Centre for Longitudinal Studies

CLS Cohort Studies

Data Note 2007/2

Programming
Employment Histories in NCDS 4, 5 and 6
1974-2000

Kelly Ward

Centre for Longitudinal Studies
Bedford Group for Lifecourse and
Statistical Studies
Institute of Education
20 Bedford Way
London WC1H 0AL
Tel: 020 7612 6860
Fax: 020 7612 6880
Email cls@ioe.ac.uk
Web http://www.cls.ioe.ac.uk

CLS Cohort Studies Data Note 2007/02

Programming Employment Histories in NCDS 4, 5 and 6 1974-2000

Kelly Ward

Centre for Longitudinal Studies, Institute of Education.

1. Introduction

The following work was undertaken as part of a Gender Network project with Professors Shirley Dex and Heather Joshi, examining the occupational mobility of men and women over the lifecourse, using the retrospective employment histories from NCDS 4, 5 and 6.

This Data Note has been designed to provide additional information along side Data Note 1-2007 '*Employment histories in NCDS 4, 5, and 6*'. In brief, Data Note 1 provides an overview of the cleaning undertaken on the NCDS 4 and 5 job spells. This cleaning specifically targeted the information collected retrospectively at each sweep on job spells focusing mainly on the start (month and year) and end (month and year) dates of each spell.

This Data Note outlines the further extensive work undertaken to programme these job spells into an employment history data file. SPSS syntax is provided to show the process of programming all of the job spells collected from survey sweeps 4, 5 and 6 using SPSS vector and looping syntax.

This Data Note is split into three sections as follows:

2. The first section provides an example of the work carried out to minimise as much of the missing data as possible.
3. The second section details the work undertaken to programme SPSS to hold the employment histories to a readable and workable format.
4. The third section highlights the remainder of the work carried out on the employment history file.

Section one provides example syntax only, whilst the detailed SPSS syntax relevant to Sections 2 and 3, and used in the project, is contained in the Appendices found at the end of this Data Note.

In the discussion below we use the terminology as follows:

a. Job spell – which is a job but also implies a period of employment.
b. Non-employment spell – is the gap between discontinuous job spells.
c. Hours status – is whether a job spell has been coded either working full time (code 1) or part time (code 2).

2. Recoding missing data into valid responses

As detailed in Data Note 1, it was important to recode any system missing values in each job spell into ‘valid’ missing responses, such as, ‘Not Applicable’ (-1) or ‘Don’t Know’ (-8). Having accomplished this it became apparent that there were a vast number of ‘Don’t Know’ (-8) values in mainly the start and end *months* of each job spell. A smaller number of -8 values were found in the start and end *years* of each job spell. For the purposes of our project we decided to convert as many of these -8 values into valid responses as possible. The work undertaken on this has not been and will not be fully documented, so other researchers need to make their own decisions on these issues; it was deemed important to provide an example of the work carried out for the benefit of other users. This work was carried out for the purposes of our project only and therefore the dataset deposited in the Data Archive has the original -8 (Don’t Know) values and no recoded values.

Recoding the -8 values required thorough checks of all the start and end month and year variables for each job spell. The sample syntax (see Appendix One) was used to select variables that had a -8 (Don’t Know) response and check whether those individual cases had any previous and post job details (such as valid month and year dates). Any additional dates were then used to establish whether a valid year/month date (start and end) could be derived for spells with these missing dates. For example, an end job date (month and year) could often be the same as the (missing) start job date (month and year) of the next job spell if there were no intervening periods of non-employment recorded. Further checks of the non-employment spells were carried out to examine this. The reference to the end dates of non-employment spells also indicated dates of labour market re-entry.

Even after this inspection (using the non-employment spells) there were still a number of job spells that had missing (don’t know) values. For these spells we adopted a policy to create a month date (start or end) to have closure on a particular spell. For example if job spell 1 started in January 1980 and the end date of this spell was also 1980 but the end month was missing (i.e don’t know) then the policy would be to assign the missing end month a value that took the mid-point of the year, i.e. June. Other cases that were unclear or had relatively little information remained as -8 (don’t know). Examples are provided in Appendix One of the syntax used to make these changes.

3. Programming SPSS

Once all the variables from the different NCDS datasets (4, 5, 6) are merged together (using the link variable *serial*) more cases will appear as system missing. These values will refer to those who did not complete that interview sweep. It is important to ensure that all missing values, including the system missing, use the same categories.

Working with employment histories means working with a number of job spells that contain valid month and year dates (start and end) as well as information relating to the hours and occupation category. To begin programming SPSS, it is necessary to run a command that will set up an SPSS vector of 'empty' variables. In this employment history we needed 25 empty variables that we could use to load the information contained in each job spell. To begin with we loaded the start month of each job spell taken from NCDS 4, 5 and 6, of which there were 25 spells in total. Two job spells are taken from NCDS 4, 12 from NCDS 5 and 11 from NCDS 6. SPSS sets up these variables by using the following command which will compute 25 variables that SPSS will call mth(1), mth(2), mth(3)…..mth(25). All variables in the mth vector are initially set to 0 (missing values need to be turned off in SPSS for all of this syntax; this is shown in more detail in Appendix Two).

```
vector mth(25).
loop #a=1 to 25.
+ compute mth(#a)=0.
end loop.
```

Once we have defined the number of empty variables that we need in a vector we must then load in the information that we require. For example, this means that we must tell SPSS which start month for each spell belongs in which empty vector variable defined in the above box. This process is continued until we reach the end of the available job spell data. The following syntax provides an example of this. Mth(1) and mth(2) relate to the start month of the job spells taken from NCDS 4. Mth(3) to mth14 relate to the start month of the job spells taken from NCDS 5 and mth(15) onwards relate to NCDS 6.

NCDS 6 differed slightly because the non-employment spells were collected alongside the job spells. Therefore when taking the start month of a job spell from NCDS 6 it was necessary to ensure these month values were only copied into the vector that related to a job spell (and if the spell was a non-employment spell, it was passed over).

Note: When working with vectors SPSS often requires that parts of the variable name are in parentheses, for example Mth(1), however once you have defined the vector as outlined in the example above SPSS will load the information shown below without the need for parentheses.

```
compute mth1=dmjbstm1.
if (dmjbstm1=0) mth1=-1.
compute mth2=dmjm2.
if (dmjm2=0) mth2=-1.
compute mth3=emjbstm1.
if (emjbstm1=0) mth3=-1.
compute mth4=emjbstm2.
if (emjbstm2=0) mth4=-1.
compute mth5=emjbstm3.
if (emjbstm3=0) mth5=-1.
compute mth6=emjbstm4.
if (emjbstm4=0) mth6=-1.
compute mth7=emjbstm5.
if (emjbstm5=0) mth7=-1.
compute mth8=emjbstm6.
if (emjbstm6=0) mth8=-1.
compute mth9=emjbstm7.
if (emjbstm7=0) mth9=-1.
compute mth10=emjbstm8.
if (emjbstm8=0) mth10=-1.
compute mth11=emjbstm9.
if (emjbstm9=0) mth11=-1.
compute mth12=emjbstm10.
if (emjbstm10=0) mth12=-1.
compute mth13=emjbstm11.
if (emjbstm11=0) mth13=-1.
compute mth14=emjbstm12.
if (emjbstm12=0) mth14=-1.
compute mth15=startm10.
if (activi10>4) mth15=-1.
if (startm10=0) mth15=-1.
compute mth16=startmo9.
if (activit9>4) mth16=-1.
if (startmo9=0) mth16=-1.
compute mth17=startmo8.
if (activit8>4) mth17=-1.
if (startmo8=0) mth17=-1.
compute mth18=startmo7.
if (activit7>4) mth18=-1.
if (startmo7=0) mth18=-1.
compute mth19=startmo6.
if (activit6>4) mth19=-1.
if (startmo6=0) mth19=-1.
compute mth20=startmo5.
if (activit5>4) mth20=-1.
if (startmo5=0) mth20=-1.
compute mth21=startmo4.
if (activit4>4) mth21=-1.
if (startmo4=0) mth21=-1.
compute mth22=startmo3.
if (activit3>4) mth22=-1.
if (startmo3=0) mth22=-1.
compute mth23=startmo2.
if (activit2>4) mth23=-1.
if (startmo2=0) mth23=-1.
compute mth24=startmo.
if (activity>4) mth24=-1.
if (startmo=0) mth24=-1.
compute mth25=cstartmo.
if (econact>4) mth25=-1.
execute.

variable label mth1 "ncds 4 - start month job one".
variable label mth2 "ncds 4 - start month job two".
variable label mth3 "ncds 5 - start month job one".
variable label mth4 "ncds 5 - start month job two".
variable label mth5 "ncds 5 - start month job three".
variable label mth6 "ncds 5 - start month job four".
variable label mth7 "ncds 5 - start month job five".
variable label mth8 "ncds 5 - start month job six".
variable label mth9 "ncds 5 - start month job seven".
variable label mth10 "ncds 5 - start month job eight".
variable label mth11 "ncds 5 - start month job nine".
variable label mth12 "ncds 5 - start month job ten".
variable label mth13 "ncds 5 - start month job eleven".
variable label mth14 "ncds 5 - start month job twelve".
variable label mth15 "ncds 6 - start month job ten".
variable label mth16 "ncds 6 - start month job nine".
variable label mth17 "ncds 6 - start month job eight".
variable label mth18 "ncds 6 - start month job seven".
variable label mth19 "ncds 6 - start month job six".
variable label mth20 "ncds 6 - start month job five".
variable label mth21 "ncds 6 - start month job four".
variable label mth22 "ncds 6 - start month job three".
variable label mth23 "ncds 6 - start month job two".
variable label mth24 "ncds 6 - start month job one".
variable label mth25 "ncds 6 - start month current job".
value labels mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14 mth15 mth16
mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25
0 'doing same job'
-8 'don t know'
-1 'not applicable'.
Execute.
```

Once this syntax has been run, the empty vector variables that were originally set to zero should now contain the start month of all the job spells taken from NCDS 4, 5 and 6. Once this information is loaded into the empty vector variables, the data will not be in any particular chronological order, as information obtained at each sweep can potentially overlap with the previous sweep. It is also possible that respondents did not participate at all three sweeps which will also mean that the vector data at this point will not be chronological and will have missing spells. Therefore the vector should look like this:

Table 1: Start month data

	Details obtained from						
	NCDS 4		**NCDS 5**			**NCDS 6**	
How the original sweep data appears:							
Variable names of the start month variables in each job spell (as loaded into the vector in the example syntax above)	dmjbstm1	dmstm2	emjstm1	emjstm2	emjstm3	startm10	startm11
Month start date – values that should be loaded into the empty vector variables	**2**	**8**	**5**	**-1 (Not Applicable)**	**-1 (Not Applicable)**	**4**	**6**
How the vector data will appear:							
Vector name	Mth(1)	Mth(2)	Mth(3)	Mth(4)	Mth(5)	Mth(16)	Mth(17)
Values found in these vectors should match the variable were the information was obtained, as represented here	**2**	**8**	**5**	**-1 (Not Applicable)**	**-1 (Not Applicable)**	**4**	**6**

Setting up a vector of each job spell's start month represents only the initial stages of programming the employment histories.   Once the start month details had been programmed, the other information about each spell needs also to be stored in its own 25-space vector, namely the spell's start year, hour status, end month and end year dates and the occupation code. This requires a total of 6 vectors of 25 variables in each vector. Throughout each stage, it was necessary to recode all system missing values into a common missing category which I set to -5. Appendix Two provides full details of the syntax used at this stage of the process.

Job spell end dates

The end dates of each job spell were not collected in the NCDS employment histories available in survey sweeps 4, 5 and 6. To identify the end dates of the job spells we decided to use the start date of the following job spell. Empty vectors were declared to load in the end month and year dates for each job spell, and the dates used to load the information were the start dates of a job spell. If a start date represented a spell of non-employment it was still used to identify an end date of a previous job spell. Therefore, if job spell (1) was followed by a spell of non-employment, the start date of the non-employment spell would be used to represent the end date for job spell (1). If job spell (1) was followed by job spell (2), the start date of the job spell (2) would be used to represent the end date of job spell (1).

4. Restructuring the data file

Once all the required variables are programmed into vectors it is important to restructure the files. The first restructuring undertaken was to remove the missing values from the histories (as shown in Table 1, these missing values can crop up in the middle of the employment history).   Restructuring therefore meant re-ordering the dataset so that all 'valid' values for each job spell followed in a chronological order and all missing not applicable values were found at the end of the employment history file.

To ensure that the correct variables were removed I focused on the start year of each job spell and selected on those with valid values greater than 0. If a missing cell was removed using its start year then it was necessary to remove the corresponding information relating to this particular spell from the other vectors i.e. the start month, end month, end year etc. To facilitate this, storage vectors were set up to receive the correct valid responses and these responses were then re-ordered in each vector accordingly. Storage vectors will temporarily hold the information in the SPSS memory but do not set up any further variables.

The next step was to reorder the job spells chronologically, so that if job(3) started in 1989 and job(10) started in 1987, job(10) would be moved within the vector to come before job(3) and the variable names were reordered to continue in the format of job(1), job(2),…. job(25). Again, any reordering that takes place must also carry through to reorder the corresponding details for that spell's position in all of the vectors. So for example the occupation category for job (10) would also need to be moved to be the occupation category for job(3).

To complete the reordering process efficiently, it was necessary to identify the cases where successive spells started in the same year. To find out which spell was earliest, the start and end months of each spell needed to be compared to order the job spells in the correct chronological sequence. For example, if the start year of job spell (1) and job spell (2) were both 1999 and the start months equalled August and March respectively, then the file was reordered again so that the month and year of job spell (2) became job spell 1 and job spell 1 became job spell 2. Due to the complexity of this, I again made use of temporary storage vectors.

At the seam points between NCDS interviews (where the retrospective employment history from NCDS5 (or NCDS6) reached back to NCDS4's (or NCDS5's) interview date, it was possible for duplicate details of job spells to be recorded. Where duplicate cases were identified we kept the data closest to the time of its collection. For example, if a respondent identified a job in NCDS 4 as starting on 9/1980 and this respondent identified this same job in NCDS 5, then the details given in NCDS 5 would be dropped from the employment history and the NCDS4 information about this job spell would be kept.

It is also important to identify the respondents who had said they were still working when the interview took place – these cases are censored at interview. For these cases the end date values were set to -6. Step by step details are outlined in Appendix Three. The final job spell for any individual depends on whether they participated in all three NCDS interview sweeps, or only two or only one of the sweeps. Where the final job spell appears in the 25 space vector record will depend on how many jobs were reported in total across the survey sweeps. A value of -6 in the end dates of a respondent's employment history appears only once for respondents who had entered their most recent job across the three interviews, but the tenure of this job was censored at interview.

Derived Variables

I have included four additional derived variables in the employment history dataset deposited at the Data Archive These variables are as follows:

a) Censored – 0/1 variable which identifies the cases censored at interview (code 1) by picking up all the -6 values. A value of -1 also indicates respondents who had participated at earlier sweeps but had not participated in survey sweep 4, 5 or 6.
b) NCDS4 – A 1/2 variable where code 1 indicates a response in this sweep using the longitudinal sample. Code 2 indicates no data was collected at this sweep.

c) NCDS 5 – A 1/2 variable where code 1 indicates a response in this sweep using the longitudinal sample. Code 2 indicates no data was collected at this sweep.
d) NCDS 6 – A 1/2 variable where code 1 indicates a response in this sweep using the longitudinal sample. Code 2 indicates no data was collected at this sweep.

APPENDIX ONE: Recoding -8 values into valid responses

The syntax below is an example of the start date (month and year) of the first ever job after leaving full-time education. Recoding the -8 (Don’t know) values for these variables was relatively easy as all first-ever jobs were set to the date the respondent would have left full-time education. Any respondents who had continued in education would have had their first-ever job spell coded as a non-employment spell.

START YEAR

```
Compute emjsty1=emjbsty1.
if ((emjbsty1=-8) and (emjbedy1>0) and (n507816>0) and (emjbedy1<=n507816) and (emjbedy1>=74)) emjsty1=74.
if ((emjbsty1=-8) and (emjbedy1>0) and (n507816>0) and (emjbedy1>n507816)) emjsty1=-8.
if ((emjbsty1=-8) and (emjbedy1>0) and (n507816<0) and (emjbedy1>=74)) emjsty1=74.
if ((emjbsty1=-8) and (emjbedy1<=0) and (n507816>0) and (n507816>74)) emjsty1=74.
if ((emjbsty1=-8) and (emjbedy1<=0) and (n507816>0) and (n507816<=74) and (n507820>0)) emjsty1=n507820.
if ((emjbsty1=-8) and (emjbedy1<=0) and (n507816>0) and (n507816<=76) and (n507820<0)) emjsty1=n507816.
if ((emjbsty1=-8) and (emjbedy1<0) and (n507816<0) and (n507820<0)) emjsty1=74.
Variable labels emjsty1 "Main: Job ONE start year".
Value labels emjsty1
-8 'don t know'
-1 'Not applicable'.
Execute.
```

START MONTH

```
Compute Emjstm1=Emjbstm1.
if ((emjbstm1=-8) and (emjsty1>0 and emjsty1<=76) and (emjsty1<emjedy1)) emjstm1=9.
if ((emjbstm1=-8) and (emjsty1>0 and emjsty1<=76) and (emjsty1=emjedy1) and (emjbedm1>0))
emjstm1=RND(mean(emjbedm1,0)).
if ((emjbstm1=-8) and (emjsty1>0 and emjsty1<=76) and (emjsty1=emjedy1) and (emjbedm1<0)) emjstm1=9.
if ((emjbstm1=-8) and (emjsty1>76) and (emjsty1<emjedy1)) emjstm1=6.
if ((emjbstm1=-8) and (emjsty1>76) and (emjsty1=emjedy1) and (emjbedm1>0)) emjstm1=RND(mean(emjbedm1,0)).
if ((emjbstm1=-8) and (emjsty1>76) and (emjsty1=emjedy1) and (emjbedm1<0)) emjstm1=6.
if ((emjbstm1=-8) and (emjedy1=0|emjedy1=-1)) emjstm1=6.
if ((emjbstm1=-8) and (emjsty1>0 and emjsty1<=76) and (emjedy1=-8) and (emjsty2>0) and (emjsty2>emjedy1))
emjstm1=9.
if ((emjbstm1=-8) and (emjsty1>0 and emjsty1<=76) and (emjedy1=-8 and emjsty2=-8)) emjstm1=9.
Variable labels Emjstm1 "Main: Job ONE start month".
Value labels Emjstm1
-8 'Don t know'
-1 'Not applicable'.
Execute.
```

END YEAR

```
Compute emjedy1=emjbedy1.
if ((emjbedy1=-8) and (emjbsty2>0)) emjedy1=emjbsty2.
if ((emjbedy1=-8) and (emjbsty2<0) and (emjbedy2>0) and (emjsty1>0) and (emjbedy2>emjsty1))
emjedy1=RND(((emjbedy2-emjsty1)/2)+emjsty1).
if ((emjbedy1=-8) and (emjbsty2<0) and (emjbedy2>0) and (emjsty1>0) and (emjbedy2=emjsty1))
emjedy1=emjbedy2.
if ((emjbedy1=-8) and (emjbsty2<0) and (emjbedy2<0) and (emjbsty3>0) and (emjsty1>0) and (emjbsty3>emjsty1))
emjedy1=RND(((emjbsty3-emjsty1)/2)+emjsty1).
if ((emjbedy1=-8) and (emjbsty2<0) and (emjbedy2<0) and (emjbsty3>0) and (emjsty1>0) and (emjbsty3=emjsty1))
emjedy1=emjbsty3.
if ((emjbedy1=-8) and (emjbsty2<0) and (emjbedy2<0) and (n507816>emjsty1) and (n507820>0)) emjedy1=n507820.
Variable labels emjedy1 "Main: Job ONE end year".
Value labels emjedy1
0 'Doing same job'
-8 'don t know'
-1 'Not applicable'.
Execute.
```

END MONTH

```
Compute Emjedm1=Emjbedm1.
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2>0) and (emjedy1=emjsty2) and (emjbstm2>0))
Emjedm1=emjbstm2.
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2>0) and (emjedy1=emjsty2) and (emjbstm2<0)) emjedm1=6.
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2>0) and (emjedy1<emjsty2) and (emjedy1>emjsty1)) emjedm1=6.
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2>0) and (emjedy1<emjsty2) and (emjedy1=emjsty1) and
(emjstm1>0)) Emjedm1=RND(mean(emjstm1,12)).
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2>0) and (emjedy1>emjsty2) and (emjedy1>emjsty1)) emjedm1=6.
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2<0) and (emjedy1>emjsty1)) emjedm1=6.
if ((emjbedm1=-8) and (emjedy1>0) and (emjsty2<0) and (emjedy1=emjsty1)) emjedm1=RND(mean(emjstm1,12)).
if ((emjbedm1=-8) and (emjedy1=n507820) and (n507818>0)) emjedm1=n507818.
if ((emjbedm1=-8) and (emjedy1=n507820) and (n507818<0) and (emjedy1>emjsty1)) emjedm1=6.
Variable labels Emjedm1 "Main: Job ONE end month".
Value labels Emjedm1
-8 'Don t know'
-1 'Not applicable'.
Execute.
```

APPENDIX TWO: Programming SPSS

Once the variables at each cross-sectional sweep are cleaned we start by merging the datasets together. The syntax for the path to these data files (e.g. ‘C:\NCDS EMPLOYMENT HISTORY\...) will need to be amended to point at the directory in which you have stored them on your own PC.

```
GET
  FILE='C:\NCDS EMPLOYMENT HISTORY\ncds4 employment history.sav'.
SORT CASES BY
  serial (A) .
SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\ncds4 employment history.sav'
 /COMPRESSED.

GET
  FILE='C:\NCDS EMPLOYMENT HISTORY\ncds5 employment history.sav'.
SORT CASES BY
  serial (A) .
SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\ncds5 employment history.sav'
 /COMPRESSED.

GET
  FILE='C:\NCDS EMPLOYMENT HISTORY\ncds6 employment history.sav'.
SORT CASES BY
  serial (A) .
SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\ncds6 employment history.sav'
 /COMPRESSED.

**now obtain ncds 4 datafile and sort and save as history dataset, then merge in ncds 5, sort and save, then merge in
ncds 6 sort and save***

GET
  FILE='C:\NCDS EMPLOYMENT HISTORY\ncds4 employment history.sav'.
SORT CASES BY
  serial (A) .

SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\Master Data\NCDS Employment'+
 ' History Dataset.sav'
 /COMPRESSED.

MATCH FILES /FILE=*
 /FILE='C:\NCDS EMPLOYMENT HISTORY\ncds5 employment'+
 ' history.sav'
 /BY serial.
EXECUTE.
SORT CASES BY
  serial (A) .
SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\NCDS Employment'+
 ' History Dataset.sav'
 /COMPRESSED.

MATCH FILES /FILE=*
 /FILE='C:\NCDS EMPLOYMENT HISTORY\ncds6 employment'+
 ' history.sav'
 /BY serial.
EXECUTE.
SORT CASES BY
  serial (A) .
SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\NCDS Employment'+
 ' History Dataset.sav'
 /COMPRESSED.
```

First need to recode the system-missing values for NCDS 6 to have the same format throughout. Also need to recode the activity status of each job in NCDS 6 because these variables include information of non-employment episodes.

```
GET
  FILE='C:\NCDS EMPLOYMENT HISTORY\NCDS Employment'+
 ' History Dataset.sav'.
```

```
recode starty10 startm10 startyr9 startmo9 startyr8 startmo8 startyr7 startmo7 startyr6 startmo6 startyr5 startmo5
startyr4 startmo4 startyr3 startmo3 startyr2 startmo2 cstartyr cstartmo activi10 activit9 activit8 activit7 activit6 activit5
activit4 activit3 activit2 activity econact (98,998, 9998=-8) (99,999,9999=-2) (sysmis=-5) (-1=-1) (else=copy).
add value labels starty10 startm10 startyr9 startmo9 startyr8 startmo8 startyr7 startmo7 startyr6 startmo6 startyr5
startmo5 startyr4 startmo4 startyr3 startmo3 startyr2 startmo2 cstartyr cstartmo activi10 activit9 activit8 activit7
activit6 activit5 activit4 activit3 activit2 activity econact
-2 'not answered'
-5 'system missing'.
execute.
missing values starty10 startm10 startyr9 startmo9 startyr8 startmo8 startyr7 startmo7 startyr6 startmo6 startyr5
startmo5 startyr4 startmo4 startyr3 startmo3 startyr2 startmo2 cstartyr cstartmo activi10 activit9 activit8 activit7
activit6 activit5 activit4 activit3 activit2 activity econact (-9 thru -1).
freq starty10 startm10 startyr9 startmo9 startyr8 startmo8 startyr7 startmo7 startyr6 startmo6 startyr5 startmo5
startyr4 startmo4 startyr3 startmo3 startyr2 startmo2 cstartyr cstartmo activi10 activit9 activit8 activit7 activit6 activit5
activit4 activit3 activit2 activity econact .


recode activi10 activit9 activit8 activit7 activit6 activit5 activit4 activit3 activit2 activity econact
(1,3=1) (2,4=2) (5 thru hi=-1) (98,998, 9998=-8) (99,999,9999=-2) (else=copy) into act10 act9 act8 act7 act6 act5
act4 act3 act2 act1 cact .
variable labels act10 "NCDS 6 Job status activity 10".
variable labels act9 "NCDS 6 Job status activity 9".
variable labels act8 "NCDS 6 Job status activity 8".
variable labels act7 "NCDS 6 Job status activity 7".
variable labels act6 "NCDS 6 Job status activity 6".
variable labels act5 "NCDS 6 Job status activity 5".
variable labels act4 "NCDS 6 Job status activity 4".
variable labels act3 "NCDS 6 Job status activity 3".
variable labels act2 "NCDS 6 Job status activity 2".
variable labels act1 "NCDS 6 Job status activity 1".
variable labels cact "NCDS 6 Job status activity at interview".
add value labels act10 act9 act8 act7 act6 act5 act4 act3 act2 act1 cact
-8 'don t know'
-5 'system missing'
-2 'not answered'
-1 'not applicable'
1 'full time'
2 'part time'.
execute.
missing values act10 act9 act8 act7 act6 act5 act4 act3 act2 act1 cact  (-9 thru -1).
```

VECTORS

```
*LOADS THE START MONTH VALUES INTO SPSS*

missing values dmjbstm1 dmjm2 emjbstm1 emjbstm2 emjbstm3 emjbstm4 emjbstm5 emjbstm6
emjbstm7 emjbstm8 emjbstm9 emjbstm10 emjbstm11 emjbstm12 startm10 startmo9 startmo8 startmo7
startmo6 startmo5 startmo4 startmo3 startmo2 startyr cstartmo activi10 activit9 activit8 activit7 activit6
activit5 activit4 activit3 activit2 activity econact ().

vector mth(25).
loop #a=1 to 25.
+ compute mth(#a)=0.
end loop.

compute mth1=dmjbstm1.
if (dmjbstm1=0) mth1=-1.

compute mth2=dmjm2.
if (dmjm2=0) mth2=-1.

compute mth3=emjbstm1.
if (emjbstm1=0) mth3=-1.

compute mth4=emjbstm2.
if (emjbstm2=0) mth4=-1.

compute mth5=emjbstm3.
if (emjbstm3=0) mth5=-1.

compute mth6=emjbstm4.
if (emjbstm4=0) mth6=-1.

compute mth7=emjbstm5.
if (emjbstm5=0) mth7=-1.

compute mth8=emjbstm6.
if (emjbstm6=0) mth8=-1.

compute mth9=emjbstm7.
if (emjbstm7=0) mth9=-1.

compute mth10=emjbstm8.
if (emjbstm8=0) mth10=-1.

compute mth11=emjbstm9.
if (emjbstm9=0) mth11=-1.

compute mth12=emjbstm10.
if (emjbstm10=0) mth12=-1.

compute mth13=emjbstm11.
if (emjbstm11=0) mth13=-1.

compute mth14=emjbstm12.
if (emjbstm12=0) mth14=-1.

compute mth15=startm10.
if (activi10>4) mth15=-1.
if (startm10=0) mth15=-1.

compute mth16=startmo9.
if (activit9>4) mth16=-1.
if (startmo9=0) mth16=-1.

compute mth17=startmo8.
if (activit8>4) mth17=-1.
if (startmo8=0) mth17=-1.

compute mth18=startmo7.
if (activit7>4) mth18=-1.
if (startmo7=0) mth18=-1.

compute mth19=startmo6.
if (activit6>4) mth19=-1.
if (startmo6=0) mth19=-1.

compute mth20=startmo5.
if (activit5>4) mth20=-1.
if (startmo5=0) mth20=-1.

compute mth21=startmo4.
if (activit4>4) mth21=-1.
if (startmo4=0) mth21=-1.

compute mth22=startmo3.
if (activit3>4) mth22=-1.
if (startmo3=0) mth22=-1.

compute mth23=startmo2.
if (activit2>4) mth23=-1.
if (startmo2=0) mth23=-1.

compute mth24=startmo.
if (activity>4) mth24=-1.
if (startmo=0) mth24=-1.

compute mth25=cstartmo.
if (econact>4) mth25=-1.

execute.

variable label mth1 "ncds 4 - start month job one".
variable label mth2 "ncds 4 - start month job two".
variable label mth3 "ncds 5 - start month job one".
variable label mth4 "ncds 5 - start month job two".
variable label mth5 "ncds 5 - start month job three".
variable label mth6 "ncds 5 - start month job four".
variable label mth7 "ncds 5 - start month job five".
variable label mth8 "ncds 5 - start month job six".
variable label mth9 "ncds 5 - start month job seven".
variable label mth10 "ncds 5 - start month job eight".
variable label mth11 "ncds 5 - start month job nine".
variable label mth12 "ncds 5 - start month job ten".
variable label mth13 "ncds 5 - start month job eleven".
variable label mth14 "ncds 5 - start month job twelve".
variable label mth15 "ncds 6 - start month job ten".
variable label mth16 "ncds 6 - start month job nine".
variable label mth17 "ncds 6 - start month job eight".
variable label mth18 "ncds 6 - start month job seven".
variable label mth19 "ncds 6 - start month job six".
variable label mth20 "ncds 6 - start month job five".
variable label mth21 "ncds 6 - start month job four".
variable label mth22 "ncds 6 - start month job three".
variable label mth23 "ncds 6 - start month job two".
variable label mth24 "ncds 6 - start month job one".
variable label mth25 "ncds 6 - start month current job".
value labels mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14 mth15
mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25
0 'doing same job'
```

```
-8 'don t know'
-1 'not applicable'.
missing values mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14
mth15 mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25
 dmjbstm1 dmjm2  emjbstm1 emjbstm2 emjbstm3 emjbstm4 emjbstm5 emjbstm6 emjbstm7 emjbstm8
emjbstm9 emjbstm10
 emjbstm11 emjbstm12 startm10 startmo9 startmo8 startmo7 startmo6 startmo5 startmo4 startmo3
startmo2 startmo cstartmo activi10 activit9 activit8 activit7 activit6 activit5 activit4 activit3 activit2 activity
econact (-9 thru -1).


recode mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14 mth15
mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25 (sysmis=-5) (else=copy).
add value labels mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14
mth15 mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25
-5 'system missing'.

missing values mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14
mth15 mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25
 dmjbstm1 dmjm2 emjbstm1 emjbstm2 emjbstm3 emjbstm4 emjbstm5 emjbstm6 emjbstm7 emjbstm8
emjbstm9 emjbstm10
 emjbstm11 emjbstm12 startm10 startmo9 startmo8 startmo7 startmo6 startmo5 startmo4 startmo3
startmo2 startmo cstartmo activi10 activit9 activit8 activit7 activit6 activit5 activit4 activit3 activit2 activity
econact (-9 thru -1).

exe.

*LOADS THE START YEAR VALUES INTO SPSS*

missing values dmjbsty1 dmj2 emjbsty1 emjbsty2 emjbsty3 emjbsty4 emjbsty5 emjbsty6 emjbsty7
emjbsty8 emjbsty9 emjbsty10 emjbsty11 emjbsty12
 starty10 startyr9 startyr8 startyr7 startyr6 startyr5 startyr4 startyr3 startyr2 startyr cstartyr ().

vector jobs(25).
loop #e=1 to 25.
+ compute jobs(#e)=0.
end loop.

compute jobs1=dmjbsty1.
compute jobs2=dmj2.
compute jobs3=emjbsty1.

compute jobs4=emjbsty2.
if (emjbsty2=0) jobs4=-1.

compute jobs5=emjbsty3.
if (emjbsty3=0) jobs5=-1.

compute jobs6=emjbsty4.

compute jobs7=emjbsty5.
if (emjbsty5=0) jobs7=-1.

compute jobs8=emjbsty6.
compute jobs9=emjbsty7.
compute jobs10=emjbsty8.
compute jobs11=emjbsty9.
compute jobs12=emjbsty10.
compute jobs13=emjbsty11.
compute jobs14=emjbsty12.

compute jobs15=starty10.
if (activi10>4) jobs15=-1.

compute jobs16=startyr9.
if (activit9>4) jobs16=-1.
```

```
compute jobs17=startyr8.
if (activit8>4) jobs17=-1.

compute jobs18=startyr7.
if (activit7>4) jobs18=-1.

compute jobs19=startyr6.
if (activit6>4) jobs19=-1.

compute jobs20=startyr5.
if (activit5>4) jobs20=-1.

compute jobs21=startyr4.
if (activit4>4) jobs21=-1.

compute jobs22=startyr3.
if (activit3>4) jobs22=-1.

compute jobs23=startyr2.
if (activit2>4) jobs23=-1.

compute jobs24=startyr.
if (activity>4) jobs24=-1.

compute jobs25=cstartyr.
if (econact>4) jobs25=-1.

execute.

variable label jobs1 "ncds 4 - start year job one".
variable label jobs2 "ncds 4 - start year job two".
variable label jobs3 "ncds 5 - start year job one".
variable label jobs4 "ncds 5 - start year job two".
variable label jobs5 "ncds 5 - start year job three".
variable label jobs6 "ncds 5 - start year job four".
variable label jobs7 "ncds 5 - start year job five".
variable label jobs8 "ncds 5 - start year job six".
variable label jobs9 "ncds 5 - start year job seven".
variable label jobs10 "ncds 5 - start year job eight".
variable label jobs11 "ncds 5 - start year job nine".
variable label jobs12 "ncds 5 - start year job ten".
variable label jobs13 "ncds 5 - start year job eleven".
variable label jobs14 "ncds 5 - start year job twelve".
variable label jobs15 "ncds 6 - start year job ten".
variable label jobs16 "ncds 6 - start year job nine".
variable label jobs17 "ncds 6 - start year job eight".
variable label jobs18 "ncds 6 - start year job seven".
variable label jobs19 "ncds 6 - start year job six".
variable label jobs20 "ncds 6 - start year job five".
variable label jobs21 "ncds 6 - start year job four".
variable label jobs22 "ncds 6 - start year job three".
variable label jobs23 "ncds 6 - start year job two".
variable label jobs24 "ncds 6 - start year job one".
variable label jobs25 "ncds 6 - start year current job".
value labels jobs1 jobs2 jobs3 jobs4 jobs5 jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12
jobs13 jobs14 jobs15 jobs16 jobs17 jobs18 jobs19 jobs20 jobs21 jobs22 jobs23 jobs24 jobs25
0 'doing same job'
-8 'don t know'
-2 'not answered'
-1 'not applicable'.
missing values jobs1 jobs2 jobs3 jobs4 jobs5 jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12
 jobs13 jobs14 jobs15 jobs16 jobs17 jobs18 jobs19 jobs20 jobs21 jobs22 jobs23 jobs24 jobs25
 dmjbsty1 dmj2 emjbsty1 emjbsty2 emjbsty3 emjbsty4 emjbsty5 emjbsty6 emjbsty7 emjbsty8 emjbsty9
emjbsty10 emjbsty11 emjbsty12
 starty10 startyr9 startyr8 startyr7 startyr6 startyr5 startyr4 startyr3 startyr2 startyr cstartyr (-9 thru -1).
```

```
recode jobs1 jobs2 jobs3 jobs4 jobs5 jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12 jobs13 jobs14
jobs15 jobs16 jobs17 jobs18 jobs19
  jobs20 jobs21 jobs22 jobs23 jobs24 jobs25 (sysmis=-5) (else=copy).
add value labels jobs1 jobs2 jobs3 jobs4 jobs5 jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12 jobs13
jobs14 jobs15 jobs16 jobs17 jobs18 jobs19
  jobs20 jobs21 jobs22 jobs23 jobs24 jobs25
-5 'system missing'.

execute.

**LOADS THE JOB STATUS - part time or full time**

missing values dmjbst1 dmjst2 emjbst1 emjbst2 emjbst3 emjbst4 emjbst5 emjbst6 emjbst7 emjbst8
emjbst9 emjbst10 emjbst11 emjbst12
 act10 act9 act8 act7 act6 act5 act4 act3 act2 act1 cact ().

vector stat(25).
loop #e=1 to 25.
+ compute stat(#e)=0.
end loop.

compute stat1=dmjbst1.
compute stat2=dmjst2.
compute stat3=emjbst1.
compute stat4=emjbst2.
compute stat5=emjbst3.
compute stat6=emjbst4.
compute stat7=emjbst5.
compute stat8=emjbst6.
compute stat9=emjbst7.
compute stat10=emjbst8.
compute stat11=emjbst9.
compute stat12=emjbst10.
compute stat13=emjbst11.
compute stat14=emjbst12.
compute stat15=act10.
compute stat16=act9.
compute stat17=act8.
compute stat18=act7.
compute stat19=act6.
compute stat20=act5.
compute stat21=act4.
compute stat22=act3.
compute stat23=act2.
compute stat24=act1.
compute stat25=cact.

execute.

variable label stat1 "ncds 4 - job status one".
variable label stat2 "ncds 4 - job status two".
variable label stat3 "ncds 5 - job status one".
variable label stat4 "ncds 5 - job status two".
variable label stat5 "ncds 5 - job status three".
variable label stat6 "ncds 5 - job status four".
variable label stat7 "ncds 5 - job status five".
variable label stat8 "ncds 5 - job status six".
variable label stat9 "ncds 5 - job status seven".
variable label stat10 "ncds 5 - job status eight".
variable label stat11 "ncds 5 - job status nine".
variable label stat12 "ncds 5 - job status ten".
variable label stat13 "ncds 5 - job status eleven".
variable label stat14 "ncds 5 - job status twelve".
variable label stat15 "ncds 6 - job status ten".
variable label stat16 "ncds 6 - job status nine".
variable label stat17 "ncds 6 - job status eight".
variable label stat18 "ncds 6 - job status seven".
variable label stat19 "ncds 6 - job status six".
variable label stat20 "ncds 6 - job status five".
variable label stat21 "ncds 6 - job status four".
variable label stat22 "ncds 6 - job status three".
variable label stat23 "ncds 6 - job status two".
variable label stat24 "ncds 6 - job status one".
variable label stat25 "ncds 6 - job status current job".
value labels stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10 stat11 stat12 stat13 stat14
 stat15 stat16 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25
-8 'don t know'
-1 'not applicable'
1 'Full time'
2 'Part time'.
missing values stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10 stat11 stat12 stat13 stat14
stat15 stat16 stat17 stat18
 stat19 stat20 stat21 stat22 stat23 stat24 stat25 dmjbst1 dmjst2 emjbst1 emjbst2 emjbst3 emjbst4
emjbst5 emjbst6 emjbst7 emjbst8
 emjbst9 emjbst10 emjbst11 emjbst12 act10 act9 act8 act7 act6 act5 act4 act3 act2 act1 cact (-9 thru -
1).

recode stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10 stat11 stat12 stat13 stat14 stat15 stat16
 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25 (sysmis=-5) (else=copy).
add value labels stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10 stat11 stat12 stat13 stat14
stat15 stat16
 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25
-5 'system missing'.

execute.

*LOADS THE END MONTH VALUES*

missing values dmjbedm1 dmjenm2 emjbedm1 emjbedm2 emjbedm3 emjbedm4 emjbedm5 emjbedm6
emjbedm7 emjbedm8 emjbedm9 emjbedm10 emjbedm11 emjbedm12
 startmo9 startmo8 startmo7 startmo6 startmo5 startmo4 startmo3 startmo2 startyr cstartmo ().

vector mend(25).
loop #e=1 to 25.
+ compute mend(#e)=0.
end loop.

compute mend1=dmjbedm1.
compute mend2=dmjenm2.
compute mend3=emjbedm1.
compute mend4=emjbedm2.
compute mend5=emjbedm3.
compute mend6=emjbedm4.
compute mend7=emjbedm5.
compute mend8=emjbedm6.
compute mend9=emjbedm7.
compute mend10=emjbedm8.
compute mend11=emjbedm9.
compute mend12=emjbedm10.
compute mend13=emjbedm11.
compute mend14=emjbedm12.
compute mend15=startmo9.
compute mend16=startmo8.
compute mend17=startmo7.
compute mend18=startmo6.
compute mend19=startmo5.
compute mend20=startmo4.
compute mend21=startmo3.
compute mend22=startmo2.
compute mend23=startmo.
compute mend24=cstartmo.
compute mend25=-1.
```

```
execute.

variable label mend1 "ncds 4 - end month job one".
variable label mend2 "ncds 4 - end month job two".
variable label mend3 "ncds 5 - end month job one".
variable label mend4 "ncds 5 - end month job two".
variable label mend5 "ncds 5 - end month job three".
variable label mend6 "ncds 5 - end month job four".
variable label mend7 "ncds 5 - end month job five".
variable label mend8 "ncds 5 - end month job six".
variable label mend9 "ncds 5 - end month job seven".
variable label mend10 "ncds 5 - end month job eight".
variable label mend11 "ncds 5 - end month job nine".
variable label mend12 "ncds 5 - end month job ten".
variable label mend13 "ncds 5 - end month job eleven".
variable label mend14 "ncds 5 - end month job twelve".
variable label mend15 "ncds 6 - end month job ten".
variable label mend16 "ncds 6 - end month job nine".
variable label mend17 "ncds 6 - end month job eight".
variable label mend18 "ncds 6 - end month job seven".
variable label mend19 "ncds 6 - end month job six".
variable label mend20 "ncds 6 - end month job five".
variable label mend21 "ncds 6 - end month job four".
variable label mend22 "ncds 6 - end month job three".
variable label mend23 "ncds 6 - end month job two".
variable label mend24 "ncds 6 - end month job one".
variable label mend25 "ncds 6 - end month - interview date".
value labels mend1 mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11
mend12
mend13 mend14 mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23
mend24 mend25
0 'doing same job'
-8 'don t know'
-5 'system missing'
-2 'not answered'
-1 'not applicable'.
missing values mend1 mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11
mend12
mend13 mend14 mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23
mend24 mend25
dmjbedm1 dmjenm2 emjbedm1 emjbedm2 emjbedm3 emjbedm4 emjbedm5 emjbedm6 emjbedm7
emjbedm8 emjbedm9 emjbedm10 emjbedm11 emjbedm12
 startm10 startmo9 startmo8 startmo7 startmo6 startmo5 startmo4 startmo3 startmo2 startyr cstartmo (-
9 thru -1).

recode mend1 mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11 mend12
 mend13 mend14 mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23
mend24 mend25 (sysmis=-5) (else=copy).
add value labels mend1 mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11
mend12
 mend13 mend14 mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23
mend24 mend25
-5 'system missing'.

execute.


**LOADS THE END YEAR VALUES**

missing values dmjbedy1 dmjeny2 emjbedy1 emjbedy2 emjbedy3 emjbedy4 emjbedy5 emjbedy6
emjbedy7 emjbedy8 emjbedy9 emjbedy10 emjbedy11 emjbedy12 startyr9 startyr8 startyr7 startyr6
startyr5 startyr4 startyr3 startyr2 startyr cstartyr ().

vector endy(25).
loop #e=1 to 25.
+ compute endy(#e)=0.
end loop.
```

```
compute endy1=dmjbedy1.
compute endy2=dmjeny2.
compute endy3=emjbedy1.
compute endy4=emjbedy2.
compute endy5=emjbedy3.
compute endy6=emjbedy4.
compute endy7=emjbedy5.
compute endy8=emjbedy6.
compute endy9=emjbedy7.
compute endy10=emjbedy8.
compute endy11=emjbedy9.
compute endy12=emjbedy10.
compute endy13=emjbedy11.
compute endy14=emjbedy12.
compute endy15=startyr9.
compute endy16=startyr8.
compute endy17=startyr7.
compute endy18=startyr6.
compute endy19=startyr5.
compute endy20=startyr4.
compute endy21=startyr3.
compute endy22=startyr2.
compute endy23=startyr.
compute endy24=cstartyr.
compute endy25=-1.

execute.

variable label endy1 "ncds 4 - end year job one".
variable label endy2 "ncds 4 - end year job two".
variable label endy3 "ncds 5 - end year job one".
variable label endy4 "ncds 5 - end year job two".
variable label endy5 "ncds 5 - end year job three".
variable label endy6 "ncds 5 - end year job four".
variable label endy7 "ncds 5 - end year job five".
variable label endy8 "ncds 5 - end year job six".
variable label endy9 "ncds 5 - end year job seven".
variable label endy10 "ncds 5 - end year job eight".
variable label endy11 "ncds 5 - end year job nine".
variable label endy12 "ncds 5 - end year job ten".
variable label endy13 "ncds 5 - end year job eleven".
variable label endy14 "ncds 5 - end year job twelve".
variable label endy15 "ncds 6 - end year job ten".
variable label endy16 "ncds 6 - end year job nine".
variable label endy17 "ncds 6 - end year job eight".
variable label endy18 "ncds 6 - end year job seven".
variable label endy19 "ncds 6 - end year job six".
variable label endy20 "ncds 6 - end year job five".
variable label endy21 "ncds 6 - end year job four".
variable label endy22 "ncds 6 - end year job three".
variable label endy23 "ncds 6 - end year job two".
variable label endy24 "ncds 6 - end year job one".
variable label endy25 "ncds 6 - end year - interview date".
value labels endy1 endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12
endy13 endy14 endy15 endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24 endy25
0 'doing same job'
-8 'don t know'
-5 'system missing'
-2 'not answered'
-1 'not applicable'.
missing values endy1 endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12
endy13 endy14 endy15 endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24 endy25
dmjbedy1 dmjeny2 emjbedy1 emjbedy2 emjbedy3 emjbedy4
 emjbedy5 emjbedy6 emjbedy7 emjbedy8 emjbedy9 emjbedy10 emjbedy11 emjbedy12 startyr9 startyr8
startyr7 startyr6 startyr5 startyr4 startyr3 startyr2
 startyr cstartyr (-9 thru -1).
```

```
recode endy1 endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12
 endy13 endy14 endy15 endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24
endy25 (sysmis=-5) (else=copy).
add value labels endy1 endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12
 endy13 endy14 endy15 endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24
endy25
-5 'system missing'.

execute.


**LOADS THE OCCUPATION CATEGORIES**

missing values fstocc secocc emsoc1 emsoc2 emsoc3 emsoc4 emsoc5 emsoc6 emsoc7 emsoc8
emsoc9 emsoc10 emsoc11 emsoc12
 soc soc2 soc3 soc4 soc5 soc6 soc7 soc8 soc9 soc10 soc11().

vector occ(25).
loop #e=1 to 25.
+ compute occ(#e)=0.
end loop.

compute occ1=fstocc.
compute occ2=secocc.
compute occ3=emsoc1.
compute occ4=emsoc2.
compute occ5=emsoc3.
compute occ6=emsoc4.
compute occ7=emsoc5.
compute occ8=emsoc6.
compute occ9=emsoc7.
compute occ10=emsoc8.
compute occ11=emsoc9.
compute occ12=emsoc10.
compute occ13=emsoc11.
compute occ14=emsoc12.

compute occ15=soc11.
if (activi10>4) occ15=-1.

compute occ16=soc10.
if (activit9>4) occ16=-1.

compute occ17=soc9.
if (activit8>4) occ17=-1.

compute occ18=soc8.
if (activit7>4) occ18=-1.

compute occ19=soc7.
if (activit6>4) occ19=-1.

compute occ20=soc6.
if (activit5>4) occ20=-1.

compute occ21=soc5.
if (activit4>4) occ21=-1.

compute occ22=soc4.
if (activit3>4) occ22=-1.

compute occ23=soc3.
if (activit2>4) occ23=-1.

compute occ24=soc2.
if (activity>4) occ24=-1.
```

```
compute occ25=soc.
if (econact>4) occ25=-1.

execute.

variable label occ1 "ncds 4 - occupation one".
variable label occ2 "ncds 4 - occupation two".
variable label occ3 "ncds 5 - occupation one".
variable label occ4 "ncds 5 - occupation two".
variable label occ5 "ncds 5 - occupation three".
variable label occ6 "ncds 5 - occupation four".
variable label occ7 "ncds 5 - occupation five".
variable label occ8 "ncds 5 - occupation six".
variable label occ9 "ncds 5 - occupation seven".
variable label occ10 "ncds 5 - occupation eight".
variable label occ11 "ncds 5 - occupation nine".
variable label occ12 "ncds 5 - occupation ten".
variable label occ13 "ncds 5 - occupation eleven".
variable label occ14 "ncds 5 - occupation twelve".
variable label occ15 "ncds 6 - occupation ten".
variable label occ16 "ncds 6 - occupation nine".
variable label occ17 "ncds 6 - occupation eight".
variable label occ18 "ncds 6 - occupation seven".
variable label occ19 "ncds 6 - occupation six".
variable label occ20 "ncds 6 - occupation five".
variable label occ21 "ncds 6 - occupation four".
variable label occ22 "ncds 6 - occupation three".
variable label occ23 "ncds 6 - occupation two".
variable label occ24 "ncds 6 - occupation one".
variable label occ25 "ncds 6 - occupation current job".
value labels occ1 occ2 occ3 occ4 occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15
occ16 occ17
 occ18 occ19 occ20 occ21 occ22 occ23 occ24 occ25
-8 'don t know'
-1 'not applicable'.
missing values occ1 occ2 occ3 occ4 occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15
occ16 occ17
 occ18 occ19 occ20 occ21 occ22 occ23 occ24 occ25 fstocc secocc emsoc1 emsoc2 emsoc3 emsoc4
emsoc5 emsoc6 emsoc7 emsoc8 emsoc9 emsoc10 emsoc11 emsoc12
 soc soc2 soc3 soc4 soc5 soc6 soc7 soc8 soc9 soc10 soc11 (-9 thru -1).

recode occ1 occ2 occ3 occ4 occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15 occ16
occ17
 occ18 occ19 occ20 occ21 occ22 occ23 occ24 occ25 (-88=-99) (sysmis=-5) (else=copy).
add value labels occ1 occ2 occ3 occ4 occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14
occ15 occ16 occ17
 occ18 occ19 occ20 occ21 occ22 occ23 occ24 occ25
-5 'system missing'.

execute.

************************************************************************************************************************
*The years collected in NCDS 4 and 5 are in a two digit format (80, 90 etc) and the years collected in
NCDS 6 are in four digit format (1992, 1993 etc) therefore need to make the start year and end year
dates into the same format*
************************************************************************************************************************

if (jobs1>1 and jobs1<100) jobs1=(jobs1+1900).
if (jobs2>1 and jobs2<100) jobs2=(jobs2+1900).
if (jobs3>1 and jobs3<100) jobs3=(jobs3+1900).
if (jobs4>1 and jobs4<100) jobs4=(jobs4+1900).
if (jobs5>1 and jobs5<100) jobs5=(jobs5+1900).
if (jobs6>1 and jobs6<100) jobs6=(jobs6+1900).
if (jobs7>1 and jobs7<100) jobs7=(jobs7+1900).
if (jobs8>1 and jobs8<100) jobs8=(jobs8+1900).
if (jobs9>1 and jobs9<100) jobs9=(jobs9+1900).
if (jobs10>1 and jobs10<100) jobs10=(jobs10+1900).
if (jobs11>1 and jobs11<100) jobs11=(jobs11+1900).
if (jobs12>1 and jobs12<100) jobs12=(jobs12+1900).
if (jobs13>1 and jobs13<100) jobs13=(jobs13+1900).
if (jobs14>1 and jobs14<100) jobs14=(jobs14+1900).
if (jobs15>1 and jobs15<100) jobs15=(jobs15+1900).
if (jobs16>1 and jobs16<100) jobs16=(jobs16+1900).
if (jobs17>1 and jobs17<100) jobs17=(jobs17+1900).
if (jobs18>1 and jobs18<100) jobs18=(jobs18+1900).
if (jobs19>1 and jobs19<100) jobs19=(jobs19+1900).
if (jobs20>1 and jobs20<100) jobs20=(jobs20+1900).
if (jobs21>1 and jobs21<100) jobs21=(jobs21+1900).
if (jobs22>1 and jobs22<100) jobs22=(jobs22+1900).
if (jobs23>1 and jobs23<100) jobs23=(jobs23+1900).
if (jobs24>1 and jobs24<100) jobs24=(jobs24+1900).
if (jobs25>1 and jobs25<100) jobs25=(jobs25+1900).



if (endy1>1 and endy1<100) endy1=(endy1+1900).
if (endy2>1 and endy2<100) endy2=(endy2+1900).
if (endy3>1 and endy3<100) endy3=(endy3+1900).
if (endy4>1 and endy4<100) endy4=(endy4+1900).
if (endy5>1 and endy5<100) endy5=(endy5+1900).
if (endy6>1 and endy6<100) endy6=(endy6+1900).
if (endy7>1 and endy7<100) endy7=(endy7+1900).
if (endy8>1 and endy8<100) endy8=(endy8+1900).
if (endy9>1 and endy9<100) endy9=(endy9+1900).
if (endy10>1 and endy10<100) endy10=(endy10+1900).
if (endy11>1 and endy11<100) endy11=(endy11+1900).
if (endy12>1 and endy12<100) endy12=(endy12+1900).
if (endy13>1 and endy13<100) endy13=(endy13+1900).
if (endy14>1 and endy14<100) endy14=(endy14+1900).
if (endy15>1 and endy15<100) endy15=(endy15+1900).
if (endy16>1 and endy16<100) endy16=(endy16+1900).
if (endy17>1 and endy17<100) endy17=(endy17+1900).
if (endy18>1 and endy18<100) endy18=(endy18+1900).
if (endy19>1 and endy19<100) endy19=(endy19+1900).
if (endy20>1 and endy20<100) endy20=(endy20+1900).
if (endy21>1 and endy21<100) endy21=(endy21+1900).
if (endy22>1 and endy22<100) endy22=(endy22+1900).
if (endy23>1 and endy23<100) endy23=(endy23+1900).
if (endy24>1 and endy24<100) endy24=(endy24+1900).
if (endy25>1 and endy25<100) endy25=(endy25+1900).

execute.

*********************************************************************************************************************
Need to recode all the system missing values now.  Cases that are system missing because only
entered one or two of the surveys are set to -33. Not applicable and refusals are set to -22 and cases
set to Don’t Know remain as -8 for recoding purposes (recoding -8 values are up to individual researcher
************************************************************************************************************************

recode mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14 mth15
mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25 jobs1 jobs2 jobs3 jobs4 jobs5
jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12 jobs13 jobs14 jobs15 jobs16 jobs17 jobs18 jobs19 jobs20
jobs21 jobs22 jobs23 jobs24 jobs25 stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10 stat11
stat12 stat13 stat14 stat15 stat16 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25 mend1
mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11 mend12 mend13 mend14
mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23 mend24 mend25 endy1
endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12 endy13 endy14 endy15
endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24 endy25 occ1 occ2 occ3 occ4
occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15 occ16 occ17 occ18 occ19 occ20
occ21 occ22 occ23 occ24 occ25 (-5=-33) (9999 =-22) (-99, -9, -1=-22) (9998, -8=-8).

add value labels mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14
mth15 mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25 jobs1 jobs2 jobs3 jobs4
```

```
jobs5 jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12 jobs13 jobs14 jobs15 jobs16 jobs17 jobs18 jobs19
jobs20 jobs21 jobs22 jobs23 jobs24 jobs25 stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10
stat11 stat12 stat13 stat14 stat15 stat16 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25
mend1 mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11 mend12 mend13
mend14 mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23 mend24
mend25 endy1 endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12 endy13
endy14 endy15 endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24 endy25 occ1
occ2 occ3 occ4 occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15 occ16 occ17 occ18
occ19 occ20 occ21 occ22 occ23 occ24 occ25
-8 'don t know - value can be recoded on dates'
-22 'na and refusals'
-33 'system missing'.
execute.

*******************************************
*SAVE FILE*
******************************************

SAVE OUTFILE='C:\NCDS EMPLOYMENT HISTORY\NCDS Employment'+
 ' History Dataset.sav'
 /COMPRESSED.
```

APPENDIX THREE: Restructuring the data file

```
***********************************************************************************************************
Need to turn the missing values off so that the vectors do not pick up the -33 and -22 values – these are
now turned to -1. DO need to pick up the -8 values still so these remain as they are.
***********************************************************************************************************

recode mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14 mth15
mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25 jobs1 jobs2 jobs3 jobs4 jobs5
jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12 jobs13 jobs14 jobs15 jobs16 jobs17 jobs18 jobs19 jobs20
jobs21 jobs22 jobs23 jobs24 jobs25 stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10 stat11
stat12 stat13 stat14 stat15 stat16 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25 mend1
mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11 mend12 mend13 mend14
mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23 mend24 mend25 endy1
endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12 endy13 endy14 endy15
  endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24 endy25 occ1 occ2 occ3 occ4
occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15 occ16 occ17 occ18 occ19 occ20
occ21 occ22 occ23 occ24 occ25 (-33, -22=-1) (else=copy).
exe.

missing values mth1 mth2 mth3 mth4 mth5 mth6 mth7 mth8 mth9 mth10 mth11 mth12 mth13 mth14
mth15 mth16 mth17 mth18 mth19 mth20 mth21 mth22 mth23 mth24 mth25 jobs1 jobs2 jobs3 jobs4
jobs5 jobs6 jobs7 jobs8 jobs9 jobs10 jobs11 jobs12 jobs13 jobs14 jobs15 jobs16 jobs17 jobs18 jobs19
jobs20 jobs21 jobs22 jobs23 jobs24 jobs25 stat1 stat2 stat3 stat4 stat5 stat6 stat7 stat8 stat9 stat10
stat11 stat12 stat13 stat14 stat15 stat16 stat17 stat18 stat19 stat20 stat21 stat22 stat23 stat24 stat25
mend1 mend2 mend3 mend4 mend5 mend6 mend7 mend8 mend9 mend10 mend11 mend12 mend13
mend14 mend15 mend16 mend17 mend18 mend19 mend20 mend21 mend22 mend23 mend24
mend25 endy1 endy2 endy3 endy4 endy5 endy6 endy7 endy8 endy9 endy10 endy11 endy12 endy13
endy14 endy15 endy16 endy17 endy18 endy19 endy20 endy21 endy22 endy23 endy24 endy25 occ1
occ2 occ3 occ4 occ5 occ6 occ7 occ8 occ9 occ10 occ11 occ12 occ13 occ14 occ15 occ16 occ17 occ18
occ19 occ20 occ21 occ22 occ23 occ24 occ25 ().

************************************************************************************************************************
These first sets of vectors use the information loaded from Appendix One and remove any missing cells
from each job spell to order all spells one after the other.
*************************************************************************************************************************

vector stomth(25).
vector stojob(25).
vector stostat(25).
vector stomend(25).
vector stoendy(25).
vector stoocc(25).
vector jobsmt=mth1 to mth25.
vector jobsyr=jobs1 to jobs25.
vector jobstat=stat1 to stat25.
vector jobmend=mend1 to mend25.
vector jobendy=endy1 to endy25.
vector jobocc=occ1 to occ25.
loop #B=1 to 25.
+ compute stomth(#B)=-1.
+ compute stojob(#B)=-1.
+ compute stostat(#B)=-1.
+ compute stomend(#B)=-1.
+ compute stoendy(#B)=-1.
+ compute stoocc(#B)=-1.
end loop.


compute #J=1.
loop #I=1 to 25.
do if (jobsyr(#I)>0|jobsyr(#I)=-8).
+ compute stojob(#J)=jobsyr(#I).
+ compute stomth(#J)=jobsmt(#I).
+ compute stostat(#J)=jobstat(#I).
+ compute stomend(#J)=jobmend(#I).
+ compute stoendy(#J)=jobendy(#I).
+ compute stoocc(#J)=jobocc(#I).
+ compute #J=#J+1.
end if.
end loop.
execute.


*******************************************************************************************************************
NOW NEED TO ORDER THE START MONTH AND YEAR VALUES CONSECUTIVELY

This vector makes a COPY of the values from the above vectors into new vectors.  This is because re-
ordering the values will change the values within the specified vectors and best to have a copy in case
of error.
*******************************************************************************************************************

vector smth(25).
vector syear(25).
vector jstat(25).
vector emth(25).
vector eyear(25).
vector jocc(25).
vector mt=stomth1 to stomth25.
vector yr=stojob1 to stojob25.
vector jst=stostat1 to stostat25.
vector emt=stomend1 to stomend25.
vector eyr=stoendy1 to stoendy25.
vector oc=stoocc1 to stoocc25.
loop #e=1 to 25.
+ compute smth(#e)=-1.
+ compute syear(#e)=-1.
+ compute jstat(#e)=-1.
+ compute emth(#e)=-1.
+ compute eyear(#e)=-1.
+ compute jocc(#e)=-1.
end loop.

compute #J=1.
loop #I=1 to 25.
do if (yr(#I)>=0|yr(#I)=-8).
+ compute syear(#J)=yr(#I).
+ compute smth(#J)=mt(#I).
+ compute jstat(#J)=jst(#I).
+ compute jocc(#J)=oc(#I).
+ compute emth(#J)=emt(#I).
+ compute eyear(#J)=eyr(#I).
+ compute #J=#J+1.
end if.
end loop.
exe.

************************************************************************************************************************
This now re-orders the dates chronologically
************************************************************************************************************************

vector year=syear1 to syear25.
vector month=smth1 to smth25.
vector status=jstat1 to jstat25.
vector mthend=emth1 to emth25.
vector yrend=eyear1 to eyear25.
vector occup=jocc1 to jocc25.
loop #i = 1 to 24 .
   loop #j = #i+1 to 25 .
      do if year(#i) >= year(#j) .
         compute #tempy=year(#j) .
         compute #tempm=month(#j).
```

```
        compute #tempst=status(#j).
        compute #tenmth=mthend(#j).
        compute #tenyr=yrend(#j).
        compute #tempocc=occup(#j).
        compute year(#j)=year(#i) .
        compute month(#j)=month(#i).
        compute status(#j)=status(#i) .
        compute mthend(#j)=mthend(#i).
        compute yrend(#j)=yrend(#i).
        compute occup(#j)=occup(#i).
        compute year(#i)=#tempy .
        compute month(#i)=#tempm.
        compute status(#i)=#tempst .
        compute mthend(#i)=#tenmth.
        compute yrend(#i)=#tenyr.
        compute occup(#i)=#tempocc.
            end if .
  end loop .
end loop .
exe .
```

****

This date re-ordering sets all system missing values to appear first and all job spell values appear last, so we need to re-order the actual structure now so that the years and months of each spell start from the first cell onward and the missing are set to the last cells (this does not mean that the -8 values will be set to last, they will appear where they already naturally appear in each spell).

****

```
vector stmth(25).
vector styr(25).
vector jobst(25).
vector edmth(25).
vector edyr(25).
vector jbocc(25).
vector mt=smth1 to smth25.
vector yr=syear1 to syear25.
vector jst=jstat1 to jstat25.
vector emt=emth1 to emth25.
vector eyr=eyear1 to eyear25.
vector oc=jocc1 to jocc25.
loop #e=1 to 25.
+ compute stmth(#e)=-1.
+ compute styr(#e)=-1.
+ compute jobst(#e)=-1.
+ compute edmth(#e)=-1.
+ compute edyr(#e)=-1.
+ compute jbocc(#e)=-1.
end loop.
compute #J=1.
loop #I=1 to 25.
do if (yr(#I)~=-1).
+ compute styr(#J)=yr(#I).
+ compute stmth(#J)=mt(#I).
+ compute jobst(#J)=jst(#I).
+ compute edmth(#J)=emt(#I).
+ compute edyr(#J)=eyr(#I).
+ compute jbocc(#J)=oc(#I).
+ compute #J=#J+1.
end if.
end loop.
exe.
```

****

A large number of cases have start years that are the same over two job spells, so we need to make sure that the start months of each job spell are ordered chronologically. Therefore if job spell one started in 8, 1978 and job spell two started in 2, 1978 – spell two will become the first job and spell one will become the second job. Therefore re-ordering cases where yr=yr but month>month.

```
**************************************************************************************************************

vector styear=styr1 to styr25.
vector stmonth=stmth1 to stmth25.
vector status=jobst1 to jobst25.
vector mthend=edmth1 to edmth25.
vector yrend=edyr1 to edyr25.
vector occup=jbocc1 to jbocc25.
loop #i = 1 to 24 .
  loop #j = #i+1 to 25 .
    do if ((styear(#i)=styear(#j)) and (stmonth(#i) > stmonth(#j))) .
      compute #tempy=styear(#j) .
            compute #tempm=stmonth(#j).
      compute #tempst=status(#j).
     compute #tenmth=mthend(#j).
      compute #tenyr=yrend(#j).
            compute #tempocc=occup(#j).
      compute styear(#j)=styear(#i) .
            compute stmonth(#j)=stmonth(#i).
      compute status(#j)=status(#i) .
            compute mthend(#j)=mthend(#i).
      compute yrend(#j)=yrend(#i).
      compute occup(#j)=occup(#i).
      compute styear(#i)=#tempy .
            compute stmonth(#i)=#tempm.
      compute status(#i)=#tempst .
            compute mthend(#i)=#tenmth.
      compute yrend(#i)=#tenyr.
      compute occup(#i)=#tempocc.
          end if .
  end loop .
end loop .
exe .

************************************************************************************************************
Having merged three surveys together there will be duplicate job spells and now need to remove any
duplicates, taking the information from the survey nearest to the time of the job spell and dropping the
information obtained in a later survey.
************************************************************************************************************

vector jstmth(25).
vector jstyr(25).
vector statjob(25).
vector endjmt(25).
vector endjyr(25).
vector occjob(25).
loop #k=1 to 25.
 compute jstmth(#k)=-1.
 compute jstyr(#k)=-1.
 compute statjob(#k)=-1.
 compute endjmt(#k)=-1.
 compute endjyr(#k)=-1.
 compute occjob(#k)=-1.
end loop.
 compute jstmth1=stmth1.
 compute jstyr1=styr1.
 compute statjob1=jobst1.
 compute endjmt1=edmth1.
 compute endjyr1=edyr1.
 compute occjob1=jbocc1.
vector year=styr1 to styr25.
vector month=stmth1 to stmth25.
vector status=jobst1 to jobst25.
vector mthend=edmth1 to edmth25.
vector yrend=edyr1 to edyr25.
vector occup=jbocc1 to jbocc25.
compute #j=2.
```

```
loop #i=1 to 24.
  do if year(#i) < year(#i+1).
      compute jstyr(#j)=year(#i+1).
      compute jstmth(#j)=month(#i+1).
      compute statjob(#j)=status(#i+1).
      compute endjmt(#j)=mthend(#i+1).
      compute endjyr(#j)=yrend(#i+1).
      compute occjob(#j)=occup(#i+1).
      compute #j=#j+1.
  end if.
  do if ((year(#i)=year(#i+1)) and (month(#i) < month(#i+1))).
      compute jstyr(#j)=year(#i+1).
      compute jstmth(#j)=month(#i+1).
      compute statjob(#j)=status(#i+1).
      compute endjmt(#j)=mthend(#i+1).
      compute endjyr(#j)=yrend(#i+1).
      compute occjob(#j)=occup(#i+1).
      compute #j=#j+1.
  end if.
end loop.
exe.


********************************************************************************************************************
The end values on month and year were allowed to have a 0 or 1 to identify where a respondent said
that the job had continued.  However if they then gave details of a new job in a following spell we now
need to amend the 0 and 1 to take the value of the corresponding start values.
*********************************************************************************************************************

recode endjmt1 endjmt2 endjmt3 endjmt4 endjmt5 endjmt6 endjmt7 endjmt8 endjmt9 endjmt10
endjmt11 endjmt12 endjmt13 endjmt14 endjmt15 endjmt16 endjmt17 endjmt18 endjmt19 endjmt20
endjmt21 endjmt22 endjmt23 endjmt24 endjmt25 endjyr1 endjyr2 endjyr3 endjyr4 endjyr5 endjyr6
endjyr7 endjyr8 endjyr9 endjyr10 endjyr11 endjyr12 endjyr13 endjyr14 endjyr15 endjyr16
endjyr17 endjyr18 endjyr19 endjyr20 endjyr21 endjyr22 endjyr23 endjyr24 endjyr25 (sysmis=-1)
(else=copy).

missing values endjyr1 endjyr2 endjyr3 endjyr4 endjyr5 endjyr6 endjyr7 endjyr8 endjyr9 endjyr10
endjyr11 endjyr12 endjyr13 endjyr14 endjyr15 endjyr16 endjyr17 endjyr18 endjyr19 endjyr20 endjyr21
endjyr22 endjyr23 endjyr24 endjyr25 ().
exe.

vector starty=jstyr1 to jstyr25.
vector startm=jstmth1 to jstmth25.
vector enm=endjmt1 to endjmt25.
vector eny=endjyr1 to endjyr25.
loop #i=1 to 25.
    do if ((eny(#i)=0|eny(#i)=1) and (starty(#i+1)>1)).
          compute eny(#i)=starty(#i+1).
          compute enm(#i)=startm(#i+1).
    end if.
    do if ((eny(#i)=-1) and (enm(#i)=-1) and (starty(#i)>0) and (starty(#i+1)>0)).
          compute eny(#i)=starty(#i+1).
          compute enm(#i)=startm(#i+1).
    end if.
end loop.
exe.

**need to reorder the end dates also**

vector mthend=endjmt1 to endjmt25.
vector yrend=endjyr1 to endjyr25.
loop #i = 1 to 24 .
   loop #j = #i+1 to 25 .
      do if ( (yrend(#i)=yrend(#j)) and (mthend(#i) > mthend(#j)) ) .
              compute #tempy=yrend(#j) .
                compute #tempm=mthend(#j).
              compute yrend(#j)=yrend(#i) .
                compute mthend(#j)=mthend(#i).
              compute yrend(#i)=#tempy .
                compute mthend(#i)=#tempm.
              compute #j=#j+1.
      end if .
      do if  (yrend(#i) >yrend(#j)) .
              compute #tempy=yrend(#j) .
                compute #tempm=mthend(#j).
              compute yrend(#j)=yrend(#i) .
                compute mthend(#j)=mthend(#i).
              compute yrend(#i)=#tempy .
                compute mthend(#i)=#tempm.
              compute #j=#j+1.
      end if .
   end loop .
end loop .
exe .

**now reorder these dates so that the years and months start as from first cell onward and the missing **
**are set to the last cells**

vector mendj(25).
vector yendj(25).
vector mthend=endjmt1 to endjmt25.
vector yrend=endjyr1 to endjyr25.
compute #J=1.
loop #I=1 to 25.
do if (yrend(#I)>0).
+ compute mendj(#J)=mthend(#I).
+ compute yendj(#J)=yrend(#I).
+ compute #J=#J+1.
end if.
do if (yendj(#I)>0 and yendj(#j)=-1 and yrend(#i)=-6).
+ compute mthend(#J)=mthend(#I).
+ compute yrend(#J)=yrend(#I).
+ compute #J=#J+1.
end if.
end loop.
exe.

recode mendj1 mendj2 mendj3 mendj4 mendj5 mendj6 mendj7 mendj8 mendj9 mendj10 mendj11
mendj12 mendj13 mendj14 mendj15 mendj16 mendj17 mendj18 mendj19 mendj20 mendj21 mendj22
mendj23 mendj24 mendj25 yendj1 yendj2 yendj3 yendj4 yendj5 yendj6 yendj7 yendj8 yendj9 yendj10
yendj11 yendj12 yendj13 yendj14 yendj15 yendj16 yendj17 yendj18 yendj19 yendj20 yendj21 yendj22
yendj23 yendj24 yendj25 (sysmis=-1) (else=copy).
exe.

vector starty=jstyr1 to jstyr25.
vector startm=jstmth1 to jstmth25.
vector enm=mendj1 to mendj25.
vector eny=yendj1 to yendj25.
loop #i=1 to 24.
 loop #j=#i+1 to 25.
    do if ((eny(#i)=0|eny(#i)=1) and (eny(#i+1)>1)).
              compute #tempy=eny(#i+1) .
                compute #tempm=enm(#i+1).
              compute eny(#i+1)=eny(#i) .
                compute enm(#i+1)=enm(#i).
              compute eny(#i)=#tempy .
                compute enm(#i)=#tempm.
           compute #j=#j+1.
    end if.
 end loop.
end loop.
exe.

vector starty=jstyr1 to jstyr25.
vector startm=jstmth1 to jstmth25.
vector enm=mendj1 to mendj25.
vector eny=yendj1 to yendj25.
loop #i=1 to 24.
 loop #j=#i+1 to 25.
    do if ((eny(#i)=starty(#j)) and (enm(#i)<startm(#j))).
      compute enm(#i)=startm(#j).
    end if.
 end loop.
end loop.
exe.

*********************************************************************************************************************
For cases that remained as 0 and 1 because the respondent had no new jobs, I amend these values
now to -6 to represent the fact that they are censored at the interview.
*********************************************************************************************************************

vector starty=jstyr1 to jstyr25.
vector startm=jstmth1 to jstmth25.
vector enm=mendj1 to mendj25.
vector eny=yendj1 to yendj25.
loop #i=1 to 24.
 loop #j=#i+1 to 25.
    do if ((eny(#i)=0|eny(#i)=1) and (starty(#j)=-1)).
            compute eny(#i)=-6.
            compute enm(#i)=-6.
    end if.
    do if ((eny(#i)>1) and (enm(#i)=0)).
            compute enm(#i)=-8.
    end if.
    do if ((enm(#i)=0) and (eny(#i)=-1)).
            compute enm(#i)=-1.
    end if.
    do if ((eny(#i)=-1) and (enm(#i)=-1) and (starty(#i)>0) and (starty(#j)=-1)).
            compute eny(#i)=-6.
            compute enm(#i)=-6.
    end if.
 end loop.
end loop.
exe.

*****************************************************************************************************************
Drops any duplicate end (year and month) job spells and checks for any overlapping spells where the
end year of a job spell is greater than the start year of the next job spell.
******************************************************************************************************************

vector starty=jstyr1 to jstyr25.
vector startm=jstmth1 to jstmth25.
vector enm=mendj1 to mendj25.
vector eny=yendj1 to yendj25.
loop #i=1 to 24.
 loop #j=#i+1 to 25.
 do if ( (eny(#i)>0 and starty(#j)>0 and starty(#j+1)>0) and (eny(#i)>starty(#j)) ).
   compute eny(#i)=starty(#j).
   compute enm(#i)=startm(#j).
 end if.
 do if ((eny(#i)>0 and starty(#j)>0 and starty(#j+1)<0) and (eny(#i)>starty(#j))).
   compute #tempy=eny(#i) .
   compute #tempm=enm(#i).
   compute eny(#i)=starty(#i+1).
   compute enm(#i)=startm(#i+1).
   compute eny(#j)=#tempy .
   compute enm(#j)=#tempm.
 end if.
 do if (eny(#i)>0 and eny(#j)<0 and (starty(#j)>0)).
   compute eny(#j)=starty(#j+1).
   compute enm(#j)=startm(#j+1).
 end if.
 do if ((eny(#i)>0 and eny(#j)>0 and eny(#j+1)<0) and (eny(#i)=eny(#j))).
   compute #tempy=eny(#i) .
   compute #tempm=enm(#i).
   compute eny(#i)=eny(#i+1).
   compute enm(#i)=enm(#i+1).
   compute eny(#j)=#tempy .
   compute enm(#j)=#tempm.
 end if.
 do if ((eny(#i)=-1) and (starty(#j)>0) and (starty(#j+1)<0)).
   compute eny(#i)=starty(#j).
   compute enm(#i)=startm(#j).
 end if.
 do if ((eny(#i)=-1 and enm(#i)=-1) and (starty(#i)>0) and (starty(#j)=-1)).
     compute eny(#i)=-6.
     compute enm(#i)=-6.
 end if.
 end loop.
end loop.
exe.



*******************************************************************************************************************
This variable identifies cases that were censored at interview by picking up all the -6 values in the
employment history.
*******************************************************************************************************************

compute censored=0.
vector enm=mendj1 to mendj25.
vector eny=yendj1 to yendj25.
loop #i=1 to 25.
 do if (eny(#i)=-6).
    compute censored=1.
 end if.
end loop.
exe.
freq censored.

********************************************************************************************************************
Label values and names
********************************************************************************************************************

variable label jstmth1 "Start month job one".
variable label jstmth2 "Start month job two".
variable label jstmth3 "Start month job three".
variable label jstmth4 "Start month job four".
variable label jstmth5 "Start month job five".
variable label jstmth6 "Start month job six".
variable label jstmth7 "Start month job seven".
variable label jstmth8 "Start month job eight".
variable label jstmth9 "Start month job nine".
variable label jstmth10 "Start month job ten".
variable label jstmth11 "Start month job eleven".
variable label jstmth12 "Start month job twelve".
variable label jstmth13 "Start month job thirteen".
variable label jstmth14 "Start month job fourteen".
variable label jstmth15 "Start month job fifteen".
variable label jstmth16 "Start month job sixteen".
variable label jstmth17 "Start month job seventeen".
variable label jstmth18 "Start month job eighteen".
variable label jstmth19 "Start month job nineteen".
variable label jstmth20 "Start month job twenty".
variable label jstmth21 "Start month job twenty-one".
variable label jstmth22 "Start month job twenty-two".
variable label jstmth23 "Start month job twenty-three".
variable label jstmth24 "Start month job twenty-four".
variable label jstmth25 "Start month job twenty-five".
add value lables jstmth1 jstmth2 jstmth3 jstmth4 jstmth5 jstmth6 jstmth7 jstmth8 jstmth9 jstmth10
jstmth11 jstmth12 jstmth13 jstmth14 jstmth15
  jstmth16 jstmth17 jstmth18 jstmth19 jstmth20 jstmth21 jstmth22 jstmth23 jstmth24 jstmth25
-8 'don t know - can be recoded'
-1 'not applicable'.
missing values jstmth1 jstmth2 jstmth3 jstmth4 jstmth5 jstmth6 jstmth7 jstmth8 jstmth9 jstmth10
jstmth11 jstmth12 jstmth13 jstmth14 jstmth15
  jstmth16 jstmth17 jstmth18 jstmth19 jstmth20 jstmth21 jstmth22 jstmth23 jstmth24 jstmth25 (-9 thru -1).

variable label jstyr1 "Start year job one".
variable label jstyr2 "Start year job two".
variable label jstyr3 "Start year job three".
variable label jstyr4 "Start year job four".
variable label jstyr5 "Start year job five".
variable label jstyr6 "Start year job six".
variable label jstyr7 "Start year job seven".
variable label jstyr8 "Start year job eight".
variable label jstyr9 "Start year job nine".
variable label jstyr10 "Start year job ten".
variable label jstyr11 "Start year job eleven".
variable label jstyr12 "Start year job twelve".
variable label jstyr13 "Start year job thirteen".
variable label jstyr14 "Start year job fourteen".
variable label jstyr15 "Start year job fifteen".
variable label jstyr16 "Start year job sixteen".
variable label jstyr17 "Start year job seventeen".
variable label jstyr18 "Start year job eighteen".
variable label jstyr19 "Start year job nineteen".
variable label jstyr20 "Start year job twenty".
variable label jstyr21 "Start year job twenty-one".
variable label jstyr22 "Start year job twenty-two".
variable label jstyr23 "Start year job twenty-three".
variable label jstyr24 "Start year job twenty-four".
variable label jstyr25 "Start year job twenty-five".
add value lables jstyr1 jstyr2 jstyr3 jstyr4 jstyr5 jstyr6 jstyr7 jstyr8 jstyr9 jstyr10 jstyr11 jstyr12 jstyr13
jstyr14 jstyr15 jstyr16
  jstyr17 jstyr18 jstyr19 jstyr20 jstyr21 jstyr22 jstyr23 jstyr24 jstyr25
-8 'don t know - can be recoded'
-1 'not applicable'.
missing values jstyr1 jstyr2 jstyr3 jstyr4 jstyr5 jstyr6 jstyr7 jstyr8 jstyr9 jstyr10 jstyr11 jstyr12 jstyr13
jstyr14 jstyr15 jstyr16
  jstyr17 jstyr18 jstyr19 jstyr20 jstyr21 jstyr22 jstyr23 jstyr24 jstyr25 (-9 thru -1).

variable label statjob1 "Status job one".
variable label statjob2 "Status job two".
variable label statjob3 "Status job three".
variable label statjob4 "Status job four".
variable label statjob5 "Status job five".
variable label statjob6 "Status job six".
variable label statjob7 "Status job seven".
variable label statjob8 "Status job eight".
variable label statjob9 "Status job nine".
variable label statjob10 "Status job ten".
variable label statjob11 "Status job eleven".
variable label statjob12 "Status job twelve".
variable label statjob13 "Status job thirteen".
variable label statjob14 "Status job fourteen".
variable label statjob15 "Status job fifteen".
variable label statjob16 "Status job sixteen".
variable label statjob17 "Status job seventeen".
variable label statjob18 "Status job eighteen".
variable label statjob19 "Status job nineteen".
variable label statjob20 "Status job twenty".
variable label statjob21 "Status job twenty-one".
variable label statjob22 "Status job twenty-two".
variable label statjob23 "Status job twenty-three".
variable label statjob24 "Status job twenty-four".
variable label statjob25 "Status job twenty-five".
add value labels statjob1 statjob2 statjob3 statjob4 statjob5 statjob6 statjob7 statjob8 statjob9 statjob10
statjob11 statjob12 statjob13
  statjob14 statjob15 statjob16 statjob17 statjob18 statjob19 statjob20 statjob21 statjob22 statjob23
statjob24 statjob25
-8 'don t know - can be recoded'
-1 'not applicable'
1 'Full time'
2 'Part time'.
missing values statjob1 statjob2 statjob3 statjob4 statjob5 statjob6 statjob7 statjob8 statjob9 statjob10
statjob11 statjob12 statjob13
  statjob14 statjob15 statjob16 statjob17 statjob18 statjob19 statjob20 statjob21 statjob22 statjob23
statjob24 statjob25 (-9 thru -1).

variable label mendj1 "End month job one".
variable label mendj2 "End month job two".
variable label mendj3 "End month job three".
variable label mendj4 "End month job four".
variable label mendj5 "End month job five".
variable label mendj6 "End month job six".
variable label mendj7 "End month job seven".
variable label mendj8 "End month job eight".
variable label mendj9 "End month job nine".
variable label mendj10 "End month job ten".
variable label mendj11 "End month job eleven".
variable label mendj12 "End month job twelve".
variable label mendj13 "End month job thirteen".
variable label mendj14 "End month job fourteen".
variable label mendj15 "End month job fifteen".
variable label mendj16 "End month job sixteen".
variable label mendj17 "End month job seventeen".
variable label mendj18 "End month job eighteen".
variable label mendj19 "End month job nineteen".
variable label mendj20 "End month job twenty".
variable label mendj21 "End month job twenty-one".
variable label mendj22 "End month job twenty-two".
variable label mendj23 "End month job twenty-three".
variable label mendj24 "End month job twenty-four".
variable label mendj25 "End month job twenty-five".
add value lables mendj1 mendj2 mendj3 mendj4 mendj5 mendj6 mendj7 mendj8 mendj9 mendj10
mendj11 mendj12 mendj13 mendj14 mendj15 mendj16
  mendj17 mendj18 mendj19 mendj20 mendj21 mendj22 mendj23 mendj24 mendj25
-8 'don t know - can be recoded'
-6 'case censored at interview'
-1 'not applicable'.
missing values mendj1 mendj2 mendj3 mendj4 mendj5 mendj6 mendj7 mendj8 mendj9 mendj10
mendj11 mendj12 mendj13 mendj14 mendj15 mendj16
  mendj17 mendj18 mendj19 mendj20 mendj21 mendj22 mendj23 mendj24 mendj25 (-9 thru -1).

variable label yendj1 "End year job one".
variable label yendj2 "End year job two".
variable label yendj3 "End year job three".
variable label yendj4 "End year job four".
variable label yendj5 "End year job five".
variable label yendj6 "End year job six".
variable label yendj7 "End year job seven".
variable label yendj8 "End year job eight".
variable label yendj9 "End year job nine".
variable label yendj10 "End year job ten".
variable label yendj11 "End year job eleven".
variable label yendj12 "End year job twelve".
variable label yendj13 "End year job thirteen".
variable label yendj14 "End year job fourteen".
variable label yendj15 "End year job fifteen".
variable label yendj16 "End year job sixteen".
variable label yendj17 "End year job seventeen".
variable label yendj18 "End year job eighteen".
variable label yendj19 "End year job nineteen".
variable label yendj20 "End year job twenty".
variable label yendj21 "End year job twenty-one".
variable label yendj22 "End year job twenty-two".
variable label yendj23 "End year job twenty-three".
variable label yendj24 "End year job twenty-four".
variable label yendj25 "End year job twenty-five".
add value lables yendj1 yendj2 yendj3 yendj4 yendj5 yendj6 yendj7 yendj8 yendj9 yendj10 yendj11
yendj12 yendj13 yendj14 yendj15 yendj16
  yendj17 yendj18 yendj19 yendj20 yendj21 yendj22 yendj23 yendj24 yendj25
-8 'don t know - can be recoded'
-6 'case censored at interview'
-1 'not applicable'.
missing values yendj1 yendj2 yendj3 yendj4 yendj5 yendj6 yendj7 yendj8 yendj9 yendj10 yendj11
yendj12 yendj13 yendj14 yendj15 yendj16
  yendj17 yendj18 yendj19 yendj20 yendj21 yendj22 yendj23 yendj24 yendj25 (-9 thru -1).

variable label occjob1 "Occupation category job one".
variable label occjob2 "Occupation category job two".
variable label occjob3 "Occupation category job three".
variable label occjob4 "Occupation category job four".
variable label occjob5 "Occupation category job five".
variable label occjob6 "Occupation category job six".
variable label occjob7 "Occupation category job seven".
variable label occjob8 "Occupation category job eight".
variable label occjob9 "Occupation category job nine".
variable label occjob10 "Occupation category job ten".
variable label occjob11 "Occupation category job eleven".
variable label occjob12 "Occupation category job twelve".
variable label occjob13 "Occupation category job thirteen".
variable label occjob14 "Occupation category job fourteen".
variable label occjob15 "Occupation category job fifteen".
variable label occjob16 "Occupation category job sixteen".
variable label occjob17 "Occupation category job seventeen".
variable label occjob18 "Occupation category job eighteen".
variable label occjob19 "Occupation category job nineteen".
variable label occjob20 "Occupation category job twenty".
variable label occjob21 "Occupation category job twenty-one".
variable label occjob22 "Occupation category job twenty-two".
variable label occjob23 "Occupation category job twenty-three".
variable label occjob24 "Occupation category job twenty-four".
variable label occjob25 "Occupation category job twenty-five".
 add value labels occjob1 occjob2 occjob3 occjob4 occjob5 occjob6 occjob7 occjob8 occjob9 occjob10
occjob11 occjob12 occjob13 occjob14 occjob15
  occjob16 occjob17 occjob18 occjob19 occjob20 occjob21 occjob22 occjob23 occjob24 occjob25
-8 'don t know - can be recoded'
-1 'not applicable'.
missing values occjob1 occjob2 occjob3 occjob4 occjob5 occjob6 occjob7 occjob8 occjob9 occjob10
occjob11 occjob12 occjob13 occjob14 occjob15
  occjob16 occjob17 occjob18 occjob19 occjob20 occjob21 occjob22 occjob23 occjob24 occjob25 (-9
thru -1).
execute.

************************************************************************************************************************
The following syntax saves the variables needed for the employment history dataset
************************************************************************************************************************

save outfile='C:\NCDS EMPLOYMENT HISTORY\NCDS EMPLOYMENT DATA.sav'
 /keep= serial censored jstmth1 jstmth2 jstmth3 jstmth4 jstmth5 jstmth6 jstmth7 jstmth8 jstmth9 jstmth10
jstmth11 jstmth12 jstmth13 jstmth14 jstmth15 jstmth16 jstmth17 jstmth18 jstmth19 jstmth20 jstmth21
jstmth22 jstmth23 jstmth24 jstmth25 jstyr1 jstyr2 jstyr3 jstyr4 jstyr5 jstyr6 jstyr7 jstyr8 jstyr9 jstyr10
jstyr11 jstyr12 jstyr13 jstyr14 jstyr15 jstyr16 jstyr17 jstyr18 jstyr19 jstyr20 jstyr21 jstyr22 jstyr23 jstyr24
jstyr25 statjob1 statjob2 statjob3 statjob4 statjob5 statjob6 statjob7 statjob8 statjob9 statjob10 statjob11
statjob12 statjob13 statjob14 statjob15 statjob16 statjob17 statjob18 statjob19 statjob20 statjob21
statjob22 statjob23 statjob24 statjob25 mendj1 mendj2 mendj3 mendj4 mendj5 mendj6 mendj7 mendj8
mendj9 mendj10 mendj11 mendj12 mendj13 mendj14 mendj15 mendj16 mendj17 mendj18 mendj19
```

mendj20 mendj21 mendj22 mendj23 mendj24 mendj25 yendj1 yendj2 yendj3 yendj4 yendj5 yendj6 yendj7 yendj8 yendj9 yendj10 yendj11 yendj12 yendj13 yendj14 yendj15 yendj16 yendj17 yendj18 yendj19 yendj20 yendj21 yendj22 yendj23 yendj24 yendj25 occjob1 occjob2 occjob3 occjob4 occjob5 occjob6 occjob7 occjob8 occjob9 occjob10 occjob11 occjob12 occjob13 occjob14 occjob15 occjob16 occjob17 occjob18 occjob19 occjob20 occjob21 occjob22 occjob23 occjob24 occjob25 .

Centre for Longitudinal Studies
Bedford Group for Lifecourse and
Statistical Studies
Institute of Education
20 Bedford Way
London WC1H 0AL
Tel: 020 7612 6900
Fax: 020 7612 6880
Email cls@cls.ioe.ac.uk
Web http://www.cls.ioe.ac.uk