

# ELSA Wave 2 - Derived Variables, Version 1

## *User Guide*

Authors: Susan Nunn, Kate Cox, James Trinder

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
<b>2</b>	<b>Household Demographics .....</b>	<b>3</b>
	Number of Natural Children Outside Household.....	3
	Number of Adopted Children Outside the Household.....	4
	Number of Foster Children Outside the Household .....	5
	Number of Stepchildren Outside the Household.....	6
	Total Number of Children Outside the Household .....	7
	Number of Natural Children Inside the Household.....	7
	Number of Adopted Children Inside the Household.....	7
	Number of Foster Children Inside the Household .....	7
	Number of Stepchildren Inside the Household.....	8
	Total Number of Children Inside the Household .....	8
	Total Number of Children (In and Out of Household) .....	9
	Total Number of Natural Children (In and Out of Household) .....	9
	Total Number of Adopted Children (In and Out of Household) .....	9
	Total Number of Foster Children (In and Out of Household) .....	9
	Total Number of Stepchildren (In and Out of Household) .....	9
	Whether Natural Mother is Still Alive.....	10
	Natural Mother's Age (or Estimated Age) at Death.....	11
	Whether Natural Father is Still Alive .....	11
	Natural Father's Age (or Estimated Age) at Death.....	12
<b>3</b>	<b>Health .....</b>	<b>13</b>
	Whether Respondent was in Wave 1, Wave 2 or both .....	13
	Ever Reported High Blood Pressure (Diagnosed) .....	14
	Ever Reported Angina (Diagnosed) .....	15
	Ever Reported Myocardial Infarction (Diagnosed) .....	16
	Ever Reported Congestive Heart Failure (Diagnosed).....	17
	Ever Reported Heart Murmur (Diagnosed) .....	18
	Ever Reported Arrhythmia (Diagnosed) .....	19
	Ever Reported Diabetes or High Blood Sugar (Diagnosed).....	20
	Ever Reported Diabetes (Diagnosed) .....	21
	Ever Reported a Stroke (Diagnosed) .....	23
	Ever Reported any of 7 CVD-related Diseases (Excluding High Blood Pressure) .....	24
	Ever Reported any of 7 CVD-related Diseases (Excluding High Blood Sugar and High Blood Pressure).....	25
	Ever Reported Hedibonic Lung Disease (Diagnosed) .....	26
	Ever Reported Asthma (Diagnosed) .....	27
	Ever Reported Arthritis (Diagnosed) .....	28
	Ever Reported Osteoporosis (Diagnosed) .....	29
	Ever Reported Cancer (Diagnosed).....	30
	Ever Reported Parkinson's Disease (Diagnosed).....	31
	Ever Reported Psychiatric Disorder (Diagnosed) .....	32
	Ever Reported Alzheimer's Disease (Diagnosed).....	34
	Ever Reported Dementia or Memory Impairment (Diagnosed).....	35
	Ever Reported Glaucoma (Diagnosed).....	36
	Ever Reported Diabetic Eye Disease (Diagnosed) .....	37
	Ever Reported Macular Degeneration (Diagnosed) .....	38
	Ever Reported Cataracts (Diagnosed) .....	39
	Physical Activity Summary .....	40
	Maximum Alcohol Units in a Day Last Week .....	41
	Maximum Alcohol Units in a Day Last Week .....	43
	Edinburgh Claudication .....	45
	Breathlessness, MRC Respiratory Questionnaire.....	46
	Phlegm, MRC Respiratory Questionnaire .....	48
<b>4</b>	<b>Cognitive Function .....</b>	<b>49</b>
	Response To Prospective Memory Test.....	49
	Prospective Memory Score .....	50
	Literacy Score .....	50
	Delayed Word Recall as % of Immediate Word Recall.....	51
	% Accuracy of Letter Search.....	52

Memory Function Index.....	52
Number of Letters Missed Recoded for Executive Function Index .....	53
Number of Items Completed Recoded for Executive Function Index .....	53
Fluency Recoded for Executive Function Index.....	54
Executive Function Index (Cognitive).....	55
Total Cognitive Index (Memory + Executive) .....	55
Total Cognitive Index (Memory + Executive) .....	56
<b>5 Physical Function .....</b>	<b>56</b>
Mean Gait Speed (m/s) .....	56
Walks Used to Derive Mean Gait Speed.....	58
Walks Used to Derive Mean Gait Speed.....	59
Mean Gait Speed (Grouped).....	59
Fastest Gait Speed (m/s) .....	60
<b>6 CASP 19 .....</b>	<b>62</b>
CASP 19 .....	62
Control (CASP) .....	62
Autonomy (CASP).....	63
Pleasure (CASP).....	63
Self realisation (CASP) .....	64
Number of Organisations .....	64
Number of Organisations (Grouped).....	65
<b>7 Socio-Economic Features.....</b>	<b>65</b>
Whether New NS-SEC Code Was Assigned at ELSA Wave 2.....	65
Whether New SIC Code Was Assigned at ELSA Wave 2 .....	66
Whether New SOC Code Was Assigned at ELSA Wave 2.....	66
FINAL NS-SEC - Long Version (Including Unclassifiable and Non-Workers).....	68
Reason Respondent Has No NS-SEC Code .....	70
FINAL NS-SEC - Long Version.....	71
NS-SEC 8 Variable Classification (Individual).....	73
NS-SEC 5 Variable Classification (Individual).....	74
NS-SEC 3 Variable Classification (Individual).....	74
Combined ELSA Wave 2/Wave 1/HSE: SOC2000 (without dots) .....	75
Reason Why Respondent Has No SOC2000 .....	76
Combined ELSA Wave 2/Wave 1/HSE: 2-Digit SIC92 Code.....	77
Reason Why Respondent Has No SIC92 .....	78
Highest Other Educational Qualification at ELSA Wave 2.....	79
Highest Educational Qualification Obtained by ELSA W2 .....	80
<b>8 Contact details .....</b>	<b>81</b>
<b>9 Acknowledgements .....</b>	<b>81</b>

# 1 Introduction

This documentation describes the first version of the derived variables deposited on ELSA Wave 2 (2004 – 2005). There are a number of derived variables that have been deposited already, namely the Financial Derived Variables (Waves 1, 2 & 3), Pension Wealth Derived Variables (Wave 1 currently), and additionally within the Core Data: definitive age, year of birth and sex; financial unit type; letter cancellation task results; SOC and NS-SEC (Wave 0 and Wave 1 only); and highest educational qualification (Wave 1 only).

The variables have been derived on the latest version of the Wave 2 data available from the UK Data Archive (Wave 2 Core Data v3). If a derived variable is based on a variable that has not been archived then this has been indicated in the documentation.

In order to derive most of the variables in the Health section (“Ever Diagnosed...”) it was necessary to match in data (and respondents) from the Wave 1 archived dataset. This means that there are 12,207 respondents in the dataset rather than 9,432. Only those with a value of INWAVE of 2 or 3 (see page 13) will have values for variables outside of the Health section – you may wish to select out these cases only if you are not looking at the Health variables.

All information relevant to the derivation of each variable is provided in the format that is described below, with the variables first grouped into subject areas.

**Variable Label:** This is the label given to the variable. Derived variables can be distinguished from other variables by their labels, which always contain ‘(D)’ at the beginning. Note that the variable labels for all the derived variables are listed in the Contents Table (on the previous page), but that they do not include ‘(D)’ at the beginning (whereas they do in the actual dataset).

**Variable Name:** The name of the variable on the dataset.

**Value Labels:** These are the labels given to the different values that the data can take on the variable in question. Where there are no value labels (for example, on a continuous variable) this is indicated by ‘None’.

**Missing Value Labels:** These are the labels given to the values that are considered to be missing data. The standard missing values and their labels are as follows:

- 1 Not Applicable
- 8 Don’t Know
- 9 Refusal

Note that these are the same values as are used on the majority of the core ELSA data (with the exception of the self-completion module). On some derived variables these are not used, as there is no missing data (indicated by ‘None’), while on others there are additional categories of missing data.

**Description of Variable:** A brief description of the variable and any key issues to note. Any applicable background information or references will be included in this section.

**Population:** This defines whom the variable applies to and who should have a value for it.

On some variables, there are respondents to whom the variable applies but for whom the data is missing, usually due to item level non-response to the questions from which the variable is derived. These cases have a missing value that indicates why the data is missing e.g. -8 indicates it is missing due to a ‘Don’t Know’ response being given to the question.

**Level:** This indicates the level of the information given by the derived variable. For all of the variables in this dataset, the level is 'individual'.

## Derivation:

### **SPSS Code:**

This section provides the SPSS code used to derive and format the variable and assign variable and value labels.

**Note:** Before running the SPSS code it is crucial to ensure that the missing values are not declared (i.e. missing values are turned off) for all the variables used in the derivation. After the code has been run, these missing values must be re-declared as missing (turned on) for the variables used in the derivation and for the resulting derived variable.

This is because SPSS will not recognise the values of variables if they are declared missing. **The code for this is not supplied for each individual derivation.** The command enclosed in quotation marks should be used to turn off missing values: "missing values *varname* ()." and the following command will turn them back on again "missing values *varname* (-9 thru -1)."

## 2 Household Demographics

### Number of Natural Children Outside Household

**Variable name:** CHONAT

**Value labels:** None

**Description of variable:**

This variable gives the number of natural children a respondent has outside the household.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
numeric chonat (f2.0).
do if perid=1.
count chonat=dhcr01 dhcr17 dhcr33 dhcr49 dhcr65 dhcr81 dhcr97 dhcr113
dhcr129 dhcr145 dhcr161 dhcr177 dhcr193 dhcr209(3).
end if.
do if perid=2.
count chonat=dhcr02 dhcr18 dhcr34 dhcr50 dhcr66 dhcr82 dhcr98 dhcr114
dhcr130 dhcr146 dhcr162 dhcr194(3).
end if.
do if perid=3.
count chonat=dhcr03 dhcr19 dhcr35 dhcr51 dhcr67 dhcr83 dhcr99 dhcr115
dhcr131 (3).
end if.
do if perid=4.
count chonat=dhcr04 dhcr20 dhcr36 dhcr52 dhcr68 dhcr84 dhcr100 dhcr116
dhcr132 (3).
end if.
do if perid=5.
count chonat=dhcr05 dhcr21 dhcr37 dhcr53 dhcr69 dhcr85 dhcr101 dhcr117
dhcr133 (3).
end if.
do if perid=6.
count chonat=dhcr06 dhcr22 dhcr38 dhcr54 dhcr70 dhcr86 dhcr102 (3).
end if.
do if perid=7.
count chonat=dhcr07 dhcr23 dhcr39 dhcr55 dhcr71 dhcr87 dhcr103 (3).
end if.
do if perid=8.
count chonat=dhcr08 dhcr24 dhcr40 dhcr56 dhcr72 dhcr88 dhcr104 (3).
end if.
variable labels chonat '(D) Number of natural children outside the
household'.
execute.
```

## Number of Adopted Children Outside the Household

**Variable name:** CHOADP

**Value labels:** None

**Description of variable:**

This variable gives the number of adopted children a respondent has outside the household.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
numeric choadp (f2.0).
do if perid=1.
count choadp=dhcr01 dhcr17 dhcr33 dhcr49 dhcr65 dhcr81 dhcr97 dhcr113
dhcr129 dhcr145 dhcr161 dhcr177 (4).
end if.
do if perid=2.
count choadp=dhcr02 dhcr18 dhcr34 dhcr50 dhcr66 dhcr82 dhcr98 dhcr114
dhcr130 dhcr146 dhcr162 (4).
end if.
do if perid=3.
count choadp=dhcr03 dhcr19 dhcr35 dhcr51 dhcr67 dhcr83 dhcr99 dhcr115
dhcr131 (4).
end if.
do if perid=4.
count choadp=dhcr04 dhcr20 dhcr36 dhcr52 dhcr68 dhcr84 dhcr100 dhcr116
dhcr132 (4).
end if.
do if perid=5.
count choadp=dhcr05 dhcr21 dhcr37 dhcr53 dhcr69 dhcr85 dhcr101 dhcr117
dhcr133 (4).
end if.
do if perid=6.
count choadp=dhcr06 dhcr22 dhcr38 dhcr54 dhcr70 dhcr86 dhcr102 (4).
end if.
do if perid=7.
count choadp=dhcr07 dhcr23 dhcr39 dhcr55 dhcr71 dhcr87 dhcr103 (4).
end if.
do if perid=8.
count choadp=dhcr08 dhcr24 dhcr40 dhcr56 dhcr72 dhcr88 dhcr104 (4).
end if.
variable labels choadp "(D) Number of adopted children outside the
household".
execute.
```

## Number of Foster Children Outside the Household

**Variable name:** CHOFST

**Value labels:** None

**Description of variable:**

This variable gives the number of foster children the respondent has outside the household.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
numeric chofst (f2.0).
do if perid=1.
count chofst=dhcr01 dhcr17 dhcr33 dhcr49 dhcr65 dhcr81 dhcr97 dhcr113
dhcr129 dhcr145 dhcr161 dhcr177 dhcr193 (5).
end if.
do if perid=2.
count chofst=dhcr02 dhcr18 dhcr34 dhcr50 dhcr66 dhcr82 dhcr98 dhcr114
dhcr130 dhcr146 dhcr162 (5).
end if.
do if perid=3.
count chofst=dhcr03 dhcr19 dhcr35 dhcr51 dhcr67 dhcr83 dhcr99 dhcr115
dhcr131 (5).
end if.
do if perid=4.
count chofst=dhcr04 dhcr20 dhcr36 dhcr52 dhcr68 dhcr84 dhcr100 dhcr116
dhcr132 (5).
end if.
do if perid=5.
count chofst=dhcr05 dhcr21 dhcr37 dhcr53 dhcr69 dhcr85 dhcr101 dhcr117
dhcr133 (5).
end if.
do if perid=6.
count chofst=dhcr06 dhcr22 dhcr38 dhcr54 dhcr70 dhcr86 dhcr102 (5).
end if.
do if perid=7.
count chofst=dhcr07 dhcr23 dhcr39 dhcr55 dhcr71 dhcr87 dhcr103 (5).
end if.
do if perid=8.
count chofst=dhcr08 dhcr24 dhcr40 dhcr56 dhcr72 dhcr88 dhcr104 (5).
end if.
variable labels chofst "(D) Number of foster children outside the
household".
execute.
```

## Number of Stepchildren Outside the Household

**Variable name:** CHOSTP

**Value labels:** None

**Description of variable:**

This variable gives the number of stepchildren a respondent has outside the household.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
numeric chostp (f2.0).
do if perid=1.
count chostp=dhcr01 dhcr17 dhcr33 dhcr49 dhcr65 dhcr81 dhcr97 dhcr113
dhcr129 dhcr145 dhcr161 dhcr177 dhcr193 (6).
end if.
do if perid=2.
count chostp=dhcr02 dhcr18 dhcr34 dhcr50 dhcr66 dhcr82 dhcr98 dhcr114
dhcr130 dhcr146 dhcr162 (6).
end if.
do if perid=3.
count chostp=dhcr03 dhcr19 dhcr35 dhcr51 dhcr67 dhcr83 dhcr99 dhcr115
dhcr131 (6).
end if.
do if perid=4.
count chostp=dhcr04 dhcr20 dhcr36 dhcr52 dhcr68 dhcr84 dhcr100 dhcr116
dhcr132 (6).
end if.
do if perid=5.
count chostp=dhcr05 dhcr21 dhcr37 dhcr53 dhcr69 dhcr85 dhcr101 dhcr117
dhcr133 (6).
end if.
do if perid=6.
count chostp=dhcr06 dhcr22 dhcr38 dhcr54 dhcr70 dhcr86 dhcr102 (6).
end if.
do if perid=7.
count chostp=dhcr07 dhcr23 dhcr39 dhcr55 dhcr71 dhcr87 dhcr103 (6).
end if.
do if perid=8.
count chostp=dhcr08 dhcr24 dhcr40 dhcr56 dhcr72 dhcr88 dhcr104 (6).
end if.
variable labels chostp "(D) Number of stepchildren outside the
household".
execute.
```

## Total Number of Children Outside the Household

**Variable name:** CHOTOT

**Value labels:** None

**Description of variable:**

This variable gives the total number of children the respondent has outside the household.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
numeric chotot (f2.0).  
compute chotot= chonat + choadp + chofst + chostp .  
variable labels chotot "(D) Total number of children outside the  
household".  
execute.
```

## Number of Natural Children Inside the Household

**Variable name:** CHINAT

**Value labels:** None

**Description of variable:**

This variable gives the number of natural children the respondent has in the household.

## Number of Adopted Children Inside the Household

**Variable name:** CHIADP

**Value labels:** None

**Description of variable:**

This variable gives the number of adopted children the respondent has in the household.

## Number of Foster Children Inside the Household

**Variable name:** CHIFST

**Value labels:** None

**Description of variable:**

This variable gives the number of foster children the respondent has in the household.

## Number of Stepchildren Inside the Household

**Variable name:** CHISTP

**Value labels:** None

**Description of variable:**

This variable gives the number of stepchildren the respondent has in the household.

## Total Number of Children Inside the Household

**Variable name:** CHITOT

**Value labels:** None

**Description of variable:**

This variable gives the total number of children the respondent has in the household.

**Population (for all of the above):** All respondents      **Level (for all of the above):** Individual

**Derivation:** The syntax below is for all the five variables above.

```
numeric chinat chiadp chifst chistp chitot (f2.0).
count chinat = dhr dhr2 dhr3 dhr4 dhr5 dhr6 dhr7 dhr8 dhr9 dhr10 dhr11
dhr12 (8).
count chiadp = dhr dhr2 dhr3 dhr4 dhr5 dhr6 dhr7 dhr8 dhr9 dhr10 dhr11
dhr12 (9).
count chifst = dhr dhr2 dhr3 dhr4 dhr5 dhr6 dhr7 dhr8 dhr9 dhr10 dhr11
dhr12 (10).
count chistp = dhr dhr2 dhr3 dhr4 dhr5 dhr6 dhr7 dhr8 dhr9 dhr10 dhr11
dhr12 (11).
compute chitot= chinat + chiadp + chifst + chistp .
variable labels chinat "(D) Number of natural children inside the
household".
variable labels chiadp "(D) Number of adopted children inside the
household".
variable labels chifst "(D) Number of foster children inside the
household".
variable labels chistp "(D) Number of stepchildren inside the
household".
variable labels chitot "(D) Total number of children inside the
household".
execute.
```

## Total Number of Children (In and Out of Household)

**Variable name:** ALLTOTCH

**Value labels:** None

**Description of variable:**

This variable gives the total number of children the respondent has both inside and outside the household.

## Total Number of Natural Children (In and Out of Household)

**Variable name:** ALLNATCH

**Value labels:** None

**Description of variable:**

This variable gives the total number of natural children the respondent has both inside and outside the household.

## Total Number of Adopted Children (In and Out of Household)

**Variable name:** ALLADCH

**Value labels:** None

**Description of variable:**

This variable gives the total number of adopted children the respondent has both inside and outside the household.

## Total Number of Foster Children (In and Out of Household)

**Variable name:** ALLFSCH

**Value labels:** None

**Description of variable:**

This variable gives the total number of foster children the respondent has both inside and outside the household.

## Total Number of Stepchildren (In and Out of Household)

**Variable name:** ALLSPCH

**Value labels:** None

**Description of variable:**

This variable gives the total number of stepchildren the respondent has both inside and outside the household.

**Population (for all of the above):** All respondents      **Level (for all of the above):** Individual

**Derivation:** The syntax below is for all the five variables above.

```
numeric alltotch allnatch alladch allfsch allspch (f2.0).
compute alltotch= chotot + chitot.
compute allnatch= chonat + chinat.
compute alladch= choadp + chiadp.
compute allfsch= chofst + chifst.
compute allspch= chostp + chistp.
variable labels alltotch "(D) Total number of children (in and out of
household)".
variable labels allnatch "(D) Total number of natural children (in and
out of household)".
variable labels alladch "(D) Total number of adopted children (in and
out of household)".
variable labels allfsch "(D) Total number of foster children (in and
out of household)".
variable labels allspch "(D) Total number of step children (in and out
household)".
execute.
```

## Whether Natural Mother is Still Alive

**Variable name:** MOTHER

**Value labels:**

- 1 Mother alive
- 2 Mother dead

**Description of variable:**

This variable shows whether a respondent's natural mother was alive at the time of the Wave 2 interview.

**Population:** All respondents      **Level:** Individual

**Derivation:** This variable was derived by combining the data that was fedforward into the Wave 2 interview (FFDINMAD, not archived) as well as the answers given in the Wave 2 interview (for those who said their mother was alive previously or had not answered the question before), and whether their mother was present in the household (MAINHH1).

Note that for this variable and the three that follow it was necessary to recode some individual values. Details of this are not given in the syntax.

```

numeric mother (f2.0).
compute mother=dinma.
if dinma=-1 and mainhh1=1 mother=1.
if dinma=-1 and mainhh1~=1 mother=ffdinmad.
variable labels mother '(D) Whether natural mother is still alive'.
value labels mother 1 "Mother alive" 2 "Mother dead" -8 "Don't know"
-9 "Refusal".
execute.

```

## Natural Mother's Age (or Estimated Age) at Death

**Variable name:** MTHAGD

**Value labels:**

-5 Inaccurate response (age given under 12 years old)

**Description of variable:**

This variable shows the age of death of the natural mother of the respondent, if she has died.

**Population:** All respondents whose mother has died

**Level:** Individual

**Derivation:** This variable was derived by combining the data that was fedforward into the Wave 2 interview (FFDIMADD, not archived) as well as the answers given in the Wave 2 interview (for those who said their mother had died at Wave 2 or whose answer was missing before). As with the data on age of mother's death in the archived Wave 2 dataset (DIMAD), the ages deemed to be too young to be accurate answers have been re-coded to a value of -5.

```

numeric mthagd (f3.0).
compute mthagd=dimad.
if dimad=-1 mthagd=ffdimadd.
variable labels mthagd "(D) Natural mother's age (or estimated age) at
death".
recode mthagd (0 thru 11=-5).
execute.
value labels mthagd -1 'Not applicable' -5 'Inaccurate response (age
given under 12 years old)' -8 "Don't know" -9 'Refused'.

```

## Whether Natural Father is Still Alive

**Variable name:** FATHER

**Value labels:**

- 1 Father alive
- 2 Father dead

**Description of variable:**

This variable shows whether a respondent's natural mother was alive at the time of the Wave 2 interview.

**Population:** All respondents

**Level:** Individual

**Derivation:** This variable was derived by combining the data that was fedforward into the Wave 2 interview (FFDINFAD, not archived) as well as the answers given in the Wave 2 interview (for those who said their father was alive previously or had not answered the question before), and whether their mother was present in the household (PAINHH1).

```
numeric father (f2.0).
compute father=dinfa.
if dinfa=-1 and painhh1=1 father=1.
if dinfa=-1 and painhh1~=1 father=ffdinfad.
execute.
variable labels father '(D) Whether natural father is still alive'.
value labels father 1 "Father alive" 2 "Father dead" -8 "Don't know"
-9 "Refusal".
```

## Natural Father's Age (or Estimated Age) at Death

**Variable name:** FTHAGD

**Value labels:**

-5 Inaccurate response (age given under 12 years old)

**Description of variable:**

This variable shows the age of death of the natural father of the respondent, if he has died.

**Population:** All respondents whose father has died

**Level:** Individual

**Derivation:** This variable was derived by combining the data that was fedforward into the Wave 2 interview (FFDIFADD, not archived) as well as the answers given in the Wave 2 interview (for those who said their father had died at Wave 2 or whose answer was missing before). As with the data on age of father's death in the archived Wave 2 dataset (DIFAD), the ages deemed to be too young to be accurate answers have been re-coded to a value of -5.

```
numeric fthagd (f3.0).
compute fthagd=difad.
if difad=-1 fthagd=ffdifadd.
variable labels fthagd "(D) Natural father's age (or estimated age) at
death".
recode fthagd (0 thru 11=-5).
execute.
value labels fthagd -1 'Not applicable' -5 'Inaccurate response (age
given under 12 years old)' -8 "Don't know" -9 'Refused'.
```

## 3 Health

Many of the derived variables in this section give cumulated information regarding whether a respondent has ever reported a diagnosed disease or condition. The codes indicate in which wave the condition was first reported. It is important to note that respondents may give a date of diagnosis prior to an earlier interview. Therefore, for example, "By Wave 1" should be interpreted as "Reported by Wave 1" rather than "Diagnosed by Wave 1".

These variables do not necessarily indicate that a respondent still has a particular condition, just that they have reported it.

Categories where values were not consistent between Wave 1 and Wave 2 have been excluded from the derivations below.

There are two variables for diagnosed diabetes – one including those respondents who reported high blood sugar but not diabetes and one excluding those respondents.

There are a small number of cases who have missing values (i.e. -1, -8 or -9) for either wave's source variables. These cases usually either responded at the 'other' wave or have been coded -3 or -2 in the derived variables.

### Whether Respondent was in Wave 1, Wave 2 or both

**Variable name:** INWAVE

**Value labels:**

- 1 Wave 1 only
- 2 Wave 2 only
- 3 Both waves

**Description of variable:**

This variable shows whether a respondent is present in the archived data sets for Wave 1, Wave 2 or both.

**Population:** All respondents

**Level:** Individual

**Derivation:** This variable was derived by combining the Wave 1 and Wave 2 archived Core datasets and then looking at who was in each. A dummy variable with a value of 1 for each respondent in each wave was computed (INWAVE1 for Wave 1, INWAVE2 for Wave 2, neither archived) for all cases in each dataset before combining the datasets together.

```
NUMERIC inwave (F2.0).
IF inwave1 = 1 AND inwave2 = 1 inwave = 3.
IF inwave1 = 1 AND SYSMIS(inwave2) inwave = 1.
IF SYSMIS(inwave1) AND inwave2 = 1 inwave = 2.
VAR LAB inwave 'Whether respondent was in Wave 1, Wave 2 or both'.
VAL LAB inwave
1 'Wave 1 only'
2 'Wave 2 only'
3 'Both waves'.
```

## Ever Reported High Blood Pressure (Diagnosed)

**Variable name:** HEDIMBP

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported high blood pressure and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```
NUMERIC hedimbp (F2.0).
DO IF bhedim01 ~= 1 AND bhedim02 ~= 1 AND bhedim03 ~= 1 AND bhedim04 ~=
1 AND bhedim05 ~= 1 AND bhedim06 ~= 1 AND bhedim07 ~= 1 AND chedim01 ~=
1 AND chedim02 ~= 1 AND chedim03 ~= 1 AND chedim04 ~= 1 AND chedim05 ~=
1 AND chedim06 ~= 1 AND chedim07 ~= 1 AND chedim08 ~= 1.
COMPUTE hedimbp = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 1 OR bhedim02 = 1 OR bhedim03 = 1 OR bhedim04 = 1 OR
bhedim05 = 1 OR bhedim06 = 1 OR bhedim07 = 1.
COMPUTE hedimbp = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedimbp) AND (chedim01 = 1 OR chedim02 = 1 OR chedim03 = 1
OR chedim04 = 1 OR chedim05 = 1 OR chedim06 = 1 OR chedim07 = 1 OR
chedim08 = 1).
COMPUTE hedimbp = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedimbp).
COMPUTE hedimbp = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedimbp).
COMPUTE hedimbp = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedimbp = -1.
END IF.
EXECUTE.
VAR LAB hedimbp '(D) Ever reported high blood pressure (diagnosed)'.
VAL LAB hedimbp
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.
```

## Ever Reported Angina (Diagnosed)

**Variable name:** HEDIMAN

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported angina and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```
NUMERIC hediman (F2.0).
DO IF bhedim01 ~= 2 AND bhedim02 ~= 2 AND bhedim03 ~= 2 AND bhedim04
  ~= 2 AND bhedim05 ~= 2 AND bhedim06 ~= 2 AND bhedim07 ~= 2 AND
  chedim01 ~= 2 AND chedim02 ~= 2 AND chedim03 ~= 2 AND chedim04 ~= 2
  AND chedim05 ~= 2 AND chedim06 ~= 2 AND chedim07 ~= 2 AND chedim08 ~=
  2.
COMPUTE hediman = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 2 OR bhedim02 = 2 OR bhedim03 = 2 OR bhedim04 = 2 OR
  bhedim05 = 2 OR bhedim06 = 2 OR bhedim07 = 2.
COMPUTE hediman = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hediman) AND (chedim01 = 2 OR chedim02 = 2 OR chedim03 =
  2 OR chedim04 = 2 OR chedim05 = 2 OR chedim06 = 2 OR chedim07 = 2 OR
  chedim08 = 2).
COMPUTE hediman = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hediman).
COMPUTE hediman = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hediman).
COMPUTE hediman = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hediman = -1.
END IF.
EXECUTE.
```

```

VAR LAB hediman '(D) Ever reported angina (diagnosed)'.
VAL LAB hediman
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Myocardial Infarction (Diagnosed)

**Variable name:** HEDIMMI

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported a heart attack (including myocardial infarction or coronary thrombosis) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```

NUMERIC hedimmi (F2.0).
DO IF bhedim01 ~= 3 AND bhedim02 ~= 3 AND bhedim03 ~= 3 AND bhedim04
~= 3 AND bhedim05 ~= 3 AND bhedim06 ~= 3 AND bhedim07 ~= 3 AND
chedim01 ~= 3 AND chedim02 ~= 3 AND chedim03 ~= 3 AND chedim04 ~= 3
AND chedim05 ~= 3 AND chedim06 ~= 3 AND chedim07 ~= 3 AND chedim08 ~=
3.
COMPUTE hedimmi = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 3 OR bhedim02 = 3 OR bhedim03 = 3 OR bhedim04 = 3 OR
bhedim05 = 3 OR bhedim06 = 3 OR bhedim07 = 3.
COMPUTE hedimmi = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedimmi) AND (chedim01 = 3 OR chedim02 = 3 OR chedim03 =
3 OR chedim04 = 3 OR chedim05 = 3 OR chedim06 = 3 OR chedim07 = 3 OR
chedim08 = 3).
COMPUTE hedimmi = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedimmi).
COMPUTE hedimmi = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedimmi).
COMPUTE hedimmi = -2.
END IF.
EXECUTE.

```

```

DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedimmi = -1.
END IF.
EXECUTE.
VAR LAB hedimmi '(D) Ever reported myocardial infarction (diagnosed)'.
VAL LAB hedimmi
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Congestive Heart Failure (Diagnosed)

**Variable name:** HEDIMHF

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported congestive heart failure and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```

NUMERIC hedimhf (F2.0).
DO IF bhedim01 ~= 4 AND bhedim02 ~= 4 AND bhedim03 ~= 4 AND bhedim04
~= 4 AND bhedim05 ~= 4 AND bhedim06 ~= 4 AND bhedim07 ~= 4 AND
chedim01 ~= 4 AND chedim02 ~= 4 AND chedim03 ~= 4 AND chedim04 ~= 4
AND chedim05 ~= 4 AND chedim06 ~= 4 AND chedim07 ~= 4 AND chedim08 ~=
4.
COMPUTE hedimhf = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 4 OR bhedim02 = 4 OR bhedim03 = 4 OR bhedim04 = 4 OR
bhedim05 = 4 OR bhedim06 = 4 OR bhedim07 = 4.
COMPUTE hedimhf = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedimhf) AND (chedim01 = 4 OR chedim02 = 4 OR chedim03 =
4 OR chedim04 = 4 OR chedim05 = 4 OR chedim06 = 4 OR chedim07 = 4 OR
chedim08 = 4).
COMPUTE hedimhf = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedimhf).
COMPUTE hedimhf = -3.
END IF.
EXECUTE.

```

```

DO IF inwave = 2 AND SYSMIS(hedimhf).
COMPUTE hedimhf = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedimhf = -1.
END IF.
EXECUTE.
VAR LAB hedimhf '(D) Ever reported congestive heart failure
(diagnosed)'.
VAL LAB hedimhf
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Heart Murmur (Diagnosed)

**Variable name:** HEDIMHM

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported a heart murmur and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```

NUMERIC hedimhm (F2.0).
DO IF bhedim01 ~= 5 AND bhedim02 ~= 5 AND bhedim03 ~= 5 AND bhedim04 ~= 5
AND bhedim05 ~= 5 AND bhedim06 ~= 5 AND bhedim07 ~= 5 AND chedim01 ~= 5
AND chedim02 ~= 5 AND chedim03 ~= 5 AND chedim04 ~= 5 AND chedim05 ~= 5
AND chedim06 ~= 5 AND chedim07 ~= 5 AND chedim08 ~= 5.
COMPUTE hedimhm = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 5 OR bhedim02 = 5 OR bhedim03 = 5 OR bhedim04 = 5 OR
bhedim05 = 5 OR bhedim06 = 5 OR bhedim07 = 5.
COMPUTE hedimhm = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedimhm) AND (chedim01 = 5 OR chedim02 = 5 OR chedim03 = 5
OR chedim04 = 5 OR chedim05 = 5 OR chedim06 = 5 OR chedim07 = 5 OR
chedim08 = 5).
COMPUTE hedimhm = 2.
END IF.
EXECUTE.

```

```

DO IF inwave = 1 AND SYSMIS(hedimhm).
COMPUTE hedimhm = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedimhm).
COMPUTE hedimhm = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedimhm = -1.
END IF.
EXECUTE.
VAR LAB hedimhm '(D) Ever reported heart murmur (diagnosed)'.
VAL LAB hedimhm
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Arrhythmia (Diagnosed)

**Variable name:** HEDIMAR

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported arrhythmia (an abnormal heart rhythm) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```

NUMERIC hedimar (F2.0).
DO IF bhedim01 ~= 6 AND bhedim02 ~= 6 AND bhedim03 ~= 6 AND bhedim04 ~=
6 AND bhedim05 ~= 6 AND bhedim06 ~= 6 AND bhedim07 ~= 6 AND chedim01 ~=
6 AND chedim02 ~= 6 AND chedim03 ~= 6 AND chedim04 ~= 6 AND chedim05 ~=
6 AND chedim06 ~= 6 AND chedim07 ~= 6 AND chedim08 ~= 6.
COMPUTE hedimar = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 6 OR bhedim02 = 6 OR bhedim03 = 6 OR bhedim04 = 6 OR
bhedim05 = 6 OR bhedim06 = 6 OR bhedim07 = 6.
COMPUTE hedimar = 1.
END IF.
EXECUTE.

```

```

DO IF SYSMIS(hedimar) AND (chedim01 = 6 OR chedim02 = 6 OR chedim03 =
6 OR chedim04 = 6 OR chedim05 = 6 OR chedim06 = 6 OR chedim07 = 6 OR
chedim08 = 6).
COMPUTE hedimar = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedimar).
COMPUTE hedimar = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedimar).
COMPUTE hedimar = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedimar = -1.
END IF.
EXECUTE.
VAR LAB hedimar '(D) Ever reported arrhythmia (diagnosed)'.
VAL LAB hedimar
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Diabetes or High Blood Sugar (Diagnosed)

**Variable name:** HEDIMDI

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported diabetes or high blood sugar and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```

NUMERIC hedimdi (F2.0).
DO IF bhedim01 ~= 7 AND bhedim02 ~= 7 AND bhedim03 ~= 7 AND bhedim04
~= 7 AND bhedim05 ~= 7 AND bhedim06 ~= 7 AND bhedim07 ~= 7 AND
chedim01 ~= 7 AND chedim02 ~= 7 AND chedim03 ~= 7 AND chedim04 ~= 7
AND chedim05 ~= 7 AND chedim06 ~= 7 AND chedim07 ~= 7 AND chedim08 ~=
7.
COMPUTE hedimdi = 0.
END IF.
EXECUTE.
DO IF bhedim01 = 7 OR bhedim02 = 7 OR bhedim03 = 7 OR bhedim04 = 7 OR
bhedim05 = 7 OR bhedim06 = 7 OR bhedim07 = 7.
COMPUTE hedimdi = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedimdi) AND (chedim01 = 7 OR chedim02 = 7 OR chedim03 =
7 OR chedim04 = 7 OR chedim05 = 7 OR chedim06 = 7 OR chedim07 = 7 OR
chedim08 = 7).
COMPUTE hedimdi = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedimdi).
COMPUTE hedimdi = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedimdi).
COMPUTE hedimdi = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedimdi = -1.
END IF.
EXECUTE.
VAR LAB hedimdi '(D) Ever reported diabetes or high blood sugar
(diagnosed)'.
VAL LAB hedimdi
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Diabetes (Diagnosed)

**Variable name:** HEDBTS

**Value labels:**

- 8 Don't know
- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported diabetes and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08, CHEACD and INWAVE were used in this derivation. CHEACD was used to distinguish between diabetes *or* high blood sugar and diabetes only.

```

NUMERIC hedbts (F2.0).
DO IF bhedim01 ~= 7 AND bhedim02 ~= 7 AND bhedim03 ~= 7 AND bhedim04 ~=
7 AND bhedim05 ~= 7 AND bhedim06 ~= 7 AND bhedim07 ~= 7 AND chedim01 ~=
7 AND chedim02 ~= 7 AND chedim03 ~= 7 AND chedim04 ~= 7 AND chedim05 ~=
7 AND chedim06 ~= 7 AND chedim07 ~= 7 AND chedim08 ~= 7.
COMPUTE hedbts = 0.
END IF.
EXECUTE.
DO IF cheacd = 1 AND (bhedim01 = 7 OR bhedim02 = 7 OR bhedim03 = 7 OR
bhedim04 = 7 OR bhedim05 = 7 OR bhedim06 = 7 OR bhedim07 = 7).
COMPUTE hedbts = 1.
END IF.
EXECUTE.
DO IF cheacd = 1 AND (chedim01 = 7 OR chedim02 = 7 OR chedim03 = 7 OR
chedim04 = 7 OR chedim05 = 7 OR chedim06 = 7 OR chedim07 = 7 OR chedim08
= 7).
COMPUTE hedbts = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedbts).
COMPUTE hedbts = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedbts).
COMPUTE hedbts = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedim01 = -1 AND chedim01 = -1.
COMPUTE hedbts = -1.
END IF.
EXECUTE.
DO IF inwave = 3 AND cheacd = 2.
COMPUTE hedbts = 0.
END IF.
EXECUTE.
DO IF SYSMIS(hedbts).
COMPUTE hedbts = -8.
END IF.
EXECUTE.
VAR LAB hedbts '(D) Ever reported diabetes (diagnosed)'.
VAL LAB hedbts
-8 "Don't know"
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported a Stroke (Diagnosed)

**Variable name:** HEDIMST

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported a stroke (cerebral vascular disease) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIM01-BHEDIM07, CHEDIM01-CHEDIM08 and INWAVE were used in this derivation.

```
NUMERIC hedimst (F2.0).
IF (bhedim01 ~= 8 AND bhedim02 ~= 8 AND bhedim03 ~= 8 AND bhedim04 ~=
8 AND bhedim05 ~= 8 AND bhedim06 ~= 8 AND bhedim07 ~= 8 AND chedim01
~= 8 AND chedim02 ~= 8 AND chedim03 ~= 8 AND chedim04 ~= 8 AND
chedim05 ~= 8 AND chedim06 ~= 8 AND chedim07 ~= 8 AND chedim08 ~= 8)
hedimst = 0.
IF (bhedim01 = 8 OR bhedim02 = 8 OR bhedim03 = 8 OR bhedim04 = 8 OR
bhedim05 = 8 OR bhedim06 = 8 OR bhedim07 = 8) hedimst = 1.
IF (SYSMIS(hedimst) AND (chedim01 = 8 OR chedim02 = 8 OR chedim03 = 8
OR chedim04 = 8 OR chedim05 = 8 OR chedim06 = 8 OR chedim07 = 8 OR
chedim08 = 8)) hedimst = 2.
EXECUTE.
DO IF (inwave = 1 AND SYSMIS(hedimst)).
COMPUTE hedimst = -3.
END IF.
EXECUTE.
DO IF (inwave = 2 AND SYSMIS(hedimst)).
COMPUTE hedimst = -2.
END IF.
EXECUTE.
IF (inwave = 3 AND bhedim01 = -1 AND chedim01 = -1) hedimst = -1.
EXECUTE.
VAR LAB hedimst '(D) Ever reported stroke (diagnosed)'.
VAL LAB hedimst
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.
```

## Ever Reported any of 7 CVD-related Diseases (Excluding High Blood Pressure)

**Variable name:** CVD7DIHB

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported any of 7 CVD related diseases (arrhythmia, myocardial infarction, congestive heart failure, angina, heart murmur, diabetes or high blood sugar, stroke) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The previously derived variables HEDIMAN, HEDIMMI, HEDIMHF, HEDIMAR, HEDIMHM, HEDIMDI, HEDIMST and INWAVE were used in this derivation.

```
NUMERIC cvd7dihb (F2.0).
IF ANY (1, hediman, hedimmi, hedimhf, hedimar, hedimhm, hedimdi,
hedimst) cvd7dihb = 1.
EXECUTE.
IF (SYSMIS(cvd7dihb) AND (hediman = 2 OR hedimmi = 2 OR hedimhf = 2 OR
hedimar = 2 OR hedimhm = 2 OR hedimdi = 2 OR hedimst = 2)) cvd7dihb =
2.
EXECUTE.
IF (SYSMIS(cvd7dihb) AND hediman = 0 AND hedimmi = 0 AND hedimhf = 0
AND hedimar = 0 AND hedimhm = 0 AND hedimdi = 0 AND hedimst = 0)
cvd7dihb = 0.
EXECUTE.
DO IF (inwave = 1 AND SYSMIS(cvd7dihb)).
COMPUTE cvd7dihb = -3.
END IF.
EXECUTE.
DO IF (inwave = 2 AND SYSMIS(cvd7dihb)).
COMPUTE cvd7dihb = -2.
END IF.
EXECUTE.
IF (inwave = 3 AND bhedim01 = -1 AND chedim01 = -1) cvd7dihb = -1.
EXECUTE.
VAR LAB cvd7dihb "(D) Ever reported any of 7 cvd-related diseases
(excluding high blood pressure)".
VAL LAB cvd7dihb
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.
```

## Ever Reported any of 7 CVD-related Diseases (Excluding High Blood Sugar and High Blood Pressure)

**Variable name:** CVD7DBTS

**Value labels:**

- 8 Don't know
- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported any of 7 CVD related diseases (arrhythmia, myocardial infarction, congestive heart failure, angina, heart murmur, diabetes, and stroke) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The previously derived variables HEDIMAN, HEDIMMI, HEDIMHF, HEDIMAR, HEDIMHM, HEDBTS, HEDIMST and INWAVE were used in this derivation.

```
NUMERIC cvd7dbts (F2.0).
IF ANY (1, hediman, hedimmi, hedimhf, hedimar, hedimhm, hedbts,
hedimst) cvd7dbts = 1.
EXECUTE.
IF (SYSMIS(cvd7dbts) AND (hediman = 2 OR hedimmi = 2 OR hedimhf = 2 OR
hedimar = 2 OR hedimhm = 2 OR hedbts = 2 OR hedimst = 2)) cvd7dbts =
2.
EXECUTE.
IF (SYSMIS(cvd7dbts) AND hediman = 0 AND hedimmi = 0 AND hedimhf = 0
AND hedimar = 0 AND hedimhm = 0 AND hedbts = 0 AND hedimst = 0)
cvd7dbts = 0.
EXECUTE.
DO IF (inwave = 1 AND SYSMIS(cvd7dbts)).
COMPUTE cvd7dbts = -3.
END IF.
EXECUTE.
DO IF (inwave = 2 AND SYSMIS(cvd7dbts)).
COMPUTE cvd7dbts = -2.
END IF.
EXECUTE.
DO IF (inwave = 3 AND bhedim01 = -1 AND chedim01 = -1).
COMPUTE cvd7dbts = -1.
END IF.
EXECUTE.
IF (SYSMIS(cvd7dbts)) cvd7dbts = -8.
EXECUTE.
```

```

VAR LAB cvd7dbts "(D) Ever reported any of 7 cvd-related diseases
(excluding high blood sugar and high blood pressure)".
VAL LAB cvd7dbts
-8 "Don't know"
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Hedibonic Lung Disease (Diagnosed)

**Variable name:** HEDIBLU

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported hedibonic lung disease (chronic lung disease such as chronic bronchitis or emphysema) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```

NUMERIC hediblu (F2.0).
DO IF bhedib01 ~= 1 AND bhedib02 ~= 1 AND bhedib03 ~= 1 AND bhedib04
~= 1 AND bhedib05 ~= 1 AND bhedib06 ~= 1 AND bhedib07 ~= 1 AND
bhedib08 ~= 1 AND bhedib09 ~= 1 AND bhedib10 ~= 1 AND chedib01 ~= 1
AND chedib02 ~= 1 AND chedib03 ~= 1 AND chedib04 ~= 1.
COMPUTE hediblu = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 1 OR bhedib02 = 1 OR bhedib03 = 1 OR bhedib04 = 1 OR
bhedib05 = 1 OR bhedib06 = 1 OR bhedib07 = 1 OR bhedib08 = 1 OR
bhedib09 = 1 OR bhedib10 = 1.
COMPUTE hediblu = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hediblu) AND (chedib01 = 1 OR chedib02 = 1 OR chedib03 =
1 OR chedib04 = 1).
COMPUTE hediblu = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hediblu).
COMPUTE hediblu = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hediblu).
COMPUTE hediblu = -2.
END IF.
EXECUTE.

```

```

DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hediblu = -1.
END IF.
EXECUTE.
VAR LAB hediblu '(D) Ever reported hedibonic lung disease
(diagnosed)'.
VAL LAB hediblu
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Asthma (Diagnosed)

**Variable name:** HEDIBAS

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported asthma and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```

NUMERIC hedibas (F2.0).
DO IF bhedib01 ~= 2 AND bhedib02 ~= 2 AND bhedib03 ~= 2 AND bhedib04
~= 2 AND bhedib05 ~= 2 AND bhedib06 ~= 2 AND bhedib07 ~= 2 AND
bhedib08 ~= 2 AND bhedib09 ~= 2 AND bhedib10 ~= 2 AND chedib01 ~= 2
AND chedib02 ~= 2 AND chedib03 ~= 2 AND chedib04 ~= 2.
COMPUTE hedibas = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 2 OR bhedib02 = 2 OR bhedib03 = 2 OR bhedib04 = 2 OR
bhedib05 = 2 OR bhedib06 = 2 OR bhedib07 = 2 OR bhedib08 = 2 OR
bhedib09 = 2 OR bhedib10 = 2.
COMPUTE hedibas = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibas) AND (chedib01 = 2 OR chedib02 = 2 OR chedib03 =
2 OR chedib04 = 2).
COMPUTE hedibas = 2.
END IF.
EXECUTE.

```

```

DO IF inwave = 1 AND SYSMIS(hedibas).
COMPUTE hedibas = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibas).
COMPUTE hedibas = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibas = -1.
END IF.
EXECUTE.
VAR LAB hedibas '(D) Ever reported asthma (diagnosed)'.
VAL LAB hedibas
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Arthritis (Diagnosed)

**Variable name:** HEDIBAR

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported arthritis (including osteoarthritis and rheumatism) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```

NUMERIC hedibar (F2.0).
DO IF bhedib01 ~= 3 AND bhedib02 ~= 3 AND bhedib03 ~= 3 AND bhedib04
~= 3 AND bhedib05 ~= 3 AND bhedib06 ~= 3 AND bhedib07 ~= 3 AND
bhedib08 ~= 3 AND bhedib09 ~= 3 AND bhedib10 ~= 3 AND chedib01 ~= 3
AND chedib02 ~= 3 AND chedib03 ~= 3 AND chedib04 ~= 3.
COMPUTE hedibar = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 3 OR bhedib02 = 3 OR bhedib03 = 3 OR bhedib04 = 3 OR
bhedib05 = 3 OR bhedib06 = 3 OR bhedib07 = 3 OR bhedib08 = 3 OR
bhedib09 = 3 OR bhedib10 = 3.
COMPUTE hedibar = 1.
END IF.
EXECUTE.

```

```

DO IF SYSMIS(hedibar) AND (chedib01 = 3 OR chedib02 = 3 OR chedib03 =
3 OR chedib04 = 3).
COMPUTE hedibar = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibar).
COMPUTE hedibar = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibar).
COMPUTE hedibar = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibar = -1.
END IF.
EXECUTE.
VAR LAB hedibar '(D) Ever reported arthritis (diagnosed)'.
VAL LAB hedibar
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Osteoporosis (Diagnosed)

**Variable name:** HEDIBOS

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported osteoporosis (sometimes called thin or brittle bones) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```

NUMERIC hedibos (F2.0).
DO IF bhedib01 ~= 4 AND bhedib02 ~= 4 AND bhedib03 ~= 4 AND bhedib04
~= 4 AND bhedib05 ~= 4 AND bhedib06 ~= 4 AND bhedib07 ~= 4 AND
bhedib08 ~= 4 AND bhedib09 ~= 4 AND bhedib10 ~= 4 AND chedib01 ~= 4
AND chedib02 ~= 4 AND chedib03 ~= 4 AND chedib04 ~= 4.
COMPUTE hedibos = 0.
END IF.
EXECUTE.

```

```

DO IF bhedib01 = 4 OR bhedib02 = 4 OR bhedib03 = 4 OR bhedib04 = 4 OR
bhedib05 = 4 OR bhedib06 = 4 OR bhedib07 = 4 OR bhedib08 = 4 OR
bhedib09 = 4 OR bhedib10 = 4.
COMPUTE hedibos = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibos) AND (chedib01 = 4 OR chedib02 = 4 OR chedib03 =
4 OR chedib04 = 4).
COMPUTE hedibos = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibos).
COMPUTE hedibos = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibos).
COMPUTE hedibos = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibos = -1.
END IF.
EXECUTE.
VAR LAB hedibos '(D) Ever reported osteoporosis (diagnosed)'.
VAL LAB hedibos
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Cancer (Diagnosed)

**Variable name:** HEDIBCA

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported cancer or a malignant tumour (excluding minor skin cancers) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```

NUMERIC hedibca (F2.0).
DO IF bhedib01 ~= 5 AND bhedib02 ~= 5 AND bhedib03 ~= 5 AND bhedib04
~= 5 AND bhedib05 ~= 5 AND bhedib06 ~= 5 AND bhedib07 ~= 5 AND
bhedib08 ~= 5 AND bhedib09 ~= 5 AND bhedib10 ~= 5 AND chedib01 ~= 5
AND chedib02 ~= 5 AND chedib03 ~= 5 AND chedib04 ~= 5.
COMPUTE hedibca = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 5 OR bhedib02 = 5 OR bhedib03 = 5 OR bhedib04 = 5 OR
bhedib05 = 5 OR bhedib06 = 5 OR bhedib07 = 5 OR bhedib08 = 5 OR
bhedib09 = 5 OR bhedib10 = 5.
COMPUTE hedibca = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibca) AND (chedib01 = 5 OR chedib02 = 5 OR chedib03 =
5 OR chedib04 = 5).
COMPUTE hedibca = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibca).
COMPUTE hedibca = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibca).
COMPUTE hedibca = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibca = -1.
END IF.
EXECUTE.
VAR LAB hedibca '(D) Ever reported cancer (diagnosed)'.
VAL LAB hedibca
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Parkinson's Disease (Diagnosed)

**Variable name:** HEDIBPD

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported Parkinson's disease and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```
NUMERIC hedibpd (F2.0).
DO IF bhedib01 ~= 6 AND bhedib02 ~= 6 AND bhedib03 ~= 6 AND bhedib04
~= 6 AND bhedib05 ~= 6 AND bhedib06 ~= 6 AND bhedib07 ~= 6 AND
bhedib08 ~= 6 AND bhedib09 ~= 6 AND bhedib10 ~= 6 AND chedib01 ~= 6
AND chedib02 ~= 6 AND chedib03 ~= 6 AND chedib04 ~= 6.
COMPUTE hedibpd = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 6 OR bhedib02 = 6 OR bhedib03 = 6 OR bhedib04 = 6 OR
bhedib05 = 6 OR bhedib06 = 6 OR bhedib07 = 6 OR bhedib08 = 6 OR
bhedib09 = 6 OR bhedib10 = 6.
COMPUTE hedibpd = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibpd) AND (chedib01 = 6 OR chedib02 = 6 OR chedib03 =
6 OR chedib04 = 6).
COMPUTE hedibpd = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibpd).
COMPUTE hedibpd = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibpd).
COMPUTE hedibpd = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibpd = -1.
END IF.
EXECUTE.
VAR LAB hedibpd "(D) Ever reported Parkinson's Disease (diagnosed)".
VAL LAB hedibpd
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.
```

## Ever Reported Psychiatric Disorder (Diagnosed)

**Variable name:** HEDIBPS

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported a psychiatric disorder (including any emotional, nervous or psychiatric problems) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```
NUMERIC hedibps (F2.0).
DO IF bhedib01 ~= 7 AND bhedib02 ~= 7 AND bhedib03 ~= 7 AND bhedib04
  ~= 7 AND bhedib05 ~= 7 AND bhedib06 ~= 7 AND bhedib07 ~= 7 AND
  bhedib08 ~= 7 AND bhedib09 ~= 7 AND bhedib10 ~= 7 AND chedib01 ~= 7
  AND chedib02 ~= 7 AND chedib03 ~= 7 AND chedib04 ~= 7.
COMPUTE hedibps = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 7 OR bhedib02 = 7 OR bhedib03 = 7 OR bhedib04 = 7 OR
  bhedib05 = 7 OR bhedib06 = 7 OR bhedib07 = 7 OR bhedib08 = 7 OR
  bhedib09 = 7 OR bhedib10 = 7.
COMPUTE hedibps = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibps) AND (chedib01 = 7 OR chedib02 = 7 OR chedib03 =
  7 OR chedib04 = 7).
COMPUTE hedibps = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibps).
COMPUTE hedibps = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibps).
COMPUTE hedibps = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibps = -1.
END IF.
EXECUTE.
VAR LAB hedibps '(D) Ever reported psychiatric disorder (diagnosed)'.
VAL LAB hedibps
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.
```

## Ever Reported Alzheimer's Disease (Diagnosed)

**Variable name:** HEDIBAD

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported Alzheimer's disease and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```
NUMERIC hedibad (F2.0).
DO IF bhedib01 ~= 8 AND bhedib02 ~= 8 AND bhedib03 ~= 8 AND bhedib04
  ~= 8 AND bhedib05 ~= 8 AND bhedib06 ~= 8 AND bhedib07 ~= 8 AND
  bhedib08 ~= 8 AND bhedib09 ~= 8 AND bhedib10 ~= 8 AND chedib01 ~= 8
  AND chedib02 ~= 8 AND chedib03 ~= 8 AND chedib04 ~= 8.
COMPUTE hedibad = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 8 OR bhedib02 = 8 OR bhedib03 = 8 OR bhedib04 = 8 OR
  bhedib05 = 8 OR bhedib06 = 8 OR bhedib07 = 8 OR bhedib08 = 8 OR
  bhedib09 = 8 OR bhedib10 = 8.
COMPUTE hedibad = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibad) AND (chedib01 = 8 OR chedib02 = 8 OR chedib03 =
  8 OR chedib04 = 8).
COMPUTE hedibad = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibad).
COMPUTE hedibad = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibad).
COMPUTE hedibad = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibad = -1.
END IF.
EXECUTE.
VAR LAB hedibad "(D) Ever reported Alzheimer's Disease (diagnosed)".
VAL LAB hedibad -3 'In Wave 1 only and did not report'
  -2 'In Wave 2 only and did not report' 0 'Neither wave'
  1 'By Wave 1' 2 'By Wave 2'.
```

## Ever Reported Dementia or Memory Impairment (Diagnosed)

**Variable name:** HEDIBDE

**Value labels:**

- 3 In Wave 1 only and did not report
- 2 In Wave 2 only and did not report
- 0 Neither wave
- 1 By Wave 1
- 2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported dementia (senility or another serious memory impairment) and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEDIB01-BHEDIB10, CHEDIB01-CHEDIB04 and INWAVE were used in this derivation.

```
NUMERIC hedibde (F2.0).
DO IF bhedib01 ~= 9 AND bhedib02 ~= 9 AND bhedib03 ~= 9 AND bhedib04
  ~= 9 AND bhedib05 ~= 9 AND bhedib06 ~= 9 AND bhedib07 ~= 9 AND
  bhedib08 ~= 9 AND bhedib09 ~= 9 AND bhedib10 ~= 9 AND chedib01 ~= 9
  AND chedib02 ~= 9 AND chedib03 ~= 9 AND chedib04 ~= 9.
COMPUTE hedibde = 0.
END IF.
EXECUTE.
DO IF bhedib01 = 9 OR bhedib02 = 9 OR bhedib03 = 9 OR bhedib04 = 9 OR
  bhedib05 = 9 OR bhedib06 = 9 OR bhedib07 = 9 OR bhedib08 = 9 OR
  bhedib09 = 9 OR bhedib10 = 9.
COMPUTE hedibde = 1.
END IF.
EXECUTE.
DO IF SYSMIS(hedibde) AND (chedib01 = 9 OR chedib02 = 9 OR chedib03 =
  9 OR chedib04 = 9).
COMPUTE hedibde = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(hedibde).
COMPUTE hedibde = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(hedibde).
COMPUTE hedibde = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bhedib01 = -1 AND chedib01 = -1.
COMPUTE hedibde = -1.
END IF.
EXECUTE.
```

```

VAR LAB hedibde '(D) Ever reported dementia or memory impairment
(diagnosed)'.
VAL LAB hedibde
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Glaucoma (Diagnosed)

**Variable name:** HEOPTGL

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported glaucoma and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEOPT1-BHEOPT5, CHEOPT1-CHEOPT2 and INWAVE were used in this derivation.

```

NUMERIC heoptgl (F2.0).
DO IF bheopt1 ~= 1 AND bheopt2 ~= 1 AND bheopt3 ~= 1 AND bheopt4 ~= 1
AND bheopt5 ~= 1 AND cheopt1 ~= 1 AND cheopt2 ~= 1.
COMPUTE heoptgl = 0.
END IF.
EXECUTE.
DO IF bheopt1 = 1 OR bheopt2 = 1 OR bheopt3 = 1 OR bheopt4 = 1 OR
bheopt5 = 1.
COMPUTE heoptgl = 1.
END IF.
EXECUTE.
DO IF SYSMIS(heoptgl) AND (cheopt1 = 1 OR cheopt2 = 1).
COMPUTE heoptgl = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(heoptgl).
COMPUTE heoptgl = -3.
END IF.
EXECUTE.
DO IF inwave = 2 AND SYSMIS(heoptgl).
COMPUTE heoptgl = -2.
END IF.
EXECUTE.

```

```

DO IF inwave = 3 AND bheopt1 = -1 AND cheopt1 = -1.
COMPUTE heoptgl = -1.
END IF.
EXECUTE.
VAR LAB heoptgl '(D) Ever reported glaucoma (diagnosed)'.
VAL LAB heoptgl
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Diabetic Eye Disease (Diagnosed)

**Variable name:** HEOPTDI

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported diabetic eye disease and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEOPT1-BHEOPT5, CHEOPT1-CHEOPT2 and INWAVE were used in this derivation.

```

NUMERIC heoptdi (F2.0).
DO IF bheopt1 ~= 2 AND bheopt2 ~= 2 AND bheopt3 ~= 2 AND bheopt4 ~= 2
AND bheopt5 ~= 2 AND cheopt1 ~= 2 AND cheopt2 ~= 2.
COMPUTE heoptdi = 0.
END IF.
EXECUTE.
DO IF bheopt1 = 2 OR bheopt2 = 2 OR bheopt3 = 2 OR bheopt4 = 2 OR
bheopt5 = 2.
COMPUTE heoptdi = 1.
END IF.
EXECUTE.
DO IF SYSMIS(heoptdi) AND (cheopt1 = 2 OR cheopt2 = 2).
COMPUTE heoptdi = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(heoptdi).
COMPUTE heoptdi = -3.
END IF.
EXECUTE.

```

```

DO IF inwave = 2 AND SYSMIS(heoptdi).
COMPUTE heoptdi = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bheopt1 = -1 AND cheopt1 = -1.
COMPUTE heoptdi = -1.
END IF.
EXECUTE.
VAR LAB heoptdi '(D) Ever reported diabetic eye disease (diagnosed)'.
VAL LAB heoptdi
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Macular Degeneration (Diagnosed)

**Variable name:** HEOPTMD

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported macular degeneration and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEOPT1-BHEOPT5, CHEOPT1-CHEOPT2 and INWAVE were used in this derivation.

```

NUMERIC heoptmd (F2.0).
DO IF bheopt1 ~= 3 AND bheopt2 ~= 3 AND bheopt3 ~= 3 AND bheopt4 ~= 3
AND bheopt5 ~= 3 AND cheopt1 ~= 3 AND cheopt2 ~= 3.
COMPUTE heoptmd = 0.
END IF.
EXECUTE.
DO IF bheopt1 = 3 OR bheopt2 = 3 OR bheopt3 = 3 OR bheopt4 = 3 OR
bheopt5 = 3.
COMPUTE heoptmd = 1.
END IF.
EXECUTE.
DO IF SYSMIS(heoptmd) AND (cheopt1 = 3 OR cheopt2 = 3).
COMPUTE heoptmd = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(heoptmd).
COMPUTE heoptmd = -3.
END IF.
EXECUTE.

```

```

DO IF inwave = 2 AND SYSMIS(heoptmd).
COMPUTE heoptmd = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bheopt1 = -1 AND cheopt1 = -1.
COMPUTE heoptmd = -1.
END IF.
EXECUTE.
VAR LAB heoptmd '(D) Ever reported macular degeneration (diagnosed)'.
VAL LAB heoptmd
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Ever Reported Cataracts (Diagnosed)

**Variable name:** HEOPTCA

**Value labels:**

-3 In Wave 1 only and did not report  
-2 In Wave 2 only and did not report  
0 Neither wave  
1 By Wave 1  
2 By Wave 2

**Description of variable:**

This variable shows whether a respondent has ever reported cataracts and if so, by which wave.

**Population:** All respondents

**Level:** Individual

**Derivation:** Datasets for Waves 1 and 2 were merged and the variable names from Wave 1 prefixed with b and those from Wave 2 prefixed with c. The variables BHEOPT1-BHEOPT5, CHEOPT1-CHEOPT2 and INWAVE were used in this derivation.

```

NUMERIC heoptca (F2.0).
DO IF bheopt1 ~= 4 AND bheopt2 ~= 4 AND bheopt3 ~= 4 AND bheopt4 ~= 4
AND bheopt5 ~= 4 AND cheopt1 ~= 4 AND cheopt2 ~= 4.
COMPUTE heoptca = 0.
END IF.
EXECUTE.
DO IF bheopt1 = 4 OR bheopt2 = 4 OR bheopt3 = 4 OR bheopt4 = 4 OR
bheopt5 = 4.
COMPUTE heoptca = 1.
END IF.
EXECUTE.
DO IF SYSMIS(heoptca) AND (cheopt1 = 4 OR cheopt2 = 4).
COMPUTE heoptca = 2.
END IF.
EXECUTE.
DO IF inwave = 1 AND SYSMIS(heoptca).
COMPUTE heoptca = -3.
END IF.
EXECUTE.

```

```

DO IF inwave = 2 AND SYSMIS(heoptca).
COMPUTE heoptca = -2.
END IF.
EXECUTE.
DO IF inwave = 3 AND bheopt1 = -1 AND cheopt1 = -1.
COMPUTE heoptca = -1.
END IF.
EXECUTE.
VAR LAB heoptca '(D) Ever reported cataract (diagnosed)'.
VAL LAB heoptca
-3 'In Wave 1 only and did not report'
-2 'In Wave 2 only and did not report'
0 'Neither wave'
1 'By Wave 1'
2 'By Wave 2'.

```

## Physical Activity Summary

**Variable name:** PALEVEL

**Value labels:**

- 8 Not known
- 1 Not applicable
- 0 Sedentary
- 1 Low
- 2 Moderate
- 3 High

**Description of variable:**

This variable summarises the answers to the level of work activity (WPJACT) in the work and pensions section and three questions on physical activity “in daily life” in the health section (HEACTA - HEACTC). It approximates as closely as possible the classification used in the Allied Dunbar Survey of Fitness.

Reference

Activity and Health Research (1992) *Allied Dunbar National Fitness Survey: main findings*, London: Sports Council and Health Education Authority.

Levels used in this variable are defined as follows

**Sedentary:** Not working or sedentary occupation, engages in mild exercise 1–3 times a month or less, with no moderate or vigorous activity.

**Low:** Standing occupation, engages in moderate leisure-time exercise once a week or less and no vigorous activity; OR engages in mild leisure-time activity at least 1–3 times a month, moderate once a week or less and no vigorous; OR has a sedentary or no occupation and engages in moderate leisure-time activity once a week or 1–3 times a month, with no vigorous activity.

**Moderate:** Does physical work; OR engages in moderate leisure-time activity more than once a week; OR engages in vigorous activity once a week to 1–3 times a month.

**High:** Heavy manual work or vigorous leisure activity more than once a week.

**Population:** All respondents

**Level:** Individual

## Derivation:

```
NUMERIC palevel (F2.0).
DO IF ((wpjact = -1 OR wpjact = 1) AND (heactc = 3 OR heactc = 4) AND
heactb = 4 AND heacta = 4).
COMPUTE palevel = 0.
END IF.
EXECUTE.
DO IF (wpjact = 2 AND (heactb = 2 OR heactb = 3 OR heactb = 4) AND
heacta = 4) OR ((heactc > 0 AND heactc < 4) AND (heactb > 1 AND heactb
<= 4) AND heacta = 4) OR ((wpjact = -1 OR wpjact = 1) AND (heactb = 2
OR heactb = 3) AND heacta = 4).
COMPUTE palevel = 1.
END IF.
EXECUTE.
DO IF (wpjact = 3 OR heactb = 1 OR (heacta = 2 OR heacta = 3)).
COMPUTE palevel = 2.
END IF.
EXECUTE.
DO IF wpjact = 4 OR heacta = 1.
COMPUTE palevel = 3.
END IF.
EXECUTE.
DO IF wpjact = -9 OR wpjact = -8 OR heacta = -9 OR heacta = -8 OR
heactb = -9 OR heactb = -8 OR heactc = -9 OR heactc = -8.
COMPUTE palevel = -8.
END IF.
EXECUTE.
DO IF heacta = -1.
COMPUTE palevel = -1.
END IF.
EXECUTE.
VAR LAB palevel '(D) Physical activity summary'.
VAL LAB palevel
-8 'Not known'
-1 'Not applicable'
0 'Sedentary'
1 'Low'
2 'Moderate'
3 'High'.
```

## Maximum Alcohol Units in a Day Last Week

**Variable name:** MAXALC

### Value labels:

- 8 Not known
- 1 Not applicable
- 5 No drinks last week
- 4 No drinks last year

### Description of variable:

This variable gives the number of alcohol units consumed on the heaviest day's drinking in the week prior to interview. Respondents were first asked how often they drank alcohol in the previous 12 months, then the frequency in the previous week, then the amounts of different types of drink on the heaviest day in that week. The answers for specific drinks are converted as follows:

- 1 pint normal beer = 2 units
- 1 large can or bottle normal beer = 1.5 units
- 1 small can or bottle normal beer = 1 unit
- 1 pint strong beer = 3 units
- 1 large can or bottle strong beer = 2.25 units
- 1 small can or bottle normal beer = 1.5 units
- 1 glass spirits = 1 unit
- 1 glass sherry = 1 unit
- 1 glass wine = 1 unit
- 1 small can or bottle alcopops = 1 unit

Respondents were asked to count a double of spirits as two drinks.

Note: High values may be mis-recorded and reflect weekly rather than daily consumption; these have not been removed as there was no direct means of verifying this.

This variable is set to 'Not known' if any of the quantity variables are blank.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

COMPUTE maxalc = 0.
DO IF scabnp >= 0.
COMPUTE maxalc = maxalc + scabnp*2.
END IF.
DO IF scabnlc >= 0.
COMPUTE maxalc = maxalc + scabnlc*1.5.
END IF.
DO IF scabnsc >= 0.
COMPUTE maxalc = maxalc + scabnsc.
END IF.
DO IF scabsp >= 0.
COMPUTE maxalc = maxalc + scabsp*3.
END IF.
DO IF scabslc >= 0.
COMPUTE maxalc = maxalc + scabslc*2.25.
END IF.
DO IF scabssc >= 0.
COMPUTE maxalc = maxalc + scabssc*1.5.
END IF.
DO IF scaspir >= 0.
COMPUTE maxalc = maxalc + scaspir.
END IF.
DO IF scasher >= 0.
COMPUTE maxalc = maxalc + scasher.
END IF.
DO IF scawin >= 0.
COMPUTE maxalc = maxalc + scawin.
END IF.
DO IF scapopg >= 0.
COMPUTE maxalc = maxalc + scapopg.
END IF.

```

```

*Temporary variables to capture the amount of missing information.
DO IF ANY(-9,scdri01,scdri02,scdri03,scdri04,scdri05,scdri06).
COMPUTE maxdk = 1.
END IF.
DO IF ANY(-
9,scabnp,scabnlc,scabnsc,scabsp,scabslc,scabssc,scaspir,scasher,scawin
,scapopg).
COMPUTE maxdk2 = 1.
END IF.
DO IF maxdk = 1 OR maxdk2 = 1.
COMPUTE maxalc = -8.
END IF.
DO IF scako = -1.
COMPUTE maxalc = -1.
END IF.
DO IF scako = -9 OR scal7a = -9 OR scal7b = -9.
COMPUTE maxalc = -8.
END IF.
DO IF scako = 8.
COMPUTE maxalc = -5.
END IF.
DO IF maxalc = 0 AND scal7a = 2.
COMPUTE maxalc = -4.
END IF.
EXECUTE.
VAR LAB maxalc '(D) Maximum alcohol units in a day last week'.
VAL LAB maxalc
-8 'Not known'
-1 'Not applicable'
-5 'No drinks last year'
-4 'No drinks last week'.

```

## Maximum Alcohol Units in a Day Last Week

**Variable name:** MAXALC2

**Value labels:**

- 8 Not known
- 1 Not applicable
- 5 No drinks last week
- 4 No drinks last year

**Description of variable:**

This variable gives the number of alcohol units consumed on the heaviest day's drinking in the week prior to interview. Respondents were first asked how often they drank alcohol in the previous 12 months, then the frequency in the previous week, then the amounts of different types of drink on the heaviest day in that week. The answers for specific drinks are converted as follows:

- 1 pint normal beer = 2 units
- 1 large can or bottle normal beer = 1.5 units
- 1 small can or bottle normal beer = 1 unit
- 1 pint strong beer = 3 units
- 1 large can or bottle strong beer = 2.25 units
- 1 small can or bottle normal beer = 1.5 units
- 1 glass spirits = 1 unit
- 1 glass sherry = 1 unit

1 glass wine = 1 unit  
1 small can or bottle alcopops = 1 unit

Respondents were asked to count a double of spirits as two drinks.

Note: High values may be mis-recorded and reflect weekly rather than daily consumption; these have not been removed as there was no direct means of verifying this.

This variable assigns a quantity if a participant reported drinking beer but then gave information on only one or two of the three quantity variables (pints, large cans, small cans), e.g. it assumes that they only drank pints if they reported a quantity for this and left quantities for large and small cans blank.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
COMPUTE maxalc2 = 0.
DO IF scabnp >= 0.
COMPUTE maxalc2 = maxalc2 + scabnp*2.
END IF.
DO IF scabnlc >= 0.
COMPUTE maxalc2 = maxalc2 + scabnlc*1.5.
END IF.
DO IF scabnsc >= 0.
COMPUTE maxalc2 = maxalc2 + scabnsc.
END IF.
DO IF scabsp >= 0.
COMPUTE maxalc2 = maxalc2 + scabsp*3.
END IF.
DO IF scabslc >= 0.
COMPUTE maxalc2 = maxalc2 + scabslc*2.25.
END IF.
DO IF scabssc >= 0.
COMPUTE maxalc2 = maxalc2 + scabssc*1.5.
END IF.
DO IF scaspir >= 0.
COMPUTE maxalc2 = maxalc2 + scaspir.
END IF.
DO IF scasher >= 0.
COMPUTE maxalc2 = maxalc2 + scasher.
END IF.
DO IF scawin >= 0.
COMPUTE maxalc2 = maxalc2 + scawin.
END IF.
DO IF scapopg >= 0.
COMPUTE maxalc2 = maxalc2 + scapopg.
END IF.
*Temporary variables to capture the amount of missing information.
DO IF ANY(-9,scdri01,scdri02,scdri03,scdri04,scdri05,scdri06).
COMPUTE maxdk3 = 1.
END IF.
DO IF (scabnp = -9 AND scabnlc = -9 AND scabnsc = -9) OR (scabsp = -9
AND scabslc = -9 AND scabssc = -9) OR scaspir = -9 OR scasher = -9 OR
scawin = -9 OR scapopg = -9.
COMPUTE maxdk4 = 1.
END IF.
```

```

DO IF maxdk3 = 1 OR maxdk4 = 1.
COMPUTE maxalc2 = -8.
END IF.
DO IF scako = -1.
COMPUTE maxalc2 = -1.
END IF.
DO IF scako = -9 OR scal7a = -9 OR scal7b = -9.
COMPUTE maxalc2 = -8.
END IF.
DO IF scako = 8.
COMPUTE maxalc2 = -5.
END IF.
DO IF maxalc2 = 0 AND scal7a = 2.
COMPUTE maxalc2 = -4.
END IF.
EXECUTE.
VAR LAB maxalc2 '(D) Maximum alcohol units in a day last week
(assuming quantity of beer complete if at least one amount given)'.
VAL LAB maxalc2
-8 'Not known'
-1 'Not applicable'
-5 'No drinks last year'
-4 'No drinks last week'.

```

## Edinburgh Claudication

**Variable name:** CLAUD

**Value labels:**

- 8 Not known
- 2 Cannot/never walk
- 1 Not applicable
- 0 None
- 1 Grade 1
- 2 Grade 2

**Description of variable:**

This variable gives the summary value for the Edinburgh claudication scale, which relies on report of symptoms. To qualify as having claudication a respondent has to i) experience pain or discomfort on walking ii) not get it when standing still or sitting iv) the pain disappears in 10 minutes or less when stop v) experience the pain in the calf. Grade 1 applies if experiences the pain when walking uphill or when in a hurry; grade 2 applies if experiences the pain when walking at an ordinary pace on level ground.

Reference

Leng GC, Fowkes FG. The Edinburgh Claudication Questionnaire: an improved version of the WHO/Rose Questionnaire for use in epidemiological surveys. *J Clin Epidemiol* 1992; 45(10):1101-9.

**Population:** All respondents

**Level:** Individual

### Derivation:

```
*Temporary variable indicating whether pain on calf or not.
DO IF ANY(1,hecdf1,hecdf2,hecdf3).
COMPUTE calfm = 1.
END IF.
*Temporary variables indicating whether respondent gets pain going
uphill or on level ground.
DO IF hecdc = 1 AND hecdb = 2.
COMPUTE cuphill = 1.
END IF.
DO IF hecdd = 1 AND hecdb = 2.
COMPUTE clevel = 1.
END IF.
NUMERIC claud (F2.0).
DO IF cuphill = 1 AND hecde = 2 AND calfm = 1.
COMPUTE claud = 1.
END IF.
DO IF clevel = 1 AND hecde = 2 AND calfm = 1.
COMPUTE claud = 2.
END IF.
EXECUTE.
DO IF SYSMIS(claud) AND (hecda = 2 OR (hecdc = 2 AND hecdd = 2) OR
hecdb = 1 OR hecde = 1 OR (SYSMIS(calfm) AND hecdf1 > 0)).
COMPUTE claud = 0.
END IF.
DO IF hecda = -1.
COMPUTE claud = -1.
END IF.
EXECUTE.
DO IF SYSMIS(claud) AND (hecda = 3 OR hecdc = 3 OR hecdd = 3).
COMPUTE claud = -2.
END IF.
EXECUTE.
DO IF SYSMIS(claud) AND ((hecda = -8 OR hecda = -9) OR (hecdb = -8 OR
hecdb = -9) OR (hecdc = -8 OR hecdc = -9) OR (hecdd = -8 OR hecdd = -
9) OR (hecde = -8 OR hecde = -9) OR (hecdf1 = -8 OR hecdf1 = -9)).
COMPUTE claud = -8.
END IF.
EXECUTE.
VAR LAB claud '(D) Edinburgh claudication'.
VAL LAB claud
0 'None'
1 'Grade 1'
2 'Grade 2'
-1 'Not applicable'
-2 'Cannot/never walk'
-8 'Not known'.
```

## Breathlessness, MRC Respiratory Questionnaire

**Variable name:** BRETHS

### Value labels:

-9 Refused  
-8 Don't know  
-2 Cannot/never walk  
-1 Not applicable  
0 None

- 1 Grade 1
- 2 Grade 2
- 3 Grade 3

**Description of variable:**

This variable gives the summary value for the MRC respiratory questionnaire, which relies on report of symptoms of breathlessness or dyspnoea. To qualify as being breathless a respondent has to experience being out of breath on exertion. Grade 1 applies if experiences the breathlessness when walking up a slight hill or when in hurrying on the level; Grade 2 applies if experiences the breathlessness when walking at own pace on level ground; Grade 3 applies if has to stop when feeling breathlessness while walking at own pace on the level. Grade 3 coding takes priority over the other codes. Code -2 is only used if have not been coded as Grade 2 or Grade 3 and answers “never” or “cannot” walk at one of the other questions.

The dyspnoea aspects of the MRC respiratory questionnaire are recommended by the National Institute for Clinical Excellence for use as one element in diagnosing Chronic Obstructive Airways Disease.

Reference

National Institute for Clinical Excellence; Chronic obstructive pulmonary disease. Management of chronic obstructive pulmonary disease in adults in primary and secondary care Clinical Guideline 12. February 2004. Page 9.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC breths (F2.0).
DO IF herpe = 1.
  COMPUTE breths = 2.
END IF.
EXECUTE.
DO IF herpf = 1.
  COMPUTE breths = 3.
END IF.
EXECUTE.
DO IF SYSMIS(breths) AND (herpd = 3 OR herpd = 4 OR herpe = 3 OR herpe = 4).
  COMPUTE breths = -2.
END IF.
EXECUTE.
DO IF SYSMIS(breths) AND herpd = 1.
  COMPUTE breths = 1.
END IF.
EXECUTE.
DO IF SYSMIS(breths) AND (herpd = 2 OR (herpe = 2 AND herpf = 2)).
  COMPUTE breths = 0.
END IF.
EXECUTE.
DO IF herpd < 0.
  COMPUTE breths = herpd.
END IF.
EXECUTE.
```

```

VAR LAB breths '(D) Breathlessness, MRC respiratory questionnaire'.
VAL LAB breths
-1 "Not applicable"
-8 "Don't know"
-9 "Refused"
0 "None"
1 "Grade 1"
2 "Grade 2"
3 "Grade 3"
-2 "Cannot/never walk".

```

## Phlegm, MRC Respiratory Questionnaire

**Variable name:** PHLEGM

**Value labels:**

-8 Not known  
-1 Not applicable  
0 None or not regular  
1 First thing in winter  
2 Other

**Description of variable:**

This variable gives the summary value for the MRC respiratory questionnaire, which relies on report of symptoms of phlegm or sputum production. Code 1 includes those who usually bring up phlegm from the chest first thing in winter and bring up phlegm on most days for as much as 3 months in the year. Code 2 applies to those who answered 'no' to bringing up phlegm first thing but yes to bringing up phlegm from the chest during day or night in winter and say they bring up phlegm most days for as much as 3 months in the year. Code 0 includes those who satisfy only one of the two criteria needed for Code 1 or for Code 2.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC phlegm (F2.0).
DO IF herpa = 1 AND herpc = 1.
COMPUTE phlegm = 1.
END IF.
EXECUTE.
DO IF SYSMIS(phlegm) AND herpb = 1 AND herpc = 1.
COMPUTE phlegm = 2.
END IF.
EXECUTE.
DO IF SYSMIS(phlegm) AND ((herpa = 2 AND herpb = 2) OR herpc = 2).
COMPUTE phlegm = 0.
END IF.
EXECUTE.
DO IF herpa < -1 OR herpb < -1 OR herpc < -1 OR herpa = 3.
COMPUTE phlegm = -8.
END IF.
EXECUTE.

```

```

DO IF herpa = -1.
COMPUTE phlegm = -1.
END IF.
EXECUTE.
VAR LAB phlegm '(D) Phlegm, MRC respiratory questionnaire'.
VAL LAB phlegm
0 'None or not regular'
1 'First thing in winter'
2 'Other'
-8 'Not known'
-1 'Not applicable'.

```

## 4 Cognitive Function

### Response to Prospective Memory Test

**Variable name:** CFMERSP

**Value labels:**

- 1 Correct response without prompt
- 2 Incorrect response
- 1 Not applicable

**Description of variable:**

This variable gives the number of those who failed to make the correct response without prompt. Only those who did the test without being prompted are the only successes.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC cfmersp (F2.0).
IF (cfmems = 1 & cfmem = 1) cfmersp = 1 .
EXECUTE .
IF (cfmems = 2 | cfmems=3 | cfmems=4 | cfmems = 5 | (cfmems = 1 &
cfmem = 2) ) cfmersp = 2 .
EXECUTE .
IF (cfmem = - 9 | cfmem = - 8 | cfmem = - 1) cfmersp = cfmem.
EXECUTE .
IF (SYSMIS(cfmersp) AND (cfmems = -1 | cfmems = -8 | cfmems = -9))
cfmersp = cfmems.
EXECUTE.
VAR LAB cfmersp '(D) Response to prospective memory test'.
VAL LAB cfmersp
1 "Correct response without prompt"
2 "Incorrect response"
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal".

```

## Prospective Memory Score

**Variable name:** CFPRMEM

**Value labels:**

- 1 Not applicable
- 0 'Prompt given or did nothing'
- 1 'Did something else without prompt'
- 2 'Wrote initials elsewhere or wrote something else without prompt'
- 3 'Correct response without prompt'

**Description of variable:**

This variable gives a score from 0 to 3 for the prospective memory test. The value 3 corresponds to the correct answer to the test.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC cfprmem (F2.0).
IF (cfmems = 5 | cfmem = 2) cfprmem = 0.
EXECUTE.
IF (cfmems = 1 & cfmem = 1) cfprmem = 3 .
EXECUTE .
IF (cfmems = 2 & cfmem = 1) cfprmem = 2 .
EXECUTE .
IF (cfmems = 3 & cfmem = 1) cfprmem = 2 .
EXECUTE .
IF (cfmems = 4 & cfmem = 1) cfprmem = 1 .
EXECUTE .
IF (cfmems = -1) cfprmem = -1 .
EXECUTE.
VARIABLE LABELS cfprmem '(D) Prospective memory score'.
VALUE LABELS cfprmem
3 'Correct response without prompt'
2 'Wrote initials elsewhere or wrote something else without prompt'
1 'Did something else without prompt'
0 'Prompt given or did nothing'
-1 "Not applicable".
```

## Literacy Score

**Variable name:** CFLITSC

**Value labels:**

- 1 Not applicable, includes Refusals and Don't Know's

**Description of variable:**

This variable gives the combined literacy score for a respondent if applicable.

This variable was present on the archived Wave 2 Core Data, but its derivation had not been documented before and was in fact incorrect, so the variable has been re-derived and included here.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
COMPUTE cflitsc = -1.
RECODE cflitb (1=1) (-1=-1) (ELSE=0) INTO cflitbn .
RECODE cflitc (1=1) (-1=-1) (ELSE=0) INTO cflitcn .
RECODE cflitd (1=1) (-1=-1) (ELSE=0) INTO cflitdn .
RECODE cflite (1=1) (-1=-1) (ELSE=0) INTO cfliten .
EXECUTE.
IF (cflitbn > - 1) cflitsc = cflitbn + cflitcn + cflitdn+cfliten .
VARIABLE LABELS cflitsc '(D) Literacy score'.
VALUE LABELS cflitsc -1 "Not applicable, includes Refusals and Don't Know's".
```

## Delayed Word Recall as % of Immediate Word Recall

**Variable name:** CFRECAL

**Value labels:**

- 9 Refusal
- 8 Don't know
- 7 Missing delayed word recall or zero immediate word recall
- 1 Not applicable

**Description of variable:**

This variable gives the delayed recall as % of immediate recall. Note that for a relatively small number of cases, this figure is more than 100%.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
IF (cflisen=-9) cfrecal=-9.
IF (cflisen=-8) cfrecal=-8.
IF (cflisen=-1 & cflisd=-1) cfrecal=-1.
IF (cflisen=0) cfrecal=-7.
IF (cflisen>0 & (cflisd=-9 | cflisd=-1 | cflisd=-8)) cfrecal=-7.
EXECUTE.
DO IF (cflisen>0 & cflisd >= 0).
COMPUTE cfrecal=(cflisd*100)/cflisen.
END IF.
EXECUTE.
VAR LAB cfrecal "(D) Delayed word recall as % of immediate recall".
VAL LAB cfrecal
-8 "Don't know"
-9 "Refusal"
-1 "Not applicable"
-7 "Missing delayed word recall or zero immediate word recall".
```

## % Accuracy of Letter Search

**Variable name:** CFACCUR

**Value labels:**

-1 Not applicable

**Description of variable:**

This variable gives the letters correctly searched as a % of total letters searched.

Note that the variables NNCORRE and NNMISSSE were named NCORREC and NMISSED in Wave 1 respectively.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
DO IF (nncorre > 0) .
COMPUTE cfaccur=(nncorre*100)/(nncorre+nnmisse) .
ELSE IF (nncorre=0) .
COMPUTE cfaccur=0.
ELSE IF nncorre = -1.
COMPUTE cfaccur = -1.
END IF.
EXECUTE.
VAR LAB cfaccur "(D) % Accuracy of letter search".
VALUE LABELS cfaccur
  -1 "Not applicable".
EXECUTE.
```

## Memory Function Index

**Variable name:** CFMEIND

**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable is a Memory Function Index, which gives the total score of all the memory tests.

Please note that the second prospective memory test was not completed at Wave 2.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC cfmeind (F2.0).
COMPUTE cfmeind=-1.
EXECUTE.
DO IF cfdscr>=0 & cflisen>=0 & cflisd>=0 & cfprmem>=0.
COMPUTE cfmeind = cfdscr + cflisen + cflisd + cfprmem.
END IF.
EXECUTE .
VARIABLE LABELS cfmeind '(D) Memory Function Index'.
VALUE LABELS cfmeind -1 "Not applicable (including Refusals and Don't Knows)".
```

## Number of Letters Missed Recoded for Executive Function Index

**Variable name:** NNMISRC

**Value labels:**

- 1 Not applicable
- 1 9+ missed
- 2 6-8 missed
- 3 4-5 missed
- 4 3 missed
- 5 2 missed
- 6 0-1 missed

**Description of variable:**

This new variable is NNMISSE recoded into groups to give a score from 1 to 6 so that it can be used to create the Executive Function Index.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC nnmisrc (F2.0).
RECODE nnmisse
  (-1=-1) (0 thru 1=6) (2=5) (3=4) (4 thru 5=3) (6 thru 8=2) (9
thru Highest=1) INTO nnmisrc .
EXECUTE .
VARIABLE LABELS nnmisrc '(D) Number of letters missed recoded for
Executive Function Index'.
VALUE LABELS nnmisrc
  1 "9+ missed"
  2 "6-8 missed"
  3 "4-5 missed"
  4 "3 missed"
  5 "2 missed"
  6 "0-1 missed"
  -1 "Not applicable".
EXECUTE .
```

## Number of Items Completed Recoded for Executive Function Index

**Variable name:** CFSPEED

**Value labels:**

- 1 Not applicable
- 1 1-174 items completed
- 2 175-249 items completed
- 3 250-274 items completed
- 4 275-299 items completed
- 5 300-324 items completed
- 6 325-399 items completed
- 7 400-449 items completed
- 8 450+ items completed

**Description of variable:**

This new variable is NROWCL recoded into groups to give a score from 1 to 8 so that it can be used to create the Executive Function Index.

Note that the variables NROWCL was named NROWCLM in Wave 1.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC cfspeed (F2.0).
RECODE nrowcl
(-1=-1) (1 thru 174=1) (175 thru 249=2) (250 thru 274=3) (275 thru
299=4) (300 thru 324=5) (325 thru 399=6) (400 thru 449=7) (450 thru
Highest=8) INTO cfspeed.
EXECUTE .
VARIABLE LABELS cfspeed '(D) Number of items completed recoded for
Executive Function Index'.
VALUE LABELS cfspeed
  1 "1-174 items completed"
  2 "175-249 items completed"
  3 "250-274 items completed"
  4 "275-299 items completed"
  5 "300-324 items completed"
  6 "325-399 items completed"
  7 "400-449 items completed"
  8 "450+ items completed"
-1 "Not applicable".
EXECUTE .

```

## Fluency Recoded for Executive Function Index

**Variable name:** CFANIG

**Value labels:**

-9 Refusal  
-1 Not applicable  
0 0 animals  
1 1-7 animals  
2 8-12 animals  
3 13-15 animals  
4 16-17 animals  
5 18-19 animals  
6 20-21 animals  
7 22-24 animals  
8 25-29 animals  
9 30+ animals

**Description of variable:**

This new variable is CFANI recoded into groups to give a score from 1 to 8 so that it can be used to create the Executive Function Index.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC cfanig (F2.0).
RECODE cfani
  (0=0) (-1=-1) (-8=0) (-9=-9) (1 thru 7=1) (8 thru 12=2) (13
thru 15=3) (16 thru 17=4) (18 thru 19=5) (20 thru 21=6) (22 thru
24=7) (25 thru 29=8) (30 thru Highest=9) INTO cfanig .
EXECUTE .
VARIABLE LABELS cfanig '(D) Fluency recoded for Executive Function
Index' .
VALUE LABELS cfanig
  0 "0 animals"
  1 "1-7 animals"
  2 "8-12 animals"
  3 "13-15 animals"
  4 "16-17 animals"
  5 "18-19 animals"
  6 "20-21 animals"
  7 "22-24 animals"
  8 "25-29 animals"
  9 "30+ animals"
  -1 "Not applicable"
  -9 "Refusal".
EXECUTE.

```

## Executive Function Index (Cognitive)

**Variable name:** CFEXIND

**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable is the Executive Function Index, which gives the total score for the verbal fluency and letter cancellation tests.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC cfexind (F2.0).
COMPUTE cfexind=-1.
EXECUTE.
DO IF cfanig >= 0 AND cfspeed >= 0 AND nnmisrc >= 0.
COMPUTE cfexind = cfanig + cfspeed+ nnmisrc.
END IF.
EXECUTE .
VARIABLE LABELS cfexind '(D) Executive Function Index (cognitive)'.
VALUE LABELS cfexind
  -1 "Not applicable (including Refusals and Don't know's)".
EXECUTE.

```

## Total Cognitive Index (Memory + Executive)

**Variable name:** CFIND

**Value labels:**

-1 Not applicable

**Description of variable:**

This variable gives the total cognitive score, which is the sum of the Memory and Executive indices.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC cfind (F2.0).
COMPUTE cfind=-1.
EXECUTE.
DO IF cfmeind >= 0 AND cfexind > 0.
COMPUTE cfind = cfmeind + cfexind.
END IF.
EXECUTE .
VARIABLE LABELS cfind '(D) Total Cognitive Index (Memory +
Executive)'.
VALUE LABELS cfind
  -1 "Not applicable".
EXECUTE.
```

## 5 Physical Function

### Mean Gait Speed (m/s)

**Variable name:** GTSPD\_AV

**Value labels:**

-98 "Under 60 years of age"  
-97 "Proxy interview"  
-96 "Not applicable throughout"  
-95 "Participant refused at some stage"  
-94 "Participant gave 'don't know' response at some stage"  
-93 "Unable to walk alone"  
-92 "Able to walk alone but no aid available"  
-91 "Health restriction"  
-90 "Unwilling to do test"  
-89 "Interviewer refused to answer"  
-88 "Interviewer gave DK response"  
-87 "Interviewer felt not safe"  
-86 "No available space"  
-85 "Participant felt unsafe"  
-84 "Walk A not completed or stopped"  
-83 "No speed - both times extreme"  
-82 "No speed - A extreme, B not completed or refused"

**Description of variable:**

This variable gives the mean gait speed in metres per second for each respondent, with extensive missing values for the respondents who did not complete the timed walk.

**Population:** All respondents**Level:** Individual**Derivation:**

```
NUMERIC gtspd_av (F8.4).
DO IF indager < 60.
COMPUTE gtspd_av = -98.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND (w2indout = 13 OR w2indout = 23).
COMPUTE gtspd_av = -97.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND mmschs = -1 AND mmalone = -1 AND mmhss = -1 AND mmwill = -1 AND mmsaf = -1 AND mmavsp = -1 AND mmwala = -1 AND mmtrya = -1 AND mmwlka = -1 AND mmtryb = -1 AND mmwlkb = -1).
COMPUTE gtspd_av = -96.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmschs=-9 OR mmalone = -9 OR mmhss = -9 OR mmwill = -9 OR mmwala = -9 OR mmtrya=-9 OR mmtrya=4 OR mmwlka=-9)).
COMPUTE gtspd_av = -95.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmalone = -8 OR mmwala=-8 OR mmhss=-8 OR mmwill=-8)).
COMPUTE gtspd_av = -94.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmalone = 3.
COMPUTE gtspd_av = -93.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmalone = 2.
COMPUTE gtspd_av = -92.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND (mmhss = 2 OR mmhss = 3 OR mmhss = 4).
COMPUTE gtspd_av = -91.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmwill = 2.
COMPUTE gtspd_av = -90.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmsaf = -9 OR mmavsp=-9)).
COMPUTE gtspd_av = -89.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmsaf = -8 OR mmavsp=-8)).
COMPUTE gtspd_av = -88.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmsaf = 2.
COMPUTE gtspd_av = -87.
END IF.
EXECUTE.
```

```

DO IF SYSMIS(gtspd_av) AND mmavsp = 2.
COMPUTE gtspd_av = -86.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmwala = 2.
COMPUTE gtspd_av = -85.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmtrya = 2 OR mmtrya=3)).
COMPUTE gtspd_av = -84.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmwlka >= 30 OR (mmwlka>0 AND mmwlka <=
0.54)) AND (mmwlkb >= 30 OR (mmwlkb>0 AND mmwlkb <= 0.54))).
COMPUTE gtspd_av = -83.
END IF.
EXECUTE.
DO IF (SYSMIS(gtspd_av) AND (mmwlka >= 30 OR mmwlka <= 0.54) AND
mmwlkb < 0).
COMPUTE gtspd_av = -82.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmwlka < 30 AND mmwlka > 0.54 AND mmwlkb <
30 AND mmwlkb > 0.54.
COMPUTE gtspd_av = ((2.4384/mmwlka) + (2.4384/mmwlkb))/2.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmwlka < 30 AND mmwlka > 0.54.
COMPUTE gtspd_av = 2.4384/mmwlka.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_av) AND mmwlkb < 30 AND mmwlkb > 0.54.
COMPUTE gtspd_av = 2.4384/mmwlkb.
END IF.
EXECUTE.
VAR LAB gtspd_av "(D) Mean gait speed (m/s)".
VAL LAB gtspd_av
-98 "Under 60 years of age"
-97 "Proxy interview"
-96 "Not applicable throughout"
-95 "Participant refused at some stage"
-94 "Participant gave 'don't know' response at some stage"
-93 "Unable to walk alone"
-92 "Able to walk alone but no aid available"
-91 "Health restriction"
-90 "Unwilling to do test"
-89 "Interviewer refused to answer"
-88 "Interviewer gave DK response"
-87 "Interviewer felt not safe"
-86 "No available space"
-85 "Participant felt unsafe"
-84 "Walk A not completed or stopped"
-83 "No speed - both times extreme"
-82 "No speed - A extreme, B not completed or refused".

```

## Walks Used to Derive Mean Gait Speed

**Variable name:** GTSPD\_ID

**Value labels:**

- 1 Not applicable – see GTSPD\_AV
- 1 Only Walk A - B was extreme, not completed, refused or don't know
- 2 Only Walk B – A was extreme
- 3 From both walks

**Description of variable:**

This variable states whether times from either Walk A, Walk B or both were used to calculate the average walk speed.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC gtspd_id (F2.0).
DO IF gtspd_av > 0 AND mmwlka < 30 AND mmwlka > 0.54 AND mmwlkb < 30
AND mmwlkb > 0.54.
COMPUTE gtspd_id = 3.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_id) AND gtspd_av > 0 AND mmwlka < 30 AND mmwlka >
0.54.
COMPUTE gtspd_id = 1.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_id) AND gtspd_av > 0 AND mmwlkb < 30 AND mmwlkb >
0.54.
COMPUTE gtspd_id = 2.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_id).
COMPUTE gtspd_id = -1.
END IF.
EXECUTE.
VAR LAB gtspd_id "(D) Walks used to derive mean gait speed".
VAL LAB gtspd_id
-1 "Not applicable - see GTSPD_AV"
1 "Only Walk A - B was extreme, not completed, refused or don't know"
2 "Only Walk B - A was extreme"
3 "From both walks".
```

## Mean Gait Speed (Grouped)

**Variable name:** GTSPD\_AG

**Value labels:**

- 5 Unavailable due to other circumstances
- 4 Unable due to physical limitation
- 3 Participant unwilling/refused
- 2 Missing - unknown reason
- 1 Not eligible
- 1 Speed <= 0.4 m/s

2 Speed > 0.4 m/s

**Description of variable:**

This variable shows whether the mean gait speed was greater than 0.4 metres per second or not.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC gtspd_ag (F2.0).
DO IF gtspd_av > 0 AND gtspd_av <= 0.4.
COMPUTE gtspd_ag = 1.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_ag) AND gtspd_av > 0.4.
COMPUTE gtspd_ag = 2.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_ag) AND gtspd_av <= -97.
COMPUTE gtspd_ag = -1.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_ag) AND (gtspd_av = -96 OR gtspd_av = -94 OR
gtspd_av = -89 OR gtspd_av = -88).
COMPUTE gtspd_ag = -2.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_ag) AND (gtspd_av = -95 OR gtspd_av = -90).
COMPUTE gtspd_ag = -3.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_ag) = 1 AND (gtspd_av = -93 OR gtspd_av = -84).
COMPUTE gtspd_ag = -4.
END IF.
EXECUTE.
DO IF SYSMIS(gtspd_ag) = 1 AND (gtspd_av = -92 OR gtspd_av = -86 OR
gtspd_av = -83 OR gtspd_av = -82 OR gtspd_av = -91 OR gtspd_av = -87
OR gtspd_av = -81 OR gtspd_av = -85).
COMPUTE gtspd_ag = -5.
END IF.
EXECUTE.
VAR LAB gtspd_ag "(D) Mean gait speed (grouped)".
VAL LAB gtspd_ag
-5 "Unavailable due to other circumstances"
-4 "Unable due to physical limitation"
-3 "Participant unwilling/refused"
-2 "Missing - unknown reason"
-1 "Not eligible"
1 "Speed <= 0.4 m/s"
2 "Speed > 0.4 m/s".
```

## Fastest Gait Speed (m/s)

**Variable name:** FST\_TM

**Value labels:**

- 98 "Under 60 years of age"
- 97 "Proxy interview"
- 96 "Not applicable throughout"

- 95 "Participant refused at some stage"
- 94 "Participant gave 'don't know' response at some stage"
- 93 "Unable to walk alone"
- 92 "Able to walk alone but no aid available"
- 91 "Health restriction"
- 90 "Unwilling to do test"
- 89 "Interviewer refused to answer"
- 88 "Interviewer gave DK response"
- 87 "Interviewer felt not safe"
- 86 "No available space"
- 85 "Participant felt unsafe"
- 84 "Walk A not completed or stopped"
- 83 "No speed - both times extreme"
- 82 "No speed - A extreme, B not completed or refused"

**Description of variable:**

This variable gives the faster gait speed of the 2 walks.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC fst_tm (F5.2).
COMPUTE fst_tm = gtspd_av.
DO IF gtspd_id = 3 AND mmwlka < mmwlkb.
COMPUTE fst_tm = mmwlka.
END IF.
EXECUTE.
DO IF gtspd_id = 3 AND mmwlka > mmwlkb.
COMPUTE fst_tm = mmwlkb.
END IF.
EXECUTE.
DO IF gtspd_id = 3 AND mmwlka = mmwlkb.
COMPUTE fst_tm = mmwlkb.
END IF.
EXECUTE.
IF gtspd_id = 1 fst_tm = mmwlka.
IF gtspd_id = 2 fst_tm = mmwlkb.
EXECUTE.
VAR LAB fst_tm '(D) Fastest gait speed (m/s)'.
VAL LAB fst_tm
-98 "Under 60 years of age"
-97 "Proxy interview"
-96 "Not applicable throughout"
-95 "Participant refused at some stage"
-94 "Participant gave 'don't know' response at some stage"
-93 "Unable to walk alone"
-92 "Able to walk alone but no aid available"
-91 "Health restriction"
-90 "Unwilling to do test"
-89 "Interviewer refused to answer"
-88 "Interviewer gave DK response"
-87 "Interviewer felt not safe"
-86 "No available space"
-85 "Participant felt unsafe"
-84 "Walk A not completed or stopped"
-83 "No speed - both times extreme"
-82 "No speed - A extreme, B not completed or refused".

```

## 6 CASP 19

### CASP 19

**Variable name:** CASP19

**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable gives the CASP 19 score for each respondent. Those who have a value of -1 did not have a valid response to every component. This applies to all of the CASP variables.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
NUMERIC CASP19 (F2.0).
RECODE scqola scqolb scqolc scqold scqole scqolf scqolg scqolh scqoli
scqolj scqolk scqoll scqolm scqoln scqolo scqolp scqolq scqolr scqols
(1=3) (2=2) (3=1) (4=0) (-9 thru -1 = sysmis)
INTO qolarc qolbrc qolcrc qoldrc qolerc qolfrc qolgrc qolhrc qolirc
qoljrc qolkrc qollrc qolmrc qolnrc qolorc qolprc qolqrc qolrrc qolsrc.
EXECUTE.
RECODE qolarc qolbrc qoldrc qolfrc qolhrc qolirc (3=0) (2=1) (1=2)
(0=3).
EXECUTE.
COMPUTE CASP19 = qolarc + qolbrc + qolcrc + qoldrc + qolerc + qolfrc
+ qolgrc + qolhrc + qolirc + qoljrc + qolkrc + qollrc + qolmrc +
qolnrc + qolorc + qolprc + qolqrc + qolrrc + qolsrc.
EXECUTE.
IF SYSMIS(CASP19) CASP19 = -1.
EXECUTE.
VAR LAB CASP19 '(D) CASP 19'.
VAL LAB CASP19
-1 "Not applicable (including Refusals and Don't Know's)".
```

### Control (CASP)

**Variable name:** CASPCTL

**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable gives the score on the control domain of the CASP 19 score for each respondent. It uses QOLARC, QOLBRC, QOLCRC and QOLDRC as created in the derivation of CASP19.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

NUMERIC CASPCTL (F2.0).
COMPUTE CASPCTL = qolarc + qolbrc + qolcrc + qoldrc.
EXECUTE.
IF SYSMIS(CASPCTL) CASPCTL = -1.
EXECUTE.
VAR LAB CASPCTL '(D) Control (CASP)'.
VAL LAB CASPCTL
-1 "Not applicable (including Refusals and Don't Know's)".

```

## Autonomy (CASP)

**Variable name:** CASPAUT**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable gives the score on the autonomy domain of the CASP 19 score for each respondent. It uses QOLERC, QOLFRC, QOLGRC, QOLHRC and QOLIRC as created in the derivation of CASP19.

**Population:** All respondents**Level:** Individual**Derivation:**

```

NUMERIC CASPAUT (F2.0).
COMPUTE CASPAUT = qolerc + qolfrc + qolgrc + qolhrc + qolirc.
EXECUTE.
IF SYSMIS(CASPAUT) CASPAUT = -1.
EXECUTE.
VAR LAB CASPAUT '(D) Autonomy (CASP)'.
VAL LAB CASPAUT
-1 "Not applicable (including Refusals and Don't Know's)".

```

## Pleasure (CASP)

**Variable name:** CASPPLE**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable gives the score on the pleasure domain of the CASP 19 score for each respondent. It uses QOLJRC, QOLKRC, QOLLRC, QOLMRC and QOLNRC as created in the derivation of CASP19.

**Population:** All respondents**Level:** Individual

**Derivation:**

```
NUMERIC CASPPLE (F2.0).  
COMPUTE CASPPLE = qoljrc + qolkrc + qollrc + qolmrc + qolnrc.  
EXECUTE.  
IF SYSMIS(CASPPLE) CASPPLE = -1.  
EXECUTE.  
VAR LAB CASPPLE '(D) Pleasure (CASP)'.  
VAL LAB CASPPLE  
-1 "Not applicable (including Refusals and Don't Know's)".
```

## Self realisation (CASP)

**Variable name:** CASPSR**Value labels:**

-1 Not applicable (including Refusals and Don't Know's)

**Description of variable:**

This variable gives the score on the self realisation domain of the CASP 19 score for each respondent.

It uses QOLORC, QOLPRC, QOLQRC, QOLRRC and QOLSRC as created in the derivation of CASP19.

**Population:** All respondents**Level:** Individual**Derivation:**

```
NUMERIC CASPSR (F2.0).  
COMPUTE CASPSR = qolorc + qolprc + qolqrc + qolrrc + qolsrc.  
EXECUTE.  
IF SYSMIS(CASPSR) CASPSR = -1.  
EXECUTE.  
VAR LAB CASPSR '(D) Self realisation (CASP)'.  
VAL LAB CASPSR  
-1 "Not applicable (including Refusals and Don't Know's)".
```

## Number of Organisations

**Variable name:** ORGANIS**Value labels:**

-9 Not answered

-1 Not applicable

**Description of variable:**

This variable gives the number of organisations (e.g. political parties, religious groups, and social clubs) that a respondent belongs to. It is derived using the variables SCORG01-SCORG08.

**Population:** All respondents**Level:** Individual

**Derivation:**

```
NUMERIC organis (F2.0).  
DO IF scorg01 >= 0.  
COMPUTE organis = scorg01 + scorg02 + scorg03 + scorg04 + scorg05 +  
scorg06 + scorg07 + scorg08.  
END IF.  
DO IF scorg01 = -9 OR scorg01 = -1.  
COMPUTE organis = scorg01.  
END IF.  
VAR LAB organis '(D) Number of organisations'.  
VAL LAB organis  
-9 'Not answered'  
-1 'Not applicable'.
```

## Number of Organisations (Grouped)

**Variable name:** ORGNO**Value labels:**

- 9 Not answered
- 1 Not applicable
- 1 No organisation
- 2 One organisation
- 3 2-3 organisations
- 4 4 or more organisations

**Description of variable:**

This variable shows whether the respondent is in none, one, two or three or four or more organisations.

**Population:** All respondents**Level:** Individual**Derivation:**

```
NUMERIC orgno (F2.0).  
RECODE organis (0=1) (1=2) (2 thru 3=3) (4 thru Highest=4) (ELSE =  
COPY) INTO orgno.  
EXECUTE.  
VAR LAB orgno '(D) Number of organisations (grouped)'.  
VAL LAB orgno  
-9 'Not answered'  
-1 'Not applicable'  
1 'No organisation'  
2 'One organisation'  
3 '2-3 organisations'  
4 '4 or more organisations'.
```

## 7 Socio-Economic Features

### Whether New NS-SEC Code Was Assigned at ELSA Wave 2

**Variable name:** W2SEC

**Value labels:**

0 No new NS-SEC code

1 New NS-SEC code

**Description of variable:**

This variable shows whether the respondent was assigned a new NS-SEC code at Wave 2 or not by checking whether the NS-SEC code at Wave 2 is non-negative (NSSEC, not archived – see W2NSSEC).

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
COMPUTE w2sec = 0.  
IF nssec >= 0 w2sec = 1.  
variable label w2sec "(D) Whether new NS-SEC code was assigned at ELSA  
Wave 2".  
value labels w2sec 0 "No new NS-SEC code" 1 "New NS-SEC code".
```

### Whether New SIC Code Was Assigned at ELSA Wave 2

**Variable name:** W2SIC

**Value labels:**

0 No new SIC code

1 New SIC code

**Description of variable:**

This variable shows whether the respondent was assigned a new SIC92 code at Wave 2 or not by checking whether the SIC92 code at Wave 2 is non-negative (SIC92, not archived).

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
compute w2sic=0.  
if sic92>=0 w2sic=1.  
variable label w2sic "(D) Whether new SIC code was assigned at ELSA  
Wave 2".  
value labels w2sic 0 "No new SIC code" 1 "New SIC code".
```

## Whether New SOC Code Was Assigned at ELSA Wave 2

**Variable name:** W2SOC

**Value labels:**

- 0 No new SOC code
- 1 New SOC code

**Description of variable:**

This variable shows whether the respondent was assigned a new SOC2000 code at Wave 2 or not by checking whether the SOC2000 code at Wave 2 is non-negative. ESOC2000 is used, which is the 2-digit version of the 4-digit SOC200 code assigned at Wave 2. The latter has not been archived as it is considered disclosive, but is available on request from NatCen. Please see Section 8 for contact details.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
compute w2soc=0.  
if esoc2000>=0 w2soc=1.  
variable label w2soc "(D) Whether new SOC code was assigned at ELSA  
Wave 2".  
value labels w2soc 0 "No new SOC code" 1 "New SOC code".
```

## FINAL NS-SEC - Long Version (Including Unclassifiable and Non-Workers)

**Variable name:** W2NSSEC

### **Value labels:**

- 9 Refusal
- 8 Don't know
- 1 Not applicable
- 1.0 Employers in large organisations
- 2.0 Higher managerial occupations
- 3.1 Higher professional occupations - Traditional employee
- 3.2 Higher professional occupations - New employee
- 3.3 Higher professional occupations - Traditional self-employed
- 3.4 Higher professional occupations - New self-employed
- 4.1 Lower professional & higher technical - Traditional employee
- 4.2 Lower professional & higher technical - New employee
- 4.3 Lower professional & higher technical - Traditional self-employed
- 4.4 Lower professional & higher technical - New self-employed
- 5.0 Lower managerial occupations
- 6.0 Higher supervisory occupations
- 7.1 Intermediate clerical and administrative
- 7.2 Intermediate sales and service
- 7.3 Intermediate technical and auxiliary
- 7.4 Intermediate engineering
- 8.1 Employers in small organisations - non-professional
- 8.2 Employers in small organisations - agriculture
- 9.1 Own account workers - non-professional
- 9.2 Own account workers - agriculture
- 10.0 Lower supervisory occupations
- 11.1 Lower technical craft
- 11.2 Lower technical process operative
- 12.1 Semi-routine sales
- 12.2 Semi-routine service
- 12.3 Semi-routine technical
- 12.4 Semi-routine operative
- 12.5 Semi-routine agricultural
- 12.6 Semi-routine clerical
- 12.7 Semi-routine childcare
- 13.1 Routine sales and service
- 13.2 Routine production
- 13.3 Routine technical
- 13.4 Routine operative
- 13.5 Routine agricultural
- 14.0 Never worked
- 15.0 Full-time students
- 16.0 Occupations not stated or inadequately described
- 17.0 Not classifiable for other reasons

### **Description of variable:**

This variable gives the final NS-SEC code for each respondent at Wave 2 (including unclassifiable and non-workers). This variable is derived using a combination of Wave 1 NS-SEC (W1NSSEC, derived itself from the variables ANSSEC and ENSSEC, which were archived at Wave 1) and Wave 2 NS-SEC (NSSEC, not archived), because if a respondent had already been asked questions pertaining to their NS-SEC prior to Wave 2 and their circumstances had not changed at Wave 2 then they would have a missing value of NSSEC.

Also used in the derivation are the variables WPACTW and WPEVER.

**Population:** All respondents

**Level:** Individual

## Derivation:

```
if w2sec=1 w2nssec=nssec.
if w2sec=0 w2nssec=w1nssec.
EXECUTE.
DO IF w2sec=0 AND SYSMIS(w1nssec) AND SYSMIS(nssec).
COMPUTE w2nssec = -1.
END IF.
execute.
DO IF w2sec=0 AND SYSMIS(w1nssec) AND nssec = -9.
COMPUTE w2nssec = -1.
END IF.
execute.
if wpactw=4 and w2nssec=-1 and wpever = 2 w2nssec = 14.0.
execute.
variable label w2nssec "(D) FINAL NS-SEC - long version (including
unclassifiable and non-workers)".
value labels w2nssec
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal"
1.0 "Employers in large organisations"
2.0 "Higher managerial occupations"
3.1 "Higher professional occupations - Traditional employee"
3.2 "Higher professional occupations - New employee"
3.3 "Higher professional occupations - Traditional self-employed"
3.4 "Higher professional occupations - New self-employed"
4.1 "Lower professional & higher technical - Traditional employee"
4.2 "Lower professional & higher technical - New employee"
4.3 "Lower professional & higher technical - Traditional self-
employed"
4.4 "Lower professional & higher technical - New self-employed"
5.0 "Lower managerial occupations"
6.0 "Higher supervisory occupations"
7.1 "Intermediate clerical and administrative"
7.2 "Intermediate sales and service"
7.3 "Intermediate technical and auxiliary"
7.4 "Intermediate engineering"
8.1 "Employers in small organisations - non-professional"
8.2 "Employers in small organisations - agriculture"
9.1 "Own account workers - non-professional"
9.2 "Own account workers - agriculture"
10.0 "Lower supervisory occupations"
11.1 " Lower technical craft"
11.2 " Lower technical process operative"
12.1 "Semi-routine sales"
12.2 "Semi-routine service"
12.3 "Semi-routine technical"
12.4 "Semi-routine operative"
12.5 "Semi-routine agricultural"
12.6 "Semi-routine clerical"
12.7 "Semi-routine childcare"
13.1 "Routine sales and service"
13.2 "Routine production"
13.3 "Routine technical"
13.4 "Routine operative"
13.5 "Routine agricultural"
14.0 "Never worked"
15.0 "Full-time students"
16.0 "Occupations not stated or inadequately described"
17.0 "Not classifiable for other reasons".
```

## Reason Respondent Has No NS-SEC Code

**Variable name:** W2NSSECMIS

### Value labels:

- 9 Refusal
- 8 Don't know
- 1 Not applicable
- 0 Has NSSEC
- 1 No job info collected but in work at W2
- 2 Proxy: No job info collected but in work at W2
- 3 Proxy: No job info, not work W2, don't know if ever worked
- 4 No job info, not work at W2, don't know if ever worked
- 5 Not work W2, ever worked, no job info available
- 6 Proxy: Not work W2, ever worked, no job info available

### Description of variable:

This variable gives the reason why W2NSSEC is missing, using information from the Wave 2 interview outcome (W2INDOUT), WPACTW, WPEVER as well as W2NSSEC.

**Population:** All respondents

**Level:** Individual

### Derivation:

```
IF w2nssec > 0 w2nssecmis = 0.
if w2nssec=-1 and (wpactw=1 or wpactw=2) and any(w2indout,11,21)
w2nssecmis=1.
if w2nssec=-1 and (wpactw=1 or wpactw=2) and any(w2indout,13,23)
w2nssecmis=2.
if w2nssec=-1 and (wpactw=4 or wpactw = -1) and any(w2indout,13,23)
and wpever = -1 w2nssecmis=3.
if w2nssec=-1 and (wpactw=4 or wpactw = -1) and any(w2indout,11,21)
and wpever = -1 w2nssecmis=4.
if wpactw=4 and any(w2indout,11,21) and w2nssec=-1 and wpever = 1
w2nssecmis=5.
if wpactw=4 and any(w2indout,13,23) and w2nssec=-1 and wpever = 1
w2nssecmis=6.
EXECUTE.
format w2nssecmis (f2.0).
execute.
variable labels w2nssecmis "(D) Reason respondent has no NS-SEC code".
add value labels w2nssecmis
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal"
0 "Has NSSEC"
1 "No job info collected but in work at W2"
2 "Proxy: No job info collected but in work at W2"
3 "Proxy: No job info, not work W2, don't know if ever worked"
4 "No job info, not work at W2, don't know if ever worked"
5 "Not work W2, ever worked, no job info available"
6 "Proxy: Not work W2, ever worked, no job info available".
```

## FINAL NS-SEC - Long Version

**Variable name:** W2NSSEC13

### **Value labels:**

- 9 Refusal
- 8 Don't know
- 1 Not applicable
- 1.0 Employers in large organisations
- 2.0 Higher managerial occupations
- 3.1 Higher professional occupations - Traditional employee
- 3.2 Higher professional occupations - New employee
- 3.3 Higher professional occupations - Traditional self-employed
- 3.4 Higher professional occupations - New self-employed
- 4.1 Lower professional & higher technical - Traditional employee
- 4.2 Lower professional & higher technical - New employee
- 4.3 Lower professional & higher technical - Traditional self-employed
- 4.4 Lower professional & higher technical - New self-employed
- 5.0 Lower managerial occupations
- 6.0 Higher supervisory occupations
- 7.1 Intermediate clerical and administrative
- 7.2 Intermediate sales and service
- 7.3 Intermediate technical and auxiliary
- 7.4 Intermediate engineering
- 8.1 Employers in small organisations - non-professional
- 8.2 Employers in small organisations - agriculture
- 9.1 Own account workers - non-professional
- 9.2 Own account workers - agriculture
- 10.0 Lower supervisory occupations
- 11.1 Lower technical craft
- 11.2 Lower technical process operative
- 12.1 Semi-routine sales
- 12.2 Semi-routine service
- 12.3 Semi-routine technical
- 12.4 Semi-routine operative
- 12.5 Semi-routine agricultural
- 12.6 Semi-routine clerical
- 12.7 Semi-routine childcare
- 13.1 Routine sales and service
- 13.2 Routine production
- 13.3 Routine technical
- 13.4 Routine operative
- 13.5 Routine agricultural

### **Description of variable:**

This variable is based on W2NSSEC but does not include unclassifiable and non-workers.

**Population:** All respondents

**Level:** Individual

## Derivation:

```
RECODE w2nssec (14 = -1) (16 = -1) (ELSE = COPY) INTO w2nssec13.
variable label w2nssec13 "(D) FINAL NS-SEC - long version".
value labels w2nssec13
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal"
1.0 "Employers in large organisations"
2.0 "Higher managerial occupations"
3.1 "Higher professional occupations - Traditional employee"
3.2 "Higher professional occupations - New employee"
3.3 "Higher professional occupations - Traditional self-employed"
3.4 "Higher professional occupations - New self-employed"
4.1 "Lower professional & higher technical - Traditional employee"
4.2 "Lower professional & higher technical - New employee"
4.3 "Lower professional & higher technical - Traditional self-
employed"
4.4 "Lower professional & higher technical - New self-employed"
5.0 "Lower managerial occupations"
6.0 "Higher supervisory occupations"
7.1 "Intermediate clerical and administrative"
7.2 "Intermediate sales and service"
7.3 "Intermediate technical and auxiliary"
7.4 "Intermediate engineering"
8.1 "Employers in small organisations - non-professional"
8.2 "Employers in small organisations - agriculture"
9.1 "Own account workers - non-professional"
9.2 "Own account workers - agriculture"
10.0 "Lower supervisory occupations"
11.1 " Lower technical craft"
11.2 " Lower technical process operative"
12.1 "Semi-routine sales"
12.2 "Semi-routine service"
12.3 "Semi-routine technical"
12.4 "Semi-routine operative"
12.5 "Semi-routine agricultural"
12.6 "Semi-routine clerical"
12.7 "Semi-routine childcare"
13.1 "Routine sales and service"
13.2 "Routine production"
13.3 "Routine technical"
13.4 "Routine operative"
13.5 "Routine agricultural".
```

## NS-SEC 8 Variable Classification (Individual)

**Variable name:** W2NSSEC8

### Value labels:

- 1 Not applicable
- 1 Higher managerial and professional occupations
- 2 Lower managerial and professional occupations
- 3 Intermediate occupations
- 4 Small employers and own account workers
- 5 Lower supervisory and technical occupations
- 6 Semi-routine occupations
- 7 Routine occupations
- 8 Never worked and long term unemployed
- 99 Other

### Description of variable:

This variable gives the 8 category version of the NS-SEC classification for individuals.

**Population:** All respondents

**Level:** Individual

### Derivation:

```
RECODE w2nssec (1 thru 3.4=1) (4 thru 6=2) (7 thru 7.4=3) (8 thru
9.2=4) (10 thru 11.2=5) (12 thru 12.7=6) (13 thru 13.5=7) (14 thru
14.2=8) (15 thru 17=99) (else=copy) into w2nssec8.
Variable labels w2nssec8 "(D) NS-SEC 8 variable classification
(individual)".
Value labels w2nssec8
-1 "Not applicable"
 1 "Higher managerial and professional occupations"
 2 "Lower managerial and professional occupations"
 3 "Intermediate occupations"
 4 "Small employers and own account workers"
 5 "Lower supervisory and technical occupations"
 6 "Semi-routine occupations"
 7 "Routine occupations"
 8 "Never worked and long term unemployed"
99 "Other".
format w2nssec8 (f2.0).
```

## NS-SEC 5 Variable Classification (Individual)

**Variable name:** W2NSSEC5

**Value labels:**

- 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 occupations
- 99 Other

**Description of variable:**

This variable gives the 5 category version of the NS-SEC classification for individuals.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
RECODE w2nssec (1 thru 6=1) (7 thru 7.4=2) (8 thru 9.2=3) (10 thru
11.2=4) (12 thru 13.5=5) (14 thru 17=99) (else=copy) INTO w2nssec5.
Variable label w2nssec5 "(D) NS-SEC 5 variable classification
(individual)".
Value label w2nssec5
-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 occupations"
99 "Other".
format w2nssec5 (f2.0).
```

## NS-SEC 3 Variable Classification (Individual)

**Variable name:** W2NSSEC3

**Value labels:**

- 1 Not applicable
- 1 Managerial and professional occupations
- 2 Intermediate occupations
- 3 Routine and manual occupations
- 99 Other

**Description of variable:**

This variable gives the 3 category version of the NS-SEC classification for individuals.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```

RECODE w2nssec (1 thru 6=1) (7 thru 9.2=2) (10 thru 13.5=3) (14 thru
17=99) (else=copy) INTO w2nssec3.
Variable label w2nssec3 "(D) NS-SEC 3 variable classification
(individual)".
Value label w2nssec3
-1 "Not applicable"
 1 "Managerial and professional occupations"
 2 "Intermediate occupations"
 3 "Routine and manual occupations"
99 "Other".
format w2nssec3 (f2.0).

```

**Combined ELSA Wave 2/Wave 1/HSE: SOC2000 (without dots)****Variable name:** W2SOC2000**Value labels:**

-1 Not applicable

**Description of variable:**

This variable gives the final SOC2000 code for each respondent at Wave 2. This variable is derived using a combination of Wave 1 SOC2000 (W1SOC2000, derived itself from the variables AXSC2000 and EXS2000, which were archived at Wave 1) and Wave 2 SOC2000 (ESOC2000, archived with these derived variables), because if a respondent had already been asked questions pertaining to their SOC2000 prior to Wave 2 and their circumstances had not changed at Wave 2 then they would have a missing value of esoc2000.

**Population:** All respondents**Level:** Individual**Derivation:**

```

if w2soc=1 w2soc2000=esoc2000.
if w2soc=0 w2soc2000=w1soc2000.
DO IF SYSMIS(w1soc2000) AND esoc2000 = -1.
COMPUTE w2soc2000 = -1.
END IF.
execute.
value label w2soc2000
-1 "Not applicable".
variable label w2soc2000 "(D) Combined ELSA W2/W1/HSE: SOC2000
(without dots)".
execute.

```

## Reason Why Respondent Has No SOC2000

**Variable name:** W2SOC2000MIS

### Value labels:

- 9 Refusal
- 8 Don't know
- 1 Not applicable
- 0 Has SOC2000
- 1 No job info collected but in work at W2
- 2 Proxy: No job info collected but in work at W2
- 3 Proxy: No job info, not work W2, don't know if ever worked
- 4 No job info, not work at W2, don't know if ever worked
- 5 Not work W2, ever worked, no job info available
- 6 Proxy: Not work W2, ever worked, no job info available
- 7 Never worked
- 8 Occupation not stated or inadequately classified

### Description of variable:

This variable gives the reason why W2SOC2000 is missing, and is derived from W2NSSEC and W2NSSECMIS,

**Population:** All respondents

**Level:** Individual

### Derivation:

```
NUMERIC w2soc2000mis (F2.0).
IF w2soc2000 >= 0 w2soc2000mis = 0.
IF w2nssecmis = 1 w2soc2000mis = 1.
IF w2nssecmis = 2 w2soc2000mis = 2.
IF w2nssecmis = 3 w2soc2000mis = 3.
IF w2nssecmis = 4 w2soc2000mis = 4.
IF w2nssecmis = 5 w2soc2000mis = 5.
IF w2nssecmis = 6 w2soc2000mis = 6.
IF w2nssec = 14 w2soc2000mis = 7.
IF w2nssec = 16 w2soc2000mis = 8.
VAR LAB w2soc2000mis "(D) Reason why respondent has no SOC2000".
VAL LAB w2soc2000mis
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal"
0 "Has SOC2000"
1 "No job info collected but in work at W2"
2 "Proxy: No job info collected but in work at W2"
3 "Proxy: No job info, not work W2, don't know if ever worked"
4 "No job info, not work at W2, don't know if ever worked"
5 "Not work W2, ever worked, no job info available"
6 "Proxy: Not work W2, ever worked, no job info available"
7 "Never worked"
8 "Occupation not stated or inadequately classified".
```

## Combined ELSA Wave 2/Wave 1/HSE: 2-Digit SIC92 Code

**Variable name:** W2SIC92

**Value labels:**

- 9 Refusal
- 8 Don't know
- 1 Not applicable

**Description of variable:**

This variable gives the final SIC92 code for each respondent at Wave 2. This variable is derived using a combination of Wave 1 SIC92 (W1SIC92, itself derived from SIC92 at HSE and Wave 1) and Wave 2 SIC92 (SIC92, not archived), because if a respondent had already been asked questions pertaining to their SIC92 prior to Wave 2 and their circumstances had not changed at Wave 2 then they would have a missing value of SIC92.

**Population:** All respondents

**Level:** Individual

**Derivation:**

```
if w2sic=1 w2sic92=sic92.
if w2sic=0 w2sic92=w1sic92.
execute.
DO IF SYSMIS(w1sic92) AND sic92 = -1.
COMPUTE w2sic92 = -1.
END IF.
EXECUTE.
DO IF SYSMIS(w1sic92) AND sic92 = -8.
COMPUTE w2sic92 = -8.
END IF.
EXECUTE.
DO IF ((SYSMIS(w1sic92) AND sic92 = -9) OR (w1sic92 = -1 AND sic92 = -9)).
COMPUTE w2sic92 = -9.
END IF.
EXECUTE.
value label w2sic92
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal".
variable label w2sic92 "(D) Combined ELSA W2/W1/HSE: 2-digit SIC92 code".
EXECUTE.
```

## Reason Why Respondent Has No SIC92

**Variable name:** W2SIC92MIS

### Value labels:

- 9 Refusal
- 8 Don't know
- 1 Not applicable
- 0 Has SIC92
- 1 No job info collected but in work at W2
- 2 Proxy: No job info collected but in work at W2
- 3 Proxy: No job info, not work W2, don't know if ever worked
- 4 No job info, not work at W2, don't know if ever worked
- 5 Not work W2, ever worked, no job info available
- 6 Proxy: Not work W2, ever worked, no job info available
- 7 Never worked
- 8 Occupation not stated or inadequately classified

### Description of variable:

This variable gives the reason why W2SIC92 is missing, and is derived using W2SIC92, WPDET and WPJDO.

**Population:** All respondents

**Level:** Individual

### Derivation:

```
NUMERIC w2sic92mis (F2.0).
IF w2sic92 = -9 w2sic92mis = -9.
IF w2sic92 >= 0 w2sic92mis = 0.
IF w2nssecmis = 1 w2sic92mis = 1.
IF w2nssecmis = 2 w2sic92mis = 2.
IF w2nssecmis = 3 w2sic92mis = 3.
IF w2nssecmis = 4 w2sic92mis = 4.
IF w2nssecmis = 5 w2sic92mis = 5.
IF w2nssecmis = 6 w2sic92mis = 6.
IF w2nssec = 14 w2sic92mis = 7.
IF w2nssec = 16 OR (SYSMIS(w2sic92mis) AND wpdet = " " AND wpjdo = "
")
w2sic92mis = 8.
VAR LAB w2sic92mis "(D) Reason why respondent has no SIC92".
VAL LAB w2sic92mis
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal"
0 "Has SIC92"
1 "No job info collected but in work at W2"
2 "Proxy: No job info collected but in work at W2"
3 "Proxy: No job info, not work W2, don't know if ever worked"
4 "No job info, not work at W2, don't know if ever worked"
5 "Not work W2, ever worked, no job info available"
6 "Proxy: Not work W2, ever worked, no job info available"
7 "Never worked"
8 "Occupation not stated or inadequately classified".
```

## Highest Other Educational Qualification at ELSA Wave 2

**Variable name:** W2OTHERQ

### Value labels:

- 9 Refusal
- 8 Don't know
- 1 Not applicable
- 1 NVQ4/NVQ5/Degree or equivalent
- 2 Higher education below degree
- 3 NVQ3/GCE A Level equivalent
- 4 NVQ2/GCE O Level equivalent
- 5 NVQ1/CSE other grade equivalent
- 6 Foreign/other
- 7 No qualification

### Description of variable:

This variable gives the highest other educational qualification achieved at Wave 2, as opposed to the highest main qualification. FQQUAZ1 and FQQUAZ2 are used (not archived).

**Population:** All respondents

**Level:** Individual

### Derivation:

```
NUMERIC w2otherq (F2.0).
COMPUTE w2otherq=0.
DO REPEAT XXA=fqquaz1 to fqquaz2.
IF (ANY(XXA,1,23,24)) w2otherq=1.
IF (ANY(XXA,2,3,4,6) & w2otherq ~= 1) w2otherq=2.
IF (ANY(XXA,5,7,9,10,11,25) & ~RANGE(w2otherq,1,2)) w2otherq=3.
IF (ANY(XXA,8,12,13,15,17,20,22,26) & ~RANGE(w2otherq,1,3))
w2otherq=4.
IF (ANY(XXA,14,16,18,21,27,28) & ~RANGE(w2otherq,1,4)) w2otherq=5.
IF (ANY(XXA,29,51,52,53,54,55,56,57,58,85,95) & ~RANGE(w2otherq,1,5))
w2otherq=6.
IF ((XXA=19) | (fqquaz1=96) | (fqquaz1=86)) & ~RANGE(w2otherq,1,6)
w2otherq=7.
END REPEAT.
IF fqquaz1<0 and ~RANGE(w2otherq,1,7) w2otherq=fqquaz1.
VARIABLE LABEL w2otherq "(D) Highest Other Educational Qualification
at ELSA W2".
VALUE LABELS w2otherq
-9 Refusal
-8 Don't know
-1 Not applicable
1 NVQ4/NVQ5/Degree or equivalent
2 Higher education below degree
3 NVQ3/GCE A Level equivalent
4 NVQ2/GCE O Level equivalent
5 NVQ1/CSE other grade equivalent
6 Foreign/other
7 No qualification.
```

## Highest Educational Qualification Obtained by ELSA W2

**Variable name:** W2EDQUAL

### Value labels:

- 9 Refusal
- 8 Don't know
- 2 In W2 only
- 1 Not applicable
- 1 NVQ4/NVQ5/Degree or equivalent
- 2 Higher education below degree
- 3 NVQ3/GCE A Level equivalent
- 4 NVQ2/GCE O Level equivalent
- 5 NVQ1/CSE other grade equivalent
- 6 Foreign/other
- 7 No qualification

### Description of variable:

This variable gives the highest educational qualification achieved by Wave 2. It uses W1EDQUAL (archived as EDQUAL at Wave 1), as well as variables relating to qualifications at Wave 2 (W2OTHERQ).

**Population:** All respondents

**Level:** Individual

### Derivation:

```
NUMERIC w2edqual (F2.0).
COMPUTE w2edqual=0.
DO REPEAT XXA=fqquzm1 to fqquzm2 fqqual1 to fqqual3.
IF (ANY(XXA,1,23,24)) w2edqual=1.
IF (ANY(XXA,2,3,4,6) & w2edqual~1) w2edqual=2.
IF (ANY(XXA,5,7,9,10,11,25) & ~RANGE(w2edqual,1,2)) w2edqual=3.
IF (ANY(XXA,8,12,13,15,17,20,22,26) & ~RANGE(w2edqual,1,3))
w2edqual=4.
IF (ANY(XXA,14,16,18,21,27,28) & ~RANGE(w2edqual,1,4)) w2edqual=5.
IF (ANY(XXA,29) & ~RANGE(w2edqual,1,5)) w2edqual=6.
END REPEAT.
DO REPEAT XXA=fqquzm1 to fqquzm2.
IF (ANY(XXA,51,52,53,54,55,56,57,58,85,95) & ~RANGE(w2edqual,1,5))
w2edqual=6.
IF ((XXA=19 | (fqaqua=2 and fmqqua=2) | (fqaqua=2 and fmqqua<0) |
(fqquzm1=96) | (ANY(fqaqua,1,2) AND fmqqua=1 and fqquzm1=86)
| (fqaqua<0 & fqquzm1=86)) & ~RANGE(w2edqual,1,6)) w2edqual=7.
END REPEAT.
IF ((fqaqua=1 AND fmqqua<0) & ~RANGE(w2edqual,1,7)) w2edqual=fmqqua.
IF ((ANY(fqaqua,1,2) AND fmqqua=1 and fqquzm1<0) &
~RANGE(w2edqual,1,7)) w2edqual=fqquzm1.
IF (fqaqua<0 & fqquzm1<0 and ~RANGE(w2edqual,1,7)) w2edqual=fqquzm1.
IF (w2edqual > 0 and wledqual > 0 AND w2edqual > wledqual) w2edqual =
wledqual.
IF w2edqual = -1 AND wledqual > 0 w2edqual = wledqual.
IF (w2edqual > 0 and w2otherq > 0 AND w2edqual > w2otherq) w2edqual =
w2otherq.
IF (w2edqual = -1 AND w2otherq > 0) w2edqual = w2otherq.
IF (w2edqual2=-1 and (wledqual=-8 or wledqual=-9)) w2edqual2=wledqual.
IF (w2edqual2=-1 and (fqaqua=-8 or fqaqua=-9)) w2edqual2=fqaqua.
EXECUTE.
```

```
VARIABLE LABEL w2edqual "(D) Highest Educational Qualification
obtained by ELSA W2".
VALUE LABELS w2edqual
  1 'NVQ4/NVQ5/Degree or equivalent'
  2 'Higher education below degree'
  3 'NVQ3/GCE A Level equivalent'
  4 'NVQ2/GCE O Level equivalent'
  5 'NVQ1/CSE other grade equivalent'
  6 'Foreign/other'
  7 'No qualification'
-1 "Not applicable"
-8 "Don't know"
-9 "Refusal".
```

## 8 Contact details

In the event of any queries please contact:

Susan Nunn  
ELSA Senior Data Manager  
E-mail: [susan.nunn@natcen.ac.uk](mailto:susan.nunn@natcen.ac.uk)  
Telephone: 020 7250 1866

## 9 Acknowledgements

We are grateful to our academic collaborators at University College London for providing the syntax used to derive a number of these variables.