

Growing Up in Scotland Birth Cohort 2 Sweep 3: Age 5

Derived Variables Syntax

ScotCen Social Research

Scotiabank House
6 South Charlotte Street
Edinburgh, EH2 4AW
T 0131 240 0210

www.scotcen.org.uk

A Company Limited by Guarantee. Registered in England No.4392418.
A Charity registered in England and Wales (1091768) and Scotland
(SC038454)

Contents

Admin variables	3
Household derived variables	3
Parenting Styles	13
Parental Support	13
Sibling relationships	14
Non-resident parents	14
Pre-school	14
Childcare	14
Primary School	56
Child Health & Development	56
Self Control	63
Activities	63
Physical Activity	64
Food & Eating	64
Employment, Education & Demographics	65
Income & Financial Stress	89
Housing & Accommodation	94
BMI, Height & Weight	94

Admin variables

***DATE OF INTERVIEW - CALCULATING MONTH AND QUARTER

```
COMPUTE DeXmnth1 = XDATE.MONTH(IntDate_) .  
EXECUTE.
```

```
Var labs DeXmnth1 'De Month of interview'.
```

```
Val labs DeXmnth1
```

```
1 'January'
```

```
2 'February'
```

```
3 'March'
```

```
4 'April'
```

```
5 'May'
```

```
6 'June'
```

```
7 'July'
```

```
8 'August'
```

```
9 'September'
```

```
10 'October'
```

```
11 'November'
```

```
12 'December'.
```

```
FORMATS DeXmnth1 (F2.0).
```

```
freq DeXmnth1.
```

```
COMPUTE DeXqurt1 = XDATE.QUARTER(IntDate_) .  
EXECUTE.
```

```
Var labs DeXqurt1 'De Quarter of interview'.
```

```
Val labs DeXqurt1
```

```
1 'January to March'
```

```
2 'April to June'
```

```
3 'July to September'
```

```
4 'October to December'.
```

```
FORMATS DeXqurt1 (F2.0).
```

```
freq DeXqurt1.
```

Household derived variables

Derived variables

DeHGrsp04	De - Family Type
DeHGagC	De Study child's age at interview (months)
DeRespAg	Dd - Respondent's age (banded)
DeRPage	Dd - Resp partner's age (banded)
DeHGbord	De - Study child's birth order
DeHGkds	De Nbr of persons aged 0-18 in household
DeHGkds2	De Nbr of persons aged 0-13 in household
DeHGkds3	De Nbr of persons aged 14-18 in household
DeHGads	De Nbr of persons aged 19 or over in household
DeHGnmad	De Number of adults (16 or over) in household
DeHGnmad2	De Nbr of adults other than resp in hhold - banded
DeHGnmk2	De Nbr of children in household - Banded
DeHGprim	De Whether child was mother's first-born
DeHGrsp05	De Resp is child's mother? (incl. adopt./foster/step-mothers)
DeHGrsp06	De Resp is child's father? (incl. adopt./foster/step-fathers)
DeHGrsp07	De Resp's relationship to study child
DeHGrsp08	Da Resps partner relation to the child

DeHGmag1	Dd Mother's date of birth
DeHGmag2	Dd Childs nat mother's age at interview
DeHGmag3	De Childs nat mothers age at interview (banded)
DeHGmag4	Da Age (years) of nat mother at birth of cohort child
DeHGmag5	Dd Nat mother's age at birth of cohort child (banded)
DeHGrsp04	De - Family Type
DeHGhsiz	De Household size

**LONE PARENT VS. COUPLE HOUSEHOLD - 'Family Type'

CRO RParSpo by BSpous2.

RENAME VARS (BSpous2=DeHGnp04).

VAR LABS DeHGnp04 'De - Respondent living with spouse/partner'.
 VAL LABS DeHGnp04
 0 'Respondent not living with spouse/partner'
 1 'Respondent living with spouse/partner'.

COMPUTE DeHGrsp04=DeHGnp04.
 Execute.

var labs DeHGrsp04 'De - Family Type'.
 val labs DeHGrsp04
 0 'Lone Parent'
 1 'Couple Family'.
 FORMATS DeHGrsp04 (F2.0).

fre DeHGrsp04.

*** STUDY CHILD'S AGE IN YEARS

FREQ CDobChk ChCDoB_.

IF (CDobChk=1) x_DVAge16Full = CTIME.DAYS(IntDate_ - CDob_2)/365.24.
 IF (CDobChk=2) x_DVAge16Full = CTIME.DAYS(IntDate_ - ChCDoB_)/365.24.
 Execute.
 FORMATS x_DVAge16Full (F5.3).

COMPUTE x_DVAge16 = TRUNC(x_DVAge16Full,0.1) .
 Execute.
 FORMATS x_DVAge16 (F3.1).
 FREQ x_DVAge16.

* Range 2.7 to 3.4 yr; only 12 out of 5019 are 2.7yrs; vast majority 2.8 to 3.0 yrs, so OK.

Compute x_DeHGagC = x_DVAge16Full *12.
 Exe.
 Var Labs x_DeHGagC 'De Study child's RECALCULATED age at interview (months 1 dp)'.
 FORMATS x_DeHGagC (F2.1).
 freq x_DeHGagC.

* range 33.2 to 41.5 months.

Compute y_DeHGagC = TRUNC(x_DeHGagC).
 Exe.
 VAR LABS y_DeHGagC 'Truncated value of x_DeHGagC'.
 FORMATS y_DeHGagC (F2.0).
 FREQ y_DeHGagC.

* and renaming TRUNCATED var y_DeHGagC as the final var DeHGagC to be kept in dataset
 * label amended to final for DeHGagC

RENAME VARS (y_DeHGagC=DeHGagC).

Var Labs DeHGagC 'De Study child's age at interview (months)'.
FREQ DeHGagC.

***** RECALCULATING RESPONDENT's and PARTNER's (full truncated) AGES
** CAPI DVs 'RAge' and 'PAge' need to be recalculated as well since based on incorrect DVAge[...].
* based on Respondent ID 'MainResp='RID' and Partner ID 'PID' (which is based on RParSpo, not BSpous2).

FREQ MainResp RID PID.

** respondent is always Person 2, so just use Person's 2 full (truncated) age in years:

RECODE DeHGag2 (else=copy) INTO x_DeRespAg.
Exe.
VAR LABS x_DeRespAg "Dd - Respondent's age (years)".
VALUE LABELS x_DeRespAg
-1 'Not applicable'.
MISSING VALUES x_DeRespAg (-9 thru -1).
FORMATS x_DeRespAg (F2.0).
FREQ x_DeRespAg.

** Partner's ID changes (Person 3 to 10 in hhold) so need to allocate correct Person's age.
* Note: PID based on RParSpo, not (BSpous2=)

RENAME VARS (PID=DePartID).
Var Labs DePartID 'De - Partner ID'.
ADD VAL LABS DePartID
0 'N/A - Not in household'
1 'Person 1' 2 'Person 2' 3 'Person 3' 4 'Person 4' 5 'Person 5' 6 'Person 6' 7 'Person 7' 8 'Person 8'
9 'Person 9' 10 'Person 10' 11 'Person 11' 12 'Person 12' 13 'Person 13' 14 'Person 14' 15 'Person 15'.

* and correcting for the hholds with missing info:

IF (DeHGnp04=1) AND (RParSpo=0) DePartID=-3.
Exe.
ADD VAL LABS DePartID
-3 'Information not available'.
FREQ DePartID.

* allocating ages (starting from Person 3 since Person 2 is always the Resp in this sweep, and finishing with Person 10, as per frequency range at PID):

IF (DeHGnp04=1) AND (RParSpo=0) x_DeRPage=-3.
Exe.
IF (DePartID=3) x_DeRPage=DeHGag3.
IF (DePartID=4) x_DeRPage=DeHGag4.
IF (DePartID=5) x_DeRPage=DeHGag5.
IF (DePartID=6) x_DeRPage=DeHGag6.
IF (DePartID=7) x_DeRPage=DeHGag7.
IF (DePartID=8) x_DeRPage=DeHGag8.
IF (DePartID=9) x_DeRPage=DeHGag9.
IF (DePartID=10) x_DeRPage=DeHGag10.
Exe.
RECODE x_DeRPage (sysmis=-1).
Exe.
VAR LABS x_DeRPage "Dd - Partner's age (years)".
VALUE LABELS x_DeRPage
-1 'Not applicable'
-3 'Information not available'.
MISSING VALUES x_DeRPage (-9 thru -1).
FORMATS x_DeRPage (F2.0).
FREQ x_DeRPage.

* range 19 to 81 extremes at top end.
* comparing with (incorrect) CAPI DV 'PAge'.

FREQ PAge.

temporary.

select if (Page ge 75).
list serial DePartID x_DeRPage.

* the (archived) banded vars will be DaRespAg and DaRPage below.

* note: banding not same as used for [...]HGag2 to 12: under 20, 20 to 29, 30 to 39 and 40 or older instead of 10-year intervals

* same compressed banding as used at [...]HGmag2/3.. re mother's age (as done further down)

* so age banding varies between age DVs.

RECODE

x_DeRespAg x_DeRPage

(1 THRU 19=1) (20 THRU 29=2) (30 THRU 39=3) (40 THRU Hi=4) (else=copy) INTO DeRespAg

DeRPage.

Exe.

VAR LABS DeRespAg "Dd - Respondent's age (banded)".

VAR LABS DeRPage "Dd - Resp partner's age (banded)".

VALUE LABELS DeRespAg DeRPage

1 'Under 20'

2 '20 to 29'

3 '30 to 39'

4 '40 or older'

-1 'Not applicable'

-3 'Information not available'.

MISSING VALUES DeRespAg DeRPage (-9 thru -1).

FORMATS DeRespAg DeRPage (F2.0).

FREQ DeRespAg DeRPage.

*****To calculate SIZE OF HOUSEHOLD = DeHGhsiz.

*Note: there are equivalent CAPI derived vars = (identical vars) DVHSize2 & DMHSIZE2 but probably incorrect (as in earlier sweeps) - get different results from calculated DV

*Need first to delete the original feedforward section PersNo (row 131) to DVMarD15 (row 250) re "Person1 to 15" to be able to renumber them - Sw6 CAPI data start at "Person no 16"

delete variables Ckl to XFlags10.

* nbr of hhold members stop at max available = Person 15.

RENAME VARS (PersNo16=PersNo1) (PersNo17=PersNo2) (PersNo18=PersNo3) (PersNo19=PersNo4)

(PersNo20=PersNo5)

(PersNo21=PersNo6) (PersNo22=PersNo7) (PersNo23=PersNo8) (PersNo24=PersNo9)

(PersNo25=PersNo10)

(PersNo26=PersNo11) (PersNo27=PersNo12) (PersNo28=PersNo13) (PersNo29=PersNo14)

(PersNo30=PersNo15).

Var Labs PersNo1 'Me - ID person 1 - study child'.

Var Labs PersNo2 'Me - ID person 2'.

Var Labs PersNo3 'Me - ID person 3'.

Var Labs PersNo4 'Me - ID person 4'.

Var Labs PersNo5 'Me - ID person 5'.

Var Labs PersNo6 'Me - ID person 6'.

Var Labs PersNo7 'Me - ID person 7'.

Var Labs PersNo8 'Me - ID person 8'.

Var Labs PersNo9 'Me - ID person 9'.

Var Labs PersNo10 'Me - ID person 10'.

Var Labs PersNo11 'Me - ID person 11'.

Var Labs PersNo12 'Me - ID person 12'.

Var Labs PersNo13 'Me - ID person 13'.

Var Labs PersNo14 'Me - ID person 14'.

Var Labs PersNo15 'Me - ID person 15'.

Recode PersNo1 PersNo2 PersNo3 PersNo4 PersNo5 PersNo6 PersNo7 PersNo8 PersNo9 PersNo10

PersNo11 PersNo12

PersNo13 PersNo14 PersNo15 (-1 = 0) (else = 1) INTO RPN01 RPN02 RPN03 RPN04 RPN05 RPN06

RPN07 RPN08 RPN09

RPN010 RPN011 RPN012 RPN013 RPN014 RPN015.

Exe.

Count DeHGhsiz = RPN01 RPN02 RPN03 RPN04 RPN05 RPN06 RPN07 RPN08 RPN09
RPN010 RPN011 RPN012 RPN013 RPN014 RPN015 (1).
Exe.

var labs DeHGhsiz 'De Household size'.
FORMATS DeHGhsiz (F2.0).

FREQ DeHGhsiz.

**NUMBER OF ADULTS & CHILDREN (< 16 yr old) IN HOUSEHOLD.
* NOT USING Variable already in dataset - 'Numchild' as likely to be incorrect since all CAPI ages
calculated incorrectly,

** NUMBER OF CHILDREN IN HHOLD 'NumChHH': count if AdCh.. answers are 1:

Count NumChHH = AdCh1 AdCh2 AdCh3 AdCh4 AdCh5 AdCh6 AdCh7 AdCh8 AdCh9 AdCh10 AdCh11
AdCh12 AdCh13 AdCh14 AdCh15 (1).
Exe.

RENAME VARS (NumChHH=DeHGnmkd).
var labs DeHGnmkd 'De Number of children in household'.

** NUMBER OF ADULTS IN HHOLD 'NumAdHH': count if AdCh.. answers are 2:

Count NumAdHH = AdCh1 AdCh2 AdCh3 AdCh4 AdCh5 AdCh6 AdCh7 AdCh8 AdCh9 AdCh10 AdCh11
AdCh12 AdCh13 AdCh14 AdCh15 (2).
Exe.

RENAME VARS (NumAdHH=DeHGnmd).
var labs DeHGnmd 'De Number of adults (16 or over) in household'.

FORMATS DeHGnmkd DeHGnmd (F2.0).
FREQ DeHGnmkd DeHGnmd.

* checking that nbr of adults in HHold + nbr of children in HHold mtach total of people in HHold:

Compute x_NbrAdltChld=DeHGnmkd+DeHGnmd.
Exe.
FORMATS x_NbrAdltChld (F2.0).
FREQ x_NbrAdltChld.

Compute x_NbrAdltChld=SUM(DeHGnmkd,DeHGnmd).
Exe.
FORMATS x_NbrAdltChld (F2.0).
FREQ x_NbrAdltChld.

* total children + adults match original nbr in HHold 'DeHGhsiz' OK.

****TO RECODE DeHGnmd AND CREATE A VARIABLE INDICATING THE NUMBER OF 'OTHER'
ADULTS IN THE HOUSEHOLD.
**THAT IS, OTHER THAN THE RESPONDENT

Recode DeHGnmd (1 = 0) (2 = 1) (3 thru hi = 2) into DeHGnmd2.
Exe.
var labs DeHGnmd2 'De Nbr of adults other than resp in hhold - banded'.
val labs DeHGnmd2
0 'None'
1 'One'
2 'Two or more'.
FORMATS DeHGnmd2 (F2.0).

fre DeHGnmd2.

** To tidy up var re NUMBER OF CHILDREN IN HOUSEHOLD - Banded version

Recode DeHGnmkd (1=copy) (2 thru 3=2) (4 thru hi=3) (else=copy) into DeHGnmk2.
Execute.
Var labs DeHGnmk2 'De Nbr of children in household - Banded'.

Val labs DeHGnmk2

- 1 'One'
- 2 'Two or three'
- 3 'Four or more'.

FORMATS DeHGnmk2 (F2.0).

FREQ DeHGnmk2.

FREQ Cpay06 NumChil2.

**PRIMAPOROUS (1st time) MOTHERS - 'Prim'

RENAME VARS (BirthOrd2=DeHGbord).

VAR LABS DeHGbord 'De - Study child s birth order'.

FORMATS DeHGbord (F2.0).

FREQ DeHGbord.

RECODE DeHGbord (1=1) (2 thru Highest=2) INTO DeHGprim.

EXECUTE.

Var labs DeHGprim 'De Whether child was mothers first-born'.

Val labs DeHGprim

- 1 'Yes - first born'
- 2 'No - other children' .

FORMATS DeHGprim (F2.0).

FREQ DeHGprim.

** WHETHER RESPONDENT IS CHILD'S MOTHER (INCLUDING FOSTER, ADOPTIVE, STEP-MOTHERS)

* (syntax below copied from weight prep syntax file).

* first: MainResp renamed using same name as at BC1 Sw1 (question actually asked, not a DV 'DeRespID'

* like later sweeps in BC1).

* relationship vars R241, 256 etc (+15 each time) renamed as MeHGr21, 31 etc and gender vars also renamed.

FREQ MeCdom01.

**WHETHER RESPONDENT IS CHILD'S FATHER (INCLUDING FOSTER, ADOPTIVE, STEP-FATHERS)

Compute DeHGrsp06 = 0.

If (MeCdom01 = 2) AND (MeHGr21 > 7) AND (MeHGr21 < 12) AND (MeHGsx2 = 1) DeHGrsp06 = 1.

Exe.

Var labs DeHGrsp06 'De Resp is childs father? (incl. adopt./foster/step-fathers)'.

Val labs DeHGrsp06

- 0 'No'
- 1 'Yes'.

FORMATS DeHGrsp06 (F2.0).

fre DeHGrsp06.

**WHO IS THE RESPONDENT IN RELATION TO THE CHILD?

Compute DeHGrsp07 = 3.

If (DeHGrsp05 = 1) DeHGrsp07 = 1.

If (DeHGrsp06 = 1) DeHGrsp07 = 2.

Exe.

Val labs DeHGrsp07

- 1 'Mother'
- 2 'Father'
- 3 'Someone else'.

Var labs DeHGrsp07 'De Resp's relationship to study child'.

FORMATS DeHGrsp07 (F2.0).

fre DeHGrsp07.

**WHO IS THE PARTNER IN RELATION TO THE CHILD

FREQ DePartID.

Compute DeHGrsp08 = 3.
If (DePartID = 0) DeHGrsp08 = -1.
Exe.

If (DePartID = 2) AND (MeHGr21 > 7) AND (MeHGr21 < 12) AND (MeHGsx2 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 2) AND (MeHGr21 > 7) AND (MeHGr21 < 12) AND (MeHGsx2 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 3) AND (MeHGr31 > 7) AND (MeHGr31 < 12) AND (MeHGsx3 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 3) AND (MeHGr31 > 7) AND (MeHGr31 < 12) AND (MeHGsx3 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 4) AND (MeHGr41 > 7) AND (MeHGr41 < 12) AND (MeHGsx4 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 4) AND (MeHGr41 > 7) AND (MeHGr41 < 12) AND (MeHGsx4 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 5) AND (MeHGr51 > 7) AND (MeHGr51 < 12) AND (MeHGsx5 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 5) AND (MeHGr51 > 7) AND (MeHGr51 < 12) AND (MeHGsx5 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 6) AND (MeHGr61 > 7) AND (MeHGr61 < 12) AND (MeHGsx6 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 6) AND (MeHGr61 > 7) AND (MeHGr61 < 12) AND (MeHGsx6 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 7) AND (MeHGr71 > 7) AND (MeHGr71 < 12) AND (MeHGsx7 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 7) AND (MeHGr71 > 7) AND (MeHGr71 < 12) AND (MeHGsx7 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 8) AND (MeHGr81 > 7) AND (MeHGr81 < 12) AND (MeHGsx8 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 8) AND (MeHGr81 > 7) AND (MeHGr81 < 12) AND (MeHGsx8 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 9) AND (MeHGr91 > 7) AND (MeHGr91 < 12) AND (MeHGsx9 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 9) AND (MeHGr91 > 7) AND (MeHGr91 < 12) AND (MeHGsx9 = 1) DeHGrsp08 = 2.
Exe.

If (DePartID = 10) AND (MeHGr101 > 7) AND (MeHGr101 < 12) AND (MeHGsx10 = 2) DeHGrsp08 = 1.
Exe.

If (DePartID = 10) AND (MeHGr101 > 7) AND (MeHGr101 < 12) AND (MeHGsx10 = 1) DeHGrsp08 = 2.
Exe.

Val labs DeHGrsp08

1 'Mother' 2 'Father' 3 'Someone else' -1 'No partner in household' -3 'Information not available'.

Var labs DeHGrsp08 'Da Resps partner relation to the child'.

Missing values DeHGrsp08 (-9 thru -1).

FORMATS DeHGrsp08 (F2.0).

fre DeHGrsp08.

***AGE OF natural MOTHER AT INTERVIEW (BANDED) - 'DeHGmag3'

* (truncated ages so no dp necessary in interval boundaries below).

RECODE

DeHGmag2

(1 THRU 19=1) (20 THRU 29=2) (30 THRU 39=3) (40 THRU Hi=4) (else=copy) INTO DeHGmag3.

VARIABLE LABELS DeHGmag3 'De Childs nat mothers age at interview (banded)'.

VALUE LABELS DeHGmag3

1 'Under 20'

2 '20 to 29'

3 '30 to 39'

4 '40 or older'

-1 'Not applicable'.

EXECUTE.

MISSING VALUES DeHGmag3 (-1).

FORMATS DeHGmag3 (F2.0).

fre DeHGmag3.

***AGE OF (natural) MOTHER AT BIRTH of cohort child (BANDED) - DeHGmag5
* using CAPI DV BMum2 - WHICH MAY BE CORRECT OR NOT - re 'Resp is natural mother' to be renamed
* as DeHGrsp01 to recode the missing values as 'not known' rather than 'N/A'.

RENAME VARS (BMum2 = DeHGrsp01).
VAR LABS DeHGrsp01 'De - Whether respondent is natural mother'.

* DeHGmag4 is truncated age so OK to use integers as boundaries for DeHGmag5 below

RECODE
DeHGmag4
(1 THRU 19=1) (20 THRU 29=2) (30 THRU 39=3) (40 THRU Hi=4) (else=copy) INTO DeHGmag5.
Exe.
VARIABLE LABELS DeHGmag5 "Dd Nat mother's age at birth of cohort child (banded)".
VALUE LABELS DeHGmag5
1 'Under 20'
2 '20 to 29'
3 '30 to 39'
4 '40 or older'
-3 'Information not available'.
FORMATS DeHGmag5 (F2.0).
MISSING VALUES DeHGmag5 (-3).

fre DeHGmag5.

* hhold members leaving hhold ~ when..

*person2 ~ LineNo17 StlLivCk2 StlLiv17 LftMo_17 More17.
do if(StlLivCk2=1)or(StlLiv17 ge 3).

COMPUTE DeHgLm02 = XDATE.MONTH(LftMo_17) .

COMPUTE DeHgLy02 = XDATE.YEAR(LftMo_17) .

end if.

EXECUTE .

frequencies DeHgLm02 DeHgLy02.

*person3 ~ LineNo18 StlLivCk3 StlLiv18 LftMo_18 More18.

do if(StlLivCk3=1)or(StlLiv18 ge 3).

COMPUTE DeHgLm03 = XDATE.MONTH(LftMo_18) .

COMPUTE DeHgLy03 = XDATE.YEAR(LftMo_18) .

end if.

EXECUTE .

frequencies DeHgLm03 DeHgLy03.

*person4 ~ LineNo19 StlLivCk4 StlLiv19 LftMo_19 More19.

do if(StlLivCk4=1)or(StlLiv19 ge 3).

COMPUTE DeHgLm04 = XDATE.MONTH(LftMo_19) .

COMPUTE DeHgLy04 = XDATE.YEAR(LftMo_19) .

end if.

EXECUTE .

*person5 ~ LineNo20 StlLivCk5 StlLiv20 LftMo_20 More20.

do if(StlLivCk5=1)or(StlLiv20 ge 3).

COMPUTE DeHgLm05 = XDATE.MONTH(LftMo_20) .

COMPUTE DeHgLy05 = XDATE.YEAR(LftMo_20) .

end if.

EXECUTE .

*person6 ~ LineNo21 StlLivCk6 StlLiv21 LftMo_21 More21.

do if(StlLivCk6=1)or(StlLiv21 ge 3).

COMPUTE DeHgLm06 = XDATE.MONTH(LftMo_21) .

COMPUTE DeHgLy06 = XDATE.YEAR(LftMo_21) .

end if.

EXECUTE .

*person7 ~ LineNo22 StlLivCk7 StlLiv22 LftMo_22 More22.

do if(StlLivCk7=1)or(StlLiv22 ge 3).

COMPUTE DeHgLm07 = XDATE.MONTH(LftMo_22) .

COMPUTE DeHgLy07 = XDATE.YEAR(LftMo_22) .

end if.

EXECUTE .

*person8 ~ LineNo23 StlLivCk8 StlLiv23 LftMo_23 More23.

do if(StlLivCk8=1)or(StlLiv23 ge 3).

```

COMPUTE DeHgLm08 = XDATE.MONTH(LftMo_23) .
COMPUTE DeHgLy08 = XDATE.YEAR(LftMo_23) .
end if.
EXECUTE .
*person9 ~ LineNo24 StlLivCk9 StlLiv24 LftMo_24 More24.
do if(StlLivCk9=1)or(StlLiv24 ge 3).
COMPUTE DeHgLm09 = XDATE.MONTH(LftMo_24) .
COMPUTE DeHgLy09 = XDATE.YEAR(LftMo_24) .
end if.
EXECUTE .
*person10 ~ LineNo25 StlLivCk10 StlLiv25 LftMo_25 More25.
do if(StlLivCk10=1)or(StlLiv25 ge 3).
COMPUTE DeHgLm10 = XDATE.MONTH(LftMo_25) .
COMPUTE DeHgLy10 = XDATE.YEAR(LftMo_25) .
end if.
EXECUTE .
*person11 ~ LineNo26 StlLivCk11 StlLiv26 LftMo_26 More26.
do if(StlLivCk11=1)or(StlLiv26 ge 3).
COMPUTE DeHgLm11 = XDATE.MONTH(LftMo_26) .
COMPUTE DeHgLy11 = XDATE.YEAR(LftMo_26) .
end if.
EXECUTE .
*person12 ~ LineNo27 StlLivCk12 StlLiv27 LftMo_27 More27.
do if(StlLivCk12=1)or(StlLiv27 ge 3).
COMPUTE DeHgLm12 = XDATE.MONTH(LftMo_27) .
COMPUTE DeHgLy12 = XDATE.YEAR(LftMo_27) .
end if.
EXECUTE .
*person13 ~ LineNo28 StlLivCk13 StlLiv28 LftMo_28 More28.
do if(StlLivCk13=1)or(StlLiv28 ge 3).
COMPUTE DeHgLm13 = XDATE.MONTH(LftMo_28) .
COMPUTE DeHgLy13 = XDATE.YEAR(LftMo_28) .
end if.
EXECUTE .
*person14 ~ LineNo29 StlLivCk14 StlLiv29 LftMo_29 More29.
do if(StlLivCk14=1)or(StlLiv29 ge 3).
COMPUTE DeHgLm14 = XDATE.MONTH(LftMo_29) .
COMPUTE DeHgLy14 = XDATE.YEAR(LftMo_29) .
end if.
EXECUTE .
*person15 ~ LineNo30 StlLivCk15 StlLiv30 LftMo_30 More30.
do if(StlLivCk15=1)or(StlLiv30 ge 3).
COMPUTE DeHgLm15 = XDATE.MONTH(LftMo_30) .
COMPUTE DeHgLy15 = XDATE.YEAR(LftMo_30) .
end if.
EXECUTE .

```

```

formats DeHgLm02 DeHgLy02 DeHgLm03 DeHgLy03 DeHgLm04 DeHgLy04 DeHgLm05 DeHgLy05
DeHgLm06 DeHgLy06 DeHgLm07 DeHgLy07 DeHgLm08 DeHgLy08 DeHgLm09 DeHgLy09 DeHgLm10
DeHgLm11 DeHgLy11 DeHgLm12 DeHgLy12 DeHgLm13 DeHgLy13 DeHgLm14 DeHgLy14 DeHgLm15
DeHgLy15 (F2.0).

```

```

recode DeHgLm02 DeHgLy02 DeHgLm03 DeHgLy03 DeHgLm04 DeHgLy04 DeHgLm05 DeHgLy05
DeHgLm06 DeHgLy06 DeHgLm07 DeHgLy07 DeHgLm08 DeHgLy08 DeHgLm09 DeHgLy09 DeHgLm10
DeHgLm11 DeHgLy11 DeHgLm12 DeHgLy12 DeHgLm13 DeHgLy13 DeHgLm14 DeHgLy14 DeHgLm15
DeHgLy15 (sysmis=-1).

```

```

Value labels DeHgLm02 DeHgLm03 DeHgLm04 DeHgLm05
DeHgLm06 DeHgLm07 DeHgLm08 DeHgLm09 DeHgLm10
DeHgLm11 DeHgLm12 DeHgLm13 DeHgLm14 DeHgLm15
1 'January' 2 'February' 3 'March' 4 'April' 5 'May' 6 'June' 7 'July' 8 'August'
9 'September' 10 'October' 11 'November' 12 'December' .

```

```

add value labels DeHgLm02 DeHgLy02 DeHgLm03 DeHgLy03 DeHgLm04 DeHgLy04 DeHgLm05
DeHgLy05
DeHgLm06 DeHgLy06 DeHgLm07 DeHgLy07 DeHgLm08 DeHgLy08 DeHgLm09 DeHgLy09 DeHgLm10
DeHgLy10

```

DeHgLm11 DeHgLy11 DeHgLm12 DeHgLy12 DeHgLm13 DeHgLy13 DeHgLm14 DeHgLy14 DeHgLm15
DeHgLy15 -1 'not applicable'.

variable labels

DeHgLm02	'De Month left - person 2'
DeHgLy02	'De Year left - person 2'
DeHgLm03	'De Month left - person 3'
DeHgLy03	'De Year left - person 3'
DeHgLm04	'De Month left - person 4'
DeHgLy04	'De Year left - person 4'
DeHgLm05	'De Month left - person 5'
DeHgLy05	'De Year left - person 5'
DeHgLm06	'De Month left - person 6'
DeHgLy06	'De Year left - person 6'
DeHgLm07	'De Month left - person 7'
DeHgLy07	'De Year left - person 7'
DeHgLm08	'De Month left - person 8'
DeHgLy08	'De Year left - person 8'
DeHgLm09	'De Month left - person 9'
DeHgLy09	'De Year left - person 9'
DeHgLm10	'De Month left - person 10'
DeHgLy10	'De Year left - person 10'
DeHgLm11	'De Month left - person 11'
DeHgLy11	'De Year left - person 11'
DeHgLm12	'De Month left - person 12'
DeHgLy12	'De Year left - person 12'
DeHgLm13	'De Month left - person 13'
DeHgLy13	'De Year left - person 13'
DeHgLm14	'De Month left - person 14'
DeHgLy14	'De Year left - person 14'
DeHgLm15	'De Month left - person 15'
DeHgLy15	'De Year left - person 15'.

frequencies DeHgLm02 DeHgLy02 DeHgLm03 DeHgLy03 DeHgLm04 DeHgLy04 DeHgLm05 DeHgLy05
DeHgLm06 DeHgLy06 DeHgLm07 DeHgLy07 DeHgLm08 DeHgLy08 DeHgLm09 DeHgLy09 DeHgLm10
DeHgLy10
DeHgLm11 DeHgLy11 DeHgLm12 DeHgLy12 DeHgLm13 DeHgLy13 DeHgLm14 DeHgLy14 DeHgLm15
DeHgLy15 .

missing values DeHgLm02 DeHgLy02 DeHgLm03 DeHgLy03 DeHgLm04 DeHgLy04 DeHgLm05
DeHgLy05
DeHgLm06 DeHgLy06 DeHgLm07 DeHgLy07 DeHgLm08 DeHgLy08 DeHgLm09 DeHgLy09 DeHgLm10
DeHgLy10
DeHgLm11 DeHgLy11 DeHgLm12 DeHgLy12 DeHgLm13 DeHgLy13 DeHgLm14 DeHgLy14 DeHgLm15
DeHgLy15 (lo thru -1).

frequencies MarryD_2 MarryD_3 MarryD_4 MarryD_5 MarryD_6 MarryD_7 MarryD_8 .
*no data from this point
frequencies MarryD_9 MarryD_10 MarryD_11 MarryD_12 MarryD_13 MarryD_14 MarryD_15.

frequencies McHGmr2 DeHGmr2.

* hhold members getting married ~ when..

missing values MarryD_2 MarryD_3 MarryD_4 MarryD_5 MarryD_6 MarryD_7 MarryD_8 ().

*person2.

COMPUTE DeMarM2 = XDATE.MONTH(MarryD_2) .
COMPUTE DeMarY2 = XDATE.YEAR(MarryD_2) .
EXECUTE .
frequencies DeMarM2 DeMarY2.

*person3.

COMPUTE DeMarM3 = XDATE.MONTH(MarryD_3) .
COMPUTE DeMarY3 = XDATE.YEAR(MarryD_3) .
EXECUTE .

*person4.

COMPUTE DeMarM4 = XDATE.MONTH(MarryD_4) .
COMPUTE DeMarY4 = XDATE.YEAR(MarryD_4) .
EXECUTE .

*person5.

```

COMPUTE DeMarM5 = XDATE.MONTH(MarryD_5) .
COMPUTE DeMarY5 = XDATE.YEAR(MarryD_5) .
EXECUTE .
*person6.
COMPUTE DeMarM6 = XDATE.MONTH(MarryD_6) .
COMPUTE DeMarY6 = XDATE.YEAR(MarryD_6) .
EXECUTE .
*person7.
COMPUTE DeMarM7 = XDATE.MONTH(MarryD_7) .
COMPUTE DeMarY7 = XDATE.YEAR(MarryD_7) .
EXECUTE .
*person8.
COMPUTE DeMarM8 = XDATE.MONTH(MarryD_8) .
COMPUTE DeMarY8= XDATE.YEAR(MarryD_8) .
EXECUTE .

```

formats DeMarM2 DeMarY2 DeMarM3 DeMarY3 DeMarM4 DeMarY4 DeMarM5 DeMarY5
DeMarM6 DeMarY6 DeMarM7 DeMarY7 DeMarM8 DeMarY8 (F2.0).

recode DeMarM2 DeMarY2 DeMarM3 DeMarY3 DeMarM4 DeMarY4 DeMarM5 DeMarY5 DeMarM6
DeMarY6
DeMarM7 DeMarY7 DeMarM8 DeMarY8 (sysmis=-1).
execute.

Value labels DeMarM2 DeMarY2 DeMarM3 DeMarY3 DeMarM4 DeMarY4 DeMarM5 DeMarY5
DeMarM6 DeMarY6 DeMarM7 DeMarY7 DeMarM8 DeMarY8
1 'January' 2 'February' 3 'March' 4 'April' 5 'May' 6 'June' 7 'July' 8 'August' 9 'September' 10 'October' 11
'November' 12 'December' .

add value labels DeMarM2 DeMarY2 DeMarM3 DeMarY3 DeMarM4 DeMarY4 DeMarM5 DeMarY5
DeMarM6 DeMarY6 DeMarM7 DeMarY7 DeMarM8 DeMarY8 -1 'not applicable'.

variable labels

DeMarM2	'De Month married - person 2'
DeMarY2	'De Year married - person 2'
DeMarM3	'De Month married - person 3'
DeMarY3	'De Year married - person 3'
DeMarM4	'De Month married - person 4'
DeMarY4	'De Year married - person 4'
DeMarM5	'De Month married - person 5'
DeMarY5	'De Year married - person 5'
DeMarM6	'De Month married - person 6'
DeMarY6	'De Year married - person 6'
DeMarM7	'De Month married - person 7'
DeMarY7	'De Year married - person 7'
DeMarM8	'De Month married - person 8'
DeMarY8	'De Year married - person 8'.

missing values DeMarM2 DeMarY2 DeMarM3 DeMarY3 DeMarM4 DeMarY4 DeMarM5 DeMarY5
DeMarM6 DeMarY6 DeMarM7 DeMarY7 DeMarM8 DeMarY8 (lo thru -1).

missing values MarryD_2 MarryD_3 MarryD_4 MarryD_5 MarryD_6 MarryD_7 MarryD_8 (lo thru -1).

frequencies DeMarM2 DeMarY2 DeMarM3 DeMarY3 DeMarM4 DeMarY4 DeMarM5 DeMarY5
DeMarM6 DeMarY6 DeMarM7 DeMarY7 DeMarM8 DeMarY8.

Parenting Styles

** no DV syntax.

Parental Support

** no DV syntax:

Sibling relationships

** no DV syntax.

Non-resident parents

** no DV syntax.

Pre-school

** no DV syntax.

Childcare

Derived variables:

DeCtyp01	De Childcare from grandparents
DeCtyp02	De Childcare from other relative
DeCtyp03	De Childcare from private nursery/creche
DeCtyp04	De Childcare from childminder
DeCtyp05	De Childcare from Local Authority playgroup or pre-school
DeCtyp06	De Childcare from Local Authority creche or nursery
DeCtyp07	De Childcare from Private playgroup or pre-school
DeCtyp08	De Childcare from Community/Voluntary playgroup or pre-school
DeCtyp09	De Childcare from ex-spouse or partner
DeCtyp10	De Childcare from the child's older sibling
DeCtyp11	De Childcare from A friend or neighbour
DeCtyp12	De Childcare from Daily nanny who came to our house
DeCtyp13	De Childcare from Live-in nanny
DeCtyp14	De Childcare from Babysitter
DeCtyp15	De Childcare from Workplace creche or nursery
DeCtyp16	De Childcare from Family centre
DeCtyp17	De Childcare from nursery class attached to primary school
DeCtyp18	De Childcare from agency carer
DeCtyp19	De Childcare from other provider
DeCtyp24	De Childcare from playscheme/summer club
DeCtyp25	De Childcare from Breakfast club
DeCtyp20	De Childcare from other informal sources
DeCtyp21	De Childcare from nursery or creche

Combined (ie currently active) childcare variables

MeCtma01	Me - how many hours is child now in Prov1 care in an average week
-----------------	---

MeCdya01 Me - over how many days would Prov1 hours spread in an average week?
MeCtmb01 Me - how many hours is child now in Prov2 care in an average week
MeCdyb01 Me - over how many days would Prov2 hours spread in an average week?
MeCtmc01 Me - how many hours is child now in Prov3 care in an average week
MeCdyc01 Me - over how many days would Prov3 hours spread in an average week?
MeCtmd01 Me - how many hours is child now in Prov4 care in an average week
MeCdyd01 Me - over how many days would Prov4 hours spread in an average week?
MeCtme01 Me - how many hours is child now in Prov5 care in an average week
MeCdye01 Me - over how many days would Prov5 hours spread in an average week?

DeCtot01 De Number of ccare providers at Sweep 3
DeCtmi01 De Nbr hrs child looked after by s/one else (avge week)
DeCtmi02 De Nbr hrs child looked after by s/one else in avge week-BANDED
DeCday01 De Highest nbr of days/week in any one childcare arrangmt
DeCtyp94 De Childcare from other provider
DeCcst01 De Cost of childcare per week - sample child
DeCcst02 De Cost of ccare/week-sample child (banded)
DeCcst03 De Cost of childcare - ALL CHILDREN - per week
DeCcst04 De Cost of ccare/week-ALL children (banded)

*** NUMBER OF CURRENT CHILCARE PROVIDERS (max 5 used as noted during checks).
 *(this DV was not calculated at BC1 Sw1, only from BC1 Sw2 onwards).
 * corresponding CAPI DV in dataset is NumProv2; may or may not be correct.

frequencies ProvTyp21 ProvTyp22 ProvTyp23 ProvTyp24 ProvTyp25.
 *all active.

FREQUENCIES DvPvTy01 DvPvTy02 DvPvTy03 DvPvTy04 DvPvTy05 DvPvNo.
 *all active.

* renaming & relabelling provider types.
 RENAME VARS (ProvTyp21=MeCtya01) (ProvTyp22=MeCtyb01) (ProvTyp23=MeCtyc01)
 (ProvTyp24=MeCtyd01) (ProvTyp25=MeCtye01).

Var Labs MeCtya01 'Me - Provider 1 - type'.
 Var Labs MeCtyb01 'Me - Provider 2 - type'.
 Var Labs MeCtyc01 'Me - Provider 3 - type'.
 Var Labs MeCtyd01 'Me - Provider 4 - type'.
 Var Labs MeCtye01 'Me - Provider 5 - type'.

* new values and rearranged since previous sweep.

ADD VAL LABS MeCtya01 MeCtyb01 MeCtyc01 MeCtyd01 MeCtye01
 1 "The child's maternal grandparent(s)"
 2 "The child's paternal grandparents(s)"
 3 'Another relative'
 4 'Private creche or nursery'
 5 'Nursery class attached to primary school'
 6 'Registered childminder'
 7 'Local authority playgroup or pre-school'
 8 'Local authority creche or nursery'
 9 'Private playgroup or pre-school'
 10 'Community/Voluntary playgroup or pre-school'
 11 'My ex-spouse or partner'
 12 "The child(ren)'s older brother or sister"
 13 'A friend or neighbour'
 14 'Daily nanny who came to our house'
 15 'Live-in nanny'
 16 'Babysitter who came to our house'
 17 'Workplace creche or nursery'

18 'Family Centre'
19 'Child-carer (provided via childcare agency)'
20 'Playscheme/holiday club'
21 'Breakfast club'
22 'After-school club'
23 'Other childcare provider'
24 'None'.

FREQ MeCtya01 MeCtyb01 MeCtyc01 MeCtyd01 MeCtye01.

* Calculate total number of current provider types.

RECODE MeCtya01 MeCtyb01 MeCtyc01 MeCtyd01 MeCtye01 (1 thru 23=1) (else=0) INTO TEMP1
TEMP2 TEMP3 TEMP4 TEMP5.
COUNT DeCtot01 = TEMP1 to TEMP5 (1).
Exe.

VAR LABS DeCtot01 'De Number of ccare providers at Sweep 3'.
FORMATS DeCtot01 (F2.0).

FREQ DeCtot01.

* compared with CAPI DV NumProv2.

FREQ NumProv2.

DELETE VARS TEMP1 to TEMP5.

**** TIME SPENT IN CHILDCARE.

* (syntax adapted from BC1 Sw5 as Sw6 was more complex: split on holidays vs school term).

* renaming & relabelling first the hours & days variables:

RENAME VARS (CChrs21=MeCtma01) (CCdays21=MeCdya01) (CChrs22=MeCtmb01)
(CCdays22=MeCdya01) (CChrs23=MeCtmc01) (CCdays23=MeCdyc01) (CChrs24=MeCtmd01)
(CCdays24=MeCdya01) (CChrs25=MeCtme01) (CCdays25=MeCdye01).

Var labs MeCtma01 'Me 1st ccare provider - hours per week'.
Var labs MeCtmb01 'Me 2nd ccare provider - hours per week'.
Var labs MeCtmc01 'Me 3rd ccare provider - hours per week'.
Var labs MeCtmd01 'Me 4th ccare provider - hours per week'.
Var labs MeCtme01 'Me 5th ccare provider - hours per week'.

Var labs MeCdya01 'Me 1st ccare provider - number of days per week'.
Var labs MeCdyb01 'Me 2nd ccare provider - number of days per week'.
Var labs MeCdyc01 'Me 3rd ccare provider - number of days per week'.
Var labs MeCdyd01 'Me 4th ccare provider - number of days per week'.
Var labs MeCdye01 'Me 5th ccare provider - number of days per week'.

ADD VAL LABS MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01
MeCdya01 MeCdyb01 MeCdyc01 MeCdyd01 MeCdye01
-1 'Item not applicable' -8 "Don't know" -9 'Refusal'.

FREQ MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01 MeCdya01 MeCdyb01 MeCdyc01
MeCdyd01 MeCdye01.

RENAME VARS (CCare2=MeCany01).

VAR LABS MeCany01 'Whether resp uses regular CCare at Sweep 2'.
FREQ MeCany01.

* calculating time DV.

frequencies MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01.
missing values MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01 (lo thru -1).

Recode MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01 (1 thru 999 = copy) (else = 0) into
ReCtma01 ReCtmb01 ReCtmc01 ReCtmd01 ReCtme01.
Exe.
COMPUTE DeCtmi01 = ReCtma01 + ReCtmb01 + ReCtmc01 + ReCtmd01 + ReCtme01.
Exe.

If (MeCany01=2) DeCtmi01=-1.
Execute.
Var labels DeCtmi01 'De Nbr hrs child looked after by s/one else (avge week)'.
Val Labs DeCtmi01 -1 'Item not applicable: no childcare' 0 'Less than an hour per week'.
Missing values DeCtmi01 (0, -1).

FREQ DeCtmi01.

* MISSING VALUES DaCtmi01 (.).
* TEMP.
* SELECT IF (DaCtmi01=0).
* LIST Serial DaCtmi01 MaCtma01 MaCtmb01 MaCtmc01 MaCtmd01 MaCtme01 .
* Missing values DaCtmi01 (0, -1).

MISSING VALUES MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01 DeCtmi01().

DO IF (DeCtmi01=0).
IF (MeCtma01 =-8) OR (MeCtmb01 =-8) OR (MeCtmc01 =-8) OR (MeCtmd01 =-8) OR (MeCtme01 =-8)
DeCtmi01=-3.
END IF.
Exe.

MISSING VALUES MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 MeCtme01 (-9 thru -1).

ADD VAL LABS DeCtmi01
-3 'Information not available'.
MISSING VALUES DeCtmi01 (0, -1,-3).
FREQ DeCtmi01.

DELETE VARS ReCtma01 to ReCtme01.

**BANDED version of 'DeCtmi01' = 'DeCtmi02'
* hours are integers at DeCtmi01 so no problem about intervals for recoding below (no need to use midpoints).

Recode DeCtmi01 (1 thru 8.99=1) (9 thru 16.99=2) (17 thru 40.99 = 3) (41 thru Hi=4) (else = copy) into
DeCtmi02.
Execute.

Var labs DeCtmi02 'De Nbr hrs child looked after by s/one else in avge week-BANDED'.
Val labs DeCtmi02
-3 'Information not available'
-1 'N/A: No childcare'
0 'Less than an hour per week'
1 'Up to 8 hours'
2 'Between 9 and 16 hours'
3 'Between 17 and 40 hours'
4 'More than 40 hours'.
Missing values DeCtmi02 (0, -1,-3).
FORMATS DeCtmi02 (F2.0).

freq DeCtmi02.

* had to add decimal places to make this run correctly.

* same nbr of valid answers as DeCtmi01 OK

***TO CREATE A VARIABLE 'DeCday01' WHICH INDICATES HIGHEST NUMBER OF DAYS
CHILDCARE USED ***
*Progressing from lower count to higher count to keep only highest number of days in the variable.

Compute DeCday01 = -3.
If ANY(1, MeCdya01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 1.
Exe.
If ANY(2, MeCdya01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 2.
Exe.
If ANY(3, MeCdya01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 3.
Exe.
If ANY(4, MeCdya01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 4.
Exe.

If ANY(5, MeCday01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 5.
Exe.
If ANY(6, MeCday01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 6.
Exe.
If ANY(7, MeCday01, MeCdyb01, MeCdyc01, MeCdyd01, MeCdye01) DeCday01 = 7.
Exe.
If (MeCany01 = 2) DeCday01 = -1.
Execute.

var labs DeCday01 'De Highest nbr of days/week in any one childcare arrangmt'.
Val labs DeCday01 -1 'N/A: No childcare' -3 'Information not available'.
missing values DeCday01 (-3,-1).
FORMATS DeCday01 (F2.0).

fre DeCday01.

****TO CREATE INDIVIDUAL VARIABLES INDICATING USE OF EACH CHILDCARE TYPE**.

VAL LABS MeCtya01 MeCtyb01 MeCtyc01 MeCtyd01 MeCtye01

1 "The child's maternal grandparent(s)"
2 "The child's paternal grandparents(s)"
3 'Another relative'
4 'Private creche or nursery'
5 'Nursery class attached to primary school'
6 'Registered childminder'
7 'Local authority playgroup or pre-school'
8 'Local authority creche or nursery'
9 'Private playgroup or pre-school'
10 'Community/Voluntary playgroup or pre-school'
11 'My ex-spouse or partner'
12 "The child(ren)'s older brother or sister"
13 'A friend or neighbour'
14 'Daily nanny who came to our house'
15 'Live-in nanny'
16 'Babysitter who came to our house'
17 'Workplace creche or nursery'
18 'Family Centre'
19 'Child-carer (provided via childcare agency)'
20 'Playscheme/holiday club'
21 'Breakfast club'
22 'After-school club'
23 'Other childcare provider'
24 'None'.

*** nb take account of changes to list of childcare providers

**CURRENT USE OF either GRANDPARENTS AS CHILDCARE PROVIDERS

*1 "The child's maternal grandparent(s)"
*2 "The child's paternal grandparents(s)"

freq MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01.

Compute DeCtyp01 = 0.

If any (1, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp01 = 1.

If any (2, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp01 = 1.

If (MeCany01=2) DeCtyp01=-1.

execute.

Var labs DeCtyp01 'De Childcare from either/both grandparents'.

Val labs DeCtyp01

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp01 (-1).

FORMATS DeCtyp01 (F2.0).

fre DeCtyp01.

**CURRENT USE OF ANOTHER RELATIVE AS CHILDCARE PROVIDER

*3 'Another relative'.

Compute DeCtyp02 = 0.

If (MeCtya01=3) OR (MeCtyb01=3) OR (MeCtyc01=3) OR (MeCtyd01=3) OR (MeCtye01=3) DeCtyp02 = 1.
If (MeCany01=2) DeCtyp02=-1.
Exe.
Var labs DeCtyp02 'De Childcare from other relative'.
Val labs DeCtyp02
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp02 (-1).
FORMATS DeCtyp02 (F2.0).

fre DeCtyp02.

****CURRENT USE OF PRIVATE CRECHE/NURSERY AS CHILDCARE PROVIDER**
*4 'Private creche or nursery'

Compute DeCtyp03 = 0.
If any (4, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp03 = 1.
If (MeCany01=2) DeCtyp03=-1.
Exe.
Var labs DeCtyp03 'Da Childcare from private nursery/creche'.
Val labs DeCtyp03
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp03 (-1).
FORMATS DeCtyp03 (F2.0).

fre DeCtyp03.

****CURRENT USE OF CHILDMINDER AS CHILDCARE PROVIDER.**
* ATTENTION: watch out answer numbers for each category as there was a change between BC1 Sw1 and later sweeps, and BC2 Sw1 follows BC1 Sw1.
* childminder is answer 6, in this sweep.
*6 'Registered childminder'

Compute DeCtyp04 = 0.
If any (6, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp04 = 1.
If (MeCany01=2) DeCtyp04=-1.
Exe.
Var labs DeCtyp04 'De Childcare from childminder'.
Val labs DeCtyp04
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp04 (-1).
FORMATS DeCtyp04 (F2.0).

fre DeCtyp04.

****CURRENT USE OF LOCAL AUTHORITY PLAYGROUP AS CHILDCARE PROVIDER**
* answer 7 here, not 6 as in BC1 Sw5 syntax.
* 7 'Local authority playgroup or pre-school'

Compute DeCtyp05 = 0.
If any (7, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp05 = 1.
If (MeCany01=2) DeCtyp05=-1.
Exe.
Var labs DeCtyp05 'De Childcare from Local Auth playgroup/preschool'.
Val labs DeCtyp05 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp05 (-1).
FORMATS DeCtyp05 (F2.0).

fre DeCtyp05.

****CURRENT USE OF LOCAL AUTHORITY CRECHE/NURSERY AS CHILDCARE PROVIDER**

Compute DeCtyp06 = 0.
If any (8, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp06 = 1.

If (MeCany01=2) DeCtyp06=-1.
Exe.
Var labs DeCtyp06 'De Childcare from Local Auth nursery'.
Val labs DeCtyp06 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp06 (-1).
FORMATS DeCtyp06 (F2.0).

fre DeCtyp06.

****CURRENT USE OF Private Playgroup AS CHILDCARE PROVIDER.**

Compute DeCtyp07 = 0.
If any (9, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp07 = 1.
If (MeCany01=2) DeCtyp07=-1.
Exe.
Var labs DeCtyp07 'De Childcare from private playgroup'.
Val labs DeCtyp07 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp07 (-1).
FORMATS DeCtyp07 (F2.0).

fre DeCtyp07.

****CURRENT USE OF Community/Voluntary Playgroup AS CHILDCARE PROVIDER.**

Compute DeCtyp08 = 0.
If any (10, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp08 = 1.
If (MeCany01=2) DeCtyp08=-1.
Exe.
Var labs DeCtyp08 'De Childcare from community/vol playgroup'.
Val labs DeCtyp08 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp08 (-1).
FORMATS DeCtyp08 (F2.0).

fre DeCtyp08.

****CURRENT USE OF Ex-spouse/partner AS CHILDCARE PROVIDER**

Compute DeCtyp09 = 0.
If any (11, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp09 = 1.
If (MeCany01=2) DeCtyp09=-1.
Exe.
Var labs DeCtyp09 'De Childcare from ex partner'.
Val labs DeCtyp09 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp09 (-1).
FORMATS DeCtyp09 (F2.0).

fre DeCtyp09.

****CURRENT USE OF child's older sibling AS CHILDCARE PROVIDER.**

Compute DeCtyp10 = 0.
If any (12, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp10 = 1.
If (MeCany01=2) DeCtyp10=-1.
Exe.
Var labs DeCtyp10 'De Childcare from child's sibling'.
Val labs DeCtyp10 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp10 (-1).
FORMATS DeCtyp10 (F2.0).

fre DeCtyp10.

****CURRENT USE OF a friend or neighbour AS CHILDCARE PROVIDER**

Compute DeCtyp11 = 0.
If any (13, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp11 = 1.
If (MeCany01=2) DeCtyp11=-1.
Exe.
Var labs DeCtyp11 'De Childcare from friend/neighbour'.
Val labs DeCtyp11 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp11 (-1).
FORMATS DeCtyp11 (F2.0).

fre DeCtyp11.

****CURRENT USE OF Daily visiting nanny AS CHILDCARE PROVIDER.**

Compute DeCtyp12 = 0.

If any (14, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp12 = 1.

If (MeCany01=2) DeCtyp12=-1.

Exe.

Var labs DeCtyp12 'De Childcare from Daily nanny'.

Val labs DeCtyp12 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp12 (-1).

FORMATS DeCtyp12 (F2.0).

fre DeCtyp12.

****CURRENT USE OF live-in nanny AS CHILDCARE PROVIDER.**

Compute DeCtyp13 = 0.

If any (13, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp13 = 1.

If (MeCany01=2) DeCtyp13=-1.

Var labs DeCtyp13 'De Childcare from live-in nanny'.

Val labs DeCtyp13 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp13 (-1).

FORMATS DeCtyp13 (F2.0).

Exe.

fre DeCtyp13.

****CURRENT USE OF babysitter who comes to house AS CHILDCARE PROVIDER.**

Compute DeCtyp14 = 0.

If any (16, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp14 = 1.

If (MeCany01=2) DeCtyp14=-1.

Exe.

Var labs DeCtyp14 'De Childcare from babysitter'.

Val labs DeCtyp14 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp14 (-1).

FORMATS DeCtyp14 (F2.0).

fre DeCtyp14.

****CURRENT USE OF workplace creche/nursery AS CHILDCARE PROVIDER.**

Compute DeCtyp15 = 0.

If any (17, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp15 = 1.

If (MeCany01=2) DeCtyp15=-1.

Exe.

Var labs DeCtyp15 'De Childcare from work creche/nursery'.

Val labs DeCtyp15 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp15 (-1).

FORMATS DeCtyp15 (F2.0).

fre DeCtyp15.

****CURRENT USE OF family centre AS CHILDCARE PROVIDER.**

Compute DeCtyp16 = 0.

If any (18, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp16 = 1.

If (MeCany01=2) DeCtyp16=-1.

Exe.

Var labs DeCtyp16 'De Childcare from family centre'.

Val labs DeCtyp16 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp16 (-1).

FORMATS DeCtyp16 (F2.0).

fre DeCtyp16.

****CURRENT USE OF nursery class attached to primary school AS CHILDCARE PROVIDER.**

Compute DeCtyp17 = 0.

If any (5, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp17 = 1.

If (MeCany01=2) DeCtyp17=-1.

Exe.

Var labs DeCtyp17 'De Childcare from school nursery class'.

Val labs DeCtyp17 0 'No' 1 'Yes' -1 'Does not use childcare'.
Missing values DeCtyp17 (-1).
FORMATS DeCtyp17 (F2.0).

fre DeCtyp17.

****CURRENT USE OF childcarer (provided via childcare agency) AS CHILDCARE PROVIDER.**

Compute DeCtyp18 = 0.

If any (19, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp18 = 1.

If (MeCany01=2) DeCtyp18=-1.

Exe.

Var labs DeCtyp18 'De Childcare from agency carer'.

Val labs DeCtyp18 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp18 (-1).

FORMATS DeCtyp18 (F2.0).

fre DeCtyp18.

****CURRENT USE OF other playscheme/holiday club AS CHILDCARE PROVIDER.**

Compute DeCtyp19 = 0.

If any (20, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp19 = 1.

If (MeCany01=2) DeCtyp19=-1.

Exe.

Var labs DeCtyp19 'De Childcare from playscheme/holiday club'.

Val labs DeCtyp19 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp19 (-1).

Execute.

FORMATS DeCtyp19 (F2.0).

fre DeCtyp19.

****CURRENT USE OF Breakfast club AS CHILDCARE PROVIDER.**

Compute DeCtyp20 = 0.

If any (21, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp20 = 1.

If (MeCany01=2) DeCtyp20=-1.

Exe.

Var labs DeCtyp20 'De Childcare from Breakfast club '.

Val labs DeCtyp20 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp20 (-1).

Execute.

FORMATS DeCtyp20 (F2.0).

fre DeCtyp20.

****CURRENT USE OF After-school club AS CHILDCARE PROVIDER.**

Compute DeCtyp21 = 0.

If any (22, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp21 = 1.

If (MeCany01=2) DeCtyp21=-1.

Exe.

Var labs DeCtyp21 'De Childcare from After-school club '.

Val labs DeCtyp21 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp21 (-1).

Execute.

FORMATS DeCtyp21 (F2.0).

fre DeCtyp21.

****CURRENT USE OF other childcare provider AS CHILDCARE PROVIDER.**

Compute DeCtyp94 = 0.

If any (23, MeCtya01, MeCtyb01, MeCtyc01, MeCtyd01, MeCtye01) DeCtyp94 = 1.

If (MeCany01=2) DeCtyp94=-1.

Exe.

Var labs DeCtyp94 'De Childcare from other provider'.

Val labs DeCtyp94 0 'No' 1 'Yes' -1 'Does not use childcare'.

Missing values DeCtyp94 (-1).

Execute.

FORMATS DeCtyp94 (F2.0).

fre DeCtyp94.

*****CHILDCARE COST *****.

** renaming & relabelling payment-related raw variables.

RENAME VARS (Cpay03=MeCpay03) (Cpay04=MeCpay04) (Cpay05=MeCpay05) (Cpay06=MeCpay06)
(Cpay07=MeCpay07) (Cpay08=MeCpay08) (Cpay09=MeCpay09) (Cphr=MeCphr).

Var Labs MeCpay03 'Me Pay for childs childcare wk/month'.
Var Labs MeCpay04 'Me Amount for childs childcare per week'.
Var Labs MeCpay05 'Me Amount for childs childcare per month'.
Var Labs MeCpay06 'Me Pay for all childcare wk/month'.
Var Labs MeCpay07 'Me Amount for all childcare per week'.
Var Labs MeCpay08 'Me Amount for all childcare per month'.
Var Labs MeCpay09 'Me How easy/diff to pay for childcare'.
Var Labs MeCphr 'Me And for how many hours would that amount be?'

ADD VAL LABS MeCpay04 MeCpay05 MeCpay07 MeCpay08 MeCphr
-1 'Item not applicable'
-8 "Don't know"
-9 'Refusal'.

**COST OF CHILDCARE PER WEEK FOR COHORT CHILD.

*are they paying weekly or monthly? checkvar to ascertain which to divide amount by to get weekly cost.

FREQ MeCpay03.

Recode MeCpay03 (1 thru 2 = 1) (3 thru 4 = 0) (-1=-2) (else=copy) INTO DeCcst01.
Exe.
formats DeCcst01 (F8.0).
FREQ DeCcst01.

* attention: syntax above may lead to some incorrect values of £1 by default due to (1 thru 2 = 1) - checked further down.

* if ccare for study child and payment made, weekly or monthly (MeCpay03=1 or 2):

FREQ MeCpay04 MeCpay05.

If (MeCpay03=2) DeCcst01 = (MeCpay05*12)/52.
Exe.
If (MeCpay03=1) DeCcst01 = MeCpay04.
Exe.

FREQ DeCcst01.

MISSING VALUES MeCpay03 MeCpay04 MeCpay05 ().
IF ANY (-8, MeCpay03, MeCpay04, MeCpay05) DeCcst01=-3.
IF ANY (-9, MeCpay03, MeCpay04, MeCpay05) DeCcst01=-3.
Exe.
MISSING VALUES MeCpay03 MeCpay04 MeCpay05 (-9 thru -1).
FREQ DeCcst01.

* if no ccare MaCany01=No=2:
IF (MeCany01=2) DeCcst01=-1.
Exe.

FREQ DeCcst01.

Var labs DeCcst01 'De Cost of childcare per week - sample child'.
Val labs DeCcst01 0 'Do not pay for childcare' -1 'Do not use childcare' -3 'Information not available'.
FORMATS DeCcst01 (F8.2).
Missing values DeCcst01 (-9 thru -1, 0).
FREQ DeCcst01.

**BANDED VERSION of DaCcst01.

** answer labels equivalent to rounding so 'up to £20' includes £20.01 to £20.50, and 'from £21' starts at £20.51.

Recode DeCcst01 (0.1 thru 20.50=1) (20.51 thru 50.50=2) (50.51 thru 100.50=3) (100.51 thru hi = 4) (ELSE=Copy) into DeCcst02.

Execute.

Var labs DeCcst02 'De Cost of ccare/week-sample child (banded)'.

Val labs DeCcst02

0 'Do not pay for childcare' -1 'Do not use childcare' -3 'Information not available' 1 'Up to £20'

2 'From £21 to £50' 3 'From £51 to £100' 4 'Over £100'.

MISSING VALUES DeCcst02 (-9 thru -1,0).

FORMATS DeCcst02 (F2.0).

FREQ DeCcst02.

**COST FOR ALL CHILDCARE IN HHOLD PER WEEK.

* MeCpay06 asked only if more than one child in hhold, so need to include cost for study child as default value, to be overridden by cost for all childcare if more than 1 child in hhold.

* routing indicates a fresh start for this section of the Qre so does not take into account whether study child has CCare (MeCany01=1) or not

FREQ NumChil2 DeHGnmkd.

freq MeCpay06.

MISSING VALUES MeCpay06 ().

CRO MeCpay06 by DeHGnmkd.

MISSING VALUES MeCpay06 (-9 thru -1).

* the default value of DeCcst03 is set as the cost for the study child (if any ccare cost) as calculated in DeCcst01.

* then updated with info for the rest of the hhold (incl study child or not) given in MeCpay07 & 08 for HHolds>1 kid (based on CAPI DV NumChil2, not recalculated DeHGnmkd: so 8 N/A too Meny in MeCpay06)

FREQ MeCpay07 MeCpay08.

Recode DeCcst01(else=copy) into DeCcst03.

Exe.

If (MeCpay06=2) DeCcst03 = (MeCpay08*12)/52.

Exe.

If (MeCpay06=1) DeCcst03 = MeCpay07.

Exe.

FREQ DeCcst03.

Recode DeCcst03 (sysmis=-3) (else=copy).

Exe.

MISSING VALUES MeCpay06 ().

IF (DeHGnmkd>1) AND (MeCpay06=-1) DeCcst03=-3.

Exe.

MISSING VALUES MeCpay06 (-9 thru -1).

Var labs DeCcst03 'De Cost of childcare - ALL CHILDREN - per week'.

Val labs DeCcst03 0 'Do not pay for childcare' -1 'Do not use childcare' -3 'Information not available'.

MISSING VALUES DeCcst03 (-9 thru -1,0).

FORMATS DeCcst03 (F8.2).

fre DeCcst03.

* range £0.92 to £3825, same as for study child's childcare.

* Final nbrs = 20 'no info' (-3), 2715 'no ccare' (-1), and 1713 '0's 'do not pay for ccare'

** checking the 20 'No info' are OK

MISSING VALUES DeCcst03 ().

TEMP.

SELECT IF (DeCcst03=-3).

LIST Serial DeHGnmkd DeCcst03 DeCcst01 MeCpay06 MeCpay07 MeCpay08.

MISSING VALUES DeCcst03 (-9 thru -1,0).

MISSING VALUES DeCcst03 DeCcst01 ().
MISSING VALUES DeCcst03 DeCcst01 (-9 thru -1).
TEMP.
SELECT IF (DeCcst03<DeCcst01).
LIST Serial DeCcst01 DeCcst03 MeCpay03 MeCpay04 MeCpay05 MeCpay06 MeCpay07 MeCpay08.
MISSING VALUES DeCcst03 DeCcst01 (-9 thru -1,0).

*** banded cost of ALL childcare DeCcst04.
** as noted for banded cost of study child's ccare, answer labels equivalent to rounding, so 'up to £20'
includes £20.50
* and 'from £21' starts at £20.51, and so on)

Recode DeCcst03 (0.01 thru 20.50=1) (20.51 thru 50.50=2) (50.51 thru 100.50=3) (100.51 thru hi = 4)
(ELSE=Copy) into DeCcst04.

Execute.

Var labs DeCcst04 'De Cost of ccare/week-ALL children (banded)'.
Val labs DeCcst04 0 'Do not pay for childcare' -1 'Do not use childcare' -3 'Information not available'

1 'Up to £20' 2 'From £21 to £50' 3 'From £51 to £100' 4 'Over £100'.
Missing values DeCcst04 (0,-9 thru -1).

FORMATS DeCcst04 (F2.0).

fre DeCcst04.

fre DeCcst04.

*****DeCany01 Whether respondent using childcare for cohort child at sweep 3
*Comprised of anyone who was still using a previous provider (MeCsta01, MeCstb01...)
*New providers: Anyone who was using childcare at sw2 and has a new arrangement
(CCare3/MeCany02)
*New providers: Anyone who wasn't using at sw2 and is now (MeCany03)

**Info for up to 5 previous providers was fed forward

**Compute var

fre mecsta01 mecstb01 mecstc01 mecstd01 mecste01.

Compute DeCAny01 = 0.

If MeCsta01 < 3 or MeCstb01 < 3 OR MeCstc01 < 3 OR MeCstd01 < 3
OR MeCste01 < 3 or mecany02 = 1 or mecany03 = 1 DeCAny01= 1.

fre decany01.

*Suggests 2844 respondents (64%) were using childcare other than pre-school provider

var labs DeCany01 'De Whether respondent using childcare for cohort child at sweep 3'.
val labs decany01

0 'No'

1 'Yes'.

fre decany01.

save outfile = inprogress.

*Create a DV that provides a count of 'new' providers

recode mectya01 mectyb01 mectyc01 mectyd01 mectye01 (1 thru hi = 1) (else = 0)
(sysmis = 0) into rectya01 rectyb01 rectyc01 rectyd01 rectye01.

Count DeCnnp = rectya01 rectyb01 rectyc01 rectyd01 rectye01 (1).

var labs decnnp 'De Number of NEW childcare providers being used'.
fre decnnp.

fre decnnp.

*Create a DV that provides a count of total childcare providers - sw2 still being used plus new

Compute DeCnnp = DeCnnpv01 + DeCnnp.

If missing(decnnpv01) decnnp = decnnp.

var labs decnnp 'De Number of childcare providers currently being used'.
fre decnnp.

fre decnnp.

*Recode 'still using' variables into binaries

fre mecsta01.

Recode mecsta01 mecstb01 mecstc01 mecstd01 mecste01 (1 = 1) (2=2) (3=0) (4=0) (else = -1)
into recsta01 recstb01 recstc01 recstd01 recste01.

val labs recsta01 recstb01 recstc01 recstd01 recste01

1 'Still using - no change to hours or days'

2 'Still using - hours or days changed'

0 'No longer using'.

fre recsta01 recstb01 recstc01 recstd01 recste01.

save outfile = inprogress.

****PROVIDER TYPES****

*Check CAPI DVs on childcare provider types - make sure only 'current' childcare users are included

Compute tempdvpvck = 0.

If (dvpvty01 >0) or (dvpvty02 >0) or (dvpvty03 >0) or (dvpvty04 >0) or (dvpvty05 >0) tempdvpvck = 1.

fre tempdvpvck.

*This var only has 2729 valid cases, so there are about 115 missing from these DVs.

temp.

select if tempdvpvck = 0 and decany01 = 1.

list dvpvty01 to dvpvty05 mecany02 mecany03.

*Looks like they have missed new providers amongst those with previous/existing arrangements (MeCany03)

*which have been recorded at MeCtya01 etc. Check in MeCtya01.

temp.

select if tempdvpvck = 0 and decany01 = 1.

list mecany03 dvpvty01 dvpvty02 dvpvty03 dvpvty04 dvpvty05 mectya01 mectyb01 mectyc01.

*Yep. Almost all are accounted for by use of a new provider. And for those cases dvpvty01 is always missing.

*Suggest computing dvpvty01 and 02 to eq mectya01 and tyb01.

fre dvpvty01 dvpvty02.

If missing(dvpvty01) and mectya01 > 0 dvpvty01 = mectya01.

If missing(dvpvty02) and mectyb01 > 0 dvpvty02 = mectyb01.

EXECUTE.

**Check coverage of these vars again

Compute tempdvpvck = 0.

If (dvpvty01 >0) or (dvpvty02 >0) or (dvpvty03 >0) or (dvpvty04 >0) or (dvpvty05 >0) tempdvpvck = 1.

fre tempdvpvck.

*Now have 2832 valid cases against 2844 who should have something, only 12 missing from these DVs.

*Possible some people said they used childcare but didn't give details

temp.

select if tempdvpvck = 0 and decany01 = 1.

list mecsta01 mecstb01 mecstc01 mecstd01 mecste01.

*They are all cases using a previous provider.

*Double check.

temp.

select if tempdvpvck = 0 and decany01 = 1.

list mecsta01 mecstb01 dvpvty01 dvpvty02.

*Yep. Details of the provider still being used should be sitting in dvpvty01 and/or dvpvty02 but isn't

*This needs to be retrieved from feed forward which has been removed from this dataset.

**NOTE: There are some cases where the FF details have been taken from sweep 1 - where the case wasn't achieved

**at sweep 2. Need to get time and days info for these cases from there.

sort cases by serial.

***Use the age 3 data to correct the missing information in dvpvty vars

temp.

select if missing(dvpvty01) and mecsta01 < 3.
list mecsta01 dcctya01.

If missing(dvpvty01) and mecsta01 < 3 dvpvty01 = dcctya01.
If missing(dvpvty01) and mecsta01 > 2 and mecstb01 < 3 dvpvty01 = dcctyb01.
If dvpvty01 > 0 and missing(dvpvty02) and mecstb01 > 2 dvpvty02 = dcctyb01.
EXECUTE.

**Check coverage of these vars again

Compute tempdvpvck = 0.

If (dvpvty01 > 0) or (dvpvty02 > 0) or (dvpvty03 > 0) or (dvpvty04 > 0) or (dvpvty05 > 0) tempdvpvck = 1.
fre tempdvpvck.

*Now have 2844 valid cases against 2844 who should have something - yay!

temp.

select if missing(dvpvty01) and decany01 = 1.
*No cases selected.

**To rename the provider type details

Rename vars (DvPvTy01 = DeCtya01) (DvPvTy02 = DeCtyb01) (DvPvTy03 = DeCtyc01)
(DvPvTy04 = DeCtyd01) (DvPvTy05 = DeCtye01).

****CHECK THAT DeCtyb01 and DeCtyc01 etc variables have captured provider details
****in all valid cases

***Variable to check how many previous providers from sweep 2 a respondent was still using at sweep 3
***(this variable is used later to derive current provider details)

fre mecsta01.

Recode mecsta01 mecstb01 mecstc01 mecstd01 mecste01 (1 thru 2 = 1) (3 thru 4 = 0) (else = copy)
Into Rmecsta01 RStlCar2 RStlCar3 RStlCar4 RStlCar5.

Count DeCnpv01 = Rmecsta01 RStlCar2 RStlCar3 RStlCar4 RStlCar5 (1).

If (dccany01 = 0) DeCnpv01 = -1.

Var labs DeCnpv01 'Number of sw2 providers still being used '.

Missing values DeCnpv01 (-1).

val labs DeCnpv01

-1 'Didnt use childcare at sweep 2'.

fre DeCnpv01 .

*Check that all cases with a 2nd provider type in dectyb01 are using at least 2 providers

temp.

select if dectyb01 > 0 and decncp < 2.
list recsta01 recstb01 recstc01 recstd01 recste01.

*1 case where there is info in dectyc01 and less than 2 providers being used

*****Check that all cases with details of 2 providers have a provider type in dectyb01

temp.

select if missing(dectyb01) and decncp > 1.
list recsta01 recstb01 recstc01 recstd01 recste01.

*19 cases where provider type 2 data is missing

****Correct the 19 cases where provider type is missing.

Missing values DeCtyb01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtyb01
DcCtme01 DcCtmd01 DcCtme01 meCcta01 meCctb01 meCCdc01 meCctd01 meCcte01
mectma01 mectyb01 mectmc01 mectmd01 mectme01().

Do if missing(dectyb01) and decncp > 1.
 If (recsta01 > 0) and (recstb01 > 0) DeCtyb01 = DcCtyb01.
 If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) DeCtyb01 = DcCtmc01.
 If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) DeCtyb01 = DcCtmd01.
 If (recsta01 = 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCtyb01 = DcCtme01.
 If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) DeCtyb01 = DcCtmc01.
 If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) DeCtyb01 = DcCtmd01.
 If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 > 0) DeCtyb01 = DcCtme01.
 If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) DeCtyb01 = DcCtmd01.
 If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 > 0) DeCtyb01 = DcCtme01.
 If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 > 0) DeCtyb01 = DcCtme01.

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = -1) AND (MeCany03 = 1) AND (MeCtyb01 >= 0) DeCtyb01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (MeCany03 = 1) AND (MeCtyb01 > 0) DeCtyb01 = MeCtyb01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCtyb01 = MeCtya01.

If ((DeCnpv01 = 0) or missing(decnpv01)) and mecany03 = 1 DeCtyb01 = MeCtyb01.
 If (recsta01 = 1) and (recstb01 = 1) and missing(Dcctyb01) and deffwds = 1 DeCtyb01 = mactyb01.
 If (recsta01 = 1) and (recstb01 = 1) and missing(dcctyb01) dectyb01 = -8.
 If (MeCany02 = 1) DeCtyb01 = MeCtyb01.
 end if.

Missing values DeCtyb01 (-1, -8, 0).
 EXECUTE.

temp.
 select if missing(dectyb01) and decncp > 1.
 list recsta01 recstb01 recstc01 recstd01 recste01.
 *0 cases where provider type 2 data is missing

get file = inprogress.

*****Check that all cases with a 3rd provider type in dectyc01 are using at least 3rd providers

temp.
 select if dectyc01 > 0 and decncp < 3.
 list recsta01 recstb01 recstc01 recstd01 recste01.
 *No cases where there is info in dectyc01 and less than 3 providers being used

*Check that all cases with details of 3 providers have a provider type in dectyc01

temp.

select if missing(dectyc01) and decncp > 2.

list recsta01 recstb01 recstc01 recstd01 recste01.

*40 cases where there is info on at least 3 providers but no 3rd provider type is being used.

****Correct the 40 cases where provider type is missing.

*If still using at least 3 providers from previous sweep

do if missing(dectyc01) and decncp > 2.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) DeCtyc01 = DcCtyc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 > 0) DeCtyc01 = DcCtyd01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) DeCtyc01 = DcCtyd01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) DeCtyc01 = DcCtyd01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 > 0) DeCtyc01 = DcCtye01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 > 0) DeCtyc01 = DcCtye01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 > 0) DeCtyc01 = DcCtye01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 > 0) DeCtyc01 = DcCtye01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 > 0) DeCtyc01 = DcCtye01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 > 0) DeCtyc01 = DcCtye01.

*If using 2 providers from previous sweep and has at least 1 new providers

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1)

DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)

DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)

DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 > 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND

(MeCany03 = 1) DeCtyc01 = MeCtya01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCtyc01 = MeCtya01.

*If using 1 provider from previous sweep and has at least 2 new providers

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCtyb01 > 0) DeCtyc01 = MeCtyb01.
If (recsta01 > 0) AND (recstb01 = 0) AND (MeCtyb01 > 0) DeCtyc01 = MeCtyb01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCtyb01 > 0) DeCtyc01 = MeCtyb01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCtyb01 > 0 = 1) DeCtyc01 = MeCtyb01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtyb01 > 0 = 1) DeCtyc01 = MeCtyb01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCtyb01 > 0 = 1) DeCtyc01 = MeCtyb01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtyb01 > 0 = 1) DeCtyc01 = MeCtyb01.

*If no providers from previous sweep and at least three new providers

If (recsta01 = 0) AND (recstb01 = -1) AND (MeCtyc01 > 0) DeCtyc01 = MeCtyc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCtyc01 > 0) DeCtyc01 = MeCtyc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtyc01 > 0) DeCtyc01 = MeCtyc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCtyc01 > 0) DeCtyc01 = MeCtyc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyc01 > 0) DeCtyc01 = MeCtyc01.
If (recsta01 = -1) AND (MeCtyc01 > 0 = 1) DeCtyc01 = MeCtyc01.

**Additional syntax to pick up sw1 FF data etc

If ((DeCnpv01 = 0) or missing(decnpv01)) and mecany03 = 1 DeCtyc01 = MeCtyc01.
If (reCsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and missing(Dcctyc01) and deffwdsw = 1 DeCtyc01 = mactyc01.
If (recsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and missing(dcctyc01) dectyc01 = -8.

If (MeCany02 = 1) DeCtyc01 = MeCtyc01.
If (DeCtyc01 = -1) DeCtyc01 = -1.
end if.

Missing values DeCtyc01 (-1, -8, 0).
val labs DeCtyc01
-8 'Dont know'.
EXECUTE.

temp.
select if missing(dectyc01) and decncp > 2.
list recsta01 recstb01 recstc01 recstd01 recste01.
*FIXED! 0 cases where there is info on at least 3 providers but no 3rd provider type is being used.

*****Check that all cases with a 4th provider type in dectyd01 are using at least 4 providers

temp.
select if dectyd01 > 0 and decncp < 4.
list recsta01 recstb01 recstc01 recstd01 recste01.
*No cases where there is info in dectyd01 and less than 4 providers being used

*Check that all cases with details of 3 providers have a provider type in dectyc01

temp.
select if missing(dectyd01) and decncp > 3.
list recsta01 recstb01 recstc01 recstd01 recste01.
*7 cases where there is info on at least 4 providers but no 4th provider type is being used according to dectyd01.

****Correct the 7 cases where provider type D is missing.

Do if missing(dectyd01) and decncp > 3.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCtyb01 > 0) DeCtyd01 = MeCtyb01.

*If using 1 providers from previous sweep and has at least 3 new providers

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtyc01 > 0) DeCtyd01 = MeCtyc01.

*If no providers from previous sweep and at least four new providers

If ((decnpv01 = 0) or missing(decnpv01)) AND (MeCtyd01 > 0) DeCtyd01 = MeCtyd01.
 If (recsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and (recstd01 = 1) and missing(dcctyd01) dectyd01 = -8.
 If (reCsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and (recstd01 = 1) and missing(Dcctyd01) and deffwdsw = 1 DeCtyd01 = mactyd01.
 If (MeCany02 = 1) DeCtyd01 = MeCtyd01.
 end if.

Missing values DeCtyd01 (-1, -8, 0).

val labs DeCtyd01

-8 'Dont know'.

EXECUTE.

temp.

select if missing(dectyd01) and decncp > 3.

list recsta01 recstb01 recstc01 recstd01 recste01.
*0 cases where there is info on at least 4 providers but no 4th provider type is being used according to dectyd01.
*FIXED!

*****Check that all cases with a 5th provider type in dectye01 are using at least 5 providers

temp.
select if dectye01 > 0 and decncp < 5.
list recsta01 recstb01 recstc01 recstd01 recste01.
*No cases where there is info in dectye01 and less than 5 providers being used

*Check that all cases with details of 5 providers have a provider type in dectye01

temp.
select if missing(dectye01) and decncp > 4.
list recsta01 recstb01 recstc01 recstd01 recste01.
*1 cases where there is some info on at least 5 providers but no 5th provider type is being used according to dectye01.

***Correct the 1 case where provider type E is missing. ***TO FOLLOW**

Do if missing(dectye01) and decncp > 4.
If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCtyc01 > 0) DeCtye01 = MeCtyc01.
end if.

exe.

temp.
select if missing(dectye01) and decncp > 4.
list recsta01 recstb01 recstc01 recstd01 recste01.
**FIXED

missing values dectya01 dectyb01 dectyc01 dectyd01 dectye01 (-1).
fre dectya01 dectyb01 dectyc01 dectyd01 dectye01 dvpvty06.

save outfile = inprogress.

***Number of childcare providers being used at sweep 3 - rename existing variable

***USE THIS VARIABLE FOR NUMBER OF PROVIDERS
***Not DvPvNo as this sometimes counts providers no longer being used

fre DeCtya01 DeCtyb01 DeCtyc01 DeCtyd01 DeCtye01 .

Recode DeCtya01 DeCtyb01 DeCtyc01 DeCtyd01 DeCtye01 (1 thru 24 =1) (Else = Copy) Into rDeCtya01 rDeCtyb01 rDeCtyc01 rDeCtyd01 rDeCtye01 .

Count DeCtot02 = rDeCtya01 rDeCtyb01 rDeCtyc01 rDeCtyd01 rDeCtye01 (1).
If (DeCany01 = 2) DeCtot02 = -1.
Var labs DeCtot02 'Number of childcare providers being used' .
Missing values DeCtot02 (0, -1).
val labs DeCtot02
-1 'Not using childcare at sweep 3'.

fre Decany01 DeCtot02.
*Numbers match

**To derive the number of hours in care of each childcare provider

fre DeCtya01 DeCtyb01 DeCtyc01 DeCtyd01 DeCtye01 .

**Number of hours for Provider A - i.e. that provider which is detailed in DeCtya01.

temp.
select if missing(mecsta01) and mecstb01 > 0.
list mecsta01 mecstb01 mecstc01 mecstd01 mecste01.
*Later provider details are not present if earlier providers are 'missing'

Missing values DeCtya01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtmb01
DcCtme01 DcCtmd01 DcCtme01 meCcta01 meCctb01 meCCdc01 meCctd01 meCcte01
mectma01 mectmb01 mectmc01 mectmd01 mectme01().

If (mecsta01 = 1) DeCtma01 = DcCtma01.
If (mecsta01 = 2) DeCtma01 = meCcta01.
exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 1) DeCtma01 = DcCtmb01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 2) DeCtma01 = meCctb01.
exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 1) DeCtma01 = DcCtmc01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 2) DeCtma01 = meCCtc01.
exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 1)
DeCtma01 = DcCtmd01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 2)
DeCtma01 = meCctd01.
exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 3) AND
(mecste01 = 1) DeCtma01 = DcCtme01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 3) AND
(mecste01 = 2) DeCtma01 = meCcte01.
exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = -1) AND (meCany03 = 1) DeCtma01 =
meCtma01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = -1) AND (meCany03 = 1)
DeCtma01 = meCtma01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = -1) AND
(meCany03 = 1) DeCtma01 = meCtma01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 3) AND
(mecste01 = -1)
AND (meCany03 = 1) DeCtma01 = meCtma01.
exe.

If ((mecsta01 = -1) or (mecsta01 = 3) or (mecsta01 = 4)) and mecany03 = 1 dectma01 = mectma01.
exe.

If (meCany02 = 1) DeCtma01 = meCtma01.
exe.

If MeCsta01 = 1 and missing(Dcctma01) and deffwdsw = 1 DeCtma01 = mactma01.
If MeCsta01 = 1 and missing(Dcctma01) and deffwdsw = 2 DeCtma01 = -8.
If (DeCtya01 = -1) DeCtma01 = -1.
Var labs DeCtma01 'No of hours child looked after per week by provider 1'.
Missing values DeCtma01 (-1, -8, 0).
val labs DeCtma01
-8 'Dont know'.
EXECUTE.

missing values DeCtya01 (-1).

fre DeCtya01 DeCtma01 .

*2841 have at least one provider and we have time info for 2834. Plus 4 DKs (2838) and 3 zero hours =
*2841!

**Number of days looked after by 1st childcare provider

**NOTE: The 'days' syntax is just the 'hours' syntax but with the end variables changed from e.g.

**DeCtma01 to DeCdya01 and MeCtma01 to MeCdma01

Missing values DeCtya01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtmb01
DcCtme01 DcCtmd01 DcCtme01 meCcda01 meCcdb01 meCCdc01 meCcdd01 meCcde01
mecdya01 mecdyb01 mecdyc01 mecdyd01 mecdye01().

If (mecsta01 = 1) DeCdya01 = DcCdya01.

If (mecsta01 = 2) DeCdya01 = meCcda01.

exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 1) DeCdya01 = DcCdyb01.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 2) DeCdya01 = meCcdb01.

exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 1) DeCdya01 = DcCdyc01.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 2) DeCdya01 = meCCdc01.

exe.

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 1)

DeCdya01 = DcCdyd01.

```

If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 2)
DeCdy01 = meCdd01.
exe.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 3) AND
(mecste01 = 1) DeCdy01 = DcCdy01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 3) AND
(mecste01 = 2) DeCdy01 = meCde01.
exe.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = -1) AND (meCany03 = 1) DeCdy01 = meCdy01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = -1) AND (meCany03 = 1)
DeCdy01 = meCdy01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = -1) AND
(meCany03 = 1) DeCdy01 = meCdy01.
If ((mecsta01 = 3) or (mecsta01 = 4)) AND (mecstb01 = 3) AND (mecstc01 = 3) AND (mecstd01 = 3) AND
(mecste01 = -1)
AND (meCany03 = 1) DeCdy01 = meCdy01.
exe.
If ((mecsta01 = -1) or (mecsta01 = 3) or (mecsta01 = 4)) and mecany03 = 1 DeCdy01 = meCdy01.
exe.
If (meCany02 = 1) DeCdy01 = meCdy01.
exe.
If MeCsta01 = 1 and missing(Dccdy01) and deffwds = 1 mecdy01 = macdy01.
If MeCsta01 = 1 and missing(Dccdy01) and deffwds = 2 mecdy01 = -8.
If (DeCtya01 = -1) DeCdy01 = -1.
Var labs DeCdy01 'Sw3 No of days child looked after per week by provider 1'.
Missing values DeCdy01 (-1, -8, 0).
val labs DeCdy01
-8 'Dont know'.
EXECUTE.

```

missing values dectya01 (-1).

fre DeCtya01 DeCdy01 .

*2841 have at least one provider and we have days info for 2827, plus 4 'Don't Knows' = 2831

*and 3 cases where the days value is 0 = 2834 and 7 sysmis where FF data is missing = 2841

temp.

select if DeCtma01 > 0 and missing(decdy01).

list dectma01 deCdy01.

*****Number of hours for Provider B - i.e. that provider which is detailed in DeCtyb01.

fre dectyb01.

*1291 cases with a 2nd provider

**When checking the syntax, it appeared that some cases had a 2nd provider type recorded

**but shouldn't have because they didn't actually have a 2nd provider. The syntax below corrects

**the 2nd provider type variable, setting incorrect cases to -1.

**Think this may be related to CAPI DV syntax picking up former arrangements now used for pre-school

If DeCnpv01 = 0 and mecany03 = 1 and mectmb01 = -1 dectyb01 = -1.

*Not using any previous providers, report only one 'new' provider

If DeCnpv01 < 2 and mecany03 = 2 dectyb01 = -1.

*Using only one previous providers, not using any new providers

Missing values DeCtyb01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtmb01

DcCtme01 DcCtmd01 DcCtme01 meCcta01 meCctb01 meCCdc01 meCctd01 meCcte01

mectma01 mectmb01 mectmc01 mectmd01 mectme01().

If (recsta01 > 0) and (recstb01 = 1) DeCtmb01 = DcCtmb01.

If (recsta01 > 0) and (recstb01 = 2) DeCtmb01 = MeCctb01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 1) DeCtmb01 = DcCtmc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 2) DeCtmb01 = MeCctc01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 1) DeCtmb01 = DcCtmd01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 2) DeCtmb01 = MeCctd01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCtmb01 = DcCtme01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCtmb01 = MeCcte01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 1) DeCtmb01 = DcCtmc01.

```

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 2) DeCtmb01 = MeCctc01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 = 1) DeCtmb01 = DcCtmd01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 = 2) DeCtmb01 = MeCctd01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 1) DeCtmb01
= DcCtme01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 2) DeCtmb01
= MeCcte01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 1) DeCtmb01 = DcCtmd01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 2) DeCtmb01 = MeCctd01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 1) DeCtmb01
= DcCtme01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 2) DeCtmb01
= MeCcte01.
If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 1) DeCtmb01
= DcCtme01.
If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 2) DeCtmb01
= MeCcte01.
exe.
If (recsta01 > 0) AND (recstb01 = -1) AND (MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = -1) AND (MeCany03 = 1) AND (MeCtmb01 >= 0) DeCtmb01 =
MeCtmb01.
If (recsta01 = 0) AND (recstb01 = 0) AND (MeCany03 = 1) AND (MeCtmb01 > 0) DeCtmb01 = MeCtmb01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmb01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCany03 = 1) DeCtmb01 = MeCtma01.
exe.
If ((DeCnpv01 = 0) or missing(decnpv01)) and mecany03 = 1 DeCtmb01 = MeCtmb01.
If (recsta01 = 1) and (recstb01 = 1) and missing(Dcctmb01) and deffwdsw = 1 DeCtmb01 = mactmb01.
If (recsta01 = 1) and (recstb01 = 1) and missing(dcctmb01) dectmb01 = -8.
If (MeCany02 = 1) DeCtmb01 = MeCtmb01.
exe.
If (DeCtyb01 = -1) DeCtmb01 = -1.
Var labs DeCtmb01 'Sw3 No of hours child looked after per week by provider 2'.
Missing values DeCtmb01 (-1, -8, 0).
val labs DeCtmb01
-8 'Dont know'.
EXECUTE.

missing values DeCtyb01 (-1).
fre DeCtyb01 DeCtmb01 .
*1291 have a second provider type and we have time info for 1281 plus 6 DKs (1287), 3 Zeros (1290), 1
sysmis = 1291
*Sysmis explained by a DK recorded for 'whether using any new providers' - so info not captured

TEMPORARY.
missing value dectmb01 ().
select if dectyb01 > 0 and missing(dectmb01).
list recsta01 recstb01 recstc01 DeCnpv01 dcctmb01 dectyb01 dcctmb01 mecany02 mecany03 MeCtmb01
dectmb01.

```


Var labs DeCdyb01 'Sw3 No of days child looked after per week by provider 2'.
Missing values DeCdyb01 (-1, -8, 0).
val labs DeCdyb01
-8 'Dont know'.
EXECUTE.

missing values DeCtyb01 (-1).
fre DeCtyb01 DeCdyb01 .
*1291 have a second provider and we have time info for 1280 plus 6 don't knows (1286)
*4 'zeros' (1290) and 1 sysmis (1291).

*****Number of hours for Provider C - i.e. that provider which is detailed in DcCtyc01.

fre dectyc01.
*382 cases with a 3rd provider

Missing values DeCtya01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtmb01
DcCtmc01 DcCtmd01
DcCtme01 meCcta01 meCctb01 meCctc01 meCctd01 meCcte01 MeCtmb01 ().

*If still using at least 3 providers from previous sweep

delete vars dectmc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 1) DeCtmc01 = DcCtmc01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 2) DeCtmc01 = MeCctc01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 1) DeCtmc01 = DcCtmd01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 2) DeCtmc01 = MeCctd01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 1) DeCtmc01 = DcCtmd01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 2) DeCtmc01 = MeCctd01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 1) DeCtmc01 = DcCtmd01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 2) DeCtmc01 = MeCctd01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 1) DeCtmc01
= DcCtme01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 2) DeCtmc01
= MeCcte01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 1) DeCtmc01
= DcCtme01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 2) DeCtmc01
= MeCcte01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCtmc01
= DcCtme01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCtmc01
= MeCcte01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 1) DeCtmc01
= DcCtme01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 2) DeCtmc01
= MeCcte01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCtmc01
= DcCtme01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCtmc01
= MeCcte01.
If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 1) DeCtmc01
= DcCtme01.
If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 2) DeCtmc01
= MeCcte01.
exe.

*If using 2 providers from previous sweep and has at least 1 new providers

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmc01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmc01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmc01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmc01 = MeCtma01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCtmc01 = MeCtma01.
 exe.

*If using 1 provider from previous sweep and has at least 2 new providers

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCtmb01 >= 0) DeCtmc01 = MeCtmb01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (MeCtmb01 > 0) DeCtmc01 = MeCtmb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCtmb01 > 0) DeCtmc01 = MeCtmb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCtmb01 > 0) DeCtmc01 = MeCtmb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtmb01 > 0) DeCtmc01 = MeCtmb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCtmb01 > 0) DeCtmc01 = MeCtmb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtmb01 > 0) DeCtmc01 = MeCtmb01.
 exe.

*If no providers from previous sweep and at least three new providers

If (recsta01 = 0) AND (recstb01 = -1) AND (MeCtmc01 > 0) DeCtmc01 = MeCtmc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCtmc01 > 0) DeCtmc01 = MeCtmc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtmc01 > 0) DeCtmc01 = MeCtmc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCtmc01 > 0) DeCtmc01 = MeCtmc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtmc01 > 0) DeCtmc01 = MeCtmc01.
 If (recsta01 = -1) AND (MeCtmc01 > 0) DeCtmc01 = MeCtmc01.
 exe.

**TO DO Add the additional syntax and adjustments to the end of the big chunk, e.g. to pick up sw1 FF data

If ((DeCnpv01 = 0) or missing(decnpv01)) and mecany03 = 1 DeCtmc01 = MeCtmc01.
 If (reCsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and missing(Dccdyc01) and deffwdsw = 1 DeCdyc01 = macdyc01.
 If (recsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and missing(dccdyc01) decdyc01 = -8.

If (MeCany02 = 1) DeCtmc01 = MeCtmc01.
 If (DeCtyc01 = -1) DeCtmc01 = -1.

exe.

Var labs DeCtmc01 'No of hours child looked after per week by provider 3'.

Missing values DeCtmc01 (-1, -8, 0).

val labs DeCtmc01

-8 'Dont know'.

EXECUTE.

missing values DeCtyc01 (-1).

fre DeCtyc01 DeCtmc01 .

*382 have a third provider and we have time info for 380 plus 2 zeros

save outfile = inprogress.

**Number of DAYS for Provider C - i.e. that provider which is detailed in DeCtyc01.

**NOTE: The 'days' syntax is just the 'hours' syntax but with the end variables changed from e.g.

**DeCtma01 to DcCdya01 and McCtma01 to McCdma01

*If still using at least 3 providers from previous sweep

delete vars DeCdyc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 = 0) and (recste01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCdyc01 = MeCcadc01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 1) DeCdyc01 = DcCdyc01.

If (recsta01 = 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 2) DeCdyc01 = MeCcadc01.

*If using 2 providers from previous sweep and has at least 1 new providers

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCany03 = 1)

DeCdyc01 = MeCdya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)

DeCdyc01 = MeCdya01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)

DeCdyc01 = MeCdya01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCdyc01 = MeCdya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND

(MeCany03 = 1) DeCdyc01 = MeCdya01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCany03 = 1) DeCdyc01 = MeCdya01.

*If using 1 provider from previous sweep and has at least 2 new providers

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCdyb01 >= 0) DeCdyc01 = MeCdyb01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (MeCdyb01 > 0) DeCdyc01 = MeCdyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCdyb01 > 0) DeCdyc01 = MeCdyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCdyb01 > 0) DeCdyc01 = MeCdyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCdyb01 > 0) DeCdyc01 = MeCdyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCdyb01 > 0) DeCdyc01 = MeCdyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCdyb01 > 0) DeCdyc01 = MeCdyb01.

*If no providers from previous sweep and at least three new providers

If (recsta01 = 0) AND (recstb01 = -1) AND (MeCtmc01 > 0) DeCdyc01 = MeCdyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCdyc01 > 0) DeCdyc01 = MeCdyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCdyc01 > 0) DeCdyc01 = MeCdyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCdyc01 > 0) DeCdyc01 = MeCdyc01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCdyc01 > 0) DeCdyc01 = MeCdyc01.
 If (recsta01 = -1) AND (MeCdyc01 > 0) DeCdyc01 = MeCdyc01.

**TO DO Add the additional syntax and adjustments to the end of the big chunk, e.g. to pick up sw1 FF data

If ((DeCnpv01 = 0) or missing(decnpv01)) and mecany03 = 1 DeCdyc01 = MeCdyc01.
 If (reCsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and missing(dccdyc01) and deffwdsw = 1 DeCdyc01 = macedyc01.
 If (recsta01 = 1) and (recstb01 = 1) and (recstc01 = 1) and missing(dccdyc01) DeCdyc01 = -8.

If (MeCany02 = 1) DeCdyc01 = MeCdyc01.
 If (DeCtyc01 = -1) DeCdyc01 = -1.
 exe.

Var labs DeCdyc01 'No of days child looked after per week by provider 3'.
 Missing values DeCdyc01 (-1, -8, 0).
 val labs DeCdyc01

-8 'Dont know'.
EXECUTE.

missing values DeCtyc01 (-1).
fre DeCtyc01 DeCdyc01 .
*382 have a third provider and we have day info for 379 plus 2 zeros

temp.
select if dectyc01>0 and missing(decdyc01).
list mecany02 mecany03 recsta01 recstb01 recstc01 recstd01 DcCdyc01 maCdyc01 decdyc01.
*There is a rogue sysmis value where there should be FF data

*****Number of hours for Provider D - i.e. that provider which is detailed in DcCtyd01.

fre dectyd01.
*48 cases with a 4th provider

Missing values DeCtyd01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtmb01
DcCtmc01 DcCtmd01
DcCtme01 meCcta01 meCctb01 meCctc01 meCctd01 meCcte01 MeCtmb01 MeCtmd01 ().

*If still using at least 4 providers from previous sweep

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 1) DeCtmd01 = DcCtmd01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 2) DeCtmd01 = MeCctd01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 1) DeCtmd01
= DcCtme01.
If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 2) DeCtmd01
= MeCcte01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 1) DeCtmd01 =
DcCtme01.
If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 2) DeCtmd01
= MeCcte01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCtmd01
= DcCtme01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCtmd01
= MeCcte01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 1) DeCtmd01
= DcCtme01.
If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 0) and (recste01 = 2) DeCtmd01
= MeCcte01.

*If using 3 providers from previous sweep and has at least 1 new provider

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCany03 = 1)
DeCtmd01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 > 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCany03 = 1) DeCtmd01 = MeCtma01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

*If no providers from previous sweep and at least four new providers

If (recsta01 = -1) AND ((MeCtmd01 > 0) or (MeCtmd01 = -8)) DeCtmd01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = -1) AND (MeCtmd01 > 0) DeCtmd01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCtmd01 > 0) DeCtmd01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtmd01 > 0) DeCtmd01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCtmd01 > 0) DeCtmd01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCtmc01 > 0) DeCtmd01 = MeCtmc01.

If (DeCtyd01 = -1) DeCtmd01 = -1.

Var labs DeCtmd01 'No of hours child looked after per week by provider 4'.

Missing values DeCtmd01 (-1, -8, 0).

val labs DeCtmd01

-8 'Dont know'.

EXECUTE.

missing values DeCtyd01 (-1).

fre DeCtyd01 DeCtmd01 .

*48 have a third provider and we have time info for 47 plus 1 don't know

temp.

select if dectyd01 > 0 and missing(dectmd01).

list mecany02 mecany03 recsta01 recstb01 recstc01 recstd01 MeCtma01 MeCtmb01 MeCtmc01 MeCtmd01 dectmd01.

save outfile = inprogress.

**Number of DAYS for Provider D - i.e. that provider which is detailed in DcCtyd01.

**NOTE: The 'days' syntax is just the 'hours' syntax but with the end variables changed from e.g.

**DcCtma01 to DcCdyd01 and MaCtma01 to MaCdmd01

fre dectyd01.

*48 cases with a 4th provider

*If still using at least 4 providers from previous sweep

delete vars dectyd01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 1) DeCdyd01= DcCdyd01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 = 2) DeCdyd01= MeCcdd01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 1) DeCdyd01= DcCdye01.

If (recsta01 = 0) and (recstb01 > 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 2) DeCdyd01= MeCcde01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 1) DeCdyd01= DcCdye01.

If (recsta01 > 0) and (recstb01 = 0) and (recstc01 > 0) and (recstd01 > 0) and (recste01 = 2) DeCdyd01= MeCcde01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 1) DeCdyd01= DcCdye01.

If (recsta01 > 0) and (recstb01 > 0) and (recstc01 = 0) and (recstd01 > 0) and (recste01 = 2) DeCdyd01= MeCcde01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND (MeCdyb01 > 0) DeCdyd01= MeCdyb01.

*If using 1 providers from previous sweep and has at least 3 new providers

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.

*If no providers from previous sweep and at least four new providers

If (recsta01 = -1) AND ((MeCdyd01 > 0) or (MeCdyd01 = -8)) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.
 If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND (MeCdyd01 > 0) DeCdyd01= MeCdyd01.

If (DeCtyd01 = -1) DeCdyd01 = -1.
 Var labs DeCdyd01 'No of days child looked after per week by provider 4'.
 Missing values DeCdyd01 (-1, -8, 0).
 val labs DeCdyd01
 -8 'Dont know'.
 EXECUTE.

missing values DeCtyd01 (-1).
 fre DeCtyd01 DeCdyd01 .
 *48 have a fourth provider and we have time info for 48

*****Number of hours for Provider E - i.e. that provider which is detailed in DeCtye01.

fre DeCtye01.

*4 cases with a 5th provider

TEMPORARY.

select if DeCtye01 > 0.

list mecany02 mecany03 recsta01 recstb01 recstc01 recstd01 recste01 MeCtma01 MeCtmb01 MeCtmc01
MeCtmd01 MeCtme01.

*No-one using any more than 2 previous providers

Missing values DeCtye01 mecsta01 mecstb01 mecstc01 mecstd01 mecste01 DcCtma01 DcCtmb01
DcCtmc01 DcCtmd01
DcCtme01 meCcta01 meCctb01 meCctc01 meCctd01 meCcte01 MeCtmb01 MeCtmd01 ().

*If using 2 providers from previous sweep and has at least 3 new providers

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCtmc01 > 0)
DeCtme01 = MeCtmc01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCtmc01 > 0)
DeCtme01 = MeCtmc01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.
If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 > 0) AND
(MeCtmc01 > 0) DeCtme01 = MeCtmc01.

*If using 1 provider from previous sweep and has at least 4 new providers

If (recsta01 > 0) AND (recstb01 = -1) AND (MeCtmd01 > 0) DeCtme01 = MeCtmd01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCtmd01 > 0) DeCtme01 = MeCtmd01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCtmd01 > 0) DeCtme01 = MeCtmd01.
If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtmd01 > 0)
DeCtme01 = MeCtmd01.
If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtmd01 > 0)
DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCtmd01 > 0)
DeCtme01 = MeCtmd01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = 0) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 > 0) AND
(MeCtmd01 > 0) DeCtme01 = MeCtmd01.

*If no provider from previous sweep and at least five new providers

If (recsta01 = -1) AND (MeCtme01 > 0) DeCtme01 = MeCtme01.

If (recsta01 = 0) AND (recstb01 = -1) AND (MeCtme01 > 0) DeCtme01 = MeCtme01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = -1) AND (MeCtme01 > 0) DeCtme01 = MeCtme01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = -1) AND (MeCtme01 > 0)
DeCtme01 = MeCtme01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCtme01 > 0) DeCtme01 = MeCtme01.

If (recsta01 = 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = 0) AND
(MeCtme01 > 0) DeCtme01 = MeCtme01.

If (DeCtme01 = -1) DeCtme01 = -1.

Var labs DeCtme01 'No of hours child looked after per week by provider 5'.

Missing values DeCtme01 (-1, -8, 0).

val labs DeCtme01

-8 'Dont know'.

EXECUTE.

missing values DeCtme01 (-1).

fre DeCtme01 DeCtme01 .

*4 have a 5th provider and we have days info for 4

missing values DeCtme01 ().

If (DeCtme01 = -1) DeCtme01 = -1.

exe.

*****NUMBER OF DAYS FOR PROVIDER E

*If using 2 providers from previous sweep and has at least 3 new providers

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = -1) AND (MeCdyc01 > 0) DeCdye01 = MeCdyc01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCdyc01 > 0)

DeCdye01 = MeCdyc01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = -1) AND (MeCdyc01 > 0)

DeCdye01 = MeCdyc01.

If (recsta01 > 0) AND (recstb01 > 0) AND (recstc01 = 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCdyc01 > 0) DeCdye01 = MeCdyc01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCdyc01 > 0) DeCdye01 = MeCdyc01.

If (recsta01 > 0) AND (recstb01 = 0) AND (recstc01 = 0) AND (recstd01 > 0) AND (recste01 = -1) AND
(MeCdyc01 > 0) DeCdye01 = MeCdyc01.

If (recsta01 = 0) AND (recstb01 > 0) AND (recstc01 > 0) AND (recstd01 = 0) AND (recste01 = -1) AND
(MeCdyc01 > 0) DeCdye01 = MeCdyc01.


```
If (DeCtye01 = -1) DeCdye01 = -1.
Var labs DeCdye01 'No of days child looked after per week by provider 5'.
Missing values DeCdye01 (-1, -8, 0).
val labs DeCdye01
-8 'Dont know'.
EXECUTE.
```

```
missing values DeCtye01 (-1).
fre DeCtye01 DeCdye01 .
*4 have a 5th provider and we have days info for 4
```

```
save outfile = inprogress.
```

```
*****I think there is a sixth provider
```

```
temp.
select if decncp = 6.
list mecany02 mecany03 recsta01 recstb01 recstc01 recstd01 recste01 MeCtyf01 MeCtmd01 MeCtme01.
*2 prev providers, 4 new providers so information is stored in MeCtyd01, MeCtmd01 and MeCdyd01.
```

```
Compute DeCtyf01 = -1.
Compute DeCtmf01 = -1.
Compute DeCdyf01 = -1.
If decncp = 6 DeCtyf01 = MeCtyd01.
If decncp = 6 DeCtmf01 = MeCtmd01.
If decncp = 6 DeCdyf01 = MeCdyd01.
exe.
```

```
var labs DeCtyf01 'De Provider Type 6'.
Var labs DeCtmf01 'No of hours child looked after per week by provider 6'.
Var labs DeCdyf01 'No of days child looked after per week by provider 6'.
missing values dectyf01 DeCtmf01 DeCdyf01 (-1).
```

```
val labs dectyf01
1 'The child(ren s) grandparents'
2 'Child s maternal grandparents'
3 'Child s paternal grandparents'
4 'Another relative'
5 'Private creche or nursery'
6 'Nursery class attached to primary school'
7 'Childminder'
8 'Local Authority playgroup or pre-school'
9 'Local Authority creche or nursery'
10 'Private playgroup or pre-school'
11 'Community/Voluntary playgroup or pre-school'
12 'My ex-spouse or partner'
13 'The child(ren s) older brother or sister'
14 'A friend or neighbour'
15 'Daily nanny who came to our house'
16 'Live-in nanny'
17 'Babysitter who came to our house'
18 'Workplace creche or nursery'
19 'Family Centre'
20 'Child-carer (provided via childcare agency)'
21 'Playscheme / summer / holiday club'
22 'Breakfast club'
24 'Other childcare provider (PLEASE SPECIFY)'.
```

```
fre dectyf01 DeCtmf01 DeCdyf01.
```

```
*****Renaming and relabelling to match previous sweeps/cohort
```

```
var labs DeCtya01 'Sweep 3 childcare provider A: provider type'.
var labs DeCtma01 'Sw3 provider A: no. of hours used per week'.
var labs DeCdy01 'Sw3 provider A: no. of days used per week'.
var labs DeCtyb01 'Sweep 3 childcare provider B: provider type'.
var labs DeCtmb01 'Sw3 provider B: no. of hours used per week'.
var labs DeCdyb01 'Sw3 provider B: no. of days used per week'.
```

```

var labs DeCtyc01      'Sweep 3 childcare provider C: provider type'.
var labs DeCtmc01     'Sw3 provider C: no. of hours used per week'.
var labs DeCdyc01     'Sw3 provider C: no. of days used per week'.
var labs DeCtyd01     'Sweep 3 childcare provider D: provider type'.
var labs DeCtmd01     'Sw3 provider D: no. of hours used per week'.
var labs DeCdyd01     'Sw3 provider D: no. of days used per week'.
var labs DeCtye01     'Sweep 3 childcare provider E: provider type'.
var labs DeCtme01     'Sw3 provider E: no. of hours used per week'.
var labs DeCdye01     'Sw3 provider E: no. of days used per week'.
var labs DeCtyf01     'Sweep 3 childcare provider F: provider type'.
var labs DeCtmf01     'Sw3 provider F: no. of hours used per week'.
var labs DeCdyf01     'Sw3 provider F: no. of days used per week'.

```

missing values DeCtya01 DeCtyb01 DeCtyc01 DeCtyd01 DeCtye01 DeCtyf01 (-1, -8).
fre DeCtya01 DeCtyb01 DeCtyc01 DeCtyd01 DeCtye01 DeCtyf01.

**Creating DVs of use of specific childcare provider types for comparison with BC1

**NOTE: Single person using Provider F is Breakfast Club (code 22) so only needs

**added there.

fre DeCtya01.

Compute DeCtyp01 = 0.

If (DeCtya01=1) or (DeCtyb01=1) or (DeCtyc01=1) or (DeCtyd01=1) or (DeCtye01=1) or
(DeCtya01=2) or (DeCtyb01=2) or (DeCtyc01=2) or (DeCtyd01=2) or (DeCtye01=2) or
(DeCtya01=3) or (DeCtyb01=3) or (DeCtyc01=3) or (DeCtyd01=3) or (DeCtye01=3) DeCtyp01 = 1.

If (DeCany01=0) DeCtyp01 = -1.

var labs DeCtyp01 'De Childcare from grandparents'.

Val labs DeCtyp01

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp01 (-1).

FORMATS DeCtyp01 (F2.0).

fre DeCtyp01.

Compute DeCtyp02 = 0.

If (DeCtya01=4) or (DeCtyb01=4) or (DeCtyc01=4) or (DeCtyd01=4) or (DeCtye01=4) DeCtyp02 = 1.

If (DeCany01=0) DeCtyp02 = -1.

var labs DeCtyp02 'De Childcare from other relative'.

Val labs DeCtyp02

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp02 (-1).

FORMATS DeCtyp02 (F2.0).

fre DeCtyp02.

Compute DeCtyp03 = 0.

If (DeCtya01=5) or (DeCtyb01=5) or (DeCtyc01=5) or (DeCtyd01=5) or (DeCtye01=5) DeCtyp03 = 1.

If (DeCany01=0) DeCtyp03 = -1.

var labs DeCtyp03 'De Childcare from private nursery/creche'.

Val labs DeCtyp03

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp03 (-1).

FORMATS DeCtyp03 (F2.0).

fre DeCtyp03.

Compute DeCtyp04 = 0.

If (DeCtya01=7) or (DeCtyb01=7) or (DeCtyc01=7) or (DeCtyd01=7) or (DeCtye01=7) DeCtyp04 = 1.

If (DeCany01=0) DeCtyp04 = -1.

var labs DeCtyp04 'De Childcare from childminder'.

Val labs DeCtyp04

0 'No'

1 'Yes'

-1 'Does not use childcare'.
Missing values DeCtyp04 (-1).
FORMATS DeCtyp04 (F2.0).

fre DeCtyp04.

Compute DeCtyp05 = 0.
If (DeCtya01=8) or (DeCtyb01=8) or (DeCtyc01=8) or (DeCtyd01=8) or (DeCtye01=8) DeCtyp05 = 1.
If (DeCany01=0) DeCtyp05 =-1.
var labs DeCtyp05 'De Childcare from Local Authority playgroup or pre-school'.
Val labs DeCtyp05
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp05 (-1).
FORMATS DeCtyp05 (F2.0).

fre DeCtyp05.

Compute DeCtyp06 = 0.
If (DeCtya01=9) or (DeCtyb01=9) or (DeCtyc01=9) or (DeCtyd01=9) or (DeCtye01=9) DeCtyp06 = 1.
If (DeCany01=0) DeCtyp06 =-1.
var labs DeCtyp06 'De Childcare from Local Authority creche or nursery'.
Val labs DeCtyp06
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp06 (-1).
FORMATS DeCtyp06 (F2.0).

fre DeCtyp06.

Compute DeCtyp07 = 0.
If (DeCtya01=10) or (DeCtyb01=10) or (DeCtyc01=10) or (DeCtyd01=10) or (DeCtye01=10) DeCtyp07 = 1.
If (DeCany01=0) DeCtyp07 =-1.
var labs DeCtyp07 'De Childcare from Private playgroup or pre-school'.
Val labs DeCtyp07
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp07 (-1).
FORMATS DeCtyp07 (F2.0).

fre DeCtyp07.

Compute DeCtyp08 = 0.
If (DeCtya01=11) or (DeCtyb01=11) or (DeCtyc01=11) or (DeCtyd01=11) or (DeCtye01=11) DeCtyp08 = 1.
If (DeCany01=0) DeCtyp08 =-1.
var labs DeCtyp08 'De Childcare from Community/Voluntary playgroup or pre-school'.
Val labs DeCtyp08
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp08 (-1).
FORMATS DeCtyp08 (F2.0).

fre DeCtyp08.

Compute DeCtyp09 = 0.
If (DeCtya01=12) or (DeCtyb01=12) or (DeCtyc01=12) or (DeCtyd01=12) or (DeCtye01=12) DeCtyp09 = 1.
If (DeCany01=0) DeCtyp09 =-1.
var labs DeCtyp09 'De Childcare from ex-spouse or partner'.
Val labs DeCtyp09
0 'No'
1 'Yes'
-1 'Does not use childcare'.
Missing values DeCtyp09 (-1).

FORMATS DeCtyp09 (F2.0).

fre DeCtyp09.

Compute DeCtyp10 = 0.

If (DeCtya01=13) or (DeCtyb01=13) or (DeCtyc01=13) or (DeCtyd01=13) or (DeCtye01=13) DeCtyp10 = 1.

If (DeCany01=0) DeCtyp10 =-1.

var labs DeCtyp10 'De Childcare from the childs older sibling'.

Val labs DeCtyp10

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp10 (-1).

FORMATS DeCtyp10 (F2.0).

fre DeCtyp10.

Compute DeCtyp11 = 0.

If (DeCtya01=14) or (DeCtyb01=14) or (DeCtyc01=14) or (DeCtyd01=14) or (DeCtye01=14) DeCtyp11 = 1.

If (DeCany01=0) DeCtyp11 =-1.

var labs DeCtyp11 'De Childcare from A friend or neighbour'.

Val labs DeCtyp11

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp11 (-1).

FORMATS DeCtyp11 (F2.0).

fre DeCtyp11.

Compute DeCtyp12 = 0.

If (DeCtya01=15) or (DeCtyb01=15) or (DeCtyc01=15) or (DeCtyd01=15) or (DeCtye01=15) DeCtyp12 = 1.

If (DeCany01=0) DeCtyp12 =-1.

var labs DeCtyp12 'De Childcare from Daily nanny who came to our house'.

Val labs DeCtyp02

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp12 (-1).

FORMATS DeCtyp12 (F2.0).

fre DeCtyp12.

Compute DeCtyp13 = 0.

If (DeCtya01=16) or (DeCtyb01=16) or (DeCtyc01=16) or (DeCtyd01=16) or (DeCtye01=16) DeCtyp13 = 1.

If (DeCany01=0) DeCtyp13 =-1.

var labs DeCtyp13 'De Childcare from Live-in nanny'.

Val labs DeCtyp13

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp13 (-1).

FORMATS DeCtyp13 (F2.0).

fre DeCtyp13.

Compute DeCtyp14 = 0.

If (DeCtya01=17) or (DeCtyb01=17) or (DeCtyc01=17) or (DeCtyd01=17) or (DeCtye01=17) DeCtyp14 = 1.

If (DeCany01=0) DeCtyp14 =-1.

var labs DeCtyp14 'De Childcare from Babysitter'.

Val labs DeCtyp14

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp14 (-1).

FORMATS DeCtyp14 (F2.0).

fre DeCtyp14.

Compute DeCtyp15 = 0.

If (DeCtya01=18) or (DeCtyb01=18) or (DeCtyc01=18) or (DeCtyd01=18) or (DeCtye01=18) DeCtyp15 = 1.

If (DeCany01=0) DeCtyp15 =-1.

var labs DeCtyp15 'De Childcare from Workplace creche or nursery'.

Val labs DeCtyp15

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp15 (-1).

FORMATS DeCtyp15 (F2.0).

fre DeCtyp15.

Compute DeCtyp16 = 0.

If (DeCtya01=19) or (DeCtyb01=19) or (DeCtyc01=19) or (DeCtyd01=19) or (DeCtye01=19) DeCtyp16 = 1.

If (DeCany01=0) DeCtyp16 =-1.

var labs DeCtyp16 'De Childcare from Family centre'.

Val labs DeCtyp16

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp16 (-1).

FORMATS DeCtyp16 (F2.0).

fre DeCtyp16.

Compute DeCtyp17 = 0.

If (DeCtya01=6) or (DeCtyb01=6) or (DeCtyc01=6) or (DeCtyd01=6) or (DeCtye01=6) DeCtyp17 = 1.

If (DeCany01=0) DeCtyp17 =-1.

var labs DeCtyp17 'De Childcare from nursery class attached to primary school'.

Val labs DeCtyp17

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp17 (-1).

FORMATS DeCtyp17 (F2.0).

fre DeCtyp17.

Compute DeCtyp18 = 0.

If (DeCtya01=20) or (DeCtyb01=20) or (DeCtyc01=20) or (DeCtyd01=20) or (DeCtye01=20) DeCtyp18 = 1.

If (DeCany01=0) DeCtyp18 =-1.

var labs DeCtyp18 'De Childcare from agency carer'.

Val labs DeCtyp18

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp18 (-1).

FORMATS DeCtyp18 (F2.0).

fre DeCtyp18.

Compute DeCtyp19 = 0.

If (DeCtya01=24) or (DeCtyb01=24) or (DeCtyc01=24) or (DeCtyd01=24) or (DeCtye01=24) DeCtyp19 = 1.

If (DeCany01=0) DeCtyp19 =-1.

var labs DeCtyp19 'De Childcare from other provider'.

Val labs DeCtyp19

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp19 (-1).

FORMATS DeCtyp19 (F2.0).

fre DeCtyp19.

Compute DeCtyp24 = 0.

If (DeCtya01=21) or (DeCtyb01=21) or (DeCtyc01=21) or (DeCtyd01=21) or (DeCtye01=21) DeCtyp24 = 1.

If (DeCany01=0) DeCtyp24 =-1.

var labs DeCtyp24 'De Childcare from playscheme/summer club'.

Val labs DeCtyp24

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp24 (-1).

FORMATS DeCtyp24 (F2.0).

fre DeCtyp24.

Compute DeCtyp25 = 0.

If (DeCtya01=22) or (DeCtyb01=22) or (DeCtyc01=22) or (DeCtyd01=22) or (DeCtye01=22) or (DeCtyf01 = 22) DeCtyp25 = 1.

If (DeCany01=0) DeCtyp25 =-1.

var labs DeCtyp25 'De Childcare from Breakfast club'.

Val labs DeCtyp25

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp25 (-1).

FORMATS DeCtyp25 (F2.0).

fre DeCtyp25.

**Use of other informal

**Other informal - another relative(2), ex-spouse/partner (9), older sibling (10),

**friend/neighbour(11), babysitter (14)

Compute DeCtyp20 = 0.

If any (1, DeCtyp02, DeCtyp09, DeCtyp10, DeCtyp11, DeCtyp14) DeCtyp20 = 1.

If (DeCany01=0) DeCtyp20 =-1.

Var labs DeCtyp20 'De Childcare from other informal sources'.

Val labs DeCtyp20

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp20 (-1).

FORMATS DeCtyp20 (F2.0).

execute.

fre DeCtyp20.

**Use of nursery or creche

**Nursery or creche - private nursery(3), local authority nursery(6), workplace nursery(15),

**nursery class attached to primary school (17)

Compute DeCtyp21 = 0.

If any (1, DeCtyp03, DeCtyp06, DeCtyp15, DeCtyp17) DeCtyp21 = 1.

If (DeCany01=0) DeCtyp21 =-1.

Var labs DeCtyp21 'De Childcare from nursery or creche'.

Val labs DeCtyp21

0 'No'

1 'Yes'

-1 'Does not use childcare'.

Missing values DeCtyp21 (-1).

FORMATS DeCtyp21 (F2.0).

execute.

fre DeCtyp21.

save outfile = inprogress.

***MISC CORRECTIONS

fre DeCtya01 DeCtyb01 DeCtyc01 DeCtyd01 DeCtye01 DeCtyf01.
fre DeCtma01 DeCtmb01 DeCtmc01 DeCtmd01 DeCtme01 DeCtmf01.
fre DeCdya01 DeCdyb01 DeCdyc01 DeCdyd01 DeCdye01 DeCdyf01.

val labs dectyc01 dectyd01
1 'The child(ren s) grandparents'
2 'Child s maternal grandparents'
3 'Child s paternal grandparents'
4 'Another relative'
5 'Private creche or nursery'
6 'Nursery class attached to primary school'
7 'Childminder'
8 'Local Authority playgroup or pre-school'
9 'Local Authority creche or nursery'
10 'Private playgroup or pre-school'
11 'Community/Voluntary playgroup or pre-school'
12 'My ex-spouse or partner'
13 'The child(ren s) older brother or sister'
14 'A friend or neighbour'
15 'Daily nanny who came to our house'
16 'Live-in nanny'
17 'Babysitter who came to our house'
18 'Workplace creche or nursery'
19 'Family Centre'
20 'Child-carer (provided via childcare agency)'
21 'Playscheme / summer / holiday club'
22 'Breakfast club'
24 'Other childcare provider (PLEASE SPECIFY)'.
EXECUTE.

Primary School

** no DV syntax

Child Health & Development

freq DAccAny.

* freq Daccb Daccb2 Daccb3 Daccb4 Daccb5 Haca32 Haca33 Haca34 Haca35 Haca36.

**** DV for BANDED NUMBER OF ACCIDENTS *** for analysis only: this DV has never been kept for archived dataset*.

FREQ DAccAny.

RENAME VARS (DAccAny=MeHac).
VAR LABS MeHac 'Me Number of accidents/injuries'.

RECODE MeHac (0 = 0) (1=1) (2 thru hi = 2) (else=copy) INTO DeHacbd.
variable labels DeHacbd 'De Banded number of accidents'.
value labels DeHacbd 0 'None' 1 'One' 2 'Two or more' -1 'Item not applicable' -8 "Don't know" -9 'Refusal'.
formats DeHacbd (F8.0).
MISSING VALUES DeHacbd (-9 thru -1).
fre DeHacbd.

**** DV for BANDED AGE OF CHILD WHEN ACCIDENT HAPPENED *** for analysis only: this DV has not kept for archived dataset*

* FREQ Daccb Daccb2 Daccb3.

* RENAME VARS (Daccb=MeHaca31) (Daccb2=MeHacb31) (Daccb3=MeHacc31)
(Daccb4=MeHacd31) (Daccb5=MeHace31).

* VAR LABS MeHaca31 'Me Age in months at accident 1'.
* VAR LABS MeHacb31 'Me Age in months at accident 2'.
* VAR LABS MeHacc31 'Me Age in months at accident 3'.
* VAR LABS MeHacd31 'Me Age in months at accident 4'.
* VAR LABS MeHace31 'Me Age in months at accident 5'.

* ADD VAL LABS MeHaca31 MeHacb31 MeHacc31 MeHacd31 MeHace31
-1 'Item not applicable'
-8 "Don't know"
-9 'Refusal'.

* freq MeHaca31 MeHacb31 MeHacc31 MeHacd31 MeHace31.

* original syntax in 'DVs for Health & Dev Sweep 1.sps' for 5 sets of accident vars, as details collected for EACH accident the child has***
also "(98 = 98) (99 = 99)" using old NK & REF codes (replaced with else=copy) and missings val labs adjusted.

* recode MeHaca31 (0 thru 6 = 1) (7 thru 12 =2) (13 thru 18 = 3) (19 thru 24 = 4) (25 thru 35 = 5)
(36 thru 97 = 5) (else=copy) INTO DeHacag1.
* variable labels DeHacag1 'De Banded age of child when 1st accident happened'.
* value labels DeHacag1 1 '0-6 months' 2 '7-12 months' 3 '13-18 months' 4 '19-24 months' 5 '25-35 months'
6 '36 months and over' -9 'Refused' -8 "Don't Know" -1 'Item not applicable'.
* MISSING VALUES DeHacag1 (-9 thru -1).
* fre DeHacag1.

* recode MeHacb31 (0 thru 6 = 1) (7 thru 12 =2) (13 thru 18 = 3) (19 thru 24 = 4) (25 thru 35 = 5)
(36 thru 97 = 5) (else=copy) INTO DeHacag2.
* variable labels DeHacag2 'De Banded age of child when 2nd accident happened'.
* value labels DeHacag2 1 '0-6 months' 2 '7-12 months' 3 '13-18 months' 4 '19-24 months' 5 '25-35 months'
6 '36 months and over' -9 'Refused' -8 "Don't Know" -1 'Item not applicable'.
* MISSING VALUES DeHacag2 (-9 thru -1).
* fre DeHacag2.

* recode MeHacc31 (0 thru 6 = 1) (7 thru 12 =2) (13 thru 18 = 3) (19 thru 24 = 4) (25 thru 35 = 5)
(36 thru 97 = 5) (else=copy) INTO DeHacag3.
* variable labels DeHacag3 'De Banded age of child when 3rd accident happened'.
* value labels DeHacag3 1 '0-6 months' 2 '7-12 months' 3 '13-18 months' 4 '19-24 months' 5 '25-35 months'
6 '36 months and over' -9 'Refused' -8 "Don't Know" -1 'Item not applicable'.
* MISSING VALUES DeHacag3 (-9 thru -1).
* fre DeHacag3.

* recode MeHacd31 (0 thru 6 = 1) (7 thru 12 =2) (13 thru 18 = 3) (19 thru 24 = 4) (25 thru 35 = 5)
(36 thru 97 = 5) (else=copy) INTO DeHacag4.
* variable labels DeHacag4 'De Banded age of child when 4th accident happened'.
* value labels DeHacag4 1 '0-6 months' 2 '7-12 months' 3 '13-18 months' 4 '19-24 months' 5 '25-35 months'
6 '36 months and over' -9 'Refused' -8 "Don't Know" -1 'Item not applicable'.
* MISSING VALUES DeHacag4 (-9 thru -1).
* fre DeHacag4.

* recode MeHace31 (0 thru 6 = 1) (7 thru 12 =2) (13 thru 18 = 3) (19 thru 24 = 4) (25 thru 35 = 5)
(36 thru 97 = 5) (else=copy) INTO DeHacag5.
* variable labels DeHacag5 'De Banded age of child when 5th accident happened'.
* value labels DeHacag5 1 '0-6 months' 2 '7-12 months' 3 '13-18 months' 4 '19-24 months' 5 '25-35 months'
6 '36 months and over' -9 'Refused' -8 "Don't Know" -1 'Item not applicable'.
* MISSING VALUES DeHacag5 (-9 thru -1).
* fre DeHacag5.

* formats DeHacag1 DeHacag2 DeHacag3 DeHacag4 DeHacag5 (f8.0).
* freq DeHacag1 DeHacag2 DeHacag3 DeHacag4 DeHacag5.

* checking that age in months at accident is not higher than RECALCULATED child's age in months DaHGagC.

* TEMP.
* SELECT IF (MeHaca31>DdHGagC) OR (MeHac31>DdHGagC) OR (MeHacc31>DdHGagC)
OR (MeHacd31>DdHGagC) OR (MeHace31>DdHGagC) .
* LIST Serial x_DdHGagC DeHacbd DeHGagC MeHaca31 MeHac31 MeHacc31 MeHacd31 MeHace31
BabyAge BabyAge2 BabyAge3.

***** HEALTH & DEVELOPMENT *****

**** ADDENDUM: DV re BANDED NUMBER OF HOSPITALISATIONS for analysis only; was not archived.
* ('DlnHsp' in original BC1 Sw1 syntax is 'Hhsp01' in current dataset').

* FREQ Hhsp01.

* RENAME VARS (Hhsp01=MaHhsp01).

* VAR LABS MaHhsp01 'Ma Freq child went to hospital for health prb'.

* no missings in original count var so no need to add val labs.

* recode MaHhsp01 (0 = 0) (1=1) (2 thru 97 = 2) (else=copy) INTO DaHhsp01.

* variable labels DaHhsp01 'Da Banded number of hospital admissions for illness or health prob'.

* value labels DaHhsp01 0 'None' 1 'One' 2 'Two or more' -9 'Refused' -8 "Don't Know" -1 'Item not applicable'.

* MISSING VALUES DaHhsp01 (-9 thru -1).

* fre DaHhsp01.

*** Changing multi-coded into new single variables for each illness a, b, c

frequencies PvProb Hlsi201 Hlsi202 Hlsi203 Hlsi204 Hlsi205 Hlsi206 Hlsi207 Hlsi208 Hlsi209 Hlsi30.
frequencies PvProb2 Hlsi210 Hlsi211 Hlsi212 Hlsi213 Hlsi214 Hlsi215 Hlsi216 Hlsi217 Hlsi218 Hlsi31.
frequencies PvProb3 Hlsi219 Hlsi220 Hlsi221 Hlsi222 Hlsi223 Hlsi224 Hlsi225 Hlsi226 Hlsi227 Hlsi32.
frequencies PvProb4 Hlsi228 Hlsi229 Hlsi230 Hlsi231 Hlsi232 Hlsi233 Hlsi234 Hlsi235 Hlsi236 Hlsi33.

*no illnesses from 4 onwards, keep filter, drop subsequent variables.

frequencies DisProbC DisPrbM DisPrbM2.
missing values DisProbC DisPrbM DisPrbM2 ().

**

RECODE DisProbC (1=0) (else=-1) INTO MeHlsa01.

Exe.

FREQ MeHlsa01.

IF (XDisPrb1=1) MeHlsa01=1.

Exe.

IF (XDisPrb2=1) MeHlsa01=2.

Exe.

IF (XDisPrb3=1) MeHlsa01=3.

Exe.

IF (XDisPrb4=1) MeHlsa01=4.

Exe.

IF (XDisPrb5=1) MeHlsa01=5.

Exe.

IF (XDisPrb6=1) MeHlsa01=6.

Exe.

IF (XDisPrb7=1) MeHlsa01=7.

Exe.

IF (XDisPrb8=1) MeHlsa01=8.

Exe.

IF (XDisPrb9=1) MeHlsa01=9.

Exe.

IF (XDisPrb10=1) MeHlsa01=10.

Exe.

IF (XDisPrb11=1) MeHlsa01=11.

Exe.

IF (XDisPrb12=1) MeHlsa01=12.

Exe.

IF (XDisPrb13=1) MeHlsa01=13.
Exe.
IF (XDisPrb14=1) MeHlsa01=14.
Exe.
IF (XDisPrb15=1) MeHlsa01=15.
Exe.
IF (XDisPrb16=1) MeHlsa01=16.
Exe.
IF (XDisPrb17=1) MeHlsa01=17.
Exe.
IF (XDisPrb18=1) MeHlsa01=18.
Exe.
IF (XDisPrb19=1) MeHlsa01=19.
Exe.
IF (XDisPrb20=1) MeHlsa01=20.
Exe.
IF (XDisPrb21=1) MeHlsa01=21.
Exe.
IF (XDisPrb22=1) MeHlsa01=22.
Exe.
IF (XDisPrb23=1) MeHlsa01=23.
Exe.
IF (XDisPrb24=1) MeHlsa01=24.
Exe.
IF (XDisPrb25=1) MeHlsa01=25.
Exe.
IF (XDisPrb26=1) MeHlsa01=26.
Exe.
IF (XDisPrb27=1) MeHlsa01=27.
Exe.
IF (XDisPrb28=1) MeHlsa01=28.
Exe.
IF (XDisPrb29=1) MeHlsa01=29.
Exe.
IF (XDisPrb30=1) MeHlsa01=30.
Exe.
IF (XDisPrb31=1) MeHlsa01=31.
Exe.
IF (XDisPrb32=1) MeHlsa01=32.
Exe.
IF (XDisPrb33=1) MeHlsa01=33.
Exe.
IF (XDisPrb34=1) MeHlsa01=34.
Exe.
IF (XDisPrb35=1) MeHlsa01=35.
Exe.
IF (XDisPrb36=1) MeHlsa01=36.
Exe.
IF (XDisPrb37=1) MeHlsa01=37.
Exe.
IF (XDisPrb38=1) MeHlsa01=38.
Exe.
IF (XDisPrb39=1) MeHlsa01=39.
Exe.
IF (XDisPrb40=1) MeHlsa01=40.
Exe.
IF (XDisPrb41=1) MeHlsa01=41.
Exe.
IF (XDisPrb94=1) MeHlsa01=42.
Exe.

FREQ MeHlsa01.

* 2nd series of illnesses : setting base and filling default values

RECODE DisPrbM (1=0) (else=-1) INTO MeHlsb01.

Exe.

FREQ MeHlsb01.

IF (XDisPrb42=1) MeHlsb01=1.

Exe.
IF (XDisPrb43=1) MeHlsb01=2.
Exe.
IF (XDisPrb44=1) MeHlsb01=3.
Exe.
IF (XDisPrb45=1) MeHlsb01=4.
Exe.
IF (XDisPrb46=1) MeHlsb01=5.
Exe.
IF (XDisPrb47=1) MeHlsb01=6.
Exe.
IF (XDisPrb48=1) MeHlsb01=7.
Exe.
IF (XDisPrb49=1) MeHlsb01=8.
Exe.
IF (XDisPrb50=1) MeHlsb01=9.
Exe.
IF (XDisPrb51=1) MeHlsb01=10.
Exe.
IF (XDisPrb52=1) MeHlsb01=11.
Exe.
IF (XDisPrb53=1) MeHlsb01=12.
Exe.
IF (XDisPrb54=1) MeHlsb01=13.
Exe.
IF (XDisPrb55=1) MeHlsb01=14.
Exe.
IF (XDisPrb56=1) MeHlsb01=15.
Exe.
IF (XDisPrb57=1) MeHlsb01=16.
Exe.
IF (XDisPrb58=1) MeHlsb01=17.
Exe.
IF (XDisPrb59=1) MeHlsb01=18.
Exe.
IF (XDisPrb60=1) MeHlsb01=19.
Exe.
IF (XDisPrb61=1) MeHlsb01=20.
Exe.
IF (XDisPrb62=1) MeHlsb01=21.
Exe.
IF (XDisPrb63=1) MeHlsb01=22.
Exe.
IF (XDisPrb64=1) MeHlsb01=23.
Exe.
IF (XDisPrb65=1) MeHlsb01=24.
Exe.
IF (XDisPrb66=1) MeHlsb01=25.
Exe.
IF (XDisPrb67=1) MeHlsb01=26.
Exe.
IF (XDisPrb68=1) MeHlsb01=27.
Exe.
IF (XDisPrb69=1) MeHlsb01=28.
Exe.
IF (XDisPrb70=1) MeHlsb01=29.
Exe.
IF (XDisPrb71=1) MeHlsb01=30.
exe.
IF (XDisPrb72=1) MeHlsb01=31.
Exe.
IF (XDisPrb73=1) MeHlsb01=32.
Exe.
IF (XDisPrb74=1) MeHlsb01=33.
Exe.
IF (XDisPrb75=1) MeHlsb01=34.
Exe.
IF (XDisPrb76=1) MeHlsb01=35.
Exe.
IF (XDisPrb77=1) MeHlsb01=36.

Exe.
IF (XDisPrb78=1) MeHlsb01=37.
Exe.
IF (XDisPrb79=1) MeHlsb01=38.
Exe.
IF (XDisPrb80=1) MeHlsb01=39.
Exe.
IF (XDisPrb81=1) MeHlsb01=40.
Exe.
IF (XDisPrb82=1) MeHlsb01=41.
Exe.
IF (XDisPrb97=1) MeHlsb01=42.
exe.

FREQ MeHlsb01.

* 3rd series of illnesses : setting base and filling default values

* caution: nbr gap between XDisPrb93 and XDPrb100 because nbrs used before for 'Other'/'Vague' etc

RECODE DisPrbM2 (1=0) (else=-1) INTO MeHlsc01.

Exe.
FREQ MeHlsc01.

IF (XDisPrb83=1) MeHlsc01=1.
Exe.
IF (XDisPrb84=1) MeHlsc01=2.
Exe.
IF (XDisPrb85=1) MeHlsc01=3.
Exe.
IF (XDisPrb86=1) MeHlsc01=4.
Exe.
IF (XDisPrb87=1) MeHlsc01=5.
Exe.
IF (XDisPrb88=1) MeHlsc01=6.
Exe.
IF (XDisPrb89=1) MeHlsc01=7.
Exe.
IF (XDisPrb90=1) MeHlsc01=8.
Exe.
IF (XDisPrb91=1) MeHlsc01=9.
Exe.
IF (XDisPrb92=1) MeHlsc01=10.
Exe.
IF (XDisPrb93=1) MeHlsc01=11.
Exe.
IF (XDisPrb100=1) MeHlsc01=12.
Exe.
IF (XDisPrb101=1) MeHlsc01=13.
Exe.
IF (XDisPrb102=1) MeHlsc01=14.
Exe.
IF (XDisPrb103=1) MeHlsc01=15.
Exe.
IF (XDisPrb104=1) MeHlsc01=16.
Exe.
IF (XDisPrb105=1) MeHlsc01=17.
Exe.
IF (XDisPrb106=1) MeHlsc01=18.
Exe.
IF (XDisPrb107=1) MeHlsc01=19.
Exe.
IF (XDisPrb108=1) MeHlsc01=20.
Exe.
IF (XDisPrb109=1) MeHlsc01=21.
Exe.
IF (XDisPrb110=1) MeHlsc01=22.
Exe.
IF (XDisPrb111=1) MeHlsc01=23.
Exe.
IF (XDisPrb112=1) MeHlsc01=24.

Exe.
IF (XDisPrb113=1) MeHlsc01=25.
Exe.
IF (XDisPrb114=1) MeHlsc01=26.
Exe.
IF (XDisPrb115=1) MeHlsc01=27.
Exe.
IF (XDisPrb116=1) MeHlsc01=28.
Exe.
IF (XDisPrb117=1) MeHlsc01=29.
Exe.
IF (XDisPrb118=1) MeHlsc01=30.
Exe.
IF (XDisPrb119=1) MeHlsc01=31.
Exe.
IF (XDisPrb120=1) MeHlsc01=32.
Exe.
IF (XDisPrb121=1) MeHlsc01=33.
Exe.
IF (XDisPrb122=1) MeHlsc01=34.
Exe.
IF (XDisPrb123=1) MeHlsc01=35.
Exe.
IF (XDisPrb124=1) MeHlsc01=36.
Exe.
IF (XDisPrb125=1) MeHlsc01=37.
Exe.
IF (XDisPrb126=1) MeHlsc01=38.
Exe.
IF (XDisPrb127=1) MeHlsc01=39.
Exe.
IF (XDisPrb128=1) MeHlsc01=40.
Exe.
IF (XDisPrb129=1) MeHlsc01=41.
Exe.
IF (XDisPrb130=1) MeHlsc01=42.
exe.

FREQ MeHlsc01.

Var labs MeHlsa01 'Me - New Illness/disability 1'.

Var labs MeHlsb01 'Me -New Illness/disability 2'.

Var labs MeHlsc01 'Me - New Illness/disability 3'.

Value labels MeHlsa01 MeHlsb01 MeHlsc01

1 'Cancer (neoplasm) including lumps, masses, tumours and growths and benign (non-malignant) lumps and cysts'

2 'Diabetes'

3 'Other endocrine/metabolic'

4 'Mental illness/anxiety/depression/nerves (nes)'

5 'Mental handicap'

6 'Epilepsy/fits/convulsions'

7 'Migraine/headaches'

8 'Other problems of nervous system'

9 'Cataract/poor eye sight/blindness'

10 'Other eye complaints'

11 'Poor hearing/deafness'

12 'Tinnitus/noises in the ear'

13 'Meniere s disease/ear complaints causing balance problems'

14 'Other ear complaints'

15 'Stroke/cerebral haemorrhage/cerebral thrombosis'

16 'Heart attack/angina'

17 'Hypertension/high blood pressure/blood pressure (nes)'

18 'Other heart problems'

19 'Piles/haemorrhoids incl. Varicose Veins in anus'

20 'Varicose veins/phlebitis in lower extremities'

21 'Other blood vessels/embolic'

22 'Bronchitis/emphysema'

23 'Asthma'

24 'Hayfever'

25 'Other respiratory complaints'

- 26 'Stomach ulcer/ulcer (nes)/abdominal hernia/rupture'
- 27 'Other digestive complaints (stomach, liver, pancreas, bile ducts, small intestine - duodenum, jejunum and ileum)'
- 28 'Complaints of bowel/colon (large intestine, caecum, bowel, colon, rectum)'
- 29 'Complaints of teeth/mouth/tongue'
- 30 'Kidney complaints'
- 31 'Urinary tract infection'
- 32 'Other bladder problems/incontinence'
- 33 'Reproductive system disorders'
- 34 'Arthritis/rheumatism/fibrositis'
- 35 'Back problems/slipped disc/spine/neck'
- 36 'Other problems of bones/joints/muscles'
- 37 'Infectious and parasitic disease'
- 38 'Disorders of blood and blood forming organs and immunity disorders'
- 39 'Skin complaints'
- 40 'Other complaints'
- 41 'Complaint no longer present'
- 42 'Other specific'
- 1 'Item not applicable'
- 8 'Don t know'
- 9 'Refusal'.

FORMATS MeHlsa01 MeHlsb01 MeHlsc01 (F2.0).
 Missing values MeHlsa01 MeHlsb01 MeHlsc01 (-9 thru -1).

FREQ MeHlsa01 MeHlsb01 MeHlsc01.

cro MeHlsa01 by DisProbC.
 cro MeHlsb01 by DisProbC.
 cro MeHlsb01 by DisProbC.

IF (DisProbC=1) AND (MeHlsa01=0) MeHlsa01=-1.
 IF (DisProbC=1) AND (MeHlsb01=0) MeHlsb01=-1.
 IF (DisProbC=1) AND (MeHlsc01=0) MeHlsc01=-1.
 Exe.

FREQ MeHlsa01 MeHlsb01 MeHlsc01.

Self Control

** no DV syntax

Activities

*** Activities DVs re TV watching in this section.

*** WATCHING TELEVISION ***.

RENAME VARS (Atv01=MeAtv01) (Atv02=MeAtv02)
 (Atvwd=MeAtvwd) (Atv24=MeAtv24) (Atv25=MeAtv25).

Var Labs MeAtv01 'Me - Days child watched TV last week'.
 Var Labs MeAtv02 'Me - Does child ever watch TV'.
 Var Labs MeAtvwd 'Me - Does child ever watch TV at weekend'.
 Var Labs MeAtv24 'Me - Time child watches TV weekends'.
 Var Labs MeAtv25 'Me - Time child watches TV week days'.

ADD VAL LABS MeAtv01 MeAtv02 MeAtvwd MeAtv24 MeAtv25

-1 'Item not applicable'
-8 'Don t know'
-9 'Refusal'.

ADD VAL LABS MeAtv02
1 'Yes'
2 'No'.

FREQ MeAtv01 MeAtv02 MeAtvwd MeAtv24 MeAtv25.

* some extreme values as already noted at routing checks (up to 20 hrs a day)
* hours recorded with 1dp.

**NUMBER OF HOURS WATCHED TV ON WEEKDAYS & weekends

RECODE
MeAtv24
(0 = copy) (0.1 thru 0.4=1) (0.5 thru 0.9=2) (1 thru 1.9=3) (2 thru 2.9=4) (3 thru 3.9=5) (4 thru 4.9=6) (5 thru Highest=7) (else=copy) INTO
DeAtv24.
Execute.
IF (MeAtv01=0) DeAtv24=-1.
EXECUTE.
VARIABLE LABELS DeAtv24 'Da Hours of TV watched (weekdays)'.

RECODE
MeAtv25
(0 = copy) (0.1 thru 0.4=1) (0.5 thru 0.9=2) (1 thru 1.9=3) (2 thru 2.9=4) (3 thru 3.9=5) (4 thru 4.9=6) (5 thru Highest=7) (else=copy) INTO
DeAtv25.
Execute.
IF (MeAtv01=0) DeAtv25=-1.
EXECUTE.
VARIABLE LABELS DeAtv25 'Da Hours of TV watched (weekends)'.

val labs DeAtv24 DeAtv25
0 'None'
1 'Up to/under 30 minutes'
2 '30 minutes to less than 1 hour'
3 '1 hour to less than 2 hours'
4 '2 hours to less than 3 hours'
5 '3 hours to less than 4 hours'
6 '4 hours to less than 5 hours'
7 '5 or more hours'
-1 'Item not applicable'
-8 'Don t know'
-9 'Refused'.
FORMATS DeAtv24 DeAtv25 (F2.0).
MISSING VALUES DeAtv24 DeAtv25 (-9 thru -1).
fre DeAtv24 DeAtv25.

Physical Activity

** no DV syntax

Food & Eating

** no DV syntax

Employment, Education & Demographics

Derived variables:

DeWendM	De - Month resp's job ended
DeWendY	De - Year resp's job ended
DeYendM	De - Month partner's job ended
DeYendY	De - Year partner's job ended
DeCfpl20	De Does employer offer any family friendly policies?
DeCfpl03	De Does employer offer flexible working?
DeCfpl04	De Does employer offer any other fam friendly policy (excl subsid. ccare & workplace nursery for BC1 comparison)?
DeCfpl05	De Does employer offer time off (paid or unpaid) when child is sick?
DeCfpl21	De Does respondent use any family friendly policies?
DeCfpl13	Da Does respondent use flexible working?
DeCfpl14	De Does resp use some other family friendly policy - excl subsidised ccare and workplace nursery?
DeCfpl15	Da Does respondent use time off when child is sick ?
DeWnow01	De Respondent's current working status
DeMsta01	De Respondents employment status ver1
DeMsta02	De Respondents employment status ver2
DeYsta01	De Partners employment status
DeYsta02	De Partners employment status ver2
DeMsta10	Da Household employment: Measure 1
DeMsta11	Da Household employment and family type
DeWsta02	De Mothers employment status (incl. adopt./foster/step-mothers)
DeWsta03	De Mothers employment status ver2 (incl. adopt./foster/step-mothers)
DeMsec01	De Respondent NSSEC - 6 Category
DeYsec01	De Partner NSSEC - 6 Category
DeMsec10	De Household NSSEC - 6 Category

***** EMPLOYMENT & ECONOMIC ACTIVITY *****.

*** DATE RESPONDENT'S JOB ENDED Wend01_ (MaWend01) *****

* derived into month & year for archive (keep only Year for archive for anonymity?).

FREQ WrkNow2 Wend_.

RENAME VARS (Wend_ = MeWend01).

VAR LABS MeWend01 "Me Date Resp's job ended".

COMPUTE DeWendM = XDATE.MONTH(MeWend01) .

EXECUTE.

RECODE DeWendM (sysmis=-1) (else=copy).

Exe.

Var labs DeWendM "De - Month resp's job ended".

Val labs DeWendM

1 'January' 2 'February' 3 'March' 4 'April' 5 'May' 6 'June' 7 'July' 8 'August'

9 'September' 10 'October' 11 'November' 12 'December' -1 'Not applicable'.

FORMATS DeWendM (F2.0).

MISSING VALUES DeWendM (-1).

freq DeWendM.

```
COMPUTE DeWendY = XDATE.YEAR(MeWend01) .
EXECUTE .
RECODE DeWendY (sysmis=-1) (else=copy).
Exe.
```

```
Var labs DeWendY "De - Year resp's job ended".
Val Labs DeWendY
-1 'Not applicable'.
FORMATS DeWendY (F8.0).
MISSING VALUES DeWendY (-1).
freq DeWendY.
```

```
*** DATE PARTNER's JOB ENDED Yend01_ (MeYend01) *****
* derived into month & year for archive (keep only Year for archive for anonymity?).
```

```
FREQ PrCurrWk PrJobEnd_.
```

```
* asked IF (PCurrWk = No).
* range 15.10.2000 to 29.3.2014.
```

```
RENAME VARS (PrJobEnd_=MeYend01).
VAR LABS MeYend01 "Me - Date Ptner's job ended".
```

```
COMPUTE DeYendM = XDATE.MONTH(MeYend01) .
EXECUTE .
RECODE DeYendM (sysmis=-1) (else=copy).
Exe.
```

```
Var labs DeYendM "De - Month partner's job ended".
Val labs DeYendM
1 'January' 2 'February' 3 'March' 4 'April' 5 'May' 6 'June' 7 'July' 8 'August' 9 'September'
10 'October' 11 'November' 12 'December' -1 'Not applicable'.
FORMATS DeYendM (F2.0).
MISSING VALUES DeYendM (-1).
freq DeYendM.
```

```
COMPUTE DeYendY = XDATE.YEAR(MeYend01) .
EXECUTE .
RECODE DeYendY (sysmis=-1) (else=copy).
Exe.
```

```
Var labs DeYendY "De - Year partner's job ended".
Val Labs DeYendY
-1 'Not applicable'.
FORMATS DeYendY (F8.0).
MISSING VALUES DeYendY (-1).
```

```
freq DeYendY.
```

```
*****FAMILY FRIENDLY EMPLOYMENT POLICIES so coded as a DeC... (note: 3rd letter C is for
CCare normally) ***
```

```
FREQ EmpSt2.
```

```
freq Wff11 Wff12 Wff13 Wff14 Wff15 Wff16 Wff17 Wff18 Wff19 Wff110 Wff111 Wff112.
```

```
***WHETHER OR NOT EMPLOYER OFFERS ANY FAMILY FRIENDLY POLICY (same syntax as used at
BC1 Sw6).
```

```
RENAME VARS (Wff11=MeWff01) (Wff12=MeWff07) (Wff13=MeWff02) (Wff14=MeWff08)
(Wff15=MeWff09) (Wff16=MeWff10) (Wff17=MeWff11) (Wff18=MeWff12)
(Wff19=MeWff13) (Wff110=MeWff14) (Wff111=MeWff94) (Wff112=MeWff95).
```

```
Var Labs MeWff01 'Me - Employer provides subsidised ccare'.
Var Labs MeWff07 'Me - Empl gives childcare vouchers'.
Var Labs MeWff02 'Me - Employer provides nursery'.
Var Labs MeWff08 'Me - Employer has flex work hrs (always)'.
Var Labs MeWff09 'Me - Employer has flex work hrs (arrange)'.
Var Labs MeWff10 'Me - Paid leave (extra) if child sick'.
```

Var Labs MeWff11 'Me - Unpaid leave if child sick'.
Var Labs MeWff12 'Me - Unpaid time off in school hols'.
Var Labs MeWff13 'Me - Can work at home some/all the time'.
Var Labs MeWff14 'Me - Can job-share'.
Var Labs MeWff94 'Me - Some other family friendly facility'.
Var Labs MeWff95 'Me - No family friendly facilities'.

MISSING VALUES MeWff95 ().

Compute DeCfpl20 = -1.

If any (1, MeWff01, MeWff02, MeWff07, MeWff08, MeWff09, MeWff10, MeWff11, MeWff12, MeWff13, MeWff14, MeWff94) DaCfpl20 = 1.

Exe.

If (MeWff01 = 0) AND (MeWff02 = 0) AND (MeWff07 = 0) AND (MeWff08 = 0) AND (MeWff09 = 0) AND (MeWff10 = 0) AND (MeWff11 = 0) AND (MeWff12 = 0)

AND (MeWff13 = 0) AND (MeWff14 = 0) AND (MeWff94=0) DeCfpl20 = 0.

Exe.

IF (MeWff95<0) DeCfpl20=MeWff95.

Execute.

var labs DeCfpl20 'De Does employer offer any family friendly policies?'.
val labs DeCfpl20

0 'No'

1 'Yes'

-1 'Item not applicable'

-8 'Don t know'.

Missing values DeCfpl20 (-8,-1).

FORMATS DeCfpl20 (F2.0).

fre DeCfpl20.

MISSING VALUES MeWff95 (-9 thru -1).

FREQ MeWff95.

MISSING VALUES MeWff95 ().

IF (MeWff95=1) DeCfpl20=0.

IF (MeWff95=0) DeCfpl20=1.

IF (MeWff95<0) DeCfpl20=MeWff95.

var labs DeCfpl20 'De Does employer offer any family friendly policies?'.
val labs DeCfpl20 0 'No' 1 'Yes' -1 'Item not applicable' -8 'Don t know'.

Missing values DeCfpl20 (-8,-1).

FORMATS DeCfpl20 (F2.0).

Execute.

fre DeCfpl20.

***BANDING of the various policies INTO BROADER GROUPS

** FLEXIBLE working - keeping same base and the 'Don't know' of the original vars in that section.

MISSING VALUES MeWff08 MeWff09 ().

If any(1, MeWff08, MeWff09) DeCfpl03 = 1.

Exe.

If ((MeWff08=0) AND (MeWff09=0)) DeCfpl03 = 0.

Exe.

If (MeWff08<0) DeCfpl03 = MeWff08.

Exe.

var labs DeCfpl03 'De Does employer offer flexible working?'.
val labs DeCfpl03 0 'No' 1 'Yes' -1 'Item not applicable' -8 'Don t know'.

Missing Values DeCfpl03 (-8,-1).

FORMATS DeCfpl03 (F2.0).

fre DeCfpl03.

** OTHER family friendly policies (does not include MeWff08 & 09 above)

* MeWff01 & MeWff02 left out to be able to compare with Sw2 & Sw3 DVs

MISSING VALUES MeWff07 to MeWff94 ().

If any (1, MeWff07, MeWff10, MeWff11, MeWff12, MeWff13, MeWff14, MeWff94) DeCfpl04 = 1.

Exe.

IF ((MeWff07=0) AND (MeWff10=0) AND (MeWff11=0) AND (MeWff12=0) AND (MeWff13=0) AND (MeWff14=0)

AND (MeWff94=0)) DeCfpl04=0.
Exe.
If (MeWff07<0) DeCfpl04 = MeWff07.
var labs DeCfpl04 'De Does employer offer any other fam friendly policy (excl subsid. ccare & workplace nursery for BC1 comparison)?'.
val labs DeCfpl04
0 'No'
1 'Yes'
-1 'Item not applicable'
-8 'Don t know'.
Missing values DeCfpl04 (-8,-1).
FORMATS DeCfpl04 (F2.0).
Exe.

fre DeCfpl04.

**TIME OFF to care for sick children (missing values already switched off above)

If any (1, MeWff10, MeWff11) DeCfpl05 = 1.
Exe.
IF ((MeWff10=0) AND (MeWff11=0)) DeCfpl05=0.
Exe.
If (MeWff10<0) DeCfpl05 = MeWff10.
Exe.
var labs DeCfpl05 'De Does employer offer time off (paid or unpaid) when child is sick?'.
val labs DeCfpl05
0 'No'
1 'Yes'
-1 'Item not applicable'
-8 'Don t know'.
Missing values DeCfpl05 (-8,-1).
FORMATS DeCfpl05 (F2.0).

FREQ DeCfpl05.

MISSING VALUES MeWff07 to MeWff94 (-9 thru -1).

****USE of ANY family friendly policy (asked if MeWff95 = 0)

FREQ FamUse11.

freq Wff21 Wff22 Wff23 Wff24 Wff25 Wff26 Wff27 Wff28 Wff29 Wff210 Wff211 Wff212.

RENAME VARS (Wff21=MeWff21) (Wff22=MeWff22) (Wff23=MeWff23) (Wff24=MeWff24)
(Wff25=MeWff25) (Wff26=MeWff26) (Wff27=MeWff27) (Wff28=MeWff28) (Wff29=MeWff29)
(Wff210=MeWff29a) (Wff211=MeWff30) (Wff212=MeWff31).

Var Labs MeWff21 'Me - Use subsidised childcare'.
Var Labs MeWff22 'Me - Use childcare vouchers'.
Var Labs MeWff23 'Me - Use workplace creche'.
Var Labs MeWff24 'Me - Use flex hours (always avail)'.
Var Labs MeWff25 'Me - Use flex hours (arrangement)'.
Var Labs MeWff26 'Me - Use extra paid leave if child sick'.
Var Labs MeWff27 'Me - Use unpaid leave if child sick'.
Var Labs MeWff28 'Me - Use unpaid leave in school hols'.
Var Labs MeWff29 'Me - Use working at home facility'.
Var Labs MeWff29a 'Me - Use job-share facility'.
Var Labs MeWff30 'Me - Use other family friendly facility'.
Var Labs MeWff31 'Me - not using any family friendly policies'.

MISSING VALUES MeWff21 ().

Compute DeCfpl21 = -1.

If any (1, MeWff21, MeWff22, MeWff23, MeWff24, MeWff25, MeWff26, MeWff27,
MeWff28, MeWff29, MeWff29a, MeWff30) DeCfpl21 = 1.

Exe.

If ((MeWff21 = 0) AND (MeWff22 = 0) AND (MeWff23 = 0) AND (MeWff24 = 0) AND (MeWff25 = 0) AND
(MeWff26 = 0) AND (MeWff27 = 0) AND (MeWff28 = 0)
AND (MeWff29 = 0) AND (MeWff29a = 0) AND (MeWff30=0)) DeCfpl21 = 0.

Exe.

IF (MeWff21=-8) DeCfpl21 = -8.
Exe.
If (DeCfpl20 = 0) DeCfpl21 = -2.
Exe.
var labs DeCfpl21 'De Does respondent use any family friendly policies?'.
val labs DeCfpl21
0 'No'
1 'Yes'
-1 'Item not applicable'
-2 'Not applicable - employer does not offer any'
-8 'Don t know'.
Missing values MeWff21 DeCfpl21 (-9 thru -1).
FORMATS DeCfpl21 (F2.0).

FREQ DeCfpl21.

**USE of FLEXIBLE working

MISSING VALUES MeWff21 ().

Compute DeCfpl13=-1.

If any (1, MeWff24, MeWff25) DeCfpl13 = 1.

Exe.

If ((MeWff24=0) AND (MeWff25=0)) DeCfpl13 = 0.

Exe.

IF (MeWff21=-8) DeCfpl13 = -8.

Exe.

If (DeCfpl03 = 0) DeCfpl13 = -2.

var labs DeCfpl13 'Da Does respondent use flexible working?'.
val labs DeCfpl13

0 'No'

1 'Yes'

-1 'Item not applicable'

-2 'Not applicable, neither policy offered'

-8 'Don t know'.

Missing values MeWff21 DeCfpl13 (-9 thru -1).

FORMATS DeCfpl13 (F2.0).

Exe.

fre DeCfpl13.

**USE of some OTHER family friendly policy (excluding subsidised ccare and workplace nursery)

MISSING VALUES MeWff22 ().

Compute DeCfpl14 = -1.

If any (1, MeWff22, MeWff26, MeWff27, MeWff28, MeWff29, MeWff29a, MeWff30) DeCfpl14 = 1.

Exe.

IF ((MeWff22=0) AND (MeWff26=0) AND (MeWff27=0) AND (MeWff28=0) AND (MeWff29=0)

AND (MeWff29a=0) AND (MeWff30)) DeCfpl14 =0.

Exe.

If (DeCfpl04 = 0) DeCfpl14 = -2.

IF (MeWff22=-8) DeCfpl14 = -8.

Exe.

var labs DeCfpl14 'De Does resp use some other family friendly policy - excl subsidised ccare and

workplace nursery?'.
val labs DeCfpl14

0 'No'

1 'Yes'

-1 'Item not applicable'

-2 'Not applicable, none of these policies offered'

-8 'Don t know'.

Missing values MeWff22 DeCfpl14 (-9 thru -1).

FORMATS DeCfpl14 (F2.0).

fre DeCfpl14.

**USE of TIME OFF when child is sick

MISSING VALUES MeWff26 ().

Compute DeCfpl15 = -1.

If any (1, MeWff26, MeWff27) DeCfpl15 = 1.
Exe.
IF ((MeWff26=0) AND (MeWff27=0)) DeCfpl15 = 0.
Exe.
If (DeCfpl05 = 0) DeCfpl15 = -2.
IF (MeWff26=-8) DeCfpl15 = -8.
Exe.
var labs DeCfpl15 'Da Does respondent use time off when child is sick ?'.
val labs DeCfpl15 0 'No' 1 'Yes' -1 'Item not applicable'
-2 'Not applicable, neither policy offered' -8 'Don t know'.
Missing values MeWff26 DeCfpl15 (-9 thru -1).
FORMATS DeCfpl15 (F2.0).

fre DeCfpl15.

MISSING VALUES MeWff01 to MeWff31 (-9 thru -1).

***** EMPLOYMENT *****

**** RESPONDENT EMPLOYMENT STATUS (based on nbr of hours worked *now*).

*** CALCULATING respondent "DV" "DaWnow01" BASED ON PAUL'S INSTRUCTIONS TAKING INTO ACCOUNT MORE COMPLICATED ROUTING.

West016 = MaWest20-25 re work in last 7 days;

** if resp is nat mother (BMum2=) DaHGrsp01=1, either
* currently on maternity leave (considered as employmt): Wmat01=MaWmat01=4
* [currently NOT on maternity leave, and has worked since child's birth: Wpnt01=1, then asked WrkNow2 and/or 3]
* or working in same job: WrkNow2=MaWsam01=1 (WrkNow2=2 can mean in diffrt job now or not working: clarified in WrkNow3 next]
* or currently working in diffrt job: WrkNow3=MaWnow01=1
* the rest being automatically considered as not working.

** if resp is nat father BDad2==DaHGrsp02=1, if they ever worked they have been asked if currently working WrkNow3=1.

FREQ West011 to West016 WrkNow2 BDad2.

* renaming & relabelling relevant vars:

RENAME VARS (West011=MeWest20) (West012=MeWest21) (West013=MeWest22)
(West014=MeWest23)
(West015=MeWest24) (West016=MeWest25).

RENAME VARS(WrkNow2=MeWnow01).

RENAME VARS (BDad2 =DeHGrsp02).

VAR LABS MeWest20 'Me Resp employed last 7 days'.
VAR LABS MeWest21 'Me Resp on Gov sponsored training last 7 days'.
VAR LABS MeWest22 'Me Resp self-employed/freelance last 7 days'.
VAR LABS MeWest23 'Me Resp working unpaid for family last 7 days'.
VAR LABS MeWest24 'Me Resp doing other paid work last 7 days'.
VAR LABS MeWest25 'Me Resp not working last 7 days'.

VAR LABS MeWnow01 'Ma Whether resp currently has job'.
VAR LABS DeHGrsp02 'De Whether respondent is natural father'.

* setting the base of new DV with MeWest25 on ALL resps .
* then updating with relevant work-related vars.

RECODE MeWest25 (0=1) (1=2) (else=-3) INTO DeWnow01.

IF (MeWnow01=2) DeWnow01=2.

IF (MeWnow01=1) DeWnow01=1.

Exe.

Var labs DeWnow01 "De Respondent's current working status".

Val labs DeWnow01

1 'currently working'

2 'Not currently working'
-3 'Information not available'.
MISSING VALUES DeWnow01 (-9 thru -1).
FORMATS DeWnow01 (F2.0).
FREQ DeWnow01.

*** Respondent's EMPLOYMENT STATUS.

RENAME VARS (WrkEv2=MeWevr01) (JbHrs2=MeWtim01).

VAR LABS MeWevr01 'Me - Resp ever had a paid job'.
VAR LABS MeWtim01 'Me - Resp - hrs worked in week'.

ADD VAL LABS MeWtim01
-1 'Item not applicable'
-8 "Don't know"
-9 'Refusal'.

FREQ MeWtim01 DeWnow01 MeWevr01.

* integers OK.

Compute DeMsta01=-3.
Compute DeMsta02=-3.
Exe.
If (MeWevr01 = 2) or (DeWnow01 = 2) DeMsta01 = 3.
Exe.
If (MeWevr01 = 2) or (DeWnow01 = 2) DeMsta02 = 3.
Exe.

FREQ DeMsta01 DeMsta02.

DO IF (DeWnow01 = 1).
RECODE MeWtim01 (Lowest thru 34=2) (35 thru Highest=1) (else=-3) INTO DeMsta01 .
RECODE MeWtim01 (Lowest thru 15=2) (16 thru Highest=1) (else=-3) INTO DeMsta02 .
END IF.
Exe.

FREQ DeMsta01 DeMsta02.

VARIABLE LABELS DeMsta01 'De Respondents employment status ver1'.
val labs DeMsta01
1 'Working - full-time (>= 35 hrs pr wk)'
2 'Working - part-time (< 35 hrs pr wk)'
3 'Not working'
-3 'Information not available'.
FORMATS DeMsta01 (F2.0).
MISSING VALUES DeMsta01 (-9 thru -1).

VARIABLE LABELS DeMsta02 'De Respondents employment status ver2'.
val labs DeMsta02
1 'Working - full-time (>= 16 hrs pr wk)'
2 'Working - part-time (< 16 hrs pr wk)'
3 'Not working'
-3 'Information not available'.
FORMATS DeMsta02 (F2.0).
MISSING VALUES DeMsta02 (-9 thru -1).

fre DeMsta01 DeMsta02.

*** Resp's PARTNER EMPLOYMENT STATUS (based on nbr of hours worked *now*).

* unlike resp where we had to create a DV to find out if currently working, this has actually been asked (to resp: proxy info) about all partners who ever worked in 'PCurrWk2'.

RENAME VARS (PrEvPdWk=MeYevr01) (PrCurrWk=MeYnow01) (PrJbHrs2=MeYtim01).

VAR LABS MeYevr01 'Me - Partner ever had a paid job'.
VAR LABS MeYnow01 'Me - Does partner have paid job'.
VAR LABS MeYtim01 'Me - Part - hours worked in week'.

ADD VAL LABS MeYtim01
-1 'Item not applicable'
-8 "Don't know"
-9 'Refusal'.

FREQ MeYevr01 MeYnow01 MeYtim01.

* hours = integers OK.

COMPUTE DeYsta01=-3.
COMPUTE DeYsta02=-3.
Exe.
DO IF (MeYevr01 = 2) or (MeYnow01 = 2).
COMPUTE DeYsta01 = 3.
COMPUTE DeYsta02 = 3.
END IF.
Exe.

FREQ DeYsta01 DeYsta02.

DO IF (MeYnow01 = 1).
RECODE MeYtim01 (Lowest thru 34=2) (35 thru Highest=1) (else=-3) INTO DeYsta01 .
RECODE MeYtim01 (Lowest thru 15=2) (16 thru Highest=1) (else=-3) INTO DeYsta02 .
END IF.
Exe.

FREQ DeYsta01 DeYsta02.

If (DeHGnp04 = 0) DeYsta01 = -1.
Exe.
If (DeHGnp04 = 0) DeYsta02 = -1.
Exe.

VARIABLE LABELS DeYsta01 'De Partners employment status'.
val labs DeYsta01
-1 'N/A - No partner'
1 'Working - full-time (>=35 hrs pr wk)'
2 'Working - part-time (<35 hrs pr wk)'
3 'Not working'
-3 'Information not available'.

VARIABLE LABELS DeYsta02 'De Partners employment status ver2'.
val labs DeYsta02
-1 'N/A - No partner'
1 'Working - full-time (>=16 hrs pr wk)'
2 'Working - part-time (<16 hrs pr wk)'
3 'Not working'
-3 'Information not available'.
MISSING VALUES DeYsta01 DeYsta02 (-9 thru -1).
FORMATS DeYsta01 DeYsta02 (F2.0).

fre DeYsta01 DeYsta02.

*** HOUSEHOLD EMPLOYMENT STATUS 'DeMsta10' and 'DeMsta11'
*If there is no Partner in the Hhold DeHGnp04 (DeHGnp04)=0. Note: put Full-time condition (1) AFTER
Part-time condition (2) so longest working hrs overwrite shortest.

** DeMsta10 based on DeMsta01 & DeYsta01 (FT= 35 hrs+).

Compute DeMsta10=-3.
Exe.
If (DeHGnp04=0) DeMsta10 = DeMsta01.
Exe.
Do If (DeHGnp04=1).
If (DeMsta01=2 OR DeYsta01=2) DeMsta10 = 2.
If (DeMsta01=1 OR DeYsta01=1) DeMsta10 = 1.
If (DeMsta01=3 AND DeYsta01=3) DeMsta10 = 3.
End If.
Exe.

FREQ DeMsta10.

* RECODE DeMsta10 (sysmis=-4).

* Exe.

* TEMP.

* SELECT IF (DeMsta10=-4).

* LIST Serial DeHGnp04 DeMsta10 DeMsta01 DeYsta01.

* OK: no ptner in hhold and info missing for resp (-3). Recode as -3 'No info'.

* RECODE DeMsta10 (-4=-3) (else=copy).

* Exe.

var labs DeMsta10 'Da Household employment: Measure 1'.

val labs DeMsta10

1 'At least one parent/carer in full-time employment (>=35 hrs per week)'

2 'At least one parent/carer in part-time employment (<35 hrs per week)'

3 'No parent/carer working'

-3 'Information not available'.

FORMATS DeMsta10 (F2.0).

MISSING VALUES DeMsta10 (-9 thru -1).

fre DeMsta10.

** DeMsta11 based on DeMsta02 & DeYsta02 (FT= 16 hrs+).

Compute DeMsta11 = -3.

Do If (DeHGnp04=0).

If (DeMsta02=1) DeMsta11 = 1.

If (DeMsta02=2) DeMsta11 = 2.

If (DeMsta02=3) DeMsta11 = 2.

End If.

Exe.

FREQ DeMsta11.

Do If (DeHGnp04=1).

If (DeMsta02=1 AND DeYsta02=1) DeMsta11 = 3 .

If (DeMsta02=1 AND DeYsta02=2) DeMsta11 = 4 .

If (DeMsta02=1 AND DeYsta02=3) DeMsta11 = 4 .

If (DeMsta02=2 AND DeYsta02=1) DeMsta11 = 4 .

If (DeMsta02=3 AND DeYsta02=1) DeMsta11 = 4 .

If (DeMsta02=2 AND DeYsta02=2) DeMsta11 = 5.

If (DeMsta02=2 AND DeYsta02=3) DeMsta11 = 5.

If (DeMsta02=3 AND DeYsta02=2) DeMsta11 = 5.

If (DeMsta02=3 AND DeYsta02=3) DeMsta11 = 5.

End If.

Exe.

FREQ DeMsta11.

TEMP.

SELECT IF (DeMsta11=-3).

LIST Serial DeMsta11 DeMsta02 DeYsta02 DeMsta10 DeMsta01 DeYsta01.

VALUE LABELS DeMsta11

1 'Lone parent in paid work for >=16 hours'

2 'Lone parent unemployed or in paid work for <16 hours'

3 'Couple family both resp and partner working >=16 hours'

4 'Couple family either resp or partner working >=16 hours (but not both)'

5 'Couple family both unemployed or working <16 hours'

-3 'Information not available for at least one parent'.

var labs DeMsta11 'Da Household employment and family type'.

FORMATS DeMsta11 (F2.0).

Missing values DeMsta11 (-9 thru -1).

fre DeMsta11.

*** MOTHER's EMPLOYMENT STATUS (NATURAL mother OR NOT).

FREQ DeHGrsp07 DeHGrsp08.

* BC1 Sw1 syntax assumed that if resp was not nat mother, but nat mother in hhold, she must be the Partner of the resp.

* not necessarily the case, hence use of relationship DVs below.

MISSING VALUES DeMsta01 DeYsta01 ().

Compute DeWsta02 = -3.

If (DeHGrsp07=1) DeWsta02 = DeMsta01.

Exe.

If (DeHGrsp08=1) DeWsta02 = DeYsta01.

Exe.

MISSING VALUES DeMsta01 DeYsta01 (-9 thru -1).

var labs DeWsta02 'De Mothers employment status (incl. adopt./foster/step-mothers)'.
val labs DeWsta02

1 'Childs mother working - full-time (35hrs or more)'

2 'Childs mother working - part-time (less than 35 hrs)'

3 'Childs mother not working'

-3 'Information not available'.

Missing values DeWsta02 (-9 thru -1).

FORMATS DeWsta02 (F2.0).

fre DeWsta02.

** 2nd version: based on +/- 16 hrs work/week, as before.

MISSING VALUES DeMsta02 DeYsta02 ().

Compute DeWsta03 = -3.

If (DeHGrsp07 = 1) DeWsta03 = DeMsta02.

Exe.

If (DeHGrsp08=1) DeWsta03 = DeYsta02.

Exe.

MISSING VALUES DeMsta02 DeYsta02 (-9 thru -1).

var labs DeWsta03 'De Mothers employment status ver2 (incl. adopt./foster/step-mothers)'.
val labs DeWsta03

1 'Childs mother working - full-time (16hrs or more)'

2 'Childs mother working - part-time (less than 16 hrs)'

3 'Childs mother not working'

-3 'Information not available'.

Missing values DeWsta03 (-9 thru -1).

FORMATS DeWsta03 (F2.0).

fre DeWsta03.

* checking the 84 cases coded 'No info'.

MISSING VALUES DeWsta02 DeWsta03 ().

TEMP.

SELECT IF (DeWsta02=-3) OR (DeWsta03=-3).

LIST Serial DeHGrsp07 DeHGrsp08 DeWsta02 DeMsta01 DeYsta01 DeWsta03 DeMsta02 DeYsta02.

MISSING VALUES DeWsta02 DeWsta03 (-9 thru -1).

***** NSSEC FOR RESPONDENT & RESP's PARTNER ("NSSEC SOC2000" scale according to CAPI label; based on most recent SOC classification in reality).

* checking raw NSSEC vars present in dataset:

FREQ NSSEC NSSEC2 NSSEC3.

* NSSEC3 is empty.

* based on numbers of N/A, NSSEC must be for Resps and NSSEC2 for Ptner, as in BC1 sweeps.

```
RECODE
  NSSEC NSSEC2 (0 thru 6.9=1) (7.0 thru 7.9=2) (8.0 thru 9.9=3) (10.0 thru 11.9=4) (12.0 thru 13.9=5)
  (else = copy) INTO DeMtsec DeYtsec.
IF (DeHGnp04=0) DeYtsec=-4.
EXECUTE.
VARIABLE LABELS DeMtsec 'Respondent NSSEC - 5 Category'.
VARIABLE LABELS DeYtsec 'Partner NSSEC - 5 Category'.
VALUE LABELS DeMtsec DeYtsec
  - 1 'Not applicable'
  - 4 'No Partner in hhold'
  - 9 'Refusal'
  1 'Managerial and professional occupations'
  2 'Intermediate occupations'
  3 'Small employers and own account workers'
  4 'Lower supervisory and technical occupations'
  5 'Semi-routine and routine occupations'.
Missing values DeMtsec DeYtsec (-9 thru -1).
FORMATS DeMtsec DeYtsec (F2.0).
```

```
FREQ DeMtsec DeYtsec.
```

```
** 6 category-NSSEC vars (incl 'never worked'):
```

```
RECODE DeMtsec (else=copy) INTO DeMsec01.
Exe.
If (MeWevr01 = 2) DeMsec01=6.
Exe.
```

```
FREQ DeMsec01.
```

```
* RECODE DeMsec01 (-1=-3).
* Exe.
```

```
Var labels DeMsec01 'De Respondent NSSEC - 6 Category'.
```

```
VALUE LABELS DeMsec01
  - 3 'Information not available'
  1 'Managerial and professional occupations'
  2 'Intermediate occupations'
  3 'Small employers and own account workers'
  4 'Lower supervisory and technical occupations'
  5 'Semi-routine and routine occupations'
  6 'Never worked'.
```

```
Missing values DeMsec01 (-9 thru -1).
```

```
FORMATS DeMsec01 (F2.0).
```

```
fre DeMsec01.
```

```
RECODE DeYtsec (else=copy) INTO DeYsec01.
```

```
Exe.
```

```
If (MeYevr01 = 2) DeYsec01=6.
```

```
Exe.
```

```
FREQ DeYsec01.
```

```
* 'N/A' -1 : missing information; recoded as -3 'No info' below.
```

```
Recode DeYsec01 (-1=-3) (-4=-1) (else=copy).
```

```
Exe.
```

```
Var labels DeYsec01 'De Partner NSSEC - 6 Category'.
```

```
VALUE LABELS DeYsec01
  - 3 'Information not available'
  - 1 'N/A - No partner in hhold'
  1 'Managerial and professional occupations'
  2 'Intermediate occupations'
  3 'Small employers and own account workers'
  4 'Lower supervisory and technical occupations'
  5 'Semi-routine and routine occupations'
  6 'Never worked'.
```

```
Missing values DeYsec01 (-9 thru -1).
```

```
FORMATS DeYsec01 (F2.0).
```

```
fre DeYsec01.
```

**** HOUSEHOLD NSSEC.

* missing values left switched on to use only valid values in resp & ptner vars:

Compute DeMsec10 = DeMsec01.
if (DeYsec01 < DeMsec10) DeMsec10 = DeYsec01.
Exe.

fre DeMsec10.

* sysmis; recoded as -3 and checked.

RECODE DeMsec10 (sysmis=-3) (else=copy).
Exe.

fre DeMsec10.

* TEMP.
* SELECT IF (DaMsec10=-3).
* LIST Serial DaMsec10 DaMsec01 DaYsec01.

DO IF (DeYsec01=1).
RECODE DeMsec10 (-3=1) (else=copy).
END IF.
Exe.

Var labs DeMsec10 'De Household NSSEC - 6 Category'.

VALUE LABELS DeMsec10
-3 'Information not available'
1 'Managerial and professional occupations'
2 'Intermediate occupations'
3 'Small employers and own account workers'
4 'Lower supervisory and technical occupations'
5 'Semi-routine and routine occupations'
6 'Never worked'.

FORMATS DeMsec10 (F2.0).
MISSING VALUES DeMsec10 (-9 thru -1).

fre DeMsec10.

***** NS-SEC VARIABLES ("NSSEC SOC2000" scale according to CAPI label) .

***RESPONDENT'S & PARTNER'S NS-SEC - 6 category - 6th category= "never worked"
****Sw3 NSSEC vars - creating INTERMEDIARY vars (so NOT KEPT) DeMtsec & DeYtsec for Resp & Partner (a lot of N/A cases = if resp/partners have kept same job or if no partner in hhold)

* freq DeMtsec DeYtsec.
* delete variables DeMtsec DeYtsec.

RECODE
NSSEC NSSEC2 (0 thru 6.9=1) (7.0 thru 7.9=2) (8.0 thru 9.9=3) (10.0 thru 11.9=4) (12.0 thru 13.9=5)
(else = copy) INTO
DeMtsec DeYtsec.

IF (DeHGnp04=0) DeYtsec=-4.

EXECUTE.

VARIABLE LABELS DeMtsec 'Respondent New NSSEC - 5 Category'.

VARIABLE LABELS DeYtsec 'Partner New NSSEC - 5 Category'.

VALUE LABELS DeMtsec DeYtsec
- 1 'Not applicable'
- 4 'No Partner in hhold'
- 9 'Refusal'
1 'Managerial and professional occupations'
2 'Intermediate occupations'
3 'Small employers and own account workers'
4 'Lower supervisory and technical occupations'
5 'Semi-routine and routine occupations'.

Missing values DeMtsec DeYtsec (-9 thru -1).

FORMATS DeMtsec DeYtsec (F2.0).

FREQ DeMtsec DeYtsec.
fre DeHGnp04.

** NSSEC for ALL RESPS:

freq DeMtsec.
* has been created for 1011 cases.

freq DaMsec01 DcMsec01 .
freq FFSweep.
rename variables FFSweep=DeFFwdSw.

MISSING VALUES DaMsec01 DcMsec01 ().
Compute DeMsec01 = DaMsec01.
If (DeFFwdSw=2) AND (DcMsec01 > 0) DeMsec01 = DcMsec01.
Exe.
If (DeMtsec > 0) DeMsec01 = DeMtsec.
Exe.
If (MeWevr01 = 2) DeMsec01=6.
Exe.
fre DeMsec01.

RECODE DeMsec01 (-2=-3) (else=copy).
Exe.

Var labels DeMsec01 'De Respondent NSSEC - 6 Category'.

VALUE LABELS DeMsec01
- 3 'Employment information incomplete'
1 'Managerial and professional occupations'
2 'Intermediate occupations'
3 'Small employers and own account workers'
4 'Lower supervisory and technical occupations'
5 'Semi-routine and routine occupations'
6 'Never worked'.

Missing values DaMsec01 DcMsec01 DeMsec01 (-9 thru -1).
FORMATS DeMsec01 (F2.0).
var level DeMsec01 (nominal).
EXECUTE.
fre DeMsec01.

* ~~~~~.

NSSEC 6 categories for ALL PARTNERS *
* '-4' used temporarily for cases where no Partner in hhold
* as noted earlier, the Sweep2 to Sweep5 'DeFFwdSw' conditions added below may not be necessary as current sweep DV gets updated whenever new value exists in sweep chronological sequence
* also as noted earlier, we do not take into account whether new partner or same partner (with or without new quals) - main problem to do this is the fact that the routing for the NSSEC vars (coding) is not clear

freq DeHGnp04.

MISSING VALUES DaYsec01 DcYsec01 ().
Compute DeYsec01 = DaYsec01.
If (DeFFwdSw=2) AND (DcYsec01 > 0) DeYsec01 = DcYsec01.
Exe.
If (DeYtsec > 0) DeYsec01 = DeYtsec.
Exe.
If (MeYevr01 = 2) DeYsec01=6.
Exe.
If (DeHGnp04= 0) DeYsec01 = -4.
Exe.

fre DeYsec01.

* -2 and -1 recoded as '-3' (incomplete info)

* and the '-4' recoded as '-1' to match previous DVs answer labels:

Recode DeYsec01 (-1=-3) (-2=-3) (else=copy).

Exe.

Recode DeYsec01 (-4=-1) (else=copy).

Exe.

Var labels DeYsec01 'De Partner NSSEC - 6 Category'.

VALUE LABELS DeYsec01

- 3 'Employment information incomplete'

- 1 'N/A - No partner in hhold'

1 'Managerial and professional occupations'

2 'Intermediate occupations'

3 'Small employers and own account workers'

4 'Lower supervisory and technical occupations'

5 'Semi-routine and routine occupations'

6 'Never worked'.

Missing values DaYsec01 DcYsec01 DeYsec01 (-9 thru -1).

FORMATS DeYsec01 (F2.0).

var level DeYsec01 (nominal).

fre DeYsec01.

*recode sysmis=-3.

Missing values DeMsec01 DeYsec01 ().

Recode DeMsec01 (sysmis=-3) (-1=-3) (-2=-3) (else=copy).

Exe.

Recode DeYsec01 (sysmis=-3) (-4=-1) (else=copy).

Exe.

Missing values DeMsec01 DeYsec01 (lo thru -1).

fre DeMsec01 DeYsec01.

*-----.

***NS-SEC 6 category for HOUSEHOLD ***

* compute below only takes valid values from DhMsec01 since missing values left switched on (so ignores all missing values: = sysmis)

* proceeding step by step to check what the ((MISSING ())) syntax does.

Missing values DeMsec01 DeYsec01 ().

Compute DeMsec10 = DeMsec01.

if (DeYsec01 < DeMsec10) DeMsec10 = DeYsec01.

Exe.

fre DeMsec10.

* list DeMsec01 DeYsec01 DeMsec10.

* no sysmis OK, so don't need indented bits of syntax below

if (MISSING (DeYsec01)) DeMsec10 = DeMsec01.

Exe.

fre DeMsec10.

* exactly as before (which is logical since Resp NSSEC was set up as the default value at the start).

if (MISSING (DeMsec01)) DeMsec10 = DeYsec01.

Exe.

fre DeMsec10.

* Recode sysmis:.

RECODE DeMsec10 (sysmis = -3) (else=copy).

exe.

Var labs DeMsec10 'De Household NSSEC - 6 Category'.

VALUE LABELS DeMsec10

-3 'Employment information incomplete'

-1 'Not applicable'

1 'Managerial and professional occupations'

2 'Intermediate occupations'

3 'Small employers and own account workers'

4 'Lower supervisory and technical occupations'

5 'Semi-routine and routine occupations'

6 'Never worked'.

FORMATS DeMsec10 (F2.0).
MISSING VALUES DeMsec10 (-9 thru -1).
var level DeMsec10 (nominal).

fre DeMsec10.

***** EDUCATION *****.

Derived variables:

DeMedu01 De Highest Education level of Respondent
DeMedu02 De Highest Education level of Respondent - Banded
DeYedu01 De Highest Education level of Partner
DeYedu02 De Highest Education level of Partner - Banded
DeMedu03 De Highest Education level of Respondent - Banded v2
DeYedu03 De Highest Education level of Partner - Banded v2
DeMedu04 De BC2 Highest Education level of Respondent (SCQF)
DeYedu04 De BC2 Highest Education level of Partner (SCQF)
DeMedu10 De Highest Education level in hhold (SCQF)
DeMedu11 De Highest Education level in hhold, (lower standard grades and other merged) (SCQF)

***HIGHEST EDUCATION LEVEL OF RESPONDENT (GROUPED).
FREQ Meds01 to Medf0225.

RENAME VARS (Meds01=MeMeds01) (Meds021=MeMeds02) (Meds022=MeMeds03)
(Meds023=MeMeds04) (Meds024=MeMeds05) (Medf01=MeMedf01) (Medf021=MeMedf02)
(Medf022=MeMedf03) (Medf023=MeMedf04) (Medf024=MeMedf05) (Medf025=MeMedf06)
(Medf026=MeMedf07) (Medf027=MeMedf08) (Medf028=MeMedf09) (Medf029=MeMedf10)
(Medf0210=MeMedf11) (Medf0211=MeMedf12) (Medf0212=MeMedf13) (Medf0213=MeMedf14)
(Medf0214=MeMedf15) (Medf0215=MeMedf16) (Medf0216=MeMedf17)
(Medf0217=MeMedf18) (Medf0218=MeMedf19) (Medf0219=MeMedf20) (Medf0220=MeMedf21)
(Medf0221=MeMedf22) (Medf0222=MeMedf23) (Medf0223=MeMedf24)
(Medf0224=MeMedf25) (Medf0225=MeMedf94).

VAR LABS MeMeds01 'Me - any school quals (list 1)'.
VAR LABS MeMeds02 'Me - Section 1: Standard Grades 4-7 etc'.
VAR LABS MeMeds03 'Me - Section 2: Standard Grades 1-3 etc'.
VAR LABS MeMeds04 'Me - Section 3: Higher etc'.
VAR LABS MeMeds05 'Me - Section 4: Overseas exam/certificate'.

Var Labs MeMedf01 'Me - any further ed quals (list 2)'.
Var Labs MeMedf02 'Me Univ/CNAA first degree/diploma'.
Var Labs MeMedf03 'Me Postgraduate degree'.
Var Labs MeMedf04 'Me Teacher training qualification'.
Var Labs MeMedf05 'Me Nursing qualification'.
Var Labs MeMedf06 'Me Foundation/adv mod apprenticeship'.
Var Labs MeMedf07 'Me Other recognised apprenticeship'.
Var Labs MeMedf08 'Me OCR/RSA - (Vocational) Certificate'.
Var Labs MeMedf09 'Me OCR/RSA - (First) Diploma'.
Var Labs MeMedf10 'Me OCR/RSA - Advanced Diploma'.
Var Labs MeMedf11 'Me OCR/RSA - Higher Diploma'.
Var Labs MeMedf12 'Me Other clerical qualification'.
Var Labs MeMedf13 'Me City&Guilds - Level 1/ Part I'.
Var Labs MeMedf14 'Me City&Guilds - Level 2 etc'.
Var Labs MeMedf15 'Me City&Guilds - Level 3 etc'.
Var Labs MeMedf16 'Me City&Guilds - Level 4 etc'.
Var Labs MeMedf17 'Me SCOTVEC/BTEC First Certificate'.
Var Labs MeMedf18 'Me SCOTVEC/BTEC First/General Diploma'.
Var Labs MeMedf19 'Me SCOTVEC etc ONC/OND'.
Var Labs MeMedf20 'Me SCOTVEC etc HNC/HND'.
Var Labs MeMedf21 'Me SVQ/NVQ Lev 1 etc'.
Var Labs MeMedf22 'Me SVQ/NVQ Lev 2 etc'.
Var Labs MeMedf23 'Me SVQ/NVQ Lev 3 etc'.

Var Labs MeMedf24 'Me SVQ/NVQ Lev 4 etc'.
Var Labs MeMedf25 'Me SVQ/NVQ Lev 5 etc'.
Var Labs MeMedf94 'Me Other qualification'.

MISSING VALUES MeMeds01 MeMedf01 ().
If (MeMeds01=2) AND (MeMedf01=2) DeMedu01 = 6.
If (MeMeds01<0) AND (MeMedf01<0) DeMedu01 = -3.
Exe.
MISSING VALUES MeMeds01 MeMedf01 (-9 thru -1).

FREQ DeMedu01.

If any (1, MeMeds05, MeMedf94) DeMedu01 = 5.
Exe.
If any (1, MeMeds02, MeMeds03) DeMedu01 =4.
Exe.
If (MeMeds04 = 1) DeMedu01 = 3.
Exe.
If ANY(1, MeMedf06 to MeMedf25) DeMedu01 =2.
Exe.
If ANY(1, MeMedf02, MeMedf03, MeMedf04, MeMedf05) DeMedu01 =1.
Exe.

FREQ DeMedu01.

* 2 sysmis left; recoded as -1 below for identification:

RECODE DeMedu01 (sysmis=-1).
Exe.

var labs DeMedu01 'De Highest Education level of Respondent'.
val labs DeMedu01
6 'No Qualifications' 5 'Other' 4 'Standard Grade or equivalent' 3 'Higher Grade or equivalent'
2 'Vocational qualification below degree' 1 'Degree or equivalent' -3 'Information not available'.
Missing values DeMedu01 (-9 thru -1).
FORMATS DeMedu01 (F2.0).

fre DeMedu01.

* banded version.

Recode DeMedu01 (1 thru 3 = 1) (4 thru 5 = 2) (6 = 3) (else = -3) into DeMedu02.
Exe.

Var labs DeMedu02 'De Highest Education level of Respondent - Banded'.
Val labs DeMedu02
1 'Higher or above' 2 'Standard grade or other' 3 'No qualifications' -3 'Information not available'.
FORMATS DeMedu02 (F2.0).
MISSING VALUES DeMedu02 (-9 thru -1).

fre DeMedu02.

***** Stephen Hinchliffe's version

MISSING VALUES MeMeds01 MeMedf01 ().
If (MeMeds01<0) AND (MeMedf01<0) DeMedu03 = -3.
Exe.
MISSING VALUES MeMeds01 MeMedf01 (-9 thru -1).

If (MeMeds01 = 2) AND (MeMedf01 = 2) DeMedu03 = 1.
If ANY(1, MeMedf07, MeMedf12, MeMedf94) DeMedu03 =2.
If ANY(1, MeMeds02, MeMeds03, MeMedf08, MeMedf09, MeMedf13, MeMedf14, MeMedf17, MeMedf18,
MeMedf21, MeMedf22) DeMedu03 = 3.
If ANY(1, MeMeds04, MeMeds05, MeMedf06, MeMedf10, MeMedf15, MeMedf19, MeMedf23) DeMedu03
= 4.
If ANY(1, MeMedf11, MeMedf16, MeMedf20, MeMedf24) DeMedu03 = 5.
If ANY(1, MeMedf02, MeMedf03, MeMedf04, MeMedf05, MeMedf25) DeMedu03 = 6.
Exe.

FREQ DeMedu03.

RECODE DeMedu03 (sysmis=-3).
Exe.

Var label DeMedu03 "De Highest Education level of Respondent - Banded v2".
Val label DeMedu03

1 "No qualification" 2 "Other" 3 "GCSEs / Standard Grades / NVQ level 2 or below"
4 "A levels / Highers / NVQ level 3 or equiv" 5 "HNC, HND, NVQ level 4 or equiv"
6 "Degree / NVQ level 5 or equiv" -3 "Information not available".
MISSING VALUES DeMedu03 (-9 thru -1).
FORMATS DeMedu03 (F2.0).

FREQ DeMedu03.

**** Scottish Credit and Qualifications Framework SCQF version.

MISSING VALUES MeMeds01 MeMedf01 ().
If (MeMeds01<0) AND (MeMedf01<0) DeMedu04 = -3.
Exe.
MISSING VALUES MeMeds01 MeMedf01 (-9 thru -1).

If (MeMeds01 = 2) AND (MeMedf01 = 2) DeMedu04 = 1.
If ANY(1, MeMedf94, MeMeds05, MeMedf07, MeMedf12) DeMedu04 =2.
If ANY(1, MeMeds02, MeMedf08) DeMedu04 = 3.
If ANY(1, MeMeds03, MeMedf09, MeMedf13, MeMedf14, MeMedf17, MeMedf18, MeMedf19, MeMedf21,
MeMedf22) DeMedu04 = 4.
If ANY(1, MeMeds04, MeMedf06, MeMedf10, MeMedf11, MeMedf15, MeMedf16, MeMedf20, MeMedf23)
DeMedu04 = 5.
If ANY(1, MeMedf02, MeMedf03, MeMedf04, MeMedf05, MeMedf24, MeMedf25) DeMedu04 = 6.
Exe.

FREQ DeMedu04.

RECODE DeMedu04 (sysmis=-3).
Exe.

FORMATS DeMedu04 (f2.0).
VAR LABS DeMedu04 'De Highest Education level of Respondent (SCQF)'.
VAL LABS DeMedu04
1 'No qualification' 2 'Other' 3 'Lower level Standard Grades and Vocational qualifications'
4 'Upper level Standard Grades and Intermediate Vocational qualifications'
5 'Higher grades and Upper level vocational qualifications'
6 'Degree level academic and vocational qualifications' -3 'Information not available'.
MISSING VALUES DeMedu04 (-9 thru -1).

FREQ DeMedu04.

****HIGHEST EDUCATION LEVEL OF Respondent's PARTNER (GROUPED).

FREQ Yeds01 to Yedf0225.

RENAME VARS (Yeds01=MeYeds01) (Yeds021=MeYeds02) (Yeds022=MeYeds03)
(Yeds023=MeYeds04) (Yeds024=MeYeds05)
(Yedf01=MeYedf01) (Yedf021=MeYedf02) (Yedf022=MeYedf03) (Yedf023=MeYedf04)
(Yedf024=MeYedf05) (Yedf025=MeYedf06) (Yedf026=MeYedf07) (Yedf027=MeYedf08)
(Yedf028=MeYedf09) (Yedf029=MeYedf10) (Yedf0210=MeYedf11) (Yedf0211=MeYedf12)
(Yedf0212=MeYedf13) (Yedf0213=MeYedf14) (Yedf0214=MeYedf15)
(Yedf0215=MeYedf16) (Yedf0216=MeYedf17) (Yedf0217=MeYedf18) (Yedf0218=MeYedf19)
(Yedf0219=MeYedf20) (Yedf0220=MeYedf21) (Yedf0221=MeYedf22)
(Yedf0222=MeYedf23) (Yedf0223=MeYedf24) (Yedf0224=MeYedf25) (Yedf0225=MeYedf94).

Var Labs MeYeds01 'Me Part - any school quals (list 1)'.
Var Labs MeYeds02 'Me Part - Section 1: SCE SG 4-7 etc'.
Var Labs MeYeds03 'Me Part - Section 2: SCE SG 1-3 etc'.
Var Labs MeYeds04 'Me Part - Section 3: SCE Higher etc'.
Var Labs MeYeds05 'Me Part - Section 4: Overseas exam qls'.

Var Labs MeYedf01 'Me Part - any further ed quals (list 2)'.
Var Labs MeYedf02 'Me Part - Univ/CNAA degree etc'.
Var Labs MeYedf03 'Me Part - Postgraduate degree'.

Var Labs MeYedf04 'Me Part - Teacher training qualification'.
 Var Labs MeYedf05 'Me Part - Nursing qualification'.
 Var Labs MeYedf06 'Me Part - Foundation/adv mod apprent'.
 Var Labs MeYedf07 'Me Part - Other apprenticeship'.
 Var Labs MeYedf08 'Me Part - OCR/RSA - (Voc) Certificate'.
 Var Labs MeYedf09 'Me Part - OCR/RSA - (First) Diploma'.
 Var Labs MeYedf10 'Me Part - OCR/RSA - Advanced Diploma'.
 Var Labs MeYedf11 'Me Part - OCR/RSA - Higher Diploma'.
 Var Labs MeYedf12 'Me Part - other clerical qual'.
 Var Labs MeYedf13 'Me Part - C &G - Level 1/ Part I'.
 Var Labs MeYedf14 'Me Part - C &G - Level 2 etc.'.
 Var Labs MeYedf15 'Me Part - C &G - Level 3 etc.'.
 Var Labs MeYedf16 'Me Part - C &G - Level 4 etc.'.
 Var Labs MeYedf17 'Me Part - SCOTVEC/BTEC First Cert'.
 Var Labs MeYedf18 'Me Part - SCOTVEC/BTEC First/Gen Dip'.
 Var Labs MeYedf19 'Me Part - SCOTVEC etc ONC/OND'.
 Var Labs MeYedf20 'Me Part - SCOTVEC etc HNC/HND'.
 Var Labs MeYedf21 'Me Part - SVQ/NVQ Lev 1 etc'.
 Var Labs MeYedf22 'Me Part - SVQ/NVQ Lev 2 etc'.
 Var Labs MeYedf23 'Me Part - SVQ/NVQ Lev 3 etc'.
 Var Labs MeYedf24 'Me Part - SVQ/NVQ Lev 4 etc'.
 Var Labs MeYedf25 'Me Part - SVQ/NVQ Lev 5 etc'.
 Var Labs MeYedf94 'Me Part - other qualification'.

* mirror of Resp syntax:

MISSING VALUES MeYeds01 MeYedf01 ().
 If (MeYeds01<0) AND (MeYedf01<0) DeYedu01 = -3.
 Exe.
 MISSING VALUES MeYeds01 MeYedf01 (-9 thru -1).

If (MeYeds01=2) AND (MeYedf01=2) DeYedu01 = 6.
 If any (1, MeYeds05, MeYedf94) DeYedu01 = 5.
 If any (1, MeYeds02, MeYeds03) DeYedu01 =4.
 If (MeYeds04 = 1) DeYedu01 = 3.
 If any (1, MeYedf06 to MeYedf25) DeYedu01 =2.
 If any (1, MeYedf02, MeYedf03, MeYedf04, MeYedf05) DeYedu01 =1.
 Exe.

FREQ DeYedu01.

* correcting for HHolds with no partner at BC2 Sw1 based on BSpous2/DaHGnp04:

If (DeHGnp04 = 0) DeYedu01 = -1.
 Exe.

FREQ DeYedu01.

RECODE DeYedu01 (sysmis=-4).
 Exe.

RECODE DeYedu01 (-4=-3) (else=copy).
 Exe.

var labs DeYedu01 'De Highest Education level of Partner'.
 val labs DeYedu01
 6 'No Qualifications' 5 'Other' 4 'Standard Grade or equivalent' 3 'Higher Grade or equivalent'
 2 'Vocational qualification below degree' 1 'Degree or equivalent' -1 'N/A - No partner'
 -3 'Information not available'.
 FORMATS DeYedu01 (F2.0).
 Missing values DeYedu01 (-9 thru-1).

fre DeYedu01.

** banded version.

Recode DeYedu01 (1 thru 3 = 1) (4 thru 5 = 2) (6 = 3) (else = Copy) into DeYedu02.
 Execute.

Var labs DeYedu02 'De Highest Education level of Partner - Banded'.

Val labs DeYedu02
1 'Higher or above' 2 'Standard grade or other' 3 'No qualifications' -1 'N/A - No partner'
-3 'Information not available'.
Missing values DeYedu02 (-9 thru -1).
FORMATS DeYedu02 (F2.0).

fre DeYedu02.

***** Stephen Hinchliffe's version.

MISSING VALUES MeYeds01 MeYedf01 ().
If (MeYeds01<0) AND (MeYedf01<0) DeYedu03 = -3.
Exe.
MISSING VALUES MeYeds01 MeYedf01 (-9 thru -1).

If (MeYeds01 = 2) AND (MeYedf01 = 2) DeYedu03 = 1.
If ANY(1, MeYedf07, MeYedf12, MeYedf94) DeYedu03 =2.
If ANY(1, MeYeds02, MeYeds03, MeYedf08, MeYedf09, MeYedf13, MeYedf14, MeYedf17, MeYedf18,
MeYedf21, MeYedf22) DeYedu03 = 3.
If ANY(1, MeYeds04, MeYeds05, MeYedf06, MeYedf10, MeYedf15, MeYedf19, MeYedf23) DeYedu03 =
4.
If ANY(1, MeYedf11, MeYedf16, MeYedf20, MeYedf24) DeYedu03 = 5.
If ANY(1, MeYedf02, MeYedf03, MeYedf04, MeYedf05, MeYedf25) DeYedu03 = 6.
Exe.

FREQ DeYedu03.

If (DeHGnp04 = 0) DeYedu03 = -1.
Exe.

FREQ DeYedu03.

RECODE DeYedu03 (sysmis=-3).
Exe.

Var label DeYedu03 "Da Highest Education level of Partner - Banded v2".
Val label DeYedu03
1 "No qualification" 2 "Other" 3 "GCSEs / Standard Grades / NVQ level 2 or below"
4 "A levels / Highers / NVQ level 3 or equiv" 5 "HNC, HND, NVQ level 4 or equiv"
6 "Degree / NVQ level 5 or equiv" -1 "N/A: no partner in hhold" -3 "Information not available".
MISSING VALUES DeYedu03 (-9 thru -1).
FORMATS DeYedu03 (F2.0).

FREQ DeYedu03.

**** Scottish Credit and Qualifications Framework SCQF version.

MISSING VALUES MeYeds01 MeYedf01 ().
If (MeYeds01<0) AND (MeYedf01<0) DeYedu04 = -3.
Exe.
MISSING VALUES MeYeds01 MeYedf01 (-9 thru -1).

If (MeYeds01 = 2) AND (MeYedf01 = 2) DeYedu04 = 1.
If ANY(1, MeYedf94, MeYeds05, MeYedf07, MeYedf12) DeYedu04 =2.
If ANY(1, MeYeds02, MeYedf08) DeYedu04 = 3.
If ANY(1, MeYeds03, MeYedf09, MeYedf13, MeYedf14, MeYedf17, MeYedf18, MeYedf19, MeYedf21,
MeYedf22) DeYedu04 = 4.
If ANY(1, MeYeds04, MeYedf06, MeYedf10, MeYedf11, MeYedf15, MeYedf16, MeYedf20, MeYedf23)
DeYedu04 = 5.
If ANY(1, MeYedf02, MeYedf03, MeYedf04, MeYedf05, MeYedf24, MeYedf25) DeYedu04 = 6.
Exe.

FREQ DeYedu04.

If (DeHGnp04 = 0) DeYedu04 = -1.
Exe.

FREQ DeYedu04.

RECODE DeYedu04 (sysmis=-3).

Exe.

FORMATS DeYedu04 (f2.0).

VAR LABS DeYedu04 'De Highest Education level of Partner (SCQF)'.

VAL LABS DeYedu04

1 'No qualification' 2 'Other' 3 'Lower level Standard Grades and Vocational qualifications'

4 'Upper level Standard Grades and Intermediate Vocational qualifications'

5 'Higher grades and Upper level vocational qualifications'

6 'Degree level academic and vocational qualifications' -3 'Information not available'

-1 'N/A: no Partner in hhold'.

MISSING VALUES DeYedu04 (-9 thru -1).

FREQ DeYedu04.

FREQ DeMedu04 DeYedu04.

Compute DeMedu10=-3.

Exe.

If (DeHGnp04=0) DeMedu10=DeMedu04.

Exe.

Do If (DeHGnp04=1).

If (DeMedu04>=DeYedu04) DeMedu10=DeMedu04.

If (DeMedu04<DaYedu04) DeMedu10=DeYedu04.

End If.

Exe.

FORMATS DeMedu10 (f2.0).

VAR LABS DeMedu10 'De Highest Education level in hhold (SCQF)'.

VAL LABS DeMedu10

1 'No qualification' 2 'Other' 3 'Lower level Standard Grades and Vocational qualifications'

4 'Upper level Standard Grades and Intermediate Vocational qualifications'

5 'Higher grades and Upper level vocational qualifications'

6 'Degree level academic and vocational qualifications' -3 'Information not available'.

MISSING VALUES DeMedu10 (-9 thru -1).

FREQ DeMedu10.

RECODE DeMedu10 (sysmis=-1).

Exe.

MISSING VALUES DeMedu10 ().

TEMP.

SELECT IF (DeMedu10=-1).

LIST Serial DeMedu10 DeMedu04 DeYedu04.

MISSING VALUES DeMedu10 (-9 thru -1).

MISSING VALUES DeMedu10 ().

TEMP.

SELECT IF (DeMedu10=-3).

LIST Serial DeMedu10 DeMedu04 DeYedu04.

MISSING VALUES DeMedu10 (-9 thru -1).

RECODE DeMedu10 (-1=-3).

Exe.

ADD VAL LABS DeMedu10

-3 'Information not available for partner, respondent or both'.

FREQ DeMedu10.

FREQ DeMedu04 DeYedu04.

Compute DeMedu10=-3.

Exe.

If (DeHGnp04=0) DeMedu10=DeMedu04.

Exe.

Do If (DeHGnp04=1).

If (DeMedu04>=DeYedu04) DeMedu10=DeMedu04.

If (DeMedu04<DaYedu04) DeMedu10=DeYedu04.

End If.

Exe.

FORMATS DeMedu10 (f2.0).
VAR LABS DeMedu10 'De Highest Education level in hhold (SCQF)'.
VAL LABS DeMedu10
1 'No qualification' 2 'Other' 3 'Lower level Standard Grades and Vocational qualifications'
4 'Upper level Standard Grades and Intermediate Vocational qualifications'
5 'Higher grades and Upper level vocational qualifications'
6 'Degree level academic and vocational qualifications' -3 'Information not available'.
MISSING VALUES DeMedu10 (-9 thru -1).
FREQ DeMedu10.

RECODE DeMedu10 (sysmis=-1).
Exe.

MISSING VALUES DeMedu10 ().
TEMP.
SELECT IF (DeMedu10=-1).
LIST IDnumber DeMedu10 DeMedu04 DeYedu04.
MISSING VALUES DeMedu10 (-9 thru -1).

MISSING VALUES DeMedu10 ().
TEMP.
SELECT IF (DeMedu10=-3).
LIST IDnumber DeMedu10 DeMedu04 DeYedu04.
MISSING VALUES DeMedu10 (-9 thru -1).

RECODE DeMedu10 (-1=-3).
Exe.

ADD VAL LABS DeMedu10
-3 'Information not available for partner, respondent or both'.
FREQ DeMedu10.

***** IDENTITY *****

Derived variables:

DeMeth07	De Ethnicity of Respondent
DeYeth07	De Ethnicity of Partner
DeMfai01	De Respondents religion
DeYfai01	De Partners religion

***ETHNICITY OF MAIN CARER = RESPONDENT - white or other ***

FREQ Ethnic09.

RECODE Ethnic09 (21=94) (else=copy).
Exe.

RENAME VARS (Ethnic09=MeMeth01).
VAR LABS MeMeth01 'Me Ethnicity of Respondent (not banded)'.

VALUE LABELS MeMeth01
-9 'Refusal' -8 "Don't Know" -1 'Item not applicable' 1 'White – Scottish' 2 'White – English'
3 'White – Welsh' 4 'White – Northern Irish' 5 'White – British' 6 'White – Irish'
7 'White – Gypsy/Traveller' 8 'White – Polish'
9 'White - Any other white background (WRITE IN)'
10 'Any mixed background (WRITE IN)'
11 'Asian, Asian Scottish or Asian British - Pakistani'
12 'Asian, Asian Scottish or Asian British - Indian'
13 'Asian, Asian Scottish or Asian British - Bangladeshi'
14 'Asian, Asian Scottish or Asian British – Chinese'
15 'Asian, Asian Scottish or Asian British – Any other Asian background (WRITE IN)'
16 'Black, Black Scottish or Black British - African'
17 'Black, Black Scottish or Black British - Caribbean'
18 'Black, Black Scottish or Black British'
19 'Black, Black Scottish or Black British - Any other black background (WRITE IN)'
20 'Arab' 94 'Any other ethnic group (WRITE IN)'.

FREQ MeMeth01.

RECODE MeMeth01 (1 thru 9=1) (10 thru hi=2) (else=copy) INTO DeMeth07.

Exe.

RECODE DeMeth07 (-1=-3) (else=copy).

Exe.

VARIABLE LABELS DeMeth07 'De Ethnicity of Respondent'.

VALUE LABELS DeMeth07

1 'White'

2 'Other ethnic background'

-3 'Information not available'

-8 'Don t know'

-9 'Refusal'.

FORMATS DeMeth07 (F2.0).

MISSING VALUES DeMeth07 (-9 thru -1).

FREQ DeMeth07.

***ETHNICITY OF STUDY CHILD - white or other ***

FREQ Chdeth.

RECODE Chdeth (21=94) (else=copy).

Exe.

RENAME VARS (Chdeth=MeEthGpC).

VAR LABS MeEthGpC 'Me Ethnicity of Child (not banded)'.

VALUE LABELS MeEthGpC

-9 'Refusal' -8 "Don't Know" -1 'Item not applicable' 1 'White – Scottish' 2 'White – English'

3 'White – Welsh' 4 'White – Northern Irish' 5 'White – British' 6 'White – Irish'

7 'White – Gypsy/Traveller' 8 'White – Polish'

9 'White - Any other white background (WRITE IN)'

10 'Any mixed background (WRITE IN)'

11 'Asian, Asian Scottish or Asian British - Pakistani'

12 'Asian, Asian Scottish or Asian British - Indian'

13 'Asian, Asian Scottish or Asian British - Bangladeshi'

14 'Asian, Asian Scottish or Asian British – Chinese'

15 'Asian, Asian Scottish or Asian British – Any other Asian background (WRITE IN)'

16 'Black, Black Scottish or Black British - African'

17 'Black, Black Scottish or Black British - Caribbean'

18 'Black, Black Scottish or Black British'

19 'Black, Black Scottish or Black British - Any other black background (WRITE IN)'

20 'Arab' 94 'Any other ethnic group (WRITE IN)'.

FREQ MeEthGpC.

RECODE MeEthGpC (1 thru 9=1) (10 thru hi=2) (else=copy) INTO DeEthGpC.

Exe.

RECODE DeEthGpC (-1=-3) (else=copy).

Exe.

VARIABLE LABELS DeEthGpC 'De Ethnicity of Child'.

VALUE LABELS DeEthGpC

1 'White'

2 'Other ethnic background'

-3 'Information not available'

-8 'Don t know'

-9 'Refusal'.

FORMATS DeEthGpC (F2.0).

MISSING VALUES DeEthGpC (-9 thru -1).

FREQ DeEthGpC.

***ETHNICITY OF Respondent's PARTNER - white or other ***

FREQ PrEthn09.

RECODE PrEthn09 (21=94) (else=copy).

Exe.

RENAME VARS (PrEthn09=MeYeth01).
VAR LABS MeYeth01 'Me Ethnicity of Partner (not banded)'.

VALUE LABELS MeYeth01

-9 'Refusal' -8 "Don't Know" -1 'Item not applicable'
1 'White – Scottish' 2 'White – English' 3 'White – Welsh' 4 'White – Northern Irish'
5 'White – British' 6 'White – Irish' 7 'White – Gypsy/Traveller' 8 'White – Polish'
9 'White - Any other white background (WRITE IN)'
10 'Any mixed background (WRITE IN)'
11 'Asian, Asian Scottish or Asian British - Pakistani'
12 'Asian, Asian Scottish or Asian British - Indian'
13 'Asian, Asian Scottish or Asian British - Bangladeshi'
14 'Asian, Asian Scottish or Asian British – Chinese'
15 'Asian, Asian Scottish or Asian British – Any other Asian background (WRITE IN)'
16 'Black, Black Scottish or Black British - African'
17 'Black, Black Scottish or Black British - Caribbean'
18 'Black, Black Scottish or Black British'
19 'Black, Black Scottish or Black British - Any other black background (WRITE IN)'
20 'Arab' 94 'Any other ethnic group (WRITE IN)'.

FREQ MeYeth01.

RECODE MeYeth01 (1 thru 9=1) (10 thru hi=2) (else=copy) INTO DeYeth07.

Exe.

RECODE DeYeth07 (-1=-3) (else=copy).

Exe.

IF (DeHGnp04=0) DeYeth07=-1.

Exe.

VARIABLE LABELS DeYeth07 'De Ethnicity of Partner'.

VALUE LABELS DeYeth07

1 'White' 2 'Other ethnic background' -1 'Not applicable: no partner in household'
-3 'No information' -8 'Don t know' -9 'Refusal'.
FORMATS DeYeth07 (F2.0).
MISSING VALUES DeYeth07 (-9 thru -1).

FREQ DeYeth07.

***** RELIGION *****

***RELIGION OF RESPONDENT ***.

**Reduces the full religion classification

FREQ ReligioS.

RECODE ReligioS (10=94) (11=12) (9=10) (8=9) (7=8) (6=7) (5=6) (4=5) (3=4) (2=3) (1=2) (0=1).

Exe.

FREQ ReligioS.

RENAME VARS (ReligioS=MeMfai01).

Var Labs MeMfai01 'Me Resp Religion (not banded)'.

VALUE LABELS MeMfai01

-9 'Refusal' -8 "Don't Know" -2 'Schedule not applicable' -1 'Item not applicable'
1 'No religion' 2 'Church of Scotland' 3 'Roman Catholic' 4 'Other Christian' 5 'Muslim'
6 'Buddhist' 7 'Sikh' 8 'Jewish' 9 'Hindu' 10 'Pagan' 94 'Another religion (please specify)'
12 'Refused'.

FREQ MeMfai01.

RECODE

MeMfai01

(1=0) (2=1) (3=2) (4 =1) (5=4) (6 thru 10=3) (94=3) (12=-9) (-1=-3) (else=copy) INTO DeMfai01.

Exe.

VARIABLE LABELS DeMfai01 'De Respondents religion'.

VALUE LABELS DeMfai01

0 'No Religion' 1 'Christian - Protestant or other non-Catholic'

2 'Roman Catholic' 3 'Other non-Christian'
4 'Muslim/Islam' -1 'Item not applicable'
-3 'Information not available: incomplete interview'
-8 'Don t know' -9 'Refusal'.
FORMATS DeMfai01 (F2.0).
MISSING VALUES DeMfai01 (-9 thru -1).

fre DeMfai01.

***RELIGION OF CHILD ***.

**Reduces the full religion classification

FREQ CRelig2.

RECODE CRelig2 (10=94) (11=12) (9=10) (8=9) (7=8) (6=7) (5=6) (4=5) (3=4) (2=3) (1=2) (0=1).
Exe.
FREQ CRelig2.

* val labs out of sync now but figures look OK.

RENAME VARS (CRelig2=MeMfai04).
Var Labs MeMfai04 'Me Childs Religion (not banded)'.
VALUE LABELS MeMfai04
-9 'Refusal' -8 "Don't Know" -2 'Schedule not applicable' -1 'Item not applicable'
1 'No religion' 2 'Church of Scotland' 3 'Roman Catholic' 4 'Other Christian' 5 'Muslim'
6 'Buddhist' 7 'Sikh' 8 'Jewish' 9 'Hindu' 10 'Pagan' 94 'Another religion (please specify)'
12 'Refused'.

FREQ MeMfai04.

RECODE
MeMfai04
(1=0) (2=1) (3=2) (4 =1) (5=4) (6 thru 10=3) (94=3) (12=-9) (-1=-3) (else=copy) INTO DeReligC.
Exe.

VARIABLE LABELS DeReligC 'De Childs religion'.

VALUE LABELS DeReligC
0 'No Religion' 1 'Christian - Protestant or other non-Catholic'
2 'Roman Catholic' 3 'Other non-Christian'
4 'Muslim/Islam' -1 'Item not applicable'
-3 'Information not available: incomplete interview'
-8 'Don t know' -9 'Refusal'.
FORMATS DeReligC (F2.0).
MISSING VALUES DeReligC (-9 thru -1).

fre DeReligC.

***RELIGION OF Respondent's PARTNER ***.

* mirror of Resp syntax (except for 'No Ptnr in hhold' correction in DV).

FREQ PrReligS.

RECODE PrReligS (10=94) (11=12) (9=10) (8=9) (7=8) (6=7) (5=6) (4=5) (3=4) (2=3) (1=2) (0=1).
Exe.
FREQ PrReligS.

* val labs out of sync now but figures look OK.

* MF 17 Jul-12: var lab error corrected at relabelling stage: partner's religion, not resp's religion as relabelled at this stage.

RENAME VARS (PrReligS=MeYfai01).
Var Labs MeYfai01 'Me Partners Religion (not banded)'.
VALUE LABELS MeYfai01
-8 "Don't Know" -2 'Schedule not applicable' -1 'Item not applicable'
1 'No religion' 2 'Church of Scotland' 3 'Roman Catholic' 4 'Other Christian' 5 'Muslim'
6 'Buddhist' 7 'Sikh' 8 'Jewish' 9 'Hindu' 10 'Pagan' 94 'Another religion (please specify)'
12 'Refused'.

FREQ MeYfai01.


```

RECODE
  MeYfai01
  (1=0) (2=1) (3=2) (4 =1) (5=4) (6 thru 10=3) (94=3) (12=-9) (-1=-3) (else=copy) INTO DeYfai01.
Exe.
If (DeHGnp04= 0) DeYfai01 = -1.
Exe.

```

```

VARIABLE LABELS DeYfai01 'De Partners religion'.
VALUE LABELS DeYfai01
  0 'No Religion' 1 'Christian - Protestant or other non-Catholic' 2 'Roman Catholic'
  3 'Other non-Christian' 4 'Muslim/Islam' -1 'N/A: no partner in household'
  -3 'Information not available: incomplete interview'
  -8 'Don t know' -9 'Refusal'.
FORMATS DeYfai01 (F2.0).
MISSING VALUES DeYfai01 (-9 thru -1).

```

```
fre DeYfai01.
```

```
*****
```

Income & Financial Stress

Derived variables:

DeWinc01	De Household income - banded
DeWinc02	De Household income - banded (alternative)
DeMcClem	De OECD-Modified McClements hhold score for equiv. income
DeEqvinc	De Equivalised income
DeEqv5	De Equivalised income - quintiles

*1) Creating McClemens score variable

*2) Creating equivalised household income variable (using McClemens score variable)

*Using syntax from BC2 Sw1 as basis for deriving McClemens score variable: '3_Derive_5_IncomeToEnd'.

*Using syntax from BC1 Sw8 for creating equivalised household income variable (BC2 sw1 and sw2 had only 9 categories for income while sw3 has 17, like BC1)

** Double-checking that the nbr of valid answers at DVAge.. vars is equal or less than nbr of corresponding Person nbr in dataset.

```

FREQ PersNo DVAge16 PersNo2 DVAge17 PersNo3 DVAge18 PersNo4 DVAge19 PersNo5 DVAge20
PersNo6 DVAge21 PersNo7 DVAge22 PersNo8 DVAge23
PersNo9 DVAge24 PersNo10 DVAge25 PersNo11 DVAge26 PersNo12 DVAge27 PersNo13 DVAge28
PersNo14 DVAge29 PersNo15 DVAge30
/FORMAT=NOTABLE.

```

**To count the NUMBER OF PEOPLE AGED 0-18 in the household.

* default value of 1 for DeHGkds because always at least one child in HHold = the study child (so DVAge16 not needed in list)

* also the syntax below assumes that Missing Values switched on so DVAge[...] always >=0, otherwise <19 would include -1=N/A.. (should be OK as missing values switched on earlier in dataset but switched on now just in case)

```

MISSING VALUES DVAge16 DVAge17 DVAge18 DVAge19 DVAge20 DVAge21 DVAge22 DVAge23
DVAge24 DVAge25
DVAge26 DVAge27 DVAge28 DVAge29 DVAge30 (-9 thru -1).
Compute DaHGkds = 1.

```

```
If (DVAge17 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge18 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge19 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge20 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge21 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge22 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge23 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge24 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge25 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge26 < 19) DaHGkds = DaHGkds + 1.
```

```
If (DVAge27 < 19) DaHGkds = DaHGkds + 1.
If (DVAge28 < 19) DaHGkds = DaHGkds + 1.
If (DVAge29 < 19) DaHGkds = DaHGkds + 1.
If (DVAge30 < 19) DaHGkds = DaHGkds + 1.
Exe.
```

```
RENAME VARIABLES DaHGkds=DeHGkds.
var labs DeHGkds 'De Nbr of persons aged 0-18 in household'.
FORMATS DeHGkds (F2.0).
fre DeHGkds.
```

**To count the NUMBER OF PEOPLE AGED 0-13 in the household.
* again default value set as 1 since always at least one child in HHold = the study child
* and line with DVAge16 = study child's age not required

```
Compute DaHGkds2 = 1.
If (DVAge17 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge18 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge19 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge20 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge21 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge22 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge23 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge24 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge25 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge26 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge27 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge28 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge29 < 14) DaHGkds2 = DaHGkds2 + 1.
If (DVAge30 < 14) DaHGkds2 = DaHGkds2 + 1.
Exe.
```

```
RENAME VARIABLES DaHGkds2 = DeHGkds2.
var labs DeHGkds2 'De Nbr of persons aged 0-13 in household'.
FORMATS DeHGkds2 (F2.0).
fre DeHGkds2.
```

*Compute a variable indicating the NUMBER OF KIDS aged 14-18 by subtracting nbr of 0-13 from total kids variable

```
Compute DeHGkds3 = DeHGkds - DeHGkds2.
var labs DeHGkds3 'De Nbr of persons aged 14-18 in household'.
exe.
FORMATS DeHGkds3 (F2.0).
```

```
fre DeHGkds3.
```

*Checking same total OK

```
compute allkds = DeHGkds3 + DeHGkds2.
cro DeHGkds by allkds.
```

* perfect X-tab OK

**To count the NUMBER OF PEOPLE AGED 19 OR OVER in the household.
* (Note: 1st line re study child's age DaHGag1 redundant as cannot be > 18)
* ">=19" used below is valid whether using cut-off or full ages for all > 18 yrs old (but compulsory if full ages, rather than using '> 18')

```
Compute DaHGads = 0.
If (DVAge16 >= 19) DaHGads = DaHGads + 1.
If (DVAge17 >= 19) DaHGads = DaHGads + 1.
If (DVAge18 >= 19) DaHGads = DaHGads + 1.
If (DVAge19 >= 19) DaHGads = DaHGads + 1.
If (DVAge20 >= 19) DaHGads = DaHGads + 1.
If (DVAge21 >= 19) DaHGads = DaHGads + 1.
If (DVAge22 >= 19) DaHGads = DaHGads + 1.
If (DVAge23 >= 19) DaHGads = DaHGads + 1.
If (DVAge24 >= 19) DaHGads = DaHGads + 1.
If (DVAge25 >= 19) DaHGads = DaHGads + 1.
```

```
If (DVAge26 >= 19) DaHGads = DaHGads + 1.
If (DVAge27 >= 19) DaHGads = DaHGads + 1.
If (DVAge28 >= 19) DaHGads = DaHGads + 1.
If (DVAge29 >= 19) DaHGads = DaHGads + 1.
If (DVAge30 >= 19) DaHGads = DaHGads + 1.
Exe.
```

```
RENAME VARIABLES DaHGads = DeHGads.
var labs DeHGads 'De Nbr of persons aged 19 or over in household'.
FORMATS DeHGads (F2.0).
fre DeHGads.
```

*** START of calculation syntax for "McClements" (OECD method in fact) and Equivalised Income vars.
* originally copied from BC1 Sw6 syntax (based on Sw2/3/4/5) - now amended with BC2 Sw3 names.

**Compute variable indicating TOTAL NUMBER OF PEOPLE in household

```
Compute allpers = DeHGads + DeHGkds.
fre allpers.
```

* same as DV re hhold size calculated in earlier syntax OK.

** Equiv income treats HEADS OF HOUSEHOLD differently, need to split adults into head/other adults in the household.

** Remember households heads <19 - if any - need to be included, make sure they're not counted again as children 14-18.

```
compute head = 1.
exe.
```

*otherads = all adults in household who are not the head.

```
compute otherads = DeHGads-head.
exe.
```

```
fre otherads.
```

*check total number in household is the same.

```
compute allpers2 = otherads+head+DeHGkds2+DeHGkds3.
cro allpers by allpers2.
```

* perfect X-tab OK.

** Allocate score.

```
compute temp1 = head*0.67.
compute temp2 = otherads*0.33.
compute temp3 = DeHGkds3 *0.33.
compute temp4 = DeHGkds2*0.20.
exe.
```

```
compute DeMcClem = temp1+temp2+temp3+temp4.
exe.
```

* 'OECD' added in var lab from BC1 Sw4 (archiving run) to avoid confusion as not true McClement score

```
VARIABLE LABEL DeMcClem 'De OECD-Modified McClements hhold score for equiv. income'.
```

* SPSS default format now 0 dp instead of 2 dp before

* so need to add FORMATS command to get 2 dp

* no missing values to tidy up OK

```
FORMATS DeMcClem (F3.2).
fre DeMcClem.
```

* range 0.87 to 4.51.

**Create equivalised household income variable

**Copied syntax used for BC1 Sw1 - amended to use BC2 Sw3 variable names

**SEE BC2 SW1 EQ INCOME SYNTAX FOR DISCUSSION ON SELECTION OF LOWER AND UPPER MID-POINT SELECTIONS

**IN SYNTAX FILE: "3_Derive_4_IncomeToEnd.sps"

dataset close all.
get file v25.

**midinc is the MIDDLE VALUE FOR INCOME GROUPS, used in the absense of actual income for each household.

* (same syntax used by Sarah)

fre MeWinc09.

**midinc is the MIDDLE VALUE FOR INCOME GROUPS, used in the absense of actual income for each household.

* (same syntax used by Sarah)

*nb returned to 17 category ~ syntax picked up from BC1 Sweep8

```
COMPUTE midinc=-1.  
RECODE MeWinc09 (1 = 3500) (2 = 4999.5) (3 = 6999.5) (4 = 8999.5) (5 = 10999.5)  
(6 = 13499.5) (7 = 16499.5) (8 = 18999.5) (9 = 21499.5) (10 = 24499.5) (11 = 27499.5)  
(12 = 30499.5) (13 = 34999.5) (14 = 40999.5) (15 = 46999.5) (16 = 52999.5)  
(17 = 60000) INTO midinc.  
fre midinc.
```

```
temp.  
select if midinc=-1.  
freq MeWinc09 .
```

* 442 values coded '-1': they correspond to the don't know refused and no drop-outs '-1' .

```
COMPUTE DeEqvinc=-1.  
IF (midinc>0) DeEqvinc=midinc/DeMcClem.  
exe.
```

```
FORMATS DeEqvinc (F8.1).  
FREQ DeEqvinc.
```

```
*Statistics  
*DeEqvinc  
*N      Valid    4434  
        *Missing  0  
*Mean           23976.070  
*Median         24200.685  
*Std. Deviation 14800.5178  
*Range          68966.5  
*Minimum        -1.0  
*Maximum        68965.5
```

*442 coded -1 like midinc; check range by setting -1s as missing

```
missing values Deeqvinc (-1).  
fre DeEqvinc.
```

```
*Statistics  
*DeEqvinc  
*N      Valid    3992  
        *Missing  442  
*Mean           26630.846  
*Median         25624.688  
*Std. Deviation 13137.5886  
*Range          67678.8  
*Minimum        1286.8  
*Maximum        68965.5
```

```
missing values DeEqvinc ().
```

** get QUINTILES.

* (only negative value for midinc and DeEqvinc are the "-1" for N/A)

```
temp.  
select if (midinc>0).  
fre midinc DeEqvinc /ntiles = 5  
/ FORMAT=NOTABLE.
```

* line " / Format=Notable" command added to get only the 1st stat table, the only one of interest:

```
*Statistics  
*midinc DeEqvinc  
*N Valid 3992 3992  
*Missing 0 0  
*Percentiles  
*20 18999.5000 13402.021  
*40 30499.5000 21785.357  
*60 46999.5000 33124.688  
*80 60000.0000 42857.143
```

* using Sarah's syntax details (originally from BC1 Sw3) from now on, with values adapted to BC2 Sw3:

* syntax below uses boundaries from LEFT column of Statistics table above:
* command RECODE is using overlapping boundaries - order of priority determined by order in RECODE sequence
* sequence below means that 18999.5000 originally included in lower boundary 0-18999.50 is *NOT* recoded as 2 in the next interval (2nd step DOES NOT overwrite the 1st one)
* assume that this is what is wanted (otherwise would need backward recoding, i.e. starting from highest interval first)

```
RECODE midinc (0 thru 18999.50=1) (18999.50 thru 30499.50=2)  
(30499.50 thru 46999.50=3) (46999.50 thru 60000.00=4) (60000.00 thru hi=5) (else = -1)  
INTO mid5.  
EXE.
```

* syntax below uses boundaries from RIGHT column of Statistics table - NOTE: when using values from table above, only c.5% of cases were in the 5th quintile
*This was due to a large number of cases having the value=42857.143 (n=667), thus being included in the 4th quintile.
*By changing the border btw the 4th and 5th quintile to 42857.0, these 667 cases were included in the top/5th quintile instead.
*After this correction, however, only 17% of cases fell into the 4th quintile. To correct for this, the border btw 3rd and 4th quintile was also adjusted,
*in order to to add 50 cases with value=33127.7 to the 4th rather than the 3rd quintile (i.e. the value was adjusted from 33124.688 to 33124.600).
** 13402.021 coded as 1 CANNOT BE recoded as 2 in the 2nd step as value no longer exists in dataset due to 1st step)

```
RECODE DeEqvinc (0 thru 13402.021=1) (13402.021 thru 21785.357 = 2)  
(21785.357 thru 33124.600 = 3) (33124.600 thru 42857.0 =4) (42857.0 thru hi = 5) (else = -1)  
INTO DeEqv5.  
EXE.
```

```
fre deeqv5.
```

```
VARIABLE LABEL DeEqvinc 'De Equivalised income'.  
VARIABLE LABEL DeEqv5 'De Equivalised income - quintiles'.  
VALUE LABELS DeEqvinc  
-1 'Item not applicable'.
```

* The value labels for DeEqv5 below use rounded boundaries

```
VALUE LABELS DeEqv5  
-1 'Item not applicable'  
1 'Bottom Quintile - up to £13,402'  
2 '2nd Quintile £13,402 to £21,785'  
3 '3rd Quintile £21,786 to £33,125'  
4 '4th Quintile £33,126 to £42,857'  
5 'Top Quintile - over £42,857'.  
MISSING VALUES DeEqvinc DeEqv5 (-1).  
FORMATS DeEqv5 (F2.0).
```

FREQ DeEqv5 DeEqvinc.

Housing & Accommodation

Derived variables:

DeZten02 De Tenure at BC2 Sweep3 - banded

***** TENURE banded DV.

* (syntax copied from weight prep files).

FREQ Zhou05.

* = full tenure: asked all, 11 N/A = drop-outs.

RENAME VARS (Zhou05=MeZhou05).

VAR LABS MeZhou05 'Me - Household own or rent accommodation'.

ADD VAL LABS MeZhou05

5 'Rents - privately, unfurnished'.

FREQ MeZhou05.

* attention: answer cat '10' old 'Other' now 10 'Part of co-ownership scheme' + 11 'Partly own/partly rent' + 12 'Crofting'

** BANDED TENURE DV:

Recode MeZhou05 (1 thru 2 = 1) (3 thru 4 = 2) (5 thru 6 = 3) (7 thru hi = 4) (-1=-3) (else = copy) into DeZten02.

Exe.

var labs DeZten02 'De Tenure at BC2 Sweep1 - banded'.

val labs DeZten02 1 'Owner occupied' 2 'Social rented' 3 'Private rented' 4 'Other'

-9 'Refusal' -8 'Don t Know' -3 'Information not available: incomplete interview'.

Missing value DeZten02 (-9 thru -1).

FORMATS DeZten02 (F2.0).

fre DeZten02.

BMI, Height & Weight

Derived variables:

DeZeht05 De - Childs measured or est height (cm)

DeZwkg01 De - Childs final measured weight

DeZewt05 We - Final measured or estimated weight (kg)

DeBMI De BMI (reliable measurements only)

DeBMI_b De BMI - incl. unreliable measurements

DeUKbmi De UK BMI national classification standards

DeINTbmi Dh International BMI cut-offs

DeINTbmi2 De BMI status (ovrwt inc. obese) - international cut-offs

DeINTbmi3 Dh BMI status (non-obese vs obese) - international cut-offs

DeISDbmc De ISD BMI centiles classification

DeISDbmi De Childrens BMI - 5 groups ISD classification

DeISDHwt De Study child weight within/outwith ISD healthy range
DeISDovW De Study child overweight, including obese (ISD)

****BMlok**
 Compute BMlok=-1 .
 IF (WeXhei01=1) AND ((WeXwei01=0) OR (WeXwei01=1)) BMlok=1.
 Exe.
 freq BMlok.

**** BMI (= weight in kg divided by squared[height in METRES, not cm] hence 100x100 in numerator below).**
 COMPUTE DeBMI_b=-1.
 compute DeBMI_b=(DeZwkg01*100*100)/(WeZhcm01*WeZhcm01).
 VARIABLE LABELS DeBMI_b "De BMI - incl. unreliable measurements".
 missing values DeBMI_b (-1).

FORMATS DeBMI_b (F4.1).
 freq DeBMI_b.

* 4301 cases (as expected from 'BMlok': OK)
 * range 7.6 to 39.9 .

**** CORRECTING this expanded BMI value DeBMI_b by removing unreliable cases (attention: nothing to do with outside height or weight ranges)**
**** -> new DV = final BMI 'DeBMI'.**

FREQ WeXhei14 WeXwei19.

RECODE DeBMI_b (else=copy) INTO DeBMI.
 Exe.
 IF ((WeXhei14=3) OR (WeXwei19=3)) AND (DeBMI_b>0) DeBMI=-2.
 Exe.

VARIABLE LABELS DeBMI "De BMI (reliable measurements only)".

ADD VAL LABS DeBMI
 -1 'Not applicable' -2 'Unreliable height or weight measurement'.
 missing values DeBMI (-9 thru -1).
 FORMATS DeBMI (F5.2).
 freq DeBMI.

****Child's AGE at time of interview in years (86400 (= 24 hrs x 3600 sec/hr) x 365.25 days in one year = nbr of seconds in a year ; GUS keeps dates as seconds so CDob_ & IntDate_ are in seconds, so final ratio = nbr of years).**

COMPUTE intexage = -1 .
 IF (cdob_2 > 0) intexage=((IntDate_-CDob_2)/(86400*365.25)) .
 VARIABLE LABELS intexage "(D) Exact child age in years at interview".
 missing values intexage (-1).
 FORMATS intexage (F5.2).
 freq intexage.

***** BMI CUT OFFS 1990 UK BMI reference
 ***** no longer used in SHeS.

* Sw3: children's age around 4.5yr to 5.5yr as birth cohort only, so syntax blocks reduced to those actually needed

DeChAgMth De Study childs age at interview (months)						
		Frequency	Percent	Valid Percent	Cumulative Percent	
Valid	57	58	1.3	1.3	1.3	
		58	2617	59.0	59.0	60.3
		59	1506	34.0	34.0	94.3
		60	205	4.6	4.6	98.9
		61	40	.9	.9	99.8

	62		6	.1	.1	100.0
	63		1	.0	.0	100.0
	64		1	.0	.0	100.0
Total		4434		100.0	100.0	

rename variables sex=MeHGsx1.

DO IF (BMlok=1) AND (WeXhei14<>3) AND (WeXwei19<>3).

COMPUTE DeUKbmi=0.

IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND DeBMI<=14.01 DeUKbmi=1.

IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND DeBMI<=13.71 DeUKbmi=1.

IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND (DeBMI>14.01 AND DeBMI<17.13)
DeUKbmi=2.

IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND (DeBMI>13.71 AND DeBMI<17.23)
DeUKbmi=2.

IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND (DeBMI>=17.13 AND DeBMI<18.08)
DeUKbmi=3.

IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND (DeBMI>=17.23 AND DeBMI<18.32)
DeUKbmi=3.

IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND DeBMI>=18.08 DeUKbmi=4.

IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND DeBMI>=18.32 DeUKbmi=4.

*

IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND DeBMI<=13.90 DeUKbmi=1.

IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND DeBMI<=13.57 DeUKbmi=1.

IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND (DeBMI>13.90 AND DeBMI<17.01)
DeUKbmi=2.

IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND (DeBMI>13.57 AND DeBMI<17.17)
DeUKbmi=2.

IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND (DeBMI>=17.01 AND DeBMI<17.97)
DeUKbmi=3.

IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND (DeBMI>=17.17 AND DeBMI<18.31)
DeUKbmi=3.

IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND DeBMI>=17.97 DeUKbmi=4.

IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND DeBMI>=18.31 DeUKbmi=4.

*

IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND DeBMI<=13.81 DeUKbmi=1.

IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND DeBMI<=13.45 DeUKbmi=1.

IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND (DeBMI>13.81 AND DeBMI<16.96)
DeUKbmi=2.

IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND (DeBMI>13.45 AND DeBMI<17.16)
DeUKbmi=2.

IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND (DeBMI>=16.96 AND DeBMI<17.95)
DeUKbmi=3.

IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND (DeBMI>=17.16 AND DeBMI<18.35)
DeUKbmi=3.

IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND DeBMI>=17.95 DeUKbmi=4.

IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND DeBMI>=18.35 DeUKbmi=4.

*

IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND DeBMI<=13.75 DeUKbmi=1.

IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND DeBMI<=13.37 DeUKbmi=1.

IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND (DeBMI>13.75 AND DeBMI<16.96)
DeUKbmi=2.

IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND (DeBMI>13.37 AND DeBMI<17.21)
DeUKbmi=2.

IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND (DeBMI>=16.96 AND DeBMI<17.99)
DeUKbmi=3.

IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND (DeBMI>=17.21 AND DeBMI<18.46)
DeUKbmi=3.

IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND DeBMI>=17.99 DeUKbmi=4.

IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND DeBMI>=18.46 DeUKbmi=4.

*

IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND DeBMI<=13.69 DeUKbmi=1.

IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND DeBMI<=13.32 DeUKbmi=1.

IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND (DeBMI>13.69 AND DeBMI<17.01)
DeUKbmi=2.

IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND (DeBMI>13.32 AND DeBMI<17.32)
DeUKbmi=2.

IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND (DeBMI>=17.01 AND DeBMI<18.10)
DeUKbmi=3.


```
IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND (DeBMI>=17.32 AND DeBMI<18.65)
DeUKbmi=3.
IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND DeBMI>=18.10 DeUKbmi=4.
IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND DeBMI>=18.65 DeUKbmi=4.
*.
ELSE.
compute DeUKbmi = -1.
END IF.
```

```
MISSING VALUES DeBMI ().
IF (DeBMI=-2) DeUKbmi=-2.
RECODE DeUKbmi (sysmis=-1).
Exe.
```

```
VAR LAB DeUKbmi 'De UK BMI national classification standards'.
Value labels DeUKbmi
1 'Underweight (less than or equal to 5th percentile)'
2 'Healthy weight (greater than 5th centile and less than 85th centile)'
3 'Overweight (greater than or equal to 85th centile and less than 95th centile)'
4 'Obese (greater than or equal to 95th centile)'
-1 'Not applicable: height and/or weight not measured'
-2 'Unreliable height or weight measurement'.
```

```
MISSING VALUES DeUKbmi DeBMI (-9 thru -1).
FORMATS DeUKbmi (F2.0).
```

```
freq DeUKbmi.
```

```
* temporary.
* select if DeUKbmi=0.
* list serial MeHGsx1 BMlok WeXhei14 WeXwei19 intexage DeUKbmi DeBMI.
```

```
***** BMI CUT OFFS * INTERNATIONAL * BMI reference *****
*** only 85th & 95th percentiles in Excel file (and could not find a table of full percentile cut offs on internet
from International Obesity Taskforce = IOTF or other)
** SO NOT AFFECTED BY CORRECTIONS ON LOWER INTERVALS (using 5% centile) as carried out
above on UK BMI
```

```
DO IF (BMlok=1) AND (WeXhei14<>3) AND (WeXwei19<>3).
COMPUTE DeINTbmi=0.
```

```
IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND DeBMI<17.55 DeINTbmi=1.
IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND DeBMI<17.28 DeINTbmi=1.
IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND (DeBMI>=17.55 AND DeBMI<19.29)
DeINTbmi=2.
IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND (DeBMI>=17.28 AND DeBMI<19.15)
DeINTbmi=2.
IF MeHGsx1=1 AND (intexage>=4 AND intexage<4.5) AND DeBMI>=19.29 DeINTbmi=3.
IF MeHGsx1=2 AND (intexage>=4 AND intexage<4.5) AND DeBMI>=19.15 DeINTbmi=3.
*.
IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND DeBMI<17.47 DeINTbmi=1.
IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND DeBMI<17.19 DeINTbmi=1.
IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND (DeBMI>=17.47 AND DeBMI<19.26)
DeINTbmi=2.
IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND (DeBMI>=17.19 AND DeBMI<19.12)
DeINTbmi=2.
IF MeHGsx1=1 AND (intexage>=4.5 AND intexage<5) AND DeBMI>=19.26 DeINTbmi=3.
IF MeHGsx1=2 AND (intexage>=4.5 AND intexage<5) AND DeBMI>=19.12 DeINTbmi=3.
*.
IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND DeBMI<17.42 DeINTbmi=1.
IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND DeBMI<17.15 DeINTbmi=1.
IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND (DeBMI>=17.42 AND DeBMI<19.30)
DeINTbmi=2.
IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND (DeBMI>=17.15 AND DeBMI<19.17)
DeINTbmi=2.
IF MeHGsx1=1 AND (intexage>=5 AND intexage<5.5) AND DeBMI>=19.30 DeINTbmi=3.
IF MeHGsx1=2 AND (intexage>=5 AND intexage<5.5) AND DeBMI>=19.17 DeINTbmi=3.
*.
```

```

IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND DeBMI<17.45 DeINTbmi=1.
IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND DeBMI<17.20 DeINTbmi=1.
IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND (DeBMI>=17.45 AND DeBMI<19.47)
DeINTbmi=2.
IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND (DeBMI>=17.20 AND DeBMI<19.34)
DeINTbmi=2.
IF MeHGsx1=1 AND (intexage>=5.5 AND intexage<6) AND DeBMI>=19.47 DeINTbmi=3.
IF MeHGsx1=2 AND (intexage>=5.5 AND intexage<6) AND DeBMI>=19.34 DeINTbmi=3.
*
IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND DeBMI<17.55 DeINTbmi=1.
IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND DeBMI<17.34 DeINTbmi=1.
IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND (DeBMI>=17.55 AND DeBMI<19.78)
DeINTbmi=2.
IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND (DeBMI>=17.34 AND DeBMI<19.65)
DeINTbmi=2.
IF MeHGsx1=1 AND (intexage>=6 AND intexage<6.5) AND DeBMI>=19.78 DeINTbmi=3.
IF MeHGsx1=2 AND (intexage>=6 AND intexage<6.5) AND DeBMI>=19.65 DeINTbmi=3.
*
ELSE.
COMPUTE DeINTbmi=-1.
END IF.

```

```

MISSING VALUES DeBMI ().
IF (DeBMI=-2) DeINTbmi=-2.
RECODE DeINTbmi (sysmis=-1).
Exe.

```

```

VAR LAB DeINTbmi 'Dh International BMI cut-offs'.
value labels DeINTbmi
1 'Normal-weight (less than 85th percentile)'
2 'Over-weight (greater than or equal to 85th centile and less than 95th centile)'
3 'Obese (greater than or equal to 95th centile)'
-1 'Not applicable' -2 'Unreliable height or weight measurement'.
MISSING VALUES DeINTbmi DeBMI (-9 thru -1).
FORMATS DeINTbmi (F2.0).
FREQ DeINTbmi.

```

```
cro DeINTbmi by DeUKbmi
```

```

* checking cases where UK overweight and international is normal weight.
temp.
select if DeUKbmi =3 and DeINTbmi =1.
list MeHGsx1 intexage DeBMI.

```

```
*** alternative groupings for international BMI DV:
```

```

RECODE DeINTbmi (1=1) (2 thru 3=2)(else=copy) INTO DeINTbmi2.
exe.
VAR LAB DeINTbmi2 'De BMI status (ovrwt incl. obese) - international cut-offs'.
VAL LAB DeINTbmi2
1 'Neither overweight nor obese'
2 'Overweight incl. obese'
-1 'Not applicable'
-2 'Unreliable height or weight measurement'.

```

```

RECODE DeINTbmi (1 thru 2=1) (3=2)(else=copy) INTO DeINTbmi3.
exe.
VAR LAB DeINTbmi3 'Dh BMI status (non-obese vs obese) - international cut-offs'.
VAL LAB DeINTbmi3
1 'Non-obese'
2 'Obese'
-1 'Not applicable'
-2 'Unreliable height or weight measurement'.

```

```

MISSING VALUES DeINTbmi2 DeINTbmi3 (-9 thru -1).
FORMATS DeINTbmi2 DeINTbmi3 (F2.0).
FREQ DeINTbmi2 DeINTbmi3.

```

```
***** BMI CUT OFFS * ISD * BMI reference *****.
```

dataset close all.
get file='M:\Current projects\GUS\Secure\Data Management\Data\Cohort 2\Sweep 3\data
cleaning\BC2Sw3 Derived 21.sav'.

** (IntDate_-CDob_2) is the child's age in seconds
* so to get the child's age in days, we need to divide this by the number of seconds in a day = 86400 (24
hrs x 3600 sec/hr)

```
COMPUTE AGE_DAYS = -1 .  
IF (cdob_2 > 0) AGE_DAYS =((IntDate_-CDob_2)/86400) .  
VARIABLE LABELS AGE_DAYS "(D) Exact child age in days at interview for ISD BMI calc".  
missing values AGE_DAYS (-1).  
freq AGE_DAYS.
```

* need to rename TEMPORARILY 'MeHGsx1' back to 'SEX' and 'Debmi' as 'BMI' to match ISD syntax
provided:
* JC note in SW7 there was already a var called SEX so had to rename it first.

```
RENAME VARS (MeHGsx1=SEX) (Debmi=BMI).
```

```
***** ISD syntax starts here*****  
***** picked up syntax from SHeS12 - revised by ISD to include wider range of BMIs.
```

* * Create variable agemth - to show child's age in months at date of interview.

```
compute agemth=12*AGE_DAYS/365.25.  
exe.
```

```
numeric ageyr (f6.3).  
compute ageyr=rnd(1000*trun(agemth)/12)/1000.  
exe.
```

```
*****.
```

* this now used the ISD lookup files to do the calculations.
* ISD lookup file is in SHeS folders (lmsbmi.sav).

*The look-up file is matched in using the lowest Whole month converted to years.
*The corresponding L,M,S are the lowest (LO) values used in the interpolation.

sort cases by ageyr.

```
match files/ file=*  
/table={table}  
LOOKUP FILE\lmsbmi.sav'  
/rename=(LM=LMLO) (MM=MMLO) (SM=SMLO) (LF=LFLO) (MF=MFLO) (SF=SFLO)  
/by=ageyr.
```

```
rename variables (ageyr=agelo).  
execute.
```

* * Child's age in months will lie between two ages in Whole months in the lookup table.
* * Line below calculates the next highest Whole month and converts to years.

```
compute ageyr=rnd(1000*(trun(agemth)+1)/12)/1000.  
execute.
```

* * The look-up file is matched in using the highest Whole month converted to years.
* * The corresponding L,M,S are the lowest (HI) values used in the interpolation.

```
match files/ file=*  
/table={table}  
LOOKUP FILE\lmsbmi.sav'  
/rename=(LM=LMHI) (MM=MMHI) (SM=SMHI) (LF=LFHI) (MF=MFHI) (SF=SFHI)  
/by=ageyr.
```

```
rename variables (AGEYR=AGEHI).  
execute.
```

*** new syntax from ISD here - age in years calculated to 2 decimal places.

*new bit - for interpolation calculate age in years to 2 decimal places.

compute ageyrs2decimal=rd((agemth/12),0.01).

```
DO IF (SEX EQ 1).
  COMPUTE LINT=LMHI-((AGEHI-(ageyrs2decimal))/(AGEHI-AGELO))*(LMHI-LMLO).
  COMPUTE MINT=MMHI-((AGEHI-(ageyrs2decimal))/(AGEHI-AGELO))*(MMHI-MMLO).
  COMPUTE SINT=SMHI-((AGEHI-(ageyrs2decimal))/(AGEHI-AGELO))*(SMHI-SMLO).
  ELSE IF (SEX EQ 2).
  COMPUTE LINT=LFHI-((AGEHI-(ageyrs2decimal))/(AGEHI-AGELO))*(LFHI-LFLO).
  COMPUTE MINT=MFHI-((AGEHI-(ageyrs2decimal))/(AGEHI-AGELO))*(MFHI-MFLO).
  COMPUTE SINT=SFHI-((AGEHI-(ageyrs2decimal))/(AGEHI-AGELO))*(SFHI-SFLO).
  END IF.
exe.
```

missing values BMI (-1, -2).

freq BMI.

* * Calculate SD (standardised) score.

*bmi rounded to 2 decimals.
compute bmi_2d=rd(bmi,0.01).

* Calculate SD (standardised) score using BMI to two decimal places.
compute SDS2=(((bmi_2d/MINT)**LINT)-1)/(LINT*SINT).
execute.

*compute SDS to 2 decimal places.
compute SDS=rd(SDS2,0.01).

* * Calculate centiles.

numeric cent (f4.1).
compute cent=100*CDF.Normal(sds,0,1).
execute.

* select out those outwith the range deemed to be real.
compute keep_new=0.
if range(sds,-7,7) keep_new=1.
execute.

freq keep_new.

* FROM PREVIOUS SYNTAX create centiles groups cent.

recode cent (lo thru 2.0=1) (2.0 thru 5.0=2) (5.0 thru 85.0=3)
(85.0 thru 95.0=4) (95.0 thru 98.0=5) (98.0 thru hi=6) into centiles.
execute.

* * The above syntax is not accurate for cases that have a "cent" of exactly 85.0, 95.0 or 98.0.

* * Therefore we need to recode these exceptional cases as follows.

if (cent = 85.0) centiles = 4.
if (cent = 95.0) centiles = 5.
if (cent = 98.0) centiles = 6.
exe.

IF keep_new=0 centiles=-3.
IF missing (BMI) centiles=-1.
freq centiles.

value labels centiles 1"less than or equal to 2nd centile"
2"greater than 2nd and less than or equal to 5th centile"
3"greater than 5th centile less than 85th centile"
4"greater than or equal to 85th centile and less than 95th centile"
5"greater than or equal to 95th and less than 98th centiles"
6"greater than or equal to 98th centile"

-1 "no BMI"
-3 "out of range".

format centiles (f8.0).
freq centiles.

rename variables (centiles= DeISDbmc).

var labels DeISDbmc "De ISD BMI centiles classification".

* 5 category version.

recode DeISDbmc (1=1)(2,3=2)(4=3)(5=4)(6=5)(else=copy) into DeISDbmi.
var label DeISDbmi "De Childrens BMI - 5 groups ISD classification".

cro DeISDbmi by DeUKbmi.

cro DeISDbmi by DeINTbmi.

value labels DeISDbmi
1 "Underweight (<= 2nd %ile)"
2 "Healthy weight (> 2nd %ile to < 85th %ile)"
3 "Overweight (>= 85th %ile to < 95th %ile)"
4 "Obese (>= 95th %ile to <98th %ile)"
5 "Morbidly obese (>= 98th %ile)"
-1 "no BMI"
-3 "out of range".
format DeISDbmi(f8.0).

freq DeISDbmi.

RENAME VARS (SEX=MeHGsx1) (BMI=DeBMI).

* delete working variables.

* adding extra DVs

*** create variable for Outwith healthy weight (<= 5th %ile or >=85th %ile).

* (variable name & label amended to follow GUS conventions)

freq DeISDbmi.

recode DeISDbmi (2=0) (1,3,4,5=1) (else=copy) into DeISDHWt.

var label DeISDHWt "De Study child weight within/outwith ISD healthy range".
value labels DeISDHWt 0 "weight in healthy range (> 5th %ile to < 85th %ile)"
1 "weight outwith healthy range (<= 5th %ile or >=85th %ile)"
-1 "Not applicable: no height or weight measurement"
-2 "Unreliable height or weight measurement".

FORMATS DeISDHWt (F2.0).

MISSING VALUES DeISDHWt DeBMI (-9 thru -1).

FREQ DeISDHWt.

cro DeISDbmi by DeISDHWt.

*** create variable for Overweight including obese (>=85th %ile).

freq DeISDbmi .

recode DeISDbmi (1,2=0) (3,4,5=1) (else=copy) into DeISDovW.

var label DeISDovW "De Study child overweight, including obese (ISD)".

value labels DeISDovW 0 "not overweight or obese (< 85th %ile)"
1 "overweight or obese (>= 95th %ile)"
-1 "Not applicable: no height or weight measurement"
-2 "Unreliable height or weight measurement".

FORMATS DeISDovW (F2.0).

MISSING VALUES DeISDovW(-9 thru -1).

FREQ DeISDovW.
