# Programs for Recipiency Event Histories

**(survey year 2020)**

- Respondent Unemployment Benefits Receipt
- Spouse/Partner Unemployment Benefits Receipt
- AFDC/TANF Receipt
- Food Stamps/SNAP Receipt
- Respondent/Dependent Child/Spouse/Partner SSI/SSDI Receipt
- Summary Welfare (AFDC/TANF, Food Stamps/SNAP, SSI/SSDI) Receipt

## Respondent Unemployment Benefits Receipt

```
%let cEndYr = 121;              *(2021-1900);
%let cEndMo = 1464;     *(121+1)*12;
%let cYrs = 44;                         *(2021-1978)+1;
%let cMos = 528;                *44*12;
%let cFillFlag = 2020;

data Q13;
 set prg2020.in_inc_2020;

rename
 'Q13RECIP-2_2020'n          =          RECIP2
 'Q13U-43B_2020'n           =    q43B
 'Q13U-43C~M_2020'n          =    q43CM
 'Q13U-43C~Y_2020'n          =    q43CY
 'Q13U-44A1_CHK1.01_2020'n    =    q44A1CHK101
 'Q13U-44A1_CHK1.02_2020'n    =    q44A1CHK102
 'Q13U-44A1_CHK1.03_2020'n    =    q44A1CHK103
 'Q13U-44A1_CHK1.04_2020'n    =    q44A1CHK104
 'Q13U-44A1_CHK1.05_2020'n    =    q44A1CHK105
 'Q13U-44A3.01_2020'n        =    q44A301
 'Q13U-44A3.02_2020'n        =    q44A302
 'Q13U-44A3.03_2020'n        =    q44A303
 'Q13U-44A3.04_2020'n        =    q44A304
 'Q13U-44A3.05_2020'n        =    q44A305
 'Q13U-44A5.01_2020'n        =    q44A501
 'Q13U-44A6.01_2020'n        =    q44A601
 'Q13U-44B.01_2020'n         =    q44B01
 'Q13U-44B.02_2020'n         =    q44B02
 'Q13U-44B.03_2020'n         =    q44B03
```

'Q13U-44B.04_2020'n = q44B04
'Q13U-44B.05_2020'n = q44B05
'Q13U-44C.02~M_2020'n = q44C02M
'Q13U-44C.02~Y_2020'n = q44C02Y
'Q13U-44C.03~M_2020'n = q44C03M
'Q13U-44C.03~Y_2020'n = q44C03Y
'Q13U-44C.04~M_2020'n = q44C04M
'Q13U-44C.04~Y_2020'n = q44C04Y
'Q13U-44C.05~M_2020'n = q44C05M
'Q13U-44C.05~Y_2020'n = q44C05Y
'Q13U-44D.01_2020'n = q44D01
'Q13U-44D.02_2020'n = q44D02
'Q13U-44D.03_2020'n = q44D03
'Q13U-44D.04_2020'n = q44D04
'Q13U-44D.05_2020'n = q44D05
'Q13U-45A.01_2020'n = q45A01
'Q13U-45A.02_2020'n = q45A02
'Q13U-45A.03_2020'n = q45A03
'Q13U-45A.04_2020'n = q45A04
'Q13U-45A.05_2020'n = q45A05
'Q13U-45B.01~M_2020'n = q45B01M
'Q13U-45B.01~Y_2020'n = q45B01Y
'Q13U-45B.02~M_2020'n = q45B02M
'Q13U-45B.02~Y_2020'n = q45B02Y
'Q13U-45B.03~M_2020'n = q45B03M
'Q13U-45B.03~Y_2020'n = q45B03Y
'Q13U-45B.04~M_2020'n = q45B04M
'Q13U-45B.04~Y_2020'n = q45B04Y
'Q13U-45B.05~M_2020'n = q45B05M
'Q13U-45B.05~Y_2020'n = q45B05Y
'Q13U-47E.01_2020'n = q47E01
'Q13U-47E.02_2020'n = q47E02
'Q13U-47E.03_2020'n = q47E03
'Q13U-47E.04_2020'n = q47E04
'Q13U-47E.05_2020'n = q47E05
'Q13U-62.01_2020'n = q6201
'Q13U-62.02_2020'n = q6202
'Q13U-62.03_2020'n = q6203
'Q13U-62.04_2020'n = q6204
'Q13U-62.05_2020'n = q6205
'Q13U-62A.01_2020'n = q62A01
'Q13U-62A.02_2020'n = q62A02
'Q13U-62A.03_2020'n = q62A03
'Q13U-62A.04_2020'n = q62A04
'Q13U-62A.05_2020'n = q62A05
'Q13U-63.01_2020'n = q6301

```
 'Q13U-63.02_2020'n             =    q6302
 'Q13U-63.03_2020'n          =    q6303
 'Q13U-63A.01_2020'n         =    q63A01
 'Q13U-63AA.01_2020'n        =    q63AA01
 'Q13U-63AB.01_2020'n        =    q63AB01
 'Q13U-63AB.02_2020'n        =    q63AB02
 'Q13U-63B.01_2020'n         =    q63B01
 'Q13U-63B.02_2020'n         =    q63B02
 'Q13U-64.01_2020'n          =    q6401
 'Q13U-64.02_2020'n          =    q6402
 'Q13U-64.03_2020'n          =    q6403
 'Q13U-64.04_2020'n          =    q6404
 'Q13U-64.05_2020'n          =    q6405
 'CURDATE~D_2020'n           =    intDay
 'CURDATE~M_2020'n           =    intMo
 'CURDATE~Y_2020'n           =    intYr
;

data recip1;
 merge R29.lintdate2020 Q13 prg2020.oldin2020 (keep=norcid uydolo:);
 by norcid;
 if (intMo > 0);
recip = RECIP2;

proc freq;
 tables recip;
 title "freq for all";
run;
proc freq;
 tables lintyr lint;
 where recip ne 0;
 title "DLI year and month after DLI month for those whose UC EH need to be updated in R29";
run;

data recip;
 set recip1;
* convert Y2K changes to simple years;
array yrs (*) intYr q43cy
            q44c01y q44c02y q44c03y q44c04y q44c05y
            q45b01y q45b02y q45b03y q45b04y q45b05y;

do i=1 to dim(yrs);
   if yrs(i) > 0 then yrs(i)=yrs(i)-1900;
end;

* date of this interview;
```

```
intDt=intYr*12+intMo;
/* ================================================================
  Merge info from several sources + convert weeks to months.
  ================================================================ */
dummy1 = -4;
dummy2 = -4;
dummy3 = -4;

if q44c01y>-4 and q43cy>-4 and q44c01y<q43cy then do;
          q44c01y=q43cy;
          q44c01m=q43cm;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y>q43cy then do;
          q44c01y=q44c01y;
          q44c01m=q44c01m;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y=q43cy then do;
          q44c01m = max(q44c01m, q43cm);
          q44c01y = max(q44c01y, q43cy);
end;
else if q44c01y<=-4 and q43cy>-4 then do;
          q44c01m = max(q44c01m, q43cm);
          q44c01y = max(q44c01y, q43cy);
end;

/* nrOfRMo - # of months x was received (used when start/stop dates are not known).*/
array nrOfRMos  (*) q63a01 q63ab02 dummy1 dummy2 dummy3;
nrOfRMos(1) = max(nrOfRMos(1), q63aa01, q63ab01);
do s = 1 to dim(nrOfRMos);
   if nrOfRMos(s) > 0 then
      nrOfRMos(s) = max(round(nrOfRMos(s)/4.3),1);
end;

* How much did R get during each month (by spells);
array howMuch (*) q47e01-q47e05;
array q63b    (*) q63b01-q63b05;
do s=1 to dim(howMuch);
   if howMuch(s) = -4 then howMuch(s) = q63b(s);
   if howMuch(s) > 0 then howMuch(s)=round(howMuch(s)*4.3,1);
end;
/* ================================================================
  Recipiency status (rec) and amounts (dol) by months
  All months from 1978jan till dec of current year, indexed from 1900jan.
  78*12+1 = 937
  ================================================================ */
```

```
array rec (937:&cEndMo);      * Did they receive x in that month?;
array dol (937:&cEndMo);      * How much?;

/* ================================================================
 * No/unknown recipiency
 * ================================================================ */
if recip=0 then do;
   do m=lint to intDt;
      rec(m)=-4;
      dol(m)=-4;
   end;
end;
else if recip in (-1,-2) then do;
   do m=lint to intDt;
      rec(m) = Recip;
      dol(m) = Recip;
   end;
end;
else if recip in (-3) then do;
   do m=lint to intDt;
      rec(m) = Recip;
      dol(m) = Recip;
   end;
end;

/* ================================================================
 * There was some recipiency, get details
 * ================================================================ */
else do;
check = 0;
array startMo (*) q44c01m q44c02m q44c03m q44c04m q44c05m;
array startYr (*) q44c01y q44c02y q44c03y q44c04y q44c05y;
array endMo (*) q45b01m q45b02m q45b03m q45b04m q45b05m;
array endYr (*) q45b01y q45b02y q45b03y q45b04y q45b05y;
array cont    (*) q44d01-q44d05;
array q44b    (*) q44b01-q44b05;
array q64     (*) q6401-q6405;

/* ----------------------------------------------------------------------
 * Calculate starts and ends of spells if possible
 * ----------------------------------------------------------------------*/

array starts (*) starts1-starts5;
array ends   (*) ends1-ends5;

* Continuous recipiency "ends" at interview date;
```

```
do s = 1 to dim(cont);
   if cont(s) = 1 then do;
      endYr(s) = intYr;
      endMo(s) = intMo;
   end;
end;

* Compute starts/stops/last for basic cases (dates are known);
do s = 1 to dim(starts);
   if startMo(s) > 0 & startYr(s) > 0 then
      starts(s) = startYr(s)*12+startMo(s);
   if endMo(s) > 0 & endYr(s) > 0 then
      ends(s) = endYr(s)*12+endMo(s);
end;

seamFlag=0;
do s = 1 to dim(starts);
         if starts(s) > 0 then do;
      if starts(s)=lint   then seamFlag=1;
      if starts(s)=lint-1 then seamFlag=2;
                  if 0<starts(s)<lint-1 then seamFlag=3;
         end;
end;
lint_real=lint-936-1;
do s=1 to dim(starts);
  if starts(s) > 0 & starts(s)<lint then starts(s)=lint;
end;

* Try to deduce starts/ends from # of mos;
do s = 1 to dim(starts);
   if nrOfRMos(s) > 0 then do;
      if ends(s) > 0 then do;
         starts(s) = ends(s) - nrOfRMos(s)+1;
                        check=5.1;
         if (starts(s) < lint) then check = 5;
      end;
      else if starts(s) > 0 then do;
         ends(s) = starts(s) + nrOfRMos(s) - 1;
                        check=100+s+0.1;
                        if ends(s)>intDt then check=100+s;
      end;
                  else if starts(s) = .  & ends(s) = .  then do;
         if s = 1 then do; starts(s) = lint; check=201+0.1; end;
                        else do; starts(s) = ends(s-1) + 1; check=200+s+0.1; end;
         ends(s) = starts(s) + nrOfRMos(s) - 1;
                        if ends(s)>intDt then check=200+s;
```

```
      end;
   end;
end;

last = 0;
if q44b01=2 or q43B = 2 then last = 1;
do s = 1 to dim(starts);
   if ((startMo(s) > 0 | startYr(s) > 0 | endMo(s) > 0 | endYr(s) > 0 |nrOfRMos (s) > 0) & last < s)
then last = s;
end;

/* ---------------------------------------------------------------------------
 * Edges of recipiency spells and periods between: n1 r1 n2 r2 ...  (NNNRRR???NRRR)
 * Under ideal conditions they touch, but often there is an unclear zone between them (-2)
 * --------------------------------------------------------------------------- */
array rs(*) rs1-rs5;       * Recipiency starts;
array re(*) re1-re5;       * Recipiency ends;
array ns(*) ns1-ns6;       * Nonrecipiency starts (# of spells + 1, potentialy bef and after);
array ne(*) ne1-ne6;       * Nonrecipiency ends;

* The fully known cases;
do s = 1 to last;
   if starts(s) > 0 then do;
      ne(s) = starts(s)-1;
      rs(s) = starts(s);
   end;
   if ends(s) > 0 then do;
      re(s)   = ends(s);
      ns(s+1) = ends(s)+1;
   end;
end;

* The unknown and partially known cases;
do s = 1 to last;
   if starts(s) = . & startYr(s) > 0 then do;
                  check=301;
      xne = startYr(s)*12;
      if ( s = 1 & lint < =xne ) |
        ( s > 1 & 0 < ends(s-1) < xne) |
        ( s > 1 & 0 < endYr(s-1) + 1 < startYr(s) )
         then do; ne(s) = xne; check=302; end;
      if startYr(s) < endYr(s) then do;
         rs(s) = startYr(s)*12+12;
                        check=303;
                  end;
   end;
```

```
   if (ends(s) = . & endYr(s) > 0) then do;
                    check=401;
      if 0 < startYr(s) < endYr(s) then do;
         re(s) = endYr(s)*12+1;
         ns(s+1) = endYr(s)*12+13;
                              check=402;
                    end;
                    if 0 < startYr(s) = endYr(s) then do;
        ns(s+1)=endYr(s)*12+13;
                     if rs(s)>0 then do;
                              re(s)=rs(s);
                     end;
                     check=403;
                    end;
   end;
end;

if (ne(1) > 0) then ns(1) = lint;              * There was a nonrecipiency at the begining;
if (ns(last+1) > 0) then ne(last+1) = intDt;    * There was a nonrecipiency at the end;

if last>=2 then do s = 2 to last;
          if 0 < rs(s) & rs(s) = re(s-1) then check = 7;
end;
 if rs1 > 0 & rs1<lint then rs1=lint;

/* ---------------------------------------------------------------------------
 * Fill the recipiency status (rec) and amounts (dol).
 * --------------------------------------------------------------------------- */
* First, assume we know nothing;
do m = lint to intDt;
   rec(m) = -2;
   dol(m) = -2;
end;

* Fill in info we know (any of the periods below can be empty);
do s = 1 to last;
   * nonrecipiency period;
   if (0 < ns(s) <= ne(s)) then
      do m = max(ns(s),lint) to min(ne(s),intDt);
         rec(m) = -4;
         dol(m) = -4;
      end;
   * recipiency period;
   if (0 < rs(s) <= re(s)) then
      do m = max(rs(s),lint) to min(re(s),intDt);
```

```
            rec(m) = 1;
            dol(m) = howMuch(s);
                                if howmuch(s) in (. -4) then do; dol(m)=-2; check5=s+0.1; end;
         end;
      * we know only start - we are sure R received smth that month;
      else if (lint <= rs(s) <= intDt & re(s) = .) then do;
         rec(rs(s)) = 1;
         dol(rs(s)) = howMuch(s);
         if howmuch(s) in (. -4) then do; dol(rs(s))=-2; check5=s+0.2; end;
      end;
      * we know only end - we are sure R received smth that month (can be even intDt if
continuous);
      else if (rs(s) = . & lint <= re(s) <= intDt) then do;
         rec(re(s)) = 1;
         dol(re(s)) = howMuch(s);
                  if howmuch(s) in (. -4) then do; dol(re(s))=-2; check5=s+0.3; end;
      end;
end;

* The last period of nonrecipiency;
if (0 < ns(last+1) <= ne(last+1)) then
   do m = ns(last+1) to min(ne(last+1),intDt);
      rec(m) = -4;
      dol(m) = -4;
   end;

end;
/* ================================================================= */
 * End if recip = .... (getting details for people who got something)
/* ================================================================= */

* fill in space at the end of year after the interview;
do m=intDt+1 to hbound(rec);
   rec(m)=-4;
   dol(m)=-4;
end;

endDt=intDt-936;
noInt=0;
if q44c01m in (-1,-2) & q44c01y in (-1,-2) then q43b = 2;

if noInt=0 then
   do i=lbound(rec) to hbound(rec);
      if rec(i)=. then rec(i)=-4;
      if dol(i)=. then dol(i)=-4;
   end;
```

```
if noInt=1 then
   do i=lbound(rec) to hbound(rec);
      if rec(i)=. then rec(i)=-5;
      if dol(i)=. then dol(i)=-5;
   end;
/* ================================================================
 * Calculating yearly variables.
 * ================================================================ */

array recYM(&cYrs,12) rec1-rec&cMos;
array dolYM(&cYrs,12) dol1-dol&cMos;
array yRec(*)   yRec78-yRec&cEndYr;
array yDol(*)   yDol78-yDol&cEndYr;
array yRecMo(*) yRecMo78-yRecMo&cEndYr;

* Calculate yRec (any recipiency in a yr?);
do y=1 to dim(yRec);
   do m=1 to 12;
      if recYM(y,m) ~= . then    yRec(y)=0;
      if recYM(y,m) > 0 then do; yRec(y)=1; leave; end;
   end;
        if yRec(y)=0 then do m=1 to 12;
           if recYM(y,m) in (-1 -2 -3) then do;  yRec(y)=recYM(y,m); leave; end;
        end;
        if yRec(y)=0 then yRec(y)=-4;
end;

* Calculate yDol ($ in yr), yRecMo (# of months of recipiency in each y);
do y=1 to dim(yRec);
   if yRec(y) > 0 then do;
                  rrefuse = 0; drefuse = 0;
                  rdk = 0;     ddk = 0;
      do m=1 to 12;
                        if dolYM(y,m) = -1 then drefuse = 1;
                        else if dolYM(y,m) in (-2 -3) then ddk = 1;
        if recYM(y,m) = -1 then rrefuse = 1;
                        else if recYM(y,m) = -2 then rdk = 1;
                  end;
                  if rrefuse = 1 then yRecMo(y) = -1;
                  else if rdk = 1 then yRecMo(y) = -2;
                  else do;
               yRecMo(y) = 0;
               do m=1 to 12;
                  if recYM(y,m) = 1 then yRecMo(y)=yRecMo(y)+1;
               end;
            end;
```

```
            if rrefuse = 1 or drefuse = 1 then yDol(y) = -3;
                  else if rdk = 1 or ddk = 1 then yDol(y) = -3;
                  else do;
               yDol(y) = 0;
               do m=1 to 12;
                  if 0 <= dolYM(y,m) then yDol(y)=yDol(y)+dolYM(y,m);
               end;
            end;
         end;
         else do;
                  yDol(y)=yRec(y);
                  yRecMo(y)=yRec(y);
         end;
end;


/* ===================================================================
 *  Flag yearly amount that are not within possible bounds.
  * ================================================================== */

array yFlg(*)   yFlg78-yFlg&cEndYr;
array cMinDol (&cYrs) _temporary_
   (36 40 43 48 50 49 51 54 56 58 61 65 68 70 70 72 74 76 76 88 88 98 98 21*109);
array cMaxDol (&cYrs) _temporary_
   (158 173 187 208 217 215 222 236 245 254 266 284 298 305 305 315 322 333 333 383 383
433 433 7*483 3*525 550 575 9*653);
array cMaxDol1 (&cYrs) _temporary_
   (158 173 187 208 217 215 222 236 245 254 266 284 298 305 305 315 322 333 333 383 383
433 433 7*483 3*525 550 575 7*653 1253 953);
do y=1 to dim(yRec);
   yFlg(y) = 0;
   if yRec(y) = -5 then yFlg(y)=-5;
   if yDol(y) < 0 then continue;
         avgDol = yDol(y) / yRecMo(y);
   if avgDol < cMinDol(y)*4.3 then
                  yFlg(y)=1;
         else if cMaxDol(y)*4.3 < avgDol then do;
      yFlg(y)=2;
                  if cMinDol(y)*4.3 <= avgDol / yRecMo(y) <= cMaxDol(y)*4.3 then yFlg(y)=3;
                  if cMaxDol1(y)*4.3 < avgDol then do;
                    yFlg(y)=2.1;
       if cMinDol(y)*4.3 <= avgDol / yRecMo(y) <= cMaxDol1(y)*4.3 then yFlg(y)=3.1;
                  end;
   end;
end;
```

```
/*
=================================================================
 *
 * ================================================================ */
 do y=1 to dim(yRec);
    if yDol(y) = .  then yDol(y)=-4;
    if yRec(y) = -4 then yRec(y)=0;
end;

yflag=0;
array ydolo(*)   uyDolo78-uyDolo&cEndYr;
do i=1 to dim(ydol);
 if lintyr=i+77 & 0<lintmo<13 then do;
  if ydol(i)=-4 then do; ydol(i)=ydolo(i); yflag=1; end;
  else if ydol(i)>=0 & ydolo(i) not in (-4 0) then do;
    if ydolo(i)>0 then do; ydol(i)=ydol(i)+ydolo(i); yflag=2; end;
          else if ydolo(i) in (-1 -2 -3) then do; ydol(i)=ydolo(i); yflag=3; end;
  end;
 end;
end;
```

## Spouse/Partner Unemployment Benefits Receipt

```
%let cEndYr = 121;              *(2021-1900);
%let cEndMo = 1464;     *(121+1)*12;
%let cYrs = 44;                         *(2021-1978)+1;
%let cMos = 528;                *44*12;
%let cFillFlag = 2020;

data Q13;
 set prg2020.in_inc_2020;

rename
 'Q13RECIP-2-SP_2020'n              =     recip2
 'Q13S-43B_2020'n                 =     q43B
 'Q13S-43C~M_2020'n               =     q43CM
 'Q13S-43C~Y_2020'n               =     q43CY
 'Q13S-44A1_CHK1.01_2020'n        =     q44A1CHK101
 'Q13S-44A1_CHK1.02_2020'n        =     q44A1CHK102
 'Q13S-44A1_CHK1.03_2020'n        =     q44A1CHK103
 'Q13S-44A1_CHK1.04_2020'n        =    q44A1CHK104
 'Q13S-44A1_CHK1.05_2020'n        =    q44A1CHK105
 'Q13S-44A3.01_2020'n             =     q44A301
 'Q13S-44A3.02_2020'n             =     q44A302
 'Q13S-44A3.03_2020'n             =     q44A303
```

'Q13S-44A3.04_2020'n = q44A304
'Q13S-44A3.05_2020'n = q44A305
'Q13S-44A5.01_2020'n = q44A501
'Q13S-44A6.01_2020'n = q44A601
'Q13S-44B.01_2020'n = q44B01
'Q13S-44B.02_2020'n = q44B02
'Q13S-44B.03_2020'n = q44B03
'Q13S-44B.04_2020'n = q44B04
'Q13S-44B.05_2020'n = q44B05
'Q13S-44C.02~M_2020'n = q44C02M
'Q13S-44C.02~Y_2020'n = q44C02Y
'Q13S-44C.03~M_2020'n = q44C03M
'Q13S-44C.03~Y_2020'n = q44C03Y
'Q13S-44C.04~M_2020'n = q44C04M
'Q13S-44C.04~Y_2020'n = q44C04Y
'Q13S-44C.05~M_2020'n = q44C05M
'Q13S-44C.05~Y_2020'n = q44C05Y
'Q13S-44D.01_2020'n = q44D01
'Q13S-44D.02_2020'n = q44D02
'Q13S-44D.03_2020'n = q44D03
'Q13S-44D.04_2020'n = q44D04
'Q13S-44D.05_2020'n = q44D05
'Q13S-45A.01_2020'n = q45A01
'Q13S-45A.02_2020'n = q45A02
'Q13S-45A.03_2020'n = q45A03
'Q13S-45A.04_2020'n = q45A04
'Q13S-45A.05_2020'n = q45A05
'Q13S-45B.01~M_2020'n = q45B01M
'Q13S-45B.01~Y_2020'n = q45B01Y
'Q13S-45B.02~M_2020'n = q45B02M
'Q13S-45B.02~Y_2020'n = q45B02Y
'Q13S-45B.03~M_2020'n = q45B03M
'Q13S-45B.03~Y_2020'n = q45B03Y
'Q13S-45B.04~M_2020'n = q45B04M
'Q13S-45B.04~Y_2020'n = q45B04Y
'Q13S-45B.05~M_2020'n = q45B05M
'Q13S-45B.05~Y_2020'n = q45B05Y
'Q13S-47E.01_2020'n = q47E01
'Q13S-47E.02_2020'n = q47E02
'Q13S-47E.03_2020'n = q47E03
'Q13S-47E.04_2020'n = q47E04
'Q13S-47E.05_2020'n = q47E05
'Q13S-62.01_2020'n = q6201
'Q13S-62.02_2020'n = q6202
'Q13S-62.03_2020'n = q6203
'Q13S-62.04_2020'n = q6204

```
 'Q13S-62.05_2020'n                =    q6205
 'Q13S-62A.01_2020'n                =    q62A01
 'Q13S-62A.02_2020'n                =    q62A02
 'Q13S-62A.03_2020'n                =          q62A03
 'Q13S-62A.04_2020'n                =    q62A04
 'Q13S-62A.05_2020'n                =          q62A05
 'Q13S-63.01_2020'n                =    q6301
 'Q13S-63A.01_2020'n                =    q63A01
 'Q13S-63AA.01_2020'n               =    q63AA01
 'Q13S-63AB.01_2020'n               =    q63AB01
 'Q13S-63B.01_2020'n                =    q63B01
 'Q13S-64.01_2020'n                =    q6401
 'Q13S-64.02_2020'n                =    q6402
 'Q13S-64.03_2020'n                =    q6403
 'Q13S-64.04_2020'n                =    q6404
 'Q13S-64.05_2020'n                =          q6405
 'CURDATE~D_2020'n                 =    intDay
 'CURDATE~M_2020'n                 =    intMo
 'CURDATE~Y_2020'n                 =    intYr
 'Q13-10_2020'n                                          =        q10
;

data recip1;
 merge R29.lintdate2020 Q13 prg2020.oldin2020  (keep=norcid pydolo:);
 by norcid;
 if (intMo > 0);

recip = RECIP2;

proc freq;
 tables recip;
 title "freq for all";
run;
proc freq;
 tables lintyr lint;
 where recip ne 0 & q10 ne 0;
 title "DLI year and month after DLI month for those whose SPUC EH need to be updated in
R29";
run;

data recip;
 set recip1;

* convert Y2K changes to simple years;
array yrs (*) intYr q43cy
            q44c01y q44c02y q44c03y q44c04y q44c05y
```

```
            q45b01y q45b02y q45b03y q45b04y q45b05y;

do i=1 to dim(yrs);
   if yrs(i) > 0 then yrs(i)=yrs(i)-1900;
end;

* date of this interview;
intDt=intYr*12+intMo;

/* ===================================================================
  Merge info from several sources + convert weeks to months.
  =================================================================== */
dummy = -4;

if q44c01y>-4 and q43cy>-4 and q44c01y<q43cy then do;
          q44c01y=q43cy;
          q44c01m=q43cm;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y>q43cy then do;
          q44c01y=q44c01y;
          q44c01m=q44c01m;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y=q43cy then do;
          q44c01m = max(q44c01m, q43cm);
          q44c01y = max(q44c01y, q43cy);
end;
else if q44c01y<=-4 and q43cy>-4 then do;
          q44c01m = max(q44c01m, q43cm);
          q44c01y = max(q44c01y, q43cy);
end;

/* nrOfRMo - # of months x was received (used when start/stop dates are not known).*/
array nrOfRMos  (*) q63aa01 dummy dummy dummy dummy;
nrOfRMos(1) = max(nrOfRMos(1), q63a01, q63ab01);
do s = 1 to dim(nrOfRMos);
   if nrOfRMos(s) > 0 then
      nrOfRMos(s) = max(round(nrOfRMos(s)/4.3),1);
end;

* How much did R get during each month (by spells);
array howMuch (*) q47e01-q47e05;
array q63b    (*) q63b01-q63b05;
array q63     (*) q6301-q6305;

do s=1 to dim(howMuch);
   if howMuch(s) = -4 then howMuch(s) = q63b(s);
```

```
   if howMuch(s) > 0 then howMuch(s)=round(howMuch(s)*4.3,1);
end;

/* ===================================================================
   Recipiency status (rec) and amounts (dol) by months
   All months from 1978jan till dec of current year, indexed from 1900jan.
   78*12+1 = 937
   =================================================================== */

array rec (937:&cEndMo);      * Did they receive x in that month?;
array dol (937:&cEndMo);      * How much?;

/* ===================================================================
 * No/unknown recipiency
 * ================================================================= */
if recip=0 or q10=0 then do;
   do m=lint to intDt;
      rec(m)=-4;
      dol(m)=-4;
   end;
end;
else if recip in (-1,-2) then do;
   do m=lint to intDt;
      rec(m) = recip;
      dol(m) = recip;
   end;
end;
else if recip=-3 then
   do m=lint to intDt;
      rec(m) = -3;
      dol(m) = -3;
   end;

/* ===================================================================
 * There was some recipiency, get details
 * ================================================================= */
else do;
check = 0;                  * the case needs a manual check;
array startMo (*) q44c01m q44c02m q44c03m q44c04m q44c05m;
array startYr (*) q44c01y q44c02y q44c03y q44c04y q44c05y;
array endMo (*) q45b01m q45b02m q45b03m q45b04m q45b05m;
array endYr (*) q45b01y q45b02y q45b03y q45b04y q45b05y;
array cont    (*) q44d01-q44d05;
array q44b    (*) q44b01-q44b05;
array q64     (*) q6401-q6405;
```

```
/* ---------------------------------------------------------------------------
 * Calculate starts and ends of spells if possible
 * ---------------------------------------------------------------------------*/

array starts (*) starts1-starts5;
array ends   (*) ends1-ends5;

* Continuous recipiency "ends" at interview date;
do s = 1 to dim(cont);
   if cont(s) = 1 then do;
      endYr(s) = intYr;
      endMo(s) = intMo;
   end;
end;

* Compute starts/stops/last for basic cases (dates are known);
do s = 1 to dim(starts);
   if startMo(s) > 0 & startYr(s) > 0 then
      starts(s) = startYr(s)*12+startMo(s);
   if endMo(s) > 0 & endYr(s) > 0 then
      ends(s) = endYr(s)*12+endMo(s);
end;

seamFlag=0;
do s = 1 to dim(starts);
          if starts(s) > 0 then do;
       if starts(s)=lint   then seamFlag=1;
       if starts(s)=lint-1 then seamFlag=2;
                   if 0<starts(s)<lint-1 then seamFlag=3;
          end;
end;

lint_real=lint-936-1;
do s=1 to dim(starts);
 if starts(s) > 0 & starts(s)<lint then starts(s)=lint;
end;

* Try to deduce starts/ends from # of mos;
do s = 1 to dim(starts);
   if nrOfRMos(s) > 0 then do;
      if ends(s) > 0 then do;
          starts(s) = ends(s) - nrOfRMos(s)+1;
                              check=5.1;
          if (starts(s) < lint) then check = 5;
       end;
       else if starts(s) > 0 then do;
```

```
         ends(s) = starts(s) + nrOfRMos(s) - 1;
                              check=100+s+0.1;
                              if ends(s)>intDt then check=100+s;
      end;
                  else if starts(s) = .  & ends(s) = .  then do;
         if s = 1 then do; starts(s) = lint; check=201+0.1; end;
                              else do; starts(s) = ends(s-1) + 1; check=200+s+0.1; end;
         ends(s) = starts(s) + nrOfRMos(s) - 1;
                              if ends(s)>intDt then check=200+s;
      end;
   end;
end;

last = 0;
if q44b01=2 or q43B = 2 then last = 1;
do s = 1 to dim(starts);
   if ((startMo(s) > 0 | startYr(s) > 0 | endMo(s) > 0 | endYr(s) > 0 |nrOfRMos (s) > 0) & last < s)
then last = s;
end;

/* ---------------------------------------------------------------------------
 * Edges of recipiency spells and periods between: n1 r1 n2 r2 ...  (NNNRRR???NRRR)
 * Under ideal conditions they touch, but often there is an unclear zone between them (-2)
 * --------------------------------------------------------------------------- */
array rs(*) rs1-rs5;       * Recipiency starts;
array re(*) re1-re5;       * Recipiency ends;
array ns(*) ns1-ns6;       * Nonrecipiency starts (# of spells + 1, potentialy bef and after);
array ne(*) ne1-ne6;       * Nonrecipiency ends;

* The fully known cases;
do s = 1 to last;
   if starts(s) > 0 then do;
      ne(s) = starts(s)-1;
      rs(s) = starts(s);
   end;

   if ends(s) > 0 then do;
      re(s)   = ends(s);
      ns(s+1) = ends(s)+1;
   end;
end;

* The unknown and partially known cases;
do s = 1 to last;
   if starts(s) = . & startYr(s) > 0 then do;
                  check=301;
```

```
      xne = startYr(s)*12;
      if ( s = 1 & lint < =xne ) |
        ( s > 1 & 0 < ends(s-1) < xne) |
        ( s > 1 & 0 < endYr(s-1) + 1 < startYr(s) )
         then do; ne(s) = xne; check=302; end;
      if startYr(s) < endYr(s) then do;
         rs(s) = startYr(s)*12+12;
                        check=303;
                end;
   end;
   if (ends(s) = . & endYr(s) > 0) then do;
                check=401;
      if 0 < startYr(s) < endYr(s) then do;
         re(s) = endYr(s)*12+1;
         ns(s+1) = endYr(s)*12+13;
                        check=402;
                end;
                if 0 < startYr(s) = endYr(s) then do;
       ns(s+1)=endYr(s)*12+13;
                  if rs(s)>0 then do;
                        re(s)=rs(s);
                  end;
                  check=403;
                end;
   end;
end;

if (ne(1) > 0) then ns(1) = lint;              * There was a nonrecipiency at the begining;
if (ns(last+1) > 0) then ne(last+1) = intDt;    * There was a nonrecipiency at the end;

if last>=2 then do s = 2 to last;
         if 0 < rs(s) & rs(s) = re(s-1) then check = 7;
end;
  if rs1 > 0 & rs1<lint then rs1=lint;

/* ---------------------------------------------------------------------------
 * Fill the recipiency status (rec) and amounts (dol).
 * --------------------------------------------------------------------------- */
* First, assume we know nothing;
do m = lint to intDt;
   rec(m) = -2;
   dol(m) = -2;
end;

* Fill in info we know (any of the periods below can be empty);
do s = 1 to last;
```

```
   * nonrecipiency period;
   if (0 < ns(s) <= ne(s)) then
      do m = max(ns(s),lint) to min(ne(s),intDt);
         rec(m) = -4;
         dol(m) = -4;
      end;
   * recipiency period;
   if (0 < rs(s) <= re(s)) then
      do m = max(rs(s),lint) to min(re(s),intDt);
         rec(m) = 1;
         dol(m) = howMuch(s);
                        if howmuch(s) in (. -4) then do; dol(m)=-2; check5=s+0.1; end;

      end;
   * we know only start - we are sure R received smth that month;
   else if (lint <= rs(s) <= intDt & re(s) = .) then do;
      rec(rs(s)) = 1;
      dol(rs(s)) = howMuch(s);
      if howmuch(s) in (. -4) then do;  dol(rs(s))=-2; check5=s+0.2; end;
   end;
   * we know only end - we are sure R received smth that month (can be even intDt if
continuous);
   else if (rs(s) = . & lint <= re(s) <= intDt) then do;
      rec(re(s)) = 1;
      dol(re(s)) = howMuch(s);
                if howmuch(s) in (. -4) then do; dol(re(s))=-2; check5=s+0.3; end;
   end;
end;

* The last period of nonrecipiency;
if (0 < ns(last+1) <= ne(last+1)) then
   do m = ns(last+1) to min(ne(last+1),intDt);
      rec(m) = -4;
      dol(m) = -4;
   end;

end;
/* ============================================================== */
/* End if recip = .... (getting details for people who got something)
/* ============================================================== */

* Fill in space at the end of year after the interview;
do m=intDt+1 to hbound(rec);
   rec(m)=-4;
   dol(m)=-4;
end;
```

```
endDt=intDt-936;
noInt=0;
if q44c01m in (-1,-2) & q44c01y in (-1,-2) then q43b = 2;

if noInt=0 then
   do i=lbound(rec) to hbound(rec);
      if rec(i)=. then rec(i)=-4;
      if dol(i)=. then dol(i)=-4;
   end;
if noInt=1 then
   do i=lbound(rec) to hbound(rec);
      if rec(i)=. then rec(i)=-5;
      if dol(i)=. then dol(i)=-5;
   end;
/* ================================================================
 * Calculating yearly variables.
 * ================================================================ */
array recYM(&cYrs,12) rec1-rec&cMos;
array dolYM(&cYrs,12) dol1-dol&cMos;
array yRec(*)   yRec78-yRec&cEndYr;
array yDol(*)   yDol78-yDol&cEndYr;
array yRecMo(*) yRecMo78-yRecMo&cEndYr;

* Calculate yRec (any recipiency in a yr?);
do y=1 to dim(yRec);
   do m=1 to 12;
      if recYM(y,m) ~= . then    yRec(y)=0;
      if recYM(y,m) > 0 then do; yRec(y)=1; leave; end;
   end;
        if yRec(y)=0 then do m=1 to 12;
           if recYM(y,m) in (-1 -2 -3) then do;  yRec(y)=recYM(y,m); leave; end;
        end;
        if yRec(y)=0 then yRec(y)=-4;
end;

* Calculate yDol ($ in yr), yRecMo (# of months of recipiency in each y);
do y=1 to dim(yRec);
   if yRec(y) > 0 then do;
                  rrefuse = 0; drefuse = 0;
                  rdk = 0;     ddk = 0;
      do m=1 to 12;
                        if dolYM(y,m) = -1 then drefuse = 1;
                        else if dolYM(y,m) = -2 then ddk = 1;
        if recYM(y,m) = -1 then rrefuse = 1;
                        else if recYM(y,m) = -2 then rdk = 1;
```

```
                end;
                if rrefuse = 1 then yRecMo(y) = -1;
                else if rdk = 1 then yRecMo(y) = -2;
                else do;
              yRecMo(y) = 0;
              do m=1 to 12;
                if recYM(y,m) = 1 then yRecMo(y)=yRecMo(y)+1;
              end;
            end;
            if rrefuse = 1 or drefuse = 1 then yDol(y) = -1;
                else if rdk = 1 or ddk = 1 then yDol(y) = -2;
                else do;
              yDol(y) = 0;
              do m=1 to 12;
                if 0 <= dolYM(y,m) then yDol(y)=yDol(y)+dolYM(y,m);
              end;
            end;
          end;
          else do;
                yDol(y)=yRec(y);
                yRecMo(y)=yRec(y);
          end;
end;

/* ===================================================================
 *  Flag yearly amount that are not within possible bounds.
 * =================================================================== */

array yFlg(*)   yFlg78-yFlg&cEndYr;
array cMinDol (&cYrs) _temporary_
   (36 40 43 48 50 49 51 54 56 58 61 65 68 70 70 72 74 27*76);
array cMaxDol (&cYrs) _temporary_
   (158 173 187 208 217 215 222 236 245 254 266 284 298 305 305 315 322 333 333 383 383
425 425 7*471 3*525 550 575 9*653);
array cMaxDol1 (&cYrs) _temporary_
   (158 173 187 208 217 215 222 236 245 254 266 284 298 305 305 315 322 333 333 383 383
433 433 7*483 3*525 550 575 7*653 1253 953);

do y=1 to dim(yRec);
   yFlg(y) = 0;
   if yRec(y) = -5 then yFlg(y)=-5;
   if yDol(y) < 0 then continue;
         avgDol = yDol(y) / yRecMo(y);
   if avgDol < cMinDol(y)*4.3 then
                yFlg(y)=1;
         else if cMaxDol(y)*4.3 < avgDol then do;
```

```
      yFlg(y)=2;
                if cMinDol(y)*4.3 <= avgDol / yRecMo(y) <= cMaxDol(y)*4.3 then yFlg(y)=3;
                if cMaxDol1(y)*4.3 < avgDol then do;
                  yFlg(y)=2.1;
       if cMinDol(y)*4.3 <= avgDol / yRecMo(y) <= cMaxDol1(y)*4.3 then yFlg(y)=3.1;
                end;
   end;
end;

/* ===================================================================
 *
 * =================================================================== */

do y=1 to dim(yRec);
   if yDol(y) = .  then yDol(y)=-4;
   if yRec(y) = -4 then yRec(y)=0;
end;

yflag=0;
array ydolo(*)   pyDolo78-pyDolo&cEndYr;
do i=1 to dim(ydol);
 if lintyr=i+77 & 0<lintmo<13 then do;
  if ydol(i)=-4 then do; ydol(i)=ydolo(i); yflag=1; end;
  else if ydol(i)>=0 & ydolo(i) not in (-4 0) then do;
   if ydolo(i)>0 then do; ydol(i)=ydol(i)+ydolo(i); yflag=2; end;
         else if ydolo(i) in (-1 -2 -3) then do; ydol(i)=ydolo(i); yflag=3; end;
  end;
 end;
end;
```

## Aid to Families with Dependent Children (AFDC)/Temporary Assistance to Needy Families (TANF) Receipt

```
%let cEndYr = 121;              *(2021-1900);
%let cEndMo = 1464;     *(121+1)*12;
%let cYrs = 44;                         *(2021-1978)+1;
%let cMos = 528;                *44*12;
%let cFillFlag = 2020;

data Q13;
 set prg2020.in_inc_2020;

rename
'Q13RECIP-4~000001_2020'n    =      recip41
'Q13RECIP-4~000002_2020'n    =      recip42
```

```
'Q13RECIP-4~000003_2020'n    =      recip43
'Q13RECIP-4~000004_2020'n    =      recip44
'Q13A-BB_2020'n              =      QBB
'Q13A-43B_2020'n             =      Q43B
'Q13A-43C~M_2020'n           =      Q43CM
'Q13A-43C~Y_2020'n           =      Q43CY
'Q13A-44A1_CHK1.01_2020'n    =      Q44A1_CHK11
'Q13A-44A1_CHK1.02_2020'n    =      Q44A1_CHK12
'Q13A-44A3.01_2020'n         =      Q44A301
'Q13A-44A3.02_2020'n         =      Q44A302
'Q13A-44A5.01_2020'n         =      Q44A501
'Q13A-44A6.01_2020'n         =      Q44A601
'Q13A-44B.01_2020'n          =      Q44B01
'Q13A-44B.02_2020'n          =      Q44B02
'Q13A-44C.02~M_2020'n        =      Q44C02M
'Q13A-44C.02~Y_2020'n        =      Q44C02Y
'Q13A-44D.01_2020'n          =      Q44D01
'Q13A-44D.02_2020'n          =      Q44D02
'Q13A-45A.01_2020'n          =      Q45A01
'Q13A-45B.01~M_2020'n        =      Q45B01M
'Q13A-45B.01~Y_2020'n        =      Q45B01Y
'Q13A-47E.01_2020'n          =      Q47E01
'Q13A-47E.02_2020'n          =      Q47E02
'Q13A-62.01_2020'n           =      Q6201
'Q13A-62.02_2020'n           =      Q6202
'Q13A-62A.01_2020'n          =      Q62A01
'Q13A-63A.01_2020'n          =      Q63A01
'Q13A-63B.01_2020'n          =      Q63B01
'Q13A-63C.01_2020'n          =      Q63C01
'Q13A-63D.01_2020'n          =      Q63D01
'Q13A-64.01_2020'n           =      Q6401
'Q13A-64A.01_2020'n          =      Q64A01
'Q13A-64A.02_2020'n          =      Q64A02
'CURDATE~D_2020'n            =      intDay
'CURDATE~M_2020'n            =      intMo
'CURDATE~Y_2020'n            =      intYr;

data recip1;
 merge R29.lintdate2020 Q13 prg2020.oldin2020 (keep=norcid aydolo:);
 by norcid;
 if (intMo > 0);

recip = RECIP41;
*****************************************************************;
proc freq;
 tables lintyr lint;
```

```
 where intYr~=-4;
 title "DLI year and month after DLI month for those interviewed in R29";
proc freq;
 tables lintyr lIntMo lint;
 where recip not in (-5 0);
 title "DLI year and month after DLI month for those whose AFDC EH need to be updated in
R29";
run;

data recip;
 set recip1;

* convert Y2K changes to simple years;
array yrs (*) intYr q43cy q44c02y 45b01y;
do i=1 to dim(yrs);
   if yrs(i) > 0 then yrs(i)=yrs(i)-1900;
end;

* date of this interview;
intDt=intYr*12+intMo;

/*============================================================
  Merge info from several sources + convert weeks to months.
  ============================================================ */
dummy = -4;
if q44c01y>-4 and q43cy>-4 and q44c01y<q43cy then do;
        q44c01y=q43cy;
        q44c01m=q43cm;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y>q43cy then do;
        q44c01y=q44c01y;
        q44c01m=q44c01m;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y=q43cy then do;
        q44c01m = max(q44c01m, q43cm);
        q44c01y = max(q44c01y, q43cy);
end;
else if q44c01y<=-4 and q43cy>-4 then do;
        q44c01m = max(q44c01m, q43cm);
        q44c01y = max(q44c01y, q43cy);
end;

/* nrOfRMos - # of months x was received (used when start/stop dates are not known).*/
array nrOfRMos (*) q63aa01 q63aa02;
nrOfRMos(1) = max(nrOfRMos(1), q63a01, q63ab01);
```

```
* How much did R receive in each spell;
array howMuch (*)  q47e01-q47e02;
array q63b    (*)  q63b01-q63b02;
do s=1 to dim(howMuch);
   if howMuch(s) = -4 then howMuch(s) = q63b(s);
end;

/*============================================================
  Recipiency status (rec) and amounts (dol) by months
  All months from 1978jan till dec of current year, indexed from 1900jan.
  78*12+1 = 937
  ============================================================ */


array rec (937:&cEndMo);      * Did they receive x in that month?;
array dol (937:&cEndMo);      * How much?;

/*============================================================
 * No/unknown recipiency
 *
============================================================ */
if recip=0 then do;
   do m=lint to intDt;
      rec(m)=-4;
      dol(m)=-4;
   end;
end;
else if recip in (-1,-2) then do;
   do m=lint to intDt;
      rec(m) = recip;
      dol(m) = recip;
   end;
end;
else if recip in (-3) then do;
   do m=lint to intDt;
      rec(m) = recip;
      dol(m) = recip;
   end;
end;

/*============================================================ */
/* There was some recipiency, get details */
/* ============================================================ */

else do;
check = 0;
```

```
array startMo (*) q44c01m q44c02m;
array startYr (*) q44c01y q44c02y;
array endMo (*) q45b01m q45b02m;
array endYr (*) q45b01y q45b02y;
array cont    (*) q44d01 q44d02;
array q44b (*) q44b01 q44b02;
array q64     (*) q6401 Q6402;
array q63c    (*) q63c01 Q63c02;

/* ---------------------------------------------------------------------------
 * Calculate starts and ends of spells if possible
 * ---------------------------------------------------------------------------*/

array starts (*) starts1-starts2;
array ends   (*) ends1-ends2;

* Continuous recipiency "ends" at interview date;
do s = 1 to dim(cont);
          if cont(s) = 1 or q63c(s)=12 then do;
                    endYr(s) = intYr;
                    endMo(s) = intMo;
          end;
end;

* Compute starts/stops/last for basic cases (dates are known);
do s = 1 to dim(starts);
   if startMo(s) > 0 & startYr(s) > 0 then
      starts(s) = startYr(s)*12+startMo(s);
   if endMo(s) > 0 & endYr(s) > 0 then
      ends(s) = endYr(s)*12+endMo(s);
end;

seamFlag=0;
do s = 1 to dim(starts);
          if starts(s) > 0 then do;
       if starts(s)=lint   then seamFlag=1;
       if starts(s)=lint-1 then seamFlag=2;
                    if 0<starts(s)<lint-1 then seamFlag=3;
          end;
end;

lint_real=lint-936-1;
do s=1 to dim(starts);
 if starts(s) > 0 & starts(s)<lint then starts(s)=lint;
end;
```

```
* Try to deduce starts/ends from # of mos;
do s = 1 to dim(starts);
   if nrOfRMos(s) > 0 then do;
            if ends(s) > 0 & starts(s) = .  then do;
         starts(s) = ends(s) - nrOfRMos(s)+1;
                        check=5.1;
         if (starts(s) < lint) then check = 5;
      end;
                else if starts(s) > 0 & ends(s) = .  then do;
         ends(s) = starts(s) + nrOfRMos(s) - 1;
                        check=100+s+0.1;
                        if ends(s)>intDt then check=100+s;
      end;
                else if starts(s) = .  & ends(s) = .  then do;
         if s = 1 then do; starts(s) = lint; check=201+0.1; end;
                        else do; starts(s) = ends(s-1) + 1; check=200+s+0.1; end;
         ends(s) = starts(s) + nrOfRMos(s) - 1;
                        if ends(s)>intDt then check=200+s;
      end;
   end;
end;

/* Determine what was the last spell (0 - there was no spell) */

last = 0;
if q44b01=2 or q43B = 2 then last = 1;
do s = 1 to dim(starts);
         if ((startMo(s) > 0 | startYr(s) > 0 | endMo(s) > 0 | endYr(s) > 0 |nrOfRMos (s) > 0) & last
< s) then last = s;
end;

/* ----------------------------------------------------------------------------
 * Edges of recipiency spells and periods between: n1 r1 n2 r2 ...  (NNNRRR???NRRR)
 * Under ideal conditions they touch, but often there is an unclear zone between them (-2)
 * ---------------------------------------------------------------------------- */
array rs(*) rs1-rs2;              * Recipiency starts;
array re(*) re1-re2;              * Recipiency ends;
array ns(*) ns1-ns3;              * Nonrecipiency starts (# of spells + 1, potentialy bef and after);
array ne(*) ne1-ne3;              * Nonrecipiency ends;

* The fully known cases;
do s = 1 to last;
   if starts(s) > 0 then do;
      ne(s) = starts(s)-1;
      rs(s) = starts(s);
   end;
```

```
   if ends(s) > 0 then do;
      re(s)   = ends(s);
      ns(s+1) = ends(s)+1;
   end;
end;

* The unknown and partially known cases;
do s = 1 to last;
   if starts(s) = . & startYr(s) > 0 then do;
                  check=301;
      xne = startYr(s)*12;
      if ( s = 1 & lint <= xne ) |
        ( s > 1 & 0 < ends(s-1) < xne) |
        ( s > 1 & 0 < endYr(s-1) + 1 < startYr(s) )
         then do; ne(s) = xne; check=302; end;
      if startYr(s) < endYr(s) then do;
         rs(s) = startYr(s)*12+12;
                        check=303;
                  end;
   end;

   if (ends(s) = . & endYr(s) > 0) then do;
                  check=401;
      if 0 < startYr(s) < endYr(s) then do;
         re(s) = endYr(s)*12+1;
         ns(s+1) = endYr(s)*12+13;
                        check=402;
                  end;
                  if 0 < startYr(s) = endYr(s) then do;
                    ns(s+1)=endYr(s)*12+13;
                    if rs(s)>0 then do;
                           re(s)=rs(s);
                    end;
                    check=403;
                  end;
   end;
end;

if (ne(1) > 0) then ns(1) = lint;                              * There was a nonrecipiency at the
begining;
if (ns(last+1) > 0) then ne(last+1) = intDt;     * There was a nonrecipiency at the end;

if last>=2 then do s = 2 to last;
          if 0 < rs(s) & rs(s) = re(s-1) then check = 7;
end;
 if rs1 > 0 & rs1<lint then rs1=lint;
```

```
/* ---------------------------------------------------------------------------
 * Fill the recipiency status (rec) and amounts (dol).
 * --------------------------------------------------------------------------- */

* First, assume we know nothing;
do m = lint to intDt;
        rec(m) = -2;
        dol(m) = -2;
end;

* Fill in info we know (any of the periods below can be empty);
do s = 1 to last;
        * nonrecipiency period;
        if (0 < ns(s) <= ne(s)) then
                do m = max(ns(s),lint) to min(ne(s),intDt);
                        rec(m) = -4;
                        dol(m) = -4;
                end;
        * recipiency period;
        if (0 < rs(s) <= re(s)) then
                do m = max(rs(s),lint) to min(re(s),intDt);
                        rec(m) = 1;
                        dol(m) = howMuch(s);
                        if howmuch(s) in (. -4) then do; dol(m)=-2; check5=s+0.1; end;
                end;
           * we know only start - we are sure R received smth that month;
      else if (lint <= rs(s) <= intDt & re(s) = .) then do;
        rec(rs(s)) = 1;
        dol(rs(s)) = howMuch(s);
                if howmuch(s) in (. -4) then do;  dol(rs(s))=-2; check5=s+0.2; end;
      end;
      * we know only end - we are sure R received smth that month (can be even intDt if
continuous);
      else if (rs(s) = . & lint <= re(s) <= intDt) then do;
        rec(re(s)) = 1;
        dol(re(s)) = howMuch(s);
                if howmuch(s) in (. -4) then do; dol(re(s))=-2; check5=s+0.3; end;
      end;
end;

* The last period of nonrecipiency;
if (0 < ns(last+1) <= ne(last+1)) then
        do m = ns(last+1) to min(ne(last+1),intDt);
                rec(m) = -4;
                dol(m) = -4;
```

```
        end;

end;

/* ========================================================== */
/* End if recip = .... (getting details for people who got something)
/*========================================================== */

* Fill in space at the end of year after the interview;
do m=intDt+1 to hbound(rec);
   rec(m)=-4;
   dol(m)=-4;
end;

endDt=intDt-936;
noInt=0;
if q44c01m in (-1,-2) & q44c01y in (-1,-2) then q43b = 2;

/* ========================================================== */

array flag(937:&cEndMo) flag1-flag&cMos;

array recYM(&cYrs,12) rec1-rec&cMos;
array dolYM(&cYrs,12) dol1-dol&cMos;


* Set fill flag for ints/non-ints;
if noInt in (0,1) then
        do i=lint to intDt;
                flag(i)=&cFillFlag;
        end;
/* ============================================================
 * fill in missings: -4 if int, -5 if non-int
 * ========================================================== */
if noInt=0 then
        do i=lbound(rec) to hbound(rec);
                if rec(i)=. then rec(i)=-4;
                if dol(i)=. then dol(i)=-4;
                if flag(i)=. then flag(i)=-4;
        end;
if noInt=1 then
        do i=lbound(rec) to hbound(rec);
                if rec(i)=. then rec(i)=-5;
                if dol(i)=. then dol(i)=-5;
                if flag(i)=. then flag(i)=-4;
        end;
```

```
/* ============================================================
 * Calculating yearly variables.
 * ============================================================ */

array yRec(*)   yRec78-yRec&cEndYr;
array yDol(*)   yDol78-yDol&cEndYr;
array yRecMo(*) yRecMo78-yRecMo&cEndYr;


* Calculate yRec (any recipiency in a yr?);
do y=1 to dim(yRec);
   do m=1 to 12;
      if recYM(y,m) ~= . then    yRec(y)=0;
      if recYM(y,m) > 0 then do; yRec(y)=1; leave; end;
   end;
         if yRec(y)=0 then do m=1 to 12;
            if recYM(y,m) in (-1 -2 -3) then do;  yRec(y)=recYM(y,m); leave; end;
         end;
         if yRec(y)=0 then yRec(y)=-4;
end;

* Calculate yDol ($ in yr), yRecMo (# of months of recipiency in each y);
do y=1 to dim(yRec);
   if yRec(y) > 0 then do;
                  rrefuse = 0; drefuse = 0;
                  rdk = 0;     ddk = 0;
      do m=1 to 12;
                        if dolYM(y,m) = -1 then drefuse = 1;
                        else if dolYM(y,m) = -2 then ddk = 1;
         if recYM(y,m) = -1 then rrefuse = 1;
                        else if recYM(y,m) = -2 then rdk = 1;
                  end;
                  if rrefuse = 1 then yRecMo(y) = -1;
                  else if rdk = 1 then yRecMo(y) = -2;
                  else do;
               yRecMo(y) = 0;
               do m=1 to 12;
                  if recYM(y,m) = 1 then yRecMo(y)=yRecMo(y)+1;
               end;
            end;
            if rrefuse = 1 or drefuse = 1 then yDol(y) = -1;
                  else if rdk = 1 or ddk = 1 then yDol(y) = -2;
                  else do;
               yDol(y) = 0;
               do m=1 to 12;
                  if 0 <= dolYM(y,m) then yDol(y)=yDol(y)+dolYM(y,m);
```

```
            end;
          end;
        end;
        else do;
            yDol(y)=yRec(y);
            yRecMo(y)=yRec(y);
        end;
end;

/* ============================================================
 *  Flag yearly amount that are not within possible bounds.
 * ============================================================ */

array yFlg(*)   yFlg78-yFlg&cEndYr;
array cMinDol (&cYrs) _temporary_ (&cYrs*44);
array cMaxDol (&cYrs) _temporary_
        (537 579 586 622 643 670 700 726 742 771 794 815 809 786 875 844 827 808 808 949
949 1025 1025 1134 1134 19*1134);

do y=1 to dim(yRec);
   yFlg(y) = 0;
   if yRec(y) = -5 then yFlg(y)=-5;
   if yDol(y) < 0 then continue;
        avgDol = yDol(y) / yRecMo(y);
   if avgDol < cMinDol(y) then
            yFlg(y)=1;
        else if cMaxDol(y) < avgDol then do;
      yFlg(y)=2;
      if cMinDol(y) <= avgDol / yRecMo(y) <= cMaxDol(y) then yFlg(y)=3;
   end;
end;

/* ============================================================
 *
 * ============================================================ */

do y=1 to dim(yRec);
   if yDol(y) = .  then yDol(y)=-4;
   if yRec(y) = -4 then yRec(y)=0;
end;

yflag=0;
array ydolo(*)   ayDolo78-ayDolo&cEndYr;
do i=1 to dim(ydol);
 if lintyr=i+77 & 0<lintmo<13 then do;
  if ydol(i)=-4 then do; ydol(i)=ydolo(i); yflag=1; end;
```

```
  else if ydol(i)>=0 & ydolo(i) not in (-4 0) then do;
    if ydolo(i)>0 then do; ydol(i)=ydol(i)+ydolo(i); yflag=2; end;
          else if ydolo(i) in (-1 -2 -3) then do; ydol(i)=ydolo(i); yflag=3; end;
  end;
 end;
end;
```

## Food Stamps/Supplemental Nutrition Assistance Program (SNAP) Receipt

```
%let cEndYr = 121;              *(2021-1900);
%let cEndMo = 1464;     *(121+1)*12;
%let cYrs = 44;                         *(2021-1978)+1;
%let cMos = 528;                *44*12;
%let cFillFlag = 2020;

data Q13;
 set prg2020.in_inc_2020;

rename
 'Q13RECIP-4~000001_2020'n   =  RECIP41
 'Q13RECIP-4~000002_2020'n   =  RECIP42
 'Q13RECIP-4~000003_2020'n   =  RECIP43
 'Q13RECIP-4~000004_2020'n   =  RECIP44
 'Q13F-B_2020'n            =    qB
 'Q13F-43B_2020'n          =  q43B
 'Q13F-43C~M_2020'n        =  q43CM
 'Q13F-43C~Y_2020'n        =  q43CY
 'Q13F-44A1_CHK1.01_2020'n   =  q44A1CHK101
 'Q13F-44A1_CHK1.02_2020'n   =  q44A1CHK102
 'Q13F-44A3.01_2020'n       =  q44A301
 'Q13F-44A3.02_2020'n       =  q44A302
 'Q13F-44A5.01_2020'n       =  q44A501
 'Q13F-44A6.01_2020'n       =  q44A601
 'Q13F-44B.01_2020'n        =  q44B01
 'Q13F-44B.02_2020'n        =  q44B02
 'Q13F-44C.01~M_2020'n      =  q44C01M
 'Q13F-44C.01~Y_2020'n      =  q44C01Y
 'Q13F-44C.02~M_2020'n      =  q44C02M
 'Q13F-44C.02~Y_2020'n      =  q44C02Y
 'Q13F-44D.01_2020'n        =  q44D01
 'Q13F-44D.02_2020'n        =  q44D02
 'Q13F-45A.01_2020'n        =  q45A01
 'Q13F-45B.01~M_2020'n      =  q45B01M
 'Q13F-45B.01~Y_2020'n      =  q45B01Y
 'Q13F-47E.01_2020'n        =  q47E01
```

```
 'Q13F-47E.02_2020'n        =  q47E02
 'Q13F-62.01_2020'n         =  q6201
 'Q13F-62.02_2020'n         =  q6202
 'Q13F-62A.01_2020'n        =  q62A01
 'Q13F-63.01_2020'n         =  q6301
 'Q13F-63A.01_2020'n        =  q63A01
 'Q13F-63AA.01_2020'n       =  q63AA01
 'Q13F-63AA.02_2020'n       =  q63AA02
 'Q13F-63AB.01_2020'n       =  q63AB01
 'Q13F-63B.01_2020'n        =  q63B01
 'Q13F-63B.02_2020'n        =  q63B02
 'Q13F-63C.01_2020'n        =  q63C01
 'Q13F-63D.01_2020'n        =  q63D01
 'Q13F-63D.02_2020'n        =  q63D02
 'Q13F-64.01_2020'n         =  q6401
 'Q13F-64A.01_2020'n        =  q64A01
 'Q13F-64A.02_2020'n        =  q64A02
 'CURDATE~D_2020'n          =  intDay
 'CURDATE~M_2020'n          =  intMo
 'CURDATE~Y_2020'n          =  intYr
;

data recip1;
 merge R29.lintdate2020 Q13 prg2020.oldin2020 (keep=norcid fydolo:);
 by norcid;
 if (intMo > 0);

recip = RECIP42;

proc freq;
 tables lintyr lint recip;
 where recip not in (-5 0);
 title "DLI year and month after DLI month for those whose FDSTMPS EH need to be updated
in R28";
run;

data recip;
 set recip1;

* convert Y2K changes to simple years;
array yrs (*) intYr q43cy
              q44c01y q44c02y
              q45b01y;

do i=1 to dim(yrs);
   if yrs(i) > 0 then yrs(i)=yrs(i)-1900;
```

```
end;

* date of this interview;
intDt=intYr*12+intMo;


/* ================================================================
  Merge info from several sources + convert weeks to months.
  ================================================================ */
dummy = -4;
if q44c01y>-4 and q43cy>-4 and q44c01y<q43cy then do;
        q44c01y=q43cy;
        q44c01m=q43cm;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y>q43cy then do;
        q44c01y=q44c01y;
        q44c01m=q44c01m;
end;
else if q44c01y>-4 and q43cy>-4 and q44c01y=q43cy then do;
        q44c01m = max(q44c01m, q43cm);
        q44c01y = max(q44c01y, q43cy);
end;
else if q44c01y<=-4 and q43cy>-4 then do;
        q44c01m = max(q44c01m, q43cm);
        q44c01y = max(q44c01y, q43cy);
end;

/* nrOfRMos - # of months x was received (used when start/stop dates are not known).*/
array nrOfRMos  (*) q63aa01 q63aa02;
nrOfRMos(1) = max(nrOfRMos(1), q63a01, q63ab01);

* How much did R receive in each spell ;
array howMuch (*) q47e01-q47e02;
array q63b    (*) q63b01-q63b02;

do s=1 to dim(howMuch);
   if howMuch(s) = -4 then howMuch(s) = q63b(s);
end;


/* ================================================================
   Recipiency status (rec) and amounts (dol) by months
   All months from 1978jan till dec of current year, indexed from 1900jan.
   78*12+1 = 937
   ================================================================ */
```

```
array rec (937:&cEndMo);      * Did they receive x in that month?;
array dol (937:&cEndMo);      * How much?;

/* ================================================================
 * No/unknown recipiency
 *
================================================================ */
if recip=0 then do;
   do m=lint to intDt;
      rec(m)=-4;
      dol(m)=-4;
   end;
end;
else if recip in (-1,-2) then do;
   do m=lint to intDt;
      rec(m) = recip;
      dol(m) = recip;
   end;
end;
else if recip in (-3) then do;
   do m=lint to intDt;
      rec(m) = recip;
      dol(m) = recip;
   end;
end;

/* ================================================================
 * There was some recipiency, get details
 *
================================================================ */
else do;
check = 0;
array startMo (*) q44c01m q44c02m;
array startYr (*) q44c01y q44c02y;
array endMo (*) q45b01m q45b02m;
array endYr (*) q45b01y q45b02y;
array cont   (*) q44d01-q44d02;
array q44b (*) q44b01 q44b02;
array q64  (*) q6401 Q6402;
array q63c (*) q63c01 Q63c02;

/* ----------------------------------------------------------------------
 * Calculate starts and ends of spells if possible
 * ----------------------------------------------------------------------*/
array starts (*) starts1-starts2;
array ends   (*) ends1-ends2;
```

```
* Continuous recipiency "ends" at interview date;
do s = 1 to dim(cont);
          if cont(s) = 1 or q63c(s)=9 then do;
       endYr(s) = intYr;
       endMo(s) = intMo;
   end;
end;

* Compute starts/stops/last for basic cases (dates are known);
do s = 1 to dim(starts);
   if startMo(s) > 0 & startYr(s) > 0 then
      starts(s) = startYr(s)*12+startMo(s);
   if endMo(s) > 0 & endYr(s) > 0 then
      ends(s) = endYr(s)*12+endMo(s);
end;

seamFlag=0;
do s = 1 to dim(starts);
          if starts(s) > 0 then do;
       if starts(s)=lint   then seamFlag=1;
       if starts(s)=lint-1 then seamFlag=2;
                   if 0<starts(s)<lint-1 then seamFlag=3;
          end;
end;

lint_real=lint-936-1;
do s=1 to dim(starts);
  if starts(s) > 0 & starts(s)<lint then starts(s)=lint;
end;

* Try to deduce starts/ends from # of mos;
do s = 1 to dim(starts);
   if nrOfRMos(s) > 0 then do;
              if ends(s) > 0 & starts(s) = .  then do;
          starts(s) = ends(s) - nrOfRMos(s)+1;
                              check=5.1;
          if (starts(s) < lint) then check = 5;
       end;
                   else if starts(s) > 0 & ends(s) = .  then do;
          ends(s) = starts(s) + nrOfRMos(s) - 1;
                              check=100+s+0.1;
                              if ends(s)>intDt then check=100+s;
       end;
                   else if starts(s) = .  & ends(s) = .  then do;
          if s = 1 then do; starts(s) = lint; check=201+0.1; end;
```

```
                        else do; starts(s) = ends(s-1) + 1; check=200+s+0.1; end;
        ends(s) = starts(s) + nrOfRMos(s) - 1;
                        if ends(s)>intDt then check=200+s;
      end;
   end;
end;

/* Determine what was the last spell (0 - there was no spell)*/
last = 0;
if q44b01=2 or q43B = 2 then last = 1;
do s = 1 to dim(starts);
   if ((startMo(s) > 0 | startYr(s) > 0 | endMo(s) > 0 | endYr(s) > 0 |nrOfRMos (s) > 0) & last < s)
then last = s;
end;

/* ---------------------------------------------------------------------------
 * Edges of recipiency spells and periods between: n1 r1 n2 r2 ...  (NNNRRR???NRRR)
 * Under ideal conditions they touch, but often there is an unclear zone between them (-2)
 * --------------------------------------------------------------------------- */
array rs(*) rs1-rs2;       * Recipiency starts;
array re(*) re1-re2;       * Recipiency ends;
array ns(*) ns1-ns3;       * Nonrecipiency starts (# of spells + 1, potentialy bef and after);
array ne(*) ne1-ne3;       * Nonrecipiency ends;

* The fully known cases;
do s = 1 to last;
   if starts(s) > 0 then do;
      ne(s) = starts(s)-1;
      rs(s) = starts(s);
   end;
   if ends(s) > 0 then do;
      re(s)   = ends(s);
      ns(s+1) = ends(s)+1;
   end;
end;


* The unknown and partially known cases;
do s = 1 to last;
   if starts(s) = . & startYr(s) > 0 then do;
                  check=301;
      xne = startYr(s)*12;
      if ( s = 1 & lint <= xne ) |
        ( s > 1 & 0 < ends(s-1) < xne) |
        ( s > 1 & 0 < endYr(s-1) + 1 < startYr(s) )
         then do; ne(s) = xne; check=302; end;
```

```
      if startYr(s) < endYr(s) then do;
         rs(s) = startYr(s)*12+12;
                        check=303;
               end;
   end;
   if (ends(s) = . & endYr(s) > 0) then do;
               check=401;
      if 0 < startYr(s) < endYr(s) then do;
         re(s) = endYr(s)*12+1;
         ns(s+1) = endYr(s)*12+13;
                        check=402;
               end;
               if 0 < startYr(s) = endYr(s) then do;
                 ns(s+1)=endYr(s)*12+13;
                 if rs(s)>0 then do;
                        re(s)=rs(s);
                 end;
                 check=403;
               end;
   end;
end;

if (ne(1) > 0) then ns(1) = lint;              * There was a nonrecipiency at the begining;
if (ns(last+1) > 0) then ne(last+1) = intDt;    * There was a nonrecipiency at the end;

if last>=2 then do s = 2 to last;
         if 0 < rs(s) & rs(s) = re(s-1) then check = 7;
end;

 if rs1 > 0 & rs1<lint then rs1=lint;

/* ---------------------------------------------------------------------------
 * Fill the recipiency status (rec) and amounts (dol).
 * --------------------------------------------------------------------------- */

* First, assume we know nothing;
do m = lint to intDt;
   rec(m) = -2;
   dol(m) = -2;
end;

* Fill in info we know (any of the periods below can be empty);
do s = 1 to last;
   * nonrecipiency period;
   if (0 < ns(s) <= ne(s)) then
      do m = max(ns(s),lint) to min(ne(s),intDt);
```

```
            rec(m) = -4;
            dol(m) = -4;
         end;
      * recipiency period;
      if (0 < rs(s) <= re(s)) then
         do m = max(rs(s),lint) to min(re(s),intDt);
            rec(m) = 1;
            dol(m) = howMuch(s);
                              if howmuch(s) in (. -4) then do; dol(m)=-2; check5=s+0.1; end;
         end;
      * we know only start - we are sure R received smth that month;
      else if (lint <= rs(s) <= intDt & re(s) = .) then do;
         rec(rs(s)) = 1;
         dol(rs(s)) = howMuch(s);
         if howmuch(s) in (. -4) then do;  dol(rs(s))=-2; check5=s+0.2; end;
      end;
      * we know only end - we are sure R received smth that month (can be even intDt if
continuous);
      else if (rs(s) = . & lint <= re(s) <= intDt) then do;
         rec(re(s)) = 1;
         dol(re(s)) = howMuch(s);
                  if howmuch(s) in (. -4) then do; dol(re(s))=-2; check5=s+0.3; end;
      end;
end;

* The last period of nonrecipiency;
if (0 < ns(last+1) <= ne(last+1)) then
   do m = ns(last+1) to min(ne(last+1),intDt);
      rec(m) = -4;
      dol(m) = -4;
   end;

end;
/* ================================================================ */
/* End if recip = .... (getting details for people who got something)
/* ================================================================ */

* Fill in space at the end of year after the interview;
do m=intDt+1 to hbound(rec);
   rec(m)=-4;
   dol(m)=-4;
end;

endDt=intDt-936;
noInt=0;
if q44c01m in (-1,-2) & q44c01y in (-1,-2) then q43b = 2;
```

```
if noInt=0 then
   do i=lbound(rec) to hbound(rec);
      if rec(i)=. then rec(i)=-4;
      if dol(i)=. then dol(i)=-4;
   end;
if noInt=1 then
   do i=lbound(rec) to hbound(rec);
      if rec(i)=. then rec(i)=-5;
      if dol(i)=. then dol(i)=-5;
   end;
/* ================================================================
 * Calculating yearly variables.
 *
================================================================ */
array recYM(&cYrs,12) rec1-rec&cMos;
array dolYM(&cYrs,12) dol1-dol&cMos;
array yRec(*)   yRec78-yRec&cEndYr;
array yDol(*)   yDol78-yDol&cEndYr;
array yRecMo(*) yRecMo78-yRecMo&cEndYr;

* Calculate yRec (any recipiency in a yr?);
do y=1 to dim(yRec);
   do m=1 to 12;
      if recYM(y,m) ~= . then    yRec(y)=0;
      if recYM(y,m) > 0 then do; yRec(y)=1; leave; end;
   end;
        if yRec(y)=0 then do m=1 to 12;
           if recYM(y,m) in (-1 -2 -3) then do;  yRec(y)=recYM(y,m); leave; end;
        end;
        if yRec(y)=0 then yRec(y)=-4;
end;

* Calculate yDol ($ in yr), yRecMo (# of months of recipiency in each y);
do y=1 to dim(yRec);
   if yRec(y) > 0 then do;
                rrefuse = 0; drefuse = 0;
                rdk = 0;     ddk = 0;
      do m=1 to 12;
                        if dolYM(y,m) = -1 then drefuse = 1;
                        else if dolYM(y,m) in (-2 -3) then ddk = 1;
        if recYM(y,m) = -1 then rrefuse = 1;
                        else if recYM(y,m) = -2 then rdk = 1;
                end;

                if rrefuse = 1 then yRecMo(y) = -1;
```

```
                else if rdk = 1 then yRecMo(y) = -2;
                else do;
              yRecMo(y) = 0;
              do m=1 to 12;
                if recYM(y,m) = 1 then yRecMo(y)=yRecMo(y)+1;
              end;
            end;

            if rrefuse = 1 or drefuse = 1 then yDol(y) = -3;
                else if rdk = 1 or ddk = 1 then yDol(y) = -3;
                else do;
              yDol(y) = 0;
              do m=1 to 12;
                if 0 <= dolYM(y,m) then yDol(y)=yDol(y)+dolYM(y,m);
              end;
            end;
          end;
          else do;
                yDol(y)=yRec(y);
                yRecMo(y)=yRec(y);
          end;
end;

/* ===================================================================
 *  Flag yearly amount that are not within possible bounds.
 *
=================================================================== */

array yFlg(*)   yFlg78-yFlg&cEndYr;
array cMinDol (&cYrs) _temporary_ (&cYrs*44);
array cMaxDol (&cYrs) _temporary_
   (244 275 316 344 342 360 364 366 400 415 472 511 549 544 550 555 560 568 568 735 735
835 835 969 20*814);

do y=1 to dim(yRec);
   yFlg(y) = 0;
   if yRec(y) = -5 then yFlg(y)=-5;
   if yDol(y) < 0 then continue;
         avgDol = yDol(y) / yRecMo(y);
   if avgDol < cMinDol(y) then
                yFlg(y)=1;
         else if cMaxDol(y) < avgDol then do;
      yFlg(y)=2;
      if cMinDol(y) <= avgDol / yRecMo(y) <= cMaxDol(y) then yFlg(y)=3;
   end;
end;
```

```
/* ================================================================
 *
 *
================================================================ */

do y=1 to dim(yRec);
   if yDol(y) = .  then yDol(y)=-4;
   if yRec(y) = -4 then yRec(y)=0;
end;

yflag=0;
array ydolo(*)   fyDolo78-fyDolo&cEndYr;
do i=1 to dim(ydol);
 if lintyr=i+77 & 0<lintmo<13 then do;
  if ydol(i)=-4 then do; ydol(i)=ydolo(i); yflag=1; end;
  else if ydol(i)>=0 & ydolo(i) not in (-4 0) then do;
    if ydolo(i)>0 then do; ydol(i)=ydol(i)+ydolo(i); yflag=2; end;
          else if ydolo(i) in (-1 -2 -3) then do; ydol(i)=ydolo(i); yflag=3; end;
  end;
 end;
end;
```

## Respondent/Dependent Child/Spouse/Partner Supplemental Security Income (SSI) or Supplemental Security Disability Income (SSDI) Receipt

```
%let cEndYr = 121;              *(2021-1900);
%let cEndMo = 1464;     *(121+1)*12;
%let cYrs = 44;                         *(2021-1978)+1;
%let cMos = 528;                *44*12;
%let cFillFlag = 2020;

data Q13;
 set prg2020.in_inc_2020;

rename
'Q13-37_2020'n                  =       'Q37'n
'Q13SSI-43A~000001_2020'n               =       'qr43A1'n
'Q13SSI-43A~000002_2020'n               =       'qr43A2'n
'Q13SSI-43A~000003_2020'n               =       'qr43A3'n
'Q13SSI-43A-NOKIDS_2020'n               =       'qr43ANOKIDS'n
'Q13SSI-A_2020'n                =       'qrA'n
'Q13SSI-B1_2020'n               =       'qrB1'n
'Q13SSI-B2_2020'n               =       'qrB2'n
```

'Q13SSI-B3_2020'n = 'qrB3'n
'Q13SSI-43D~000001_2020'n = 'qr43D1'n
'Q13SSI-43D~000002_2020'n = 'qr43D2'n
'Q13SSI-43D~000003_2020'n = 'qr43D3'n
'Q13SSI-43C.01~M_2020'n = 'qr43C1m'n
'Q13SSI-43C.01~Y_2020'n = 'qr43C1y'n
'Q13SSI-43C.02~M_2020'n = 'qr43C2m'n
'Q13SSI-43C.02~Y_2020'n = 'qr43C2y'n
'Q13SSI-43C.03~M_2020'n = 'qr43C3m'n
'Q13SSI-43C.03~Y_2020'n = 'qr43C3y'n
'Q13SSI-44A1_CHK1.01.01_2020'n = 'qr44A1CHK111'n
'Q13SSI-44A1_CHK1.01.02_2020'n = 'qr44A1CHK112'n
'Q13SSI-44A1_CHK1.01.03_2020'n = 'qr44A1CHK113'n
'Q13SSI-44A1_CHK1.02.01_2020'n = 'qr44A1CHK121'n
'Q13SSI-44A1_CHK1.02.02_2020'n = 'qr44A1CHK122'n
'Q13SSI-44A1_CHK1.02.03_2020'n = 'qr44A1CHK123'n
'Q13SSI-44A1_CHK1.03.01_2020'n = 'qr44A1CHK131'n
'Q13SSI-44A3.01.01_2020'n = 'qr44A311'n
'Q13SSI-44A3.01.02_2020'n = 'qr44A312'n
'Q13SSI-44A3.01.03_2020'n = 'qr44A313'n
'Q13SSI-44A3.02.01_2020'n = 'qr44A321'n
'Q13SSI-44A3.02.02_2020'n = 'qr44A322'n
'Q13SSI-44A3.02.03_2020'n = 'qr44A323'n
'Q13SSI-44A3.03.01_2020'n = 'qr44A331'n
'Q13SSI-44A5.01.01_2020'n = 'qr44A511'n
'Q13SSI-44A5.02.01_2020'n = 'qr44A521'n
'Q13SSI-44A5.03.01_2020'n = 'qr44A531'n
'Q13SSI-44A6.01.01_2020'n = 'qr44A611'n
'Q13SSI-44A6.02.01_2020'n = 'qr44A621'n
'Q13SSI-44A6.03.01_2020'n = 'qr44A631'n
'Q13SSI-44B.01.02_2020'n = 'qr44B12'n
'Q13SSI-44B.01.03_2020'n = 'qr44B13'n
'Q13SSI-44B.02.02_2020'n = 'qr44B22'n
'Q13SSI-44B.02.03_2020'n = 'qr44B23'n
'Q13SSI-44C.01.02~M_2020'n = 'qr44C12m'n
'Q13SSI-44C.01.02~Y_2020'n = 'qr44C12y'n
'Q13SSI-44C.02.02~M_2020'n = 'qr44C22m'n
'Q13SSI-44C.02.02~Y_2020'n = 'qr44C22y'n
'Q13SSI-44C.02.03~M_2020'n = 'qr44C23m'n
'Q13SSI-44C.02.03~Y_2020'n = 'qr44C23y'n
'Q13SSI-44D.01.01_2020'n = 'qr44D11'n
'Q13SSI-44D.01.02_2020'n = 'qr44D12'n
'Q13SSI-44D.02.01_2020'n = 'qr44D21'n
'Q13SSI-44D.02.02_2020'n = 'qr44D22'n
'Q13SSI-44D.02.03_2020'n = 'qr44D23'n
'Q13SSI-44D.03.01_2020'n = 'qr44D31'n

'Q13SSI-45A.01.01_2020'n = 'qr45A11'n
'Q13SSI-45A.02.01_2020'n = 'qr45A21'n
'Q13SSI-45A.02.02_2020'n = 'qr45A22'n
'Q13SSI-45A.03.01_2020'n = 'qr45A31'n
'Q13SSI-45B.01.01~M_2020'n = 'qr45B11m'n
'Q13SSI-45B.01.01~Y_2020'n = 'qr45B11y'n
'Q13SSI-45B.02.01~M_2020'n = 'qr45B21m'n
'Q13SSI-45B.02.01~Y_2020'n = 'qr45B21y'n
'Q13SSI-45B.02.02~M_2020'n = 'qr45B22m'n
'Q13SSI-45B.02.02~Y_2020'n = 'qr45B22y'n
'Q13SSI-45B.03.01~M_2020'n = 'qr45B31m'n
'Q13SSI-45B.03.01~Y_2020'n = 'qr45B31y'n
'Q13SSI-47E.01.01_2020'n = 'qr47E11'n
'Q13SSI-47E.01.02_2020'n = 'qr47E12'n
'Q13SSI-47E.02.01_2020'n = 'qr47E21'n
'Q13SSI-47E.02.02_2020'n = 'qr47E22'n
'Q13SSI-47E.02.03_2020'n = 'qr47E23'n
'Q13SSI-47E.03.01_2020'n = 'qr47E31'n
'Q13SSI-62.01.01_2020'n = 'qr6211'n
'Q13SSI-62.01.02_2020'n = 'qr6212'n
'Q13SSI-62.02.01_2020'n = 'qr6221'n
'Q13SSI-62.02.02_2020'n = 'qr6222'n
'Q13SSI-62.02.03_2020'n = 'qr6223'n
'Q13SSI-62.03.01_2020'n = 'qr6231'n
'Q13SSI-62A.01.01_2020'n = 'qr62A11'n
'Q13SSI-62A.01.02_2020'n = 'qr62A12'n
'Q13SSI-62A.02.01_2020'n = 'qr62A21'n
'Q13SSI-62A.02.02_2020'n = 'qr62A22'n
'Q13SSI-62A.03.01_2020'n = 'qr62A31'n
'Q13SSI-63.01.01_2020'n = 'qr6311'n
'Q13SSI-63.03.01_2020'n = 'qr6331'n
'Q13SSI-63AB.01.01_2020'n = 'qr63AB11'n
'Q13SSI-63B.01.01_2020'n = 'qr63B11'n
'Q13SSI-63C.01.01_2020'n = 'qr63C11'n
'Q13SSI-63C.01.02_2020'n = 'qr63C12'n
'Q13SSI-63C.02.01_2020'n = 'qr63C21'n
'Q13SSI-63C.02.02_2020'n = 'qr63C22'n
'Q13SSI-63C.03.01_2020'n = 'qr63C31'n
'Q13SSI-63D.01.01_2020'n = 'qr63D11'n
'Q13SSI-63D.01.02_2020'n = 'qr63D12'n
'Q13SSI-63D.02.01_2020'n = 'qr63D21'n
'Q13SSI-63D.02.02_2020'n = 'qr63D22'n
'Q13SSI-63D.03.01_2020'n = 'qr63D31'n
'Q13SSI-64.01.01_2020'n = 'qr6411'n
'Q13SSI-64.01.02_2020'n = 'qr6412'n
'Q13SSI-64.02.01_2020'n = 'qr6421'n

'Q13SSI-64.02.02_2020'n = 'qr6422'n
'Q13SSI-64.03.01_2020'n = 'qr6431'n
'Q13SSI-64A.01.01_2020'n = 'qr64A11'n
'Q13SSI-64A.01.02_2020'n = 'qr64A12'n
'Q13SSI-64A.02.01_2020'n = 'qr64A21'n
'Q13SSI-64A.02.02_2020'n = 'qr64A22'n
'Q13SSI-64A.02.03_2020'n = 'qr64A23'n
'Q13SSI-64A.03.01_2020'n = 'qr64A31'n

'Q13SSI-SP-43A_2020'n = 'qs43A'n
'Q13SSI-SP-43D~000001_2020'n = 'qs43D1'n
'Q13SSI-SP-43D~000002_2020'n = 'qs43D2'n
'Q13SSI-SP-43D~000003_2020'n = 'qs43D3'n
'Q13SSI-SP-44A1_CHK1.01.01_2020'n = 'qs44A1CHK111'n
'Q13SSI-SP-44A1_CHK1.01.02_2020'n = 'qs44A1CHK112'n
'Q13SSI-SP-44A1_CHK1.01.03_2020'n = 'qs44A1CHK113'n
'Q13SSI-SP-44A1_CHK1.02.01_2020'n = 'qs44A1CHK121'n
'Q13SSI-SP-44A1_CHK1.02.02_2020'n = 'qs44A1CHK122'n
'Q13SSI-SP-44A1_CHK1.02.03_2020'n = 'qs44A1CHK123'n
'Q13SSI-SP-44A1_CHK1.03.01_2020'n = 'qs44A1CHK131'n
'Q13SSI-SP-44A1_CHK1.03.02_2020'n = 'qs44A1CHK132'n
'Q13SSI-SP-44A3.01.01_2020'n = 'qs44A311'n
'Q13SSI-SP-44A3.01.02_2020'n = 'qs44A312'n
'Q13SSI-SP-44A3.01.03_2020'n = 'qs44A313'n
'Q13SSI-SP-44A3.02.01_2020'n = 'qs44A321'n
'Q13SSI-SP-44A3.02.02_2020'n = 'qs44A322'n
'Q13SSI-SP-44A3.02.03_2020'n = 'qs44A323'n
'Q13SSI-SP-44A3.03.01_2020'n = 'qs44A331'n
'Q13SSI-SP-44A3.03.02_2020'n = 'qs44A332'n
'Q13SSI-SP-44A5.01.01_2020'n = 'qs44A511'n
'Q13SSI-SP-44A5.02.01_2020'n = 'qs44A521'n
'Q13SSI-SP-44A5.03.01_2020'n = 'qs44A531'n
'Q13SSI-SP-44A6.01.01_2020'n = 'qs44A611'n
'Q13SSI-SP-44A6.02.01_2020'n = 'qs44A621'n
'Q13SSI-SP-44A6.03.01_2020'n = 'qs44A631'n
'Q13SSI-SP-44B.01.01_2020'n = 'qs44B11'n
'Q13SSI-SP-44B.01.02_2020'n = 'qs44B12'n
'Q13SSI-SP-44B.01.03_2020'n = 'qs44B13'n
'Q13SSI-SP-44B.02.01_2020'n = 'qs44B21'n
'Q13SSI-SP-44B.02.02_2020'n = 'qs44B22'n
'Q13SSI-SP-44B.02.03_2020'n = 'qs44B23'n
'Q13SSI-SP-44B.03.01_2020'n = 'qs44B31'n
'Q13SSI-SP-44B.03.02_2020'n = 'qs44B32'n
'Q13SSI-SP-44C.01.01~M_2020'n = 'qs44C11m'n
'Q13SSI-SP-44C.01.01~Y_2020'n = 'qs44C11y'n
'Q13SSI-SP-44C.01.02~M_2020'n = 'qs44C12m'n

'Q13SSI-SP-44C.01.02~Y_2020'n = 'qs44C12y'n
'Q13SSI-SP-44C.01.03~M_2020'n = 'qs44C13m'n
'Q13SSI-SP-44C.01.03~Y_2020'n = 'qs44C13y'n
'Q13SSI-SP-44C.02.01~M_2020'n = 'qs44C21m'n
'Q13SSI-SP-44C.02.01~Y_2020'n = 'qs44C21y'n
'Q13SSI-SP-44C.02.02~M_2020'n = 'qs44C22m'n
'Q13SSI-SP-44C.02.02~Y_2020'n = 'qs44C22y'n
'Q13SSI-SP-44C.03.01~M_2020'n = 'qs44C31m'n
'Q13SSI-SP-44C.03.01~Y_2020'n = 'qs44C31y'n
'Q13SSI-SP-44D.01.01_2020'n = 'qs44D11'n
'Q13SSI-SP-44D.01.02_2020'n = 'qs44D12'n
'Q13SSI-SP-44D.01.03_2020'n = 'qs44D13'n
'Q13SSI-SP-44D.02.01_2020'n = 'qs44D21'n
'Q13SSI-SP-44D.02.02_2020'n = 'qs44D22'n
'Q13SSI-SP-44D.03.01_2020'n = 'qs44D31'n
'Q13SSI-SP-45A.02.01_2020'n = 'qs45A21'n
'Q13SSI-SP-45B.02.01~M_2020'n = 'qs45B21m'n
'Q13SSI-SP-45B.02.01~Y_2020'n = 'qs45B21y'n
'Q13SSI-SP-47E.01.01_2020'n = 'qs47E11'n
'Q13SSI-SP-47E.01.02_2020'n = 'qs47E12'n
'Q13SSI-SP-47E.01.03_2020'n = 'qs47E13'n
'Q13SSI-SP-47E.02.01_2020'n = 'qs47E21'n
'Q13SSI-SP-47E.02.02_2020'n = 'qs47E22'n
'Q13SSI-SP-47E.03.01_2020'n = 'qs47E31'n
'Q13SSI-SP-62.01.01_2020'n = 'qs6211'n
'Q13SSI-SP-62.01.02_2020'n = 'qs6212'n
'Q13SSI-SP-62.01.03_2020'n = 'qs6213'n
'Q13SSI-SP-62.02.01_2020'n = 'qs6221'n
'Q13SSI-SP-62.02.02_2020'n = 'qs6222'n
'Q13SSI-SP-62.03.01_2020'n = 'qs6231'n
'Q13SSI-SP-62A.01.01_2020'n = 'qs62A11'n
'Q13SSI-SP-62A.02.01_2020'n = 'qs62A21'n
'Q13SSI-SP-63AA.01.01_2020'n = 'qs63AA11'n
'Q13SSI-SP-63AA.01.02_2020'n = 'qs63AA12'n
'Q13SSI-SP-63AA.02.01_2020'n = 'qs63AA21'n
'Q13SSI-SP-63AA.02.02_2020'n = 'qs63AA22'n
'Q13SSI-SP-63AA.02.03_2020'n = 'qs63AA23'n
'Q13SSI-SP-63AA.03.01_2020'n = 'qs63AA31'n
'Q13SSI-SP-63B.01.01_2020'n = 'qs63B11'n
'Q13SSI-SP-63B.01.02_2020'n = 'qs63B12'n
'Q13SSI-SP-63B.02.01_2020'n = 'qs63B21'n
'Q13SSI-SP-63B.02.02_2020'n = 'qs63B22'n
'Q13SSI-SP-63B.02.03_2020'n = 'qs63B23'n
'Q13SSI-SP-63B.03.01_2020'n = 'qs63B31'n
'Q13SSI-SP-63C.01.01_2020'n = 'qs63C11'n
'Q13SSI-SP-63C.01.02_2020'n = 'qs63C12'n

```
 'Q13SSI-SP-63C.02.01_2020'n          =   'qs63C21'n
 'Q13SSI-SP-63C.02.02_2020'n          =   'qs63C22'n
 'Q13SSI-SP-63C.02.03_2020'n          =   'qs63C23'n
 'Q13SSI-SP-63C.03.01_2020'n          =   'qs63C31'n
 'Q13SSI-SP-63D.01.01_2020'n          =   'qs63D11'n
 'Q13SSI-SP-63D.01.02_2020'n          =   'qs63D12'n
 'Q13SSI-SP-63D.02.01_2020'n          =   'qs63D21'n
 'Q13SSI-SP-63D.02.02_2020'n          =   'qs63D22'n
 'Q13SSI-SP-63D.02.03_2020'n          =   'qs63D23'n
 'Q13SSI-SP-63D.03.01_2020'n          =   'qs63D31'n
 'Q13SSI-SP-64.01.01_2020'n          =   'qs6411'n
 'Q13SSI-SP-64.01.02_2020'n          =   'qs6412'n
 'Q13SSI-SP-64.02.01_2020'n          =   'qs6421'n
 'Q13SSI-SP-64.02.02_2020'n          =   'qs6422'n
 'Q13SSI-SP-64.02.03_2020'n          =   'qs6423'n
 'Q13SSI-SP-64.03.01_2020'n          =   'qs6431'n
 'Q13SSI-SP-64A.01.01_2020'n          =   'qs64A11'n
 'Q13SSI-SP-64A.01.02_2020'n          =   'qs64A12'n
 'Q13SSI-SP-64A.01.03_2020'n          =   'qs64A13'n
 'Q13SSI-SP-64A.02.01_2020'n          =   'qs64A21'n
 'Q13SSI-SP-64A.02.02_2020'n          =   'qs64A22'n
 'Q13SSI-SP-64A.03.01_2020'n          =   'qs64A31'n
 'Q13-10_2020'n                  =      Q10
 'CURDATE~D_2020'n                =          intDay
 'CURDATE~M_2020'n                =    intMo
 'CURDATE~Y_2020'n                =    intYr;

data recip1;
 merge R29.lintdate2020 Q13
 prg2020.oldin2020  (keep=norcid sydolo: dydolo: r29bydolo:);
 by norcid;
 if (intMo > 0);
noInt=0;

proc freq;
 tables lintyr lint;
 where qr43A1 in (1 -1 -2 -3) or qr43A2=1 or qr43ANOKIDS in (1 -1 -2 -3);
 title "DLI year, and month after DLI month for those who or whose children receive SSI/SSDI
in R29";
proc freq;
 tables lintyr lint;
 where qs43a in (1 -1 -2 -3);
 title "DLI year, and month after DLI month for those whose spouse receive SSI/SSDI in R29";

data recip;
 set recip1;
```

```
* convert Y2K changes to simple years;
array yrs (*) intYr
          qr43C1y qr43C2y qr43C3y
                        qr44C12y qr44C22y qr44C23y
                        qr45B11y qr45B21y qr45B22y qr45B31y
                        qs44C11y qs44C12y qs44C13y qs44C21y qs44C22y qs44C31y
                        qs45B21y;

do i=1 to dim(yrs);
   if yrs(i) > 0 then yrs(i)=yrs(i)-1900;
end;

* date of this interview;
intDt=intYr*12+intMo;

/* ================================================================
  Merge info from several sources + convert weeks to months.
  ================================================================ */
dummy = -4;
         qr44c11y=qr43c1y; qr44c21y=qr43c2y; qr44c31y=qr43c3y;
         qr44c11m=qr43c1m; qr44c21m=qr43c2m; qr44c31m=qr43c3m;

* define arrays;
array nrOfRMos (6,3) qr63ab11 dummy dummy
                                        dummy dummy dummy
                                        dummy dummy dummy
                                        qs63aa11 qs63aa12 dummy
                                        qs63aa21 qs63aa22 qs63aa23
                                        qs63aa31 dummy dummy;
array howMuch (6,3) qr47e11 qr47e12 dummy
                                       qr47e21 qr47e22 qr47e23
                                       qr47e31 dummy dummy
                                       qs47e11 qs47e12 qs47e13
                                       qs47e21 qs47e22 dummy
                                       qs47e31 dummy dummy;
array q63b  (6,3)   qr63b11 dummy dummy
                                       dummy dummy dummy
                                       dummy dummy dummy
                                       qs63b11 qs63b12 dummy
                                       qs63b21 qs63b22 qs63b23
                                       qs63b31 dummy dummy;
do i=1 to 6;
  do j=1 to 3;
    if howMuch(i,j) = -4 then howMuch(i,j) = q63b(i,j);
  end;
```

```
end;

array check (*) check1-check6;
do i=1 to 6; check(i)=0; end;

array startMo (6,3) qr44c11m qr44c12m qr44c13m
                                    qr44c21m qr44c22m qr44c23m
                                    qr44c31m qr44c32m qr44c33m
                                    qs44c11m qs44c12m qs44c13m
                                    qs44c21m qs44c22m qs44c23m
                                    qs44c31m qs44c32m qs44c33m;
                                    *qr44c11m qr44c12m dummy
                                    qr44c21m qr44c22m qr44c23m
                                    qr44c31m dummy dummy
                                    qs44c11m qs44c12m qs44c13m
                                    qs44c21m qs44c22m dummy
                                    qs44c31m dummy dummy;
array startYr (6,3) qr44c11y qr44c12y qr44c13y
                                    qr44c21y qr44c22y qr44c23y
                                    qr44c31y qr44c32y qr44c33y
                                    qs44c11y qs44c12y qs44c13y
                                    qs44c21y qs44c22y qs44c23y
                                    qs44c31y qs44c32y qs44c33y;
                                    *qr44c11y qr44c12y dummy
                                    qr44c21y qr44c22y qr44c23y
                                    qr44c31y dummy dummy
                                    qs44c11y qs44c12y qs44c13y
                                    qs44c21y qs44c22y dummy
                                    qs44c31y dummy dummy;

array endMo   (6,3) qr45B11m qr45B12m qr45B13m
                                    qr45B21m qr45B22m qr45B23m
                                    qr45B31m qr45B32m qr45B33m
                                    qs45B11m qs45B12m qs45B13m
                                    qs45B21m qs45B22m qs45B23m
                                    qs45B31m qs45B32m qs45B33m;
                                    *qr45B11m dummy dummy
                                    qr45B21m qr45B22m dummy
                                    qr45B31m dummy dummy
                                    dummy dummy dummy
                                    qs45B21m dummy dummy
                                    dummy dummy dummy;
array endYr   (6,3) qr45B11y qr45B12y qr45B13y
                                    qr45B21y qr45B22y qr45B23y
                                    qr45B31y qr45B32y qr45B33y
                                    qs45B11y qs45B12y qs45B13y
```

qs45B21y qs45B22y qs45B23y
qs45B31y qs45B32y qs45B33y;
*qr45B11y dummy dummy
qr45B21y qr45B22y dummy
qr45B31y dummy dummy
dummy dummy dummy
qs45B21y dummy dummy
dummy dummy dummy;
array cont (6,3) qr44d11 qr44d12 qr44d13
qr44d21 qr44d22 qr44d23
qr44d31 qr44d32 qr44d33
qs44d11 qs44d12 qs44d13
qs44d21 qs44d22 qs44d23
qs44d31 qs44d32 qs44d33;
*qr44d11 qr44d12 dummy
qr44d21 qr44d22 qr44d23
qr44d31 dummy dummy
qs44d11 qs44d12 qs44d13
qs44d21 qs44d22 dummy
qs44d31 dummy dummy;
array q44b (6,3) qr44b11 qr44b12 qr44b13
qr44b21 qr44b22 qr44b23
qr44b31 qr44b32 qr44b33
qs44b11 qs44b12 qs44b13
qs44b21 qs44b22 qs44b23
qs44b31 qs44b32 qs44b33;
*dummy qr44b12 qr44b13
dummy qr44b22 qr44b23
dummy dummy dummy
qs44b11 qs44b12 qs44b13
qs44b21 qs44b22 qs44b23
qs44b31 qs44b32 dummy;
array q43b (*) qr43b1-qr43b3 qs43b1-qs43b3;
array q63c (6,3) qr63c11 qr63c12 qr63c13
qr63c21 qr63c22 qr63c23
qr63c31 qr63c32 qr63c33
qs63c11 qs63c12 qs63c13
qs63c21 qs63c22 qs63c23
qs63c31 qs63c32 qs63c33;
*qr63c11 qr63c12 dummy
qr63c21 qr63c22 dummy
qr63c31 dummy dummy
qs63c11 qs63c12 dummy
qs63c21 qs63c22 qs63c23
qs63c31 dummy dummy;
array starts (6,3) starts11-starts13 starts21-starts23 starts31-starts33

```
                                    starts41-starts43 starts51-starts53 starts61-starts63;
array ends   (6,3) ends11-ends13 ends21-ends23 ends31-ends33
                                    ends41-ends43 ends51-ends53 ends61-ends63;
array seamFlag (*) seamFlag1-seamFlag6;
array last (*) last1-last6;
array rs(6,3) rs11-rs13 rs21-rs23 rs31-rs33
        rs41-rs43 rs51-rs53 rs61-rs63;  * Recipiency starts;
array re(6,3) re11-re13 re21-re23 re31-re33
        re41-re43 re51-re53 re61-re63;  * Recipiency ends;
array ns(6,4) ns11-ns14 ns21-ns24 ns31-ns34
        ns41-ns44 ns51-ns54 ns61-ns64;  * Nonrecipiency starts (# of spells + 1, potentialy bef
and after);
array ne(6,4) ne11-ne14 ne21-ne24 ne31-ne34
        ne41-ne44 ne51-ne54 ne61-ne64;  * Nonrecipiency ends;

array chk5 (*) chk51-chk56;

/* ===================================================================
   Recipiency status (rec) and amounts (dol) by months
   All months from 1978jan till dec of current year, indexed from 1900jan.
   78*12+1 = 937
   =================================================================== */

array rec (6, 937:&cEndMo) rec1_1-rec1_&cMos rec2_1-rec2_&cMos rec3_1-rec3_&cMos
                                    rec4_1-rec4_&cMos rec5_1-rec5_&cMos rec6_1-
rec6_&cMos;
array dol (6, 937:&cEndMo) dol1_1-dol1_&cMos dol2_1-dol2_&cMos dol3_1-dol3_&cMos
                                    dol4_1-dol4_&cMos dol5_1-dol5_&cMos
dol6_1-dol6_&cMos;
array q43d (*) qr43d1-qr43d3 qs43d1-qs43d3;

/* ===================================================================
 * No/unknown recipiency
 *
=================================================================== */
if qr43a3=1 or qr43ANOKIDS=0 then do i=1 to 3;
 do m=lint to intDt;
   rec(i,m)=-4;
   dol(i,m)=-4;
 end;
end;
if q10=0 or qs43a=0 then do i=4 to 6;
 do m=lint to intDt;
   rec(i,m)=-4;
   dol(i,m)=-4;
 end;
```

```
end;
if qr43a1 in (-1 -2 -3) or qr43ANOKIDS in (-1 -2 -3) or qr43D1 in (-1 -2 -3) then do i=1 to 3;
 do m=lint to intDt;
   rec(i,m)=-3;
   dol(i,m)=-3;
 end;
end;
if qs43a in (-1 -2 -3) or qs43d1 in (-1 -2 -3) then do i=4 to 6;
 do m=lint to intDt;
   rec(i,m)=-3;
   dol(i,m)=-3;
 end;
end;

/* ================================================================
 * There was some recipiency, get details
 *
================================================================ */
do i=1 to 6;
if q43d(i)=0 then do m=lint to intDt;
  rec(i,m)=-4;
  dol(i,m)=-4;
end;
else if q43d(i)=1 then do;
/* ---------------------------------------------------------------------------
 * Calculate starts and ends of spells if possible
 * ---------------------------------------------------------------------------*/
* Continuous recipiency "ends" at interview date;
 do s = 1 to dim2(cont);
          if cont(i,s) = 1 or q63c(i,s)=13 then do;
       endYr(i,s) = intYr;
       endMo(i,s) = intMo;
    end;
 end;
* Compute starts/stops/last for basic cases (dates are known);
 do s = 1 to dim2(starts);
    if startMo(i,s) > 0 & startYr(i,s) > 0 then
       starts(i,s) = startYr(i,s)*12+startMo(i,s);
    if endMo(i,s) > 0 & endYr(i,s) > 0 then
       ends(i,s) = endYr(i,s)*12+endMo(i,s);
 end;
 seamFlag(i)=0;
 do s = 1 to dim2(starts);
          if starts(i,s) > 0 then do;
       if starts(i,s)=lint   then seamFlag(i)=1;
       if starts(i,s)=lint-1 then seamFlag(i)=2;
```

```
                if 0<starts(i,s)<lint-1 then seamFlag(i)=3;
        end;
 end;

lint_real=lint-936-1;
 do s=1 to dim2(starts);
  if starts(i,s) > 0 & starts(i,s)<lint then starts(i,s)=lint;
 end;
* Try to deduce starts/ends from # of mos;
 do s = 1 to dim2(starts);
   if nrOfRMos(i,s) > 0 then do;
            if ends(i,s) > 0 & starts(i,s) = .  then do;
        starts(i,s) = ends(i,s) - nrOfRMos(i,s)+1;
                        check(i)=5.1;
        if (starts(i,s) < lint) then check(i) = 5;
      end;
                else if starts(i,s) > 0 & ends(i,s) = .  then do;
        ends(i,s) = starts(i,s) + nrOfRMos(i,s) - 1;
                        check(i)=100+s+0.1;
                        if ends(i,s)>intDt then check(i)=100+s;
      end;
                else if starts(i,s) = .  & ends(i,s) = .  then do;
        if s = 1 then do; starts(i,s) = lint; check(i)=201+0.1; end;
                        else do; starts(i,s) = ends(i,s-1) + 1; check(i)=200+s+0.1; end;
        ends(i,s) = starts(i,s) + nrOfRMos(i,s) - 1;
                        if ends(i,s)>intDt then check(i)=200+s;
      end;
   end;
 end;

/* Determine what was the last spell */
 last(i) = 0;
 if q44b(i,1)=2 or q43B(i) = 2 then last(i) = 1;
 do s = 1 to dim2(starts);
   if ((startMo(i,s) > 0 | startYr(i,s) > 0 | endMo(i,s) > 0 | endYr(i,s) > 0 |nrOfRMos (i,s) > 0) &
last(i) < s) then last(i) = s;
 end;

/* ----------------------------------------------------------------------------
 * Edges of recipiency spells and periods between: n1 r1 n2 r2 ...  (NNNRRR???NRRR)
 * Under ideal conditions they touch, but often there is an unclear zone between them (-2)
 * ---------------------------------------------------------------------------- */
* The fully known cases;
 if last(i) ge 1 then do s = 1 to last(i);
   if starts(i,s) > 0 then do;
      ne(i,s) = starts(i,s)-1;
```

```
      rs(i,s) = starts(i,s);
   end;
   if ends(i,s) > 0 then do;
      re(i,s)   = ends(i,s);
      ns(i,s+1) = ends(i,s)+1;
   end;
 end;

* The unknown and partially known cases;
  if last(i) ge 1 then do s = 1 to last(i);
   if starts(i,s) = . & startYr(i,s) > 0 then do;
                  check(i)=301;
      xne = startYr(i,s)*12;
      if ( s = 1 & lint < =xne ) |
        ( s > 1 & 0 < ends(i,s-1) < xne) |
        ( s > 1 & 0 < endYr(i,s-1) + 1 < startYr(i,s) )
         then do; ne(i,s) = xne; check(i)=302; end;
      if startYr(i,s) < endYr(i,s) then do;
         rs(i,s) = startYr(i,s)*12+12;
                        check(i)=303;
                  end;
   end;
   if (ends(i,s) = . & endYr(i,s) > 0) then do;
                  check(i)=401;
      if 0 < startYr(i,s) < endYr(i,s) then do;
         re(i,s) = endYr(i,s)*12+1;
         ns(i,s+1) = endYr(i,s)*12+13;
                        check(i)=402;
                  end;
                  if 0 < startYr(i,s) = endYr(i,s) then do;
                    ns(i,s+1)=endYr(i,s)*12+13;
                    if rs(i,s)>0 then do;
                        re(i,s)=rs(i,s);
                    end;
                    check(i)=403;
                  end;
   end;
 end;
 if (ne(i,1) > 0) then ns(i,1) = lint;
 if (ns(i,last(i)+1) > 0) then ne(i,last(i)+1) = intDt;
 if last(i)>1 then do s = 2 to last(i);
         if 0 < rs(i,s) & rs(i,s) = re(i,s-1) then check(i) = 7;
 end;
 if rs(i,1) > 0 & rs(i,1)<lint then rs(i,1)=lint;

/* ---------------------------------------------------------------------------
```

```
 * Fill the recipiency status (rec) and amounts (dol).
 * ------------------------------------------------------------------------ */
* First, assume we know nothing;
 do m = lint to intDt;
   rec(i,m) = -2;
   dol(i,m) = -2;
 end;

* Fill in info we know (any of the periods below can be empty);
  if last(i) ge 1 then do s = 1 to last(i);
    * nonrecipiency period;
    if (0 < ns(i,s) <= ne(i,s)) then
       do m = max(ns(i,s),lint) to min(ne(i,s),intDt);
          rec(i,m) = -4;
          dol(i,m) = -4;
       end;
    * recipiency period;
    if (0 < rs(i,s) <= re(i,s)) then
       do m = max(rs(i,s),lint) to min(re(i,s),intDt);
          rec(i,m) = 1;
          dol(i,m) = howMuch(i,s);
                          if howmuch(i,s) in (. -4) then do; dol(i,m)=-2; chk5(i)=s+0.1; end;
       end;
    * we know only start - we are sure R received smth that month;
    else if (lint <= rs(i,s) <= intDt & re(i,s) = .) then do;
       rec(i, rs(i,s)) = 1;
       dol(i, rs(i,s)) = howMuch(i,s);
                  if howmuch(i,s) in (. -4) then do; dol(i, rs(i,s))=-2; chk5(i)=s+0.2; end;
    end;
    * we know only end - we are sure R received smth that month (can be even intDt if
continuous);
    else if (rs(i,s) = . & lint <= re(i,s) <= intDt) then do;
       rec(i,re(i,s)) = 1;
       dol(i,re(i,s)) = howMuch(i,s);
                  if howmuch(i,s) in (. -4) then do; dol(i, re(i,s))=-2; chk5(i)=s+0.3; end;
    end;
 end;

* The last period of nonrecipiency;
  if last(i) ge 1 then do;
    if (0 < ns(i, last(i)+1) <= ne(i, last(i)+1)) then
    do m = max(lint,ns(i, last(i)+1)) to min(ne(i,last(i)+1),intDt);
       rec(i,m) = -4;
       dol(i,m) = -4;
    end;
  end;
```

```
end;
end;
/* ================================================================ */
/* End if recip = .... (getting details for people who got something)
/* ================================================================ */

* fill in space at the end of year after the interview;
do i=1 to dim1(starts);
 do m=intDt+1 to hbound2(rec);
   rec(i,m)=-4;
   dol(i,m)=-4;
 end;
end;

endDt=intDt-936;
do i=1 to dim1(starts);
 if startMo(i,1) in (-1,-2) & startYr(i,1) in (-1,-2) then q43b(i) = 2;
end;

if noInt=0 then
   do i=1 to dim1(rec);
          do m=lbound2(rec) to hbound2(rec);
      if rec(i,m)=. then rec(i,m)=-4;
      if dol(i,m)=. then dol(i,m)=-4;
          end;
   end;

if noInt=1 then
   do i=1 to dim1(rec);
         do m=lbound2(rec) to hbound2(rec);
      if rec(i,m)=. then rec(i,m)=-5;
      if dol(i,m)=. then dol(i,m)=-5;
         end;
   end;

********** the respondent and spouse monthly amount;
***** respondent ssb (the sum of ssi and ssdi);
array rssb (937:&cEndMo) rssb1-rssb&cMos;
do m = lbound(rssb) to hbound(rssb);
  rssb(m)=-4;
end;
if qr43d3=1 then do m = lint to intdt;
 if dol(3,m)>-4 then do;
  dol(1,m)=-3;
  dol(2,m)=-3;
```

```
  rssb(m)=dol(3,m);
 end;
end;
else do m = lint to intDt;
 if dol(1,m)=-4 & dol(2,m)=-4 then rssb(m)=-4;
 else if dol(1,m) in (-1 -2 -3) or dol(2,m) in (-1 -2 -3) then rssb(m)=-3;
 else rssb(m)=max(dol(1,m),0) + max(dol(2,m),0);
end;
***** spouse ssb (both ssi and ssdi);
array sssb (937:&cEndMo) sssb1-sssb&cMos;
do m = lbound(sssb) to hbound(sssb);
   sssb(m)=-4;
end;
if qs43d3=1 then do m = lint to intdt;
 if dol(6,m)>-4 then do;
  dol(4,m)=-3;
  dol(5,m)=-3;
  sssb(m)=dol(6,m);
 end;
end;
else do m = lint to intDt;
 if dol(4,m)=-4 & dol(5,m)=-4 then sssb(m)=-4;
 else if dol(4,m) in (-1 -2 -3) or dol(5,m) in (-1 -2 -3) then sssb(m)=-3;
 else sssb(m)=max(dol(4,m),0) + max(dol(5,m),0);
end;
********** add up the respondent and the spouse monthly amount to come up with ssi, ssd and
ssb;
***** ssi;
array ssi (937:&cEndMo) ssi1-ssi&cMos;
do m = lbound(ssi) to hbound(ssi);
   ssi(m)=-4;
end;
do m = lint to intDt;
 if dol(1,m)=-4 & dol(4,m)=-4 then ssi(m)=-4;
 else if dol(1,m) in (-1 -2 -3) or dol(4,m) in (-1 -2 -3) then ssi(m)=-3;
 else ssi(m)=max(dol(1,m),0) + max(dol(4,m),0);
end;
***** ssd;
array ssd (937:&cEndMo) ssd1-ssd&cMos;
do m = lbound(ssd) to hbound(ssd);
   ssd(m)=-4;
end;
do m = lint to intDt;
 if dol(2,m)=-4 & dol(5,m)=-4 then ssd(m)=-4;
 else if dol(2,m) in (-1 -2 -3) or dol(5,m) in (-1 -2 -3) then ssd(m)=-3;
 else ssd(m)=max(dol(2,m),0) + max(dol(5,m),0);
```

```
end;
***** ssb;
array ssb (937:&cEndMo) ssb1-ssb&cMos;
do m = lbound(ssb) to hbound(ssb);
   ssb(m)=-4;
end;
do m = lint to intDt;
 if rssb(m)=-4 & sssb(m)=-4 then ssb(m)=-4;
 else if rssb(m) in (-1 -2 -3) or sssb(m) in (-1 -2 -3) then ssb(m)=-3;
 else ssb(m)=max(rssb(m),0) + max(sssb(m),0);
end;

********** ssd & ssb array starting in Jan 2017;
if lintyr=117 then do;
 *to fill the months in 2017 but before lint;
 if qr43d2 in (1 -1 -2 -3) or qr43d3=1 or qs43d2 in (1 -1 -2 -3) or qs43d3=1 then do;
  if lint>469 then do m=(469+936) to (lint-1);
   if ssd(m) in (-4 .) then ssd(m)=99999;
   if ssb(m) in (-4 .) then ssb(m)=99999;
  end;
 end;
 else do;
  if lint>469 then do m=(469+936) to (lint-1);
   if ssd(m) in (-4 .) then ssd(m)=99998;
   if ssb(m) in (-4 .) then ssb(m)=99998;
  end;
 end;
 *when we calculate the annual amount, we equate 99999 to -3 and 99998 to -4;

 *for those who indicate in the comments the SSDI start date is Jan 2017 or before;
 if  ssd469=99999 and (qr43d2=1 or qs43d2=1) and (cases found in the comments) then do;
  do m=(469+936) to &cEndMo;
   if -4<ssd(m)<99998 then do; ssd_1=ssd(m); ssd_1_m=m; m=&cEndMo;end;
  end;
  do m=(469+936) to &cEndMo;
   if ssd(m)=99999 then do; ssd(m)=ssd_1; ssb(m)=ssd_1; end;
   else if ssd(m) ne 99999 then m=&cEndMo;
  end;
 end;

end;

* for some who reported Jan 2017 as the start month;
if  ssd469=99999 and ((qr43d2=1 & qr44c21m=1 & qr44c21y=2017) or
                                          (qs43d2=1 & qs44c21m=1 & qs44c21y=2017)) then do;
 flag=1;
```

```
 do m=(469+936) to &cEndMo;
  if -4<ssd(m)<99998 then do; ssd_1a=ssd(m);ssd_1a_m=m; m=&cEndMo;end;
 end;
 do m=(469+936) to &cEndMo;
  if ssd(m)=99999 then do; ssd(m)=ssd_1a; ssb(m)=ssd_1a; end;
  else if ssd(m) ne 99999 then m=&cEndMo;
 end;
end;

********** create the current round yearly ssi, ssd & ssb;
array ssiYM(&cYrs,12) ssi1-ssi&cMos;
array ssdYM(&cYrs,12) ssd1-ssd&cMos;
array ssbYM(&cYrs,12) ssb1-ssb&cMos;

array yssi(*)   yssi78-yssi&cEndYr;
array yssd1(*)   yssd78-yssd&cEndYr;
array yssb1(*)   yssb78-yssb&cEndYr;

%macro mty (vym, vy);
do y=1 to dim(&vy);
   m4=0; m3=0; mp=0;
   do m=1 to 12;
      if &vym(y,m) in (-4 . 99998) then m4=m4+1;
            else if &vym(y,m) in (-1 -2 -3 99999) then m3=m3+1;
            else if &vym(y,m)>=0 then mp=mp+1;
   end;
   if m3>0 then &vy(y)=-3;
   else if m4=12 then &vy(y)=-4;
   else do;
     &vy(y)=0;
           do m=1 to 12;
             if 0<&vym(y,m)<99998 then &vy(y)=&vy(y)+&vym(y,m);
           end;
   end;
end;
%mend mty;

%mty(ssiYM, yssi);
%mty(ssdYM, yssd1);
%mty(ssbYM, yssb1);

********** connect the current round yearly variables to the old yearly variables;

array yssio(*)   syDolo78-syDolo&cEndYr;
array yssdo(*)   dyDolo117-dyDolo&cEndYr;
array yssbo(*)   r29byDolo117-r29byDolo&cEndYr;
```

```
array yssd(*)   yssd117-yssd&cEndYr;
array yssb(*)   yssb117-yssb&cEndYr;
yiflag=0;
do Y=1 to dim(yssio);
 if lintyr=y+77 & 0<lintmo<13 then do;
  if yssi(y)=-4 then do; yssi(y)=yssio(y); yiflag=1; end;
  else if yssi(y)>=0 & yssio(y) not in (-4 0) then do;
    if yssio(y)>0 then do; yssi(y)=yssi(y)+yssio(y); yiflag=2; end;
          else if yssio(y) in (-1 -2 -3) then do; yssi(y)=yssio(y); yiflag=3; end;
  end;
 end;
end;

ydflag=0;
do Y=1 to dim(yssdo);
 if lintyr=y+116 & 0<lintmo<13 then do;
  if yssd(y)=-4 then do; yssd(y)=yssdo(y); ydflag=1; end;
  else if yssd(y)>=0 & yssdo(y) not in (-4 0) then do;
    if yssdo(y)>0 then do; yssd(y)=yssd(y)+yssdo(y); ydflag=2; end;
          else if yssdo(y) in (-1 -2 -3) then do; yssd(y)=yssdo(y); ydflag=3; end;
  end;
 end;
end;

ybflag=0;
do Y=1 to dim(yssbo);
 if lintyr=y+116 & 0<lintmo<13 then do;
  if yssb(y)=-4 then do; yssb(y)=yssbo(y); ybflag=1; end;
  else if yssb(y)>=0 & yssbo(y) not in (-4 0) then do;
    if yssbo(y)>0 then do; yssb(y)=yssb(y)+yssbo(y); ybflag=2; end;
          else if yssbo(y) in (-1 -2 -3) then do; yssb(y)=yssbo(y); ybflag=3; end;
  end;
 end;
end;
```

## Summary Welfare (AFDC/TANF, Food Stamps/SNAP, SSI/SSDI) Receipt

```
%let cEndYr = 121;              *(2021-1900);
%let cEndMo = 1464;     *(121+1)*12;
%let cYrs = 44;                         *(2021-1978)+1;
%let cMos = 528;                *44*12;
%let cFillFlag = 2020;

***** variables created in the current round;
```

```
data afdc;
infile aIn dlm=',' lrecl=500;
input norcid ayDol106-ayDol&cEndYr;
aupdate=1;

data stamps;
infile fIn dlm=',' lrecl=500;
input norcid fyDol106-fyDol&cEndYr;
fupdate=1;

data ssi;
infile siIn dlm=',' lrecl=500;
input norcid siyDol89-siyDol&cEndYr;
siupdate=1;

data ssb;
infile sbIn dlm=',' lrecl=50;
input norcid sbyDol117-sbyDol&cEndYr;
sbupdate=1;

********************************** create afdc + food stamps + ssi as in the past;

data one;
 merge R29.lintdate2020 afdc stamps ssi
       prg2020.oldin2020 (keep=norcid aydolo: fydolo: sydolo:);
 by norcid;
 if aupdate=1 or fupdate=1 or siupdate=1;

array sY(*) siYDol89-siYDol&cEndYr;
array aY(*) aYDol89-aYDol&cEndYr;
array fY(*) fYDol89-fYDol&cEndYr;
array sYo(*) sYDolo89-sYDolo&cEndYr;
array aYo(*) aYDolo89-aYDolo&cEndYr;
array fYo(*) fYDolo89-fYDolo&cEndYr;

array wY(*) wiYDol89-wiYDol&cEndYr;

do i=1 to dim(ay);
 if ay(i)=. then ay(i)=-4;
end;
do i=1 to dim(sy);
 if sy(i)=. then sy(i)=-4;
end;
do i=1 to dim(fy);
 if fy(i)=. then fy(i)=-4;
end;
```

```
 ayflag=0; syflag=0; fyflag=0;
do i=1 to dim(ay);
 if lintyr=i+88 & 0<lintmo<13 then do;
   if ay(i)=-4 & ayo(i) > -4 then do; ay(i)=ayo(i); ayflag=1; end;
   if sy(i)=-4 & syo(i) > -4 then do; sy(i)=syo(i); syflag=1; end;
   if fy(i)=-4 & fyo(i) > -4 then do; fy(i)=fyo(i); fyflag=1; end;
 end;
end;

do i=1 to dim(wy);
 if ay(i)=-4 & sy(i)=-4 & fy(i)=-4 then wy(i)=-4;
 else if ay(i) in (-1 -2 -3) or sy(i) in (-1 -2 -3) or fy(i) in (-1 -2 -3) then wy(i)=-3;
 else wy(i)=max(ay(i), 0)+max(sy(i), 0)+max(fy(i), 0);
end;

data null;
set one;
file outiy lrecl=300;
put @1 norcid (wiyDol89-wiyDol&cEndYr)(+(-1)',');
run;

************************** create afdc + food stamps + ssi + ssdi starting in 2017;
data two;
 merge R29.lintdate2020 afdc stamps ssb
       prg2020.oldin2020 (keep=norcid aydolo: fydolo: r29bydolo:);
 by norcid;
 if aupdate=1 or fupdate=1 or sbupdate=1;

array sY(*) sbYDol117-sbYDol&cEndYr;
array aY(*) aYDol117-aYDol&cEndYr;
array fY(*) fYDol117-fYDol&cEndYr;
array sYo(*) r29bydolo117-r29bydolo&cEndYr;
array aYo(*) aYDolo117-aYDolo&cEndYr;
array fYo(*) fYDolo117-fYDolo&cEndYr;

array wY(*) wbYDol117-wbYDol&cEndYr;

do i=1 to dim(ay);
 if ay(i)=. then ay(i)=-4;
end;
do i=1 to dim(sY);
 if sY(i)=. then sY(i)=-4;
end;
do i=1 to dim(fy);
 if fy(i)=. then fy(i)=-4;
```

```
end;

 ayflag=0; syflag=0; fyflag=0;
do i=1 to dim(ay);
 if lintyr=i+116 & 0<lintmo<13 then do;
   if ay(i)=-4 & ayo(i) > -4 then do; ay(i)=ayo(i); ayflag=1; end;
   if sy(i)=-4 & syo(i) > -4 then do; sy(i)=syo(i); syflag=1; end;
   if fy(i)=-4 & fyo(i) > -4 then do; fy(i)=fyo(i); fyflag=1; end;
 end;
end;

do i=1 to dim(wy);
 if ay(i)=-4 & sy(i)=-4 & fy(i)=-4 then wy(i)=-4;
 else if ay(i) in (-1 -2 -3) or sy(i) in (-1 -2 -3) or fy(i) in (-1 -2 -3) then wy(i)=-3;
 else wy(i)=max(ay(i), 0)+max(sy(i), 0)+max(fy(i), 0);
end;
```