NLSY97 Appendix 1:

Education Variable Creation

ENROLLMENT STATUS, HIGHEST GRADE COMPLETED, AND HIGHEST DEGREE RECEIVED

Variables Created: CV_ENROLLSTAT
CV_HGC_EVER CV_HGC_YR
CV_HIGHEST_DEGREE_EVER CV_HIGHEST_DEGREE_YR

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
R1_3500	YSCH -3500(Round1)	E9335M21-E9335M22	YSCH-9335.02.01~M–.02~M
R1_5000	YSCH-5000(Round1)	E9335Y21-E9335Y22	YSCH-9335.02.01~Y–.02~Y
E1605	YSCH-1605	E9335M31-E9335M32	YSCH-9335.03.01~M–.02~M
E16151-E16152	YSCH-1615.01–.02	E9335Y31-E9335Y32	YSCH-9335.03.01~Y–.02~Y
E16241-E16242	YSCH-1624.01–.02	E9589111-E9589114	YSCH-9589.01.01.01–.04
E16271	YSCH-1627.01	E9589121-E9589122	YSCH-9589.01.02.01–.02
E2806	YSCH-2806	E9589211-E9589216	YSCH-9589.02.01.01–.06
E2857	YSCH-2857	E9589221	YSCH-9589.02.02.01
E3061	YSCH-3061	E9589311-E9589313	YSCH-9589.03.01.01–.03
E3112	YSCH-3112	E9589321	YSCH-9589.03.02.01
E340011-E340061	YSCH-3400.01.01–.06.01	E9589411	YSCH-9589.04.01.01
E3878	YSCH-3878	E9589511	YSCH-9589.05.01.01
E4793	YSCH-4793	E9946111-E9946114	YSCH-9946.01.01.01–.04
E4795	YSCH-4795	E9946121-E9946122	YSCH-9946.01.02.01–.02
E4951	YSCH-4951	E9946211-E9946216	YSCH-9946.02.01.01–.06
E67841-E67845	YSCH-6784.01–.05	E9946221	YSCH-9946.02.02.01
E69381-E69385	YSCH-6938.01–.05	E9946311-E9946313	YSCH-9946.03.01.01–.03
E69431-E69433	YSCH-6943.01.01–.03.01	E9946321	YSCH-9946.03.02.01
E71421-E71424	YSCH-7142.01.01–.04.01	E9946411	YSCH-9946.04.01.01
E71921-E71925	YSCH-7192.01–.05	E9946511	YSCH-9946.05.01.01
E841611-E841613	YSCH-8416.01.01–.03	E199M111-E199M113	YSCH-10099.01.01.01~M–.03~M
E841621-E841622	YSCH-8416.02.01–.02	E199Y111-E199Y113	YSCH-10099.01.01.01~Y–.03~Y
E841631-E841632	YSCH-8416.03.01–.02	E199M121, E199Y121	YSCH-10099.01.02.01~M, ~Y
E841641-E841642	YSCH-8416.04.01–.02	E199M211, E199Y211	YSCH-10099.02.01.01~M, ~Y
E841651	YSCH-8416.05.01	E199M212, E199Y212	YSCH-10099.02.01.02~M, ~Y
E841661	YSCH-8416.06.01	E199M214, E199Y214	YSCH-10099.02.01.04~M, ~Y
E913111-E913113	YSCH-9131.01.01–.03	E199M215, E199Y215	YSCH-10099.02.01.05~M, ~Y
E913121-E913122	YSCH-9131.02.01–.02	E199M311-E199M312	YSCH-10099.03.01.01~M–.02~M
E913131-E913132	YSCH-9131.03.01–.02	E199Y311-E199Y312	YSCH-10099.03.01.01~Y–.02~Y
E913141-E913142	YSCH-9131.04.01–.02	E11700	YSCH-11700
E913151	YSCH-9131.05.01	E11900M, E11900Y	YSCH-11900~M, ~Y
E913161	YSCH-9131.06.01	E234501-E234504	YSCH-23450.01–.04
E9335M11-E9335M13	YSCH-9335.01.01~M–.03~M	E234601M, E234601Y	YSCH-23460.01~M, ~Y
E9335Y11-E9335Y13	YSCH-9335.01.01~Y–.03~Y	E273371-E273375	YSCH-27337.01–.05

Codes for Created Variables

Enrollment Status (CV_ENROLLSTAT)

1. not enrolled, no high school degree, no GED
2. not enrolled, GED, but no high school degree
3. not enrolled, a high school degree, no GED
4. not enrolled, some college
5. not enrolled, 2 year college graduate
6. not enrolled, 4 year college graduate
7. not enrolled, graduate degree
8. enrolled in grades 1-12
9. enrolled in a 2 year college
10. enrolled in a 4 year program
11. enrolled in a graduate program

Highest Grade Completed (CV_HGC_EVER, CV_HGC_YR)

1. none
2. ungraded
3. first
4. second
5. third
12. tenth
13. eleventh
14. twelfth
15. first year college
16. second year college

6. fourth
7. fifth
8. sixth
9. seventh
10. eighth
11. ninth
17. third year college
18. fourth year college
19. fifth year college
20. sixth year college
21. seventh year college
22. eighth year college

Highest Degree Received (CV_HIGHEST_DEGREE_EVER, CV_HIGHEST_DEGREE_YR)
0. none
1. GED
2. high school diploma
3. junior college or 2-year associate degree
4. bachelor's degree (B.A., B.S., or unspecified)
5. master's degree
6. doctoral degree
7. professional degree (M.D., L.L.D., D.D.S., J.D.)

This program first creates an enrollment status variable for each respondent. It then identifies the highest grade completed by the respondent as of the round 2 interview date and as of June 30, 1998. Finally, it determines the highest degree or diploma received by the respondent as of the same two dates. In round 2, no respondent had completed a degree higher than a high school diploma and/or GED. Therefore, in practice, the range of actual values is limited to [0,2].

```
/* Initialize variables.  */
ENROLL=-16;    /* Indicator of whether respondent currently enrolled */
ENCAT=-16;     /* Enrollment status variable (including educational attainment) */
NODIP=0;       /* Indicator of whether respondent does not have a diploma (high school or GED) */
               /* Note: must be filled in from Round 1 */
HS=0;          /* Indicator of whether respondent is currently attending grades 1 to 12 */
COLLEGE=0;     /* Indicator of whether respondent is current attending 2 or 4 year college */
FLAGCOL=0;     /* Indicator to flag respondents currently attending grades 1 to 12 and enrolled in college */
FLAGLOOP=0;    /* Indicator to flag respondents going through old school re-enrollment loop twice */

/* Defining arrays for use later in the program */
array E1615 E16151 E16152;
array E1624 E16241 E16242;
array E3400 E340011 E340021 E340031 E340041 E340051 E340061;
array E5458 E54581 E54582 E54583 E54584 E54585;
array E6784 E67841 E67842 E67843 E67844 E67845;
array E6938 E69381 E69382 E69383 E69384 E69385;
array E6943 E69431 E69432 E69433;
array E7142 E71421 E71422 E71423 E71424;
array E7192 E71921 E71922 E71923 E71924 E71925;
array E84161 E841611 E841612 E841613;
array NE84161 NE841611 NE841612 NE841613;
array ME84161 ME841611 ME841612 ME841613;
array E84162 E841621 E841622;
array NE84162 NE841621 NE841622;
array ME84162 ME841621 ME841622;
array E84163 E841631 E841632;
array NE84163 NE841631 NE841632;
array ME84163 ME841631 ME841632;
array E84164 E841641 E841642;
array E91311 E913111 E913112 E913113;
array E91312 E913121 E913122;
array E91313 E913131 E913132;
array E91314 E913141 E913142;
array E9335M1 E9335M11 E9335M12 E9335M13;
array E9335Y1 E9335Y11 E9335Y12 E9335Y13;
array E9335M2 E9335M21 E9335M22;
array E9335Y2 E9335Y21 E9335Y22;
```

```
array E9335M3 E9335M31 E9335M32;
array E9335Y3 E9335Y31 E9335Y32;
array E958911 E9589111 E9589112 E9589113 E9589114;
array NE95811 NE958111 NE958112 NE958113;
array ME95811 ME958111 ME958112 ME958113;
array E958912 E9589121 E9589122;
array E958921 E9589211 E9589212 E9589213 E9589214 E9589215 E9589216;
array NE95821 NE958211 NE958212 NE958213 NE958214 NE958215 NE958216;
array ME95821 ME958211 ME958212 ME958213 ME958214 ME958215 ME958216;
array E958931 E9589311 E9589312 E9589313;
array NE95831 NE958311 NE958312;
array ME95831 ME958311 ME958312;
array E994611 E9946111 E9946112 E9946113 E9946114;
array E994612 E9946121 E9946122;
array E994621 E9946211 E9946212 E946213 E9946214 E9946215 E9946216;
array E994631 E9946311 E9946312 E9946313;
array E199M11 E199M111 E199M112 E199M113;
array E199Y11 E199Y111 E199Y112 E199Y113;
array E199M21 E199M211 E199M212 E199M214 E199M215 E199M216;
array E199Y21 E199Y211 E199Y212 E199Y214 E199Y215 E199Y216;
array E199M31 E199M311 E199M312;
array E199Y31 E199Y311 E199Y312;
array E23450 E234501 E234502 E234503 E234504;
array E27337 E273371 E273372 E273373 E273374 E273375;
```

/***** ENROLLMENT STATUS *******/**

```
/*Begin by defining the conditions when a respondent is enrolled. Note that summer vacations are included as part
of the enrollment period. */

/* Continuously enrolled in old school since dli */
if E4795=1 then ENROLL=1;
if E4793=1 then ENROLL=1;

/* Not continuously enrolled and not enrolled in any other schools since */
if E4795=0 and E4951=0 then ENROLL=0;

/* Not continuously enrolled in old school since dli, have not re-enrolled in old school since dli and do not report
any new schools since dli */
if E1605=-2 then ENROLL=-2;
if E4795=0 and E1605=0 and E4951=0 then ENROLL=0;
if E4793=0 and E1605=0 and E4951=0 then ENROLL=0;
if E3878=0 then ENROLL=0;

/* Not continuously enrolled in old school since dli, have reenrolled in old school a maximum of 2 times, have not
reported any other schools attended since */
if E4793=0 and E1605=1 and E16151=1 then ENROLL=1;
if E4793=0 and E1605=1 and E16152=1 then ENROLL=1;

/* Not continuously enrolled in old school since dli, have received degree and/or completed course work at old
school (i.e. YSCH-1600=1), and has not reported new school since */
if E4793=0 and E4951=0 and E1605=-4 then ENROLL=0;

/* RE-ENROLLMENT IN OLD SCHOOL.  Note that respondents can only go through the re-enrollment loop
twice and it is not possible to determine which individuals re-enrolled a third time.  Individuals who have gone
through the loop twice and were not continuously enrolled up to the interview during the last enrollment spell and
who do not report any new schools will be coded as not enrolled. */
```

```
/* Flag those who go through the re-enrollment loop twice */
if E16241=1 then FLAGLOOP=1;
if E16242=1 then FLAGLOOP=1;

if (E4795=0 and E1605=1) then do;
    if E16152=-2 then ENROLL=-2;
    if E16151=1 then ENROLL=1;   /* Continuously enrolled in old school, first time through loop. */
    if E16151=0 and E16241=0 and E16271=0 and E4951=0 then ENROLL=0;  /* Not continuously enrolled in
            old school, first time through loop, no other re-enrollments at old school and no new schools reported. */
    if E16152=0 and E16242=1 and E4951=0 then ENROLL=0;  /* Not continuously enrolled in old school, second
            time through loop and no new schools reported. */
    if E16152=1 then ENROLL=1; /* Continuously enrolled in old school, second time through loop. */
end;

/* ENROLLMENT IN NEW SCHOOLS */
if (E4951=1 or E3878=1) then do;
    do I=1 to 5;  /* Allow for up to five new schools to be recorded. */
        if E6784(I)=1 and E7192(I)=0 then ENROLL=1;  /* Continuously enrolled in new school, no other new
                schools reported. */
        if E6784(I)=0 and E7192(I)=0 then ENROLL=0;  /* Not continuously enrolled in new school, did not re-
                enroll in new school, no other new schools reported. */
/* ENROLLMENT PERIODS IN NEW SCHOOLS */
        if (E6784(I)=0 and E6938(I)=1) then do;
        if E6943(I)=1 and E7192(I)=0 then ENROLL=1; /* Continuously re-enrolled in new school, no other new
                schools reported. */
        if E6943(I)=0 and E7142(I)=0 and E7192(I)=0 then ENROLL=0; /* Not continuously re-enrolled in new
                school, not re-enrolled in new school for a subsequent spell, no other new schools reported. */
        end;
    end;
end;

/* ENROLLMENT STATUS CATEGORIES */
/* No respondents have GED at dli coded as don't know/refused and no respondents have between grades 9 and 12,
are not currently attending and do not already have a GED by dli.  Therefore, the GED question is skipped in
Round 2.  As a result, GED status must be filled in using variables from Round 1 and NODIP must be appropriately
updated as well. */

/*Need to fill in NODIP from Round 1 */
if E11700<1 then NODIP=1;

/* NOT ENROLLED */
if (ENROLL=0) then do;
        if NODIP=1 then ENCAT=1;                /* No high school degree, no GED */
        if E11700=1 then ENCAT=3;               /* High school degree */
        if 13<=E2857<=20 then ENCAT=4;          /* Highest grade attended is at least 1 year of college */
        if (E234501=1 or E234502=1 or E234503=1 or E234504=1) then ENCAT=5;      /* Received junior
                        college or 2 year associate degree (need to check each school, allow for up to 5 new
                        schools) */
        if (E234501=3 or E234502=3 or E234503=3 or E234504=3) then ENCAT=6;      /* Received bachelor's
                        degree (need to check each school, allow for up to 5 new schools) */
        do I=1 to 4;        /* Received master's, doctoral or professional degree (need to check each school, allow for
                        up to 5 new schools) */
                if 4<=E23450(I)<=6 then ENCAT=7;
        end;
end;
```

```
/* ENROLLED */
if (ENROLL=1) then do;
        if (E273372=-2 or E273373=-2)then ENCAT=-2;      /* Missing values */
        if (E273372=-2 or E273373=-2)then COLLEGE=-2;
        if (E273372=-2 or E273373=-2)then FLAGCOL=-2;
        if 1<=E2857<=12 and NODIP=1 then ENCAT=8;      /* Highest grade attended as of today less than or
                        equal to 12th grade and don't have a diploma/GED */
        if (E273371=1 or E273372=1 or E273373=1 or E273374=1 or E273375=1) and (13<=E2857<95) then
                        ENCAT=9;        /* Working towards an Associate/Junior College or two-year associate
                        degree (allow up to five schools) and current grade is greater than high school */
        if (E273371=3 or E273372=3 or E273373=3 or E273374=3 or E273375=3) and (13<=E2857<95) then
                        ENCAT=10;       /* Working towards a Bachelor's degree and current grade is greater
                        than high school */
        do I=1 to 5;        /* Working towards a graduate degree and current grade is greater than high school */
                if 4<=E27337(I)<=6 and 13<=E2857<95 then ENCAT=11;
        end;
end;

/* The following is for respondents who are enrolled, have no evidence of a high school diploma, yet reported that
highest grade attended was 1st year of college. Since there is no evidence these people have a diploma and have not
attended a new school, they will be placed under enrolled in grades 1-12. */

if ENROLL=1 and E2857=13 and NODIP=1 then ENCAT=8;

/* The following accounts for the respondents who reported a new school since dli.  It is assumed that for
respondents who reported working towards a master's or PhD (YSCH-27337) that also report less than 2 years of
college are actually still in undergraduate college. Therefore they were placed in the college and enrolled ENCAT
variable. */

if (ENROLL=1) then do;
        do I=1 to 5;
                if 0<=E5458(I)<=3 then ENCAT=8;
                if E5458(I)=4 then ENCAT=9;
                if E5458(I)=5 then ENCAT=10;
        end;
end;

/* The following is for respondents who have a high school diploma, a valid date for the diploma, have not enrolled
in a new school since dli, and report the highest grade completed being 12. They have been decided to be categorized
as not enrolled with a high school diploma (ENCAT=3) despite the fact that these respondents claim to be
continuously enrolled since dli. Since there is no evidence of college from the respondent, they have been grouped as
not enrolled. */

if E11700=1 and E11900M>0 and E11900Y>0 and E4951=0 and E3112=12 then ENCAT=3;

/* The following is for respondents who report the highest grade attended as ungraded (YSCH-2857=95), show no
evidence of a high school diploma, were skipped out of the homeschool question, and are enrolled. Again, since there
is no evidence of a high school diploma, they are regarded as enrolled between grades 1 and 12 */

if E2857=95 and NODIP=1 and ENROLL=1 then ENCAT=8;

/* Special hand edited case */
if PUBID=5998 then ENCAT=1;

/* Flag individuals who are enrolled in grades 1 to 12 and are enrolled in 2 or 4 year colleges (could be getting credits
towards GED).  For created enrollment status variable above, the flagged individuals will fall into category 8. */
```

```
if ENROLL=1 and 1<=E2857<=12 and NODIP=1 then HS=1;
if ENROLL=1 and (E273371=1 or E273372=1 or E273373=1 or E273374=1 or E273375=1 or E273371=3 or
E273372=3 or E273373=3 or E273374=3 or E273375=3) then COLLEGE=1;
if HS=1 and COLLEGE=1 then FLAGCOL=1;
```

/***** HIGHEST GRADE COMPLETED AS OF THE SURVEY DATE *******/**

```
/* Start with highest grade completed at dli and update. */

if -4<R1_5000<0 then GRSURV=R1_5000; /* Missing values */

/* Highest grade at dli attended is the starting grade for GRSURV */
if R1_5000=0 then GRSURV=1;
if R1_5000=95 then GRSURV=2;
if 0<R1_5000<95 then GRSURV=R1_5000+2;

/* The following accounts for corrections in highest grade completed from Round 1. YSCH-3061 is the check that
makes sure highest grade completed from Round 1 is correct. If yes, repsondent gave true highest grade completed at
Round 1 interview date at YSCH-3061. The same is repeated for R1_3500, which uses YSCH-3104 as the check for
Round1 current grade. */

if E3061>-1 then GRSURV=E3061+2;
if E3104>-1 then R1_3500=E3104;

/* Due to conflicting information, highest grade completed at survey is no longer collected at E3112. Using E3112
produced several people saying they completed 9th grade at E3112 but then saying they didn't complete 9th at
E913111 (and other questions). So, GRSURV will start at R1_5000, which is highest grade completed at Round 1.
It will be updated if and only if the respondent completes another grade and gives a valid date. Note that don't
knows or refusals of the month of completion will still be considered valid as completing that grade. Variables that
begin with "N" designate new variables created so that the original values won't be affected in the list file. */

NR1_3500=R1_3500;

/* Submit the negativity conditions */
do I=1 to 3; if -4<E91311(I)<0 then GRSURV=E91311(I); end;
do I=1 to 2;
    if -4<E91312(I)<0 then GRSURV=E91312(I);
    if -4<E91313(I)<0 then GRSURV=E91313(I);
    if -4<E91314(I)<0 then GRSURV=E91314(I);
end;
if -4<E913151<0 then GRSURV=E913151;
if -4<E913161<0 then GRSURV=E913161;

do I=1 to 3; if -4<E84161(I)<0 then GRSURV=E84161(I); end;
do I=1 to 2;
    if -4<E84162(I)<0 then GRSURV=E84162(I);
    if -4<E84163(I)<0 then GRSURV=E84163(I);
    if -4<E84164(I)<0 then GRSURV=E84164(I);
end;
if -4<E841651<0 then GRSURV=E841651;
if -4<E841661<0 then GRSURV=E841661;
do I=1 to 4; if -4<E994611(I)<0 then GRSURV=E994611(I); end;
do I=1 to 2; if -4<E994612(I)<0 then GRSURV=E994612(I); end;
do I=1 to 6; if -4<E994621(I)<0 then GRSURV=E994621(I); end;
if -4<E9946221<0 then GRSURV=E9946221;
do I=1 to 3; if -4<E994631(I)<0 then GRSURV=E994631(I); end;
if -4<E9946321<0 then GRSURV=E9946321;
```

```
if -4<E9946411<0 then GRSURV=E9946411;
if -4<E9946511<0 then GRSURV=E9946511;
do I=1 to 4; if -4<E958911(I)<0 then GRSURV=E958911(I); end;
do I=1 to 2; if -4<E958912(I)<0 then GRSURV=E958912(I); end;
do I=1 to 6; if -4<E958921(I)<0 then GRSURV=E958921(I); end;
if -4<E9589221<0 then GRSURV=E9589221;
do I=1 to 3; if -4<E958931(I)<0 then GRSURV=E958931(I); end;
if -4<E9589321<0 then GRSURV=E9589321;
if -4<E9589411<0 then GRSURV=E9589411;
if -4<E9589511<0 then GRSURV=E9589511;

/* Normally, R1_3500 or (if R1_3500 is incorrect) E2806 is used for questions about if grade completed (E913111)
and when (E9335M11). E841611 is skipped in these cases. When E841611 is not skipped, then it must take the
place of R1_3500. */

if E841611>-4 then NR1_3500=E841611;

NE841611=E841611; NE841612=E841612; NE841613=E841613;
NE841621=E841621; NE841622=E841622;
NE841631=E841631; NE841632=E841632;
NE841641=E841641;
NE841651=E841651;
NE841661=E841661;

/* Due to character limits on naming variables, created "N" variables will not be exactly the same name as the
originals */ /* The "9" is dropped to limit the name to 8 letters */
NE958111=E9589111;  NE958112=E9589112; NE958113=E9589113;
NE958121=E9589121;
NE958211=E9589211; NE958212=E9589212;  NE958214=E9589214; NE958215=E9589215;
NE958311=E9589311; NE958312=E9589312;

/* Assign missing values for grades not completed, invalid year information, or a non-interview case. */
if E913111<1 or E9335M11<-4 or E9335Y11<=0 then NR1_3500=.;
do I=2 to 3; if E91311(I)<1 or E9335M1(I)<-4 or E9335Y1(I)<=0 then NE84161(I)=.; end;
do I=1 to 2;
    if E91312(I)<1 or E9335M2(I)<-4 or E9335Y2(I)<=0 then NE84162(I)=.;
    if E91313(I)<1 or E9335M3(I)<-4 or E9335Y3(I)<=0 then NE84163(I)=.;
end;
if E913141<1 or E9335M41<-4 or E9335Y41<=0 then NE841641=.;
if E913151<1 or E9335M51<-4 or E9335Y51<=0 then NE841651=.;
if E913161<1 or E9335M61<-4 or E9335Y61<=0 then NE841661=.;

do I=1 to 3; if E994611(I)<1 or E199M11(I)<-4 or E199Y11(I)<=0 then NE95811(I)=.; end;
if E9946121<1 or E199M121<-4 or E199Y121<=0 then NE958121=.;
do I=1 to 2; if E994621(I)<1 or E199M21(I)<-4 or E199Y21(I)<=0 then NE95821(I)=.; end;
do I=4 to 5; if E994621(I)<1 or E199M21(I)<-4 or E199Y21(I)<=0 then NE95821(I)=.; end;
do I=1 to 2; if E994631(I)<1 or E199M31(I)<-4 or E199Y31(I)<=0 then NE95831(I)=.; end;

/* Define "MAXGRAD1" as highest grade in the following grade update loop */
MAXGRAD1=max (NR1_3500, NE841612, NE841613, NE841621, NE841622, NE841631, NE841632,
NE841641, NE841651, NE841661, NE958111, NE958112, NE958113, NE958121, NE958211, NE958212,
NE958214, NE958215, NE958311, NE958312);

/* Need to limit maxgrade so that it only changes GRSURV if it is not a valid skip (-4) or an ungraded year (95) */
if -4<MAXGRAD1<95 then GRSURV=MAXGRAD1+2;
```

/***** HIGHEST DEGREE RECEIVED AS OF THE SURVEY DATE *******/**

NOTE: Only available for those who were enrolled since the last interview. For respondents who not enrolled since the date of last interview, need highest degree as reported in Round 1 (skip to employment sect. at YSCH-3960).

```
/* Initialize everyone to zero */

DEGSURV=0;

/*  if GED=1 then DEGSURV=1  GED */
if E11700=1 then DEGSURV=2; /* High school diploma */
if (E234501=1 or E234502=1 or E234503=1 or E234504=1) then DEGSURV=3; /* Junior college or two-year
        associate degree */
if (E234501=3 or E234502=3 or E234503=3 or E234504=3) then DEGSURV=4; /* Bachelor's degree */
if (E234501=4 or E234502=4 or E234503=4 or E234504=4) then DEGSURV=5; /* Master's degree */
if (E234501=5 or E234502=5 or E234503=5 or E234504=5) then DEGSURV=6; /* Doctoral degree */
if (E234501=6 or E234502=6 or E234503=6 or E234504=6) then DEGSURV=7; /* Professional degree */
```

/***** HIGHEST GRADE COMPLETED AS OF JUNE 30, 1998 *******/**

```
/* Start with the highest grade completed as of the date of the last interview and update. */

/* Initialize everyone to -16 to test missing values */
GRJUNE=-16;

if -4<R1_5000<0 then GRJUNE=R1_5000; /* Missing values */

/* Highest grade at dli attended is the starting grade for GRJUNE */
if R1_5000=0 then GRJUNE=1;
if R1_5000=95 then GRJUNE=2;
if 0<R1_5000<95 then GRJUNE=R1_5000+2;

/* If the respondent answers E3061, then GRJUNE is the value for E3061 */
if E3061>-1 then GRJUNE=E3061+2;

/* Added the condition that if the year is less than 1998 and the month of the grade completed (E9335M(I)) is -2
(don't know) does not mean GRJUNE=-2. However if the year is 1998 and the month is a -2 (don't know), then
GRJUNE should be -2.  If the grade the respondent was enrolled in since the last interview was completed before
June 30, 1998, then update */
MR1_3500=R1_3500;

/* Missing values */
do I=1 to 3;
    if -4<E9335M1(I)<0 and E9335Y1(I)=1998 then GRJUNE=E9335M1(I);
    if -4<E9335Y1(I)<0 then GRJUNE=E9335Y1(I);
end;
do I=1 to 2;
    if -4<E9335M2(I)<0 and E9335Y2(I)=1998 then GRJUNE=E9335M2(I);
    if -4<E9335Y2(I)<0 then GRJUNE=E9335Y2(I);
    if -4<E9335M3(I)<0 and E9335Y3(I)=1998 then GRJUNE=E9335M3(I);
      if -4<E9335Y3(I)<0 then GRJUNE=E9335Y3(I);
end;
if -4<E9335M41<0 and E9335Y41=1998 then GRJUNE=E9335M41;
if -4<E9335Y41<0 then GRJUNE=E9335Y41;
if -4<E9335M51<0 and E9335Y51=1998 then GRJUNE=E9335M51;
if -4<E9335Y51<0 then GRJUNE=E9335Y51;
if -4<E9335M61<0 and E9335Y61=1998 then GRJUNE=E9335M61;
```

```
if -4<E9335Y61<0 then GRJUNE=E9335Y61;

do I=1 to 3; if -4<E91311(I)<0 then GRJUNE=E91311(I); end;
do I=1 to 2;
    if -4<E91312(I)<0 then GRJUNE=E91312(I);
    if -4<E91313(I)<0 then GRJUNE=E91313(I);
    if -4<E91314(I)<0 then GRJUNE=E91314(I);
end;
if -4<E913151<0 then GRJUNE=E913151;
if -4<E913161<0 then GRJUNE=E913161;

do I=1 to 3; if -4<E84161(I)<0 then GRJUNE=E84161(I); end;
do I=1 to 2;
    if -4<E84162(I)<0 then GRJUNE=E84162(I);
    if -4<E84163(I)<0 then GRJUNE=E84163(I);
    if -4<E84164(I)<0 then GRJUNE=E84164(I);
end;
if -4<E841651<0 then GRJUNE=E841651;
if -4<E841661<0 then GRJUNE=E841661;

do I=1 to 3;
    if -4<E199M11(I)<0 and E199Y11(I)=1998 then GRJUNE=E199M11(I);
    if -4<E199Y11(I)<0 then GRJUNE=E199Y11(I);
end;
if -4<E199M121<0 and E199Y121=1998 then GRJUNE=E199M121;
if -4<E199Y121<0 then GRJUNE=E199Y121;
do I=1 to 2;
    if -4<E199M21(I)<0 and E199Y21(I)=1998 then GRJUNE=E199M21(I);
    if -4<E199Y21(I)<0 then GRJUNE=E199Y21(I);
end;
do I=4 to 5;
    if -4<E199M21(I)<0 and E199Y21(I)=1998 then GRJUNE=E199M21(I);
    if -4<E199Y21(I)<0 then GRJUNE=E199Y21(I);
end;
do I=1 to 2;
    if -4<E199M31(I)<0 and E199M31(I)=1998 then GRJUNE=E199M31(I);
    if -4<E199Y31(I)<0 then GRJUNE=E199Y31(I);
end;

do I=1 to 4; if -4<E994611(I)<0 then GRJUNE=E994611(I); end;
do I=1 to 2; if -4<E994612(I)<0 then GRJUNE=E994612(I); end;
do I=1 to 6; if -4<E994621(I)<0 then GRJUNE=E994621(I); end;
if -4<E9946221<0 then GRJUNE=E9946221;
do I=1 to 3; if -4<E994631(I)<0 then GRJUNE=E994631(I); end;
if -4<E9946321<0 then GRJUNE=E9946321;
if -4<E9946411<0 then GRJUNE=E9946411;
if -4<E9946511<0 then GRJUNE=E9946511;

do I=1 to 4; if -4<E958911(I)<0 then GRJUNE=E958911(I); end;
do I=1 to 2; if -4<E958912(I)<0 then GRJUNE=E958912(I); end;
do I=1 to 6; if -4<E958921(I)<0 then GRJUNE=E958921(I); end;
if -4<E9589221<0 then GRJUNE=E9589221;
do I=1 to 3; if -4<E958931(I)<0 then GRJUNE=E958931(I); end;
if -4<E9589321<0 then GRJUNE=E9589321;
if -4<E9589411<0 then GRJUNE=E9589411;
if -4<E9589511<0 then GRJUNE=E9589511;
```

```
/* Normally, R1_3500 is used for questions about grade completed (E913111) and when (E9335M11). E841611 is
skipped in these cases. When E841611 is not skipped, then it must take the place of R1_3500. */
if E841611>-4 then MR1_3500=E841611;

ME841611=E841611; ME841612=E841612; ME841613=E841613;
ME841621=E841621; ME841622=E841622;
ME841631=E841631; ME841632=E841632;
ME841641=E841641;
ME841651=E841651;
ME841661=E841661;

/* Again, due to character limitations, created "M" variable names will not be exactly the same as the originals */
ME958111=E9589111; ME958112=E9589112; ME958113=E9589113;
ME958121=E9589121;
ME958211=E9589211; ME958212=E9589212; ME958214=E9589214; ME958215=E9589215;
ME958311=E9589311; ME958312=E9589312;

/* The following lines only count the maximum grade completed if it comes before 7/98. The first line of each loop
accounts for grades completed in 1998. During 1998, a valid month between January and June must be given to
count towards GRJUNE. However, before 1998 any month can be given, including a don't know (-2), a refusal (-1),
or an invalid skip (-3).*/
if (E913111<1 or E9335M11<=0 or E9335M11>6) and E9335Y11=1998 then MR1_3500=.;
if E913111<1 or E9335Y11>1998 or E9335Y11<=0 then MR1_3500=.;
do I=2 to 3;
    if (E91311(I)<1 or E9335M1(I)<=0 or E9335M1(I)>6) and E9335Y1(I)=1998 then ME84161(I)=.;
    if E91311(I)<1 or E9335Y1(I)>1998 or E9335Y1(I)<=0 then ME84161(I)=.;
end;
do I=1 to 2;
    if (E91312(I)<1 or E9335M2(I)<=0 or E9335M2(I)>6) and E9335Y2(I)=1998 then ME84162(I)=.;
    if E91312(I)<1 or E9335Y2(I)>1998 or E9335Y2(I)<=0 then ME84162(I)=.;
    if (E91313(I)<1 or E9335M3(I)<=0 or E9335M3(I)>6) and E9335Y3(I)=1998 then ME84163(I)=.;
    if E91313(I)<1 or E9335Y3(I)>1998 or E9335Y3(I)<=0 then ME84163(I)=.;
end;
if (E913141<1 or E9335M41<=0 or E9335M41>6) and E9335Y41=1998 then ME841641=.;
if E913141<1 or E9335Y41>1998 or E9335Y41<=0 then ME841641=.;
if (E913151<1 or E9335M51<=0 or E9335M51>6) and E9335Y51=1998 then ME841651=.;
if E913151<1 or E9335Y51>1998 or E9335Y51<=0 then ME841651=.;
if (E913161<1 or E9335M61<=0 or E9335M61>6) and E9335Y61=1998 then ME841661=.;
if E913161<1 or E9335Y61>1998 or E9335Y61<=0 then ME841661=.;

do I=1 to 3;
    if (E994611(I)<1 or E199M11(I)<=0 or E199M11(I)>6) and E199Y11(I)=1998 then ME95811(I)=.;
    if E994611(I)<1 or E199Y11(I)>1998 or E199Y11(I)<=0 then ME95811(I)=.;
end;
if (E9946121<1 or E199M121<=0 or E199M121>6) and E199Y121=1998 then ME958121=.;
if E9946121<1 or E199Y121>1998 or E199Y121<=0 then ME958121=.;
do I=1 to 2;
    if (E994621(I)<1 or E199M21(I)<=0 or E199M21(I)>6) and E199Y21(I)=1998 then ME95821(I)=.;
    if E994621(I)<1 or E199Y21(I)>1998 or E199Y21(I)<=0 then ME95821(I)=.;
end;
do I=4 to 5;
    if (E994621(I)<1 or E199M21(I)<=0 or E199M21(I)>6) and E199Y21(I)=1998 then ME95821(I)=.;
    if E994621(I)<1 or E199Y21(I)>1998 or E199Y21(I)<=0 then ME95821(I)=.;
end;
do I=1 to 2;
    if (E994631(I)<1 or E199M31(I)<=0 or E199M31(I)>6) and E199Y31(I)=1998 then ME95831(I)=.;
    if E994631(I)<1 or E199Y31(I)>1998 or E199Y31(I)<=0 then ME95831(I)=.;
```

```
end;

/* Define "MAXGRAD2" as highest grade in the following grade update loop */
MAXGRAD2=max (MR1_3500, ME841612, ME841613, ME841621, ME841622, ME841631, ME841632,
ME841641, ME841651, ME841661, ME958111, ME958112, ME958113, ME958121, ME958211, ME958212,
ME958214, ME958215, ME958311, ME958312);

/* Need to limit maxgrade so that it only changes GRSURV if it is not a valid skip (-4) or an ungraded year (95) */
if -4<MAXGRAD2<95 then GRJUNE=MAXGRAD2+2;
```

/***** HIGHEST DEGREE COMPLETED AS OF JUNE 30, 1998 *******/**

```
/* Initialize everyone to zero */
DEGJUNE=0;

/*GEDM and GEDY are month and year the GED was obtained. These variables must be added from Round 1.*/

/*  if GED=1 and 0<GEDM<=7 and 0<GEDY<=1998 then DEGJUNE=1  GED */
if E11700=1 and 0<E11900M<=7 and 0<E11900Y<=1998 then DEGJUNE=2;  /* High school diploma */

/* No respondent reported more than one degree since the last interview */

if E234501=1 and 0<E234601M<7 and 0<E234601Y<=1998 then DEGJUNE=3; /* Junior college or 2-year
          associate degree */
if E234501=3 and 0<E234601M<7 and 0<E234601Y<=1998 then DEGJUNE=4; /* Bachelor's degree */
if E234501=4 and 0<E234601M<7 and 0<E234601Y<=1998 then DEGJUNE=5; /* Master's degree */
if E234501=5 and 0<E234601M<7 and 0<E234601Y<=1998 then DEGJUNE=6; /* Doctoral degree */
if E234501=6 and 0<E234601M<7 and 0<E234601Y<=1998 then DEGJUNE=7; /* Professional degree */

if 95>GRSURV>0 and 95>E3112>0 then do;
  SURVEY=E3112+2; DIFF=GRSURV-SURVEY;
end;
miss=0;
if DIFF ne 0 then miss=1;

/* Account for non-interview cases */
if E2857=-5 then do; encat=-5; grjune=-5; grsurv=-5; degsurv=-5; degjune=-5; end;

/* GED hand edits */
if pubid in (137,4485,7949,8303) then do; degsurv=1; degjune=1; end;

if pubid in (137,4485,8303) then do; encat=2; end;

/* Hand edits for respondents with a HS diploma in Round1 */
if pubid in (64,995,1959,4163,7478,8148) then do; degsurv= ; degjune= ; end;

if pubid in (995,7478) then do; encat=3; end;

/* Hand edit for case with R1_3500=R1_5000=-4 */
if pubid=3178 then do; GRSURV=12; GRJUNE=12; end;

endsas;
```

CURRENT OR MOST RECENT SCHOOL PRIVATE OR PAROCHIAL

Variables Created: CV_SCHOOL_TYPE

Variables Used

Name in Program	Question Name on CD
S3400R1	YSCH-3400
PUBID	PUBID
TYPE1–TYPE6	NEWSCHOOL_TYPE.01–.06
SPERIOD1–SPERIOD6	NEWSCHOOL_PERIODS.01–.06
SSTOP1M1–SSTOP1M6	NEWSCHOOL_STOP1.01~M–.06~M
SSTOP1Y1–SSTOP1Y6	NEWSCHOOL_STOP1.01~Y–.06~Y
SSTOP2M1–SSTOP2M4	NEWSCHOOL_STOP2.01~M–.04~M
SSTOP2Y1–SSTOP2Y4	NEWSCHOOL_STOP2.01~Y-.04~Y
SSTOP3M1, SSTOP3Y1	NEWSCHOOL_STOP3.01~M, ~Y
SCODE1–SCODE6	NEWSCHOOL_SCHCODE.01–.06
SINTERV1–SINTERV6	NEWSCHOOL_INTERVIEW.01–.06

Codes for Created Variable

1 = Public school
2 = Private, not parochial
3 = Parochial
4 = Other

This program creates a variable indicating whether the respondent's current or most recent school is public, private, parochial, or other. The program first determines which school is the most recent school (non-college) by examining the end dates for different enrollment periods and then picks up the school type from round 1 or round 2 for the most recent school. Users should note that school type is defined only for respondents attending grades 1-12.

```
/* Create the array variables.*/
* # of enrollment periods;
        array SPERIOD SPERIOD1-SPERIOD6;
* ending date for each period;
        array SSTOP1M SSTOP1M1-SSTOP1M6;
        array SSTOP1Y SSTOP1Y1-SSTOP1Y6;
        array SSTOP2M SSTOP2M1-SSTOP2M6;
        array SSTOP2Y SSTOP2Y1-SSTOP2Y6;
        array SSTOP3M SSTOP3M1-SSTOP3M6;
        array SSTOP3Y SSTOP3Y1-SSTOP3Y6;
* school code, i.e. elementary school, high school or college;
        array SCODE SCODE1-SCODE6;
* indicate if the school info. is from round 1 or round 2;
        array SINTERV SINTERV1-SINTERV6;
*school type for round 2;
        array TYPE TYPE1-TYPE6;
* ending time for each school;
        array SRECM SRECM1-SRECM6;
        array SRECY SRECY1-SRECY6;
*school length;
        array SLENGTH SLENGTH1-SLENGTH6;

/************************* Step 1).find the most recent school. **************************/
do I=1 to 6;
SRECM[I]=-4;
SRECY[I]=-4;
```

```
if SPERIOD[I]=1 and SCODE[I] in (1,2,3) then do;
        SRECM[I]=SSTOP1M[I]; SRECY[I]=SSTOP1Y[I];
end;

if SPERIOD[I]=2 and SCODE[I] in (1,2,3) then do;
        SRECM[I]=SSTOP2M[I]; SRECY[I]=SSTOP2Y[I];
end;

if SPERIOD[I]=3 and SCODE[I] in (1,2,3) then do;
        SRECM[I]=SSTOP3M[I]; SRECY[I]=SSTOP3Y[I];
end;

if SRECM[I] gt 0 and SRECY[I] gt 0 then SLENGTH[I]=12*(SRECY[I]-1990)+SRECM[I];
else SLENGTH[I]=-4;
end;

MAXLENG=MAX(SLENGTH1,SLENGTH2, SLENGTH3, SLENGTH4, SLENGTH5, SLENGTH6);
if TYPE1=-5 then MAXLENG=-5;

/************** step 2).decide the school type from round 1 or round 2 information.**************/
SLTYPE1=-4;
do I=1 to 6;
if MAXLENG>0 and SLENGTH[I]=MAXLENG then do;
        if SINTERV[I]=1 then SLTYPE1=S3400R1;
        else if SINTERV[I]=2 then SLTYPE1=TYPE[I];
        end;
end;

if SCODE1 in (1,2,3) then do;
        if SPERIOD2=-4 and SPERIOD3=-4 and SPERIOD4=-4 and SPERIOD5=-4 and SPERIOD6=-4 then do;
        if SINTERV1=1 then SLTYPE1=S3400R1;
        if SINTERV1=2 then SLTYPE1=TYPE1;
        end;/* If the respondent only goes to one school in R2,input that school type.*/
end;

if MAXLENG=-4 and SCODE1 in (4,5)  then SLTYPE=S3400R1;
if SPERIOD1=-4 and SPERIOD2=-4 and SPERIOD3=-4 and SPERIOD4=-4 and SPERIOD5=-4 and
SPERIOD6=-4 then SLTYPE=S3400R1;

/****************************** Recode the type variable.************************************/
SLTYPE=-4;
if SLTYPE1=1 then SLTYPE=1;
if SLTYPE1=5 then SLTYPE=2;
if SLTYPE1 in (3,4) then SLTYPE=3;
if SLTYPE1=2 or SLTYPE1>=6 then SLTYPE=4;
if -4<SLTYPE1<0 then SLTYPE=-3;
if TYPE1=-5 then SLTYPE=-5;


/* if one of the school code is dk or refused, input the school type as -3.*/
do I=1 to 6;
          if SPERIOD[I] NE -4 and SCODE[I] in (0,-1,-2,-3) then SLTYPE=-3;
end;

endsas;
```

DATE RECEIVED DIPLOMA OR DEGREE

Variables Created: CV_GED
CV_HS_DIPLOMA

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
GEDR1	CV_GED	S13500M2, S13500Y2	YSCH-13500~M, ~Y
HSR1	CV_HS_DIPLOMA	S11700R2	YSCH-11700R2
S13300R2	YSCH-13300	S11900M2, S11900Y2	YSCH-11900~M, ~Y

This program creates two continuous month variables, one identifying the month that the respondent received a GED and the other a high school diploma. In future rounds, similar variables will calculate the month an associate's, bachelor's, master's, doctoral, or professional degree was recieved; no respondents had received these degrees in rounds 1 or 2. For more information on the continuous month scheme, see appendix 7 in this document.

```
/* First, create the date variables using round 2 infomation only.*/
HSR2=-4; GEDR2=-4; FLAG=0;

/* The date that the respondent received high school degree.*/
if S11700R2=1 and S11900M2 gt 0 and S11900Y2 gt 0 then HSR2=(S11900Y2-1980)*12+S11900M2;
if -4<S11700R2<0 then HSR2=S11700R2;
if -4<S11900M2<0 then HSR2=S11900M2;
if -4<S11900Y2<0 then HSR2=S11900Y2;
        /* For the two persons who report that they got high school diploma in 1980 and 1988, we change the data
        to 1998 since they finished 12th grade in 1998 and they also reported receiving high school diploma.*/
        if HSR2=6 or HSR2=101 then HSR2=(1998-1980)*12+S11900M2;

/* The date that the respondent received GED.*/
if S13300R2=1 and S13500M2>0 and S13500Y2>0 then GEDR2=(S13500Y2-1980)*12+S13500M2;
if -4<S13300R2<0 then GEDR2=S13300R2;
if -4<S13500M2<0 then GEDR2=S13500M2;
if -4<S13500Y2<0 then GEDR2=S13500Y2;

/******** Second, we will check if there is some conflict between round 1 and round 2.*********/
FLAGHS=0; FLAGHS1=0; FLAGGED=0;
if HSR2 not in (-4,-5) and HSR1 ne -4 and HSR2 ne HSR1 then FLAGHS=1;
if HSR2 not in (-4,-5) and HSR1 ne -4 and HSR2=HSR1 then FLAGHS1=1;
if GEDR2 not in (-4,-5) and GEDR1 ne -4 and GEDR2 ne GEDR1 then FLAGGED=1;

/*********** Finally, we will merge the result from round 1 and round 2 together. **************/
DTGED=-4; DTHS=-4;

if FLAGGED=0 then do;
        if GEDR1 ne -4 then DTGED=GEDR1;
        if GEDR2 ne -4 then DTGED=GEDR2; end;

if FLAGHS=0 then do;
        if HSR1 NE -4 then DTHS=HSR1;
        if HSR2 NE -4 then DTHS=HSR2; end;

/* For the one person whose answer for the date of high diploma from round 2 (209) can not match with that from
round 1 (210), we just set the date according to round 2 (most recent information)*/
if FLAGHS=1 then DTHS=HSR2;

endsas;
```

NUMBER OF GRADES REPEATED OR SKIPPED

Variables Created: CV_GRADES_REPEAT_EVER CV_GRADES_REPEAT_YR
CV_GRADE_SKIPPED_EVER CV_GRADE_SKIPPED_YR

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
DATER1D, M, Y	YINF-900_D, _M, _Y	S9589121–S9589122	YSCH-9589.01.02.01–.02
S3500	YSCH-3500	S9589211–S9589216	YSCH-9589.02.01.01–.06
PUBID	PUBID	S9589221	YSCH-9589.02.02.01
DATER2D, M, Y	YINTDATE~D, ~M, ~Y	S9589311–S9589313	YSCH-9589.03.01.01–.03
S3103	YSCH-3103	S9589321	YSCH-9589.03.02.01
S3104	YSCH-3104	S9589411	YSCH-9589.04.01.01
S8163G11–S8163G13	YSCH-8163BG.01.01–.03	S9589511	YSCH-9589.05.01.01
S8163G21–S8163G22	YSCH-8163BG.02.01–.02	S691111M, S691111Y	YSCH-9691.01.01.01~M, ~Y
S8163G31–S8163G32	YSCH-8163BG.03.01–.02	S691112M, S691112Y	YSCH-9691.01.01.02~M, ~Y
S8163G41–S8163G42	YSCH-8163BG.04.01–.02	S691113M, S691113Y	YSCH-9691.01.01.03~M, ~Y
S8163G51	YSCH-8163BG.05.01	S691114M, S691114Y	YSCH-9691.01.01.04~M, ~Y
S8163G61	YSCH-8163BG.06.01	S691121M, S691121Y	YSCH-9691.01.02.01~M, ~Y
S841611–S841613	YSCH-8416.01.01–.03	S691122M, S691122Y	YSCH-9691.01.02.02~M, ~Y
S841621–S841622	YSCH-8416.02.01–.02	S691211M, S691211Y	YSCH-9691.02.01.01~M, ~Y
S841631–S841632	YSCH-8416.03.01–.02	S691212M, S691212Y	YSCH-9691.02.01.02~M, ~Y
S841641–S841642	YSCH-8416.04.01–.02	S691213M, S691213Y	YSCH-9691.02.01.03~M, ~Y
S841651	YSCH-8416.05.01	S691214M, S691214Y	YSCH-9691.02.01.04~M, ~Y
S841661	YSCH-8416.06.01	S691215M, S691215Y	YSCH-9691.02.01.05~M, ~Y
S877411M, S877411Y	YSCH-8774.01.01~M, ~Y	S691216M, S691216Y	YSCH-9691.02.01.06~M, ~Y
S877412M, S877412Y	YSCH-8774.01.02~M, ~Y	S691221M, S691221Y	YSCH-9691.02.02.01~M, ~Y
S877413M, S877413Y	YSCH-8774.01.03~M, ~Y	S691311M, S691311Y	YSCH-9691.03.01.01~M, ~Y
S877421M, S877421Y	YSCH-8774.02.01~M, ~Y	S691312M, S691312Y	YSCH-9691.03.01.02~M, ~Y
S877422M, S877422Y	YSCH-8774.02.02~M, ~Y	S691313M, S691313Y	YSCH-9691.03.01.03~M, ~Y
S877431M, S877431Y	YSCH-8774.03.01~M, ~Y	S691321M, S691321Y	YSCH-9691.03.02.01~M, ~Y
S877432M, S877432Y	YSCH-8774.03.02~M, ~Y	S691411M, S691411Y	YSCH-9691.04.01.01~M, ~Y
S877441M, S877441Y	YSCH-8774.04.01~M, ~Y	S691511M, S691511Y	YSCH-9691.05.01.01~M, ~Y
S877442M, S877442Y	YSCH-8774.04.02~M, ~Y	S9742111 S9742114	YSCH-9742.01.01.01–.04
S877451M, S877451Y	YSCH-8774.05.01~M, ~Y	S9742121 S9742122	YSCH-9742.01.02.01–.02
S892611–S892613	YSCH-8926.01.01–.03	S9742211 S9742216	YSCH-9742.02.01.01–.06
S892621–S892622	YSCH-8926.02.01–.02	S9742221	YSCH-9742.02.02.01
S892631–S892632	YSCH-8926.03.01–.02	S9742311 S9742313	YSCH-9742.03.01.01–.03
S892641–S892642	YSCH-8926.04.01–.02	S9742321	YSCH-9742.03.02.01
S892651	YSCH-8926.05.01	S9742411	YSCH-9742.04.01.01
S892661	YSCH-8926.06.01	S9742511	YSCH-9742.05.01.01
S897711–S897712	YSCH-8977.01.01–.02	S9793111 S9793114	YSCH-9793.01.01.01–.04
S897721	YSCH-8977.02.01	S9793211 S9793213	YSCH-9793.02.01.01–.03
S897731–S897732	YSCH-8977.03.01–.02	S9793311	YSCH-9793.03.01.01
S897741	YSCH-8977.04.01	S9844111 S9844114	YSCH-9844.01.01.01–.04
S902811–S902813	YSCH-9028.01.01–.03	S9844121 S9844122	YSCH-9844.01.02.01–.02
S902821–S902822	YSCH-9028.02.01–.02	S9844211 S9844216	YSCH-9844.02.01.01–.06
S902831–S902832	YSCH-9028.03.01–.02	S9844221	YSCH-9844.02.02.01
S902841–S902842	YSCH-9028.04.01–.02	S9844311 S9844313	YSCH-9844.03.01.01–.03
S902851	YSCH-9028.05.01	S9844321	YSCH-9844.03.02.01
S902861	YSCH-9028.06.01	S9844411	YSCH-9844.04.01.01
S908011–S908013	YSCH-9080.01.01–.03	S9844511	YSCH-9844.05.01.01
S908021–S908022	YSCH-9080.02.01–.02	S9895111 S9895113	YSCH-9895.01.01.01–.03
S908031–S908032	YSCH-9080.03.01–.02	S9895211 S9895215	YSCH-9895.02.01.01–.05
S908041	YSCH-9080.04.01	S9895311	YSCH-9895.03.01.01
S908051	YSCH-9080.05.01	S9946111 S9946114	YSCH-9946.01.01.01–.04
S908061	YSCH-9080.06.01	S9946121 S9946122	YSCH-9946.01.02.01–.02
S913111–S913113	YSCH-9131.01.01–.03	S9946211 S9946216	YSCH-9946.02.01.01–.06
S913121–S913122	YSCH-9131.02.01–.02	S9946221	YSCH-9946.02.02.01
S913131–S913132	YSCH-9131.03.01–.02	S9946311 S9946313	YSCH-9946.03.01.01–.03
S913141–S913142	YSCH-9131.04.01–.02	S9946321	YSCH-9946.03.02.01

S913151	YSCH-9131.05.01
S913161	YSCH-9131.06.01
S933511M, S933511Y	YSCH-9335.01.01~M, ~Y
S933512M, S933512Y	YSCH-9335.01.02~M, ~Y
S933513M, S933513Y	YSCH-9335.01.03~M, ~Y
S933521M, S933521Y	YSCH-9335.02.01~M, ~Y
S933522M, S933522Y	YSCH-9335.02.02~M, ~Y
S933531M, S933531Y	YSCH-9335.03.01~M, ~Y
S933532M, S933532Y	YSCH-9335.03.02~M, ~Y
S933541M, S933541Y	YSCH-9335.04.01~M, ~Y
S933551M, S933551Y	YSCH-9335.05.01~M, ~Y
S933561M, S933561Y	YSCH-9335.06.01~M, ~Y
S938511–S938513	YSCH-9385.01.01–.03
S938521–S938522	YSCH-9385.02.01–.02
S938531–S938532	YSCH-9385.03.01–.02
S938541–S938542	YSCH-9385.04.01–.02
S938551	YSCH-9385.05.01
S938561	YSCH-9385.06.01
S9589111–S9589114	YSCH-9589.01.01.01–.04
S9946411	YSCH-9946.04.01.01
S9946511	YSCH-9946.05.01.01
S099111M, S099111Y	YSCH-10099.01.01.01~M, ~Y
S099112M, S099112Y	YSCH-10099.01.01.02~M, ~Y
S099113M, S099113Y	YSCH-10099.01.01.03~M, ~Y
S099121M, S099121Y	YSCH-10099.01.02.01~M, ~Y
S099211M, S099211Y	YSCH-10099.02.01.01~M, ~Y
S099212M, S099212Y	YSCH-10099.02.01.02~M, ~Y
S099214M, S099214Y	YSCH-10099.02.01.04~M, ~Y
S099215M, S099215Y	YSCH-10099.02.01.05~M, ~Y
S099311M, S099311Y	YSCH-10099.03.01.01~M, ~Y
S099312M, S099312Y	YSCH-10099.03.01.02~M, ~Y
START1M, START1Y	NEWSCHOOL_START1.01~M, ~Y
START2M, START2Y	NEWSCHOOL_START1.02~M, ~Y
START3M, START3Y	NEWSCHOOL_START1.03~M, ~Y
START4M, START4Y	NEWSCHOOL_START1.04~M, ~Y
START5M, START5Y	NEWSCHOOL_START1.05~M, ~Y
START6M, START6Y	NEWSCHOOL_START1.06~M, ~Y

This program creates variables counting the number of grades the respondent ever repeated or skipped. The first variable in each pair counts the total number of grades ever repeated/skipped through the interview date; the second variable calculates the number through June 30, 1998.

```
/********************************** Define arrays used later on. **********************************/
/* For the arrays that include both school number and the period, put the same period in one array.*/
/* Start date for the first period for each school. */
        array startm start1m start2m start3m start4m start5m start6m;
        array starty start1y start2y start3y start4y start5y start6y;
/* School is elementary, middle or high school. */
        array s8163g s8163g11 s8163g21 s8163g31 s8163g41 s8163g51 s8263g61;
/* first grade for each school and period.*/
        array s84161 s841611 s841621 s841631 s841641 s841651 s841661;
        array s84162 s841612 s841622 s841632 s841642 s841652 s841662;
        array s84163 s841613 s841623 s841633 s841643 s841653 s841663;
/* Start date for the first grade in each school and each period.*/
        array s87741m s877411m s877421m s877431m s877441m s877451m;
        array s87741y s877411y s877421y s877431y s877441y s877451y;
        array s87742m s877412m s877422m s877432m s877442m s877452m;
        array s87742y s877412y s877422y s877432y s877442y s877452y;
        array s87743m s877413m s877423m s877433m s877443m s877453m;
        array s87743y s877413y s877423y s877433y s877443y s877453y;
/* Questions about skip or repeat.*/
        array s89261 s892611 s892621 s892631 s892641 s892651 s892661;
        array s89262 s892612 s892622 s892632 s892642 s892652 s892662;
        array s89263 s892613 s892623 s892633 s892643 s892653 s892663;
        array s89771 s897711 s897721 s897731 s897741 s897751 s897761;
        array s89772 s897712 s897722 s897732 s897742 s897752 s897762;
        array s90281 s902811 s902821 s902831 s902841 s902851 s902861;
        array s90282 s902812 s902822 s902832 s902842 s902852 s902862;
        array s90283 s902813 s902823 s902833 s902843 s902853 s902863;
        array s90801 s908011 s908021 s908031 s908041 s908051 s908061;
        array s90802 s908012 s908022 s908032 s908042 s908052 s908062;
        array s90803 s908013 s908023 s908033 s908043 s908053 s908063;
/*complete the first grade or not?*/
        array s91311 s913111 s913121 s913131 s913141 s913151 s913161;
        array s91312 s913112 s913122 s913132 s913142 s913152 s913162;
        array s91313 s913113 s913123 s913133 s913143 s913153 s913163;
```

```
/* When complete the first grade? */
        array s93351m s933511m s933521m s933531m s933541m s933551m;
        array s93351y s933511y s933521y s933531y s933541y s933551y;
        array s93352m s933512m s933522m s933532m s933542m s933552m;
        array s93352y s933512y s933522y s933532y s933542y s933552y;
        array s93353m s933513m s933523m s933533m s933543m s933553m;
        array s93353y s933513y s933523y s933533y s933543y s933553y;
/* Enrolled in any other grades? */
        array s93851 s938511 s938521 s938531 s938541 s938551 s938561;
        array s93852 s938512 s938522 s938532 s938542 s938552 s938562;
        array s93853 s938513 s938523 s938533 s938543 s938553 s938563;

/* For the loops with school number, period number and the grade number, we use the period grade number for
each array. */
/* Next grades in each school and period. */
        array s958911 s9589111 s9589211 s9589311 s9589411 s9589511;
        array s958912 s9589112 s9589212 s9589312 s9589412 s9589512;
        array s958913 s9589113 s9589213 s9589313 s9589413 s9589513;
        array s958914 s9589114 s9589214 s9589314 s9589414 s9589514;
        array s958915 s9589115 s9589215 s9589315 s9589415 s9589515;
        array s958916 s9589116 s9589216 s9589316 s9589416 s9589516;
        array s958921 s9589121 s9589221 s9589321 s9589421 s9589521;
        array s958922 s9589122 s9589222 s9589322 s9589422 s9589522;
/* Start date for next grade. */
        array s69111m s691111m s691211m s691311m s691411m s691511m;
        array s69112m s691112m s691212m s691312m s691412m s691512m;
        array s69113m s691113m s691213m s691313m s691413m s691513m;
        array s69114m s691114m s691214m s691314m s691414m s691514m;
        array s69115m s691115m s691215m s691315m s691415m s691515m;
        array s69116m s691116m s691216m s691316m s691416m s691516m;
        array s69121m s691121m s691221m s691321m s691421m s691521m;
        array s69122m s691122m s691222m s691322m s691422m s691522m;
        array s69111y s691111y s691211y s691311y s691411y s691511y;
        array s69112y s691112y s691212y s691312y s691412y s691512y;
        array s69113y s691113y s691213y s691313y s691413y s691513y;
        array s69114y s691114y s691214y s691314y s691414y s691514y;
        array s69115y s691115y s691215y s691315y s691415y s691515y;
        array s69116y s691116y s691216y s691316y s691416y s691516y;
        array s69121y s691121y s691221y s691321y s691421y s691521y;
        array s69122y s691122y s691222y s691322y s691422y s691522y;
/* Questions about skip or repeat grades. */
        array s974211 s9742111 s9742211 s9742311 s9742411 s9742511;
        array s974212 s9742112 s9742212 s9742312 s9742412 s9742512;
        array s974213 s9742113 s9742213 s9742313 s9742413 s9742513;
        array s974214 s9742114 s9742214 s9742314 s9742414 s9742514;
        array s974215 s9742115 s9742215 s9742315 s9742415 s9742515;
        array s974216 s9742116 s9742216 s9742316 s9742416 s9742516;
        array s974221 s9742121 s9742221 s9742321 s9742421 s9742521;
        array s974222 s9742122 s9742222 s9742322 s9742422 s9742522;
        array s979311 s9793111 s9793211 s9793311 s9793411 s9793511;
        array s979312 s9793112 s9793212 s9793312 s9793412 s9793512;
        array s979313 s9793113 s9793213 s9793313 s9793413 s9793513;
        array s979314 s9793114 s9793214 s9793314 s9793414 s9793514;
        array s984411 s9844111 s9844211 s9844311 s9844411 s9844511;
        array s984412 s9844112 s9844212 s9844312 s9844412 s9844512;
        array s984413 s9844113 s9844213 s9844313 s9844413 s9844513;
        array s984414 s9844114 s9844214 s9844314 s9844414 s9844514;
```

```
        array s984415 s9844115 s9844215 s9844315 s9844415 s9844515;
        array s984416 s9844116 s9844216 s9844316 s9844416 s9844516;
        array s984421 s9844121 s9844221 s9844321 s9844421 s9844521;
        array s984422 s9844122 s9844222 s9844322 s9844422 s9844522;
        array s989511 s9895111 s9895211 s9895311 s9895411 s9895511;
        array s989512 s9895112 s9895212 s9895312 s9895412 s9895512;
        array s989513 s9895113 s9895213 s9895313 s9895413 s9895513;
        array s989514 s9895114 s9895214 s9895314 s9895414 s9895514;
        array s989515 s9895115 s9895215 s9895315 s9895415 s9895515;
/* complete next grade or not? */
        array s994611 s9946111 s9946211 s9946311 s9946411 s9946511;
        array s994612 s9946112 s9946212 s9946312 s9946412 s9946512;
        array s994613 s9946113 s9946213 s9946313 s9946413 s9946513;
        array s994614 s9946114 s9946214 s9946314 s9946414 s9946514;
        array s994615 s9946115 s9946215 s9946315 s9946415 s9946515;
        array s994616 s9946116 s9946216 s9946316 s9946416 s9946516;
        array s994621 s9946121 s9946221 s9946321 s9946421 s9946521;
        array s994622 s9946122 s9946222 s9946322 s9946422 s9946522;
/* When complete the next grade? */
        array s09911m s099111m s099211m s099311m s099411m s099511m;
        array s09912m s099112m s099212m s099312m s099412m s099512m;
        array s09913m s099113m s099213m s099313m s099413m s099513m;
        array s09914m s099114m s099214m s099314m s099414m s099514m;
        array s09915m s099115m s099215m s099315m s099415m s099515m;
        array s09916m s099116m s099216m s099316m s099416m s099516m;
        array s09921m s099121m s099221m s099321m s099421m s099521m;
        array s09922m s099122m s099222m s099322m s099422m s099522m;
        array s09911y s099111y s099211y s099311y s099411y s099511y;
        array s09912y s099112y s099212y s099312y s099412y s099512y;
        array s09913y s099113y s099213y s099313y s099413y s099513y;
        array s09914y s099114y s099214y s099314y s099414y s099514y;
        array s09915y s099115y s099215y s099315y s099415y s099515y;
        array s09916y s099116y s099216y s099316y s099416y s099516y;
        array s09921y s099121y s099221y s099321y s099421y s099521y;
        array s09922y s099122y s099222y s099322y s099422y s099522y;
/* Array for grades completed and the grades attended in round 2, the starting time and the completing time.*/
        array datestm datestm1-datestm9;        array datesty datesty1-datesty9;
        array datespm datespm1-datespm9;        array datespy datespy1-datespy9;
        array comp comp1-comp9;                 array attend attend1-attend9;

do i=1 to 9;
        datestm[i]=-4; datesty[i]=-4; datespm[i]=-4; datespy[i]=-4; comp[i]=-4; attend[i]=-4;
end;

/************ List the grades completed and the ending date ************/
j=1;
if s913111=1 then do;
   if s841611 not in (-4,-5) then do;
      comp[j]=s841611; datespm[j]=s933511m; datespy[j]=s933511y; j=j+1; end;
   if s841611=-4 and s3104 not in (-4,-5) then do;
      comp[j]=s3104; datespm[j]=s933511m; datespy[j]=s933511y; j=j+1; end;
   if s841611=-4 and s3104=-4 and s3500 not in (-4,-5) then do;
      comp[j]=s3500; datespm[j]=s933511m; datespy[j]=s933511y; j=j+1; end;
end;

if s9946111=1 and s9589111 not in (-4,-5) then do;
   comp[j]=s9589111; datespm[j]=s099111m; datespy[j]=s099111y; j=j+1; end;
```

```
if s9946112=1 and s9589112 not in (-4,-5) then do;
   comp[j]=s9589112; datespm[j]=s099112m; datespy[j]=s099112y; j=j+1; end;
if s9946113=1 and s9589113 not in (-4,-5) then do;
   comp[j]=s9589113; datespm[j]=s099113m; datespy[j]=s099113y; j=j+1; end;
if s9946114=1 and s9589114 not in (-4,-5) then do;
   comp[j]=s9589114; datespm[j]=s099114m; datespy[j]=s099114y; j=j+1; end;
if s913112=1 and s841612 not in (-4,-5) then do;
   comp[j]=s841612; datespm[j]=s933512m; datespy[j]=s933512y; j=j+1; end;
if s9946121=1 and s9589121 not in (-4,-5) then do;
   comp[j]=s9589121; datespm[j]=s099121m; datespy[j]=s099121y; j=j+1; end;
if s9946122=1 and s9589122 not in (-4,-5) then do;
   comp[j]=s9589122; datespm[j]=s099122m; datespy[j]=s099122y; j=j+1; end;
if s913113=1 and s841613 not in (-4,-5) then do;
   comp[j]=s841613; datespm[j]=s933513m; datespy[j]=s933513y; j=j+1; end;
if s913121=1 and s841621 not in (-4,-5) then do;
   comp[j]=s841621; datespm[j]=s933521m; datespy[j]=s933521y; j=j+1; end;
if s9946211=1 and s9589211 not in (-4,-5) then do;
   comp[j]=s9589211; datespm[j]=s099211m; datespy[j]=s099211y; j=j+1; end;
if s9946212=1 and s9589212 not in (-4,-5) then do;
   comp[j]=s9589212; datespm[j]=s099212m; datespy[j]=s099212y; j=j+1; end;
if s9946213=1 and s9589213 not in (-4,-5) then do;
   comp[j]=s9589213; datespm[j]=s099213m; datespy[j]=s099213y; j=j+1; end;
if s9946214=1 and s9589214 not in (-4,-5) then do;
   comp[j]=s9589214; datespm[j]=s099214m; datespy[j]=s099214y; j=j+1; end;
if s9946215=1 and s9589215 not in (-4,-5) then do;
   comp[j]=s9589215; datespm[j]=s099215m; datespy[j]=s099215y; j=j+1; end;
if s9946216=1 and s9589216 not in (-4,-5) then do;
   comp[j]=s9589216; datespm[j]=s099216m; datespy[j]=s099216y; j=j+1; end;
if s913122=1 and s841622 not in (-4,-5) then do;
   comp[j]=s841622; datespm[j]=s933522m; datespy[j]=s933522y; j=j+1; end;
if s9946221=1 and s9589221 not in (-4,-5) then do;
   comp[j]=s9589221; datespm[j]=s099221m; datespy[j]=s099221y; j=j+1; end;
if s913131=1 and s841631 not in (-4,-5) then do;
   comp[j]=s841631; datespm[j]=s933531m; datespy[j]=s933531y; j=j+1; end;
if s9946311=1 and s9589311 not in (-4,-5) then do;
   comp[j]=s9589311; datespm[j]=s099311m; datespy[j]=s099311y; j=j+1; end;
if s9946312=1 and s9589312 not in (-4,-5) then do;
   comp[j]=s9589312; datespm[j]=s099312m; datespy[j]=s099312y; j=j+1;  end;
if s9946313=1 and s9589313 not in (-4,-5) then do;
   comp[j]=s9589313; datespm[j]=s099313m; datespy[j]=s099313y; j=j+1; end;
if s913132=1 and s841632 not in (-4,-5) then do;
   comp[j]=s841632; datespm[j]=s933532m; datespy[j]=s933532y; j=j+1; end;
if s9946321=1 and s9589321 not in (-4,-5) then do;
   comp[j]=s9589321; datespm[j]=s099321m; datespy[j]=s099321y; j=j+1; end;
if s913141=1 and s841641 not in (-4,-5) then do;
   comp[j]=s841641; datespm[j]=s933541m; datespy[j]=s933541y; j=j+1; end;
if s9946411=1 and s9589411 not in (-4,-5) then do;
   comp[j]=s9589411; datespm[j]=s099411m; datespy[j]=s099411y; j=j+1; end;
if s913142=1 and s841642 not in (-4,-5) then do;
   comp[j]=s841642; datespm[j]=s933542m; datespy[j]=s933542y; j=j+1; end;
if s913151=1 and s841651 not in (-4,-5) then do;
   comp[j]=s841651; datespm[j]=s933551m; datespy[j]=s933551y; j=j+1; end;
if s9946511=1 and s9589511 not in (-4,-5) then do;
   comp[j]=s9589511; datespm[j]=s099511m; datespy[j]=s099511y; j=j+1; end;
if s913161=1 and s841661 not in (-4,-5) then do;
   comp[j]=s841661; datespm[j]=s933561m; datespy[j]=s933561y; j=j+1; end;
```

```
/************* List the grades started in round 2 and the starting dates.***************/
k=1;

if s913111 not in (-4,-5) then do;
   if s841611=-4 and s3104 not in (-4,-5) then do;
   attend[k]=s3104;
   if s877411m>0 then datestm[k]=s877411m;
   if s877411y>0 then datesty[k]=s877411y;
   if s877411m=-4 and s877411y=-4 and start1m>0 then datestm[k]=start1m;
   if s877411m=-4 and s877411y=-4 and start1y>0 then datesty[k]=start1y;
   if s877411m=-4 and s877411y=-4 and start1m=-4 and start1y=-4 and dater1m>0 then datestm[k]=dater1m;
   if s877411m=-4 and s877411y=-4 and start1m=-4 and start1y=-4 and dater1y>0 then datesty[k]=dater1y;
   k=k+1;
end;

if s841611=-4 and s3104=-4 and s3500 not in (-4,-5) then do;
   attend[k]=s3500;
   if s877411m>0 then datestm[k]=s877411m;
   if s877411y>0 then datesty[k]=s877411y;
   if s877411m=-4 and s877411y=-4 and start1m>0 then datestm[k]=start1m;
   if s877411m=-4 and s877411y=-4 and start1y>0 then datesty[k]=start1y;
   if s877411m=-4 and s877411y=-4 and start1m=-4 and start1y=-4 and dater1m>0 then datestm[k]=dater1m;
   if s877411m=-4 and s877411y=-4 and start1m=-4 and start1y=-4 and dater1y>0 then datesty[k]=dater1y;
   k=k+1;
end;

if s841611 not in (-4,-5) then do;
   attend[k]=s841611;
   if s877411m>0 then datestm[k]=s877411m;
   if s877411y>0 then datesty[k]=s877411y;
   if s877411m=-4 and s877411y=-4 and start1m>0 then datestm[k]=start1m;
   if s877411m=-4 and s877411y=-4 and start1y>0 then datesty[k]=start1y;
   if s877411m=-4 and s877411y=-4 and start1m=-4 and start1y=-4 and dater1m>0 then datestm[k]=dater1m;
   if s877411m=-4 and s877411y=-4 and start1m=-4 and start1y=-4 and dater1y>0 then datesty[k]=dater1y;
   k=k+1;
end;
end;

if s938511=1 then do; /* check if the respondent went to other grades in that period.*/
   if s9946111 ne -4 then do; attend[k]=s9589111; datestm[k]=s691111m; datesty[k]=s691111y; k=k+1; end;
   if s9946112 ne -4 then do; attend[k]=s9589112; datestm[k]=s691112m; datesty[k]=s691112y; k=k+1; end;
   if s9946113 ne -4 then do; attend[k]=s9589113; datestm[k]=s691113m; datesty[k]=s691113y; k=k+1; end;
   if s9946114 ne -4 then do; attend[k]=s9589114; datestm[k]=s691114m; datesty[k]=s691114y; k=k+1; end;
end;

if s913112 not in (-4,-5) then do; attend[k]=s841612; datestm[k]=s877412m; datesty[k]=s877412y; k=k+1; end;

if s938512=1 then do;/* check if the respondent sent to other grades.*/
   if s9946121 ne -4 then do; attend[k]=s9589121; datestm[k]=s691121m; datesty[k]=s691121y; k=k+1; end;
   if s9946122 ne -4 then do; attend[k]=s9589122; datestm[k]=s691122m; datesty[k]=s691122y; k=k+1; end;
end;

if s913113 not in (-4,-5) then do; attend[k]=s841613; datestm[k]=s877413m; datesty[k]=s877413y; k=k+1; end;

if s913121 not in (-4,-5) then do;
   attend[k]=s841621;
   if s877421m>0 and s877421y>0 then do; datestm[k]=s877421m; datesty[k]=s877421y; end;
```

```
   if s877421m=-4 and s877421y=-4 and start2m>0 and start2y>0 then do;
      datestm[k]=start2m; datesty[k]=start2y; end;
   k=k+1;
end;

if s938521=1 then do; /* check if the respondent went to other grades.*/
   if s9946211 ne -4 then do; attend[k]=s9589211; datestm[k]=s691211m; datesty[k]=s691211y; k=k+1; end;
   if s9946212 ne -4 then do; attend[k]=s9589212; datestm[k]=s691212m; datesty[k]=s691212y; k=k+1; end;
   if s9946213 ne -4 then do; attend[k]=s9589213; datestm[k]=s691213m; datesty[k]=s691213y; k=k+1; end;
   if s9946214 ne -4 then do; attend[k]=s9589214; datestm[k]=s691214m; datesty[k]=s691214y; k=k+1; end;
   if s9946215 ne -4 then do; attend[k]=s9589215; datestm[k]=s691215m; datesty[k]=s691215y; k=k+1; end;
   if s9946216 ne -4 then do; attend[k]=s9589216; datestm[k]=s691216m; datesty[k]=s691216y; k=k+1; end;
end;

if s913122 not in (-4,-5) then do;
   attend[k]=s841622; datestm[k]=s877422m; datesty[k]=s877422y; k=k+1; end;

if s938522=1 then do;/* enrolled in some other grade. */
   if s9946221 ne -4 then do; attend[k]=s9589221; datestm[k]=s691221m; datesty[k]=s691221y; k=k+1; end;
end;

if s913131 not in (-4,-5) then do;
   attend[k]=s841631;
   if s877431m>0 and s877431y>0 then do; datestm[k]=s877431m; datesty[k]=s877431y; end;
   if s877431m=-4 and s877431y=-4 and start3m>0 and start3y>0 then do;
      datestm[k]=start3m; datesty[k]=start3y; end;
   k=k+1;
end;

if s938531=1 then do;/* enrolled in some other grade.*/
   if s9946311 ne -4 then do; attend[k]=s9589311; datestm[k]=s691311m; datesty[k]=s691311y; k=k+1; end;
   if s9946312 ne -4 then do; attend[k]=s9589312; datestm[k]=s691312m; datesty[k]=s691312y; k=k+1; end;
   if s9946313 ne -4 then do; attend[k]=s9589313; datestm[k]=s691313m; datesty[k]=s691313y; k=k+1; end;
end;

if s913132 not in (-4,-5) then do; attend[k]=s841632; datestm[k]=s877432m; datesty[k]=s877432y; k=k+1; end;

if s938532=1 then do; /* enrolled in other grades.*/
   if s9946321 ne -4 then do; attend[k]=s9589321; datestm[k]=s691321m; datesty[k]=s691321y; k=k+1; end;
end;

if s913141 not in (-4,-5) then do;
   attend[k]=s841641;
   if s877441m>0 and s877441y>0 then do; datestm[k]=s877441m; datesty[k]=s877441y; end;
   if s877441m=-4 and s877441y=-4 and start4m>0 and start4y>0 then do;
      datestm[k]=start4m; datesty[k]=start4y; end;
   k=k+1;
end;

if s938541=1 then do;/* enrolled in other grades */
   if s9946411 ne -4 then do; attend[k]=s9589411; datestm[k]=s691411m; datesty[k]=s691411y; k=k+1; end;
end;

if s913142 not in (-4,-5) then do; attend[k]=s841642; datestm[k]=s877442m; datesty[k]=s877442y; k=k+1; end;

if s913151 not in (-4,-5) then do;
   attend[k]=s841651;
```

```
   if s877451m>0 and s877451y>0 then do; datestm[k]=s877451m; datesty[k]=s877451y; end;
   if s877451m=-4 and s877451y=-4 and start5m>0 and start5y>0 then do;
      datestm[k]=start5m; datesty[k]=start5y; end;
   k=k+1;
end;

if s938551=1 then do;/* enrolled in other grades.*/
   if s9946511 ne -4 then do; attend[k]=s9589511; datestm[k]=s691511m; datesty[k]=s691511y; k=k+1; end;
end;

if s913161 not in (-4,-5) then do; attend[k]=s841661; datestm[k]=s877461m; datesty[k]=s877461y; k=k+1; end;

/** After listing all the grades started and all the grades completed that are between 9th to 12th in round 2, we next
can count the grades skipped and the grades repeated.  For the grades skipped,we can look at all the grades
completed, if there is gaps between these grades, we would say that the grades are skipped.  For the grades repeated,
we need to combine the start date and the ending date for each grade to calculate the time that each respondent spent
finishing up each grade. If it took the respondent more than 13 month to finish that grade, we would consider that
grade is repeated.  **/

/********* Count the grades skipped  ********************/
array skip skip1-skip6;
array skipj skipj1-skipj6;
numskip=0;
numskipj=0;
p=1;
q=1;
flag=0;

/*** If the 1st grade completed is more than one greater than the 1st grade attended, count the grades skipped. ***/
if attend[1]>0 and comp[1]>0 and comp[1]>attend[1] then do;    /* find the start date for comp[1]*/
   do i=1 to 9;
      if attend[i]=comp[1] then do; datestmc=datestm[i]; datestyc=datesty[i]; i=9; end;
   end;

   if comp[1]-attend[1]=1 then do;
      skip[p]=attend[1]; p=p+1;
      if datestmc+(datestyc-1990)*12<=102 then do; skipj[q]=attend[1]; q=q+1; end;
   end;

   if comp[1]-attend[1]=2 then do;
      skip[p]=attend[1]; p=p+1; skip[p]=attend[1]+1; p=p+1;
      if datestmc+(datestyc-1990)*12<=102 then do; skipj[q]=attend[1]; q=q+1; skipj[q]=attend[1]+1; q=q+1; end;
   end;

   if comp[1]-attend[1]=3 then do;
      skip[p]=attend[1]; p=p+1; skip[p]=attend[1]+1; p=p+1; skip[p]=attend[1]+2; p=p+1;
      if datestmc+(datestyc-1990)*12<=102 then do;
         skipj[q]=attend[1]; q=q+1; skipj[q]=attend[1]+1; q=q+1; skipj[q]=attend[1]+2; q=q+1; end;
   end;

   if comp[1]-attend[1]=4 then do;
      skip[p]=attend[1]; p=p+1; skip[p]=attend[1]+1; p=p+1; skip[p]=attend[1]+2; p=p+1; skip[p]=attend[1]+3;
                     p=p+1;
       if datestmc+(datestyc-1990)*12<=102 then do;
         skipj[q]=attend[1]; q=q+1; skipj[q]=attend[1]+1; q=q+1; skipj[q]=attend[1]+2; q=q+1; skipj[q]=attend[1]+3;
                     q=q+1; end;
   end;
```

```
   if comp[1]-attend[1]=5 then do;
     skip[p]=attend[1]; p=p+1; skip[p]=attend[1]+1; p=p+1; skip[p]=attend[1]+2; p=p+1; skip[p]=attend[1]+3;
                    p=p+1; skip[p]=attend[1]+4; p=p+1;
      if datestmc+(datestyc-1990)*12<=102 then do;
        skipj[q]=attend[1]; q=q+1; skipj[q]=attend[1]+1; q=q+1; skipj[q]=attend[1]+2; q=q+1; skipj[q]=attend[1]+3;
                    q=q+1; skipj[q]=attend[1]+4; q=q+1; end;
   end;

   if comp[1]-attend[1]>5 then flag=1;
end;


/******************   The grades skipped between the grades completed.   ******************/
do i=1 to 8;

if comp[i+1]>0 and comp[i]>0 and comp[i+1]-comp[i]>1 then do;  /* find the start date for comp[i+1]*/
   do l=1 to 9;
     if attend[l]=comp[i+1] then do; datestmm=datestm[l]; datestym=datesty[l]; l=9; end;
   end;

   if comp[i+1]-comp[i]=2 then do;
     skip[p]=comp[i]+1; p=p+1;
     if datestmm+(datestym-1990)*12<=102 then do; skipj[q]=comp[i]+1; q=q+1; end;
   end;

   if comp[i+1]-comp[i]=3 then do;
     skip[p]=comp[i]+1; p=p+1; skip[p]=comp[i]+2; p=p+1;
     if datestmm+(datestym-1990)*12<=102 then do;
       skipj[q]=comp[i]+1; q=q+1; skipj[q]=comp[i]+2; q=q+1; end;
   end;

   if comp[i+1]-comp[i]=4 then do;
     skip[p]=comp[i]+1; p=p+1; skip[p]=comp[i]+2; p=p+1; skip[p]=comp[i]+3; p=p+1;
     if datestmm+(datestym-1990)*12<=102 then do;
       skipj[q]=comp[i]+1; q=q+1; skipj[q]=comp[i]+2; q=q+1; skipj[q]=comp[i]+3; q=q+1; end;
   end;

   if comp[i+1]-comp[i]=5 then do;
     skip[p]=comp[i]+1; p=p+1; skip[p]=comp[i]+2; p=p+1; skip[p]=comp[i]+3; p=p+1; skip[p]=comp[i]+4; p=p+1;
      if datestmm+(datestym-1990)*12<=102 then do;
       skipj[q]=comp[i]+1; q=q+1; skipj[q]=comp[i]+2; q=q+1; skipj[q]=comp[i]+3; q=q+1; skipj[q]=comp[i]+4;
                    q=q+1; end;
   end;

   if comp[i+1]-comp[i]>5 then flag=2;
end;
end;

/*If the grade attended in the last round is more than one year greater than the grade completed in the last round.**/
/* first find the last attended and last completed grades.*/

if j>1 and k>1 and attend[k-1]>0 and comp[j-1]>0 and attend[k-1]-comp[j-1]>1 then do;
   compl=comp[j-1]; attendl=attend[k-1]; end;

if j=1 and k>1 and attend[k-1]>0 and hgcr1>0 and attend[k-1]-hgcr1>1 then do;
   compl=hgcr1; attendl=attend[k-1]; end;
```

```
if attendl>0 and compl>0 then do;
   if attendl-compl=2 then do;
      skip[p]=compl+1; p=p+1;
      if datestm[k-1]+(datesty[k-1]-1990)*12<=102 then do; skipj[q]=compl+1; q=q+1; end;
   end;

   if attendl-compl=3 then do;
      skip[p]=compl+1; p=p+1; skip[p]=compl+2; p=p+1;
      if datestm[k-1]+(datesty[k-1]-1990)*12<=102 then do;
         skipj[q]=compl+1; q=q+1; skipj[q]=compl+2; q=q+1; end;
   end;

   if attendl-compl=4 then do;
      skip[p]=compl+1; p=p+1; skip[p]=compl+2; p=p+1; skip[p]=compl+3; p=p+1;
      if datestm[k-1]+(datesty[k-1]-1990)*12<=102 then do;
         skipj[q]=compl+1; q=q+1; skipj[q]=compl+2; q=q+1; skipj[q]=compl+3; q=q+1; end;
   end;

   if attendl-compl=5 then do;
      skip[p]=compl+1; p=p+1; skip[p]=compl+2; p=p+1; skip[p]=compl+3; p=p+1; skip[p]=compl+4; p=p+1;
       if datestm[k-1]+(datesty[k-1]-1990)*12<=102 then do;
         skipj[q]=compl+1; q=q+1; skipj[q]=compl+2; q=q+1; skipj[q]=compl+3; q=q+1; skipj[q]=compl+4; q=q+1; end;
   end;

      if datestm[k-1] in (-1,-2,-3) or datesty[k-1] in (-1,-2,-3) then skipj[q]=-3;
   if attendl-compl>5 then flag=3;
end;

do i=1 to 6;
   if skip[i]>0 and skip[i]<=12 then numskip=numskip+1;
   if skipj[i]>0 and skipj[i]<=12 then numskipj=numskipj+1;
end;

if skipj[1] in (-1,-2,-3) or skipj[2] in (-1,-2,-3) or skipj[3] in (-1,-2,-3) or skipj[4] in (-1,-2,-3) or skipj[5] in (-1,-2,-3)
or skipj[6] in (-1,-2,-3) then numskipj=-3;

do i=1 to j-1;
   if comp[i] in (-1,-2,-3) then do; numskip=-3; numskipj=-3; end;
end;

if datestmc in (-1,-2,-3) or datestyc in (-1,-2,-3) or datestmm in (-1,-2,-3) or datestym in (-1,-2,-3) then numskipj=-3;

/************** Count the grades repeated ****************/
array grade grade1-grade6;
array lenth lenth1-lenth6;
array stop stop1-stop6;/* the stop date for each grade*/
l=1;
numrep=0;
numrepj=0;
fflag=0;

do i=1 to 9;
   do t=1 to 9;
      if attend[t]>0 and comp[i]>0 and attend[t]=comp[i] then do; grade[l]=attend[t];
        if datespm[i]>0 and datestm[t]>0 and datespy[i]>0 and datesty[t]>0 then do;
            lenth[l]=datespm[i]-datestm[t]+12*(datespy[i]-datesty[t]); stop[l]=datespm[i]+(datespy[i]-1990)*12; end;
```

```
        if datespm[i] in (-1,-2,-3) or datestm[t] in (-1,-2,-3) or datespy[i] in (-1,-2,-3) or datesty[t] in (-1,-2,-3) then
                    lenth[l]=-3;
        if datespm[i] in (-1,-2,-3) or datespy[i] in (-1,-2,-3) then stop[l]=-3;
        l=l+1;
        t=9;
      end;
    end;
end;

do i=1 to 6;
    if grade[i]>0 and grade[i]<=12 and lenth[i]>13 then do;
        numrep=numrep+1;
        if stop[i]<=102 then numrepj=numrepj+1;
    end;
end;

/**** The last grade attended that are not completed in round2 and take more than 13 month.****/
if j=1 and hgcr1>0 then hgc=hgcr1;
if j>1 and comp[j-1]>0 then hgc=comp[j-1];

    if k>1 and attend[k-1]>hgc and attend[k-1]>0 and attend[k-1]<=12 then do;
      if (dater2y-datesty[k-1])*12+(dater2m-datestm[k-1])>13 then do; numrep=numrep+1; fflag=1; end;
      if (1998-datesty[k-1])*12+(6-datestm[k-1])>13 then do; numrepj=numrepj+1; fflag=2; end;
    end;

if lenth1=-3 or lenth2=-3 or lenth3=-3 or lenth4=-3 or lenth5=-3 or lenth6=-3 then do;
    numrep=-3; numrepj=-3; end;

if stop1=-3 or stop2=-3 or stop3=-3 or stop4=-3 or stop5=-3 or stop6=-3 then numrepj=-3;

/********************************** other cases ************************************/
if s3500 in (95,14) or s3104 in (95,14) or s841611 in (95,14) or s841612 in (95,14) or s841613 in (95,14) or
     s841621 in (95,14) or s841622 in (95,14) or s841631 in (95,14) or s841632 in (95,14) or s841641 in (95,14) or
     s841642 in (95,14) or s841651 in (95,14) or s841661 in (95,14) or s9589111 in (95,14) or
     s9589112 in (95,14) or s9589113 in (95,14) or s9589114 in (95,14) or s9589121 in (95,14) or
     s9589122 in (95,14) or s9589211 in (95,14) or s9589212 in (95,14) or s9589213 in (95,14) or
     s9589214 in (95,14) or s9589215 in (95,14) or s9589216 in (95,14) or s9589221 in (95,14) or
     s9589311 in (95,14) or s9589312 in (95,14) or s9589313 in (95,14) or s9589321 in (95,14) or
     s9589411 in (95,14) or s9589511 in (95,14) then do;
numrep=-3; numskp=-3; numrepj=-3; numskipj=-3;
end;

/* we get length negative because of the data problem.*/
if (lenth1<0 and lenth1 ne .) or (lenth2<0 and lenth2 ne .) or (lenth3<0 and lenth3 ne .) or (lenth4<0 and lenth4 ne
            .) or (lenth5<0 and lenth5 ne .) or (lenth6<0 and lenth6 ne .) then do;
numrep=-3; numrepj=-3; numskip=-3; numskipj=-3;
end;

if s3104=-5 then do;
    numskip=-5; numrep=-5; numskipj=-5; numrepj=-5;
end;

endsas;
```

NUMBER OF SCHOOLS ATTENDED

Variables Created: CV_SCH_ATTEND_EVER
CV_SCH_ATTEND_YR

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
CURGDR1	YSCH-3500	S826361	YSCH-8263.06.01
PUBID	PUBID	S841611–S841613	YSCH-8416.01.01–.03
PINF15	PINF-015_Y	S841621–S841622	YSCH-8416.02.01–.02
NUMJR1	CV_SCH_ATTEND_YR	S841631–S841632	YSCH-8416.03.01–.02
HGAR2	YSCH-2857	S841641–S841642	YSCH-8416.04.01–.02
S66821M–S66825M	YSCH-6682.01~M–.05~M	S841651	YSCH-8416.05.01
S66821Y–S66825Y	YSCH-6682.01~Y–.05~Y	S841661	YSCH-8416.06.01
S8163G11–S8163G13	YSCH-8163BG.01.01–.03	S9589111–S9589114	YSCH-9589.01.01.01–.04
S8163G21–S8163G22	YSCH-8163BG.02.01–.02	S9589121–S9589122	YSCH-9589.01.02.01–.02
S8163G31–S8163G32	YSCH-8163BG.03.01–.02	S9589211–S9589216	YSCH-9589.02.01.01–.06
S8163G41–S8163G42	YSCH-8163BG.04.01–.02	S9589221	YSCH-9589.02.02.01
S8163G51	YSCH-8163BG.05.01	S9589311–S9589313	YSCH-9589.03.01.01–.03
S8163G61	YSCH-8163BG.06.01	S9589321	YSCH-9589.03.02.01
S826311–S826313	YSCH-8263.01.01–.03	S9589411	YSCH-9589.04.01.01
S826321–S826322	YSCH-8263.02.01–.02	S9589511	YSCH-9589.05.01.01
S826331–S826332	YSCH-8263.03.01–.02	START1M–START6M	NEWSCHOOL_START1.01~M–.06~M
S826341–S826342	YSCH-8263.04.01–.02	START1Y–START6Y	NEWSCHOOL_START1.01~Y–.06~Y
S826351	YSCH-8263.05.01	ROUND1–ROUND6	NEWSCHOOL_INTERVIEW.01–.06

This program calculates the number of schools ever attended by the respondent as of the interview date and as of June 30, 1998. The program first counts the number of new regular schools that the youth attended since DLI to the date of survey using the first enrollment date in each new school, then adds this number to the created variable in round 1 to get the total number of regular schools that the youth have ever attended. Users should note that the variable counts only schools at which the respondent attended grades 7–12.

```
array startmx S66821M S66822M S66823M S66824M S66825M;
array startyx S66821Y S66822Y S66823Y S66824Y S66825Y;
array startm start1m start2m start3m start4m start5m start6m;
array starty start1y start2y start3y start4y start5y start6y;
array start start1-start6;
array S8263 S826311 S826321 S826331 S826341 S826351 S826361;
array dumgrd dumgrd1-dumgrd6;

/*First, create the dummy variables for each school to indicate if the youth attended grade 7–12 in that school.*/
do I=1 to 6;
   dumgrd[I]=0; start[I]=-4;
end;

if (7<=S841611<=12 or 7<=S9589111<=12 or 7<=S9589112<=12 or 7<=S9589113<=12 or 7<=S9589114<=12) or
     (7<=S841612<=12 or 7<=S9589121<=12 or 7<=S9589122<=12) or 7<=S841613<=12 then dumgrd1=1;
if (7<=S841621<=12 or 7<=S9589211<=12 or 7<=S9589212<=12 or 7<=S9589213<=12 or 7<=S9589214<=12 or
     7<=S9589215<=12 or 7<=S9589216<=12) or (7<=S841622<=12 or 7<=S9589221<=12) then dumgrd2=1;
if (7<=S841631<=12 or 7<=S9589311<=12 or 7<=S9589312<=12 or 7<=S9589313<=12) or (7<=S841632<=12 or
     7<=S9589321<=12) then dumgrd3=1;
if (7<=S841641<=12 OR 7<=S9589411<=12) or 7<=S841642<=12 then dumgrd4=1;
if (7<=S841651<=12 or 7<=S9589511<=12) then dumgrd5=1;
if 7<=S841661<=12 then dumgrd6=1;

if (-3<=S841611<=-1 or -3<=S9589111<=-1 or -3<=S95891-1<=-1 or -3<=S9589113<=-1 or -3<=S9589114<=-1) or
     (-3<=S8416-1<=-1 or -3<=S9589-11<=-1 or -3<=S9589-12<=-1) or -3<=S841613<=-1 then dumgrd1=-3;
```

```
if (-3<=S841621<=-1 or -3<=S9589211<=-1 or -3<=S95892-1<=-1 or -3<=S9589213<=-1 or -3<=S9589214<=-1 or
     -3<=S9589215<=-1 or -3<=S9589216<=-1) or (-3<=S841622<=-1 or -3<=S9589221<=-1) then dumgrd2=-3;
if (-3<=S841631<=-1 or -3<=S9589311<=-1 or -3<=S95893-1<=-1 or -3<=S9589313<=-1) or (-3<=S841632<=-1 or
       -3<=S9589321<=-1) then dumgrd3=-3;
if (-3<=S841641<=-1 or -3<=S9589411<=-1) or -3<=S841642<=-1 then dumgrd4=-3;
if (-3<=S841651<=-1 or -3<=S9589511<=-1) then dumgrd5=-3;
if -3<=S841661<=-1 then dumgrd6=-3;

/*Next, calculate numjr2 and numr2.*/
/* Calculate the accumulated month of the start date. */
do I=1 to 6;
     if starty[I]>0 and startm[I]>0 then start[I]=12*(starty[I]-1990)+startm[I];
end;

numjr2=0;
numr2=0;
do I=1 to 6;
   if S8263[I]=0 then do;
      if dumgrd[I]=1 then numr2=numr2+1;
      if dumgrd[I]=1 and (0<start[I]<=102 or 0<starty[I]<1998) then numjr2=numjr2+1;
   end;
end;

if dumgrd1=-3 or dumgrd2=-3 or dumgrd3=-3 or dumgrd4=-3 or dumgrd5=-3 or dumgrd6=-3 then do;
   numr2=-3; numjr2=-3; end;

if (S826311=0 and dumgrd1=1 and ((start1y=1998 and -4<start1m<0) or -4<start1y<0)) or (S826321=0 and
dumgrd2=1 and ((start2y=1998 and -4<start2m<0) or -4<start2y<0)) or (S826331=0 and dumgrd3=1 and
((start3y=1998 and -4<start3m<0) or -4<start3y<0)) or (S826341=0 and dumgrd4=1 and ((start4y=1998 and -
4<start4m<0) or -4<start4y<0)) or (S826351=0 and dumgrd5=1 and ((start5y=1998 and -4<start5m<0) or -
4<start5y<0)) or (S826361=0 and dumgrd6=1 and ((start6y=1998 and -4<start6m<0) or -4<start6y<0))
  then numjr2=-3;

/*We then add the above two numbers to the corresponding variables from round 1.*/
if numjr1>=0 then do;
  if numjr2>=0 then numj=numjr2+numjr1;
  if numr2>=0 then nums=numr2+numjr1;
  if numjr2=-3 then numj=-3;
  if numr2=-3 then nums=-3;
end;

/*For the youth who didn't answer the number of schools in round 1, we distinguish them into three cases 1)If the
grade currently attended in round 1 is <=7 and the highest grade attended in round 2 >= 7, we use the number from
round 2 as the total number. 2)If the grade currently attended in round 1 >7, we use -3 for the total number. 3)If
both the grade currently attended in round 1 and the highest grade attended in round 2 are <7, we use -4 for the
total number.*/

if PINF15=-4 then do;
  if curgdr1<=7 then do; numj=numjr2; nums=numr2; end;
  if curgdr1>7 then do; numj=-3; nums=-3; end;
  if curgdr1<7 and hgar2<7 then do; numj=-4; nums=-4; end;
end;

if -4<numjr1<0 then do; numj=numjr1; nums=numjr1; end;

if S826311=-5 then do; numjr2=-5; numr2=-5; numj=-5; nums=-5; end;
endsas;
```

TOTAL FRACTION OF CREDITS EARNED TOWARDS BACHELORS/ASSOCIATE DEGREE

Variables Created: CV_BA_CREDITS.01–.05
CV_ASSOC_CREDITS.01–.05

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
R1DTYPE	YSCH-1800	APCRED1–APCRED4	YSCH-26878.01–.04
R1TCRED	YSCH-16000.01	HSCRED1–HSCRED4	YSCH-26929.01–.04
R1CRED1	YSCH-22800.01	E273371–E273375	YSCH-27337.01–.05
PUBID	PUBID	E273881–E273885	YSCH-27388.01–.05
R1ASCRED	CV_ASSOC_CREDITS	E2070011–E2070015	YSCH-20700.01.01–.05
R1BACRED	CV_BA_CREDITS	E2070021–E2070025	YSCH-20700.02.01–.05
E16200	YSCH-16200.01	E2070031–E2070035	YSCH-20700.03.01–.05
E25807	YSCH-25807.01	E2070041	YSCH-20700.04.01
E25909	YSCH-25909.01	E2070051	YSCH-20700.05.01
E265211–E265215	YSCH-26521.01–.05	E230001–E230003	YSCH-23000.01–.03
XCRED1–XCRED4	YSCH-26827.01–.04		

Codes for Created Variable

These variables are created on a continuous scale. They have 2 implied decimal places.

This program calculates the fraction of credits the respondent has completed toward an associate's degree or a bachelor's degree.

```
/* Note that for all variables below the 1st loop is a school reported in Round1. Loops 2 through 5 are schools
reported in Round2. */
array E26521 (i) E265211-E265215;
array XCRED (i) XCRED1-XCRED4;
array APCRED (i) APCRED1-APCRED4;
array HSCRED (i) HSCRED1-HSCRED4;
array E27337 (i) E273371-E273375;
array E27388 (i) E273881-E273885;
array INCRED (i) INCRED1-INCRED5;
array E207001 (i) E2070011-E2070015;
array E207002 (i) E2070021-E2070025;
array E207003 (i) E2070031-E2070035;
array E207004 (i) E2070041-E2070045;
array E207005 (i) E2070051-E2070055;
array cred (i) cred1-cred5;
array tcred (i) tcred1-tcred5;
array tascred (i) tascred1-tascred5;
array tbacred (i) tbacred1-tbacred5;
array asfrac (i) asfrac1-asfrac5;          /* Created variable for associate credits */
array bafrac (i) bafrac1-bafrac5;          /* Created variable for bachelor credits */

/* Round1 created variables are in fraction form times 100. R1CRED1 is the total earned credits variable. R1DTYPE
is type of degree respondent is working toward. */

if E273371=1 and R1CRED1=>0 then do;  R1ASCRED=R1CRED1; end;
if R1DTYPE=4 and E273371=-4 and R1CRED1=>0 then do; R1ASCRED=R1CRED1; end;
if E273371=3 and R1CRED1=>0 then do; R1BACRED=R1CRED1; end;
if R1DTYPE=5 and E273371=-4 and R1CRED1=>0 then do; R1BACRED=R1CRED1; end;

/* Hand edit case */  if pubid=4163 then R1ASCRED=-4;
```

/* If a respondent reported missing information for incoming credits and then give a valid answer in E16200, then we will include that value to the Round1 credit count. If E16200 was not answered, then incoming credits for school 1 is simply the Round1 created variable. */

```
if R1BACRED=>0 and E16200=>0 then do; INCRED1=R1BACRED+E16200; end;
if R1ASCRED=>0 and E16200=>0 then do; INCRED1=R1ASCRED+E16200; end;
if R1ASCRED=>0 then do; INCRED1=R1ASCRED; end;
if R1BACRED=>0 then do; INCRED1=R1BACRED; end;

do i=1 to 5;
  if E26521=>0 then do; INCRED=E26521; end;
end;

/* 1st school */
if E2070011=>0 then do; cred1=E2070011; end;
if E2070011=>0 and E2070012=>0 then do; cred1=E2070011+E2070012; end;
if E2070011=>0 and E2070012=>0 and E2070013=>0 then do; cred1=E2070011+E2070012+E2070013; end;
if E2070011=>0 and E2070012=>0 and E2070013=>0 and E2070014=>0 then do;
     cred1=E2070011+E2070012+E2070013+E2070014; end;
if E2070011=>0 and E2070012=>0 and E2070013=>0 and E2070014=>0 and E2070015=>0 then do;
     cred1=E2070011+E2070012+E2070013+E2070014+E2070015; end;

/* 2nd school */
if E2070021=>0 then do; cred2=E2070021; end;
if E2070021=>0 and E2070022=>0 then do; cred2=E2070021+E2070022; end;
if E2070021=>0 and E2070022=>0 and E2070023=>0 then do; cred2=E2070021+E2070022+E2070023; end;
if E2070022=>0 and E2070022=>0 and E2070023=>0 and E2070024=>0 then do;
     cred2=E2070021+E2070022+E2070023+E2070024; end;
if E2070021=>0 and E2070022=>0 and E2070023=>0 and E2070024=>0 and E2070025=>0 then do;
     cred2=E2070021+E2070022+E2070023+E2070024+E2070025; end;

/* 3rd school */
if E2070031=>0 then do; cred3=E2070031; end;
if E2070031=>0 and E2070032=>0 then do; cred3=E2070031+E2070032; end;
if E2070031=>0 and E2070032=>0 and E2070033=>0 then do; cred3=E2070031+E2070032+E2070033; end;
if E2070031=>0 and E2070032=>0 and E2070033=>0 and E2070034=>0 then do;
     cred3=E2070031+E2070032+E2070033+E2070034; end;
if E2070031=>0 and E2070032=>0 and E2070033=>0 and E2070034=>0 and E2070035=>0 then do;
     cred3=E2070031+E2070032+E2070033+E2070034+E2070035; end;

/* 4th school */
if E2070041=>0 then do; cred4=E2070041; end;

/* 5th school */
if E2070051=>0 then do; cred5=E2070051; end;

/* Sum incoming credits with earned credits */
do over INCRED;
  if INCRED=>0 then do; TCRED=INCRED; end;
  if CRED=>0 then do; TCRED=CRED; end;
  if INCRED=>0 and CRED=>0 then do; TCRED=INCRED+CRED; end;
end;

/* E23000(1-3) accounts for incorrect tabulation of Round2 credits earned. */
array E23000 (i) E230001-E230003;
array OTCRED (i) OTCRED1-OTCRED3;
```

```
do i=1 to 3;
   OTCRED=.;
   if E23000>-4 then do; OTCRED=TCRED; TCRED=E23000; end;
end;

/* Here we add the Round1 created variable for fraction of credits with the Round2 count. They are separated by the
repsondent's answer to E27337(1-5), which is the type of degree question. */
/* Initialixe both created variables. */
do i=1 to 5;
   asfrac=0;
   bafrac=0;
end;

/* Associates Degree */
do i=1 to 5;
   if E27337=1 and TCRED=>0 then do; TASCRED=TCRED; end;
   if -4<E27388<0 and E27337=1 then do; asfrac=E27388; end;
   if TASCRED=>0 and E27388>0 then do; asfrac=(TASCRED/E27388)*100; end;
   asfrac=round(asfrac,1);
end;

/* Bachelors Degree */
do i=1 to 5;
   if E27337=3 and TCRED=>0 then do; TBACRED=TCRED; end;
   if -4<E27388<0 and E27337=3 then do; bafrac=E27388; end;
   if TBACRED=>0 and E27388>0 then do; bafrac=(TBACRED/E27388)*100; end;
   bafrac=round(bafrac,1);
end;

/* Non-interview cases */
if E16200=-5 then do;
   do i=1 to 5;
      bafrac=-5; asfrac=-5; end;
end;

endsas;
```

YOUTH'S MATH PIAT SCORE

Variables Created: CV_PIAT_STANDARD_SCORE
CV_PIAT_PERCENTILE_SCORE

Variables Used

Name in Program	Question Name on CD
CURD, CURM, CURY	SYMBOL!CURDATE~D, ~M, ~Y
DOB01D, M, Y	KEY!BDATE_D, _M, _Y (round 1)
PIATRAW	PIAT_RAW_SCORE

This program calculates the respondent's age utilizing the interview date information and the respondent's date of birth information. It then uses the age data and the raw PIAT score to create a standard PIAT score and percentile rank for each respondent.

```
/**calculate the interview date (continuous months format) <initialize everyone to valid skip>**/

intday=-4;          intmonth=-4;      intyear=-4;

/**if there is a valid day, month, and year interview date in round 2 use it.**/
if (curm>=0) and (curd>=0) then do;
        intday=curd;
        intmonth=curm;
        intyear=cury;
end;

contmint=((intyear-1980)*12)+intmonth;

/****************respondent age section*****************<if there is a valid date of birth for respondent, use it.>**/

dobday=dob01d;
dobmonth=dob01m;
dobyear=dob01y;

/**<ensure all respondents have an in range value.>**/
if (dobday<0) then dobday=15;
if (dobmonth<0) then dobmonth=6;
if (dobyear<1970) then dobyear=1983;

/**<calculate the youth's age in continuous months>**/
yagemnth=((intyear-dobyear)*12)+(intmonth-dobmonth);
if (intday-dobday<0) then yagemnth=yagemnth-1;

/**********************piat standard score section************<initialize everyone to valid skip>**/

piatstnd=-4;

/* At this point the program loops through each three-month age range and assigns appropriate standard scores based
on the respondent's raw score. This code is quite lengthy and is not reproduced here; instead, we include a table that
provides the standard score and percentile rank associated with each raw score in each age range. Users who need the
precise programming code should contact NLS User Services. */
```

Standard Score	Percentile Rank	Raw Score by Age Range (Years-Months)											
		13-0 to 13-2	13-3 to 13-5	13-6 to 13-8	13-9 to 13-11	14-0 to 14-2	14-3 to 14-5	14-6 to 14-8	14-9 to 14-11	15-0 to 15-2	15-3 to 15-5	15-6 to 15-8	15-9 to 15-11
55	0	Ω30	Ω31	Ω32	Ω33	Ω34	Ω35	Ω36	Ω37	Ω38	Ω39	Ω40	Ω41
56	0	31	32	33	34	35	36	37	38	39	40	41	42
57	0	32	33	34	35	36	37	38	39	40	41	42	
58	0	33	34	35	36	37	38		40	41	42	43	43
59	0	34	35	36	37			39			43		44
60	0	35	36			38	39	40	41	42		44	45
61	0	36	37	37	38	39	40	41	42	43	44	45	46
62	1	37		38	39	40	41	42	43	44	45	46	47
63	1		38	39	40	41	42	43	44	45	46	47	48
64	1	38	39	40	41	42	43	44	45	46	47	48	49
65	1	39	40	41	42	43	44	45	46	47	48	49	50
66	1	40	41	42	43	44	45	46	47	48	49	50	51
67	1	41	42	43	44	45	46	47	48	49	50	51	52
68	2	42	43	44	45	46	47	48	49	50	51	52	53
69	2	43	44	45	46	47	48	49	50	51	52	53	54
70	2	44	45	46	47	48	49	50	51	52	53	54	55
71	3	45	46	47	48	49	50	51	52	53	54	55	56
72	3	46	47	48	49	50	51	52	53	54	55	56	57
73	4	47	48	49	50	51	52	53	54	55	56	57	
74	4	48	49	50	51	52	53	54	55	56	57	58	58
75	5	49	50	51	52	53	54	55	56	57	58		59
76	5	50	51	52	53	54	55	56	57	58	59	59	60
77	6		52		54		56					60	61
78	7	51	53	53	55	55	57	57	58	59	60	61	62
79	8	52		54		56		58	59	60	61	62	63
80	9	53	54	55	56	57	58	59	60	61	62	63	64
81	10	54	55	56	57	58	59	60	61	62	63	64	65
82	12	55	56	57	58	59	60	61	62	63	64	65	66
83	13	56	57	58	59	60	61	62	63	64	65	66	
84	14	57	58	59	60	61	62	63	64	65	66	67	67
85	16	58	59	60	61	62	63	64	65	66	67		68
86	18	59	60	61	62	63	64	65	66	67		68	69
87	19						65				68	69	70
88	21	60	61	62	63	64		66	67	68	69	70	71
89	23	61	62	63	64	65	66	67	68	69	70	71	72
90	25	62	63	64	65	66	67	68	69	70	71	72	
91	27			65	66	67	68	69	70				73
92	30	63	64				69			71	72	73	74
93	32	64	65	66	67	68		70	71	72	73	74	75
94	34	65	66	67	68	69	70	71	72	73	74	75	
95	37			68	69	70	71	72	73	74	75		76
96	39	66	67									76	77
97	42	67	68	69	70	71	72	73	74	75	76	77	
98	45		69	70	71	72	73	74	75	76	77		78
99	47	68										78	79
100	50	69	70	71	72	73	74	75	76	77	78	79	80

Standard Score	Percentile Rank	Raw Score by Age Range (Years-Months)											
		13-0 to 13-2	13-3 to 13-5	13-6 to 13-8	13-9 to 13-11	14-0 to 14-2	14-3 to 14-5	14-6 to 14-8	14-9 to 14-11	15-0 to 15-2	15-3 to 15-5	15-6 to 15-8	15-9 to 15-11
101	53		71	72	73	74	75	76	77	78	79	80	
102	55	70	72	73	74	75	76	77	78	79	80		81
103	58	71			75			78				81	82
104	61	72	73	74		76	77		79	80	81	82	83
105	63	73	74	75	76	77	78	79	80	81	82	83	84
106	66	74	75	76	77	78	79	80	81	82	83	84	
107	68		76	77	78	79	80	81	82	83	84		85
108	70	75	77	78	79	80	81	82	83	84	85	85	86
109	73	76				81	82	83				86	87
110	75	77	78	79	80	82	83	84	84	85	86	87	88
111	77	78	79	80	81				85	86	87	88	89
112	79	79	80	81	82	83	84	85	86	87	88	89	90
113	81	80	81	82	83	84	85	86	87	88	89	90	91
114	82	81	82	83	84	85	86	87	88	89	90	91	92
115	84	82	83	84	85	86	87	88	89	90	91	92	93
116	86	83	84	85	86	87	88	89	90	91	92	93	94
117	87	84	85	86	87	88	89	90	91	92	93	94	
118	88		86	87	88	89	90	91	92	93	94	95	95
119	90	85	87	88	89	90	91	92	93	94	95	96	96
120	91	86	88	89	90	91	92	93	94	95	96		97
121	92	87		90	91	92	93	94	95	96		97	98
122	93	88	89	91	92	93	94	95	96	97	97	98	99
123	94	89	90								98	99	
124	95	90	91	92	93	94	95	96	97	98	99	100	100
125	95		92	93	94	95	96	97	98	99	100		
126	96	91	93	94	95	96	97	98	99	100			
127	96	92		95	96	97	98	99	100				
128	97	93	94	96	97	98	99	100					
129	97	94	95										
130	98	95	96	97	98	99	100						
131	98		97	98	99	100							
132	98	96	98	99	100								
133	99	97		100									
134	99	98	99										
135	99	99	100										
136	99	100											

Note: This table of *PIAT* scores is based on the information provided by the testing company. Users interested in more information about the *PIAT-Math Assessment* may wish to consult the following reference:

Markwardt, Frederick C. Jr. *Peabody Individual Achievement Test-Revised*. Circle Pines, Minn.: American Guidance Service, Inc., 1989.

QED DATA: SCHOOL SIZE AND STUDENT-TEACHER RATIO

Variables Created: CV_SCHOOL_SIZE
CV_STUDENT_TEACHER_RATIO

Variables Used

Name in Program	Question Name on CD
GENDER	KEY!SEX
RACE	KEY!RACE
E2857	YSCH-2857
E21625	YSCH-21625
AGE	SYMBOL!KEY!AGE
REGION	CV_CENSUS_REGION
CV_HGC	CV_HGC_EVER

Codes for Created Variable

School Size	Student-teacher ratio
1 = <100	1 = <14
2 = 100-299	2 = 14-17
3 = 300-499	3 = 18-21
4 = 500-749	4 = 22+
5 = 750-999	
6 = 1000+	

This program merges school identification information from the NLSY97 data with data from the "National Education Database," provided under copyright by Quality Education Data (QED), Inc. It then creates two variables that provide information about the respondent's school.

```
/* Before creating the QED variables, survey staff used dates of enrollment data to identify the respondent's current
or most recent school and then merged that information with the QED data. This information is excluded here due
to length and confidentiality restrictions. Researchers needing more information about this process should contact
NLS User Services. */

/* Initialize variables */
SCHSIZE=-4;
STUDTEAC=-4;

/* create SCHSIZE and STUDTEAC while checking
that the grade reported in the QED coincides with that
being reported in the NLSY97 */

if GRADSPAN='2' then GRADSPAN='3';
if GRADSPAN='A' then GRADSPAN='2';
if GRADSPAN='B' or GRADSPAN='C' or
    GRADSPAN='D' or GRADSPAN='E' or
    GRADSPAN='*' then GRADSPAN='0';

/* Redefine each GRADSPAN to GRADSPN to
eliminate character value problems */
if GRADSPAN='0' then GRADSPN=0;
if GRADSPAN='1' then GRADSPN=1;
if GRADSPAN='2' then GRADSPN=2;
if GRADSPAN='3' then GRADSPN=3;
if GRADSPAN='4' then GRADSPN=4;
if GRADSPAN='5' then GRADSPN=5;
if GRADSPAN='6' then GRADSPN=6;
if GRADSPAN='7' then GRADSPN=7;
if GRADSPAN='8' then GRADSPN=8;
if GRADSPAN='9' then GRADSPN=9;

/* Initialize the dummies to zero. */
dummy1=0;      dummy4=0;
dummy5=0;      dummy6=0;
dummy7=0;      dummy8=0;
dummy9=0;      dummy10=0;

if E2857 ge 1 and E2857 le 3 then do;
  dummy1=1;    dummy4=1;
  dummy5=1;    dummy6=0;
  dummy7=0;    dummy8=0;
  dummy9=0;    dummy10=0;
end;
if E2857 ge 4 and E2857 le 6 then do;
  dummy1=1;    dummy4=1;
  dummy5=1;    dummy6=1;
  dummy7=0;    dummy8=0;
  dummy9=0;    dummy10=0;
end;
if E2857 ge 7 and E2857 le 8 then do;
```

```
  dummy1=1;     dummy4=0;
  dummy5=1;     dummy6=1;
  dummy7=1;     dummy8=1;
  dummy9=0;     dummy10=0;
end;
if E2857 eq 9 then do;
  dummy1=1;     dummy4=0;
  dummy5=0;     dummy6=0;
  dummy7=1;     dummy8=1;
  dummy9=1;     dummy10=0;
end;
if E2857 ge 10 and E2857 le 12 then do;
  dummy1=1;     dummy4=0;
  dummy5=0;     dummy6=0;
  dummy7=0;     dummy8=1;
  dummy9=1;     dummy10=1;
end;

correct=0;
if GRADSPN=1 and dummy1=1 then correct=1;
if GRADSPN=4 and dummy4=1 then correct=1;
if GRADSPN=5 and dummy5=1 then correct=1;
if GRADSPN=6 and dummy6=1 then correct=1;
if GRADSPN=7 and dummy7=1 then correct=1;
if GRADSPN=8 and dummy8=1 then correct=1;
if GRADSPN=9 and dummy9=1 then correct=1;
if GRADSPN=2 and dummy10=1 then correct=1;

/* If the grades reported in the QED and the NLSY97
do not coincide, we make both created variables equal
to -3; otherwise, we proceed to create them according
to the definition provided at the top of the program */

/* Need to check parameters for STUDRANG */
if STUDRANG=0 or STUDRANG=1 then
    SCHSIZE=1;
if STUDRANG=2 then SCHSIZE=2;
if STUDRANG=3 then SCHSIZE=3;
if STUDRANG=4 then SCHSIZE=4;
if STUDRANG=5 then SCHSIZE=5;
if STUDRANG=6 or STUDRANG=7 or
    STUDRANG=8 or STUDRANG=9 then
    SCHSIZE=6;
if STUDRANG=. then SCHSIZE=-3;
if STUDENT1 gt 0 and TEACHERS gt 0 then
    STUDTEA1=STUDENT1/TEACHERS;
if STUDTEA1 lt 14 then STUDTEAC=1;
if STUDTEA1 ge 14 and STUDTEA1 lt 18 then
    STUDTEAC=2;
if STUDTEA1 ge 18 and STUDTEA1 lt 22 then
    STUDTEAC=3;
if STUDTEA1 ge 22 then STUDTEAC=4;
if STUDTEA1 eq . then STUDTEAC=-3;

if correct=0 then SCHSIZE=-3;
if correct=0 then STUDTEAC=-3;

/* We also make sure that there are at least six schools
in each cell determined by the region, type of school,
control, school size and student-teacher ratio to avoid
any possibilities of identifying the schools. We first run
the tabulations, then assign -3 to the created variables
for schools that fall in cells of five or less schools. */

INDEX=REGION||TYPESCH||E21625||SCHSIZE||
    STUDTEAC;
proc freq;
 tables REGION*TYPESCH*E21625*SCHSIZE*
    STUDTEAC / OUT=A;
 proc freq data=A noprint;
 tables COUNT;
data A;
set A;

INDEX=REGION||TYPESCH||E21625||SCHSIZE||
    STUDTEAC;
proc freq;
 tables INDEX / OUT=INDEXFRQ;
 proc freq data=INDEXFRQ noprint;
 tables COUNT;

if COUNT le 5 then SCHSIZE=-3;
if COUNT le 5 then STUDTEAC=-3;
if pin=-5 then do;
  studteac=-5; schsize=-5;
end;

endsas;
```