

DATA SET PARMIG.POR

Data Description:

This dataset provides derived variables on moves of address and their broad location based upon supplementary coding of addresses whose postal town was recorded but not coded previously. Data is available for NCDS members who provided a mother/child questionnaire in 1991 and up to 16 addresses were coded for each member. The coding of towns is not released at this point to protect confidentiality. Potential users of the geographical detail may apply to NCDS User Support.

Variables on the dataset:

SERIAL NCDS MEMBER SERIAL NUMBER

R01-R16 Standard Region 01-16
1.00 Northern
2.00 Yorks & Humber
3.00 North West
4.00 East Midlands
5.00 West Midlands
6.00 East Anglia
7.00 South East
8.00 South West
9.00 Wales
10.00 Scotland
99.00 Outside UK

Y01-Y16 Address #1-16: Date moved out-YEAR
YY Year of move
0 Still at address

MIG1-MIG15 Migration Indicator 1-16
1.00 same region
2.00 inter region
6.00 in migrant (to UK)
7.00 out migrant (from UK)
8.00 move outside uk
9.00 missing data

RMOVE Number of inter regional moves

LMOVE	Last move 1.00 same region 2.00 inter region 6.00 in migrant (to UK) 7.00 out migrant (from UK) 8.00 move outside uk 9.00 missing data
MOVES	Number of moves
XMOVE	Number of moves in/out of UK
YLMOVE	Year of last move
YRMOVE	Year last inter region move
YINMOVE	Year last in migration (to UK)
YOUTMOVE	Year last out migration (from UK)
YXMOVE	Year last move outside UK

SPSS code used to generate migration indicators.

Vector Y year(s) of move

```
vector y=y01 to y16.  
loop #i=1 to 16.  
if (y(#i)=0)last=#i.  
end loop.
```

Special case more than 16 moves

```
IF (y16 >= 82 and y16 <= 91) last = 16 .  
EXECUTE .
```

Vector R Standard region of residence or residing outside UK

```
vector r=r01 to r16.
```

Vector Mig will store up to 15 moves.

```
vector mig=mig1 to mig15.
```

Loop & compare regions at each move. Set mig accordingly

```
do if (last ge 2 and not(sysmis(last))).  
+ loop #j=2 to last.  
+ do if (sysmis(r(#j)) or sysmis(r(#j-1))).  
+   compute mig(#j-1)=9.  
+ else if (r(#j) eq 98 or r(#j-1) eq 98).  
+   compute mig(#j-1)=9.  
+ else if (r(#j) eq 99 and r(#j-1) eq 99).  
+   compute mig(#j-1)=8.  
+ else if (r(#j) eq 99).  
+   compute mig(#j-1)=7.  
+ else if (r(#j-1) eq 99).  
+   compute mig(#j-1)=6.  
+ else if (r(#j) ne r(#j-1)).  
+   compute mig(#j-1)=2.  
+ else if (r(#j) eq r(#j-1)).  
+   compute mig(#j-1)=1.  
+ end if.  
+ end loop.  
end if.
```

Compute summary variables (#moves etc)

```
compute rmove=0.  
compute xmove=0.  
compute x=last-1.  
do if (last ge 2 and not(sysmis(last))).  
loop #j=1 to x.  
if (mig(#j) eq 2)rmove=rmove+1.
```

```
if (mig(#j) eq 2) yrmove=y(#j).
if (mig(#j) eq 6) yinmove=y(#j).
if (mig(#j) eq 7) youtmove=y(#j).
if (mig(#j) eq 8) yxmove=y(#j).
if (mig(#j) ge 6 and mig(#j) le 8) xmove=xmove+1.
end loop.
```

```
compute lmove=mig(x).
compute ylmove=y(x).
end if.
compute moves=x.
execute.
```