

APPENDIX 42: THE NLSW01

This appendix to the National Longitudinal Survey of Women for 2001 (NLSW01) describes aspects of the data that are specific to 2001 and provides the SAS code used to edit or create variables for this release. There are 5,124 respondents with completed interviews in 2001: 2,318 are from the Mature Women cohort and 2,806 are from the Young Women cohort. Most surveys were conducted by personal interview; however respondents can request a phone interview and many chose this option (39%). The NLSW01 is comprised of over 9,300 variables.

One section for 2001, Parents and Transfers, was last included in 1997 and looks at transfers of time and money to the respondent/husband by their parents and to the parents by the respondent/husband. The content of transfer questions is similar to the 1999 child transfers and includes the purpose of the transfer (loan or gift) and the amount of money or help given or received. Transfer questions are asked separately for "in the past 12 months have you given...?" and for "have you ever given...?" If the respondent's father or mother is deceased and the estate was not settled as of last interview, information is asked about the status and resolution of the estate. Other Specify responses to PAR-56 and PAR-66 are included at the end of this appendix.

The cumulative created variables first released 1999—(1) labor force status (working/not working) by calendar week from 1994 through date of interview in 1999 and (2) ordered marriage start and end dates from 1967/8 through date of interview in 1999—have been updated with information collected in the 2001 interview.

One problem for 2001 is that some employers were mistakenly identified as self-employers and deleted when the respondent indicated that she was not self-employed by that employer. Because CHRR does not receive any data for employers that are deleted from the roster, the Census Bureau has flagged these jobs so that we can ask about them at the time of the 2003 interview. If the employer was deleted in error and not added on a new line of the employer roster, weeks worked since last interview will be underestimated for these respondents and labor force status event history will be incorrect for the period after 1999, until corrections are incorporated in 2003.

The ID numbers (variable R00001.) for respondents with a flagged job for which data will be recovered in 2003 are:

Mature Women

6	39	180	304	433	500	589	684	838	847	910	1052	1062	1076	1201
1307	1363	1401	1512	1525	1592	1631	1784	1892	1898	1976	1995	1997	2014	2029
2059	2214	2327	2341	2369	2555	2572	2595	2993	3017	3061	3594	3650	3824	3973
3977	4415	4427	4586	4633	4670	4712	4846	4894	5012					

Young Women

5	15	44	67	70	93	126	128	141	159	176	187	218	219	224
231	285	287	292	298	300	302	319	352	367	376	396	402	409	431
437	443	457	460	469	486	504	517	534	539	551	555	570	585	588
604	618	670	689	690	716	724	731	754	785	830	859	894	897	908
916	934	940	943	965	966	970	977	984	995	1008	1009	1010	1016	1018
1023	1028	1038	1045	1066	1069	1072	1075	1086	1118	1198	1217	1218	1219	1224
1248	1262	1271	1288	1322	1345	1366	1418	1425	1439	1469	1471	1491	1492	1498
1499	1500	1519	1540	1559	1561	1563	1583	1590	1604	1615	1624	1639	1640	1647
1659	1668	1682	1688	1691	1696	1711	1714	1729	1734	1759	1770	1777	1784	1805
1807	1817	1829	1851	1866	1878	1905	1915	1922	1959	1962	1974	1981	1997	2005
2015	2032	2039	2045	2052	2089	2125	2140	2145	2163	2168	2197	2204	2219	2232
2235	2237	2285	2290	2292	2304	2309	2319	2326	2334	2344	2357	2363	2448	2453
2486	2545	2554	2559	2571	2579	2596	2608	2617	2671	2681	2752	2763	2791	2816
2825	2826	2861	2878	2884	2894	2896	2988	3000	3008	3015	3026	3039	3056	3063
3070	3082	3088	3092	3104	3107	3116	3124	3135	3137	3138	3201	3222	3227	3228
3234	3235	3258	3305	3323	3366	3395	3399	3408	3412	3414	3416	3418	3448	3455
3464	3466	3470	3488	3530	3545	3634	3640	3648	3688	3692	3704	3716	3719	3728
3750	3779	3791	3792	3799	3809	3812	3865	3880	3897	3899	3918	3956	3978	4024

Appendix 42: The NLSW01

4041	4073	4083	4090	4091	4094	4127	4145	4171	4172	4180	4191	4192	4201	4205
4215	4221	4247	4287	4298	4344	4369	4382	4387	4452	4480	4505	4510	4516	4522
4533	4540	4544	4548	4551	4553	4565	4566	4602	4629	4632	4642	4655	4690	4695
4710	4711	4728	4736	4767	4769	4824	4837	4858	4898	4920	4953	4958	4961	4977
4978	4996	5013	5017	5019	5039	5044	5069	5071	5085	5087	5109	5156		

SAS CODE FOR TRADITIONAL CREATED VARIABLES

/* CREATE REASON FOR NONINTERVIEW FOR ALL RESPONDENTS */

```
if R4267000>0 then RNI=R4267000;          if R6327600=2 then rni=10;
if R6327600=5 then rni=14;                if 201 le R6320200 le 205 then RNI=0;
if R6320200=213 or R6320200=219 then RNI=11; if R6320200=214 then RNI=5;
if R6320200=216 then RNI=1;              if R6320200=217 then RNI=6;
if R6320200=218 then RNI=9;              if R6320200=223 then RNI=7;
if R6320200=234 then RNI=8;              if R6320200=250 then RNI=10;
if R6320200=251 then RNI=13;             if 261 le R6320200 le 271 then RNI=2;
if R6320200=260 then RNI=4;
```

/* CREATE OR EDIT AS FOLLOWS FOR COMPLETED INTERVIEWS*/

/* CREATE THE MONTHLY LABOR RECODE FOR THE RESPONDENT*/

```
if (R5506300 ge 0 & R5506500 ge 0) then HRUSLT=R5506300+R5506500;
else if (R5506300 ge 0 and R5506500 LT 0) then HRUSLT=R5506300;
else if (R5506300 LT 0 & R5506500 ge 0) then HRUSLT=R5506500;
if (R5507800 ge 0 & R5508200 ge 0) then HRACCTT=R5507800+R5508200;
else if (R5507800 ge 0 & R5508200 LT 0) then HRACCTT=R5507800;
else if (R5507800 LT 0 & R5508200 ge 0) then HRACCTT=R5508200;
if (HRACCTT=0 & (R5503800=-1 | R5503800=-2 | R5503800=0)) then HRCK6=1;
else if HRACCTT=0 then HRCK6=2;
else HRCK6=3;
if ((R5503800=0 | R5503800=-1 | R5503800=-2) & ((0 le R5507800 LT 15) | R5507800=-2)) then HRCK7=1;
else if (R5503800=0 | R5503800=-1 | R5503800=-2) & R5507800 ge 15 then HRCK7=2;
else if (HRUSLT ge 35 | R5506700=1) & (HRACCTT LT 35) & ((R5507800 ge 0) | (R5508200 ge 0)) then
    HRCK7=3;
else if (R5507000=1 & HRACCTT LT 35) & (1 le R5507100 le 3) then HRCK7=4;
else HRCK7=5;

if (R5503500=1 & HRCK6=3) | (R5503700=1 & HRCK7 ge 2) then MLR=1;
else if (R5505900=1 | R5505900=-1 | R5505900=-2 | R5505900=0) then MLR=2;
else if (R5509000=1 | R5509000=-1 | R5509000=-2 | R5509100=1) then MLR=3;
else if (R5513500=1 | R5513500=-1 | R5513500=-2 | R5513600=1 | R5513600=2) then MLR=4;

else if ((R5503500=3 & R5445400 ge 50) & R5504400=1 & R5509900=1 & R5513500=0
    & (R5513600=3 | R5513600=4 | R5513600=-2 | R5513600=-1)) then MLR=5;
else if ((R5504400=0 | R5516900=1 | (R5509900=3 & R5445400 ge 50))
    | (R5515000=3 & R5445400 ge 50) | R5517000=5
    | (R5445400 ge 50 & (R5531200=4 | R5531300=4 | R5531400=4 | R5531500=4
    | R5531600=4 | R5531700=4 | R5531800=4))) then MLR=5;
else if R5503500=3 & R5445400 ge 50 & R5504000=1 then MLR=5;
else if (R5504500=1 | R5504600=1 | R5504100=1 | R5517000=1) then MLR=6;
else MLR=7;
```

Appendix 42: The NLSW01

/* CREATE THE MONTHLY LABOR RECODE FOR THE HUSBAND/PARTNER*/

```
if 1 le R5490800 le 4 then do;
  if (R5892700 ge 0 & R5892900 ge 0) then HRUSLTH=R5892700+R5892900;
  else if (R5892700 ge 0 and R5892900 LT 0) then HRUSLTH=R5892700;
  else if (R5892700 LT 0 & R5892900 ge 0) then HRUSLTH=R5892900;
  if (R5894100 ge 0 & R5894500 ge 0) then HRACTTH=R5894100+R5894500;
  else if (R5894100 ge 0 & R5894500 LT 0) then HRACTTH=R5894100;
  else if (R5894100 LT 0 & R5894500 ge 0) then HRACTTH=R5894500;
  if ((HRACTTH=0) & (R5890400=-1 | R5890400=-2 | R5890400=0)) then HRCK6H=1;
  else if HRACTTH=0 then HRCK6H=2;
  else HRCK6H=3;
  if ((R5890400=0 | R5890400=-1 | R5890400=-2) & (0 le R5894100 LT 15 | R5894100=-2)) then HRCK7H=1;
  else if ((R5890400=0 | R5890400=-1 | R5890400=-2) & R5894100 ge 15) then HRCK7H=2;
  else if ((HRUSLTH ge 35 | R5893100=1) & (HRACTTH LT 35) & (R5894100 ge 0 | R5894500 ge 0)) then
    HRCK7H=3;
  else if ((R5893400=1 & HRACTTH LT 35) & (1 le R5893500 le 3 )) then HRCK7H=4;
  else HRCK7H=5;

  if (R5890100=1 & HRCK6H=3) | (R5890300=1 & HRCK7H ge 2) then HMLR=1;
  else if (R5892300=1 | R5892300=-1 | R5892300=-2 | R5892300=0) then HMLR=2;
  else if R5895300=1 | R5895300=-1 | R5895300=-2 then HMLR=3;
  else if R5897800=1 | R5897800=-1 | R5897800=-2 | R5897900=1 | R5897900=2 then HMLR=4;
  else if ((R5890100=3 & R5889900 ge 50) & R5890800=1 & R5896100=1 & R5897800=2
    & (R5897900=3 | R5897900=4 | R5897900=-2 | R5897900=-1)) then HMLR=5;
  else if (R5890800=0 | R5901200=1 | (R5896100=3 & R5889900 ge 50) | (R5899300=3 & R5889900 ge 50) |
    R5901300=5 | (R5889900 ge 50 & (R5915900=4 | R5916000=4 | R5916100=4 | R5916200=4 )) ) then
    HMLR=5;
  else if (R5890900=1 | R5891000=1 | R5901300=1) then HMLR=6;
  else HMLR=7;
end;
```

/* TAKE MOST RECENT HIGHEST GRADE COMPLETED AND UPDATE WITH DATA FROM CURRENT SURVEY*/

```
HGC=R5141500;
if R5440400=1997 then HGC=R4192800; if R5440400=1995 then HGC=R3476600;
if R5440400=1993 then HGC=R1520410; if R5440400=1991 then HGC=R1346410;
if R5440400=1988 then HGC=R1215110; if R5440400=1987 then HGC=R1097410;
if R5440400=1985 then HGC=R1051610; if R5440400=1983 then HGC=R0929510;
if R5440400=1982 then HGC=R0797110; if R5440400=1980 then HGC=R0749910;
if R5440400=1992 then HGC=R0989700; if R5440400=1989 then HGC=R0989700;
if 10001 le SERIAL le 19999 & HGC<0 then HGC=R0381500;
if 10001 le SERIAL le 19999 & HGC<0 then HGC=R0079000;
```

```
if 1 le R6236100 le 10 then HGC01=R6236100+7;
if R6236100=11 then HGC01=95;
if 1 le R6236300 le 3 then HGC01=12;
if 3 le R6236600 le 4 then HGC01=18;
if R6236600=2 then HGC01=16;
if R6236600=1 then HGC01=14;
if HGC01>HGC then HGC=HGC01;
```

Appendix 42: The NLSW01

/* CREATE TOTAL NET FAMILY ASSETS. SET TO MISSING if ANY AMOUNT IS MISSING*/

```
C=0;
if R6197700=0&(R6197800<0 | R6197800=.)&(R6197900<0 | R6197900=.) then do; HOUSE=0; PROPDE=0;
    end;
if R6197900>=0 & R6198400>=0 then PROPDE = R6197900 + R6198400;
if R6197900<0 & R6198400>=0 then PROPDE = R6198400;
if R6197900>=0 & R6198400<0 then PROPDE = R6197900;
if R6197800>=0 & PROPDE>=0 then HOUSE=R6197800-PROPDE;
if R6201900=1 & 0 le R6197800 le R6201800 & 0 le R6197900 le R6202000 then HOUSE=0;
if HOUSE^=. then ASSET=HOUSE; else C=C+1;

if (R6198700=0 | R6212500=0) & (R6198800<0 | R6198800=.) then SAVE=0;
else if R6198800>=0 then SAVE=R6198800;
if -2 le R6198800 le -1 then SAVE=.;
if SAVE^=. then ASSET=ASSET+SAVE; else C=C+1;

if (R6198900=0 | R6212500=0) & (R6199000<0 | R6199000=.) then BOND=0;
else if R6199000>=0 then BOND=R6199000;
if -2 le R6199000 le -1 then BOND=.;
if BOND^=. then ASSET=ASSET+BOND; else C=C+1;

if (R6199100=0 | R6212500=0) & (R6199200<0 | R6199200=.) then STOCK=0;
else if R6199200>=0 then STOCK=R6199200;
if R6199300=1 & R6199400=1 then STOCK=40000;
if R6199300=1 & R6199400=0 then STOCK=27500;
if R6199300=0 & R6199500=1 then STOCK=10000;
if R6199300=0 & R6199500=0 then STOCK=5000;
if R6199300=-1 | R6199400=-1 | R6199500=-1 | R6199300=-2 | R6199400=-2 | R6199500=-2 then STOCK=.;
if STOCK^=. then ASSET=ASSET+STOCK; else C=C+1;

if R6199600=0 & (R6199700<0 | R6199700=.) then IRA=0;
else if R6199700 ge 0 then IRA=R6199700;
if R6199800=1 & R6199900=1 then IRA=30000;
if R6199800=1 & R6199900=0 then IRA=22500;
if R6199800=0 & R6200000=1 then IRA=10000;
if R6199800=0 & R6200000=0 then IRA=5000;
if R6199800=-1 | R6199900=-1 | R6200000=-1 | R6199800=-2 | R6199900=-2 | R6200000=-2 then IRA=.;
if IRA^=. then ASSET=ASSET+IRA; else C=C+1;

if R6200100=0 & (R6200200<0 | R6200200=.) then LOAN=0;
else if R6200200>=0 then LOAN=R6200200;
if LOAN^=. then ASSET=ASSET+LOAN; else C=C+1;

if (R6200300=0 | R6200500=1) & (R6200600<0 | R6200600=.) then INSU=0;
else if R6200600 ge 0 then INSU=R6200600;
if INSU^=. then ASSET=ASSET+INSU; else C=C+1;

if R6200900=0 & (R6201000<0 | R6201000=.) then TRUS=0;
else if R6201000 >= 0 then TRUS = R6201000;
if TRUS^=. then ASSET=ASSET+TRUS; else C=C+1;

if R6201400=0 & (R6201500<0 | R6201500=.) then SETTLE=0;
else if R6201400=1 & R6201500<0 then SETTLE=0;
else if R6201500>=0 then SETTLE=R6201500;
if SETTLE^=. then ASSET=ASSET+SETTLE; else C=C+1;
```

Appendix 42: The NLSW01

```
if (R6201700=0 | R6211900=0) & (R6201800<0 | R6201800=.) & (R6202000<0 | R6202000=.) then IFARM=0;
else if R6201800>=0 & R6202000>=0 then IFARM=R6201800-R6202000;
if IFARM^=. then ASSET=ASSET+IFARM; else C=C+1;
```

```
if (R6202100=0 |(R6204400=0 & R6208300=0)) & (R6202200<0 | R6202200=.)
  & (R6202300<0 | R6202300=.) then IBUS=0;
else if R6202200>=0 & R6202300>=0 then IBUS=R6202200-R6202300;
if IBUS^=. then ASSET=ASSET+IBUS; else C=C+1;
```

```
if (R6202400=0 | R6212200=0) & (R6202500<0 | R6202500=.)
  & (R6202600<0 | R6202600=.) & (R6202700<0 | R6202700=.) then IREAL=0;
else if R6202500>=0 & R6202600>=0 & R6202700>=0 then IREAL=R6202500-R6202600-R6202700;
if IREAL^=. then ASSET=ASSET+IREAL; else C=C+1;
```

```
if R6202800=0 & (R6202900<0 | R6202900=.) then DEBT=0;
else if R6202900>=0 then DEBT=R6202900;
if DEBT^=. then ASSET=ASSET-DEBT; else C=C+1;
```

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

/* CREATE SUMMATION and NET FAMILY INCOME */

```
/* Sum all income amounts to create summation. If all are missing then summation is missing*/
/* Set net family income equal summation. If any amount is missing, net family income is missing*/
NI=0; CC=0;
```

```
array ALLINC  RWAGE  RBUS  RUNEM  RSUB  RSOC  RVET  RCOM  RSSD
               RDIS  REPRI  REMIL  REFED  RESTE  REUNI  REIRA  REOTH
               SWAGE  SBUS  SUNEM  SSUB  SSOC  SVET  SCOM  SSSD
               SDIS  SREPRI  SREMIL  SREFED  SRESTE  SREUNI  SREIRA  SREOTH
               FARM  RENT  INTT  FOODS  AFDC  SSI  ALIR  ALIP
               OTHER;
array ANY      R6204200 R6204400 R6204700 R6205000 R6205300 R6205700 R6205900 R6206100
               R6206300 R6206600 R6206800 R6207000 R6207200 R6207400 R6207600 R6207800
               R6208100 R6208300 R6208600 R6208900 R6209200 R6209600 R6209800 R6210000
               R6210200 R6210500 R6210700 R6210900 R6211100 R6211300 R6211500 R6211700
               R6211900 R6212200 R6212500 R6212700 R6213000 R6213300 R6213600 R6213800
               R6219400;
array AMT      R6204300 R6204500 AMT03  AMT04  AMT05  R6205800 R6206000 R6206200
               R6206400 R6206700 R6206900 R6207100 R6207300 R6207500 R6207700 R6207900
               R6208200 R6208400 AMT19  AMT20  AMT21  R6209700 R6209900 R6210100
               R6210300 R6210600 R6210800 R6211000 R6211200 R6211400 R6211600 R6211800
               R6212000 R6212300 R6212600 AMT36  AMT37  AMT38  R6213700 AMT40
               R6219500;
array LOSS     LOSS01 R6204600 LOSS03- LOSS10
               LOSS11-LOSS17 R6208500 LOSS19
               LOSS20 LOSS21-LOSS30
               LOSS31 LOSS32 R6212100 R6212400 LOSS35-LOSS39 R6213900 LOSS41;
array PER      PER01 PER02 R6204800 R6205100 R6205400 PER06-PER10 PER11-PER18 R6208700
               R6209000 R6209300 PER22-PER30 PER31-PER35 R6212800 R6213100 R6213400 PER39
               PER40 PER41;
```

Appendix 42: The NLSW01

```
array UNIT      UNIT01 UNIT02 R6204900 R6205200 R6205500 UNIT06-UNIT10 UNIT11-UNIT18 R6208800
                R6209100 R6209400 UNIT22-UNIT30 UNIT31-UNIT35 R6212900
                R6213200 R6213500 UNIT39 UNIT40 UNIT41;
```

```
if R6206500=0 then do;
  REPRI=0; REMIL=0; REFED=0; RESTE=0; REUNI=0; REIRA=0; REOTH=0;
  end;
if R6208000 NE 1 then do;
  SWAGE=0; SBUS=0; SUNEM=0; SSUB=0; SSOC=0; SVET=0; SCOM=0; SSSD=0; SDIS=0;
  end;
if R6208000 NE 1 or R6210400=0 then do;
  SREPRI=0; SREMIL=0; SREFED=0; SRESTE=0; SREUNI=0; SREIRA=0; SREOTH=0;
  end;
if R6213800=1 then R6213800=3;
do over ALLINC;
if PER ge 0 and UNIT ge 0 then AMT=PER*UNIT;
if (ANY=0 or ANY=4) and (AMT<0 or AMT=.) then ALLINC=0;
if ANY=1 and AMT ge 0 then ALLINC=AMT;
if ANY=3 and LOSS ge 0 then ALLINC=0-LOSS;
if ALLINC=. then CC=CC+1;
```

```
if ALLINC NE . then do; NI=NI+1;
if NI=1 then FAMINC=ALLINC;
if NI>1 then FAMINC=FAMINC+ALLINC; end; end;
```

```
if R6214000=0 and R6214100=0 then CHDSUP=0;
if R6215100=2 & R6217200>=0 then CHDSUP=R6217200;
if R6215100=1 & R6218000>=0 then CHDSUP=R6218000;
if R6215100=3 & R6217200>=0 & R6218000>=0 then CHDSUP=R6217200+R6218000;
if R6214200>=0 then CHDSUP=R6214200;
```

```
if R6218400=0 then RCHD=0;
if R6218500>=0 then RCHD=52*R6218500;
if R6218600>=0 then RCHD=12*R6218600;
if R6218700>=0 then RCHD=R6218700;
```

```
if (R6208000 NE 1 | R6219000=0) then HPCHD=0;
else if R6219100>=0 then HPCHD=52*R6219100;
else if R6219200>=0 then HPCHD=12*R6219200;
else if R6219300>=0 then HPCHD=R6219300;
```

```
if CHDSUP>=0 & RCHD>=0 & HPCHD>=0 then CHILD=CHDSUP-RCHD-HPCHD;
if CHILD=. then CC=CC+1;
if CHILD NE . then do; NI=NI+1; if NI=1 then FAMINC=CHILD;
if NI>1 then FAMINC=FAMINC+CHILD; end;
```

```
if R6219700 NE 2 then FAM=0;
if R6219800=1 then FAM= 2000;      else if R6219800=2 then FAM= 5000;
else if R6219800=3 then FAM= 6750; else if R6219800=4 then FAM= 8250;
else if R6219800=5 then FAM= 12500; else if R6219800=6 then FAM= 16250;
else if R6219800=7 then FAM= 18750; else if R6219800=8 then FAM= 22500;
else if R6219800=9 then FAM= 30000; else if R6219800=10 then FAM= 42500;
else if R6219800=11 then FAM= 62500; else if R6219800=12 then FAM= 87500;
else if R6219800=13 then FAM=100000; else if R6219800=14 then FAM=0;
```

```
if FAM NE . then do; NI=NI+1; if NI=1 then FAMINC=FAM;
```

Appendix 42: The NLSW01

```

if NI>1 then FAMINC=FAMINC+FAM; end;
if FAMINC NE . and FAMINC LT -99999 then FAMINC=-.99999;
if FAMINC>349999 then FAMINC=349999;
SUMMATN=FAMINC; if SUMMATN=. then SUMMATN=0;
if CC>0 then FAMINC=.;

```

/* CREATE HOURLY RATE OF PAY FOR EACH JOB ACTIVE SINCE DOLI FOR R and H/P*/

```

array HROP R5684500 R5684600 R5684700 R5684800 R5684900 R5685000 R5685100 R5685200
           R5685300 R5685400 R5685500 R5685600 R5685700 R6040700 R6040800 R6040900
           R6041000 R6041100 R6041200 R6041300 R6041400 R6041500 R6041600 HROP24
           HROP25 R6041700 HROP27 R6041800 HROP29 R6041900 HROP31;
array WROP R5707700 R5707800 R5707900 R5708000 R5708100 R5708200 R5708300 R5708400
           WROP9-WROP10 R5708500 WROP12- WROP13 R6060800 R6060900 R6061000
           R6061100 R6061200 R6061300 R6061400 R6061500 WROP22-WROP23 R6061600
           WROP25-WROP31;
array MROP R5711000 R5711100 R5711200 R5711300 R5711400 R5711500 R5711600 MROP08
           MROP09 R5711700 R5711800 MROP12- MROP13 R6063900 R6064000 R6064100
           R6064200 R6064300 MROP19-MROP22 R6064400 MROP24-MROP31;
array AROP R5714800 R5714900 R5715000 R5715100 R5715200 R5715300 R5715400 R5715500
           R5715600 R5715700 R5715800 R5715900 AROP13 R6067300 R6067400 R6067500
           R6067600 R6067700 R6067800 R6067900 R6068000 AROP22-AROP26 R6068100
           AROP28 R6068200 R6068300 AROP31;
array BROP R5719000 R5719100 R5719200 R5719300 R5719400 R5719500 R5719600 R5719700
           R5719800 BROP10-BROP13 R6071200 R6071300 R6071400 R6071500 R6071600
           R6071700 R6071800 BROP21 R6071900 BROP23-BROP31;
array OROP R5725800 R5725900 R5726000 R5726100 R5726200 R5726300 R5726400 R5726500
           R5726600 R5726700 R5726800 R5726900 OROP13 R6078600 R6078700 R6078800
           R6078900 R6079000 R6079100 R6079200 R6079300 R6079400 OROP23-OROP28
           R6079500 OROP30 OROP31;

array SWHRP SWHRP01-SWHRP13 HSWHRP01-HSWHRP18;
array USHD R5624900 R5625000 R5625100 R5625200 R5625300 R5625400 R5625500 R5625600
           R5625700 R5625800 R5625900 R5626000 R5626100 R6003900 R6004000 R6004100
           R6004200 R6004300 R6004400 R6004500 R6004600 R6004700 R6004800 R6004900
           R6005000 R6005100 R6005200 R6005300 R6005400 R6005500 R6005600;
array USHW R5631500 R5631600 R5631700 R5631800 R5631900 R5632000 R5632100 R5632200
           R5632300 R5632400 R5632500 R5632600 R5632700 R6011200 R6011300 R6011400
           R6011500 R6011600 R6011700 R6011800 R6011900 R6012000 R6012100 R6012200
           R6012300 R6012400 R6012500 R6012600 R6012700 R6012800 USHW31;
array TURP R5678700 R5678800 R5678900 R5679000 R5679100 R5679200 R5679300 R5679400
           R5679500 R5679600 R5679700 R5679800 R5679900 R6033500 R6033600 R6033700
           R6033800 R6033900 R6034000 R6034100 R6034200 R6034300 R6034400 R6034500
           TURP25 R6034600 R6034700 R6034800 R6034900 R6035000 R6035100;
array USWY R5721200 R5721300 R5721400 R5721500 R5721600 R5721700 R5721800 R5721900
           R5722000 R5722100 R5722200 R5722300 R5722400 R6073000 R6073100 R6073200
           R6073300 R6073400 R6073500 R6073600 R6073700 R6073800 R6073900 R6074000
           USWY25 R6074100 R6074200 R6074300 R6074400 R6074500 R6074600;

do over SWHRP;
if TURP=1 then SWHRP=HROP;
if USHW LT USHD then USHW=.;
if USHW>0 then do;
if (TURP=2 | TURP=3 | TURP=7) then SWHRP=(100*WROP)/USHW;
if TURP=4 then SWHRP=(100*BROP)/(USHW*2);
if (TURP=5 | TURP=8) then SWHRP=(100*MROP)/(USHW*4.33);
if (TURP=6 & USWY ge 1) then SWHRP=(100*AROP)/(USHW*USWY);

```

Appendix 42: The NLSW01

```
if OROP>0 then SWHRP=OROP;
SWHRP = FLOOR(SWHRP + .5);
end; end;
```

```
/* EDIT THE HUSBAND EMPLOYER SORT SECTION */
```

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

```
DOIDAT=MDY(R5440800,R5440900,R5441000);
DOLIDAT=MDY(R5440200,R5440300,R5440400);
```

```
HHIDAT=0;
```

```
array HSORTY   R5957900  R5958000  R5958100  R5958200  R5958300  R5958400  R5958500  R5958600
               R5958700  R5958800  R5958900  R5959000  R5959100  R5959200  R5959300  R5959400
               R5959500  R5959600;
array HSORTM   R5954300  R5954400  R5954500  R5954600  R5954700  R5954800  R5954900  R5955000
               R5955100  R5955200  R5955300  R5955400  R5955500  R5955600  R5955700  R5955800
               R5955900  R5956000;
array HSORTD   R5956100  R5956200  R5956300  R5956400  R5956500  R5956600  R5956700  R5956800
               R5956900  R5957000  R5957100  R5957200  R5957300  R5957400  R5957500  R5957600
               R5957700  R5957800;
array HSORT    HSRTDA01-HSRTDA18;
array HENDT    HNNT01-HNNT18;
```

```
do over HSORT;
```

```
  if HSORTM=2 and HSORTD>28 then HSORTD=28;
  if (HSORTM=4 or HSORTM=6 or HSORTM=9 or HSORTM=11) and HSORTD=31 then HSORTD=30;
  HSORT=HSORTY*10000+HSORTM*100+HSORTD;
  HENDT=MDY(Hsortm,Hsortd,Hsorthy);
  if HENDT>HHIDAT then HHIDAT=HENDT;
end;
```

```
HCUR=0; HNCUR=0;
```

```
  if HNNT01 ge DOLIDAT and HNNT01=HHIDAT then do;
    HCPS101=1; if HNNT01=DOIDAT then HCUR=HCUR+1;
    else if HNNT01<DOIDAT then HNCUR=HNCUR+1; end;
```

```
  if HNNT02 ge DOLIDAT and HNNT02=HHIDAT then do;
    HCPS102=1; if HNNT02=DOIDAT then HCUR=HCUR+1;
    else if HNNT02<DOIDAT then HNCUR=HNCUR+1; end;
```

```
  if HNNT03 ge DOLIDAT and HNNT03=HHIDAT then do;
    HCPS103=1; if HNNT03=DOIDAT then HCUR=HCUR+1;
    else if HNNT03<DOIDAT then HNCUR=HNCUR+1; end;
```

```
  if HNNT04 ge DOLIDAT and HNNT04=HHIDAT then do;
    HCPS104=1; if HNNT04=DOIDAT then HCUR=HCUR+1;
    else if HNNT04<DOIDAT then HNCUR=HNCUR+1; end;
```

```
  if HNNT05 ge DOLIDAT and HNNT05=HHIDAT then do;
    HCPS105=1; if HNNT05=DOIDAT then HCUR=HCUR+1;
    else if HNNT05<DOIDAT then HNCUR=HNCUR+1; end;
```

Appendix 42: The NLSW01

if HNDT06 ge DOLIDAT and HNDT06=HHIDAT then do;
HCPS106=1; if HNDT06=DOIDAT then HCUR=HCUR+1;
else if HNDT06<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT07 ge DOLIDAT and HNDT07=HHIDAT then do;
HCPS107=1; if HNDT07=DOIDAT then HCUR=HCUR+1;
else if HNDT07<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT08 ge DOLIDAT and HNDT08=HHIDAT then do;
HCPS108=1; if HNDT08=DOIDAT then HCUR=HCUR+1;
else if HNDT08<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT09 ge DOLIDAT and HNDT09=HHIDAT then do;
HCPS109=1; if HNDT09=DOIDAT then HCUR=HCUR+1;
else if HNDT09<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT10 ge DOLIDAT and HNDT10=HHIDAT then do;
HCPS110=1; if HNDT10=DOIDAT then HCUR=HCUR+1;
else if HNDT10<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT11 ge DOLIDAT and HNDT11=HHIDAT then do;
HCPS111=1; if HNDT11=DOIDAT then HCUR=HCUR+1;
else if HNDT11<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT12 ge DOLIDAT and HNDT12=HHIDAT then do;
HCPS112=1; if HNDT12=DOIDAT then HCUR=HCUR+1;
else if HNDT12<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT13 ge DOLIDAT and HNDT13=HHIDAT then do;
HCPS113=1; if HNDT13=DOIDAT then HCUR=HCUR+1;
else if HNDT13<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT14 ge DOLIDAT and HNDT14=HHIDAT then do;
HCPS114=1; if HNDT14=DOIDAT then HCUR=HCUR+1;
else if HNDT14<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT15 ge DOLIDAT and HNDT15=HHIDAT then do;
HCPS115=1; if HNDT15=DOIDAT then HCUR=HCUR+1;
else if HNDT15<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT16 ge DOLIDAT and HNDT16=HHIDAT then do;
HCPS116=1; if HNDT16=DOIDAT then HCUR=HCUR+1;
else if HNDT16<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT17 ge DOLIDAT and HNDT17=HHIDAT then do;
HCPS117=1; if HNDT17=DOIDAT then HCUR=HCUR+1;
else if HNDT17<DOIDAT then HNCUR=HNCUR+1; end;

if HNDT18 ge DOLIDAT and HNDT18=HHIDAT then do;
HCPS118=1; if HNDT18=DOIDAT then HCUR=HCUR+1;
else if HNDT18<DOIDAT then HNCUR=HNCUR+1; end;

if (HCUR>1 or (HCUR=0 and HNCUR>1)) then do;

array HCP HCPS101-HCPS118;

array HLN R5979900 R5980000 R5980100 R5980200 R5980300 R5980400 R5980500 R5980600 R5980700

Appendix 42: The NLSW01

```

R5980800 R5980900 R5981000 R5981100 R5981200 R5981300 R5981400 R5981500 R5981600;
do over HCP;
  if HLN ne 1 then HCP=.;
end;
end;

```

```

if Hcur ge 1 then K00400=1; else if Hcur=0 and Hncur ge 1 then K00400=2; else K00400=3;
if Hcur=1 then K00500=1; else if Hcur>1 then K00500=2;
if Hcur=0 and Hncur=1 then K00600=1; else if Hcur=0 and Hncur>1 then K00600=2;
if K00400=1 or K00400=2 then K00800=1; else K00800=0;

```

/* EDIT THE RESPONDENT EMPLOYER SORT SECTION */

```

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

```

```

HIDAT=0;
array SORTY R5583600 R5583700 R5583800 R5583900 R5584000 R5584100 R5584200
R5584300 R5584400 R5584500 R5584600 R5584700 R5584800 R5584900;
array SORTM R5580800 R5580900 R5581000 R5581100 R5581200 R5581300 R5581400
R5581500 R5581600 R5581700 R5581800 R5581900 R5582000 R5582100;
array SORTD R5582200 R5582300 R5582400 R5582500 R5582600 R5582700 R5582800
R5582900 R5583000 R5583100 R5583200 R5583300 R5583400 R5583500;
array SORT SORTDA01-SORTDA14;
array ENDT ENDT01-ENDT14;
array STK R5590500 R5590600 R5590700 R5590800 R5590900 R5591000 R5591100
R5591200 R5591300 R5591400 R5591500 R5591600 R5591700 R5591800;
array RFM R5586300 R5586400 R5586500 R5586600 R5586700 R5586800 R5586900
R5587000 R5587100 R5587200 R5587300 R5587400 R5587500 R5587600;
array RFD R5587700 R5587800 R5587900 R5588000 R5588100 R5588200 R5588300
R5588400 R5588500 R5588600 R5588700 R5588800 R5588900 R5589000;

```

```

do over RFD;
  if STK=2 and RFD=. then RFD=15;
  if STK>0 and RFM=. then RFM=01;
  if SORTM=2 and SORTD>28 then SORTD=28;
  if (SORTM=4 or SORTM=6 or SORTM=9 or SORTM=11) and SORTD=31 then SORTD=30;
  SORT=SortY*10000+SortM*100+SortD;
  ENDT=MDY(SORTM,SORTD,SortY);
  if ENDT>HIDAT then HIDAT=ENDT;
end;

```

```

CUR=0; NCUR=0;
if ENDT01 ge DOLIDAT and ENDT01=HIDAT then do;
  CPS101=1; if ENDT01=DOIDAT then CUR=CUR+1;
  else if ENDT01<DOIDAT then NCUR=NCUR+1; end;

```

```

if ENDT02 ge DOLIDAT and ENDT02=HIDAT then do;
  CPS102=1; if ENDT02=DOIDAT then CUR=CUR+1;
  else if ENDT02<DOIDAT then NCUR=NCUR+1; end;

```

```

if ENDT03 ge DOLIDAT and ENDT03=HIDAT then do;

```

Appendix 42: The NLSW01

```
CPS103=1; if ENDT03=DOIDAT then CUR=CUR+1;
else if ENDT03<DOIDAT then NCUR=NCUR+1; end;

if ENDT04 ge DOLIDAT and ENDT04=HIDAT then do;
  CPS104=1; if ENDT04=DOIDAT then CUR=CUR+1;
  else if ENDT04<DOIDAT then NCUR=NCUR+1; end;

if ENDT05 ge DOLIDAT and ENDT05=HIDAT then do;
  CPS105=1; if ENDT05=DOIDAT then CUR=CUR+1;
  else if ENDT05<DOIDAT then NCUR=NCUR+1; end;

if ENDT06 ge DOLIDAT and ENDT06=HIDAT then do;
  CPS106=1; if ENDT06=DOIDAT then CUR=CUR+1;
  else if ENDT06<DOIDAT then NCUR=NCUR+1; end;

if ENDT07 ge DOLIDAT and ENDT07=HIDAT then do;
  CPS107=1; if ENDT07=DOIDAT then CUR=CUR+1;
  else if ENDT07<DOIDAT then NCUR=NCUR+1; end;

if ENDT08 ge DOLIDAT and ENDT08=HIDAT then do;
  CPS108=1; if ENDT08=DOIDAT then CUR=CUR+1;
  else if ENDT08<DOIDAT then NCUR=NCUR+1; end;

if ENDT09 ge DOLIDAT and ENDT09=HIDAT then do;
  CPS109=1; if ENDT09=DOIDAT then CUR=CUR+1;
  else if ENDT09<DOIDAT then NCUR=NCUR+1; end;

if ENDT10 ge DOLIDAT and ENDT10=HIDAT then do;
  CPS110=1; if ENDT10=DOIDAT then CUR=CUR+1;
  else if ENDT10<DOIDAT then NCUR=NCUR+1; end;

if ENDT11 ge DOLIDAT and ENDT11=HIDAT then do;
  CPS111=1; if ENDT11=DOIDAT then CUR=CUR+1;
  else if ENDT11<DOIDAT then NCUR=NCUR+1; end;

if ENDT12 ge DOLIDAT and ENDT12=HIDAT then do;
  CPS112=1; if ENDT12=DOIDAT then CUR=CUR+1;
  else if ENDT12<DOIDAT then NCUR=NCUR+1; end;

if ENDT13 ge DOLIDAT and ENDT13=HIDAT then do;
  CPS113=1; if ENDT13=DOIDAT then CUR=CUR+1;
  else if ENDT13<DOIDAT then NCUR=NCUR+1; end;

if ENDT14 ge DOLIDAT and ENDT14=HIDAT then do;
  CPS114=1; if ENDT14=DOIDAT then CUR=CUR+1;
  else if ENDT14<DOIDAT then NCUR=NCUR+1; end;

if (CUR>1 or (CUR=0 and NCUR>1)) then do;
  array CP CPS101-CPS114;
  array LN R5603400 R5603500 R5603600 R5603700 R5603800 R5603900 R5604000 R5604100
          R5604200 R5604300 R5604400 R5604500 R5604600 R5604700;
  do over CP; if LN ne 1 then CP=.;
  end;
end;

if CUR ge 1 then E00300=1; else if CUR=0 and NCUR ge 1 then E00300=2; else E00300=3;
```

Appendix 42: The NLSW01

```

if CUR=1 then E00400=1; else if CUR>1 then E00400=2;
if CUR=0 and NCUR=1 then E00500=1; else if CUR=0 and NCUR>1 then E00500=2;
if E00300=1 or E00300=2 then E00700=1; else E00700=0;

```

/* CREATE THE WEEKS WORKING, UNEMPLOYED, AND OUT OF THE LABOR FORCE FOR THE RESPONDENT*/

```

CSTDAT=MDY(01,01,1979);
DLICW=INTCK('WEEK',CSTDAT,DOLIDAT);
DOICW=INTCK('WEEK',CSTDAT,DOIDAT);
WBID=0;
array W      (l)  W0001- W1196;
do l=1 to 1196;
if DLICW le L le DOICW then do; W=0; WBID=WBID+1; end;
end;

array YS      (k)  R5589100 R5589200 R5589300 R5589400 R5589500 R5589600 R5589700
                  R5589800 R5589900 R5590000 R5590100 R5590200 R5590300 R5590400;
array DS      (k)  R5587700 R5587800 R5587900 R5588000 R5588100 R5588200 R5588300
                  R5588400 R5588500 R5588600 R5588700 R5588800 R5588900 R5589000;
array MS      (k)  R5586300 R5586400 R5586500 R5586600 R5586700 R5586800 R5586900
                  R5587000 R5587100 R5587200 R5587300 R5587400 R5587500 R5587600;
array YE      (k)  R5583600 R5583700 R5583800 R5583900 R5584000 R5584100 R5584200
                  R5584300 R5584400 R5584500 R5584600 R5584700 R5584800 R5584900;
array DE      (k)  R5582200 R5582300 R5582400 R5582500 R5582600 R5582700 R5582800
                  R5582900 R5583000 R5583100 R5583200 R5583300 R5583400 R5583500;
array ME      (k)  R5580800 R5580900 R5581000 R5581100 R5581200 R5581300 R5581400
                  R5581500 R5581600 R5581700 R5581800 R5581900 R5582000 R5582100;
array ESTI    (k)  R5579400 R5579500 R5579600 R5579700 R5579800 R5579900 R5580000
                  R5580100 R5580200 R5580300 R5580400 R5580500 R5580600 R5580700;

array STDAT   (k)  STDAT01-STDAT14;
array ENDAT   (k)  ENDAT01-ENDAT14;
array STCW    (k)  STCW01-STCW14;
array ENCW    (k)  ENCW01-ENCW14;

do k=1 to 14;
if ESTI=1 then do;
STDAT=MDY(MS,DS,YS);
ENDAT=MDY(ME,DE,YE);
STCW=INTCK('WEEK',CSTDAT,STDAT);
ENCW=INTCK('WEEK',CSTDAT,ENDAT);

C=0;
do L=1 to 1196; C=C+1;
if DLICW le L le DOICW then do;
if stcw>0 and STCW le C le ENCW then do;
if 1 le k le 14 then do; w=1; end;
end;
end;
end;
end;
end;
end;

WORK=0; WUMP=0; WOLF=0;
do L=1 to 1196;

```

Appendix 42: The NLSW01

```

if 001 le L le 1196 then do;
if w=1 then work=work+1;
end;
end;

if WORK<WBID then do;
do L=1 to 1196;
if 001 le L le 1196 then do;
if W=0 then do;
array gapbeg (g) dlicw encw01-encw13;
array gapend (g) stcw01 stcw02-stcw14;
array lookall (g) R5889300 R5884500 R5884600 R5884700 R5884800 R5884900 R5885000
R5885100 R5885200 R5885300 R5885400 R5885500 R5885600 R5885700;

do G=1 to 14;
if GAPBEG le L le gapend then do;
if LOOKALL=1 and WUMP ge 0 then WUMP=WUMP+1;
else if -2 le LOOKALL le -1 then WUMP=LOOKALL; end;
if gapbeg le L le DOICW and gapend=. then do;
if lookall=1 and WUMP ge 0 then WUMP=WUMP+1;
else if -2 le lookall le -1 then WUMP=lookall; end;
end;

end;
end;
end;

if WUMP ge 0 and WORK+WUMP<WBID then do;
array SNA R5889300 R5884500 R5884600 R5884700 R5884800 R5884900 R5885000
R5885100 R5885200 R5885300 R5885400 R5885500 R5885600 R5885700;
array WK R5889400 R5885800 R5885900 R5886000 R5886100 R5886200 R5886300
R5886400 R5886500 R5886600 R5886700 R390211 R5886800 R390213;

do over SNA;
if SNA=2 and WK>0 and WUMP ge 0 then WUMP=WUMP+WK;
if SNA=2 and -2 le WK le -1 then WUMP=WK;
end; end;

if WUMP>WBID-WORK then WUMP=WBID-WORK;
if WORK ge 0 and WUMP ge 0 then WOLF=WBID-(WORK+WUMP);
if WUMP<0 then WOLF=WUMP;
end;

/*CREATE CLASS OF WORKER ALL FOR NLSW01, METHOD IS SAME FOR 1995-1999*/
array CKC R5611000 R5611100 R5611200 R5611300 R5611400 R5611500 R5611600
R5611700 R5611800 R5611900 R5612000 R5612100 R5612200 R5612300;
array CW R5612400 R5612500 R5612600 R5612700 R5612800 R5612900 R5613000
R5613100 R5613200 R5613300 R5613400 R5613500 R5613600 cw14;
array CWA CWA0101-CWA0114;

do over CWA;
CWA=CW; if CKC=1 then CWA=5; end;

```

OTHER SPECIFY RESPONSES

PAR-56: Can you tell me why FATHER's estate was not distributed equally among his children?

ID	
	Mature Women
130	DOESN'T WANT TO ANSWER
1891	THE ESTATE WAS LEFT SOLELY TO WIFE
1936	BECAUSE HE DID NOT UPDATE HIS WILL. IT WAS MADE WHEN THE CHILDREN WERE SMALL.
4248	HE LIVED SO LONG THAT CHILDREN DID NOT WANT TO FIGHT ABOUT IT NN
	Young Women
187	BAD FEELINGS AMONG HIS CHILDREN NN
1003	BECAUSE OF ARRANGEMENTS MADE PRIOR TO HIS DEATH
1904	LEFT MORE TO SISTER HE HAD LIVED WITH
3347	DON'T KNOW 1 OFF THE CHILDREN RECEIVED THE HOUSE
3499	HE LEFT EVERYTHING TO MY SISTER
3801	FATHER DIDN'T LIKE HER
3889	MY MOTHER GOT IT ALL
4487	WE GOT A CROOKED LAWYER AND WE GOT A CROOKD JUDGE IN
4866	ON BAD TERMS
4989	THE SON INHERITED MORE THAN THE GIRLS BECAUSE THE FARM WAS TO BE LEFT IN THE FAMILY

Appendix 42: The NLSW01

PAR-66: Can you tell me why MOTHER's estate was not distributed equally among her children?

ID	
	Mature Women
496	BECAUSE SHE LOVED ME MOST AND I WAS THE YOUNGEST
549	EXECUTORS GOT MORE
951	MY SISTER TOOK CARE OF HER WE DECIDED TO GIVE HER A LITTLE MORE
970	BROTHER CONVINCED MOTHER TO SIGN 10 ACRES OF PROPERTY OVER TO HIM
1176	SISTER PAID THE TAX ON THE PROPERTY
1375	IT WAS HER WISHES.
1908	THERE WAS INCLUSION OF GRANDCHILDREN
3020	BECAUSE ONE BROTHER WAS LIVING WITH AND CARING FOR HER AND THEY ALL AGREED HE SHOULD RECEIVE ALL OF ESTATE
3305	SHE TOOK THE CASH MY MOTHER HAD-SHE LIVED W/ HER
3646	N
3650	BROTHER GAVE HER ONE HALF OF WHAT HE INHERITED. HE'S VERY RICH. AND HER MOM WANTED HER TO HAVE THIS HOUSE.
4206	SIBLING WAS CAREGIVER
	Young Women
973	SHE DEEMED I HAD ENOUGH
1433	RESPONDENT WAS EXECUTRIX OF THE ESTATE AND RECEIVED THAT FEE AS WELL
1599	BECAUSE HE ALREADY RECEIVED SOME MONEY BEFORE
3020	BECAUSE I WAS HER CARE GIVER. SIBLINGS WANTED HER TO HAVE IT
3053	HE HAD ALREADY RECEIVED VALUE DURING THE PARENTS LIFETIME.
3057	HER ESTATE ALL WENT TO THE CARE OF THE HANDICAPPED BROTHER BY AGREEMENT OF ALL 6 CHILDREN
3833	BECAUSE BIG BROTHER ALLREADY WELL OFF AND YOUNGER BROTHER GOT HIS BEFORE SHE DIED AND SHE WAS LIVING WITH HIM
3889	BECAUSE I DID MORE FOR MY MOTHER THAN MY BROTHERS DID
4237	SOME RECEIVED STUFF BEFORE SHE DIED
4292	FAMILY DECIDED MONEY SHOULD GO TO YOUNGEST CHILDREN
4351	BECAUSE THE FAMILY WAS SPLIT AND STEPCHILDREN WERE INVOLVED
4487	CROOKED JUDGES AND LAWYERS
4546	HAD THE MOST NEED FOR THE HOUSE, 7 OTHER SIBLINGS AGREED.
4602	SISTER WOULD GET MORE BECAUSE SHE WAS YOUNGER
4658	IT WAS LEFT IN MY SISTER'S NAME AND AT THE TIME I DIDN'T ASK FOR IT BECAUSE SOMEONE WOULD GET IT. PERSONAL REASONS
4978	LESS TO THE YOUNGER CHILD