hrs14=-3;
  if age14wk<105 and starw<105 then r2hrs14=-3;
  if age14wk<157 and starw<157 then r2hrs14=-3;
  if age14wk<210 and starw<210 then r2hrs14=-3;
  if age14wk<262 and starw<262 then r2hrs14=-3;
  if age14wk<314 and starw<314 then r2hrs14=-3;
  if age14wk<366 and starw<366 then r2hrs14=-3;
  if age14wk<418 and starw<418 then r2hrs14=-3;
  if age14wk<471 and starw<471 then r2hrs14=-3;
  if age14wk<523 and starw<523 then r2hrs14=-3;
  if age14wk<575 and starw<575 then r2hrs14=-3;
  if age14wk<627 and starw<627 then r2hrs14=-3;
  if age14wk<679 and starw<679 then r2hrs14=-3;
  if age14wk<731 and starw<731 then r2hrs14=-3;
  if age14wk<784 and starw<784 then r2hrs14=-3;
  if age14wk<836 and starw<836 then r2hrs14=-3;
  if age14wk<888 and starw<888 then r2hrs14=-3;
  if age14wk<940 and starw<940 then r2hrs14=-3;
  if age14wk<991 and starw<991 then r2hrs14=-3;
  if age14wk<1044 and starw<1044 then r2hrs14=-3;
end;
end;

if e200=-5 then r2hrs14=-5;
if e200=0 then r2hrs14=-4;
if e200=-3 then r2hrs14=-3;
```

```
/* Section 2: Combine variables for R1 and R2 */
```

```
/* Initialize created variable for both rounds to zero. */
allhrs14=0;
```

```
/* Account for non-interview cases */
```

```
if e200=-5 then r2hrs14=-5;
```

```
/* By the construction of the Round2 created variable,
we can simply add the two created variables from
Round1 and Round2 if they are both positive. If one is
positive and one is not, then the negative value will be
the total created variable for both rounds. If neither is
positive, then the total created variable for both rounds
will be zero. */
```

```
if tohrs14>0 then do; allhrs14=tohrs14; end;
```

```
if r2hrs14>0 then do; allhrs14=r2hrs14; end;
```

```
if tohrs14>0 and r2hrs14>0 then do;
```

```
  allhrs14=tohrs14+r2hrs14; end;
```

```
/* Define negative values for the created variables. */
```

```
if -4<tohrs14<0 then allhrs14=tohrs14;
```

```
if r2hrs14=-1 or r2hrs14=-2 or r2hrs14=-3 or r2hrs14=-
5 then allhrs14=r2hrs14;
```

```
endsas;
```

NUMBER OF JOBS HELD DURING 19XX

Variables Created: CV_TTL_JOBS_YR.80 – CV_TTL_JOBS_YR.99

Programs Used

This program uses `emp_begin.sas` as input (see the first page of this appendix for details).

This program calculates the number of employee-type jobs the respondent held during each calendar year. This variable is created only for respondents who have worked at least one week since age 14.

```
/* Section 1: Create a variable for each year for number
of jobs held in that year using R2 information. */
```

```
array job1wks (i) wk1_1-wk1_1044;
array job2wks (i) wk2_1-wk2_1044;
array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;
array job5wks (i) wk5_1-wk5_1044;
array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;
array job8wks (i) wk8_1-wk8_1044;
array job9wks (i) wk9_1-wk9_1044;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
```

```
/** Indicate if worked at least one week on a job in a
given year **/
```

```
/* 1980 */
```

```
job801=0;      job802=0;      job803=0;
job804=0;      job805=0;      job806=0;
job807=0;      job808=0;      job809=0;
```

```
do i=1 to 52;
  if job1wks=-3 then do; job801=-3; end;
  /* and so on through job9wks and job809 */
end;
```

```
do i=1 to 52;
  if job1wks=1 then do; job801=1; end;
  /* and so on through job9wks and job809 */
end;
```

```
if job801 ne -3 and job802 ne -3 and job803 ne -3 and
job804 ne -3 and job805 ne -3 and job806 ne -3 and
job807 ne -3 and job808 ne -3 and job809 ne -3 then
do;
```

```
  njobs80=sum(job801,job802,job803,job804,job805,
  job806,job807,job808,job809);
```

```
end;
```

```
if job801=-3 or job802=-3 or job803=-3 or job804=-3
or job805=-3 or job806=-3 or job807=-3 or job808=-3
or job809=-3 then do;
```

```
  njobs80=-3;
```

```
end;
```

```
/* At this point the program loops through and creates
a series of job variables for each year (1981-1999).
This code is not shown here due to space limitations.
Users needing the entire program should contact NLS
User Services. The variables and "do i" statements for
each year are as follows:
```

1981	job811-job819; njobs81	do i=53 to 104
1982	job821-job829; njobs82	do i=105 to 156
1983	job831-job839; njobs83	do i=157 to 209
1984	job841-job849; njobs84	do i=210 to 261
1985	job851-job859; njobs85	do i=262 to 313
1986	job861-job869; njobs86	do i=314 to 365
1987	job871-job879; njobs87	do i=366 to 417
1988	job881-job889; njobs88	do i=418 to 470
1989	job891-job899; njobs89	do i=471 to 522
1990	job901-job909; njobs90	do i=523 to 574
1991	job911-job919; njobs91	do i=575 to 626
1992	job921-job929; njobs92	do i=627 to 679
1993	job931-job939; njobs93	do i=679 to 730
1994	job941-job949; njobs94	do i=731 to 783
1995	job951-job959; njobs95	do i=784 to 835
1996	job961-job969; njobs96	do i=836 to 887
1997	job971-job979; njobs97	do i=888 to 939
1998	job981-job989; njobs98	do i=940 to 991
1999	job991-job999; njobs99	do i=992 to 1044

```
do i=1 to 9;
```

```
/* start date invalid */
```

```
if starw<0 and starw>-4 then do;
```

```
  if stopw>1 then do; njobs80=njobs80+1; end;
```

```
  if stopw>52 then do; njobs81=njobs81+1; end;
```

```
  if stopw>104 then do; njobs82=njobs82+1; end;
```

```
  if stopw>156 then do; njobs83=njobs83+1; end;
```

```
  if stopw>209 then do; njobs84=njobs84+1; end;
```

```
  if stopw>261 then do; njobs85=njobs85+1; end;
```

```
  if stopw>313 then do; njobs86=njobs86+1; end;
```

```
  if stopw>365 then do; njobs87=njobs87+1; end;
```

```
  if stopw>417 then do; njobs88=njobs88+1; end;
```

```
  if stopw>470 then do; njobs89=njobs89+1; end;
```

```
  if stopw>522 then do; njobs90=njobs90+1; end;
```

```
  if stopw>574 then do; njobs91=njobs91+1; end;
```

```
  if stopw>626 then do; njobs92=njobs92+1; end;
```

```
  if stopw>678 then do; njobs93=njobs93+1; end;
```

```
  if stopw>730 then do; njobs94=njobs94+1; end;
```

```
  if stopw>783 then do; njobs95=njobs95+1; end;
```

```
  if stopw>835 then do; njobs96=njobs96+1; end;
```

Appendix 2: Employment Variable Creation

```

if stopw>887 then do; njobs97=njobs97+1; end;
if stopw>939 and intwk>939 then do;
    njobs98=njobs98+1; end;
if stopw>991 and intwk>991 then do;
    njobs99=njobs99+1; end;
end;

```

*/*stop date invalid*/*

```

if stopw<0 and stopw>-4 then do;
if starw<53 then do; njobs80=njobs80+1; end;
if starw<105 then do; njobs81=njobs81+1; end;
if starw<157 then do; njobs82=njobs82+1; end;
if starw<210 then do; njobs83=njobs83+1; end;
if starw<262 then do; njobs84=njobs84+1; end;
if starw<314 then do; njobs85=njobs85+1; end;
if starw<366 then do; njobs86=njobs86+1; end;
if starw<418 then do; njobs87=njobs87+1; end;
if starw<471 then do; njobs88=njobs88+1; end;
if starw<523 then do; njobs89=njobs89+1; end;
if starw<575 then do; njobs90=njobs90+1; end;
if starw<627 then do; njobs91=njobs91+1; end;
if starw<679 then do; njobs92=njobs92+1; end;
if starw<731 then do; njobs93=njobs93+1; end;
if starw<784 then do; njobs94=njobs94+1; end;
if starw<836 then do; njobs95=njobs95+1; end;
if starw<888 then do; njobs96=njobs96+1; end;
if starw<940 then do; njobs97=njobs97+1; end;
if starw<991 and intwk>939 then do;
    njobs98=njobs98+1; end;
if starw<1044 and intwk>991 then do;
    njobs99=njobs99+1; end;
end;
end;

```

*** Include valid skips;

```

if e200=0 then do;
njobs80=-4; njobs81=-4; njobs82=-4; njobs83=-4;
njobs84=-4; njobs85=-4; njobs86=-4; njobs87=-4;
njobs88=-4; njobs89=-4; njobs90=-4; njobs91=-4;
njobs92=-4; njobs93=-4; njobs94=-4; njobs95=-4;
njobs96=-4; njobs97=-4; njobs98=-4; njobs99=-4;
end;

```

```

if e200=-3 then do;
njobs80=-3; njobs81=-3; njobs82=-3; njobs83=-3;
njobs84=-3; njobs85=-3; njobs86=-3; njobs87=-3;
njobs88=-3; njobs89=-3; njobs90=-3; njobs91=-3;
njobs92=-3; njobs93=-3; njobs94=-3; njobs95=-3;
njobs96=-3; njobs97=-3; njobs98=-3; njobs99=-3;
end;

```

```

if e200=-5 then do;
njobs80=-5; njobs81=-5; njobs82=-5; njobs83=-5;
njobs84=-5; njobs85=-5; njobs86=-5; njobs87=-5;
njobs88=-5; njobs89=-5; njobs90=-5; njobs91=-5;

```

```

njobs92=-5; njobs93=-5; njobs94=-5; njobs95=-5;
njobs96=-5; njobs97=-5; njobs98=-5; njobs99=-5;
end;

```

/ Section 2: Calculate the total number of jobs in a given year by matching up the UID's from Round1 and Round2.*/*

```

array njobs (20) njobs80-njobs99; /* R2 variable */
array r1job (20) r1job80-r1job99; /* R1 variable */
array alljob (20) alljob80-alljob99; /* Total from both rounds created variable */

```

/ Initialize created variable */*
do i=1 to 20; alljob(i)=0; end;

/ Begin by adding together the created variables from both rounds. Jobs that are double counted will be subtracted off later in the program. */*

```

do i=1 to 20;
    if r1job(i)>0 then do; alljob(i)=r1job(i); end;
    if njobs(i)>0 then do; alljob(i)=njobs(i); end;
    if r1job(i)>0 and njobs(i)>0 then do;
        alljob(i)=r1job(i)+njobs(i); end;
end;

```

/ Define negative values for total created variable. */*
do i=1 to 20;
if -4<r1job(i)<0 then alljob(i)=-3;
if njobs(i)=-1 or njobs(i)=-2 or njobs(i)=-3 or
njobs(i)=-5 then do;
alljob(i)=njobs(i); end;
end;

/ Jobs that are worked in 1997 or 1998 have the possibility of being double counted since they are counted in each Round's created variable programs. Here we compare the UID's from each round to see which jobs are double counted. For example, match14 is the dummy variable that equals one when the first job on the R1 UID roster and the fourth job on the R2 roster have the same UID */*

```

array r1uid (i) r1uid1-r1uid7;
array uid (i) uid1-uid9;
array match1 (i) match11-match19;
array match2 (i) match21-match29;
array match3 (i) match31-match39;
array match4 (i) match41-match49;
array match5 (i) match51-match59;
array match6 (i) match61-match69;
array match7 (i) match71-match79;

```

```

do i=1 to 9;
    match1=0; match2=0; match3=0; match4=0;
    match5=0; match6=0; match7=0; end;

```

```

/* Determine if any UID from the first position in
Round1 matches with any UID in Round2 */
do over uid;
  if r1uid1>0 and uid>0 then do;
    if r1uid1=uid then do; match1=1; end;
  end;
end;

/* Any UID from position 2, 3, etc. matches */
do over uid;
  if r1uid2>0 and uid>0 then do; /* repeat for r1uid3-
    r1uid7 */
    if r1uid2=uid then do; /* repeat for r1uid3-r1uid7 */
      match2=1; /* repeat for match3-match7 */
    end;
  end;
end;

/* Define "same97" as a counted variable that adds up
how many matching UIDs are in the respondent's UID
roster. This will be subtracted from the total number of
jobs created variable to avoid double counting the same
job. Begin by initializing "same" variables to zero. */
same97=0; same98=0;

```

```

array stopyr (i) stopyr1-stopyr9;
/* Consider respondents with a R1 int. date in 1997. */
if dli_y=1997 then do;
  do over match1;
    if match1=1 then do; same97=same97+1; end;
  end;
/* and so on through match7 */
end;

/* Consider respondents with a R1 int. date in 1998. */
if dli_y=1998 then do;
  do over match1;
    if match1=1 then do; same98=same98+1; end;
  end;
/* and so on through match7 */
end;

/* Now subtract the "same" count variable from the
total created variable computed above. */
if alljob97=>0 then do; alljob97=alljob97-same97; end;
if alljob98=>0 then do; alljob98=alljob98-same98; end;

endsas;

```


TOTAL NUMBER OF JOBS HELD SINCE AGE 14

Variables Created: CV_TTL_JOBS_EVER

Programs Used

This program uses `bdate1.sas` and `emp_begin.sas` as input (see the first page of this appendix for details).

This program calculates the total number of employee-type jobs held by the respondent since age 14. It is only created for respondents who have worked at least one week since age 14.

/* Section 1: Create a variable for number of jobs held since age 14 using R2 data. The default value is set to zero. */

```
array job1wks (i) wk1_1-wk1_1044;   array job2wks (i) wk2_1-wk2_1044;   array job3wks (i) wk3_1-wk3_1044;
array job4wks (i) wk4_1-wk4_1044;   array job5wks (i) wk5_1-wk5_1044;   array job6wks (i) wk6_1-wk6_1044;
array job7wks (i) wk7_1-wk7_1044;   array job8wks (i) wk8_1-wk8_1044;   array job9wks (i) wk9_1-wk9_1044;
```

/** Indicate if worked at least one week on a given job since age 14 **/

```
job1=0;      job2=0;      job3=0;
job4=0;      job5=0;      job6=0;
job7=0;      job8=0;      job9=0;

if age14wk>0 then do;
  do i=age14wk to 1044;
    if job1wks=1 then do; job1=1; end;   if job2wks=1 then do; job2=1; end;   if job3wks=1 then do; job3=1; end;
    if job4wks=1 then do; job4=1; end;   if job5wks=1 then do; job5=1; end;   if job6wks=1 then do; job6=1; end;
    if job7wks=1 then do; job7=1; end;   if job8wks=1 then do; job8=1; end;   if job9wks=1 then do; job9=1; end;
  end;
end;
```

```
njobs14=sum(job1,job2,job3,job4,job5,job6,job7,job8,job9);
```

/* Invalid start dates */

```
if starw1>-4 and starw1<0 and stopw1>0 and stopw1>age14wk then do; njobs14=njobs14+1; end;
if starw2>-4 and starw2<0 and stopw2>0 and stopw2>age14wk then do; njobs14=njobs14+1; end;
if starw3>-4 and starw3<0 and stopw3>0 and stopw3>age14wk then do; njobs14=njobs14+1; end;
if starw4>-4 and starw4<0 and stopw4>0 and stopw4>age14wk then do; njobs14=njobs14+1; end;
if starw5>-4 and starw5<0 and stopw5>0 and stopw5>age14wk then do; njobs14=njobs14+1; end;
if starw6>-4 and starw6<0 and stopw6>0 and stopw6>age14wk then do; njobs14=njobs14+1; end;
if starw7>-4 and starw7<0 and stopw7>0 and stopw7>age14wk then do; njobs14=njobs14+1; end;
if starw8>-4 and starw8<0 and stopw8>0 and stopw8>age14wk then do; njobs14=njobs14+1; end;
if starw9>-4 and starw9<0 and stopw9>0 and stopw9>age14wk then do; njobs14=njobs14+1; end;
```

```
if stopw1>-4 and stopw1<0 and starw1>0 and starw1=>age14wk then do; njobs14=njobs14+1; end;
if stopw2>-4 and stopw2<0 and starw2>0 and starw2=>age14wk then do; njobs14=njobs14+1; end;
if stopw3>-4 and stopw3<0 and starw3>0 and starw3=>age14wk then do; njobs14=njobs14+1; end;
if stopw4>-4 and stopw4<0 and starw4>0 and starw4=>age14wk then do; njobs14=njobs14+1; end;
if stopw5>-4 and stopw5<0 and starw5>0 and starw5=>age14wk then do; njobs14=njobs14+1; end;
if stopw6>-4 and stopw6<0 and starw6>0 and starw6=>age14wk then do; njobs14=njobs14+1; end;
if stopw7>-4 and stopw7<0 and starw7>0 and starw7=>age14wk then do; njobs14=njobs14+1; end;
if stopw8>-4 and stopw8<0 and starw8>0 and starw8=>age14wk then do; njobs14=njobs14+1; end;
if stopw9>-4 and stopw9<0 and starw9>0 and starw9=>age14wk then do; njobs14=njobs14+1; end;
```

/* Invalid stop dates */

```
if stopw1>-4 and stopw1<0 and starw1>0 and starw1<age14wk then do; njobs14=-3; end;
if stopw2>-4 and stopw2<0 and starw2>0 and starw2<age14wk then do; njobs14=-3; end;
if stopw3>-4 and stopw3<0 and starw3>0 and starw3<age14wk then do; njobs14=-3; end;
```

Appendix 2: Employment Variable Creation

```
if stopw4>-4 and stopw4<0 and starw4>0 and starw4<age14wk then do; njobs14=-3; end;
if stopw5>-4 and stopw5<0 and starw5>0 and starw5<age14wk then do; njobs14=-3; end;
if stopw6>-4 and stopw6<0 and starw6>0 and starw6<age14wk then do; njobs14=-3; end;
if stopw7>-4 and stopw7<0 and starw7>0 and starw7<age14wk then do; njobs14=-3; end;
if stopw8>-4 and stopw8<0 and starw8>0 and starw8<age14wk then do; njobs14=-3; end;
if stopw9>-4 and stopw9<0 and starw9>0 and starw9<age14wk then do; njobs14=-3; end;

if e200=0 then do; njobs14=-4; end;
if e200=-3 then do; njobs14=-3; end;
if e200=-5 then do; njobs14=-5; end;

/* Section 2: Calculate the total number of jobs since age 14 by matching up the UID's from R1 and R2. */

alljob14=0;          /* Initialize created variable */

/* Begin by adding together the created variables from both rounds. Jobs that are double counted will be subtracted
off later in the program. */
if r1jobs14>0 then do; alljob14=r1jobs14; end;
if njobs14>0 then do; alljob14=njobs14; end;
if r1jobs14>0 and njobs14>0 then do; alljob14=r1jobs14+njobs14; end;

/* Define negative values for the total created variable. */
if -4<r1jobs14<0 then alljob14=-3;
if njobs14=-2 or njobs14=-3 or njobs14=-5 then do; alljob14=njobs14; end;

/* Jobs that are worked in 1997 or 1998 have the possibility of being double counted. Here we will compare the
UID's to see which jobs are double counted. For example, match14 is a dummy variable that equals one when the
first job on the Round1 UID roster and the fourth job on the Round2 roster have the same UID. */
array r1uid (i) r1uid1-r1uid7;          array uid (i) uid1-uid9;
array match1 (i) match11-match19;      array match2 (i) match21-match29;
array match3 (i) match31-match39;      array match4 (i) match41-match49;
array match5 (i) match51-match59;      array match6 (i) match61-match69;
array match7 (i) match71-match79;
do i=1 to 9; match1=0; match2=0; match3=0; match4=0; match5=0; match6=0; match7=0; end;

do over uid;          /* If any UID from the first position in Round1 matches with any UID in Round2 */
  if r1uid1>0 and uid>0 then do; /* and so on for r1uid2-r1uid7 and match2-match7 */
    if r1uid1=uid then do; match1=1; end;
  end;
end;

/* Define "same" as a count variable adding the # of matching UIDs are in the respondent's roster and subtract from
the total # of jobs variable to avoid double counting the same job. Begin by initializing "same" variables to zero. */
same=0;

do over match1; if match1=1 then do; same=same+1; end; end;
do over match2; if match2=1 then do; same=same+1; end; end;
do over match3; if match3=1 then do; same=same+1; end; end;
do over match4; if match4=1 then do; same=same+1; end; end;
do over match5; if match5=1 then do; same=same+1; end; end;
do over match6; if match6=1 then do; same=same+1; end; end;
do over match7; if match7=1 then do; same=same+1; end; end;

/* Now subtract the "same" count variable from the total created variable computed above. */
if alljob14=>0 then do; alljob14=alljob14-same; end;

endsas;
```