

---

**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

/\*\*\*\*\* The first part of this program reads in raw start and stop dates for each job (max=7) and converts them into NLSY97 week numbers. Some start/stop DAYS and MONTHS have been imputed if missing. Imputed days are coded as '15' and months as '6'. \*\*\*\*\*/

```

array startm (i) starmo1-starmo10;
array startd (i) stardy1-stardy10;
array starty (i) staryr1-staryr10;
array stopm (i) stopmo1-stopmo10;
array stopd (i) stopdy1-stopdy10;
array stopy (i) stopyr1-stopyr10;

array sttdays (i) sttday1-sttday10;
/* total days that year from startdate (to Jan 1) */
array stpdays (i) stpday1-stpday10;
/* total days that year from stopdate (to Jan 1) */
array startwk (i) starw1-starw10;
array stopwk (i) stopw1-stopw10;
array srflag (i) srflg1-srflg10;
array spflag (i) spflg1-spflg10;

/**/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 starty;
if starty>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; end;
  if startm<=0 and startd<=0 then do;
    startm=6; startd=15; flag3=1; srflag=1; 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; end;
  if stopm<=0 and stopd<=0 then do;
    stopm=6; stopd=15; flag6=1; spflag=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=2 then sttdays=startd+31;
    if startm=3 then sttdays=startd+59;
    if startm=4 then sttdays=startd+90;
    if startm=5 then sttdays=startd+120;
    if startm=6 then sttdays=startd+151;
    if startm=7 then sttdays=startd+181;
    if startm=8 then sttdays=startd+212;
    if startm=9 then sttdays=startd+243;
    if startm=10 then sttdays=startd+273;
    if startm=11 then sttdays=startd+304;
    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=2 then sttdays=startd+31;
      if startm=3 then sttdays=startd+60;
      if startm=4 then sttdays=startd+91;
      if startm=5 then sttdays=startd+121;
      if startm=6 then sttdays=startd+152;
      if startm=7 then sttdays=startd+182;
    end;
  end;
end;

```

## Appendix 2: Employment Variable Creation

```

    if startm=8 then sttdays=startd+213;
    if startm=9 then sttdays=startd+244;
    if startm=10 then sttdays=startd+274;
    if startm=11 then sttdays=startd+305;
    if startm=12 then sttdays=startd+335;
end;
end;
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***/
/**The basic formula for this is:
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;
    end;
    if starty<0 and starty>-4 then do;
        startwk=starty; 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;

```

## Appendix 2: Employment Variable Creation

```

    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;
end;
if stopy<0 and stopy>-4 then do;
    stopwk=stopy; end;
end;

```

/\*\* This intermediate part of the program reads in raw data on within-job gaps and converts the gap dates into NLSY97 week numbers. The converted data has been suitably transformed to account for work within the week of the gap start/stop. \*\*/

/\*\* 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 (i) BGDY1_1-BGDY1_12;
array bgmo (i) BGMO1_1-BGMO1_12;
array bgyr (i) BGYR1_1-BGYR1_12;
array egdy (i) EGDY1_1-EGDY1_12;
array egmo (i) EGMO1_1-EGMO1_12;
array egyr (i) EGYR1_1-EGYR1_12;

array bdays (i) bday1_1-bday1_12;
/* begin day of job1_gap# (internal calc.)*/
array edays (i) eday1_1-eday1_12;
/* end day of job1_gap# (internal calc.)*/
array bweek (i) bgap1_1-bgap1_12;
/* begin week of job1_gap# (created) */
array eweek (i) egap1_1-egap1_12;
/* end week of job1_gap# (created) */
array bflag (i) bflg1_1-bflg1_12;
array eflag (i) eflg1_1-eflg1_12;

```

/\* 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;
        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;
        end;
    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;
    end;
end;

```

```
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;
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 /**/
/**/ The basic formula for this is:
weekno=endweek{specific year}+
ceil[(totdays+{# of days remaining in DEC})/7]*/

/** Note: 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 do;
    bweek=ceil((bdays+2)/7);
    bweek=bweek+1; end;
  if bgyr=1981 then do;
    bweek=52+ceil((bdays+4)/7);
    bweek=bweek+1; end;
  if bgyr=1982 then do;
    bweek=104+ceil((bdays+5)/7);
    bweek=bweek+1; end;
  if bgyr=1983 then do;
    bweek=156+ceil((bdays+6)/7);
    bweek=bweek+1; end;
  if bgyr=1984 then do;
    bweek=209+ceil((bdays)/7);
    bweek=bweek+1; end;
  if bgyr=1985 then do;
    bweek=261+ceil((bdays+2)/7);
    bweek=bweek+1; end;
  if bgyr=1986 then do;
    bweek=313+ceil((bdays+3)/7);
    bweek=bweek+1; end;
  if bgyr=1987 then do;
    bweek=365+ceil((bdays+4)/7);
    bweek=bweek+1; end;
  if bgyr=1988 then do;
    bweek=417+ceil((bdays+5)/7);
```

```

    bweek=bweek+1; end;
if bgyr=1989 then do;
    bweek=470+ceil((bdays)/7);
    bweek=bweek+1; end;
if bgyr=1990 then do;
    bweek=522+ceil((bdays+1)/7);
    bweek=bweek+1; end;
if bgyr=1991 then do;
    bweek=574+ceil((bdays+2)/7);
    bweek=bweek+1; end;
if bgyr=1992 then do;
    bweek=626+ceil((bdays+3)/7);
    bweek=bweek+1; end;
if bgyr=1993 then do;
    bweek=678+ceil((bdays+5)/7);
    bweek=bweek+1; end;
if bgyr=1994 then do;
    bweek=730+ceil((bdays+6)/7);
    bweek=bweek+1; end;
if bgyr=1995 then do;
    bweek=783+ceil((bdays)/7);
    bweek=bweek+1; end;
if bgyr=1996 then do;
    bweek=835+ceil((bdays+1)/7);
    bweek=bweek+1; end;
if bgyr=1997 then do;
    bweek=887+ceil((bdays+3)/7);
    bweek=bweek+1; end;
if bgyr=1998 then do;
    bweek=939+ceil((bdays+4)/7);
    bweek=bweek+1; end;
end;
end;

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

    /***At this point the program loops through the
    same code for gaps in jobs 2-6 (there are no gaps
    for job 7). This code is eliminated here due to
    space restrictions. Users who need access to the
    entire program should contact NLS User
    Services.***/

```

/\*\*\*\*The program next calculates the total number of weeks worked on each job.\*\*\*\*/

```
array job1wks (i) wk1_1-wk1_992;
array job2wks (i) wk2_1-wk2_992;
array job3wks (i) wk3_1-wk3_992;
array job4wks (i) wk4_1-wk4_992;
array job5wks (i) wk5_1-wk5_992;
array job6wks (i) wk6_1-wk6_992;
array job7wks (i) wk7_1-wk7_992;
array alljobs (i) wks1-wks992;
array starw (i) starw1-starw7;
array stopw (i) stopw1-stopw7;
```

/\* Default Settings \*/

```
do i=1 to 992;
  job1wks=0; job2wks=0; job3wks=0;
  job4wks=0; job5wks=0; job6wks=0;
  job7wks=0;
end;
```

/\*\*\*\*\*\* Total Weeks Worked on Job 1 \*\*\*\*\*/

```
if starw1>0 & stopw1>0 then do; /* [1] */
  do i=(starw1) to (stopw1);
    job1wks=1;
  end;
```

```
** 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 1 on job 1 - begin 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;
```

```
** 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;
```

```
** Remove gap 2 on job 1 - begin gap date bad;
if bgfl1_2=1 & egap1_2>0 then do;
  do i=(starw1) to (egap1_2);
    job1wks=-3; gpfl1_2=1; end;
end;
```

```
** Remove gap 2 on job 1 - end gap date bad;
if bgap1_2>0 & egfl1_2=1 then do;
  do i=(bgap1_2) to (stopw1);
    job1wks=-3; gpfl1_2=1; end;
end;
```

```
** Remove gap 2 on job 1 - both gap dates bad;
if bgfl1_2=1 & egfl1_2=1 then do;
  do i=(starw1) to (stopw1);
    job1wks=-3; gpfl1_2=1; end;
end;
```

```
** Remove gap 3 on job 1;
if bgap1_3>0 & egap1_3>0 then do;
  do i=(bgap1_3) to (egap1_3);
    job1wks=0; end;
end;
```

```
** Remove gap 3 on job 1 - begin gap date bad;
if bgfl1_3=1 & egap1_3>0 then do;
  do i=(starw1) to (egap1_3);
    job1wks=-3; gpfl1_3=1; end;
end;
```

```
** Remove gap 3 on job 1 - end gap date bad;
if bgap1_3>0 & egfl1_3=1 then do;
  do i=(bgap1_3) to (stopw1);
    job1wks=-3; gpfl1_3=1; end;
end;
```

```
** Remove gap 3 on job 1 - both gap dates bad;
if bgfl1_3=1 & egfl1_3=1 then do;
  do i=(starw1) to (stopw1);
    job1wks=-3; gpfl1_3=1; end;
end;
```

```
** Remove gap 4 on job 1;
if bgap1_4>0 & egap1_4>0 then do;
  do i=(bgap1_4) to (egap1_4);
    job1wks=0; end;
end;
```

```
** Remove gap 4 on job 1 - begin gap date bad;
if bgfl1_4=1 & egap1_4>0 then do;
  do i=(starw1) to (egap1_4);
```



## Appendix 2: Employment Variable Creation

```

    job1wks=-3; gpfl1_4=1; end;
end;
** Remove gap 4 on job 1 - end gap date bad;
if bgap1_4>0 & egfl1_4=1 then do;
    do i=(bgap1_4) to (stopw1);
        job1wks=-3; gpfl1_4=1; end;
    end;
** Remove gap 4 on job 1 - both gap dates bad;
if bgfl1_4=1 & egfl1_4=1 then do;
    do i=(starw1) to (stopw1);
        job1wks=-3; gpfl1_4=1; end;
    end;

** Remove gap 5 on job 1;
if bgap1_5>0 & egap1_5>0 then do;
    do i=(bgap1_5) to (egap1_5);
        job1wks=0; end;
    end;

** Remove gap 5 on job 1 - begin gap date bad;
if bgfl1_5=1 & egap1_5>0 then do;
    do i=(starw1) to (egap1_5);
        job1wks=-3; gpfl1_5=1; end;
    end;
** Remove gap 5 on job 1 - end gap date bad;
if bgap1_5>0 & egfl1_5=1 then do;
    do i=(bgap1_5) to (stopw1);
        job1wks=-3; gpfl1_5=1; end;
    end;
** Remove gap 5 on job 1 - both gap dates bad;
if bgfl1_5=1 & egfl1_5=1 then do;
    do i=(starw1) to (stopw1);
        job1wks=-3; gpfl1_5=1; end;
    end;

** Remove gap 6 on job 1;
if bgap1_6>0 & egap1_6>0 then do;
    do i=(bgap1_6) to (egap1_6);
        job1wks=0; end;
    end;

** Remove gap 6 on job 1 - begin gap date bad;
if bgfl1_6=1 & egap1_6>0 then do;
    do i=(starw1) to (egap1_6);
        job1wks=-3; gpfl1_6=1; end;
    end;
** Remove gap 6 on job 1 - end gap date bad;
if bgap1_6>0 & egfl1_6=1 then do;
    do i=(bgap1_6) to (stopw1);
        job1wks=-3; gpfl1_6=1; end;
    end;
** Remove gap 6 on job 1 - both gap dates bad;

if bgfl1_6=1 & egfl1_6=1 then do;
    do i=(starw1) to (stopw1);
        job1wks=-3; gpfl1_6=1; end;
    end;

** Remove gap 7 on job 1;
if bgap1_7>0 & egap1_7>0 then do;
    do i=(bgap1_7) to (egap1_7);
        job1wks=0; end;
    end;

** Remove gap 7 on job 1 - begin gap date bad;
if bgfl1_7=1 & egap1_7>0 then do;
    do i=(starw1) to (egap1_7);
        job1wks=-3; gpfl1_7=1; end;
    end;
** Remove gap 7 on job 1 - end gap date bad;
if bgap1_7>0 & egfl1_7=1 then do;
    do i=(bgap1_7) to (stopw1);
        job1wks=-3; gpfl1_7=1; end;
    end;
** Remove gap 7 on job 1 - both gap dates bad;
if bgfl1_7=1 & egfl1_7=1 then do;
    do i=(starw1) to (stopw1);
        job1wks=-3; gpfl1_7=1; end;
    end;

** Remove gap 8 on job 1;
if bgap1_8>0 & egap1_8>0 then do;
    do i=(bgap1_8) to (egap1_8);
        job1wks=0; end;
    end;

** Remove gap 8 on job 1 - begin gap date bad;
if bgfl1_8=1 & egap1_8>0 then do;
    do i=(starw1) to (egap1_8);
        job1wks=-3; gpfl1_8=1; end;
    end;
** Remove gap 8 on job 1 - end gap date bad;
if bgap1_8>0 & egfl1_8=1 then do;
    do i=(bgap1_8) to (stopw1);
        job1wks=-3; gpfl1_8=1; end;
    end;
** Remove gap 8 on job 1 - both gap dates bad;
if bgfl1_8=1 & egfl1_8=1 then do;
    do i=(starw1) to (stopw1);
        job1wks=-3; gpfl1_8=1; end;
    end;

** Remove gap 9 on job 1;
if bgap1_9>0 & egap1_9>0 then do;
    do i=(bgap1_9) to (egap1_9);

```

```

        job1wks=0; end;
    end;

    ** Remove gap 9 on job 1 - begin gap date bad;
    if bgfl1_9=1 & egap1_9>0 then do;
        do i=(starw1) to (egap1_9);
            job1wks=-3; gpfl1_9=1; end;
        end;
    ** Remove gap 9 on job 1 - end gap date bad;
    if bgap1_9>0 & egfl1_9=1 then do;
        do i=(bgap1_9) to (stopw1);
            job1wks=-3; gpfl1_9=1; end;
        end;
    ** Remove gap 9 on job 1 - both gap dates bad;
    if bgfl1_9=1 & egfl1_9=1 then do;
        do i=(starw1) to (stopw1);
            job1wks=-3; gpfl1_9=1; end;
        end;

    ** Remove gap 10 on job 1;
    if bgap1_10>0 & egap1_10>0 then do;
        do i=(bgap1_10) to (egap1_10);
            job1wks=0; end;
        end;

    ** Remove gap 10 on job 1 - begin gap date bad;
    if bgfl1_10=1 & egap1_10>0 then do;
        do i=(starw1) to (egap1_10);
            job1wks=-3; gpfl1_10=1; end;
        end;
    ** Remove gap 10 on job 1 - end gap date bad;
    if bgap1_10>0 & egfl1_10=1 then do;
        do i=(bgap1_10) to (stopw1);
            job1wks=-3; gpfl1_10=1; end;
        end;
    ** Remove gap 10 on job 1 - both gap dates bad;
    if bgfl1_10=1 & egfl1_10=1 then do;

        do i=(starw1) to (stopw1);
            job1wks=-3; gpfl1_10=1; end;
        end;

    end; /* [1] */

    /***** At this point the program repeats the
    same code to calculate total weeks worked for jobs
    2-7. The only difference is that the number of
    gaps is smaller for some of these jobs. Users who
    need to see the complete code should contact NLS
    User Services. *****/

    endsas;

```

### BDATE1.SAS

```

/* This program creates birthwk (default=0 birth date < 12/30/79) and age14wk (default=9999 if age 14
date > 1/1/99). */
/** Calculate Age 14 year **/
AGE14YR=birthyr+14;

/** Convert Age 14 Birthdate to week # **/
/** Convert age14 month and 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;

```

Appendix 2: Employment Variable Creation

```

    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****/
/**** The basic formula for this is:
weekno=endweek{specific year}+
ceil[(totdays+{# of days remaining in DEC})/7] */

/* Default age 14 week = 9999 */
age14wk=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;
    if age14yr=1998 then do;
        age14wk=939+ceil((bbdays+4)/7); end;
end;

/**** Calulate Age 16 year ****/
AGE16YR=birthyr+16;

/**** Convert Age 16 birthdate to week # **/
/****Convert age16 month and 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 age16yr=1980 or age16yr=1984 or
    age16yr=1988 or age16yr=1992 or
    age16yr=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;

```

## Appendix 2: Employment Variable Creation

```

if birthmo=7 then bbdays=birthy+182;
if birthmo=8 then bbdays=birthy+213;
if birthmo=9 then bbdays=birthy+244;
if birthmo=10 then bbdays=birthy+274;
if birthmo=11 then bbdays=birthy+305;
if birthmo=12 then bbdays=birthy+335;
end;
end;

/**Convert days into week numbers***/
/** The basic formula for this is:
weekno=endweek{specific year}+
ceil[(totdays+{# of days remaining in DEC})/7] */

/* Default age 16 week = 9999 */
age16wk=9999;

if age16yr>0 and bbdays>0 then do;
if age16yr=1980 then do;
age16wk=ceil((bbdays+2)/7); end;
if age16yr=1981 then do;
age16wk=52+ceil((bbdays+4)/7); end;
if age16yr=1982 then do;
age16wk=104+ceil((bbdays+5)/7); end;
if age16yr=1983 then do;
age16wk=156+ceil((bbdays+6)/7); end;
if age16yr=1984 then do;
age16wk=209+ceil((bbdays)/7); end;
if age16yr=1985 then do;
age16wk=261+ceil((bbdays+2)/7); end;
if age16yr=1986 then do;
age16wk=313+ceil((bbdays+3)/7); end;
if age16yr=1987 then do;
age16wk=365+ceil((bbdays+4)/7); end;
if age16yr=1988 then do;
age16wk=417+ceil((bbdays+5)/7); end;
if age16yr=1989 then do;
age16wk=470+ceil((bbdays)/7); end;
if age16yr=1990 then do;
age16wk=522+ceil((bbdays+1)/7); end;
if age16yr=1991 then do;
age16wk=574+ceil((bbdays+2)/7); end;
if age16yr=1992 then do;
age16wk=626+ceil((bbdays+3)/7); end;
if age16yr=1993 then do;
age16wk=678+ceil((bbdays+5)/7); end;
if age16yr=1994 then do;
age16wk=730+ceil((bbdays+6)/7); end;
if age16yr=1995 then do;
age16wk=783+ceil((bbdays)/7); end;
if age16yr=1996 then do;
age16wk=835+ceil((bbdays+1)/7); end;

if age16yr=1997 then do;
age16wk=887+ceil((bbdays+3)/7); end;
if age16yr=1998 then do;
age16wk=939+ceil((bbdays+4)/7); end;
end;

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

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

endsas;

```

YOUTH'S EMPLOYMENT STATUS RECODE

Variables Created: CV\_ESR  
CV\_ESR\_COLLAPSED

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD	Name in Program	Question Name on CD
C2100	YCPS-2100	C87001	YCPS-87001	C15800	YCPS-15800
C2300	YCPS-2300	C87002	YCPS-87002	C15900	YCPS-15900
C2400	YCPS-2400	C96001	YCPS-96001	C176001	YCPS-176001
C2600	YCPS-2600	C96002	YCPS-96002	C176002	YCPS-176002
C2800	YCPS-2800	C10400	YCPS-10400	C186001	YCPS-186001
C2900	YCPS-2900	C106001	YCPS-106001	C186002	YCPS-186002
C3100	YCPS-3100	C106002	YCPS-106002	C23600	YCPS-23600
C78001	YCPS-78001	C11000	YCPS-11000	C23700	YCPS-23700
C78002	YCPS-78002	C14000	YCPS-14000	C27300	YCPS-27300
C79001	YCPS-79001	C14100	YCPS-14100	C30700	YCPS-30700
C79002	YCPS-79002	C14200	YCPS-14200	C30800	YCPS-30800
C82001	YCPS-82001	C15200	YCPS-15200	PUBID	PUBID
C82002	YCPS-82002	C15300	YCPS-15300		

Codes for Created Variable

Employment Status Recode

- 1 = at work
- 2 = with job, not at work
- 3 = layoff, unemployed
- 4 = look for work, unemployed
- 5 = not in labor force, retired
- 6 = not in labor force, disabled (long-term physical or mental illness, lasting six months or longer)
- 7 = not in labor force, other
- 8 = in active Armed Forces

Collapsed Employment Status Recode

- 1 = employed
- 2 = unemployed
- 3 = not in the labor force
- 4 = in active Armed Forces

This program creates two employment status variables: employment status recode (ESR) and collapsed ESR. These variables are created only for respondents age 15 or older as of the interview date. Note that enlistment in the active Armed Forces takes precedence over concurrent civilian employment.

```
/* First create the ESR variable: */
/*Initialize esr*/
```

```
esr=-1;
if C2100=-2 then esr=-2;
if C2100=-3 then esr=-3;
if C2100=0 or C2100=-4 then esr=-4;
```

```
/*Define some arrays to be used later in the program at various times */
```

```
array C7800 C78001 C78002;
array C7900 C79001 C79002;
array C8200 C82001 C82002;
array C8700 C87001 C87002;
array C9600 C96001 C96002;
array C10600 C106001 C106002;
array C17600 C176001 C176002;
array C18600 C186001 C186002;
```

## Appendix 2: Employment Variable Creation

---

```
/*Create hrck6 and hrck7 for when the hh owns a farm and when the hh does not own a farm*/

hrck6=3;
if (C15200=0 and C15900=0) and (C2800=0 or C2800=-2 or C2800=-1 or C2800=-3) then hrck6=1;
if (C15200=0 and C15900=0) then hrck6=2;

if (C10400 ge 0 and C11000 ge 0) then hruslt=C10400+C11000;
if (C10400 ge 0 and C11000 lt 0) then hruslt=C10400;
if (C11000 ge 0 and C10400 lt 0) then hruslt=C11000;

if (C15200 ge 0 and C15900 ge 0) then hractt=C15200+C15900;
if (C15200 ge 0 and C15900 lt 0) then hractt=C15200;
if (C15900 ge 0 and C15200 lt 0) then hractt=C15900;

hrck7=5;
if (C2900=0 or C2900=-2 or C2900=-1 or C2900=-3) and ((C15200 lt 15 and C15200 ge 0) or C15200 eq -2) then
    hrck7=1;
if (C2900=0 or C2900=-2 or C2900=-1 or C2900=-3) and (C15200 ge 15) then hrck7=2;
if (hruslt ge 35) and (hractt lt 35) and (C15200 ge 0) then hrck7=3;
if (hruslt ge 35) and (hractt lt 35) and (C15900 ge 0) then hrck7=3;
if (C14000=1 and hractt lt 35) and (C14100=1 or C14100=2 or C14100=3) then hrck7=4;

/*If hh owns a farm and worked last week or was absent from work:*/
    if C2100=1 and C2600=1 then do;
        /* If at work:*/
            do I=1 to 2;
                if (hrck7 eq 2 or hrck7 eq 3 or hrck7 eq 4 or hrck7 eq 5) then do;
                    if C2900=1 or (C2900=0 and C3100=1 and C10600(I)>=15) then esr=1;
                end;
            end;
        /* If absent:*/
            do I=1 to 2;
                if (C2900=0 and C3100=0 and C8200(I)=1 and C9600(I)=1) or (C2900=0 and
                    C3100=0 and C8200(I)=1 and C9600(I)=0 and C10600(I)>=15) then esr=2;
            end;
        end;

/*If hh does not own a farm and worked or was absent from work during last week*/
    /* If at work:*/
        if C2800=1 and hrck6 eq 3 and C2100=1 and C2600 le 1 then esr=1;
    /* If absent:*/
        do I=1 to 2;
            if C2800=0 and C8200(I)=1 and C2100=1 and C2600 le 1 then esr=2;
        end;

/*If laid off during last week*/
    do I=1 to 2;
        if (C2100=1 and C8700(I)=1 and C17600(I)=1) then esr=3;
    end;

/*If unemployed and looking for a job during last week*/
    do I=1 to 2;
        if (C2100=1 and C18600(I)=1 and C23600=1)
            or (C2100=1 and C18600(I)=1 and C23600=0 and C23700=1)
            then esr=4;
    end;
```

## Appendix 2: Employment Variable Creation

---

```
/*If not in the labor force and retired */
  if C2100=1 and (C2800=2 or C2900=3 or C30800=5) then esr=5;

/*If not in the labor force and disabled */
  /* If hh owns a farm:*/
  do I=1 to 2;
    if C2100=1 and C2600=1 then do;
      if C2900=3 and (C7800(I)=1 or C7900(I)=1) then esr=6;
    end;
  end;
  /* If hh does not own a farm:*/
  do I=1 to 2;
    if C2100=1 and C2600=0 then do;
      if C2800=3 and (C7800(I)=1 or C7900(I)=1) then esr=6;
    end;
  end;

if esr=-1 then do;

/*If not in the labor force and "other" category:*/
  if C2100=1 and (C30700=3 or C30800=1 or C30800=2 or C30800=3 or C30800=4 or C30800=6) then
    esr=7;

/*If in the army during last week*/
  if C2300=1 and C2400=1 then esr=8;

end;

/* Create the Collapsed ESR code */
  if esr=1 or esr=2 then collesr=1;
  if esr=3 or esr=4 then collesr=2;
  if esr=5 or esr=6 or esr=7 then collesr=3;
  if esr=8 then collesr=4;
  if esr=-4 then collesr=-4;
  if esr=-1 then collesr=-1;
  if esr=-3 then collesr=-3;
  if esr=-2 then collesr=-2;

endsas;
```

**HOURLY RATE OF PAY**

**Variables Created:** CV\_HRLY\_PAY

**Variables Used**

Name in Program	Question Name on CD	Name in Program	Question Name on CD
E1710001-07	YEMP-17100.01-.07	E5510001-05	YEMP-55100.01-.05
E1900001-07	YEMP-19000.01-.07	E5520001-04	YEMP-55200.01-.04
E1920001-07	YEMP-19200.01-.07	E5980001-07	YEMP-59800.01-.07
E2290001-06	YEMP-22900.01-.06	E6200001-07	YEMP-62000.01-.07
E2300001-07	YEMP-23000.01-.07	E6570001-04	YEMP-65700.01-.04
E2320001, 02	YEMP-23200.01, .02	E6580001-07	YEMP-65800.01-.07
E2380001, 02	YEMP-23800.01, .02	E6600001-05	YEMP-66000.01-.05
E2390001, 02	YEMP-23900.01, .02	E6670001	YEMP-66700.01
E2690002	YEMP-26900.02	E7200001-07	YEMP-72000.01-.07
E2920001-07	YEMP-29200.01-.07	E7620001-05	YEMP-76200.01-.05
E3340001-05	YEMP-33400.01-.05	E7630001, 02	YEMP-76300.01, .02
E3350001-04	YEMP-33500.01-.04	E7640001, 02	YEMP-76400.01, .02
E3360001-05	YEMP-33600.01-.05	E7800001	YEMP-18000.01
E3530001-03	YEMP-35300.01-.03	E7810001, 02	YEMP-78100.01, .02
E3560001-05	YEMP-35600.01-.05	E7840001-04	YEMP-78400.01-.04
E3610001-04	YEMP-36100.01-.04	E7900001-04	YEMP-79000.01-.04
E3620001-05	YEMP-36200.01-.05	E7920001-07	YEMP-79200.01-.07
E3780001-07	YEMP-37800.01-.07	E8220001-07	YEMP-82200.01-.07
E3790001-06	YEMP-37900.01-.06	E8310001-05	YEMP-83100.01-.05
E3820001-06	YEMP-38200.01-.06	E8680001-03	YEMP-86800.01-.03
E4190001-06	YEMP-41900.01-.06	E8690001-05	YEMP-86900.01-.05
E4200001-06	YEMP-42000.01-.06	E8710001	YEMP-87100.01
E4220003	YEMP-42200.03	E9310001-05	YEMP-93100.01-.05
E4290001	YEMP-42900.01	E9730001-05	YEMP-97300.01-.05
E4820001-06	YEMP-48200.01-.06	E9740001-04	YEMP-97400.01-.04
E5240001-05	YEMP-52400.01-.05	E9750001, 02	YEMP-97500.01, .02
E5250001-05	YEMP-52500.01-.05	E9830001, 02	YEMP-98300.01, .02
E5260001-05	YEMP-52600.01-.05	E9920001, 02	YEMP-99200.01, .02
E5340001, 02	YEMP-53400.01, .02	E9950001-04	YEMP-99500.01-.04
E5420001, 02	YEMP-54200.01, .02	E100e01-04	YEMP-100000.01-.04
E5430001, 02	YEMP-54300.01, .02	E1001e01-04	YEMP-100100.01-.04
E5460001, 05	YEMP-54600.01, .05	PUBID	PUBID

**Codes for Created Variable**

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

This program created 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.

The hourly rate of pay variable uses the following 4 reporting categories:

1. The start wage for youths under the age of 16 as of the survey date who only report a start wage;
2. The stop wage for youths under the age of 16 as of the survey date who report both a start wage and a stop wage;
3. The start wage for youths age 16 or older as of the survey date who only report a start wage; and
4. The stop wage for youths age 16 or older as of the survey date who report both a start wage and a stop wage.

/\*Set up the hourly wage for youths who report their wage hourly. Both the hourly wage reported in the non-overtime and the overtime loops can be used without any further calculations.\*/

ARRAY E17100 E1710001 E1710002 E1710003 E1710004 E1710005 E1710006 E1710007 ;



## Appendix 2: Employment Variable Creation

---

ARRAY E37800 E3780001 E3780002 E3780003 E3780004 E3780005 E3780006 E3780007 ;  
ARRAY E79200 E7920001 E7920002 E7920003 E7920004 E7920005 E7920006 E7920007 ;  
ARRAY E19000 E1900001 E1900002 E1900003 E1900004 E1900005 E1900006 E1900007 ;  
ARRAY E19200 E1920001 E1920002 E1920003 E1920004 E1920005 E1920006 E1920007 ;  
ARRAY E22900 E2290001 E2290002 E2290003 E2290004 E2290005 E2290006 E2290007 ;  
ARRAY E23000 E2300001 E2300002 E2300003 E2300004 E2300005 E2300006 E2300007 ;  
ARRAY E23200 E2320001 E2320002 E2320003 E2320004 E2320005 E2320006 E2320007 ;  
ARRAY E33400 E3340001 E3340002 E3340003 E3340004 E3340005 E3340006 E3340007 ;  
ARRAY E33500 E3350001 E3350002 E3350003 E3350004 E3350005 E3350006 E3350007 ;  
ARRAY E33600 E3360001 E3360002 E3360003 E3360004 E3360005 E3360006 E3360007 ;  
ARRAY E23900 E2390001 E2390002 E2390003 E2390004 E2390005 E2390006 E2390007 ;  
ARRAY E34400 E3440001 E3440002 E3440003 E3440004 E3440005 E3440006 E3440007 ;  
ARRAY E35300 E3530001 E3530002 E3530003 E3530004 E3530005 E3530006 E3530007 ;  
ARRAY E35600 E3560001 E3560002 E3560003 E3560004 E3560005 E3560006 E3560007 ;  
ARRAY E36100 E3610001 E3610002 E3610003 E3610004 E3610005 E3610006 E3610007 ;  
ARRAY E36200 E3620001 E3620002 E3620003 E3620004 E3620005 E3620006 E3620007 ;  
ARRAY E37900 E3790001 E3790002 E3790003 E3790004 E3790005 E3790006 E3790007 ;  
ARRAY E38200 E3820001 E3820002 E3820003 E3820004 E3820005 E3820006 E3820007 ;  
ARRAY E41900 E4190001 E4190002 E4190003 E4190004 E4190005 E4190006 E4190007 ;  
ARRAY E42000 E4200001 E4200002 E4200003 E4200004 E4200005 E4200006 E4200007 ;  
ARRAY E42200 E4220001 E4220002 E4220003 E4220004 E4220005 E4220006 E4220007 ;  
ARRAY E42900 E4290001 E4290002 E4290003 E4290004 E4290005 E4290006 E4290007 ;  
ARRAY E52400 E5240001 E5240002 E5240003 E5240004 E5240005 E5240006 E5240007 ;  
ARRAY E52500 E5250001 E5250002 E5250003 E5250004 E5250005 E5250006 E5250007 ;  
ARRAY E52600 E5260001 E5260002 E5260003 E5260004 E5260005 E5260006 E5260007 ;  
ARRAY E53400 E5340001 E5340002 E5340003 E5340004 E5340005 E5340006 E5340007 ;  
ARRAY E54200 E5420001 E5420002 E5420003 E5420004 E5420005 E5420006 E5420007 ;  
ARRAY E54300 E5430001 E5430002 E5430003 E5430004 E5430005 E5430006 E5430007 ;  
ARRAY E54600 E5460001 E5460002 E5460003 E5460004 E5460005 E5460006 E5460007 ;  
ARRAY E55100 E5510001 E5510002 E5510003 E5510004 E5510005 E5510006 E5510007 ;  
ARRAY E55200 E5520001 E5520002 E5520003 E5520004 E5520005 E5520006 E5520007 ;  
ARRAY E59800 E5980001 E5980002 E5980003 E5980004 E5980005 E5980006 E5980007 ;  
ARRAY E62000 E6200001 E6200002 E6200003 E6200004 E6200005 E6200006 E6200007 ;  
ARRAY E65700 E6570001 E6570002 E6570003 E6570004 E6570005 E6570006 E6570007 ;  
ARRAY E65800 E6580001 E6580002 E6580003 E6580004 E6580005 E6580006 E6580007 ;  
ARRAY E66000 E6600001 E6600002 E6600003 E6600004 E6600005 E6600006 E6600007 ;  
ARRAY E66700 E6670001 E6670002 E6670003 E6670004 E6670005 E6670006 E6670007 ;  
ARRAY E76200 E7620001 E7620002 E7620003 E7620004 E7620005 E7620006 E7620007 ;  
ARRAY E76300 E7630001 E7630002 E7630003 E7630004 E7630005 E7630006 E7630007 ;  
ARRAY E76400 E7640001 E7640002 E7640003 E7640004 E7640005 E7640006 E7640007 ;  
ARRAY E77200 E7720001 E7720002 E7720003 E7720004 E7720005 E7720006 E7720007 ;  
ARRAY E78000 E7800001 E7800002 E7800003 E7800004 E7800005 E7800006 E7800007 ;  
ARRAY E78100 E7810001 E7810002 E7810003 E7810004 E7810005 E7810006 E7810007 ;  
ARRAY E78400 E7840001 E7840002 E7840003 E7840004 E7840005 E7840006 E7840007 ;  
ARRAY E78900 E7890001 E7890002 E7890003 E7890004 E7890005 E7890006 E7890007 ;  
ARRAY E79000 E7900001 E7900002 E7900003 E7900004 E7900005 E7900006 E7900007 ;  
ARRAY E82200 E8220001 E8220002 E8220003 E8220004 E8220005 E8220006 E8220007 ;  
ARRAY E83100 E8310001 E8310002 E8310003 E8310004 E8310005 E8310006 E8310007 ;  
ARRAY E86800 E8680001 E8680002 E8680003 E8680004 E8680005 E8680006 E8680007 ;  
ARRAY E86900 E8690001 E8690002 E8690003 E8690004 E8690005 E8690006 E8690007 ;  
ARRAY E87100 E8710001 E8710002 E8710003 E8710004 E8710005 E8710006 E8710007 ;  
ARRAY E87800 E8780001 E8780002 E8780003 E8780004 E8780005 E8780006 E8780007 ;  
ARRAY E97300 E9730001 E9730002 E9730003 E9730004 E9730005 E9730006 E9730007 ;  
ARRAY E97400 E9740001 E9740002 E9740003 E9740004 E9740005 E9740006 E9740007 ;  
ARRAY E97500 E9750001 E9750002 E9750003 E9750004 E9750005 E9750006 E9750007 ;  
ARRAY E98300 E9830001 E9830002 E9830003 E9830004 E9830005 E9830006 E9830007 ;  
ARRAY E99200 E9920001 E9920002 E9920003 E9920004 E9920005 E9920006 E9920007 ;

## Appendix 2: Employment Variable Creation

---

```
ARRAY E99500 E9950001 E9950002 E9950003 E9950004 E9950005 E9950006 E9950007 ;
ARRAY E100e E100e01 E100e02 E100e03 E100e04 E100e05 E100e06 E100e07;
ARRAY E1001e E1001e01 E1001e02 E1001e03 E1001e04 E1001e05 E1001e06 E1001e07;
ARRAY E26800[7] E2680001-E2680007;
ARRAY E26900[7] E2690001-E2690007;
ARRAY E29200[7] E2920001-E2920007;
ARRAY E48200[7] E4820001-E4820007;
ARRAY E72000[7] E7200001-E7200007;
ARRAY E93100[7] E9310001-E9310007;
```

**\*\*\*\*\*SECTION 1: START WAGES REPORTED HOURLY, WEEKLY, BIWEEKLY, MONTHLY, AND ANNUALLY\*\*\*\*\*/**

**/\*Part A—Hourly:** Set up the hourly wage for youths who report an hourly wage\*/

```
ARRAY HRWAGE HRWAGE01 HRWAGE02 HRWAGE03 HRWAGE04 HRWAGE05 HRWAGE06
      HRWAGE07;
DO I=1 TO 7;
HRWAGE[I]=0;
```

**/\*under 16, start wage, no overtime\*/**

```
IF (E17100[I]=1 AND E37800[I]=0) then DO;
  IF (E22900[I]>0) then HRWAGE[I]=E22900[I];
  IF (E23000[I]>0) then HRWAGE[I]=E23000[I];
  IF (E22900[I]=-3 or E23000[I]=-3) then HRWAGE[I]=E23200[I];
  IF (E22900[I]=-2 or E23000[I]=-2) 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;
```

**/\*16+, start wage, no overtime\*/**

```
IF (E17100[I]=0 AND E79200[I]=0) then DO;
  IF (E65700[I]>0) then HRWAGE[I]=E65700[I];
  IF (E65800[I]>0) then HRWAGE[I]=E65800[I];
  IF (E65700[I]=-3 or E65800[I]=-3) then HRWAGE[I]=E66000[I];
  IF (E65700[I]=-2 or E65800[I]=-2) then HRWAGE[I]=E66000[I];
  IF (E65700[I]=-1 or E65800[I]=-1) then HRWAGE[I]=-1;
  IF E66700[I]>0 then HRWAGE[I]=E66700[I];
END;
```

END;

**\*\*\*\*\*Part B—Weekly:** Set up the hourly wage for youths who report a weekly, daily, or other time unit in E-19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in weekly units.\*\*\*

```
ARRAY WEEKLY WEEKLY01 WEEKLY02 WEEKLY03 WEEKLY04 WEEKLY05 WEEKLY06 WEEKLY07;
DO I=1 TO 7;
WEEKLY[I]=0;
```

**/\*Under 16, weekly start wage divided by the number of hours worked per week\*/**

**/\*no overtime\*/**

```
IF (E17100[I]=1 AND E37800[I]=0) then DO;
  IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E33400[I]>0
    and E19000[I]>0 then WEEKLY[I]=(E33400[I]/E19000[I]);
```

**/\*estimated\*/**

```
IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E33600[I]>0
  and E19000[I]>0 and E33400[I]>-3 then WEEKLY[I]=(E33600[I]/E19000[I]);
```

**/\*interviewer corrected\*/**

## Appendix 2: Employment Variable Creation

```
IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E34400[I]>0
and E19000[I]>0 and E33400[I]>-3 then WEEKLY[I]=(E34400[I]/E19000[I]);
/*respondent corrected*/
IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E35300[I]>0
and E19000[I]>0 and E33400[I]>-3 then WEEKLY[I]=(E35300[I]/E19000[I]);
/*missing values*/
IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and (E19000[I]=-1
or E19000[I]=-2 or E19000[I]=-3) then WEEKLY[I]=E19000[I];
IF (E19000[I]=0) then WEEKLY[I]=-6;
END;

/*16+, weekly start wage divided by the number of hours worked per week*/
/*no overtime*/
ELSE IF E17100[I]=0 AND E79200[I]=0 then DO;
IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E76200[I]>0
and E59800[I]>0 then WEEKLY[I]=(E76200[I]/E59800[I]);
/*estimated*/
IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E76400[I]>0
and E59800[I]>0 and E76200[I]>-3 then WEEKLY[I]=(E76400[I]/E59800[I]);
/*interviewer corrected*/
IF (E62000[I]=2 | E38200[I]=3 | E62000[I]=7 | E38200[I] GE 9) and E77200[I]>0
and E59800[I]>0 and E76200[I]>-3 then WEEKLY[I]=(E77200[I]/E59800[I]);
/*respondent corrected*/
IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E78100[I]>0
and E59800[I]>0 and E76200[I]>-3 then WEEKLY[I]=(E78100[I]/E59800[I]);
/*missing values*/
IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and (E59800[I]=-1
or E59800[I]=-2 or E59800[I]=-3) then WEEKLY[I]=E59800[I];
IF (E62000[I]=2 | E62000[I]=3 | E62000[I] GE 9) and E59800[I]=0 then WEEKLY[I]=-6;
END;
END;

/*****Part C—Biweekly: Set up the hourly wage for youths who report a biweekly time unit in E-19200, E-38200,
E-62000, or E-83100. All of these youths were asked to report their wage in biweekly units. */

ARRAY BIWKLY BIWKLY01 BIWKLY02 BIWKLY03 BIWKLY04 BIWKLY05 BIWKLY06 BIWKLY07;
DO I=1 TO 7;
BIWKLY[I]=0;

/*under 16, bi-wkly start wage divided by 2 times the number of hours worked per week */
/*no overtime*/
IF E17100[I]=1 AND E37800[I]=0 then DO;
IF (E19200[I]=4) and E33400[I]>0 and E19000[I]>0 then BIWKLY[I]=(E33400[I]/(2*E19000[I]));
/*estimated*/
IF (E19200[I]=4) and E33600[I]>0 and E19000[I]>0 and E33400[I]>-3
then BIWKLY[I]=(E33600[I]/(2*E19000[I]));
/*interviewer corrected*/
IF (E19200[I]=4) and E34400[I]>0 and E19000[I]>0 and E33400[I]>-3
then BIWKLY[I]=(E34400[I]/(2*E19000[I]));
/*respondent corrected*/
IF (E19200[I]=4) and E35300[I]>0 and E19000[I]>0 and E33400[I]>-3
then BIWKLY[I]=(E35300[I]/(2*E19000[I]));
/*missing values*/
IF (E19200[I]=4) and (E19000[I]=-1 or E19000[I]=-2 or E19000[I]=-3) then BIWKLY[I]=E19000[I];
IF (E19200[I]=4) and (E19000[I]=0) then BIWKLY[I]=-6;
END;
```

## Appendix 2: Employment Variable Creation

```
/*16+, bi-wkly start wage divided by 2 times the number of hours worked per week*/
/*no overtime*/
ELSE IF E17100[I]=0 AND E79200[I]=0 then DO;
  IF (E62000[I]=4) and E76200[I]>0 and E59800[I]>0 then BIWKLY[I]=(E76200[I]/(2*E59800[I]));
/*estimated*/
  IF (E62000[I]=4) and E76400[I]>0 and E59800[I]>0 and E76200[I]>-3
    then BIWKLY[I]=(E76400[I]/(2*E59800[I]));
/*interviewer corrected*/
  IF (E62000[I]=4) and E77200[I]>0 and E59800[I]>0 and E76200[I]>-3
    then BIWKLY[I]=(E77200[I]/(2*E59800[I]));
/*respondent corrected*/
  IF (E62000[I]=4) and E78100[I]>0 and E59800[I]>0 and E76200[I]>-3
    then BIWKLY[I]=(E78100[I]/(2*E59800[I]));
/*missing values*/
  IF (E62000[I]=4) and (E59800[I]=-1 or E59800[I]=-2 or E59800[I]=-3) then BIWKLY[I]=E59800[I];
  IF (E62000[I]=4) and E59800[I]=0 then BIWKLY[I]=-6;
END;
END;

/*****Part D—Monthly: Set up the hourly wage for youths who report a monthly or semi-monthly time unit in E-
19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in monthly units.*/

ARRAY MONTH MONTH01 MONTH02 MONTH03 MONTH04 MONTH05 MONTH06 MONTH07;
DO I=1 TO 7;
MONTH[I]=0;

/*under 16, month start wage divided by 4.3 times the number of hours worked per week*/
/*no overtime*/
IF E17100[I]=1 AND E37800[I]=0 then DO;
  IF (E19200[I]=5 | E19200[I]=8) and E33400[I]>0 and E19000[I]>0
    then MONTH[I]=(E33400[I]/(4.3*E19000[I]));
/*estimated*/
  IF (E19200[I]=5 | E19200[I]=8) and E33600[I]>0 and E19000[I]>0 and E33400[I]>-3
    then MONTH[I]=(E33600[I]/(4.3*E19000[I]));
/*interviewer corrected*/
  IF (E19200[I]=5 | E19200[I]=8) and E34400[I]>0 and E19000[I]>0 and E33400[I]>-3
    then MONTH[I]=(E34400[I]/(4.3*E19000[I]));
/*respondent corrected*/
  IF (E19200[I]=5 | E19200[I]=8) and E35300[I]>0 and E19000[I]>0 and E33400[I]>-3
    then MONTH[I]=(E35300[I]/(4.3*E19000[I]));
/*missing values*/
  IF (E19200[I]=5 | E19200[I]=8) and (E19000[I]=-1 or E19000[I]=-2 or E19000[I]=-3)
    then MONTH[I]=E19000[I];
  IF (E19200[I]=5 | E19200[I]=8) and E19000[I]=0 then MONTH[I]=-6;
END;

/*16+, month start wage divided by 4.3 times the number of hours worked per week*/
/*no overtime*/
ELSE IF E17100[I]=0 AND E79200[I]=0 then DO;
  IF (E62000[I]=5 | E62000[I]=8) and E76200[I]>0 and E59800[I]>0
    then MONTH[I]=(E76200[I]/(4.3*E59800[I]));
/*estimated*/
  IF (E62000[I]=5 | E62000[I]=8) and E76400[I]>0 and E59800[I]>0 and E76200[I]>-3
    then MONTH[I]=(E76400[I]/(4.3*E59800[I]));
/*interviewer corrected*/
  IF (E62000[I]=5 | E38200[I]=8) and E77200[I]>0 and E59800[I]>0 and E76200[I]>-3
    then MONTH[I]=(E77200[I]/(4.3*E59800[I]));
```

## Appendix 2: Employment Variable Creation

---

```
/*respondent corrected*/
  IF (E62000[I]=5 | E62000[I]=8) and E78100[I]>0 and E59800[I]>0 and E76200[I]>-3
    then MONTH[I]=(E78100[I]/(4.3*E59800[I]));
/*missing values*/
  IF (E62000[I]=5 | E62000[I]=8) and (E59800[I]=-1 or E59800[I]=-2 or E59800[I]=-3)
    then MONTH[I]=E59800[I];
  IF (E62000[I]=5 | E62000[I]=8) and E59800[I]=0 then MONTH[I]=-6;
END;
END;

/*****Part E—Annual: Set up the hourly wage for youths who report an annual time unit in E-19200, E-38200, E-
62000, or E-83100. All of these youths were asked to report their wage in annual units.*/

ARRAY ANNUAL ANNUAL01 ANNUAL02 ANNUAL03 ANNUAL04 ANNUAL05 ANNUAL06 ANNUAL07;
DO I=1 TO 7;
ANNUAL[I]=0;

/*under 16, annual start wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
  IF E17100[I]=1 AND E37800[I]=0 then DO;
    IF E19200[I]=6 and E19000[I]>0 and E33400[I]>0 and E35600[I]>0
      then ANNUAL[I]=(E33400[I]/(E35600[I]*E19000[I]));
/*estimated*/
    IF (E19200[I]=6) and E33600[I]>0 and E19000[I]>0 and E33400[I]>-3
      then ANNUAL[I]=(E33600[I]/(E35600[I]*E19000[I]));
/*interviewer corrected*/
    IF (E19200[I]=6) and E34400[I]>0 and E19000[I]>0 and E33400[I]>-3
      then ANNUAL[I]=(E34400[I]/(E35600[I]*E19000[I]));
/*respondent corrected*/
    IF (E19200[I]=6) and E35300[I]>0 and E19000[I]>0 and E33400[I]>-3
      then ANNUAL[I]=(E35300[I]/(E35600[I]*E19000[I]));
/*missing values*/
    IF E19200[I]=6 and (E19000[I]=-1 or E19000[I]=-2 or E19000[I]=-3) then ANNUAL[I]=E19000[I];
    IF E19200[I]=6 and (E35600[I]=-1 or E35600[I]=-2 or E35600[I]=-3) then ANNUAL[I]=E35600[I];
    IF E19200[I]=6 and E19000[I]=0 then ANNUAL[I]=-6;
  END;

/*16+, annual start wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
  ELSE IF E17100[I]=0 AND E79200[I]=0 then DO;
    IF E62000[I]=6 and E76200[I]>0 and E78400[I]>0 and E59800[I]>0
      then ANNUAL[I]=(E76200[I]/(E78400[I]*E59800[I]));
/*estimated*/
    IF (E62000[I]=6) and E76400[I]>0 and E59800[I]>0 and E76200[I]>-3
      then ANNUAL[I]=(E76400[I]/(E78400[I]*E59800[I]));
/*interviewer corrected*/
    IF (E62000[I]=6) and E77200[I]>0 and E59800[I]>0 and E76200[I]>-3
      then ANNUAL[I]=(E77200[I]/(E78400[I]*E59800[I]));
/*respondent corrected*/
    IF (E62000[I]=6) and E78100[I]>0 and E59800[I]>0 and E76200[I]>-3
      then ANNUAL[I]=(E78100[I]/(E78400[I]*E59800[I]));
/*missing values*/
    IF E62000[I]=6 and (E59800[I]=-1 or E59800[I]=-2 or E59800[I]=-3) then ANNUAL[I]=E59800[I];
    IF E62000[I]=6 and (E78400[I]=-1 or E78400[I]=-2 or E78400[I]=-3) then ANNUAL[I]=E78400[I];
    IF E62000[I]=6 and E59800[I]=0 then ANNUAL[I]=-6;
  END;
END;
```

## Appendix 2: Employment Variable Creation

---

/\*\*\*\*\*\*Part F—Other: Set up the hourly wage for youths who report a non-hourly time unit. Both the hourly wage reported in the non-overtime and the overtime loops can be used without any further calculations.\*/

```
ARRAY OTHER OTHER01 OTHER02 OTHER03 OTHER04 OTHER05 OTHER06 OTHER07;  
DO I=1 TO 7;  
OTHER[I]=0;
```

```
/*under 16, start wage*/
```

```
IF E17100[I]=1 AND E37800[I]=0 then DO;  
  IF (E36100[I]>0) then OTHER[I]=E36100[I];  
  IF (E36200[I]>0) then OTHER[I]=E36200[I];  
END;
```

```
/*16+, start wage*/
```

```
ELSE IF E17100[I]=0 AND E79200[I]=0 then DO;  
  IF (E78900[I]>0) then OTHER[I]=E78900[I];  
  IF (E79000[I]>0) then OTHER[I]=E79000[I];  
END;  
END;
```

```
/******Create the Hourly Rate of Pay*****/
```

```
ARRAY HRWG HRWG01 HRWG02 HRWG03 HRWG04 HRWG05 HRWG06 HRWG07;  
DO I=1 TO 7;  
HRWG[I]=0;
```

```
IF HRWAGE[I] EQ -4 OR WEEKLY[I] EQ -4 OR BIWKLY[I] EQ -4 OR MONTH[I] EQ -4 OR ANNUAL[I]  
  EQ -4 OR OTHER[I] EQ -4 THEN HRWG[I]=-4;  
IF HRWAGE[I] EQ -3 OR WEEKLY[I] EQ -3 OR BIWKLY[I] EQ -3 OR MONTH[I] EQ -3 OR ANNUAL[I]  
  EQ -3 OR OTHER[I] EQ -3 THEN HRWG[I]=-3;  
IF HRWAGE[I] EQ -2 OR WEEKLY[I] EQ -2 OR BIWKLY[I] EQ -2 OR MONTH[I] EQ -2 OR ANNUAL[I]  
  EQ -2 OR OTHER[I] EQ -2 THEN HRWG[I]=-2;  
IF HRWAGE[I] EQ -1 OR WEEKLY[I] EQ -1 OR BIWKLY[I] EQ -1 OR MONTH[I] EQ -1 OR ANNUAL[I]  
  EQ -1 OR OTHER[I] EQ -1 THEN HRWG[I]=-1;
```

```
/* If R refuses the first time asked, R is jumped out of wage section*/
```

```
IF E33400[I]=-1 OR E76200[I]=-1 then HRWG[I]=-1;
```

```
/*If R refuses or doesn't know the estimated wage, R is jumped out of the wage section*/
```

```
IF E33600[I]=-1 OR E76400[I]=-1 THEN HRWG[I]=-1;  
IF E33600[I]=-2 OR E76400[I]=-2 THEN HRWG[I]=-2;
```

```
IF ANNUAL[I] GT 0 THEN HRWG[I]=ANNUAL[I]; IF MONTH[I] GT 0 THEN HRWG[I]=MONTH[I];  
IF BIWKLY[I] GT 0 THEN HRWG[I]=BIWKLY[I]; IF WEEKLY[I] GT 0 THEN HRWG[I]=WEEKLY[I];  
IF HRWAGE[I] GT 0 THEN HRWG[I]=HRWAGE[I]; IF OTHER[I] GT 0 THEN HRWG[I]=OTHER[I];  
END;
```

```
/******SECTION 2: STOP DATE WAGES FOR YOUTH*****/
```

```
/*Part A—Hourly: Set up the hourly wage for youths who report an hourly wage*/
```

```
DO I=1 TO 7;
```

```
/*under 16, end wage, no overtime */
```

```
IF (E17100[I]=1 AND E37800[I]=1) then DO;  
  IF (E41900[I]>0) then HRWAGE[I]=E41900[I];
```

## Appendix 2: Employment Variable Creation

---

```
IF (E42000[I]>0) then HRWAGE[I]=E42000[I];
IF (E41900[I]=-3 or E42000[I]=-3) then HRWAGE[I]=E42200[I];
IF (E41900[I]=-2 or E42000[I]=-2) then HRWAGE[I]=E42200[I];
IF (E41900[I]=-1 or E42000[I]=-1) then HRWAGE[I]=-1;
IF E42900[I]>0 then HRWAGE[I]=E42900[I];
END;

/*16+, end wage, no overtime */
ELSE IF (E17100[I]=0 AND E79200[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]=-3 or E86900[I]=-3) then HRWAGE[I]=E87100[I];
  IF (E86800[I]=-2 or E86900[I]=-2) 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;

/*****Part B—Weekly: Set up the hourly wage for youths who report a weekly, daily, or other time unit in E-
19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in weekly units.*

DO I=1 TO 7;
/*under 16, weekly end wage divided by the number of hours worked per week*/
/*no overtime*/
IF (E17100[I]=1 AND E37800[I]=1) then DO;
  IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E52400[I]>0 and E37900[I]>0
    then WEEKLY[I]=(E52400[I]/E37900[I]);
  /*estimated*/
  IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E52600[I]>0
    and E37900[I]>0 and E52400[I]>-3 then WEEKLY[I]=(E52600[I]/E37900[I]);
  /*interviewer corrected*/
  IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E53400[I]>0
    and E37900[I]>0 and E52400[I]>-3 then WEEKLY[I]=(E53400[I]/E37900[I]);
  /*respondent corrected*/
  IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E54300[I]>0
    and E37900[I]>0 and E52400[I]>-3 then WEEKLY[I]=(E54300[I]/E37900[I]);
  /*missing values*/
  IF (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3) then WEEKLY[I]=E37900[I];
  IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E37900[I]=0 then WEEKLY[I]=0;
END;

/*16+, weekly end wage divided by the number of hours worked per week*/
/*no overtime*/
ELSE IF E17100[I]=0 AND E79200[I]=1 then DO;
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E97300[I]>0 and E82200[I]>0
    then WEEKLY[I]=(E97300[I]/E82200[I]);
  /*estimated*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E97500[I]>0
    and E82200[I]>0 and E97300[I]>-3 then WEEKLY[I]=(E97500[I]/E82200[I]);
  /*interviewer corrected*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E98300[I]>0
    and E82200[I]>0 and E97300[I]>-3 then WEEKLY[I]=(E98300[I]/E82200[I]);
  /*respondent corrected*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E99200[I]>0
    and E82200[I]>0 and E97300[I]>-3 then WEEKLY[I]=(E99200[I]/E82200[I]);
  /*missing values*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and (E82200[I]=-1
```

---

## Appendix 2: Employment Variable Creation

---

```
    or E82200[I]=-2 or E82200[I]=-3) then WEEKLY[I]=E82200[I];
    IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E82200[I]=0 then WEEKLY[I]=-6;
END;
END;
```

**\*\*\*\*\*Part C—Biweekly:** Set up the hourly wage for youths who report a biweekly time unit in E-19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in biweekly units.\*

```
DO I=1 TO 7;
```

```
/*Under 16, bi-wkly end wage divided by 2 times the number of hours worked per week*/
```

```
/*no overtime*/
```

```
    IF E17100[I]=1 AND E37800[I]=1 then DO;
```

```
        IF (E38200[I]=4) and E52400[I]>0 and E37900[I]>0 then BIWKLY[I]=(E52400[I]/(2*E37900[I]));
```

```
    /*estimated*/
```

```
        IF (E38200[I]=4) and E52600[I]>0 and E37900[I]>0 and E52400[I]>-3
            then BIWKLY[I]=(E52600[I]/(2*E37900[I]));
```

```
    /*interviewer corrected*/
```

```
        IF (E38200[I]=4) and E53400[I]>0 and E37900[I]>0 and E52400[I]>-3
            then BIWKLY[I]=(E53400[I]/(2*E37900[I]));
```

```
    /*respondent corrected*/
```

```
        IF (E38200[I]=4) and E54300[I]>0 and E37900[I]>0 and E52400[I]>-3
            then BIWKLY[I]=(E54300[I]/(2*E37900[I]));
```

```
    /*missing values*/
```

```
        IF (E38200[I]=4) and (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3) then BIWKLY[I]=E37900[I];
```

```
        IF (E38200[I]=4) and E37900[I]=0 then BIWKLY[I]=-6;
```

```
    END;
```

```
/*16+, bi-wkly end wage divided by 2 times the number of hours worked per week*/
```

```
/*no overtime*/
```

```
    ELSE IF E17100[I]=0 AND E79200[I]=1 then DO;
```

```
        IF (E83100[I]=4) and E97300[I]>0 and E82200[I]>0 then BIWKLY[I]=(E97300[I]/(2*E82200[I]));
```

```
    /*estimated*/
```

```
        IF (E83100[I]=4) and E97500[I]>0 and E82200[I]>0 and E97300[I]>-3
            then BIWKLY[I]=(E97500[I]/(2*E82200[I]));
```

```
    /*interviewer corrected*/
```

```
        IF (E83100[I]=4) and E98300[I]>0 and E82200[I]>0 and E97300[I]>-3
            then BIWKLY[I]=(E98300[I]/(2*E82200[I]));
```

```
    /*respondent corrected*/
```

```
        IF (E83100[I]=4) and E99200[I]>0 and E82200[I]>0 and E97300[I]>-3
            then BIWKLY[I]=(E99200[I]/(2*E82200[I]));
```

```
    /*missing values*/
```

```
        IF (E83100[I]=4) and (E82200[I]=-1 or E82200[I]=-2 or E82200[I]=-3) then BIWKLY[I]=E82200[I];
```

```
        IF (E83100[I]=4) and E82200[I]=0 then BIWKLY[I]=-6;
```

```
    END;
```

```
END;
```

**\*\*\*\*\*Part D—Monthly:** Set up the hourly wage for youths who report a month, or semi-monthly time unit in E-19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in month units.\*

```
DO I=1 TO 7;
```

```
/*under 16, month end wage divided by 4.3 times the number of hours worked per week*/
```

```
/*no overtime*/
```

```
    IF E17100[I]=1 AND E37800[I]=1 then DO;
```

```
        IF (E38200[I]=5 | E38200[I]=8) and E52400[I]>0 and E37900[I]>0
            then MONTH[I]=(E52400[I]/(4.3*E37900[I]));
```

```
    /*estimated*/
```

```
        IF (E38200[I]=5 | E38200[I]=8) and E52600[I]>0 and E37900[I]>0 and E52400[I]>-3
```



## Appendix 2: Employment Variable Creation

```
    then MONTH[I]=(E52600[I]/(4.3*E37900[I]));
/*interviewer corrected*/
    IF (E38200[I]=5 | E38200[I]=8) and E53400[I]>0 and E37900[I]>0 and E52400[I]>-3
    then MONTH[I]=(E53400[I]/(4.3*E37900[I]));
/*respondent corrected*/
    IF (E38200[I]=5 | E38200[I]=8) and E54300[I]>0 and E37900[I]>0 and E52400[I]>-3
    then MONTH[I]=(E54300[I]/(4.3*E37900[I]));
/*missing values*/
    IF (E38200[I]=5 | E38200[I]=8) and (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3)
    then MONTH[I]=E37900[I];
    IF (E38200[I]=5 | E38200[I]=8) and E37900[I]=0 then MONTH[I]=-6;
END;

/*16+, month end wage divided by 4.3 times the number of hours worked per week*/
/*no overtime*/
ELSE IF E17100[I]=0 AND E79200[I]=1 then DO;
    IF (E83100[I]=5 | E83100[I]=8) and E97300[I]>0 and E82200[I]>0
    then MONTH[I]=(E97300[I]/(4.3*E82200[I]));
/*estimated*/
    IF (E83100[I]=5 | E83100[I]=8) and E97500[I]>0 and E82200[I]>0 and E97300[I]>-3
    then MONTH[I]=(E97500[I]/(4.3*E82200[I]));
/*interviewer corrected*/
    IF (E83100[I]=5 | E83100[I]=8) and E98300[I]>0 and E82200[I]>0 and E97300[I]>-3
    then MONTH[I]=(E98300[I]/(4.3*E82200[I]));
/*respondent corrected*/
    IF (E83100[I]=5 | E83100[I]=8) and E99200[I]>0 and E82200[I]>0 and E97300[I]>-3
    then MONTH[I]=(E99200[I]/(4.3*E82200[I]));
/*missing values*/
    IF (E83100[I]=5 | E83100[I]=8) and (E82200[I]=-1 or E82200[I]=-2 or E82200[I]=-3)
    then MONTH[I]=E82200[I];
    IF (E83100[I]=5 | E83100[I]=8) and E82200[I]=0 then MONTH[I]=-6;
END;
END;

/*****Part E—Annual: Set up the hourly wage for youths who report an annual time unit in E-19200, E-38200, E-
62000, or E-83100. All of these youths were asked to report their wage in annual units.*/

DO I=1 TO 7;
/*Under 16, annual end wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
    IF E17100[I]=1 AND E37800[I]=1 then DO;
        IF E38200[I]=6 and E52400[I]>0 and E54600[I]>0 and E37900[I]>0
        then ANNUAL[I]=(E52400[I]/(E54600[I]*E37900[I]));
/*estimated*/
        IF (E38200[I]=6) and E52600[I]>0 and E37900[I]>0 and E52400[I]>-3
        then ANNUAL[I]=(E52600[I]/(E54600[I]*E37900[I]));
/*interviewer corrected*/
        IF (E38200[I]=6) and E53400[I]>0 and E37900[I]>0 and E52400[I]>-3
        then ANNUAL[I]=(E53400[I]/(E54600[I]*E37900[I]));
/*respondent corrected*/
        IF (E38200[I]=6) and E54300[I]>0 and E37900[I]>0 and E52400[I]>-3
        then ANNUAL[I]=(E54300[I]/(E54600[I]*E37900[I]));
/*missing values*/
        IF E38200[I]=6 and (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3) then ANNUAL[I]=E37900[I];
        IF E38200[I]=6 and (E54600[I]=-1 or E54600[I]=-2 or E54600[I]=-3) then ANNUAL[I]=E54600[I];
        IF E38200[I]=6 and E37900[I]=0 then ANNUAL[I]=-6;
    END;
END;
```

## Appendix 2: Employment Variable Creation

```
/*16+, annual end wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
  ELSE IF E17100[I]=0 AND E79200[I]=1 then DO;
    IF E83100[I]=6 and E97300[I]>0 and E99500[I]>0 and E82200[I]>0
      then ANNUAL[I]=(E97300[I]/(E99500[I]*E82200[I]));
  /*estimated*/
    IF (E83100[I]=6) and E97500[I]>0 and E82200[I]>0 and E97300[I]>-3
      then ANNUAL[I]=(E97500[I]/(E99500[I]*E82200[I]));
  /*interviewer corrected*/
    IF (E83100[I]=6) and E98300[I]>0 and E82200[I]>0 and E97300[I]>-3
      then ANNUAL[I]=(E98300[I]/(E99500[I]*E82200[I]));
  /*respondent corrected*/
    IF (E83100[I]=6) and E99200[I]>0 and E82200[I]>0 and E97300[I]>-3
      then ANNUAL[I]=(E99200[I]/(E99500[I]*E82200[I]));
  /*missing values*/
    IF E83100[I]=6 and (E82200[I]=-1 or E82200[I]=-2 or E82200[I]=-3) then ANNUAL[I]=E82200[I];
    IF E83100[I]=6 and (E99500[I]=-1 or E99500[I]=-2 or E99500[I]=-3) then ANNUAL[I]=E99500[I];
    IF E83100[I]=6 and E82200[I]=0 then ANNUAL[I]=-6;
  END;
END;

/****Part F—Other: Set up the hourly wage for youths who report a non-hourly time unit. Both the hourly wage
reported in the non-overtime and the overtime loops can be used without any further calculations.*/

DO I=1 TO 7;
/*under 16, end wage*/
  IF E17100[I]=1 AND E37800[I]=1 then DO;
    IF (E55100[I]>0) then OTHER[I]=E55100[I];
    IF (E55200[I]>0) then OTHER[I]=E55200[I];
  END;

/*16+, end wage*/
  ELSE IF E17100[I]=0 AND E79200[I]=1 then DO;
    IF (E100e[I]>0) then OTHER[I]=E100e[I];
    IF (E1001e[I]>0) then OTHER[I]=E1001e[I];
  END;
END;

/*****Create the Hourly Rate of Pay*****/

DO I=1 TO 7;
IF HRWAGE[I] EQ -4 OR WEEKLY[I] EQ -4 OR BIWKLY[I] EQ -4 OR MONTH[I] EQ -4 OR ANNUAL[I]
  EQ -4 OR OTHER[I] EQ -4 THEN HRWG[I]=-4;
IF HRWAGE[I] EQ -3 OR WEEKLY[I] EQ -3 OR BIWKLY[I] EQ -3 OR MONTH[I] EQ -3 OR ANNUAL[I]
  EQ -3 OR OTHER[I] EQ -3 THEN HRWG[I]=-3;
IF HRWAGE[I] EQ -2 OR WEEKLY[I] EQ -2 OR BIWKLY[I] EQ -2 OR MONTH[I] EQ -2 OR ANNUAL[I]
  EQ -2 OR OTHER[I] EQ -2 THEN HRWG[I]=-2;
IF HRWAGE[I] EQ -1 OR WEEKLY[I] EQ -1 OR BIWKLY[I] EQ -1 OR MONTH[I] EQ -1 OR ANNUAL[I]
  EQ -1 OR OTHER[I] EQ -1 THEN HRWG[I]=-1;

/* If R refuses the first time asked, R is jumped out of wage section*/
IF E52400[I]=-1 OR E97300[I]=-1 then HRWG[I]=-1;

/*If R refuses or doesn't know the estimated wage, R is jumped out of the wage section*/
IF E52600[I]=-1 OR E97500[I]=-1 THEN HRWG[I]=-1;
IF E52600[I]=-2 OR E97500[I]=-2 THEN HRWG[I]=-2;
```

## Appendix 2: Employment Variable Creation

---

```
IF ANNUAL[I] GT 0 THEN HRWG[I]=ANNUAL[I];   IF MONTH[I] GT 0 THEN HRWG[I]=MONTH[I];
IF BIWKLY[I] GT 0 THEN HRWG[I]=BIWKLY[I];   IF WEEKLY[I] GT 0 THEN HRWG[I]=WEEKLY[I];
IF HRWAGE[I] GT 0 THEN HRWG[I]=HRWAGE[I];   IF OTHER[I] GT 0 THEN HRWG[I]=OTHER[I];
```

```
END;
```

```
do I=1 to 7;
```

```
  if hrwg[I]=0 and (E19200[I]=-1 or E38200[I]=-1 or E62000[I]=-1 or E83100[I]=-1) then hrwg[I]=-1;
```

```
  if hrwg[I]=0 and (E19200[I]=-2 or E38200[I]=-2 or E62000[I]=-2 or E83100[I]=-2) then hrwg[I]=-2;
```

```
  if hrwg[I]=0 and (E19200[I]=-3 or E38200[I]=-3 or E62000[I]=-3 or E83100[I]=-3) then hrwg[I]=-3;
```

```
end;
```

```
do I=1 to 7;
```

```
  if hrwg[I]=0 and (E33500[I]>-4 or E52500[I]>-4 or E76300[I]>-4 or E97400[I]>-4) then hrwg[I]=-3;
```

```
end;
```

```
do I=1 to 7; if E17100[I]=-4 then hrwg[I]=-4; end;
```

```
do I=1 to 7; if pubid=1714 then hrwg[I]=-3; end;
```

```
HRWGR01=round(hrwg01,1);   HRWGR02=round(hrwg02,1);
```

```
HRWGR03=round(hrwg03,1);   HRWGR04=round(hrwg04,1);
```

```
HRWGR05=round(hrwg05,1);   HRWGR06=round(hrwg06,1);
```

```
HRWGR07=round(hrwg07,1);
```

```
endsas;
```

HOURLY MONETARY COMPENSATION AND JOB LENGTH < 13 WEEKS INDICATOR

Variables Created: CV\_HRLY\_COMPENSATION  
CV\_JOB<13\_WKS

Variables Used

This program uses all the variables listed under “Hourly Rate of Pay,” plus the following additional variables:

Name in Program	Question Name on CD	Name in Program	Question Name on CD
E2410001-04	YEMP-24100.01-.04	E6960001-03	YEMP-69600.01-.03
E3180001, 02	YEMP-31800.01, .02	E6970001	YEMP-69700.01
E3190001, 02	YEMP-31900.01, .02	E7460001, 02	YEMP-74600.01, .02
E4310001, 02	YEMP-43100.01, .02	E7470001, 02	YEMP-74700.01, .02
E4580001, 02	YEMP-45800.01, .02	E8800001-03	YEMP-88000.01-.03
E5080001	YEMP-50800.01	E9070001, 02	YEMP-90700.01, .02
E5090001	YEMP-50900.01	E9570001, 02	YEMP-95700.01, .02
E6690001-04	YEMP-66900.01-.04	E9580001, 02	YEMP-95800.01, .02

This program creates an hourly monetary compensation 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 the previously described hourly rate of pay variable, which calculates only the base pay rate. This program also constructs a dummy variable indicating whether the respondent worked less than 13 weeks at a given job.

Hourly wage 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. The program uses the following reporting categories:

1. The start wage for youths under the age of 16 as of the survey date who only report a start wage;
2. The stop wage for youths under the age of 16 as of the survey date who report both a start wage and a stop wage;
3. The start wage for youths age 16 or older as of the survey date who only report a start wage; and
4. The stop wage for youths age 16 or older as of the survey date who report both a start wage and a stop wage.

/\* Set up the hourly wage for youths who report their wage hourly. Both the hourly wage reported in the non-overtime and the overtime loops can be used without any further calculations. \*/

/\* NOTE: This program uses all of the arrays in the “Hourly Rate of Pay” program included above. The duplicate arrays are not reproduced here due to space considerations. Thus, this program includes the “Hourly Rate of Pay” arrays plus the following additional arrays. \*/

ARRAY E24100 E2410001 E2410002 E2410003 E2410004 E2410005 E2410006 E2410007;  
 ARRAY E31800 E3180001 E3180002 E3180003 E3180004 E3180005 E3180006 E3180007;  
 ARRAY E31900 E3190001 E3190002 E3190003 E3190004 E3190005 E3190006 E3190007;  
 ARRAY E24000 E2400001 E2400002 E2400003 E2400004 E2400005 E2400006 E2400007;  
 ARRAY E43100 E4310001 E4310002 E4310003 E4310004 E4310005 E4310006 E4310007;  
 ARRAY E45800 E4580001 E4580002 E4580003 E4580004 E4580005 E4580006 E4580007;  
 ARRAY E45900 E4590001 E4590002 E4590003 E4590004 E4590005 E4590006 E4590007;  
 ARRAY E50800 E5080001 E5080002 E5080003 E5080004 E5080005 E5080006 E5080007;  
 ARRAY E50900 E5090001 E5090002 E5090003 E5090004 E5090005 E5090006 E5090007;  
 ARRAY E66900 E6690001 E6690002 E6690003 E6690004 E6690005 E6690006 E6690007;  
 ARRAY E69600 E6960001 E6960002 E6960003 E6960004 E6960005 E6960006 E6960007;  
 ARRAY E69700 E6970001 E6970002 E6970003 E6970004 E6970005 E6970006 E6970007;  
 ARRAY E74600 E7460001 E7460002 E7460003 E7460004 E7460005 E7460006 E7460007;  
 ARRAY E74700 E7470001 E7470002 E7470003 E7470004 E7470005 E7470006 E7470007;  
 ARRAY E88000 E8800001 E8800002 E8800003 E8800004 E8800005 E8800006 E8800007;  
 ARRAY E90700 E9070001 E9070002 E9070003 E9070004 E9070005 E9070006 E9070007;

## Appendix 2: Employment Variable Creation

---

```
ARRAY E90800 E9080001 E9080002 E9080003 E9080004 E9080005 E9080006 E9080007;  
ARRAY E95700 E9570001 E9570002 E9570003 E9570004 E9570005 E9570006 E9570007;  
ARRAY E95800 E9580001 E9580002 E9580003 E9580004 E9580005 E9580006 E9580007;  
E99100=0;  
E35200=0;
```

```
/******SECTION 1: START WAGES REPORTED HOURLY, WEEKLY, BIWEEKLY, MONTHLY, AND  
ANNUALLY*****/
```

```
/*Part A—Hourly: Set up the hourly wage for youths who report an hourly wage*/
```

```
ARRAY HRWAGE HRWAGE01 HRWAGE02 HRWAGE03 HRWAGE04 HRWAGE05 HRWAGE06  
HRWAGE07;  
DO I=1 TO 7;  
HRWAGE[I]=0;
```

```
/*under 16, start wage, no overtime*/
```

```
/* This segment contains the same code as the identically titled segment in Section 1, Part A—Hourly, in the  
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
```

```
/*under 16, start wage, overtime included */
```

```
IF (E17100[I]=1 AND E37800[I]=0) THEN DO;  
IF E19000[I]>0 AND E29200[I]>0 THEN HRWAGE[I]=E29200[I]/E19000[I];  
IF E24100[I]>0 AND E29200[I]>0 THEN HRWAGE[I]=E29200[I]/E24100[I];  
IF E31800[I]>0 AND E19000[I]>0 THEN HRWAGE[I]=E31800[I]/E19000[I];  
IF E31800[I]>0 AND E24100[I]>0 THEN HRWAGE[I]=E31800[I]/E24100[I];  
IF E31900[I]=0 THEN HRWAGE[I]=-3;
```

```
END;
```

```
/*16+, start wage, no overtime*/
```

```
/* This segment contains the same code as the identically titled segment in Section 1, Part A—Hourly, in the  
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
```

```
/*16+, start wage, overtime included */
```

```
IF (E17100[I]=0 AND E79200[I]=0) THEN DO;  
IF E59800[I]>0 AND E72000[I]>0 THEN HRWAGE[I]=E72000[I]/E59800[I];  
IF E66900[I]>0 AND E72000[I]>0 THEN HRWAGE[I]=E72000[I]/E66900[I];  
IF E74600[I]>0 AND E59800[I]>0 THEN HRWAGE[I]=E74600[I]/E59800[I];  
IF E74600[I]>0 AND E66900[I]>0 THEN HRWAGE[I]=E74600[I]/E66900[I];  
IF E74700[I]=0 THEN HRWAGE[I]=-3;
```

```
END;
```

```
END;
```

```
/******Part B—Weekly: Set up the hourly wage for youths who report a weekly, daily, or other time unit in E-  
19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in weekly units.*/
```

```
ARRAY WEEKLY WEEKLY01 WEEKLY02 WEEKLY03 WEEKLY04 WEEKLY05 WEEKLY06 WEEKLY07;  
DO I=1 TO 7;  
WEEKLY[I]=0;
```

```
/*under 16, weekly start wage divided by the number of hours worked per week*/
```

```
/*no overtime*/
```

```
/* This segment contains the same code as the identically titled segment in Section 1, Part B—Weekly, in the  
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
```

```
/*overtime included*/
```

```
IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) AND E33500[I]>0 AND E19000[I]>0  
then WEEKLY[I]=(E33500[I]/E19000[I]);
```

```
/*estimated*/
```

```
IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) AND E33600[I]>0 AND E19000[I]>0 and
```

## Appendix 2: Employment Variable Creation

```
      E33500[I]>-3 then WEEKLY[I]=(E33600[I]/E19000[I]);
/*interviewer corrected*/
      IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E34400[I]>0 and E19000[I]>0 and
      E33500[I]>-3 then WEEKLY[I]=(E34400[I]/E19000[I]);
/*respondent corrected*/
      IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E35300[I]>0 and E19000[I]>0 and
      E33500[I]>-3 then WEEKLY[I]=(E35200[I]/E19000[I]);
/*missing values*/
      IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and (E19000[I]=-1 or E19000[I]=-2
      or E19000[I]=-3) then WEEKLY[I]=E19000[I];
      IF (E19200[I]=2 | E19200[I]=3 | E19200[I]=7 | E19200[I] GE 9) and E19000[I]=0 then WEEKLY[I]=-6;
END;

/*16+, weekly start wage divided by the number of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 1, Part B—Weekly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
      IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E76300[I]>0 and E59800[I]>0
      then WEEKLY[I]=(E76300[I]/E59800[I]);
/*estimated*/
      IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E76400[I]>0 and E59800[I]>0 and
      E76300[I]>-3 then WEEKLY[I]=(E76400[I]/E59800[I]);
/*interviewer corrected*/
      IF (E62000[I]=2 | E38200[I]=3 | E62000[I]=7 | E38200[I] GE 9) and E77200[I]>0 and E59800[I]>0 and
      E76300[I]>-3 then WEEKLY[I]=(E77200[I]/E59800[I]);
/*respondent corrected*/
      IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E78100[I]>0 and E59800[I]>0 and
      E76300[I]>-3 then WEEKLY[I]=(E78000[I]/E59800[I]);
/*missing values*/
      IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and (E59800[I]=-1 or E59800[I]=-2
      or E59800[I]=-3) then WEEKLY[I]=E59800[I];
      IF (E62000[I]=2 | E62000[I]=3 | E62000[I]=7 | E62000[I] GE 9) and E59800[I]=0 then WEEKLY[I]=-6;
END;
END;

/*****Part C—Biweekly: Set up the hourly wage for youths who report a biweekly time unit in E-19200, E-38200,
E-62000, or E-83100. All of these youths were asked to report their wage in biweekly units.*/

ARRAY BIWKLY BIWKLY01 BIWKLY02 BIWKLY03 BIWKLY04 BIWKLY05 BIWKLY06 BIWKLY07;
DO I=1 TO 7;
BIWKLY[I]=0;

/*under 16, bi-wkly start wage divided by 2 times the number of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 1, Part C—Biweekly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
      IF (E19200[I]=4) and E33500[I]>0 and E19000[I]>0 then BIWKLY[I]=(E33500[I]/(2*E19000[I]));
/*estimated*/
      IF (E19200[I]=4) and E33600[I]>0 and E19000[I]>0 and E33500[I]>-3
      then BIWKLY[I]=(E33600[I]/(2*E19000[I]));
/*interviewer corrected*/
      IF (E19200[I]=4) and E34400[I]>0 and E19000[I]>0 and E33500[I]>-3
      then BIWKLY[I]=(E34400[I]/(2*E19000[I]));
/*respondent corrected*/
      IF (E19200[I]=4) and E35300[I]>0 and E19000[I]>0 and E33500[I]>-3
```

## Appendix 2: Employment Variable Creation

```
    then BIWKLY[I]=(E35200/(2*E19000[I]));
/*missing values*/
    IF (E19200[I]=4) and (E19000[I]=-1 or E19000[I]=-2 or E19000[I]=-3) then BIWKLY[I]=E19000[I];
    IF (E19200[I]=4) and E19000[I]=0 then BIWKLY[I]=-6;
END;

/*16+, bi-wkly start wage divided by 2 times the number of hours worked per week*/
/*no overtime*/
    /* This segment contains the same code as the identically titled segment in Section 1, Part C—Biweekly, in the
    “Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
    IF (E62000[I]=4) and E76300[I]>0 and E59800[I]>0 then BIWKLY[I]=(E76300[I]/(2*E59800[I]));
/*estimated*/
    IF (E62000[I]=4) and E76400[I]>0 and E59800[I]>0 and E76300[I]>-3
        then BIWKLY[I]=(E76400[I]/(2*E59800[I]));
/*interviewer corrected*/
    IF (E62000[I]=4) and E77200[I]>0 and E59800[I]>0 and E76300[I]>-3
        then BIWKLY[I]=(E77200[I]/(2*E59800[I]));
/*respondent corrected*/
    IF (E62000[I]=4) and E78100[I]>0 and E59800[I]>0 and E76300[I]>-3
        then BIWKLY[I]=(E78000[I]/(2*E59800[I]));
/*missing values*/
    IF (E62000[I]=4) and (E59800[I]=-1 or E59800[I]=-2 or E59800[I]=-3) then BIWKLY[I]=E59800[I];
    IF (E62000[I]=4) and E59800[I]=0 then BIWKLY[I]=-6;
END;
END;

/*****Part D—Monthly: Set up the hourly wage for youths who report a monthly, or semi-monthly time unit in
E-19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in monthly units.*/

ARRAY MONTH MONTH01 MONTH02 MONTH03 MONTH04 MONTH05 MONTH06 MONTH07;
DO I=1 TO 7;
MONTH[I]=0;

/*under 16, month start wage divided by 4.3 times the number of hours worked per week*/
/*no overtime*/
    /* This segment contains the same code as the identically titled segment in Section 1, Part D—Monthly, in the
    “Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
    IF (E19200[I]=5 | E19200[I]=8) and E33500[I]>0 and E19000[I]>0
        then MONTH[I]=(E33500[I]/(4.3*E19000[I]));
/*estimated*/
    IF (E19200[I]=5 | E19200[I]=8) and E33600[I]>0 and E19000[I]>0 and E33500[I]>-3
        then MONTH[I]=(E33600[I]/(4.3*E19000[I]));
/*interviewer corrected*/
    IF (E19200[I]=5 | E19200[I]=8) and E34400[I]>0 and E19000[I]>0 and E33500[I]>-3
        then MONTH[I]=(E34400[I]/(4.3*E19000[I]));
/*respondent corrected*/
    IF (E19200[I]=5 | E19200[I]=8) and E35300[I]>0 and E19000[I]>0 and E33500[I]>-3
        then MONTH[I]=(E35200/(4.3*E19000[I]));
/*missing values*/
    IF (E19200[I]=5 | E19200[I]=8) and (E19000[I]=-1 or E19000[I]=-2 or E19000[I]=-3)
        then MONTH[I]=E19000[I];
    IF (E19200[I]=5 | E19200[I]=8) and E19000[I]=0 then MONTH[I]=-6;
END;

/*16+, month start wage divided by 4.3 times the number of hours worked per week*/
```

## Appendix 2: Employment Variable Creation

---

```
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 1, Part D—Monthly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
  IF (E62000[I]=5 | E62000[I]=8) and E76300[I]>0 and E59800[I]>0
    then MONTH[I]=(E76300[I]/(4.3*E59800[I]));
/*estimated*/
  IF (E62000[I]=5 | E62000[I]=8) and E76400[I]>0 and E59800[I]>0 and E76300[I]>-3
    then MONTH[I]=(E76400[I]/(4.3*E59800[I]));
/*interviewer corrected*/
  IF (E62000[I]=5 | E38200[I]=8) and E77200[I]>0 and E59800[I]>0 and E76300[I]>-3
    then MONTH[I]=(E77200[I]/(4.3*E59800[I]));
/*respondent corrected*/
  IF (E62000[I]=5 | E62000[I]=8) and E78100[I]>0 and E59800[I]>0 and E76300[I]>-3
    then MONTH[I]=(E78000[I]/(4.3*E59800[I]));
/*missing values*/
  IF (E62000[I]=5 | E62000[I]=8) and (E59800[I]=-1 or E59800[I]=-2 or E59800[I]=-3)
    then MONTH[I]=E59800[I];
  IF (E62000[I]=5 | E62000[I]=8) and E59800[I]=0 then MONTH[I]=-6;
END;
END;
```

```
/******Part E—Annual: Set up the hourly wage for youths who report an annual time unit in E-19200, E-38200, E-
62000, or E-83100. All of these youths were asked to report their wage in annual units.*/
```

```
ARRAY ANNUAL ANNUAL01 ANNUAL02 ANNUAL03 ANNUAL04 ANNUAL05 ANNUAL06 ANNUAL07;
DO I=1 TO 7;
ANNUAL[I]=0;
```

```
/*under 16, annual start wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 1, Part E—Annual, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
  IF E19200[I]=6 and E33500[I]>0 and E35600[I]>0 and E19000[I]>0
    then ANNUAL[I]=(E33500[I]/(E35600[I]*E19000[I]));
/*estimated*/
  IF (E19200[I]=6) and E33600[I]>0 and E19000[I]>0 and E33500[I]>-3
    then ANNUAL[I]=(E33600[I]/(E35600[I]*E19000[I]));
/*interviewer corrected*/
  IF (E19200[I]=6) and E34400[I]>0 and E19000[I]>0 and E33500[I]>-3
    then ANNUAL[I]=(E34400[I]/(E35600[I]*E19000[I]));
/*respondent corrected*/
  IF (E19200[I]=6) and E35300[I]>0 and E19000[I]>0 and E33500[I]>-3
    then ANNUAL[I]=(E35200/(E35600[I]*E19000[I]));
/*missing values*/
  IF E19200[I]=6 and (E19000[I]=-1 or E19000[I]=-2 or E19000[I]=-3) then ANNUAL[I]=E19000[I];
  IF E19200[I]=6 and (E35600[I]=-1 or E35600[I]=-2 or E35600[I]=-3) then ANNUAL[I]=E35600[I];
  IF E19200[I]=6 and E19000[I]=0 then ANNUAL[I]=0;
END;
```

```
/*16+, annual start wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 1, Part E—Annual, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
  IF E62000[I]=6 and E76300[I]>0 and E78400[I]>0 and E59800[I]>0
```



## Appendix 2: Employment Variable Creation

```

    then ANNUAL[I]=(E76300[I]/(E78400[I]*E59800[I]));
/*estimated*/
    IF (E62000[I]=6) and E76400[I]>0 and E59800[I]>0 and E76300[I]>-3
        then ANNUAL[I]=(E76400[I]/(E78400[I]*E59800[I]));
/*interviewer corrected*/
    IF (E62000[I]=6) and E77200[I]>0 and E59800[I]>0 and E76300[I]>-3
        then ANNUAL[I]=(E77200[I]/(E78400[I]*E59800[I]));
/*respondent corrected*/
    IF (E62000[I]=6) and E78100[I]>0 and E59800[I]>0 and E76300[I]>-3
        then ANNUAL[I]=(E78000[I]/(E78400[I]*E59800[I]));
/*missing values*/
    IF E62000[I]=6 and (E59800[I]=-1 or E59800[I]=-2 or E59800[I]=-3) then ANNUAL[I]=E59800[I];
    IF E62000[I]=6 and (E78400[I]=-1 or E78400[I]=-2 or E78400[I]=-3) then ANNUAL[I]=E78400[I];
    IF E62000[I]=6 and E59800[I]=0 then ANNUAL[I]=-6;
END;
END;

/*****Part F—Other: Set up the hourly wage for youths who report a non-hourly time unit. Both the hourly
wage reported in the non-overtime and the overtime loops can be used without any further calculations.*/

/* This segment contains the same code as Section 1, Part F—Other, in the “Hourly Rate of Pay” program
above. It is not reproduced here for space reasons. */

/*****Create the Hourly Rate of Pay*****/

ARRAY HRCOMP HRCOMP01 HRCOMP02 HRCOMP03 HRCOMP04 HRCOMP05 HRCOMP06
      HRCOMP07;
DO I=1 TO 7;
HRCOMP[I]=0;

IF HRWAGE[I] EQ -4 OR WEEKLY[I] EQ -4 OR BIWKLY[I] EQ -4 OR MONTH[I] EQ -4 OR ANNUAL[I]
  EQ -4 OR OTHER[I] EQ -4 THEN HRCOMP[I]=-4;
IF HRWAGE[I] EQ -3 OR WEEKLY[I] EQ -3 OR BIWKLY[I] EQ -3 OR MONTH[I] EQ -3 OR ANNUAL[I]
  EQ -3 OR OTHER[I] EQ -3 THEN HRCOMP[I]=-3;
IF HRWAGE[I] EQ -2 OR WEEKLY[I] EQ -2 OR BIWKLY[I] EQ -2 OR MONTH[I] EQ -2 OR ANNUAL[I]
  EQ -2 OR OTHER[I] EQ -2 THEN HRCOMP[I]=-2;
IF HRWAGE[I] EQ -1 OR WEEKLY[I] EQ -1 OR BIWKLY[I] EQ -1 OR MONTH[I] EQ -1 OR ANNUAL[I]
  EQ -1 OR OTHER[I] EQ -1 THEN HRCOMP[I]=-1;

/* if R refuses the first time asked, R is jumped out of wage section*/
IF E33400[I]=-1 or E33500[I]=-1 then HRCOMP[I]=-1;
IF E76200[I]=-1 or E76300[I]=-1 then HRCOMP[I]=-1;

/*if R refuses or doesn't know the estimated wage, R is jumped out of the wage section*/
IF E33600[I]=-1 or E76400[I]=-1 then HRCOMP[I]=-1;
IF E33600[I]=-2 or E76400[I]=-2 then HRCOMP[I]=-2;

IF ANNUAL[I] GT 0 THEN HRCOMP[I]=ANNUAL[I];
IF MONTH[I] GT 0 THEN HRCOMP[I]=MONTH[I];
IF BIWKLY[I] GT 0 THEN HRCOMP[I]=BIWKLY[I];
IF WEEKLY[I] GT 0 THEN HRCOMP[I]=WEEKLY[I];
IF HRWAGE[I] GT 0 THEN HRCOMP[I]=HRWAGE[I];
IF OTHER[I] GT 0 THEN HRCOMP[I]=OTHER[I];
END;

/*****SECTION 2: STOP DATE WAGES FOR YOUTH*****/
```

## Appendix 2: Employment Variable Creation

---

*\*\*\*\*\*Part A—Hourly: Set up the hourly wage for youths who report an hourly wage\**

DO I=1 TO 7;

*/\*under 16, end wage, no overtime \*/*

*/\* This segment contains the same code as the identically titled segment in Section 2, Part A—Hourly, in the “Hourly Rate of Pay” program above. It is not reproduced here for space reasons. \*/*

*/\*under 16, end wage, overtime included \*/*

IF (E17100[I]=1 AND E37800[I]=1) THEN DO;  
IF E37900[I]>0 AND E48200[I]>0 THEN HRWAGE[I]=E48200[I]/E37900[I];  
IF E43100[I]>0 AND E48200[I]>0 THEN HRWAGE[I]=E48200[I]/E43100[I];  
IF E50800[I]>0 AND E37900[I]>0 THEN HRWAGE[I]=E50800[I]/E37900[I];  
IF E50800[I]>0 AND E43100[I]>0 THEN HRWAGE[I]=E50800[I]/E43100[I];  
IF E50900[I]>0 THEN HRWAGE[I]=-3;

END;

*/\*16+, end wage, no overtime \*/*

*/\* This segment contains the same code as the identically titled segment in Section 2, Part A—Hourly, in the “Hourly Rate of Pay” program above. It is not reproduced here for space reasons. \*/*

*/\*16+, end wage, overtime included \*/*

IF (E17100[I]=0 AND E79200[I]=1) THEN DO;  
IF E82200[I]>0 AND E93100[I]>0 THEN HRWAGE[I]=E93100[I]/E82200[I];  
IF E88000[I]>0 AND E93100[I]>0 THEN HRWAGE[I]=E93100[I]/E88000[I];  
IF E95700[I]>0 AND E82200[I]>0 THEN HRWAGE[I]=E95700[I]/E82200[I];  
IF E95700[I]>0 AND E88000[I]>0 THEN HRWAGE[I]=E95700[I]/E88000[I];  
IF E95800[I]=0 THEN HRWAGE[I]=-3;

END;

END;

*\*\*\*\*\*Part B—Weekly: Set up the hourly wage for youths who report a weekly, daily, or other time unit in E-19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in weekly units.\*/*

DO I=1 TO 7;

*/\*under 16, weekly end wage divided by the number of hours worked per week\*/*

*/\*no overtime\*/*

*/\* This segment contains the same code as the identically titled segment in Section 2, Part B—Weekly, in the “Hourly Rate of Pay” program above. It is not reproduced here for space reasons. \*/*

*/\*overtime included\*/*

IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E52500[I]>0 and E37900[I]>0  
then WEEKLY[I]=(E52500[I]/E37900[I]);

*/\*estimated\*/*

IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E52600[I]>0 and E37900[I]>0 and  
E52500[I]>-3 then WEEKLY[I]=(E52600[I]/E37900[I]);

*/\*interviewer corrected\*/*

IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E53400[I]>0 and E37900[I]>0 and  
E52500[I]>-3 then WEEKLY[I]=(E53400[I]/E37900[I]);

*/\*respondent corrected\*/*

IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E54200[I]>0 and E37900[I]>0 and  
E52500[I]>-3 then WEEKLY[I]=(E54200[I]/E37900[I]);

*/\*missing values\*/*

IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and (E37900[I]=-1 or E37900[I]=-2  
or E37900[I]=-3) then WEEKLY[I]=E37900[I];

IF (E38200[I]=2 | E38200[I]=3 | E38200[I]=7 | E38200[I] GE 9) and E37900[I]=0 then WEEKLY[I]=0;

END;

*/\*16+, weekly end wage divided by the number of hours worked per week\*/*

## Appendix 2: Employment Variable Creation

---

```
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part B—Weekly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E97400[I]>0 and E82200[I]>0
    then WEEKLY[I]=(E97400[I]/E82200[I]);
/*estimated*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E97500[I]>0 and E82200[I]>0 and
  E97400[I]>-3 then WEEKLY[I]=(E97500[I]/E82200[I]);
/*interviewer corrected*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E98300[I]>0 and E82200[I]>0 and
  E97400[I]>-3 then WEEKLY[I]=(E98300[I]/E82200[I]);
/*respondent corrected*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E99100>0 and E82200[I]>0 and
  E97400[I]>-3 then WEEKLY[I]=(E99100/E82200[I]);
/*missing values*/
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and (E82200[I]=-1 or E82200[I]=-2
  or E82200[I]=-3) then WEEKLY[I]=E82200[I];
  IF (E83100[I]=2 | E83100[I]=3 | E83100[I]=7 | E83100[I] GE 9) and E82200[I]=0 then WEEKLY[I]=-6;
END;
END;
```

\*\*\*\*\***Part C—Biweekly:** Set up the hourly wage for youths who report a biweekly time unit in E-19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in biweekly units.\*/

DO I=1 TO 7;

```
/*under 16, bi-wkly end wage divided by 2 times the number of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part C—Biweekly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
  IF (E38200[I]=4) and E52500[I]>0 and E37900[I]>0 then BIWKLY[I]=(E52500[I]/(2*E37900[I]));
/*estimated*/
  IF (E38200[I]=4) and E52600[I]>0 and E37900[I]>0 and E52500[I]>-3
    then BIWKLY[I]=(E52600[I]/(2*E37900[I]));
/*interviewer corrected*/
  IF (E38200[I]=4) and E53400[I]>0 and E37900[I]>0 and E52500[I]>-3
    then BIWKLY[I]=(E53400[I]/(2*E37900[I]));
/*respondent corrected*/
  IF (E38200[I]=4) and E54200[I]>0 and E37900[I]>0 and E52500[I]>-3
    then BIWKLY[I]=(E54200[I]/(2*E37900[I]));
/*missing values*/
  IF (E38200[I]=4) and (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3) then BIWKLY[I]=E37900[I];
  IF (E38200[I]=4) and E37900[I]=0 then BIWKLY[I]=-6;
END;
```

```
/*16+, bi-wkly end wage divided by 2 times the number of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part C—Biweekly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
  IF (E83100[I]=4) and E97400[I]>0 and E82200[I]>0 then BIWKLY[I]=(E97400[I]/(2*E82200[I]));
/*estimated*/
  IF (E83100[I]=4) and E97500[I]>0 and E82200[I]>0 and E97400[I]>-3
    then BIWKLY[I]=(E97500[I]/(2*E82200[I]));
/*interviewer corrected*/
```

## Appendix 2: Employment Variable Creation

---

```
IF (E83100[I]=4) and E98300[I]>0 and E82200[I]>0 and E97400[I]>-3
  then BIWKLY[I]=(E98300[I]/(2*E82200[I]));
/*respondent corrected*/
IF (E83100[I]=4) and E99100>0 and E82200[I]>0 and E97400[I]>-3
  then BIWKLY[I]=(E99100/(2*E82200[I]));
/*missing values*/
IF (E83100[I]=4) and (E82200[I]=-1 or E82200[I]=-2 or E82200[I]=-3) then BIWKLY[I]=E82200[I];
IF (E83100[I]=4) and E82200[I]=0 then BIWKLY[I]=-6;
END;
END;

/*****Part D—Monthly: Set up the hourly wage for youths who report a month, or semi-month time unit in E-
19200, E-38200, E-62000, or E-83100. All of these youths were asked to report their wage in month units.*/

DO I=1 TO 7;

/*under 16, month end wage divided by 4.3 times the number of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part D—Monthly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
IF (E38200[I]=5 | E38200[I]=8) and E52500[I]>0 and E37900[I]>0
  then MONTH[I]=(E52500[I]/(4.3*E37900[I]));
/*estimated*/
IF (E38200[I]=5 | E38200[I]=8) and E52600[I]>0 and E37900[I]>0 and E52500[I]>-3
  then MONTH[I]=(E52600[I]/(4.3*E37900[I]));
/*interviewer corrected*/
IF (E38200[I]=5 | E38200[I]=8) and E53400[I]>0 and E37900[I]>0 and E52500[I]>-3
  then MONTH[I]=(E53400[I]/(4.3*E37900[I]));
/*respondent corrected*/
IF (E38200[I]=5 | E38200[I]=8) and E54200[I]>0 and E37900[I]>0 and E52500[I]>-3
  then MONTH[I]=(E54200[I]/(4.3*E37900[I]));
/*missing values*/
IF (E38200[I]=5 | E38200[I]=8) and (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3)
  then MONTH[I]=E37900[I];
IF (E38200[I]=5 | E38200[I]=8) and E37900[I]=0 then MONTH[I]=-6;
END;

/*16+, month end wage divided by 4.3 times the number of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part D—Monthly, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
IF (E83100[I]=5 | E83100[I]=8) and E97400[I]>0 and E82200[I]>0
  then MONTH[I]=(E97400[I]/(4.3*E82200[I]));
/*estimated*/
IF (E83100[I]=5 | E83100[I]=8) and E97500[I]>0 and E82200[I]>0 and E97400[I]>-3
  then MONTH[I]=(E97500[I]/(4.3*E82200[I]));
/*interviewer corrected*/
IF (E83100[I]=5 | E83100[I]=8) and E98300[I]>0 and E82200[I]>0 and E97400[I]>-3
  then MONTH[I]=(E98300[I]/(4.3*E82200[I]));
/*respondent corrected*/
IF (E83100[I]=5 | E83100[I]=8) and E99100>0 and E82200[I]>0 and E97400[I]>-3
  then MONTH[I]=(E99100/(4.3*E82200[I]));
/*missing values*/
IF (E83100[I]=5 | E83100[I]=8) and (E82200[I]=-1 or E82200[I]=-2 or E82200[I]=-3)
  then MONTH[I]=E82200[I];
```

## Appendix 2: Employment Variable Creation

```
IF (E83100[I]=5 | E83100[I]=8) and E82200[I]=0 then MONTH[I]=-6;
END;
END;

/*****Part E—Annual: Set up the hourly wage for youths who report an annual time unit in E-19200, E-38200,
E-62000, or E-83100. All of these youths were asked to report their wage in annual units.*/

DO I=1 TO 7;

/*under 16, annual end wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part E—Annual, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
IF E38200[I]=6 and E52500[I]>0 and E54600[I]>0 and E37900[I]>0
then ANNUAL[I]=(E52500[I]/(E54600[I]*E37900[I]));
/*estimated*/
IF (E38200[I]=6) and E52600[I]>0 and E37900[I]>0 and E52500[I]>-3
then ANNUAL[I]=(E52600[I]/(E54600[I]*E37900[I]));
/*interviewer corrected*/
IF (E38200[I]=6) and E53400[I]>0 and E37900[I]>0 and E52500[I]>-3
then ANNUAL[I]=(E53400[I]/(E54600[I]*E37900[I]));
/*respondent corrected*/
IF (E38200[I]=6) and E54200[I]>0 and E37900[I]>0 and E52500[I]>-3
then ANNUAL[I]=(E54200[I]/(E54600[I]*E37900[I]));
/*missing values*/
IF E38200[I]=6 and (E37900[I]=-1 or E37900[I]=-2 or E37900[I]=-3) then ANNUAL[I]=E37900[I];
IF E38200[I]=6 and (E54600[I]=-1 or E54600[I]=-2 or E54600[I]=-3) then ANNUAL[I]=E54600[I];
IF E38200[I]=6 and E37900[I]=0 then ANNUAL[I]=-6;
END;

/*16+, annual end wage divided by the # of weeks per year paid for times the # of hours worked per week*/
/*no overtime*/
/* This segment contains the same code as the identically titled segment in Section 2, Part E—Annual, in the
“Hourly Rate of Pay” program above. It is not reproduced here for space reasons. */
/*overtime included*/
IF E83100[I]=6 and E97400[I]>0 and E99500[I]>0 and E82200[I]>0
then ANNUAL[I]=(E97400[I]/(E99500[I]*E82200[I]));
/*estimated*/
IF (E83100[I]=6) and E97500[I]>0 and E82200[I]>0 and E97400[I]>-3
then ANNUAL[I]=(E97500[I]/(E99500[I]*E82200[I]));
/*interviewer corrected*/
IF (E83100[I]=6) and E98300[I]>0 and E82200[I]>0 and E97400[I]>-3
then ANNUAL[I]=(E98300[I]/(E99500[I]*E82200[I]));
/*respondent corrected*/
IF (E83100[I]=6) and E99100>0 and E82200[I]>0 and E97400[I]>-3
then ANNUAL[I]=(E99100/(E99500[I]*E82200[I]));
/*missing values*/
IF E83100[I]=6 and (E82200[I]=-1 or E82200[I]=-2 or E82200[I]=-3) then ANNUAL[I]=E82200[I];
IF E83100[I]=6 and (E99500[I]=-1 or E99500[I]=-2 or E99500[I]=-3) then ANNUAL[I]=E99500[I];
IF E83100[I]=6 and E82200[I]=0 then ANNUAL[I]=-6;
END;
END;

/*****Part F—Other: Set up the hourly wage for youths who report a non-hourly time unit. Both the hourly
wage reported in the non-overtime and the overtime loops can be used without any further calculations.*/
```

## Appendix 2: Employment Variable Creation

/\* This segment contains the same code as Section 2, Part F—Other, in the “Hourly Rate of Pay” program above. It is not reproduced here for space reasons. \*/

/\*\*\*\*\*\*Create the Hourly Compensation\*\*\*\*\*\*/

DO I=1 TO 7;

```
IF HRWAGE[I] EQ -4 OR WEEKLY[I] EQ -4 OR BIWKLY[I] EQ -4 OR MONTH[I] EQ -4 OR ANNUAL[I]
  EQ -4 OR OTHER[I] EQ -4 THEN HRCOMP[I]=-4;
IF HRWAGE[I] EQ -3 OR WEEKLY[I] EQ -3 OR BIWKLY[I] EQ -3 OR MONTH[I] EQ -3 OR ANNUAL[I]
  EQ -3 OR OTHER[I] EQ -3 THEN HRCOMP[I]=-3;
IF HRWAGE[I] EQ -2 OR WEEKLY[I] EQ -2 OR BIWKLY[I] EQ -2 OR MONTH[I] EQ -2 OR ANNUAL[I]
  EQ -2 OR OTHER[I] EQ -2 THEN HRCOMP[I]=-2;
IF HRWAGE[I] EQ -1 OR WEEKLY[I] EQ -1 OR BIWKLY[I] EQ -1 OR MONTH[I] EQ -1 OR ANNUAL[I]
  EQ -1 OR OTHER[I] EQ -1 THEN HRCOMP[I]=-1;
```

/\* if R refuses the first time asked, R is jumped out of wage section\*/

```
IF E52400[I]=-1 or E52500[I]=-1 then HRCOMP[I]=-1;
IF E97300[I]=-1 or E97400[I]=-1 then HRCOMP[I]=-1;
```

/\*if R refuses or doesn't know the estimated wage, R is jumped out of the wage section\*/

```
IF E52600[I]=-1 OR E97500[I]=-1 THEN HRCOMP[I]=-1;
IF E52600[I]=-2 OR E97500[I]=-2 THEN HRCOMP[I]=-2;
```

```
IF ANNUAL[I] GT 0 THEN HRCOMP[I]=ANNUAL[I];
IF MONTH[I] GT 0 THEN HRCOMP[I]=MONTH[I];
IF BIWKLY[I] GT 0 THEN HRCOMP[I]=BIWKLY[I];
IF WEEKLY[I] GT 0 THEN HRCOMP[I]=WEEKLY[I];
IF HRWAGE[I] GT 0 THEN HRCOMP[I]=HRWAGE[I];
IF OTHER[I] GT 0 THEN HRCOMP[I]=OTHER[I];
END;
```

/\*\*\*\*\*\*SECTION 3: JOB LASTS 13 WEEKS OR LESS\*\*\*\*\*\*/

```
ARRAY JLENG JLENG01 JLENG02 JLENG03 JLENG04 JLENG05 JLENG06 JLENG07;
DO I=1 TO 7;
  JLENG[I]=0
```

```
  JLENG[I]=-4;
  IF E37800[I]>-1 OR E79200[I]>-1 then DO;
    IF E37800[I]=0 OR E79200[I]=0 then JLENG[I]=1;
    ELSE JLENG[I]=0;
```

```
  END;
END;
```

/\* to correct for those not eligible for the employment section \*/

```
do I=1 to 7;
  if E17100[I]=-4 then HRCOMP[I]=-4 and JLENG[I]=-4;
end;
```

/\*correct for youth's being unable to report time unit rate of pay\*/

```
do I=1 to 7;
  if HRCOMP[I]=0 and (E19200[I]=-1 or E38200[I]=-1 or E62000[I]=-1 or E83100[I]=-1) then HRCOMP[I]=-1;
  if HRCOMP[I]=0 and (E19200[I]=-2 or E38200[I]=-2 or E62000[I]=-2 or E83100[I]=-2) then HRCOMP[I]=-2;
  if HRCOMP[I]=0 and (E19200[I]=-3 or E38200[I]=-3 or E62000[I]=-3 or E83100[I]=-3) then HRCOMP[I]=-3;
end;
```

## Appendix 2: Employment Variable Creation

---

```
/*correct for invalid skip in employment section*/  
do I=1 to 7;  
  if pubid=1714 then HRCOMP[I]=-3 and JLENG[I]=-3;  
end;
```

```
HRCOMR01=HRCOMP01;  
HRCOMR02=HRCOMP02;  
HRCOMR03=HRCOMP03;  
HRCOMR04=HRCOMP04;  
HRCOMR05=HRCOMP05;  
HRCOMR06=HRCOMP06;  
HRCOMR07=HRCOMP07;
```

```
HRCOMR01=round(HRCOMR01,1);  
HRCOMR02=round(HRCOMR02,1);  
HRCOMR03=round(HRCOMR03,1);  
HRCOMR04=round(HRCOMR04,1);  
HRCOMR05=round(HRCOMR05,1);  
HRCOMR06=round(HRCOMR06,1);  
HRCOMR07=round(HRCOMR07,1);
```

```
endsas;
```

NUMBER OF WEEKS WORKED DURING 19XX/SINCE LAST INTERVIEW (OR BIRTH)/SINCE AGE 14

Variables Created: CV\_WKSWK\_YR.80 – CV\_WKSWK\_YR.98  
CV\_WKSWK\_DLI  
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).

This program creates three variables counting the number of weeks the respondent worked at any employee-type job during several different time periods.

1. CV\_WKSWK\_YR is created for each individual for each year of potential work activity (1980-98). Respondents not working in a given year are given a default value of zero (0) weeks worked. Otherwise, the variable the actual cumulative weeks worked on all jobs in that year.
2. CV\_WKSWK\_DLI is created for each individual since the last interview. In the first year, this variable measures the worked since birth.
3. CV\_WKSWK\_EVER is created for each individual to measure the number of weeks worked since the age of 14.

\*\*\*\*\* Overlay multiple jobs over JOB 1 work weeks.  
This section is common to the 3 variables. \*\*\*\*\*;

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;

do over alljobs;  
if job3wks=1 then do; alljobs=job3wks; end;  
if job3wks=-3 and alljobs=0 then do;  
    alljobs=job3wks; end;  
end;

do over alljobs;  
if job4wks=1 then do; alljobs=job4wks; end;  
if job4wks=-3 and alljobs=0 then do;  
    alljobs=job4wks; end;  
end;

do over alljobs;  
if job5wks=1 then do; alljobs=job5wks; end;  
if job5wks=-3 and alljobs=0 then do;  
    alljobs=job5wks; end;  
end;

do over alljobs;  
if job6wks=1 then do; alljobs=job6wks; end;  
if job6wks=-3 and alljobs=0 then do;  
    alljobs=job6wks; end;  
end;

do over alljobs;  
if job7wks=1 then do; alljobs=job7wks; end;  
if job7wks=-3 and alljobs=0 then do;  
    alljobs=job7wks; end;

end;

\*\*\*\*\* 1. This section calculates cumulative weeks on all jobs for each year. \*\*\*\*\*

/\* 1980 \*/  
wks80=0;  
do i=1 to 52;  
    if alljobs=1 then do; wks80=wks80+1; end;  
end;  
do i=1 to 52;  
    if alljobs=-3 then do; wks80=-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 wks98. 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	1990	523-574
1982	105-156	1991	575-626
1983	157-209	1992	627-678
1984	210-261	1993	679-730
1985	262-313	1994	731-783
1986	314-365	1995	784-835
1987	366-417	1996	836-887
1988	418-470	1997	888-939
1989	471-522	1998	940-991 *****/

do i=1 to 7;  
/\* start date invalid \*/  
if starw<0 and starw>-4 then do;  
    if stopw>1 then do; wks80=-3; end;  
    if stopw>52 then do; wks81=-3; end;  
    if stopw>104 then do; wks82=-3; end;  
    if stopw>156 then do; wks83=-3; end;  
    if stopw>209 then do; wks84=-3; end;



## Appendix 2: Employment Variable Creation

```

if stopw>261 then do; wks85=-3; end;
if stopw>313 then do; wks86=-3; end;
if stopw>365 then do; wks87=-3; end;
if stopw>417 then do; wks88=-3; end;
if stopw>470 then do; wks89=-3; end;
if stopw>522 then do; wks90=-3; end;
if stopw>574 then do; wks91=-3; end;
if stopw>626 then do; wks92=-3; end;
if stopw>678 then do; wks93=-3; end;
if stopw>730 then do; wks94=-3; end;
if stopw>783 then do; wks95=-3; end;
if stopw>834 then do; wks96=-3; end;
if stopw>887 then do; wks97=-3; end;
if stopw>939 then do; wks98=-3; end;
end;

/* stop date invalid */
if stopw<0 and stopw>-4 then do;
  if starw<53 then do; wks80=-3; end;
  if starw<105 then do; wks81=-3; end;
  if starw<157 then do; wks82=-3; end;
  if starw<210 then do; wks83=-3; end;
  if starw<262 then do; wks84=-3; end;
  if starw<314 then do; wks85=-3; end;
  if starw<366 then do; wks86=-3; end;
  if starw<418 then do; wks87=-3; end;
  if starw<471 then do; wks88=-3; end;
  if starw<523 then do; wks89=-3; end;
  if starw<575 then do; wks90=-3; end;
  if starw<627 then do; wks91=-3; end;
  if starw<679 then do; wks92=-3; end;
  if starw<731 then do; wks93=-3; end;
  if starw<784 then do; wks94=-3; end;
  if starw<836 then do; wks95=-3; end;
  if starw<888 then do; wks96=-3; end;
  if starw<940 then do; wks97=-3; end;
  if starw<991 then do; wks98=-3; end;
end;
end;

*****2. This section calculates cumulative
weeks on all jobs since the last interview. *****;

allwks=0;

if birthwk>0 then do;
  do i=birthwk to 992;
    if alljobs=1 then do; allwks=allwks+1; end;
  end;
  do i=birthwk to 992;
    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 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;
end;

/* [END CV_WKSWK_DLI] */

*****3. This section calculates cumulative
weeks on all jobs since age 14. *****;

wks14=0;

if age14wk>0 then do;
  do i=age14wk to 992;
    if alljobs=1 then do; wks14=wks14+1; end;
    if alljobs=-3 then do; wks14=-3; end;
  end;
end;

do i=1 to 7;
if age14wk<53 then do;
  if starw<0 and starw>-4 and stopw>1 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<53 then do;
    wks14=-3; end;
end;
if age14wk<105 then do;
  if starw<0 and starw>-4 and stopw>52 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<105 then do;
    wks14=-3; end;
end;
if age14wk<157 then do;
  if starw<0 and starw>-4 and stopw>104 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<157 then do;
    wks14=-3; end;
end;
if age14wk<210 then do;
  if starw<0 and starw>-4 and stopw>156 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<210 then do;
    wks14=-3; end;
end;
if age14wk<262 then do;
  if starw<0 and starw>-4 and stopw>209 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<262 then do;
    wks14=-3; end;
end;
if age14wk<314 then do;

```

## Appendix 2: Employment Variable Creation

```

if starw<0 and starw>-4 and stopw>261 then do;
  wks14=-3; end;
if stopw<0 and stopw>-4 and starw<314 then do;
  wks14=-3; end;
end;
if age14wk<366 then do;
  if starw<0 and starw>-4 and stopw>313 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<366 then do;
    wks14=-3; end;
end;
if age14wk<418 then do;
  if starw<0 and starw>-4 and stopw>365 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<418 then do;
    wks14=-3; end;
end;
if age14wk<471 then do;
  if starw<0 and starw>-4 and stopw>417 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<471 then do;
    wks14=-3; end;
end;
if age14wk<523 then do;
  if starw<0 and starw>-4 and stopw>470 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<523 then do;
    wks14=-3; end;
end;
if age14wk<575 then do;
  if starw<0 and starw>-4 and stopw>522 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<575 then do;
    wks14=-3; end;
end;
if age14wk<627 then do;
  if starw<0 and starw>-4 and stopw>574 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<627 then do;
    wks14=-3; end;
end;
if age14wk<679 then do;
  if starw<0 and starw>-4 and stopw>626 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<679 then do;
    wks14=-3; end;
end;
if age14wk<731 then do;
  if starw<0 and starw>-4 and stopw>678 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<731 then do;
    wks14=-3; end;
end;
if age14wk<784 then do;
  if starw<0 and starw>-4 and stopw>730 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<784 then do;
    wks14=-3; end;
end;
if age14wk<836 then do;
  if starw<0 and starw>-4 and stopw>783 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<836 then do;
    wks14=-3; end;
end;
if age14wk<888 then do;
  if starw<0 and starw>-4 and stopw>835 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<888 then do;
    wks14=-3; end;
end;
if age14wk<940 then do;
  if starw<0 and starw>-4 and stopw>887 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<940 then do;
    wks14=-3; end;
end;
if age14wk<991 then do;
  if starw<0 and starw>-4 and stopw>939 then do;
    wks14=-3; end;
  if stopw<0 and stopw>-4 and starw<991 then do;
    wks14=-3; end;
end;
end;

***** Include valid skips for all variables *****,

if e200=0 then do;
  wks80=-4; wks81=-4; wks82=-4; wks83=-4; wks84=-4;
  wks85=-4; wks86=-4; wks87=-4; wks88=-4;
  wks89=-4; wks90=-4; wks91=-4; wks92=-4;
  wks93=-4; wks94=-4; wks95=-4; wks96=-4;
  wks97=-4; wks98=-4; allwks=-4; wks14=-4; end;

if e200=-3 then do;
  wks80=-3; wks81=-3; wks82=-3; wks83=-3; wks84=-3;
  wks85=-3; wks86=-3; wks87=-3; wks88=-3;
  wks89=-3; wks90=-3; wks91=-3; wks92=-3;
  wks93=-3; wks94=-3; wks95=-3; wks96=-3;
  wks97=-3; wks98=-3; allwks=-3; wks14=-3; end;

*if start year is missing;*
if pubid=411 or pubid=631 then do;
  wks80=-3; wks81=-3; wks82=-3; wks83=-3; wks84=-3;
  wks85=-3; wks86=-3; wks87=-3; wks88=-3;
  wks89=-3; wks90=-3; wks91=-3; wks92=-3;
  wks93=-3; wks94=-3; wks95=-3; wks96=-3;
  wks97=-3; wks98=-3; allwks=-3; wks14=-3; end;

endsas;

```

WEEKS WORKED AT EMPLOYEE JOB #X DURING 19XX

Variables Created: CV\_WKSWK\_JOB\_YR.01.80 – CV\_WKSWK\_JOB\_YR.01.98  
CV\_WKSWK\_JOB\_YR.02.80 – CV\_WKSWK\_JOB\_YR.02.98 etc. through job #7

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 in round 1 was seven, so variables are created for seven jobs for each respondent.

\*\*\*\*\* Calculate cumulative weeks on individual jobs for each year \*\*\*\*\*;

/\* 1980 \*/

wks801=0; wks802=0; wks803=0; wks804=0;  
wks805=0; wks806=0; wks807=0;

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;

end;

do i=1 to 52;

if job1wks=-3 then do; wks801=-3; end;  
if job2wks=-3 then do; wks802=-3; end;  
if job3wks=-3 then do; wks803=-3; end;  
if job4wks=-3 then do; wks804=-3; end;  
if job5wks=-3 then do; wks805=-3; end;  
if job6wks=-3 then do; wks806=-3; end;  
if job7wks=-3 then do; wks807=-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 wks811 through wks817 for 1981, wks821 through wks827 for 192=82, and so on up to wks981 through wks987 for 1998. 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	1990	523-574
1982	105-156	1991	575-626
1983	157-209	1992	627-678
1984	210-261	1993	679-730
1985	262-313	1994	731-783
1986	314-365	1995	784-835
1987	366-417	1996	836-887
1988	418-470	1997	888-939

1989 471-522 1998 940-991 \*\*\*\*\*/

\*\*\* Insert valid skips;

if e200=0 then do;

wks801=-4; wks802=-4; wks803=-4; wks804=-4;

wks805=-4; wks806=-4; wks807=-4;

/\*\* The seven weeks variables for each intervening year are similarly set to -4 \*\*/

wks981=-4; wks982=-4; wks983=-4; wks984=-4;

wks985=-4; wks986=-4; wks987=-4;

end;

if e200=-3 then do;

wks801=-3; wks802=-3; wks803=-3; wks804=-3;

wks805=-3; wks806=-3; wks807=-3;

/\*\* The seven weeks variables for each intervening year are similarly set to -3 \*\*/

wks981=-3; wks982=-3; wks983=-3; wks984=-3;

wks985=-3; wks986=-3; wks987=-3;

end;

\*if start year is missing;

if pubid=411 or pubid=631 then do;

wks801=-3; wks811=-3; wks821=-3; wks831=-3;

wks841=-3; wks851=-3; wks861=-3; wks871=-3;

wks881=-3; wks891=-3; wks901=-3; wks911=-3;

wks921=-3; wks931=-3; wks941=-3; wks951=-3;

wks961=-3; wks971=-3; wks981=-3;

end;

if starw1<0 and starw1>-4 then do;

/\*1980\*/

if stopw1>1 then do; wks801=-3; end;

if stopw2>1 then do; wks802=-3; end;

if stopw3>1 then do; wks803=-3; end;

if stopw4>1 then do; wks804=-3; end;

if stopw5>1 then do; wks805=-3; end;

if stopw6>1 then do; wks806=-3; end;

if stopw7>1 then do; wks807=-3; end;

/\*\* The same seven commands applied to the 1980 weeks variables are used for each subsequent year.

## Appendix 2: Employment Variable Creation

However, only the first job (first weeks variable) for each year is shown here due to space considerations. \*\*/

```

if stopw1>52 then do; wks811=-3; end;
  /*** and so on through job 7 for 1981 ***/

if stopw1>104 then do; wks821=-3; end;
  /*** and so on through job 7 for 1982 ***/

if stopw1>156 then do; wks831=-3; end;
  /*** and so on through job 7 for 1983 ***/

if stopw1>209 then do; wks841=-3; end;
  /*** and so on through job 7 for 1984 ***/

if stopw1>261 then do; wks851=-3; end;
  /*** and so on through job 7 for 1985 ***/

if stopw1>313 then do; wks861=-3; end;
  /*** and so on through job 7 for 1986 ***/

if stopw1>365 then do; wks871=-3; end;
  /*** and so on through job 7 for 1987 ***/

if stopw1>417 then do; wks881=-3; end;
  /*** and so on through job 7 for 1988 ***/

if stopw1>470 then do; wks891=-3; end;
  /*** and so on through job 7 for 1989 ***/

if stopw1>522 then do; wks901=-3; end;
  /*** and so on through job 7 for 1990 ***/

if stopw1>574 then do; wks911=-3; end;
  /*** and so on through job 7 for 1991 ***/

if stopw1>626 then do; wks921=-3; end;
  /*** and so on through job 7 for 1992 ***/

if stopw1>678 then do; wks931=-3; end;
  /*** and so on through job 7 for 1993 ***/

if d stopw1>730 then do; wks941=-3; end;
  /*** and so on through job 7 for 1994 ***/

if stopw1>783 then do; wks951=-3; end;
  /*** and so on through job 7 for 1995 ***/

if stopw1>835 then do; wks961=-3; end;
  /*** and so on through job 7 for 1996 ***/

if stopw1>887 then do; wks971=-3; end;
  /*** and so on through job 7 for 1997 ***/

if stopw1>939 then do; wks981=-3; end;
  /*** and so on through job 7 for 1998 ***/

end;

/* stop date invalid */
if stopw1<0 and stopw1>-4 then do;

/*1980*/
  if starw1<53 then do; wks801=-3; end;
  if starw2<53 then do; wks802=-3; end;
  if starw3<53 then do; wks803=-3; end;

```

```

if starw4<53 then do; wks804=-3; end;
if starw5<53 then do; wks805=-3; end;
if starw6<53 then do; wks806=-3; end;
if starw7<53 then do; wks807=-3; end;

```

/\*\*/ Again, the same seven commands applied to the 1980 weeks variables are used for each subsequent year; only the first job (first weeks variable) is shown here. \*/

```

if starw1<105 then do; wks811=-3; end;
  /*** and so on through job 7 for 1981 ***/

if starw1<157 then do; wks821=-3; end;
  /*** and so on through job 7 for 1982 ***/

if starw1<210 then do; wks831=-3; end;
  /*** and so on through job 7 for 1983 ***/

if starw1<262 then do; wks841=-3; end;
  /*** and so on through job 7 for 1984 ***/

if starw1<314 then do; wks851=-3; end;
  /*** and so on through job 7 for 1985 ***/

if starw1<366 then do; wks861=-3; end;
  /*** and so on through job 7 for 1986 ***/

if starw1<418 then do; wks871=-3; end;
  /*** and so on through job 7 for 1987 ***/

if starw1<471 then do; wks881=-3; end;
  /*** and so on through job 7 for 1988 ***/

if starw1<523 then do; wks891=-3; end;
  /*** and so on through job 7 for 1989 ***/

if starw1<575 then do; wks901=-3; end;
  /*** and so on through job 7 for 1990 ***/

if starw1<627 then do; wks911=-3; end;
  /*** and so on through job 7 for 1991 ***/

if starw1<679 then do; wks921=-3; end;
  /*** and so on through job 7 for 1992 ***/

if starw1<731 then do; wks931=-3; end;
  /*** and so on through job 7 for 1993 ***/

if starw1<784 then do; wks941=-3; end;
  /*** and so on through job 7 for 1994 ***/

if starw1<836 then do; wks951=-3; end;
  /*** and so on through job 7 for 1995 ***/

if starw1<888 then do; wks961=-3; end;
  /*** and so on through job 7 for 1996 ***/

if starw1<940 then do; wks971=-3; end;
  /*** and so on through job 7 for 1997 ***/

if starw1<991 then do; wks981=-3; end;
  /*** and so on through job 7 for 1998 ***/

end;

endsas;

```

TOTAL TENURE AT JOB #X AS OF THE SURVEY DATE

Variables Created: CV\_WKSWK\_JOB\_DLI.01 – CV\_WKSWK\_JOB\_DLI.07

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. 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 in round 1 was seven, so variables are created for seven jobs for each respondent.

\*\*\*\*\* Calculate cumulative weeks on individual jobs for all years \*\*\*\*\*;

```

tenure1=0; tenure2=0; tenure3=0; tenure4=0;
tenure5=0; tenure6=0; tenure7=0;

do i=1 to 992;
  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;
end;

flag=0;

do i=1 to 992;
  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;
end;

do i=1 to 992;
  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 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;
end;

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

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

*if start year is missing;
if pubid=411 or pubid=631 then do; tenure1=-3; end;

endsas;

```

TOTAL HOURS WORKED IN 19XX/SINCE AGE 14

Variables Created: CV\_HOURS\_WK\_YR.80 – CV\_HOURS\_WK\_YR.98  
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).

This program calculates the number of hours worked by the respondent at all **employee-type** jobs in each calendar year and 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

```
*****Organize hours for each job. This
section is common to both programs. *****;

/* shrs161 = starting hours age<16
   hrwk161 = ending hours age<16
   shrs1 = starting hours age 16+
   hrwk1 = ending hours age 16+ */

array s16hrs (k) shrs161-shrs167;
array hr16wk (k) hrwk161-hrwk167;
array shrs (k) shrs1-shrs7;
array hrwk (k) hrwk1-hrwk7;

array hour16s (k) hours161-hours167;
array hours (k) hours1-hours7;
array hrck16 (k) hrck161-hrck167;
array hrck (k) hrck1-hrck7;

array starw (i) starw1-starw7;
array stopw (i) stopw1-stopw7;
* Set default hours to zero (0);
do over hours;
   hour16s=0;
   hours=0;
   hrck16=0;
   hrck=0;
end;

* Take starting hours if reported (eliminates -4's);
do over s16hrs;
   if s16hrs>0 then do; hour16s=s16hrs; end;
end;

do over s16hrs;
   if s16hrs<0 and s16hrs ne -4 and s16hrs ne . then do;
      hrck16=s16hrs; end;
end;

* Write over if end hours reported;
do over hr16wk;
   if hr16wk>0 then do; hour16s=hr16wk; end;
end;

do over hr16wk;
   if hr16wk<0 and hr16wk ne -4 and hr16wk ne . then
      do; hrck16=hr16wk; end;
   if hr16wk=0 or hr16wk>0 then do; hrck16=0; end;
end;

* Take starting hours if reported (eliminates -4's);
do over shrs;
   if shrs>0 then do; hours=shrs; end;
end;

do over shrs;
   if shrs<0 and shrs ne -4 and shrs ne . then do;
      hrck=shrs; end;
end;

* Write over if end hours reported;
do over hrwk;
   if hrwk>0 then do; hours=hrwk; end;
end;

do over hrwk;
   if hrwk<0 and hrwk ne -4 and hrwk ne . then do;
      hrck=hrwk; end;
   if hrwk=0 or hrwk>0 then do; hrck=0; end;
end;

***** 1. This section calculates
cumulative weeks on each job for each year. Ends at:
[END CV_HOURS_WK_YR] *****;

/* 1980 */
wks16801=0; wks16802=0; wks16803=0;
   wks16804=0; wks16805=0; wks16806=0;
   wks16807=0;

do i=1 to 52;
   if job1wks=1 then do; wks16801=wks16801+1; end;
   if job2wks=1 then do; wks16802=wks16802+1; end;
   if job3wks=1 then do; wks16803=wks16803+1; end;
   if job4wks=1 then do; wks16804=wks16804+1; end;
   if job5wks=1 then do; wks16805=wks16805+1; end;
   if job6wks=1 then do; wks16806=wks16806+1; end;
   if job7wks=1 then do; wks16807=wks16807+1; end;
end;
```

## Appendix 2: Employment Variable Creation

```

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 then do;
    ah16801=hours161*wks16801;
    ah16802=hours162*wks16802;
    ah16803=hours163*wks16803;
    ah16804=hours164*wks16804;
    ah16805=hours165*wks16805;
    ah16806=hours166*wks16806;
    ah16807=hours167*wks16807;
  end;
  if job1wks=-3 or job2wks=-3 or job3wks=-3 or
    job4wks=-3 or job5wks=-3 or job6wks=-3 or
    job7wks=-3 then do;
    ah16801=-3; ah16802=-3; ah16803=-3;
    ah16804=-3; ah16805=-3; ah16806=-3;
    ah16807=-3;
    goto exit1;
  end;
end; exit1;

if ah16801 ne -3 and ah16802 ne -3 and ah16803 ne -
  3 and ah16804 ne -3 and ah16805 ne -3 and
  ah16806 ne -3 and ah16807 ne -3 then do;
  tothrs80=ah16801+ah16802+ah16803+
    ah16804+ah16805+ah16806+ah16807;
end;
if ah16801=-3 or ah16802=-3 or ah16803=-3 or
  ah16804=-3 or ah16805=-3 or ah16806=-3 or
  ah16807=-3 then do;
  tothrs80=-3;
end;

if wks16801>0 and hrck161<0 then tothrs80=hrck161;
if wks16802>0 and hrck162<0 then tothrs80=hrck162;
if wks16803>0 and hrck163<0 then tothrs80=hrck163;
if wks16804>0 and hrck164<0 then tothrs80=hrck164;
if wks16805>0 and hrck165<0 then tothrs80=hrck165;
if wks16806>0 and hrck166<0 then tothrs80=hrck166;
if wks16807>0 and hrck167<0 then tothrs80=hrck167;

/***** At this point the program loops through the
same code used above for 1980 for each year 1981-85,
creating the variables wks16811 through 16817 for
1981, wks16821 through wks16827 for 1982, and so
on through 1985 (at which point the pattern changes).
These loops are deleted due to space considerations;
users who need to see the entire code should contact
NLS User Services. The week numbers used in the "do
i" statement for 1981-85 are as follows:
  1981  53-104
  1982  105-156
  1983  157-209
  1984  210-261
  1985  262-313*****/

```

```

/***** Beginning in 1986 the creation pattern changes
due to different questions asked of those age 16 and
older. The complete code for 1986 is shown here; the
similar code for 1987-98 is deleted due to space
considerations. The week numbers used in the "do i"
statements for 1987-98 are as follows:
  1987  366-417      1993  679-730
  1988  418-470      1994  731-783
  1989  471-522      1995  784-835
  1990  523-574      1996  836-887
  1991  575-626      1997  888-939
  1992  627-678      1998  940-991 *****/

/* 1986 */
wks861=0; wks862=0; wks863=0; wks864=0;
  wks865=0; wks866=0; wks867=0;
wks16861=0; wks16862=0; wks16863=0;
  wks16864=0; wks16865=0; wks16866=0;
  wks16867=0;

do i=314 to 365;
  if job1wks=1 then do; wks861=wks861+1; end;
  if job2wks=1 then do; wks862=wks862+1; end;
  if job3wks=1 then do; wks863=wks863+1; end;
  if job4wks=1 then do; wks864=wks864+1; end;
  if job5wks=1 then do; wks865=wks865+1; end;
  if job6wks=1 then do; wks866=wks866+1; end;
  if job7wks=1 then do; wks867=wks867+1; end;
end;

do i=314 to 365;
  if job1wks=1 then do; wks16861=wks16861+1; end;
  if job2wks=1 then do; wks16862=wks16862+1; end;
  if job3wks=1 then do; wks16863=wks16863+1; end;
  if job4wks=1 then do; wks16864=wks16864+1; end;
  if job5wks=1 then do; wks16865=wks16865+1; end;
  if job6wks=1 then do; wks16866=wks16866+1; end;
  if job7wks=1 then do; wks16867=wks16867+1; end;
end;

do i=314 to 365;
  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 then do;
    ah16861=hours161*wks16861;
    ah16862=hours162*wks16862;
    ah16863=hours163*wks16863;
    ah16864=hours164*wks16864;
    ah16865=hours165*wks16865;
    ah16866=hours166*wks16866;
    ah16867=hours167*wks16867;
  end;
  ah861=hours1*wks861;
  ah862=hours2*wks862;
  ah863=hours3*wks863;
  ah864=hours4*wks864;
  ah865=hours5*wks865;

```

## Appendix 2: Employment Variable Creation

```

    ah866=hours6*wks866;
    ah867=hours7*wks867;
end;
if job1wks=-3 or job2wks=-3 or job3wks=-3 or
    job4wks=-3 or job5wks=-3 or job6wks=-3 or
    job7wks=-3 then do;
    ah16861=-3; ah16862=-3; ah16863=-3;
    ah16864=-3; ah16865=-3; ah16866=-3;
    ah16867=-3;
    ah861=-3; ah862=-3; ah863=-3; ah864=-3;
    ah865=-3; ah866=-3; ah867=-3;
    goto exit7;
end;
end; exit7;

if ah16861 ne -3 and ah16862 ne -3 and ah16863 ne -3
    and ah16864 ne -3 and ah16865 ne -3 and
    ah16866 ne -3 and ah16867 ne -3 and
    ah861 ne -3 and ah862 ne -3 and ah863 ne -3 and
    ah864 ne -3 and ah865 ne -3 and ah866 ne -3
    and ah867 ne -3 then do;
    tothrs86=ah16861+ah16862+ah16863+ah16864+
        ah16865+ah16866+ah16867+ah861+
        ah862+ah863+ah864+ah865+ah866+ ah867;
end;
if ah16861=-3 or ah16862=-3 or ah16863=-3 or
    ah16864=-3 or ah16865=-3 or ah16866=-3 or
    ah16867=-3 or
    ah861=-3 or ah862=-3 or ah863=-3 or ah864=-3 or
    ah865=-3 or ah866=-3 or ah867=-3 then do;
    tothrs86=-3;
end;

if wks16861>0 and hrck161<0 then tothrs86=hrck161;
if wks16862>0 and hrck162<0 then tothrs86=hrck162;
if wks16863>0 and hrck163<0 then tothrs86=hrck163;
if wks16864>0 and hrck164<0 then tothrs86=hrck164;
if wks16865>0 and hrck165<0 then tothrs86=hrck165;
if wks16866>0 and hrck166<0 then tothrs86=hrck166;
if wks16867>0 and hrck167<0 then tothrs86=hrck167;

if wks861>0 and hrck1<0 then tothrs86=hrck1;
if wks862>0 and hrck2<0 then tothrs86=hrck2;
if wks863>0 and hrck3<0 then tothrs86=hrck3;
if wks864>0 and hrck4<0 then tothrs86=hrck4;
if wks865>0 and hrck5<0 then tothrs86=hrck5;
if wks866>0 and hrck6<0 then tothrs86=hrck6;
if wks867>0 and hrck7<0 then tothrs86=hrck7;

/* 1987-98 deleted */

do i=1 to 7;
/* 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;
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;
end;
end;

/* [END CV_HOURS_WK_YR] */

***** 2. This section calculates
cumulative weeks on all jobs since age 14. Ends at:
[END CV_HOURS_WK_EVER] *****;

wks1=0; wks2=0; wks3=0; wks4=0; wks5=0; wks6=0;
    wks7=0;

if 0<age14wk<992 then do;
    do i=age14wk to 992;
        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;
    end;
end;

```



## Appendix 2: Employment Variable Creation

```

    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;
end;
end;

if 0<age14wk<992 then do;
do i=age14wk to 991;
    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 then do;
        thrs161=hours161*wks1;
        thrs162=hours162*wks2;
        thrs163=hours163*wks3;
        thrs164=hours164*wks4;
        thrs165=hours165*wks5;
        thrs166=hours166*wks6;
        thrs167=hours167*wks7;
        thrs1=hours1*wks1;
        thrs2=hours2*wks2;
        thrs3=hours3*wks3;
        thrs4=hours4*wks4;
        thrs5=hours5*wks5;
        thrs6=hours6*wks6;
        thrs7=hours7*wks7;
    end;
    if job1wks=-3 or job2wks=-3 or job3wks=-3 or
        job4wks=-3 or job5wks=-3 or job6wks=-3 or
        job7wks=-3 then do;
        thrs1=-3;
        tothrs14=-3;
        goto exit1;
    end;
end;
end; exit1;

if thrs161 ne -3 and thrs162 ne -3 and thrs163 ne -3
    and thrs164 ne -3 and thrs165 ne -3 and
    thrs166 ne -3 and thrs167 ne -3 and
    thrs1 ne -3 and thrs2 ne -3 and thrs3 ne -3 and thrs4
    ne -3 and thrs5 ne -3 and thrs6 ne -3 and
    thrs7 ne -3 then do;
    othrs14=thrs161+thrs162+thrs163+thrs164+thrs165+
        thrs166+thrs167+thrs1+thrs2+thrs3+thrs4+
        thrs5+thrs6+thrs7;
end;
if thrs161=-3 or thrs162=-3 or thrs163=-3 or thrs164=-3
    or thrs165=-3 or thrs166=-3 or thrs167=-3 or
    thrs1=-3 or thrs2=-3 or thrs3=-3 or thrs4=-3 or
    thrs5=-3 or thrs6=-3 or thrs7=-3 then do;
    tothrs14=-3;
end;

if wks1>0 and hrck161<0 and tothrs14>0 then
    tothrs14=hrck161;
if wks2>0 and hrck162<0 and tothrs14>0 then
    tothrs14=hrck162;

if wks3>0 and hrck163<0 and tothrs14>0 then
    tothrs14=hrck163;
if wks4>0 and hrck164<0 and tothrs14>0 then
    tothrs14=hrck164;
if wks5>0 and hrck165<0 and tothrs14>0 then
    tothrs14=hrck165;
if wks6>0 and hrck166<0 and tothrs14>0 then
    tothrs14=hrck166;
if wks7>0 and hrck167<0 and tothrs14>0 then
    tothrs14=hrck167;

if wks1>0 and hrck1<0 and tothrs14>0 then
    tothrs14=hrck1;
if wks2>0 and hrck2<0 and tothrs14>0 then
    tothrs14=hrck2;
if wks3>0 and hrck3<0 and tothrs14>0 then
    tothrs14=hrck3;
if wks4>0 and hrck4<0 and tothrs14>0 then
    tothrs14=hrck4;
if wks5>0 and hrck5<0 and tothrs14>0 then
    tothrs14=hrck5;
if wks6>0 and hrck6<0 and tothrs14>0 then
    tothrs14=hrck6;
if wks7>0 and hrck7<0 and tothrs14>0 then
    tothrs14=hrck7;

do i=1 to 7;
if age14wk<53 then do;
    if starw<0 and starw>-4 and stopw>1 then do;
        tothrs14=-3; end;
    if stopw<0 and stopw>-4 and starw<53 then do;
        tothrs14=-3; end;
end;
if age14wk<105 then do;
    if starw<0 and starw>-4 and stopw>52 then do;
        tothrs14=-3; end;
    if stopw<0 and stopw>-4 and starw<105 then do;
        tothrs14=-3; end;
end;
if age14wk<157 then do;
    if starw<0 and starw>-4 and stopw>104 then do;
        tothrs14=-3; end;
    if stopw<0 and stopw>-4 and starw<157 then do;
        tothrs14=-3; end;
end;
if age14wk<210 then do;
    if starw<0 and starw>-4 and stopw>156 then do;
        tothrs14=-3; end;
    if stopw<0 and stopw>-4 and starw<210 then do;
        tothrs14=-3; end;
end;
if age14wk<262 then do;
    if starw<0 and starw>-4 and stopw>209 then do;
        tothrs14=-3; end;
    if stopw<0 and stopw>-4 and starw<262 then do;
        tothrs14=-3; end;
end;
end;

```

## Appendix 2: Employment Variable Creation

```

if age14wk<314 then do;
  if starw<0 and starw>-4 and stopw>261 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<314 then do;
    tothrs14=-3; end;
end;
if age14wk<366 then do;
  if starw<0 and starw>-4 and stopw>313 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<366 then do;
    tothrs14=-3; end;
end;
if age14wk<418 then do;
  if starw<0 and starw>-4 and stopw>365 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<418 then do;
    tothrs14=-3; end;
end;
if age14wk<471 then do;
  if starw<0 and starw>-4 and stopw>417 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<471 then do;
    tothrs14=-3; end;
end;
if age14wk<523 then do;
  if starw<0 and starw>-4 and stopw>470 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<523 then do;
    tothrs14=-3; end;
end;
if age14wk<575 then do;
  if starw<0 and starw>-4 and stopw>522 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<575 then do;
    tothrs14=-3; end;
end;
if age14wk<627 then do;
  if starw<0 and starw>-4 and stopw>574 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<627 then do;
    tothrs14=-3; end;
end;
if age14wk<679 then do;
  if starw<0 and starw>-4 and stopw>626 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<679 then do;
    tothrs14=-3; end;
end;
if age14wk<731 then do;
  if starw<0 and starw>-4 and stopw>678 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<731 then do;
    tothrs14=-3; end;
end;
if age14wk<784 then do;
  if starw<0 and starw>-4 and stopw>730 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<784 then do;
    tothrs14=-3; end;
end;
if age14wk<836 then do;
  if starw<0 and starw>-4 and stopw>783 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<836 then do;
    tothrs14=-3; end;
end;
if age14wk<888 then do;
  if starw<0 and starw>-4 and stopw>835 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<888 then do;
    tothrs14=-3; end;
end;
if age14wk<940 then do;
  if starw<0 and starw>-4 and stopw>887 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<940 then do;
    tothrs14=-3; end;
end;
if age14wk<991 then do;
  if starw<0 and starw>-4 and stopw>939 then do;
    tothrs14=-3; end;
  if stopw<0 and stopw>-4 and starw<991 then do;
    tothrs14=-3; end;
end;
end;

/* [END CV_HOURS_WK_EVER] */

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; tothrs14=-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; tothrs14=-3;
end;

*if start year is missing;
if pubid=411 or pubid=631 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; tothrs14=-3;
end;

endsas;

```

NUMBER OF JOBS HELD DURING 19XX

Variables Created: CV\_TTL\_JOBS\_YR.80 – CV\_TTL\_JOBS\_YR.98

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.

\*\*\*\*\* 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;

```
do i=1 to 52;
  if job1wks=-3 then do; job801=-3; end;
  if job2wks=-3 then do; job802=-3; end;
  if job3wks=-3 then do; job803=-3; end;
  if job4wks=-3 then do; job804=-3; end;
  if job5wks=-3 then do; job805=-3; end;
  if job6wks=-3 then do; job806=-3; end;
  if job7wks=-3 then do; job807=-3; end;
end;
```

```
do i=1 to 52;
  if job1wks=1 then do; job801=1; end;
  if job2wks=1 then do; job802=1; end;
  if job3wks=1 then do; job803=1; end;
  if job4wks=1 then do; job804=1; end;
  if job5wks=1 then do; job805=1; end;
  if job6wks=1 then do; job806=1; end;
  if job7wks=1 then do; job807=1; end;
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 then do;
  njobs80=sum(job801, job802, job803, job804,
  job805, job806, job807);
end;
```

```
if job801=-3 or job802=-3 or job803=-3 or job804=-3
or job805=-3 or job806=-3 or job807=-3 then do;
  njobs80=-3;
end;
```

/\*\*\*\*\*\* At this point the program loops through the same code used above for 1980 for each year 1981–98, creating the variables job811 through job817 for 1981, job821 through job827 for 1982, and so on up to job981 through job987 for 1998. 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	1990	523-574
1982	105-156	1991	575-626
1983	157-209	1992	627-678
1984	210-261	1993	679-730
1985	262-313	1994	731-783
1986	314-365	1995	784-835
1987	366-417	1996	836-887
1988	418-470	1997	888-939
1989	471-522	1998	940-991 *****/

```
do i=1 to 7;
/* 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;
  if stopw>887 then do; njobs97=njobs97+1; end;
  if stopw>939 and intwk>939 then do;
    njobs98=njobs98+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;
```

## Appendix 2: Employment Variable Creation

---

```
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;
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;
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;
end;

*if start year is missing;
if pubid=411 or pubid=631 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;
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 age14. It is only created for respondents who have worked at least one week since age 14.

\*\*\*\*\* 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;

```

if age14wk>0 then do;
  do i=age14wk to 992;
    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;
  end;
end;

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

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 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 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;
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 e200=0 then do; njobs14=-4; end;
if e200=-3 then do; njobs14=-3; end;
/*if start year is missing */; if pubid=411 or pubid=631 then do; njobs14=-3; end;

endsas;

```

