SHARE ERIC
SURVEY OF HEALTH, AGEING AND RETIREMENT IN EUROPE

SHARE WORKING PAPER SERIES

Generating *easy*SHARE Release 7.0.0

Guidelines, Structure, Content and Programming

Stefan Gruber

Working Paper Series 37-2019
April 10, 2019

SHARE-ERIC | Amalienstr. 33 | 80799 Munich | Germany | share-eric.eu

MAX PLANCK INSTITUTE FOR SOCIAL LAW AND SOCIAL POLICY
Munich Center for the Economics of Aging

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676536.

SPONSORED BY THE Federal Ministry of Education and Research

Supported by the National Institute on Aging

Generating *easy*SHARE Release 7.0.0

Guidelines, Structure, Content and Programming

Stefan Gruber[1]

April 10, 2019

Abstract

This paper provides an overview of the construction, structure and content of *easy*SHARE – a simplified dataset that is based on the scientific release of the Survey of Health, Ageing and Retirement in Europe (SHARE). *easy*SHARE contains a subset of ready-to-analyse variables and indices. It is designed for student training on country-comparative and longitudinal analyses as well as for introducing SHARE to researchers who have little experience in quantitative analyses of complex survey data. This paper puts special emphasis on the extraction and generation of the dataset using the statistical software package Stata and provides additionally an example of how to extract more variables from the scientific release of SHARE to include them in the *easy*SHARE dataset.

Acknowledgement

This paper uses data from SHARE Waves 1, 2, 3, 4, 5, 6 and 7 (DOIs: 10.6103/SHARE.w1.700, 10.6103/SHARE.w2.700, 10.6103/SHARE.w3.700, 10.6103/SHARE.w4.700, 10.6103/SHARE.w5.700, 10.6103/SHARE.w6.700, 10.6103/SHARE.w7.700), see Börsch-Supan et al. (2013) for methodological details. The SHARE data collection has been primarily funded by the European Commission through FP5 (QLK6-CT-2001-00360), FP6 (SHARE-I3: RII-CT-2006-062193, COMPARE: CIT5-CT-2005-028857, SHARELIFE: CIT4-CT-2006-028812) and FP7 (SHARE-PREP: N°211909, SHARE-LEAP: N°227822, SHARE M4: N°261982). Additional funding from the German Ministry of Education and Research, the Max Planck Society for the Advancement of Science, the U.S. National Institute on Aging (U01_AG09740-13S2, P01_AG005842, P01_AG08291, P30_AG12815, R21_AG025169, Y1-AG-4553-01, IAG_BSR06-11, OGHA_04-064, HHSN271201300071C) and from various national funding sources is gratefully acknowledged (see www.share-project.org).

[1] Munich Center for the Economics of Aging (MEA), Max Planck Institute for Social Law and Social Policy

1. Introduction

The Survey of Health, Ageing and Retirement in Europe (SHARE) is a rich database that provides information on health, socio-economic status and social and family networks of more than 140,000 individuals aged 50 and over. Up to the present, SHARE has released seven waves of which wave 3 and wave 7 contain retrospective life history data (SHARELIFE). For more details on SHARELIFE see Börsch-Supan et al. (2013a) and Schröder (2011). Further information on the methodology and content of the regular waves is provided by Börsch-Supan and Jürges (2005), Börsch-Supan et al. (2005, 2008, 2011 and 2013b,) and Malter and Börsch-Supan (2013).

Due to its richness SHARE is also complex: the multitude of countries participating in SHARE with their institutional variety and different languages, combined with a large interdisciplinary set of variables stored in single data modules, and the differentiation of individual-, couple- and household-level data, add to this complexity.To reduce complexity, the *easy*SHARE dataset was generated. It is a single simplified file generated for student training and for researchers who are less experienced in the quantitative analysis of complex panel data.

*easy*SHARE includes the same number of observations as the main release of SHARE but is restricted to a subset of variables. It contains the regular panel waves of SHARE (waves 1, 2, 4, 5, 6 and 7) and some information collected in the retrospective SHARELIFE modules of waves 3 and 7. This allows storing *easy*SHARE as one single data file without the need for complex merging of waves and/or modules. The file includes selected information from the SHARE CAPI (Computer Assisted Personal Interview) modules and in some instances measures collected in the the drop-off questionnaires are included, too.Variables from the "generated variables datasets" are also incorporated as well as additional "easy to use"-variables.

To facilitate analyzing the data, information collected on couple- and household-level is transferred to the individual level. Several generated variables, such as indices and recoded health, demographic, social and economic measures allow analyses without the need for extensive data preparation. Additionally, a series of missing value codes is added to give detailed information on why exactly information is missing.

This SHARE working paper gives an overview on the *easy*SHARE dataset. First we summarize the guidelines that *easy*SHARE is based on, followed by a description of the data structure. The next section depicts the subset of variables that the dataset contains. Section 5 explains the general structure of the Stata program ("do-file") that we use to extract information from the scientific release files of SHARE to generate *easy*SHARE. We also describe some procedures in more detail, e.g. the generation of age at interview or how we deal with possible deviations in supposedly stable information. Section 6 provides an example of how to add more SHARE variables to *easy*SHARE.

2. Guidelines

The creation of longitudinal interdisciplinary databases such as the U.S. Health and Retirement Study (HRS), the English Longitudinal Study on Ageing (ELSA), and the Survey of Health, Ageing and Retirement in Europe (SHARE) has helped enormously to understand the life situation of older people in Europe and the US. One special interest of these studies is to understand the process of ageing in its multifaceted dimensions by facilitating interdisciplinary research on topics like health and health behaviour, wealth, retirement, social networks, etc. (cf. Börsch-Supan et al. 2013b, Banks et al. 2012, National Institute on Aging 2007). A main challenge of studies in these fields of research is to identify causal pathways. Two crucial ingredients for successful identification are a) a set of reliable and multidimensional measures at different points in time and b) sufficiently exogenous policy variation shaping the socio-economic environment and the repondents´ behaviour.

A major significance of SHARE lies in the second ingredient: exogenous policy variation. The strategy of SHARE and its sister surveys thus is to exploit cross-national variation. The essential argument is that the variety of circumstances and policies is in general much larger across countries than within a single country. Researchers and policy makers can learn from what has happened and what has been tried elsewhere. Therefore, *cross-country comparability* is a guiding principle in SHARE in general and also in *easy*SHARE.

To facilitate longitudinal analyses, another substantial guideline is to attain a maximum of *cross-wave comparability*. The majority of variables and indices included in the *easy*SHARE dataset have been collected in all waves of SHARE. Of course, data on respondents´ life histories (SHARELIFE) collected in waves 3 and 7 is a challenge to this principle. Due to its retrospective scope, the majority of variables in SHARELIFE does not contain the same information as the regular panel waves that focus on respondents´ current living conditions. Consequently, many variables of the *easy*SHARE dataset contain missing values for waves 3 and 7.

The *inclusion of new variables and indices* that are helpful for users but have not been made available in the SHARE data so far is another guideline. Among the examples are the wave-specific income percentiles, the CASP index measuring quality of life and well-being as well as a broad variety of physical and mental health measures and indices, most of them adapted to the HRS.

A further guideline is to reduce the complexity of data handling and data preparation by *copying information*. Time-constant information that is only asked once in the baseline interview (i.e. educational level) is copied to all later records of the person. Additionally, information collected only from one person of a couple or in a household (e.g. whether the household is able to make ends meet) is transferred to all respective respondents if applicable.

Furthermore, an *extended missing code scheme* is provided in *easy*SHARE in order to reduce complexity and to make the reasons behind missing information more comprehensible.

3. Structure of the data

Panel data can be displayed in long or in wide format. *easy*SHARE is stored in long format, i.e. one data line represents one wave in which a respondent participated. For example, if there are seven observations with the same respondent identifier (`mergeid`), it means that this respondent took part in all seven waves of SHARE. The columns represent variables and store the information reported. One advantage of the long format is that for most statistical packages this is the default setting for panel analyses.

Table 1 shows the structure of the data for two respondents. The first respondent was interviewed in Austria, which can be derived from the prefix "AT" in the `mergeid` or from the country variables (`country` or `country_mod`). This respondent took part in all seven waves as represented by seven data lines. The second respondent who was interviewed in Sweden (prefix "SE") participated in wave 1, did not participate in waves 2 to 6 , and then took part again in wave 7. Hence, for this respondent the dataset contains only two data lines. The information on respondents´ wave participation is also stored in the variable `wavepart`. You can use this variable to select balanced panels, e.g. all respondents who participated in all seven waves.

Table 1: Data structure

mergeid	wave	country	wavepart
AT-004855-02	1	Austria	1234567
AT-004855-02	2	Austria	1234567
AT-004855-02	3	Austria	1234567
AT-004855-02	4	Austria	1234567
AT-004855-02	5	Austria	1234567
AT-004855-02	6	Austria	1234567
AT-004855-02	7	Austria	1234567
SE-923080-02	1	Sweden	17
SE-923080-02	7	Sweden	17

Another handy feature of the *easy*SHARE dataset is that it contains no system-missing values. Compared to the scientific SHARE release, for the subset of variables selected the missing values were recoded to one of the following (partially new) missing codes:

-3: "implausible value/suspected wrong"
-7: "not yet coded"
-9: "not applicable filtered"
-10: "SHARELIFE interview" (only in wave 7)
-11: "regular interview" (only in wave 7)
-12: "don't know / refusal"
-13: "not asked in this wave"
-14: "not asked in this country"
-15: "no information"
-16: "no drop-off (information in drop-off in this wave)"

In very rare cases we set non-system-missing values to -3: "implausible value/suspected wrong". This missing code is applied e.g. when the respondent´s body mass index (`bmi`) is too low (between 0 and 12) or too high (over 100) to be realistic. A few variables are not yet coded in the data of some countries. For those the missing code -7: "not yet coded" is used. The missing value -9: "not applicable; filtered" is applied in case that a question was not asked to specific respondents due to questionnaire routing. Whenever possible, we add the reason for the filtering to the label. The -9 code is only used for filtering "within the questionnaire", but not for country or wave "filters". The wave 7 questionnaire contains SHARELIFE modules for all respondents who did not participate in wave 3 (first SHARELIFE wave), as well as standard modules for respondents who already participated in the SHARELIFE interview of wave 3. In order to identify missings based on the interview version of wave 7, the missing codes -10 "SHARELIFE interview" and -11 "regular interview" are introduced for *easy*SHARE. There is only one combined missing value code -12 for "don't know" and "refusal" in *easy*SHARE. The reason is that when combining two or more variables of the main release of SHARE into a new variable, it cannot be distinguished between possibly different missing codes of the original variables. For the consistency within *easy*SHARE the combined "don't know/refusal" code is used for all variables. The missing code -13: "not asked in this wave" is implemented for items that are not included in a specific wave. If a question is not asked in one of the SHARE countries, the code -14 "not asked in this country" is assigned. The code -15: "no information" is used for system-missing values that are not explained by any of the described reasons, for example in cases where respondents may not have completed the interview.

The so-called 'drop-off' is a paper-and-pencil questionnaire that includes additional questions not asked in the main interview on issues like mental and physical health, health care or social relations. If respondents did not fill in and send back the drop-off questionnaire or if the drop-off cannot be correctly linked to a respondent, the missing code -16: "no drop-off (information in drop-off in this wave)" is used.

4. Content of *easy*SHARE

The major criterions for including variables in *easy*SHARE are *cross-country* and *cross-wave comparability*. Moreover, the variables should cover the central topics of SHARE and attend to requests by SHARE users.

We select the following topics for *easy*SHARE:

1) **Demographics**: age, gender, country of birth, citizenship, education, religion, marital status, age and gender of partner
2) **Household composition**: living with partner in the same household, household size, children living in the household
3) **Social support & network**: mother/father alive, number of children, residential proximity of children, number of grandchildren, number of living siblings, social activities, received and given social support
4) **Childhood conditions**: number of books at age ten, relative mathematical skills at age ten, relative language skills at age ten, childhood health status, being vaccinated during childhood
5) **Health and health behavior**: self-perceived health, number of chronic diseases, mental health variables, depression scale EURO-D, CASP-12 index for quality of life and well-being, health care utilization, grip strength, body mass index, smoking and drinking behavior, vigorous activities/sports
6) **Functional limitation indices**: mobility index, large muscle index, activities of daily living index, gross motor skills index, fine motor skills index, instrumental activities of daily living index, cognitive functions
7) **Work & money**: current job situation, term of main job, working hours per week, satisfaction with main job, early retirement plans, able to make ends meet, imputed household net income, wave specific household income percentiles

Some variables in *easy*SHARE are copies of the respective variables included in the main release of SHARE that are only complemented by recoding the system-missing values due to country-/wave-missing patterns and questionnaire routing. These variables are labelled identical in *easy*SHARE as in the SHARE main release.

Variables that have been generated specifically for *easy*SHARE receive a new variable label. Indices or scores are named according to general conventions (i.e. CASP, IADLA, etc.). Variables that have been modified, but do not represent a new concept, have got an additional "_mod" suffix.

5. The Stata program generating *easy*SHARE

The Stata program (do-file) that generates the *easy*SHARE dataset based on the scientific release files of SHARE is included in full length in the appendix. For the most part, the aim is a comprehensible code that also allows users with little experience to trace the generation process of *easy*SHARE. This section describes the structure of the program and gives details on some parts that require additional explanation.

In a first step a couple of folders are created to store the various datasets used for generating *easy*SHARE. The datasets of the regular scientific release are copied into a new folder system. There are two reasons to engage in this time- and disc-space-intensive procedure. First, it is to make sure that the original release files of SHARE are not accidentally modified. Second, it is to document which version of the release files is used to build the respective version of *easy*SHARE.

*easy*SHARE release 7.0.0 is based on the respective current release of each wave, i.e. release 7.0.0. Apart from the coverscreen module (cv_r) and the drop-off questionnaire, information from the questionnaire modules demographics (DN respectively ST in wave 3), activities (AC), behavioural risks (BR), cognitive functioning (CF), children (CH), consumption (CO), the SHARELIFE childhood section (CS), employment (EP), grip strength (GS), health care (HC), housing (HO), interviewer observations (IV), mini childhood (MC), physical health (PH), social networks (SN), social support (SP), restrospective health section (HS), and retrospective health care (RH) is used. From the generated variable modules gv_health, gv_isced, gv_imputations, gv_gross_net as well as the technical variables module are used. Note that the final *easy*SHARE dataset only stores the observations of *household members who were interviewed* in the respective wave.

After preparing the coverscreen module for each wave (step 4 in the do-file), the extraction of variables from the single questionnaire module datasets starts. Each of the datasets is opened and only those variables are kept that should be included in the final *easy*SHARE dataset or that are needed to generate other variables. This section is sorted per module and is the longest part of the do-file (steps 5 to 28). If necessary, variables are recoded and some new variables are generated in this section before the reduced module datasets are saved separately. In the next step these reduced datasets are merged per wave so we get one dataset for each of the seven waves that stores all variables extracted before (step 29). Next, adjustments within the merged wave-specific datasets are made, like assigning information that was only gathered by one member of a household to the other household members (step 30).

Subsequently the waves are appended, thereby generating a panel 'long format dataset'. In this step (step 31) the variables are integrated to 'long format variables' and a variable that stores the information on the respondents´ wave participation is generated (`wavepart;` see table 1).

Based on the long format dataset the 'long' variables can be checked and recoded, e.g. in cases in which deviations across waves exist. To generate an `age` at interview variable for as many cases as possible, we impute missing information for the date of interview and use all information on birth year and birth month collected in the up to seven interviews. This also

implies to first check whether birth year and month and other stable information is consistent over time and to deal with deviating information. Missing information for the date of interview is imputed by substituting the wave- and country-specific mode[3]. If birth month deviates across waves, the mode month of all self-reported month-of-birth values is taken (`dn002_` in the main release of SHARE)[4].

If no information on the month of birth is available, it is assumed that these respondents are born in June. If available, missing information on gender, year and month of birth is replaced by values collected in other waves. Afterwards the respondents´ age and age of partner variables are generated. Note that even though a couple of replacements in various variables are made, the resulting `age` variable can only deviate up to approx. 2 years. This is due to the fact that the imputation in the year of birth, and all other changes introduce a maximum of 11 or 12 months of deviation to the true values. The magnitude of these (potential) "measurement" errors is in general regarded as less harmful than a listwise deletion of all cases with the described problems. However, all researchers working with *easy*SHARE data should be aware about those data preparation procedures. Whether or not to modify the data in the ways described is mainly a decision that depends on the specific research question.

In the subsequent part of the program time-constant information collected once, usually in the first interview of a respondent – the so-called baseline interview – is transferred to later waves (step 33). This is done for the variable `isced_r` containing the ISCED-97[5] values, years of education (`eduyears`), country of birth (`birth_country`), and the respondents´ `citizenship`. For various reasons, e.g. if the baseline interview was not completed or interviewers mixed up the respondents, it is possible to have the same time-invariant information from more than one interview. In very few cases, information can deviate. As there is no way of determining which of the reports is accurate, all information given is ignored.

Furthermore, some time-variant information in the SHARE main release is asked and stored in ways that call for an easier handling in *easy*SHARE. If, for example, the information on marital status, living siblings and parents, or smoking behaviour did not change, the actual question is not asked and hence the target variable is system-missing. Only a screening question that assesses whether the detail has changed since the previous interview stores this information. Therefore, the information is copied from the previous interview to the present one if the respondent indicates that the respective information has not changed (step 34).

Afterwards, both value and variable labels are added, reassigned, and completed where necessary (step 35). In the final steps, the *easy*SHARE missing code scheme is completed (steps 36 and 37), the list of relevant variables is kept and ordered and the final easySHARE dataset is saved and compressed (step 38).

[3] If the mode was not unique (equal number of interviews in two different months or years in the respective country) we take the ealier year/month.
[4] Again the minimum month is used if the mode was not unique. This occurs often, e.g. when we have two reports that deviate.
[5] ISCED stands for International Standard Classification of Education. SHARE assumes that its respondents who are mostly aged 50 or older do not upgrade or change their education level.

The above explanations show that various assumptions and decisions had to be made in the generation of the *easy*SHARE dataset. For many research questions and analyses these assumptions might be adequate. Of course, there are analyses for which the procedures are not appropriate or not the best solution. All *easy*SHARE users should be aware of this.

6. How to include other variables

The *easy*SHARE dataset is restricted to specific variables chosen according to the guidelines described above. Of course, there are many other variables in the SHARE data that might be of interest for users. One solution is to modify the *easy*SHARE do-file to extract more variables. The advantage is that the routines for imputing missing information or for solving deviations in stable information can easily be used for additional variables if necessary. Users should be aware that this usually requires changes at multiple points in the do-file. Therefore, a second strategy to extend the *easy*SHARE dataset is described in the following using the example of respondents´ area of residence (`areabldgi`) as stored in the gv_housing module. The variable contains information as to whether respondents live in a big city, suburb, large town, small town or rural area, respectively a village.

If the SHARE main release data are not downloaded yet, users first have to download the gv_housing modules of the waves they are interested in. The example uses all waves except wave 3 because area of residence was not part of the wave 3 SHARELIFE questionnaire. The first step of the Stata command below opens the wave-specific gv_housing modules, generates a `wave` variable and keeps the variables `areabldgi`, `mergeid` and `wave`.

```
use  "C:\SHARE\data\sharew1_rel7-0-0_gv_housing.dta", clear
gen wave = 1
keep mergeid wave areabldgi
save "C:\SHARE\data\area_w1.dta", replace

use  "C:\SHARE\data\sharew2_rel7-0-0_gv_housing.dta", clear
gen wave = 2
keep mergeid wave areabldgi
save "C:\SHARE\data\area_w2.dta", replace

use  "C:\SHARE\data\sharew4_rel7-0-0_gv_housing.dta", clear
gen wave = 4
keep mergeid wave areabldgi
save "C:\SHARE\data\area_w4.dta", replace

use  "C:\SHARE\data\sharew5_rel7-0-0_gv_housing.dta", clear
gen wave = 5
keep mergeid wave areabldgi
save "C:\SHARE\data\area_w5.dta", replace

use  "C:\SHARE\data\sharew6_rel7-0-0_gv_housing.dta", clear
gen wave = 6
keep mergeid wave areabldgi
save "C:\SHARE\data\area_w6.dta", replace

use  "C:\SHARE\data\sharew7_rel7-0-0_gv_housing.dta", clear
gen wave = 7
keep mergeid wave areabldgi
save "C:\SHARE\data\area_w7.dta", replace
```

Afterwards the stored wave-specific “area datasets” are appended and the new dataset is saved.

```
use  "C:\SHARE\data\area_w1.dta", clear
append using "C:\SHARE\data\area_w2.dta"
append using "C:\SHARE\data\area_w4.dta"
append using "C:\SHARE\data\area_w5.dta"
append using "C:\SHARE\data\area_w6.dta"
append using "C:\SHARE\data\area_w7.dta"
save "C:\SHARE\data\area_appended.dta", replace
```

Then the *easy*SHARE dataset and the appended area datafile are merged using the person identifier `mergeid` and the `wave` variable.

```
use  "C:\SHARE\data\easySHARE_rel7-0-0.dta", clear
merge 1:1 mergeid wave using "C:\SHARE\data\area_appended.dta"
```

In a final step the *easy*SHARE missing code scheme is assigned to the newly included variable `areabldgi` and the extended *easy*SHARE dataset is saved.

```
replace areabldgi = -13 if wave  == 3
replace areabldgi = -12 if areabldgi == -1 | areabldgi == -2
replace areabldgi = -15 if areabldgi ==.

lab def areabldgi -15 "no information"               ///
                  -13 "not asked in this wave"       ///
                  -12 "don´t know / refusal", add

numlabel areabldgi, add

save "C:\SHARE\data\easySHARE_area.dta", replace
```

When integrating new variables into the *easy*SHARE dataset, users should be aware of possible deviations across waves like deviating question texts or response options. A cross-wave comparison of both the regular SHARE items and the SHARELIFE items of waves 3 and 7 is available here.

7. Summary

The simplified *easy*SHARE dataset contains all respondents of the so far released seven waves of data collection but a restricted subset of variables. It is stored as panel dataset in long format covering all SHARE countries and hence is tailor-made for teaching longitudinal as well as country-comparative analyses.

This working paper aims to contribute to a better understanding and handling of the *easy*SHARE dataset by explaining data structure, content as well as the guidelines that the selection of variables is based on. Additionally, the Stata program ("do-file") used to generate *easy*SHARE is illustrated. In the process of generating the dataset various assumptions and decisions had to be made that users should be aware of. For many analyses these assumptions might be adequate. For others these procedures might be less appropriate. Finally, the last section of this paper shows how users can add further SHARE variables to the *easy*SHARE dataset using the example of respondents´ area of residence.

A full list of all variables included in the dataset is stored in the *easy*SHARE release guide (Gruber 2019). The release guide describes in detail all variables and all implemented modifications. Additionally, it provides an exemplary analysis using the statistical software packages Stata, SPSS and R. Before starting to work with the *easy*SHARE dataset a study of the *easy*SHARE release guide is highly recommended.

References

Banks, J., J. Nazroo, and A. Steptoe (Eds.) (2012): The Dynamics of Ageing: Evidence from the English Longitudinal Study of Ageing 2002-10 (Wave 5). London: The Institute for Fiscal Studies.

Börsch-Supan, A. and H. Jürges (Eds.) (2005): The Survey of Health, Ageing and Retirement in Europe – Methodology. Mannheim: Mannheim Research Institute for the Economics of Aging (MEA).

Börsch-Supan, A., A. Brugiavini, H. Jürges, J. Mackenbach, J. Siegrist, and G. Weber (2005): Health, ageing and retirement in Europe – First results from the Survey of Health, Ageing and Retirement in Europe. Mannheim: Mannheim Research Institute for the Economics of Aging (MEA).

Börsch-Supan, A., A. Brugiavini, H. Jürges, A. Kapteyn, J. Mackenbach, J. Siegrist, and G. Weber (2008). First results from the Survey of Health, Ageing and Retirement in Europe (2004-2007). Starting the longitudinal dimension. Mannheim: Mannheim Research Institute for the Economics of Aging (MEA).

Börsch-Supan, A., M. Brandt, K. Hank, and M. Schröder (Eds.) (2011): The individual and the welfare state. Life histories in Europe. Heidelberg: Springer.

Börsch-Supan, A., M. Brandt, and M. Schröder (2013a): SHARELIFE – One century of life histories in Europe, *Advances in Life Course Research*, 18, 1: 1-5.

Börsch-Supan, A., M. Brandt, H. Litwin, and G. Weber (Eds.) (2013b): Active ageing and solidarity between generations in Europe: First results from SHARE after the economic crisis. Berlin: De Gruyter.

Gruber S. (2019): Release Guide to *easy*SHARE Release 7.0.0, Munich: Munich Center for the Economics of Aging (MEA). Avilable at: http://www.share-project.org/special-data-sets/easyshare.html

Malter, F. and A. Börsch-Supan (Eds.) (2013): SHARE Wave 4: Innovations & Methodology. Munich: Munich Center for the Economics of Aging (MEA).

National Institute on Aging (2007): Growing Older in America: The Health and Retirement Study. Washington, DC: National Institutes of Health.

Schröder, M. (2011): Retrospective Data Collection in the Survey of Health, Ageing and Retirement in Europe. SHARELIFE Methodology. Mannheim: Mannheim Research Institute for the Economics of Aging (MEA).

Appendix: Stata do-file generating *easy*SHARE release 7.0.0

```
/******************************************************************************

                                    ##
                             ##    ###   ###
                            ##           ##   #
                   ####                     ###

             ##
           ####
                       #######   ###    ###       ###     ########    #######
                      ####  #    ###    ###      #####    #### ####   ####
     ###              ####       ###### ###     ### ###   ###   ###   ###
    ###                ######    ##########     ##  ###   ########    #######
                          ####   ###    ###    #########  #### ###    ###
   #                   ##  ###   ###    ###   ########### ###   ###   #######
####                  #######    ###    ###   ###    ###  ###   ###   #######
#

 ###                    #                              ___ _  _   _   ___ ___
###                   #####         ___ __ _ ____  _  / __| || | /_\ | _ \ __|
     ###      ###     ###          / -_) _` (_-< || | \__ \ __ |/ _ \|   / _|
   ####     ####                   \___\__,_/__/\_, | |___/_||_/_/ \_\_|_\___|
   #        ##                                  |__/

                                              R E L E A S E  7.0.0

                - All countries & various modules of SHARE waves 1,2,3,4,5,6,7 -


Author: Stefan Gruber
Date:   March, 2019

This Stata program generates easySHARE based on the main release of SHARE,
it is based on the data of:

wave 1  release 7.0.0
wave 2  release 7.0.0
wave 3  release 7.0.0
wave 4  release 7.0.0
wave 5  release 7.0.0
wave 6  release 7.0.0
wave 7  release 7.0.0

******************************************************************************/

*----[ O v e r v i e w  o f  C o n t e n t s ]---------------------------------

*----[ 0. Stata Version & Settings]--------------------------------------------

*----[ 1. Define paths and open log file]--------------------------------------

*----[ 2. Define w, m, c_* globals]--------------------------------------------

*----[ 3. Copy Main SHARE Release to easySHARE directory]----------------------

*----[ 4. Extract & Recode Variables from cv_r]--------------------------------
*----[ 5. Extract & Recode Variables from DN / ST / RP]------------------------
```

```
*----[ 6. Extract & Recode Variables from AC ]--------------------------------
*----[ 7. Extract & Recode Variables from BR ]--------------------------------
*----[ 8. Extract & Recode Variables from CF]--------------------------------
*----[ 9. Extract & Recode Variables from CH ]--------------------------------
*----[10. Extract & Recode Variables from CO ]--------------------------------
*----[11. Extract & Recode Variables from CS ]--------------------------------
*----[12. Extract & Recode Variables from EP ]--------------------------------
*----[13. Extract & Recode Variables from GS ]--------------------------------
*----[14. Extract & Recode Variables from HC ]--------------------------------
*----[15. Extract & Recode Variables from HO ]--------------------------------
*----[16. Extract & Recode Variables from IV ]--------------------------------
*----[17. Extract & Recode Variables from MC ]--------------------------------
*----[18. Extract & Recode Variables from PH ]--------------------------------
*----[19. Extract & Recode Variables from SN ]--------------------------------
*----[20. Extract & Recode Variables from SP ]--------------------------------
*----[21. Extract & Recode Variables from HS ]--------------------------------
*----[22. Extract & Recode Variables from RH ]--------------------------------
*----[23. Extract & Recode Variables from GV_Health]--------------------------
*----[24. Extract & Recode Variables from GV_ISCED ]--------------------------
*----[25. Extract & Recode Variables from GV_Imputations]---------------------
*----[26. Extract & Recode Variables from GV_gross_net]-----------------
*----[27. Extract & Recode Variables from DROPOFF]----------------------------
*----[28. Extract & Recode Variables from Technical_Variables]-----------------

*----[29. Merge modules per wave ]--------------------------------------------

*----[30. Other recodes per wave ]--------------------------------------------

*----[31. Append waves to panel long format & integrate "long" variables]------

*----[32. Fix date intv., year/month birth, gender, & partnervars & gen age]---

*----[33. Transfer information collected once (in baseline interviews)]--------

*----[34. Pass on information to next wave that may have changed/not changed]--

*----[35. Fix & re-generate variables, labels, etc.]--------------------------

*----[36. Implement/complete wave/country skip patterns]----------------------

*----[37. Integrate DK/RF and implement no information missing code]-----------

*----[38. Keep, add easy missing codes & labels, order, data labels & save]----


*******************************************************************************/

*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[ 0. Stata Version & Settings]-------------------------------------------

version 14
clear
clear matrix
set more off
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 1. Define paths and open log file]---------------------------------------

*>> Define location of waves 1, 2, 3/SHARELIFE, 4, 5, 6 and 7 (release 7.0.0)

   global wave1 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w1"
   global wave2 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w2"
   global wave3 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w3"
   global wave4 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w4"
   global wave5 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w5"
   global wave6 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w6"
   global wave7 "R:\waveX\waveX_vers_Q_RELEASE\data\scrambled_final_Stata\w7"

   global relw1 "rel7-0-0"
   global relw2 "rel7-0-0"
   global relw3 "rel7-0-0"
   global relw4 "rel7-0-0"
   global relw5 "rel7-0-0"
   global relw6 "rel7-0-0"
   global relw7 "rel7-0-0"

*>> Define location of easySHARE directory (here the new dataset is generated)

    global easy "R:\DataCleaning\waveX\waveX_vers_Q_easySHARE"

*>> Generate data folders within the pre-existing $easy directory:

    cd $easy                        // change to directory stored above
    capture mkdir "log"             // log  folder
    capture mkdir "data"            // data folder
    capture mkdir "data/release"    // to store a copy of the ingoing datasets
    capture mkdir "data/temp"       // to store temporary data versions

*>> Name and open a log file and store in the log directory

    capture log close
    local h =  substr("`c(current_time)'",1,2) // These commands
    local m =  substr("`c(current_time)'",4,2) // are only to
    local s =  substr("`c(current_time)'",7,2) // automatically
    local d = "`c(current_date)'"              // generate the
    local u = "`c(username)'"                  // name of the log file

    log using "$easy/log/EasySHARE_LOG_`u'__`d'_`h'-`m'-`s'.log", replace

*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 2. Define w, m, c_* globals]---------------------------------------------

// The following globals are mostly used to copy the main release data into the
// easySHARE directory. It is not really necessary to do this from within Stata
// using the rather complex loop commands. However, this is how we do it.
// In addition, these lists provide a good overview on countries, waves &
// modules.
```

```
*>> Define what waves and modules to use

    global w "1 2 3 4 5 6 7"
    global m "ac as br cv_r cf ch co cs cc dn ep gs hc hh ho hs mc mh ph rh rp ///
              sn sp st xt gv_health gv_isced gv_isco iv gv_imputations dropoff ///
              technical_variables gv_big5"

*>> Check global lists - do never change these

    global c_w1  "at be_fr be_nl ch de dk es fr gr it il nl se"
    global c_w2  "at be_fr be_nl ch cz de dk es fr gr it il ie nl pl se"
    global c_w3  "at be_fr be_nl ch cz de dk es fr gr it nl pl se"
    global c_w4  "at be_fr be_nl ch cz de dk ee es fr hu it nl pl pt se si"
    global c_w5  "at be_fr be_nl ch cz de dk ee es fr hu it il lu nl se si"
    global c_w6  "at be_fr be_nl ch cr cz de dk ee es fr hu it il lu nl se si"
    global c_w7  "at be_fr be_nl bg ch cr cy cz de dk ee es fi fr gr hu it il ///
                  li lu lv mt nl pl pt ro se si sk"


    global m_w1 "ac as br cf ch co cv_r dn ep ex ft gs hc hh ho iv mh ph sp vi ///
                 ws gv_health gv_isced gv_isco gv_imputations dropoff ///
                 technical_variables"
    global m_w2 "ac as br cf ch co cs cv_r dn ep ex ft gs hc hh ho iv mh pf ph ///
                 sp vi ws xt gv_health gv_isced gv_imputations dropoff ///
                 technical_variables"
    global m_w3 "ac cs cv_r fq fs gs gl hc hs iv ls mn rc re rp st xt"
    global m_w4 "ac as bi br cf ch co cv_r dn ep ex ft gs hc hh ho iv li mh pf ///
                 ph sn sp xt gv_health gv_isced gv_imputations dropoff ///
                 technical_variables"
    global m_w5 "ac as br cf ch co cs cv_r dn ep ex ft gs hc hh ho iv li mc mh ///
                 ph sp xt gv_health gv_isced gv_imputations dropoff ///
                 technical_variables"
    global m_w6 "ac as br cf ch co cv_r dn ep ex ft gs hc hh ho iv li mh ph sn ///
                 sp xt gv_health gv_isced gv_imputations dropoff ///
                 technical_variables"
    global m_w7 "ac as br cc cf ch co cv_r dn ep ex ft gs hc hh ho hs iv li mh ///
                 ph rh sp xt gv_health gv_isced gv_imputations dropoff ///
                 technical_variables gv_big5"

*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 3. Copy Main SHARE Release to easySHARE directory]----------------------

// Here we copy the original datasets into the easySHARE project. This is not
// very efficient in terms of storage, but we can always see which data
// versions we have used to build easySHARE, even if we accidently delete the
// data in folder we copy from.

quietly { // quitely loop starts

set more off
  noisily: di as result "[Copy main SHARE release to easySHARE directory]----"
  foreach w in $w { // wave loop starts
    noi di as result "Wave: " as txt "`w'"
  foreach m in $m { // module loop starts
    if strpos("${m_w`w'}","`m'") !=0 {    // check if module available in wave
        noisily: di as text _continue "         .. doing: mod `m', w`w'"
```

```
        copy "${wave`w'}\sharew`w'_${relw`w'}_`m'.dta" ///
             "$easy\data\release\sharew`w'_${relw`w'}_`m'.dta" , replace

        noisily: di  as text " ... DONE :-)"
    }   // end of check if module is available in wave
    else noisily: di as text   "      NOT: module `m' not in wave `w'"

  } // module loop end
  } // wave loop end


      // gv_grossnet in w1 has to be copied separately
         copy "$wave1\sharew1_rel7-0-0_gv_grossnet.dta" ///
              "$easy\data\release\sharew1_rel7-0-0_gv_grossnet.dta" , replace

}   // quietly loop end

*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[ 4. Extract & Recode Variables from cv_r]-------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_cv_r.dta, clear

    gen     wave=1
    lab var wave          "Wave"
    lab var hhid1         "Household identifier wave 1"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
    lab var int_month     "Interview month"
    lab var partnerinhh   "Partner in household"
    lab var fam_resp      "Family respondent"
    lab var hou_resp      "Household respondent"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female

    sort coupleid1
    gen     gender_partner=.
    replace gender_partner=female[_n-1] if coupleid1==coupleid1[_n-1] ///
                                         & coupleid1!=""
    replace gender_partner=female[_n+1] if coupleid1==coupleid1[_n+1] ///
                                         & coupleid1!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner: female=1, male=0"
    lab val gender_partner lblfemale
    lab def lblfemale       ///
            0 "0. male"     ///
            1 "1. female"   ///
           -9 "filtered: single or no partner in household", add

    * re-number coupleID
    bysort  hhid1: gen temp =_n
    gsort -coupleid1 -temp

    drop if interview == 0 // we only keep persons with interview for easySHARE
```

```
    * interview of partner available
    gen     int_partner=.
    replace int_partner=-9 if coupleid1==""
    replace int_partner=1  ///
               if (coupleid1==coupleid1[_n+1] | coupleid1==coupleid1[_n-1]) ///
               &  coupleid1!=""
    replace int_partner=5  ///
               if  coupleid1!=coupleid1[_n+1] & coupleid1!=coupleid1[_n-1]  ///
               &  coupleid1!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes" ///
                          5 "5. no"  ///
                         -9 "filtered: single or no partner in household", add

    keep mergeid mergeidp coupleid1 hhid1 wave female partnerinhh hhsize ///
         int_year int_month fam_resp hou_resp gender_partner  ///
         mobirthp yrbirthp int_partner

save $easy\data\temp\sharew1_cv_r_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_cv_r.dta, clear

    gen wave=2
    lab var wave          "Wave"
    lab var hhid2         "Household identifier wave 2"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
    lab var int_month     "Interview month"
    lab var partnerinhh   "Partner in household"
    lab var fam_resp      "Family respondent"
    lab var hou_resp      "Household respondent"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female

    sort coupleid2
    gen     gender_partner=.
    replace gender_partner=female[_n-1] if coupleid2==coupleid2[_n-1] ///
                                        & coupleid2!=""
    replace gender_partner=female[_n+1] if coupleid2==coupleid2[_n+1] ///
                                        & coupleid2!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner"
    lab val gender_partner lblfemale
    lab def lblfemale -9 "filtered: single or no partner in hh", add

    bysort  hhid2: gen temp =_n
    gsort -coupleid2 -temp

    drop if interview == 0 // we only keep respondents with interview
    drop if deceased == 1 // we only keep alive respondents for easySHARE

    gen int_partner=.
    replace int_partner=-9 if  coupleid2==""
```

```
    replace int_partner=1  ///
            if (coupleid2==coupleid2[_n+1] | coupleid2==coupleid2[_n-1]) ///
             &  coupleid2!=""
    replace int_partner=5  ///
            if  coupleid2!=coupleid2[_n+1] & coupleid2!=coupleid2[_n-1]  ///
             &  coupleid2!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes" ///
                          5 "5. no"  ///
                         -9 "filtered: single or no partner in household", add

    keep mergeid mergeid wave coupleid2 hhid2 female partnerinhh hhsize ///
         int_year int_month fam_resp hou_resp gender_partner ///
         mobirthp yrbirthp int_partner

save $easy\data\temp\sharew2_cv_r_a.dta, replace

*>> w3:

use $easy\data\release\sharew3_rel7-0-0_cv_r.dta, clear

    gen wave=3
    lab var wave          "Wave"
    lab var hhid3         "Household identifier wave 3"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
    lab var int_month     "Interview month"
    lab var partnerinhh   "Partner in household"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female
    *rename mstat partnerinhh

    sort coupleid3
    gen     gender_partner=.
    replace gender_partner=female[_n-1] if coupleid3==coupleid3[_n-1] ///
                                        & coupleid3!=""
    replace gender_partner=female[_n+1] if coupleid3==coupleid3[_n+1] ///
                                        & coupleid3!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner"
    lab val gender_partner lblfemale
    lab def lblfemale -9 "filtered: single or no partner in hh", add

    drop if interview == 0 // we only keep respondents with interview

    gen int_partner=.
    replace int_partner=-9 if coupleid3==""
    replace int_partner=1  ///
            if (coupleid3==coupleid3[_n+1] | coupleid3==coupleid3[_n-1]) ///
             & coupleid3!=""
    replace int_partner=5  ///
            if coupleid3!=coupleid3[_n+1] & coupleid3!=coupleid3[_n-1] ///
             & coupleid3!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes" ///
                          5 "5. no"  ///
```

```
                        -9 "filtered: single or no partner in household", add

    drop if deceased == 1 // we only keep alive respondents for easySHARE

    keep mergeid mergeidp wave coupleid3 hhid3 female partnerinhh hhsize ///
         int_year int_month gender_partner ///
         mobirthp yrbirthp int_partner

save $easy\data\temp\sharew3_cv_r_a.dta, replace

*>> w4:

use $easy\data\release\sharew4_rel7-0-0_cv_r.dta, clear

    gen wave=4
    lab var wave          "Wave"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
    lab var int_month     "Interview month"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female

    sort coupleid4
    gen     gender_partner=.
    replace gender_partner=female[_n-1] if coupleid4==coupleid4[_n-1] ///
                                        & coupleid4!=""
    replace gender_partner=female[_n+1] if coupleid4==coupleid4[_n+1] ///
                                        & coupleid4!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner: female=1, male=0"
    lab val gender_partner lblfemale
    lab def lblfemale -9 "filtered: single or no partner in hh", add

    bysort  hhid4: gen temp =_n
    gsort -coupleid4 -temp

    drop if interview == 0 // we only keep respondents with interview
    drop if deceased == 1 // we only keep alive respondents for easySHARE

    gen     int_partner=.
    replace int_partner=-9 if coupleid4==""
    replace int_partner=1 ///
            if (coupleid4==coupleid4[_n+1] | coupleid4==coupleid4[_n-1]) ///
             & coupleid4!=""
    replace int_partner=5  ///
            if coupleid4!=coupleid4[_n+1] & coupleid4!=coupleid4[_n-1] ///
             & coupleid4!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes"  ///
                          5 "5. no"   ///
                         -9 "filtered: single or no partner in household", add

    keep    mergeid mergeidp wave waveid coupleid4 hhid4 female partnerinhh ///
            hhsize int_year int_month fam_resp hou_resp gender_partner ///
            mobirthp yrbirthp int_partner

save $easy\data\temp\sharew4_cv_r_a.dta, replace
```

```
*>> w5:

use $easy\data\release\sharew5_rel7-0-0_cv_r.dta, clear

    gen wave=5
    lab var wave          "Wave"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
    lab var int_month     "Interview month"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female

    sort coupleid5
    gen     gender_partner=.
    replace gender_partner=female[_n-1] if coupleid5==coupleid5[_n-1] ///
                                        & coupleid5!=""
    replace gender_partner=female[_n+1] if coupleid5==coupleid5[_n+1] ///
                                        & coupleid5!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner: female=1, male=0"
    lab val gender_partner lblfemale
    lab def lblfemale -9 "filtered: single or no partner in hh", add

    bysort  hhid5: gen temp =_n
    gsort -coupleid5 -temp

    drop if interview == 0 // we only keep respondents with interview
    drop if deceased == 1 // we only keep alive respondents for easySHARE

    gen     int_partner=.
    replace int_partner=-9 if coupleid5==""
    replace int_partner=1 ///
            if (coupleid5==coupleid5[_n+1] | coupleid5==coupleid5[_n-1]) ///
             & coupleid5!=""
    replace int_partner=5 ///
            if coupleid5!=coupleid5[_n+1] & coupleid5!=coupleid5[_n-1] ///
             & coupleid5!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes"  ///
                          5 "5. no"   ///
                         -9 "filtered: single or no partner in household", add

     keep   mergeid mergeidp wave waveid coupleid5 hhid5 female partnerinhh ///
            hhsize int_year int_month fam_resp hou_resp gender_partner ///
            mobirthp yrbirthp int_partner

save $easy\data\temp\sharew5_cv_r_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_cv_r.dta, clear

    gen wave=6
    lab var wave          "Wave"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
```

```
    lab var int_month     "Interview month"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female

    sort coupleid6
    gen     gender_partner=.
    replace gender_partner=female[_n-1] if coupleid6==coupleid6[_n-1] ///
                                        & coupleid6!=""
    replace gender_partner=female[_n+1] if coupleid6==coupleid6[_n+1] ///
                                        & coupleid6!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner: female=1, male=0"
    lab val gender_partner lblfemale
    lab def lblfemale -9 "filtered: single or no partner in hh", add

    bysort  hhid6: gen temp =_n
    gsort -coupleid6 -temp

    drop if interview == 0 // we only keep respondents with interview
    drop if deceased == 1 // we only keep alive respondents for easySHARE

    gen     int_partner=.
    replace int_partner=-9 if coupleid6==""
    replace int_partner=1 ///
            if (coupleid6==coupleid6[_n+1] | coupleid6==coupleid6[_n-1]) ///
             & coupleid6!=""
    replace int_partner=5 ///
            if coupleid6!=coupleid6[_n+1] & coupleid6!=coupleid6[_n-1] ///
             & coupleid6!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes"  ///
                          5 "5. no"   ///
                         -9 "filtered: single or no partner in household", add

    keep    mergeid mergeidp wave waveid coupleid6 hhid6 female partnerinhh ///
            hhsize int_year int_month fam_resp hou_resp gender_partner ///
            mobirthp yrbirthp int_partner

save $easy\data\temp\sharew6_cv_r_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_cv_r.dta, clear

    gen wave=7
    lab var wave          "Wave"
    lab var hhsize        "Household size"
    lab var int_year      "Interview year"
    lab var int_month     "Interview month"
    recode gender 2=1 1=0, gen(female)
    lab var female "Gender: female=1, male=0"
    lab def female 1 "female" 0 "male"
    lab val female female

    sort coupleid7
    gen     gender_partner=.
```

```
    replace gender_partner=female[_n-1] if coupleid7==coupleid7[_n-1] ///
                                        & coupleid7!=""
    replace gender_partner=female[_n+1] if coupleid7==coupleid7[_n+1] ///
                                        & coupleid7!=""
    replace gender_partner=-9 if partnerinhh==3 | partnerinhh==97
    lab var gender_partner "Gender of partner: female=1, male=0"
    lab val gender_partner lblfemale
    lab def lblfemale -9 "filtered: single or no partner in hh", add

    bysort  hhid7: gen temp =_n
    gsort -coupleid7 -temp

    drop if interview == 0 // we only keep respondents with interview
    drop if deceased == 1 // we only keep alive respondents for easySHARE

    gen     int_partner=.
    replace int_partner=-9 if coupleid7==""
    replace int_partner=1 ///
            if (coupleid7==coupleid7[_n+1] | coupleid7==coupleid7[_n-1]) ///
             & coupleid7!=""
    replace int_partner=5 ///
            if coupleid7!=coupleid7[_n+1] & coupleid7!=coupleid7[_n-1] ///
             & coupleid7!=""
    lab var int_partner "Interview of partner available"
    lab val int_partner lblinterviewp
    lab def lblinterviewp 1 "1. yes"  ///
                          5 "5. no"   ///
                         -9 "filtered: single or no partner in household", add

     keep   mergeid mergeidp wave waveid coupleid7 hhid7 female partnerinhh ///
            hhsize int_year int_month fam_resp hou_resp gender_partner ///
            mobirthp yrbirthp int_partner

save $easy\data\temp\sharew7_cv_r_a.dta, replace


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 5. Extract & Recode Variables from DN / ST / RP ]-----------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_dn.dta, clear

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
    replace country_mod = 380 if country==16
    replace country_mod = 250 if country==17
    replace country_mod = 208 if country==18
    replace country_mod = 300 if country==19
    replace country_mod = 756 if country==20
    replace country_mod =  56 if country==23
    replace country_mod = 376 if country==25
```

```
    lab def country_mod  40 "Austria"       276 "Germany"   752 "Sweden"    ///
                        528 "Netherlands"   724 "Spain"     380 "Italy"     ///
                        250 "France"        208 "Denmark"   300 "Greece"    ///
                        756 "Switzerland"    56 "Belgium"   376 "Israel"    ///
                        203 "Czechia"       616 "Poland"    372 "Ireland"   ///
                        348 "Hungary"       620 "Portugal"  705 "Slovenia"  ///
                        233 "Estonia"       442 "Luxembourg" 191 "Croatia"
    lab val country_mod country_mod
    lab var country_mod "Country identifier (ISO coded)"

  * birth_country - filled for born in country and codes adapted to ISO-codes
    gen     birth_country = dn005c   if dn004==5
    replace birth_country = country_mod if dn004==1
    lab var birth_country "Country of birth (ISO coded)"
    // value labels will be taken from wave 4

  * Citizenship - filled for country citizenship and codes adapted to ISO-codes
    gen     citizenship = dn008c   if dn007==5
    replace citizenship = country_mod if dn007==1
    lab var citizenship "Citizenship of respondent (ISO coded)"
    // value labels will be taken from wave 4

  * Siblings_alive
    gen     siblings_alive = dn037_ + dn036_ if dn036_>=0 & dn037>=0
    replace siblings_alive = -9 if dn034_==5
    lab var siblings_alive ///
            "Number of siblings alive (based on: dn036_,dn037_,dn034_)"
    lab def siblings_alive -9 "filtered: no siblings ever", add
    lab val siblings_alive siblings_alive

  keep  mergeid country country_mod language dn002_ dn003_ dn004_ dn006_ ///
        birth_country citizenship dn014_ siblings_alive dn026_1 dn026_2

save $easy\data\temp\sharew1_dn_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_dn.dta, clear

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
    replace country_mod = 380 if country==16
    replace country_mod = 250 if country==17
    replace country_mod = 208 if country==18
    replace country_mod = 300 if country==19
    replace country_mod = 756 if country==20
    replace country_mod =  56 if country==23
    replace country_mod = 203 if country==28
    replace country_mod = 616 if country==29
    replace country_mod = 372 if country==30
    replace country_mod = 376 if country==25
```

```
    lab def country_mod  40 "Austria"       276 "Germany"   752 "Sweden"    ///
                        528 "Netherlands"   724 "Spain"     380 "Italy"     ///
                        250 "France"        208 "Denmark"   300 "Greece"    ///
                        756 "Switzerland"    56 "Belgium"   376 "Israel"    ///
                        203 "Czechia"       616 "Poland"    372 "Ireland"   ///
                        348 "Hungary"       620 "Portugal"  705 "Slovenia"  ///
                        233 "Estonia"       442 "Luxembourg" 191 "Croatia"
    lab val country_mod country_mod
    lab var country_mod "Country identifier (ISO coded)"

  * birth_country - filled for born in country and codes adapted to ISO-codes
    gen     birth_country = dn005c   if dn004==5
    replace birth_country = country_mod if dn004==1
    lab var birth_country "Country of birth (ISO coded)"
    // value labels will be taken from wave 4

  * Citizenship - filled for country citizenship and codes adapted to ISO-codes
    gen     citizenship = dn008c   if dn007==5
    replace citizenship = country_mod if dn007==1
    lab var citizenship "Citizenship of respondent (ISO coded)"
    // value labels will be taken from wave 4

  * Siblings
    gen     siblings_alive = dn037_ + dn036_ if dn036_>=0 & dn037>=0
    replace siblings_alive =-9 if dn034_==5
    lab var siblings_alive ///
            "Number of siblings alive (based on: dn036_, dn037_, dn034_)"
    lab def siblings_alive -9 "filtered: no siblings ever", add
    lab val siblings_alive siblings_alive

  * Years of education:
    rename dn041_ eduyears

    keep mergeid country country_mod language dn002_ dn003_ dn004_ dn006_ ///
         birth_country citizenship dn014_ siblings_alive dn026_1 dn026_2 ///
         dn044_ dn043_ eduyears

save $easy\data\temp\sharew2_dn_a.dta, replace


*>> w3:
// information on country and birthdate based on ST module
// information on partnership taken from RP module

// ST Module:
use $easy\data\release\sharew3_rel7-0-0_st.dta, clear
    rename sl_st006_ dn002_
    rename sl_st007_ dn003_

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
    replace country_mod = 380 if country==16
    replace country_mod = 250 if country==17
    replace country_mod = 208 if country==18
    replace country_mod = 300 if country==19
```

```
    replace country_mod = 756 if country==20
    replace country_mod =  56 if country==23
    replace country_mod = 203 if country==28
    replace country_mod = 616 if country==29
    replace country_mod = 372 if country==30

    lab def country_mod  40 "Austria"       276 "Germany"    752 "Sweden"    ///
                        528 "Netherlands"   724 "Spain"      380 "Italy"     ///
                        250 "France"        208 "Denmark"    300 "Greece"    ///
                        756 "Switzerland"    56 "Belgium"    376 "Israel"    ///
                        203 "Czechia"       616 "Poland"     372 "Ireland"   ///
                        348 "Hungary"       620 "Portugal"   705 "Slovenia" ///
                        233 "Estonia"       442 "Luxembourg" 191 "Croatia"
    lab val country_mod country_mod
    lab var country_mod "Country identifier (ISO coded)"

    keep mergeid hhid3 country country_mod language dn002_ dn003_ sl_st011_
          // sl_st011_ taken to replace  missing lines/info in cv_r file

save $easy\data\temp\sharew3_st_a.dta, replace

// RP Module
use $easy\data\release\sharew3_rel7-0-0_rp.dta, clear

             * Decease of partner between w2 and w3
             egen pdeath_last = rowmax(sl_rp011_*)
             label var pdeath_last "Year of widowdhood - last (from sharelife)"

             * Divorce between w2 and w3
             egen pdivorce_last = rowmax(sl_rp014_*)
             label var pdivorce_last "Year of divorce - last (from sharelife)"

    keep mergeid pdeath_last pdivorce_last

save $easy\data\temp\sharew3_rp_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_dn.dta, clear

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
    replace country_mod = 380 if country==16
    replace country_mod = 250 if country==17
    replace country_mod = 208 if country==18
    replace country_mod = 300 if country==19
    replace country_mod = 756 if country==20
    replace country_mod =  56 if country==23
    replace country_mod = 203 if country==28
    replace country_mod = 616 if country==29
    replace country_mod = 348 if country==32
    replace country_mod = 620 if country==33
    replace country_mod = 705 if country==34
    replace country_mod = 233 if country==35
```

```
    lab def country_mod  40 "Austria"       276 "Germany"   752 "Sweden"   ///
                        528 "Netherlands"   724 "Spain"     380 "Italy"    ///
                        250 "France"        208 "Denmark"   300 "Greece"   ///
                        756 "Switzerland"    56 "Belgium"   376 "Israel"   ///
                        203 "Czechia"       616 "Poland"    372 "Ireland"  ///
                        348 "Hungary"       620 "Portugal"  705 "Slovenia" ///
                        233 "Estonia"       442 "Luxembourg" 191 "Croatia"
    lab val country_mod country_mod
    lab var country_mod "Country identifier (ISO coded)"

  * birth_country - filled for born in country and codes adapted to ISO-codes
    gen     birth_country = dn005c   if dn004==5
    replace birth_country = country_mod if dn004==1
    lab var birth_country "Country of birth (ISO coded)"

  * Citizenship - filled for country citizenship and codes adapted to ISO-codes
    gen     citizenship = dn008c   if dn007==5
    replace citizenship = country_mod if dn007==1
    lab var citizenship "Citizenship of respondent (ISO coded)"

  * Siblings
    gen     siblings_alive= dn037_ + dn036_ if dn036_>=0 & dn037>=0
    replace siblings_alive=-9 if dn034_==5
    lab var siblings_alive ///
            "Number of siblings alive (based on: dn036_, dn037_, dn034_)"
    lab def siblings_alive -9 "filtered: no siblings ever", add
    lab val siblings_alive siblings_alive

    rename dn041_ eduyears

    keep mergeid country country_mod language dn002_ dn003_ dn004_ dn006_ ///
         birth_country citizenship dn014_ siblings_alive dn026_1 dn026_2 ///
         dn044_ dn043_ eduyears dn005c dn008c
         // dn005c dn008c are kept for their value labels

save $easy\data\temp\sharew4_dn_a.dta, replace

*>> w5:

use $easy\data\release\sharew5_rel7-0-0_dn.dta, clear

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
    replace country_mod = 380 if country==16
    replace country_mod = 250 if country==17
    replace country_mod = 208 if country==18
    replace country_mod = 756 if country==20
    replace country_mod =  56 if country==23
    replace country_mod = 376 if country==25
    replace country_mod = 203 if country==28
    replace country_mod = 616 if country==29
    replace country_mod = 442 if country==31
    replace country_mod = 705 if country==34
    replace country_mod = 233 if country==35
```

```
    lab def country_mod  40 "Austria"       276 "Germany"   752 "Sweden"   ///
                        528 "Netherlands"   724 "Spain"     380 "Italy"    ///
                        250 "France"        208 "Denmark"   300 "Greece"   ///
                        756 "Switzerland"    56 "Belgium"   376 "Israel"   ///
                        203 "Czechia"       616 "Poland"    372 "Ireland"  ///
                        348 "Hungary"       620 "Portugal"  705 "Slovenia" ///
                        233 "Estonia"       442 "Luxembourg" 191 "Croatia"
    lab val country_mod country_mod
    lab var country_mod "Country identifier (ISO coded)"

  * birth_country - filled for born in country and codes adapted to ISO-codes
    gen     birth_country = dn005c   if dn004==5
    replace birth_country = country_mod if dn004==1
    lab var birth_country "Country of birth (ISO coded)"

  * Citizenship - filled for country citizenship and codes adapted to ISO-codes
    gen     citizenship = dn008c   if dn007==5
    replace citizenship = country_mod if dn007==1
    lab var citizenship "Citizenship of respondent (ISO coded)"

  * Siblings
    gen     siblings_alive= dn037_ + dn036_ if dn036_>=0 & dn037>=0
    replace siblings_alive=-9 if dn034_==5
    lab var siblings_alive ///
            "Number of siblings alive (based on: dn036_, dn037_, dn034_)"
    lab def siblings_alive -9 "filtered: no siblings ever", add
    lab val siblings_alive siblings_alive

    rename dn041_ eduyears

    keep mergeid country country_mod language dn002_ dn003_ dn004_ dn006_ ///
         birth_country citizenship dn014_ siblings_alive dn026_1 dn026_2 ///
         dn044_ dn043_ eduyears dn005c dn008c
         // dn005c dn008c are kept for their value labels

save $easy\data\temp\sharew5_dn_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_dn.dta, clear

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
    replace country_mod = 380 if country==16
    replace country_mod = 250 if country==17
    replace country_mod = 208 if country==18
    replace country_mod = 300 if country==19
    replace country_mod = 756 if country==20
    replace country_mod =  56 if country==23
    replace country_mod = 376 if country==25
    replace country_mod = 203 if country==28
    replace country_mod = 616 if country==29
    replace country_mod = 442 if country==31
```

```
    replace country_mod = 620 if country==33
    replace country_mod = 705 if country==34
    replace country_mod = 233 if country==35
    replace country_mod = 191 if country==47

    lab def country_mod  40 "Austria"       276 "Germany"   752 "Sweden"   ///
                        528 "Netherlands"   724 "Spain"     380 "Italy"    ///
                        250 "France"        208 "Denmark"   300 "Greece"   ///
                        756 "Switzerland"    56 "Belgium"   376 "Israel"   ///
                        203 "Czechia"       616 "Poland"    372 "Ireland"  ///
                        348 "Hungary"       620 "Portugal"  705 "Slovenia" ///
                        233 "Estonia"         442 "Luxembourg" 191 "Croatia"
    lab val country_mod country_mod
    lab var country_mod "country identifier (ISO coded)"

  * birth_country - filled for born in country and codes adapted to ISO-codes
    gen     birth_country = dn005c   if dn004==5
    replace birth_country = country_mod if dn004==1
    lab var birth_country "country of birth (ISO coded)"

  * Citizenship - filled for country citizenship and codes adapted to ISO-codes
    gen     citizenship = dn008c   if dn007==5
    replace citizenship = country_mod if dn007==1
    lab var citizenship "citizenship of respondent (ISO coded)"

  * Siblings
    gen     siblings_alive= dn037_ + dn036_ if dn036_>=0 & dn037>=0
    replace siblings_alive=-9 if dn034_==5
    lab var siblings_alive ///
            "number of siblings alive (based on: dn036_, dn037_, dn034_)"
    lab def siblings_alive -9 "filtered: no siblings ever", add
    lab val siblings_alive siblings_alive

    rename dn041_ eduyears

      gen parent_500km = .
      replace parent_500km = 0 if dn030_1 <8
      replace parent_500km = 0 if dn030_2 <8
      replace parent_500km = 1 if dn030_1 >=8 & dn030_1 <=9
      replace parent_500km = 1 if dn030_2 >=8 & dn030_2 <=9

    keep mergeid country country_mod language dn002_ dn003_ dn004_ dn006_ ///
         dn007_ dn502 dn503_ birth_country citizenship dn014_ siblings_alive ///
         dn026_1 dn026_2 dn044_ dn043_ eduyears dn005c dn008c
         // dn005c dn008c are kept for their value labels

save $easy\data\temp\sharew6_dn_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_dn.dta, clear

  * country_mod - codes adapted to ISO-codes
    gen     country_mod =   .
    replace country_mod =  40 if country==11
    replace country_mod = 276 if country==12
    replace country_mod = 752 if country==13
    replace country_mod = 528 if country==14
    replace country_mod = 724 if country==15
```

```
  replace country_mod = 380 if country==16
  replace country_mod = 250 if country==17
  replace country_mod = 208 if country==18
  replace country_mod = 300 if country==19
  replace country_mod = 756 if country==20
  replace country_mod =  56 if country==23
  replace country_mod = 376 if country==25
  replace country_mod = 203 if country==28
  replace country_mod = 616 if country==29
  replace country_mod = 442 if country==31
  replace country_mod = 348 if country==32
  replace country_mod = 620 if country==33
  replace country_mod = 705 if country==34
  replace country_mod = 233 if country==35
  replace country_mod = 191 if country==47
  replace country_mod = 440 if country==48
  replace country_mod = 100 if country==51
  replace country_mod = 196 if country==53
  replace country_mod = 246 if country==55
  replace country_mod = 428 if country==57
  replace country_mod = 470 if country==59
  replace country_mod = 642 if country==61
  replace country_mod = 703 if country==63

  lab def country_mod7 40 "Austria"       276 "Germany"    752 "Sweden"   ///
                       528 "Netherlands"   724 "Spain"      380 "Italy"    ///
                       250 "France"        208 "Denmark"    300 "Greece"   ///
                       756 "Switzerland"    56 "Belgium"    376 "Israel"   ///
                       203 "Czechia"       616 "Poland"     372 "Ireland"  ///
                       348 "Hungary"       620 "Portugal"   705 "Slovenia" ///
                       233 "Estonia"       442 "Luxembourg" 191 "Croatia"  ///
                       440 "Lithuania"     100 "Bulgaria"   196 "Cyprus" ///
                       246 "Finland"       428 "Latvia"     470 "Malta" ///
                       642 "Romania"       703 "Slovakia"
  lab val country_mod country_mod7
  lab var country_mod "country identifier (ISO coded)"

* birth_country - filled for born in country and codes adapted to ISO-codes
  gen     birth_country = dn005c   if dn004==5
  replace birth_country = country_mod if dn004==1
  lab var birth_country "country of birth (ISO coded)"

* Citizenship - filled for country citizenship and codes adapted to ISO-codes
  gen     citizenship = dn008c   if dn007==5
  replace citizenship = country_mod if dn007==1
  lab var citizenship "citizenship of respondent (ISO coded)"

* Siblings
  gen     siblings_alive= dn037_ + dn036_ if dn036_>=0 & dn037>=0
  replace siblings_alive=-9 if dn034_==5
  lab var siblings_alive ///
          "number of siblings alive (based on: dn036_, dn037_, dn034_)"
  lab def siblings_alive -9 "filtered: no siblings ever", add
  lab val siblings_alive siblings_alive

  rename dn041_ eduyears

    gen parent_500km = .
    replace parent_500km = 0 if dn030_1 <8
    replace parent_500km = 0 if dn030_2 <8
```

```
      replace parent_500km = 1 if dn030_1 >=8 & dn030_1 <=9
      replace parent_500km = 1 if dn030_2 >=8 & dn030_2 <=9

    keep mergeid country country_mod language dn002_ dn003_ dn004_ dn006_ ///
         dn007_ dn502 dn503_ birth_country citizenship dn014_ siblings_alive ///
         dn026_1 dn026_2 dn044_ dn043_ eduyears dn005c dn008c
         // dn005c dn008c are kept for their value labels

save $easy\data\temp\sharew7_dn_a.dta, replace


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 6. Extract & Recode Variables from AC ]---------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_ac.dta, clear
    keep mergeid ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6 ac002d7 ///
         ac002dno
save $easy\data\temp\sharew1_ac_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_ac.dta, clear

    mvdecode ac02* ac03* ac01*, mv(-1=.a \ -2=.b)

  * Quality of Life Score (CASP-12)
    // inverse into  *_inv (high value / high control etc.)
    recode ac017_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac017_inv)
    recode ac020_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac020_inv)
    recode ac021_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac021_inv)
    recode ac022_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac022_inv)
    recode ac023_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac023_inv)
    recode ac024_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac024_inv)
    recode ac025_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac025_inv)

    // Subscale Control
    gen     con= ac014_ + ac015_ + ac016_
    alpha        ac014_   ac015_   ac016_

    // Subscale Autonomy
    gen     aut= ac017_inv + ac018_ + ac019_
    alpha        ac017_inv   ac018_   ac019_

    // Subscale Pleasure
    gen     ple= ac020_inv + ac021_inv + ac022_inv
    alpha        ac020_inv   ac021_inv   ac022_inv

    // Subscale Self-Realisation
    gen     sel= ac023_inv + ac024_inv + ac025_inv
    alpha        ac023_inv   ac024_inv   ac025_inv

    // CASP
    gen     casp= con + aut + ple + sel
    lab var casp "CASP quality of life (high is high quality)"
```

```
    mvencode ac02* ac03* ac01*, mv(.a=-1 \ .b=-2)

    keep mergeid ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6 ac002d7 ///
         ac002dno casp

save $easy\data\temp\sharew2_ac_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_ac.dta, clear

    mvdecode ac02* ac03* ac01*, mv(-1=.a \ -2=.b)

  * Quality of Life Score (CASP-12)

    // inverse into  *_inv (high value / high control etc.)
    recode ac017_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac017_inv)
    recode ac020_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac020_inv)
    recode ac021_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac021_inv)
    recode ac022_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac022_inv)
    recode ac023_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac023_inv)
    recode ac024_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac024_inv)
    recode ac025_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac025_inv)

    // Subscale Control
    gen     con = ac014_ + ac015_ + ac016_
    alpha         ac014_   ac015_   ac016_

    // Subscale Autonomy
    gen aut= ac017_inv + ac018_ + ac019_
    alpha    ac017_inv   ac018_   ac019_

    // Subscale Pleasure
    gen     ple= ac020_inv + ac021_inv + ac022_inv
    alpha        ac020_inv   ac021_inv   ac022_inv

    // Subscale Self-Realisation
    gen     sel = ac023_inv + ac024_inv + ac025_inv
    alpha         ac023_inv   ac024_inv   ac025_inv

    gen     casp = con + aut + ple + sel
    lab var casp "CASP quality of life (high is high quality)"

    mvencode ac02* ac03* ac01*, mv(.a=-1 \ .b=-2)

    keep mergeid casp

save $easy\data\temp\sharew4_ac_a.dta, replace


*>> w5:

use $easy\data\release\sharew5_rel7-0-0_ac.dta, clear

    mvdecode ac02* ac03* ac01*, mv(-1=.a \ -2=.b)

  * Quality of Life Score (CASP-12)
```

```
    // inverse into  *_inv (high value / high control etc.)
    recode ac017_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac017_inv)
    recode ac020_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac020_inv)
    recode ac021_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac021_inv)
    recode ac022_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac022_inv)
    recode ac023_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac023_inv)
    recode ac024_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac024_inv)
    recode ac025_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac025_inv)

    // Subscale Control
    gen     con = ac014_ + ac015_ + ac016_
    alpha         ac014_   ac015_   ac016_

    // Subscale Autonomy
    gen aut= ac017_inv + ac018_ + ac019_
    alpha    ac017_inv   ac018_   ac019_

    // Subscale Pleasure
    gen     ple= ac020_inv + ac021_inv + ac022_inv
    alpha        ac020_inv   ac021_inv   ac022_inv

    // Subscale Self-Realisation
    gen     sel = ac023_inv + ac024_inv + ac025_inv
    alpha         ac023_inv   ac024_inv   ac025_inv

    gen     casp = con + aut + ple + sel
    lab var casp "CASP quality of life (high is high quality)"

    mvencode ac02* ac03* ac01*, mv(.a=-1 \ .b=-2)

    keep mergeid casp

save $easy\data\temp\sharew5_ac_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_ac.dta, clear

    mvdecode ac02* ac03* ac01*, mv(-1=.a \ -2=.b)

  * Quality of Life Score (CASP-12)

    // inverse into  *_inv (high value / high control etc.)
    recode ac017_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac017_inv)
    recode ac020_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac020_inv)
    recode ac021_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac021_inv)
    recode ac022_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac022_inv)
    recode ac023_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac023_inv)
    recode ac024_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac024_inv)
    recode ac025_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac025_inv)

    // Subscale Control
    gen     con = ac014_ + ac015_ + ac016_
    alpha         ac014_   ac015_   ac016_

    // Subscale Autonomy
    gen aut= ac017_inv + ac018_ + ac019_
    alpha    ac017_inv   ac018_   ac019_
```

```
    // Subscale Pleasure
    gen     ple= ac020_inv + ac021_inv + ac022_inv
    alpha        ac020_inv   ac021_inv   ac022_inv

    // Subscale Self-Realisation
    gen     sel = ac023_inv + ac024_inv + ac025_inv
    alpha         ac023_inv   ac024_inv   ac025_inv

    gen     casp = con + aut + ple + sel
    lab var casp "CASP quality of life (high is high quality)"

    mvencode ac02* ac03* ac01*, mv(.a=-1 \ .b=-2)

   rename ac012 satisfied

    keep mergeid casp

save $easy\data\temp\sharew6_ac_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_ac.dta, clear

    mvdecode ac02* ac03* ac01*, mv(-1=.a \ -2=.b)

  * Quality of Life Score (CASP-12)

    // inverse into  *_inv (high value / high control etc.)
    recode ac017_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac017_inv)
    recode ac020_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac020_inv)
    recode ac021_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac021_inv)
    recode ac022_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac022_inv)
    recode ac023_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac023_inv)
    recode ac024_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac024_inv)
    recode ac025_ (4=1) (3=2) (2=3) (1=4) (else=.) , gen(ac025_inv)

    // Subscale Control
    gen     con = ac014_ + ac015_ + ac016_
    alpha         ac014_   ac015_   ac016_

    // Subscale Autonomy
    gen aut= ac017_inv + ac018_ + ac019_
    alpha    ac017_inv   ac018_   ac019_

    // Subscale Pleasure
    gen     ple= ac020_inv + ac021_inv + ac022_inv
    alpha        ac020_inv   ac021_inv   ac022_inv

    // Subscale Self-Realisation
    gen     sel = ac023_inv + ac024_inv + ac025_inv
    alpha         ac023_inv   ac024_inv   ac025_inv

    gen     casp = con + aut + ple + sel
    lab var casp "CASP quality of life (high is high quality)"

    mvencode ac02* ac03* ac01*, mv(.a=-1 \ .b=-2)

    rename ac012 satisfied

    keep mergeid casp

save $easy\data\temp\sharew7_ac_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 7. Extract & Recode Variables from BR ]--------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_br.dta, clear

    recode br010_ (7=1) (6=2) (5=3) (4=4) (3=5) (2=6) (1=7) (-1=-1) (-2=-2) ///
                  (else=.)
    lab def lblalcohol -2 "refusal" -1 "don't know" 1 "not at all"      ///
                2 "less than once a month" 3 "once or twice a month"    ///
                4 "once or twice a week" 5 "three or four days a week"  ///
                6 "five or six days a week" 7 "almost every day"
    lab val br010_ lblalcohol
    rename br010_ br010_mod

    keep mergeid br001_ br002_ br010_mod br015_

save $easy\data\temp\sharew1_br_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_br.dta, clear

    recode br010_ (7=1) (6=2) (5=3) (4=4) (3=5) (2=6) (1=7) (-1=-1) (-2=-2) ///
                  (else=.)
    lab def lblalcohol -2 "refusal" -1 "don't know" 1 "not at all"      ///
                2 "less than once a month" 3 "once or twice a month"    ///
                4 "once or twice a week" 5 "three or four days a week"  ///
                6 "five or six days a week" 7 "almost every day"
    lab val br010_ lblalcohol
    rename br010_ br010_mod

    keep mergeid br001_ br002_ br010_mod br015_

save $easy\data\temp\sharew2_br_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_br.dta, clear

    recode br010_ (7=1) (6=2) (5=3) (4=4) (3=5) (2=6) (1=7) (-1=-1) (-2=-2) ///
                  (else=.)
    lab def lblalcohol -2 "refusal" -1 "don't know" 1 "not at all"      ///
                2 "less than once a month" 3 "once or twice a month"    ///
                4 "once or twice a week" 5 "three or four days a week"  ///
                6 "five or six days a week" 7 "almost every day"
    lab val br010_ lblalcohol
    rename br010_ br010_mod

    keep mergeid br001_ br002_ br010_mod br015_

save $easy\data\temp\sharew4_br_a.dta, replace
```

```
*>> w5:

use $easy\data\release\sharew5_rel7-0-0_br.dta, clear

    recode br010_ (7=1) (6=2) (5=3) (4=4) (3=5) (2=6) (1=7) (-1=-1) (-2=-2) ///
                  (else=.)
    lab def lblalcohol -2 "refusal" -1 "don't know" 1 "not at all"      ///
                2 "less than once a month" 3 "once or twice a month"    ///
                4 "once or twice a week" 5 "three or four days a week"  ///
                6 "five or six days a week" 7 "almost every day"
    lab val br010_ lblalcohol
    rename br010_ br010_mod

    keep mergeid br001_ br002_ br010_mod br015_

save $easy\data\temp\sharew5_br_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_br.dta, clear

    * br010_ not asked in wave 6
    keep mergeid br001_ br002_ br015_

save $easy\data\temp\sharew6_br_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_br.dta, clear

    * br010_ not asked in wave 7
    keep mergeid br001_ br002_ br015_

save $easy\data\temp\sharew7_br_a.dta, replace


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 8. Extract & Recode Variables from CF]---------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_cf.dta, clear

    gen recall_1=cf008tot
    gen recall_2=cf016tot
    lab var recall_1 "Recall of words, first trial (based on cf008tot)"
    lab var recall_2 "Recall of words, delayed (based on cf016tot)"

    gen numeracy_2 =.
    lab var numeracy_2 "Score of second numeracy test (subtraction)"
    lab val numeracy_2 lbl_num_2

    keep mergeid recall* numeracy_2

save $easy\data\temp\sharew1_cf_a.dta, replace
```

```
*>> w2:

use $easy\data\release\sharew2_rel7-0-0_cf.dta, clear

    gen recall_1=cf008tot
    gen recall_2=cf016tot
    lab var recall_1 "Recall of words, first trial (based on cf008tot)"
    lab var recall_2 "Recall of words, delayed (based on cf016tot)"

    gen numeracy_2 =.
    lab var numeracy_2 "Score of second numeracy test (subtraction)"
    lab def lbl_num_2 0 "bad" 5 "good"
    lab val numeracy_2 lbl_num_2
    keep mergeid recall* numeracy_2
    save $easy\data\temp\sharew2_cf_a.dta, replace


*>> w4: recall in wave 4 is generated based on gv_health-module

use $easy\data\release\sharew4_rel7-0-0_cf.dta, clear

    gen count_cf108_ = 1 if cf108_==93
    gen count_cf109_ = 1 if cf109_==cf108_-7 & cf109_!=.
    gen count_cf110_ = 1 if cf110_==cf109_-7 & cf110_!=.
    gen count_cf111_ = 1 if cf111_==cf110_-7 & cf111_!=.
    gen count_cf112_ = 1 if cf112_==cf111_-7 & cf112_!=.
    recode count_cf108_ .=0
    recode count_cf109_ .=0
    recode count_cf110_ .=0
    recode count_cf111_ .=0
    recode count_cf112_ .=0

    gen numeracy_2 = count_cf108_ + count_cf109_ + count_cf110_ + count_cf111_ ///
        + count_cf112_
    replace numeracy_2 =. if cf108_ ==.
    lab var numeracy_2 "Score of second numeracy test (subtraction)"
    lab def lbl_num_2 0 "bad" 5 "good"
    lab val numeracy_2 lbl_num_2
    keep mergeid numeracy_2

save $easy\data\temp\sharew4_cf_a.dta, replace


*>> w5: recall in wave 5 is generated based on gv_health-module

use $easy\data\release\sharew5_rel7-0-0_cf.dta, clear

    gen count_cf108_ = 1 if cf108_==93
    gen count_cf109_ = 1 if cf109_==cf108_-7 & cf109_!=.
    gen count_cf110_ = 1 if cf110_==cf109_-7 & cf110_!=.
    gen count_cf111_ = 1 if cf111_==cf110_-7 & cf111_!=.
    gen count_cf112_ = 1 if cf112_==cf111_-7 & cf112_!=.
    recode count_cf108_ .=0
    recode count_cf109_ .=0
    recode count_cf110_ .=0
    recode count_cf111_ .=0
    recode count_cf112_ .=0

    gen numeracy_2 = count_cf108_ + count_cf109_ + count_cf110_ + count_cf111_ ///
        + count_cf112_
```

```
    replace numeracy_2 =. if cf108_ ==.
    lab var numeracy_2 "Score of second numeracy test (subtraction)"
    lab def lbl_num_2 0 "bad" 5 "good"
    lab val numeracy_2 lbl_num_2

    keep mergeid numeracy_2

save $easy\data\temp\sharew5_cf_a.dta, replace


*>> w6: recall in wave 6 is generated based on gv_health-module

use $easy\data\release\sharew6_rel7-0-0_cf.dta, clear

    gen count_cf108_ = 1 if cf108_==93
    gen count_cf109_ = 1 if cf109_==cf108_-7 & cf109_!=.
    gen count_cf110_ = 1 if cf110_==cf109_-7 & cf110_!=.
    gen count_cf111_ = 1 if cf111_==cf110_-7 & cf111_!=.
    gen count_cf112_ = 1 if cf112_==cf111_-7 & cf112_!=.
    recode count_cf108_ .=0
    recode count_cf109_ .=0
    recode count_cf110_ .=0
    recode count_cf111_ .=0
    recode count_cf112_ .=0

   gen numeracy_2 = count_cf108_ + count_cf109_ + count_cf110_ + count_cf111_ ///
      + count_cf112_
    replace numeracy_2 =. if cf108_ ==.

    lab var numeracy_2 "score of second numeracy test (subtraction)"
    lab def lbl_num_2 0 "bad" 5 "good"
    lab val numeracy_2 lbl_num_2

    mergeid numeracy_2

save $easy\data\temp\sharew6_cf_a.dta, replace

*>> w7: recall in wave 7 is generated based on gv_health-module

use $easy\data\release\sharew7_rel7-0-0_cf.dta, clear

    gen count_cf108_ = 1 if cf108_==93
    gen count_cf109_ = 1 if cf109_==cf108_-7 & cf109_!=.
    gen count_cf110_ = 1 if cf110_==cf109_-7 & cf110_!=.
    gen count_cf111_ = 1 if cf111_==cf110_-7 & cf111_!=.
    gen count_cf112_ = 1 if cf112_==cf111_-7 & cf112_!=.
    recode count_cf108_ .=0
    recode count_cf109_ .=0
    recode count_cf110_ .=0
    recode count_cf111_ .=0
    recode count_cf112_ .=0

    gen numeracy_2 = count_cf108_ + count_cf109_ + count_cf110_ + count_cf111_ ///
        + count_cf112_
    replace numeracy_2 =. if cf108_ ==.

    lab var numeracy_2 "score of second numeracy test (subtraction)"
    lab def lbl_num_2 0 "bad" 5 "good"
    lab val numeracy_2 lbl_num_2

    keep mergeid numeracy_2

save $easy\data\temp\sharew7_cf_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[ 9. Extract & Recode Variables from CH ]--------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_ch.dta, clear

    gen ch007_hh =.
    forval n=1/16 {
      replace ch007_hh = 0 if ch007_`n' > 2 & ch007_`n' < .
      }
    forval n=1/16 {
             replace ch007_hh = 1 if ch007_`n' == 1
             replace ch007_hh = 1 if ch007_`n' == 2
             }
    replace ch007_hh=-9 if ch001_==0
      lab var ch007_hh ///
     "At least one child lives in household/building"
    lab def lblch007_hh -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_hh lblch007_hh

    gen ch007_km =.
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km =-9 if ch007_hh ==-9
    forval n=1/16 {
             replace ch007_km = 0 if ch007_`n' > 3 & ch007_`n' < .
             replace ch007_km = 1 if ch007_`n' == 3
             }
    lab var ch007_km ///
    "At least one child lives less than 1km away"
    lab def lblch007_km -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_km lblch007_km

    replace ch021_ = -9 if ch001_ == 0
    lab def lblch021_ -9 "filtered: no children" ///
                          -2 "refusal" -1 "don't know"
    lab val ch021_ lblch021_
    rename ch021_ ch021_mod

    keep mergeid ch001_ ch021_mod ch023_ ch007_hh ch007_km

save $easy\data\temp\sharew1_ch_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_ch.dta, clear

    gen ch007_hh =.
    forval n=1/16 {
      replace ch007_hh = 0 if ch007_`n' > 2 & ch007_`n' < .
             }
    forval n=1/16 {
             replace ch007_hh = 1 if ch007_`n' == 1
             replace ch007_hh = 1 if ch007_`n' == 2
             }
    replace ch007_hh=-9 if ch001_==0
```

```
    lab var ch007_hh ///
     "At least one child lives in household/building"
    lab def lblch007_hh -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_hh lblch007_hh

    gen ch007_km =.
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km =-9 if ch007_hh ==-9
    forval n=1/16 {
             replace ch007_km = 0 if ch007_`n' > 3 & ch007_`n' < .
             replace ch007_km = 1 if ch007_`n' == 3
             }
       lab var ch007_km ///
    "At least one child lives less than 1km away"
    lab def lblch007_km -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_km lblch007_km

    replace ch021_ = -9 if ch001_ == 0
    lab def lblch021_ -9 "filtered: no children" ///
                          -2 "refusal" -1 "don't know"
    lab val ch021_ lblch021_
    rename ch021_ ch021_mod

    keep mergeid ch001_ ch021_mod ch023_ ch007_hh ch007_km

save $easy\data\temp\sharew2_ch_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_ch.dta, clear

    gen ch007_hh =.
    forval n=1/20 {
             replace ch007_hh = 0 if ch007_`n' > 2 & ch007_`n' < .
             }
    forval n=1/20 {
             replace ch007_hh = 1 if ch007_`n' == 1
             replace ch007_hh = 1 if ch007_`n' == 2
             }
    forval n=1/20 {
             replace ch007_hh = 0 if ch526_`n' > 2 & ch526_`n' < .
             }
    forval n=1/20 {
             replace ch007_hh = 1 if ch526_`n' == 1
             replace ch007_hh = 1 if ch526_`n' == 2
             }
             replace ch007_hh=-9 if ch001_==0
    lab var ch007_hh ///
     "At least one child lives in household/building"
    lab def lblch007_hh -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_hh lblch007_hh

    gen ch007_km =.
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km =-9 if ch007_hh ==-9
    forval n=1/20 {
             replace ch007_km = 0 if ch007_`n' > 3 & ch007_`n' < .
             replace ch007_km = 1 if ch007_`n' == 3
             }
```

```
    forval n=1/20 {
             replace ch007_km = 0 if ch526_`n' > 3 & ch526_`n' < .
             replace ch007_km = 1 if ch526_`n' == 3
             }
    lab var ch007_km ///
     "At least one child lives less than 1km away"
    lab def lblch007_km -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_km lblch007_km

    replace ch021_ = -9 if ch001_ == 0
    replace ch021_ = -3 if ch021_ >= 500
    lab def lblch021_ -9 "filtered: no children" ///
                      -3 "implausible value/suspected wrong" ///
                      -2 "refusal" -1 "don't know"
    lab val ch021_ lblch021_
    rename ch021_ ch021_mod

    keep mergeid ch001_ ch021_mod ch023_ ch007_hh ch007_km ch524_ ch526*

save $easy\data\temp\sharew4_ch_a.dta, replace


*>> w5:

use $easy\data\release\sharew5_rel7-0-0_ch.dta, clear

    gen ch007_hh =.
    forval n=1/20 {
             replace ch007_hh = 0 if ch007_`n' > 2 & ch007_`n' < .
             }
    forval n=1/20 {
             replace ch007_hh = 1 if ch007_`n' == 1
             replace ch007_hh = 1 if ch007_`n' == 2
             }
    forval n=1/20 {
             replace ch007_hh = 0 if ch526_`n' > 2 & ch526_`n' < .
             }
    forval n=1/20 {
             replace ch007_hh = 1 if ch526_`n' == 1
             replace ch007_hh = 1 if ch526_`n' == 2
             }
             replace ch007_hh=-9 if ch001_==0
    lab var ch007_hh ///
     "At least one child lives in household/building"
    lab def lblch007_hh -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_hh lblch007_hh

    gen ch007_km =.
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km =-9 if ch007_hh ==-9
    forval n=1/20 {
             replace ch007_km = 0 if ch007_`n' > 3 & ch007_`n' < .
             replace ch007_km = 1 if ch007_`n' == 3
             }
    forval n=1/20 {
             replace ch007_km = 0 if ch526_`n' > 3 & ch526_`n' < .
             replace ch007_km = 1 if ch526_`n' == 3
             }
    lab var ch007_km ///
     "At least one child lives less than 1km away"
```

```
    lab def lblch007_km -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_km lblch007_km

    replace ch021_ = -9 if ch001_ == 0
    replace ch021_ = -3 if ch021_ >= 500
    lab def lblch021_ -9 "filtered: no children" ///
                      -3 "implausible value/suspected wrong" ///
                      -2 "refusal" -1 "don't know"
    lab val ch021_ lblch021_
    rename ch021_ ch021_mod

    keep mergeid ch001_ ch021_mod ch023_ ch007_hh ch007_km ch524_ ch526*

save $easy\data\temp\sharew5_ch_a.dta, replace

*>> w6:

use $easy\data\release\sharew6_rel7-0-0_ch.dta, clear

    gen ch007_hh =.
      forval n=1/20 {
             replace ch007_hh = 0 if ch007_`n' > 2 & ch007_`n' < .
             replace ch007_hh = 1 if ch007_`n' == 1
             replace ch007_hh = 1 if ch007_`n' == 2
             }
    forval n=1/20 {
             replace ch007_hh = 0 if ch526_`n' > 2 & ch526_`n' < .
             replace ch007_hh = 1 if ch526_`n' == 1
             replace ch007_hh = 1 if ch526_`n' == 2
             }
             replace ch007_hh=-9 if ch001_==0
    lab var ch007_hh ///
     "at least one child lives in household/building"
    lab def lblch007_hh -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_hh lblch007_hh

    gen ch007_km =.
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km =-9 if ch007_hh ==-9
    forval n=1/20 {
             replace ch007_km = 0 if ch007_`n' > 3 & ch007_`n' < .
             replace ch007_km = 1 if ch007_`n' == 3
             }
    forval n=1/20 {
             replace ch007_km = 0 if ch526_`n' > 3 & ch526_`n' < .
             replace ch007_km = 1 if ch526_`n' == 3
             }
    lab var ch007_km ///
     "at least one child lives less than 1km away"
    lab def lblch007_km -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_km lblch007_km

    replace ch021_ = -9 if ch001_ == 0
    replace ch021_ = -3 if ch021_ >= 500
    lab def lblch021_ -9 "filtered: no children" ///
                      -3 "implausible value/suspected wrong" ///
                      -2 "refusal" -1 "don't know"
    lab val ch021_ lblch021_
    rename ch021_ ch021_mod

    keep mergeid ch001_ ch021_mod ch023_ ch007_hh ch007_km ch524_ ch526*

save $easy\data\temp\sharew6_ch_a.dta, replace
```

```
*>> w7:

use $easy\data\release\sharew7_rel7-0-0_ch.dta, clear

    gen ch007_hh =.
    forval n=1/20 {
             replace ch007_hh = 0 if ch007_`n' > 2 & ch007_`n' < .
             replace ch007_hh = 1 if ch007_`n' == 1
             replace ch007_hh = 1 if ch007_`n' == 2
             }
    forval n=1/20 {
             replace ch007_hh = 0 if ch526_`n' > 2 & ch526_`n' < .
             replace ch007_hh = 1 if ch526_`n' == 1
             replace ch007_hh = 1 if ch526_`n' == 2
             }
             replace ch007_hh=-9 if ch001_==0
    lab var ch007_hh ///
     "at least one child lives in household/building"
    lab def lblch007_hh -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_hh lblch007_hh

    gen ch007_km =.
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km =-9 if ch007_hh ==-9
    forval n=1/20 {
             replace ch007_km = 0 if ch007_`n' > 3 & ch007_`n' < .
             replace ch007_km = 1 if ch007_`n' == 3
             }
    forval n=1/20 {
             replace ch007_km = 0 if ch526_`n' > 3 & ch526_`n' < .
             replace ch007_km = 1 if ch526_`n' == 3
             }
    lab var ch007_km ///
     "at least one child lives less than 1km away"
    lab def lblch007_km -9 "filtered: no children" 0 "no" 1 "yes"
    lab val ch007_km lblch007_km

    replace ch021_ = -9 if ch001_ == 0
    replace ch021_ = -3 if ch021_ >= 500
    lab def lblch021_ -9 "filtered: no children" ///
                      -3 "implausible value/suspected wrong" ///
                      -2 "refusal" -1 "don't know"
    lab val ch021_ lblch021_
    rename ch021_ ch021_mod

    keep mergeid ch001_ ch021_mod ch023_ ch007_hh ch007_km ch524_ ch526*

save $easy\data\temp\sharew7_ch_a.dta, replace
```

```
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[10. Extract & Recode Variables from CO ]--------------------------------

*>> w1:

use  $easy\data\release\sharew1_rel7-0-0_co.dta, clear
     keep mergeid co007_
save $easy\data\temp\sharew1_co_a.dta, replace

*>> w2:

use  $easy\data\release\sharew2_rel7-0-0_co.dta, clear
     keep mergeid co007_
save $easy\data\temp\sharew2_co_a.dta, replace

*>> w4:

use  $easy\data\release\sharew4_rel7-0-0_co.dta, clear
     keep mergeid co007_
save $easy\data\temp\sharew4_co_a.dta, replace

*>> w5:

use  $easy\data\release\sharew5_rel7-0-0_co.dta, clear
     keep mergeid co007_
save $easy\data\temp\sharew5_co_a.dta, replace

*>> w6:

use  $easy\data\release\sharew6_rel7-0-0_co.dta, clear
     keep mergeid co007_
save $easy\data\temp\sharew6_co_a.dta, replace


*>> w7:

use  $easy\data\release\sharew7_rel7-0-0_co.dta, clear
     keep mergeid co007_
save $easy\data\temp\sharew7_co_a.dta, replace


*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[11. Extract & Recode Variables from CS ]--------------------------------


*>> w3:

use $easy\data\release\sharew3_rel7-0-0_cs.dta, clear

    lab def lblbooks -2 "refusal" -1 "don't know" ///
                      1 "none or very few (0-10 books)" ///
                      2 "enough to fill one shelf (11-25 books)" ///
                      3 "enough to fill one bookcase (26-100 books)" ///
                      4 "enough to fill two bookcases (101-200 books)" ///
```

```
                    5 "enough to fill two or more bookcases (more than 200)"
    lab val sl_cs008_ lblbooks
    rename sl_cs008_ books_age10


    recode sl_cs010_ 9=-9
    lab def lblmaths -2 "refusal" -1 "don't know" ///
                    -9 "filtered: did not go to school" ///
                     1 "much better" 2 "better" 3 "about the same" ///
                     4 "worse" 5 "much worse"
    lab val sl_cs010_ lblmaths
    rename sl_cs010_ maths_age10

    recode sl_cs010a_ .=-9 if sl_cs010a_==-9
    lab def lbllanguage -2 "refusal" -1 "don't know" ///
                        -9 "filtered: did not go to school" ///
                         1 "much better" 2 "better" 3 "about the same" ///
                         4 "worse" 5 "much worse"
    lab val sl_cs010a_ lbllanguage
    rename sl_cs010a_ language_age10

    keep mergeid books_age10 maths_age10 language_age10

save $easy\data\temp\sharew3_cs_a.dta, replace



*>> w7:

use $easy\data\release\sharew7_rel7-0-0_cc.dta, clear

    lab def lblbooks -2 "refusal" -1 "don't know" ///
                     1 "none or very few (0-10 books)" ///
                     2 "enough to fill one shelf (11-25 books)" ///
                     3 "enough to fill one bookcase (26-100 books)" ///
                     4 "enough to fill two bookcases (101-200 books)" ///
                     5 "enough to fill two or more bookcases (more than 200)"
    lab val cc008_ lblbooks
    rename  cc008_ books_age10


    recode cc010_ 9=-9
    lab def lblmaths -2 "refusal" -1 "don't know" ///
                    -9 "filtered: did not go to school" ///
                     1 "much better" 2 "better" 3 "about the same" ///
                     4 "worse" 5 "much worse"
    lab val cc010_ lblmaths
    rename  cc010_ maths_age10

    recode cc010a_ .=-9 if cc010a_==-9
    lab def lbllanguage -2 "refusal" -1 "don't know" ///
                        -9 "filtered: did not go to school" ///
                         1 "much better" 2 "better" 3 "about the same" ///
                         4 "worse" 5 "much worse"
    lab val cc010a_ lbllanguage
    rename  cc010a_ language_age10

    keep mergeid books_age10 maths_age10 language_age10

save $easy\data\temp\sharew7_cc_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[12. Extract & Recode Variables from EP ]--------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_ep.dta, clear

    lab def lblep005_ -2 "refusal" -1 "don't know"      1 "retired"    ///
                       2 "employed or self-employed"    3 "unemployed" ///
                       4 "permanently sick or disabled" 5 "homemaker"  ///
                      97 "other"
    lab val ep005_ lblep005_

    replace ep009_1=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep009_1 -9 "filtered" -2 "refusal"          ///
                       -1 "don't know"            1 "employee"         ///
                        2 "civil servant"         3 "self-employed"
    lab val ep009_1 lblep009_1
    rename ep009_1 ep009_mod

    replace ep011_1=-9 if ep009_mod==3 |  ep009_mod==-9
        // -9 for "not worked" added
    lab def lblep011_1 -9 "filtered"   ///
                       -2 "refusal" -1 "don't know" 1 "short-term"  ///
                        2 "permanent"
    lab val ep011_1 lblep011_1
    rename ep011_1 ep011_mod

    replace ep013_1=-9 if ep002_==5 & ep003_==5 & ep005_!=2 & ep005_>0
    lab def lblep013_1 -9 "filtered" -2 "refusal" ///
                       -1 "don't know"
    lab val ep013_1 lblep013_1
    rename ep013_1 ep013_mod

    replace ep026_=-9 if ep003_==5 & ep005_!=2 & ep005_>0
    lab def lblep026_ -9 "filtered" -2 "refusal"        ///
                         -1 "don't know"       1 "strongly agree" ///
                          2 "agree"            3 "disagree"       ///
                          4 "strongly disagree"
    lab val ep026_ lblep026_
    rename ep026_ ep026_mod

    replace ep036_=-9 if ep005_ !=. & (ep005_>2 | ep005_==1)
    lab def lblep036_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know" 1 "yes" 5 "no"
    lab val ep036_ lblep036_
    rename ep036_ ep036_mod

    keep mergeid ep005_ ep009_mod ep011_mod ep013_mod ep026_mod ep036_mod

save $easy\data\temp\sharew1_ep_a.dta, replace
```

```
*>> w2:

use $easy\data\release\sharew2_rel7-0-0_ep.dta, clear

    lab def lblep005_ -2 "refusal" -1 "don't know"      1 "retired"     ///
                       2 "employed or self-employed"    3 "unemployed"  ///
                       4 "permanently sick or disabled" 5 "homemaker"   ///
                      97 "other"
    lab val ep005_ lblep005_

    replace ep009_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep009_ -9 "filtered" -2 "refusal"            ///
                      -1 "don't know"            1 "employee"           ///
                       2 "civil servant"         3 "self-employed"
    lab val ep009_ lblep009_
    rename ep009_ ep009_mod


    replace ep011_=-9 if ep009_==3 | ep009_==-9
    lab def lblep011_ -9 "filtered"        ///
                      -2 "refusal" -1 "don't know" 1 "short-term"       ///
                       2 "permanent"
    lab val ep011_ lblep011_
    rename ep011_ ep011_mod

    replace ep013_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep013_ -9 "filtered" -2 "refusal"    ///
                      -1 "don't know"
    lab val ep013_ lblep013_
    rename ep013_ ep013_mod

    replace ep026_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep026_ -9 "filtered" -2 "refusal"             ///
                      -1 "don't know" 1 "strongly agree"                  ///
                       2 "agree"      3 "disagree" 4 "strongly disagree"
    lab val ep026_ lblep026_
    rename ep026_ ep026_mod

    replace ep036_=-9 if ep005_ !=. & (ep005_>2 | ep005_==1)
    lab def lblep036_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"  1 "yes" 5 "no"
    lab val ep036_ lblep036_
    rename ep036_ ep036_mod

    keep mergeid ep005_ ep009_mod ep011_mod ep013_mod ep026_mod ep036_mod

save $easy\data\temp\sharew2_ep_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_ep.dta, clear

    lab def lblep005_ -2 "refusal" -1 "don't know"      1 "retired"     ///
                       2 "employed or self-employed"    3 "unemployed"  ///
                       4 "permanently sick or disabled" 5 "homemaker"   ///
                      97 "other"
    lab val ep005_ lblep005_

    replace ep009_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
```

```
    lab def lblep009_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"    1 "employee"        ///
                       2 "civil servant" 3 "self-employed"
    lab val ep009_ lblep009_
    rename ep009_ ep009_mod


    replace ep011_=-9 if ep009_==3 | ep009_==-9
    lab def lblep011_ -9 "filtered"  ///
                      -2 "refusal" -1 "don't know" 1 "short-term" ///
                       2 "permanent"
    lab val ep011_ lblep011_
    rename ep011_ ep011_mod

    replace ep013_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep013_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"
    lab val ep013_ lblep013_
    rename ep013_ ep013_mod

    replace ep026_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep026_ -9 "filtered" -2 "refusal"        ///
                      -1 "don't know" 1 "strongly agree"            ///
                       2 "agree" 3 "disagree" 4 "strongly disagree"
    lab val ep026_ lblep026_
    rename ep026_ ep026_mod

    replace ep036_=-9 if ep005_ !=. & (ep005_>2 | ep005_==1)
    lab def lblep036_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"  1 "yes" 5 "no"
    lab val ep036_ lblep036_
    rename ep036_ ep036_mod

    keep mergeid ep005_ ep009_mod ep011_mod ep013_mod ep026_mod ep036_mod

save $easy\data\temp\sharew4_ep_a.dta, replace


*>> w5:

use $easy\data\release\sharew5_rel7-0-0_ep.dta, clear

    lab def lblep005_ -2 "refusal" -1 "don't know"      1 "retired"     ///
                       2 "employed or self-employed"    3 "unemployed"  ///
                       4 "permanently sick or disabled" 5 "homemaker"   ///
                      97 "other"
    lab val ep005_ lblep005_

    replace ep009_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep009_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"    1 "employee"        ///
                       2 "civil servant" 3 "self-employed"
    lab val ep009_ lblep009_
    rename ep009_ ep009_mod

    replace ep011_=-9 if ep009_==3 | ep009_==-9
    lab def lblep011_ -9 "filtered"  ///
                      -2 "refusal" -1 "don't know" 1 "short-term" ///
                       2 "permanent"
    lab val ep011_ lblep011_
```

```
    rename ep011_ ep011_mod

    replace ep013_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep013_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"
    lab val ep013_ lblep013_
    rename ep013_ ep013_mod

    replace ep026_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep026_ -9 "filtered" -2 "refusal"        ///
                      -1 "don't know" 1 "strongly agree"            ///
                       2 "agree" 3 "disagree" 4 "strongly disagree"
    lab val ep026_ lblep026_
    rename ep026_ ep026_mod

    replace ep036_=-9 if ep005_ !=. & (ep005_>2 | ep005_==1)
    lab def lblep036_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"  1 "yes" 5 "no"
    lab val ep036_ lblep036_
    rename ep036_ ep036_mod

    keep mergeid ep005_ ep009_mod ep011_mod ep013_mod ep026_mod ep036_mod

save $easy\data\temp\sharew5_ep_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_ep.dta, clear

    lab def lblep005_ -2 "refusal" -1 "don't know"      1 "retired"     ///
                       2 "employed or self-employed"    3 "unemployed"  ///
                       4 "permanently sick or disabled" 5 "homemaker"   ///
                      97 "other"
    lab val ep005_ lblep005_

    replace ep009_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep009_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"    1 "employee"        ///
                       2 "civil servant" 3 "self-employed"
    lab val ep009_ lblep009_
    rename ep009_ ep009_mod

    replace ep011_=-9 if ep009_==3 | ep009_==-9
    lab def lblep011_ -9 "filtered"  ///
                      -2 "refusal" -1 "don't know" 1 "short-term" ///
                       2 "permanent"
    lab val ep011_ lblep011_
    rename ep011_ ep011_mod

    replace ep013_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep013_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"
    lab val ep013_ lblep013_
    rename ep013_ ep013_mod

    replace ep026_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep026_ -9 "filtered" -2 "refusal"        ///
                      -1 "don't know" 1 "strongly agree"            ///
                       2 "agree" 3 "disagree" 4 "strongly disagree"
```

```
    lab val ep026_ lblep026_
    rename ep026_ ep026_mod

    replace ep036_=-9 if ep005_ !=. & (ep005_>2 | ep005_==1)
    lab def lblep036_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"  1 "yes" 5 "no"
    lab val ep036_ lblep036_
    rename ep036_ ep036_mod

    keep mergeid ep005_ ep009_mod ep011_mod ep013_mod ep026_mod ep036_mod

save $easy\data\temp\sharew6_ep_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_ep.dta, clear

    lab def lblep005_ -2 "refusal" -1 "don't know"      1 "retired"     ///
                       2 "employed or self-employed"    3 "unemployed"  ///
                       4 "permanently sick or disabled" 5 "homemaker"   ///
                      97 "other"
    lab val ep005_ lblep005_

    replace ep009_=-9 if ep005_!=2 //& ep005_>0 & ep005_ !=.
    lab def lblep009_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"    1 "employee"        ///
                       2 "civil servant" 3 "self-employed"
    lab val ep009_ lblep009_
    rename ep009_ ep009_mod

    replace ep011_=-9 if ep009_==3 | ep009_==-9
    lab def lblep011_ -9 "filtered"  ///
                      -2 "refusal" -1 "don't know" 1 "short-term" ///
                       2 "permanent"
    lab val ep011_ lblep011_
    rename ep011_ ep011_mod

    replace ep013_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep013_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"
    lab val ep013_ lblep013_
    rename ep013_ ep013_mod

    replace ep026_=-9 if ep005_!=2 & ep005_>0 & ep005_ !=.
    lab def lblep026_ -9 "filtered" -2 "refusal"        ///
                      -1 "don't know" 1 "strongly agree"            ///
                       2 "agree" 3 "disagree" 4 "strongly disagree"
    lab val ep026_ lblep026_
    rename ep026_ ep026_mod

    replace ep036_=-9 if ep005_ !=. & (ep005_>2 | ep005_==1)
    lab def lblep036_ -9 "filtered" -2 "refusal" ///
                      -1 "don't know"  1 "yes" 5 "no"
    lab val ep036_ lblep036_
    rename ep036_ ep036_mod

    keep mergeid ep005_ ep009_mod ep011_mod ep013_mod ep026_mod ep036_mod

save $easy\data\temp\sharew7_ep_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[13. Extract & Recode Variables from GS ]--------------------------------

// Although release 7.0.0 of wave 3 already contains the variable maxgrip
// we keep the generation commends here for illustrative reasons

use $easy\data\release\sharew3_rel7-0-0_gs.dta, clear

*>> recode left and right vars
    generate left1 = sl_gs006_
    replace  left1 = .  if   (sl_gs006_ <= 0 | sl_gs006_ >= 100)

    generate left2 = sl_gs007_
    replace  left2 = .  if   (sl_gs007_ <= 0 | sl_gs007_ >= 100)

    generate right1 = sl_gs008_
    replace  right1 = .  if  (sl_gs008_ <= 0 | sl_gs008_ >= 100)

    generate right2 = sl_gs009_
    replace  right2 = .  if  (sl_gs009_ <= 0 | sl_gs009_ >= 100)


    // Counting no. of right hand grip strength measures
    egen  nright = robs(right1 right2)

    // Counting no. of left hand grip strength measures
    egen  nleft = robs(left1 left2)

*>> generating maxgrip:
    // if difference between the two measurements on one hand is < 20
    // only if two valid measurements on the measuring hand

    generate goodR = 1  if  (nright == 2 &  abs(right1 - right2) < 20)
    generate goodL = 1  if  (nleft  == 2 &  abs(left1  - left2 ) < 20)

    egen    maxgrip = rmax(left1 left2 right1 right2) if (goodR==1 & goodL==1)

    replace maxgrip = max(left1, left2)    if  (goodR == . & goodL == 1)
    replace maxgrip = max(right1, right2)  if  (goodR == 1 & goodL == .)

    lab var maxgrip  "Max. of grip strength measures"


    keep mergeid maxgrip

save $easy\data\temp\sharew3_gs_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[14. Extract & Recode Variables from HC ]--------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_hc.dta, clear

    rename hc002_ hc002_mod

    keep mergeid hc002_mod hc012_ hc029_

save $easy\data\temp\sharew1_hc_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_hc.dta, clear

    rename hc002_ hc002_mod

    keep mergeid hc002_mod hc012_ hc029_

save $easy\data\temp\sharew2_hc_a.dta, replace

*>> w3:

use $easy\data\release\sharew3_rel7-0-0_hc.dta, clear

    lab def lblvaccination -2 "refusal" -1 "don't know" ///
                           1 "Yes" 5 "No"
    lab val sl_hc002_ lblvaccination
    rename sl_hc002_  vaccinated

    keep mergeid vaccinated

save $easy\data\temp\sharew3_hc_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_hc.dta, clear

    rename hc002_ hc002_mod

    keep mergeid hc002_mod hc012_ hc029_

save $easy\data\temp\sharew4_hc_a.dta, replace

*>> w5:

use $easy\data\release\sharew5_rel7-0-0_hc.dta, clear

    rename hc002_ hc002_mod

    keep mergeid hc002_mod hc012_ hc029_

save $easy\data\temp\sharew5_hc_a.dta, replace
```

```
*>> w6: hc602_ instead of hc002_

use $easy\data\release\sharew6_rel7-0-0_hc.dta, clear

    replace hc602_ = 98 if hc602 >=98
    rename hc602_ hc002_mod // question text is slightly different in w6

    keep mergeid hc002_mod hc012_ hc029_

save $easy\data\temp\sharew6_hc_a.dta, replace


*>> w7: hc602_ instead of hc002_

use $easy\data\release\sharew7_rel7-0-0_hc.dta, clear

    replace hc602_ = 98 if hc602 >=98
    rename hc602_ hc002_mod // question text is slightly different in w6

    keep mergeid hc002_mod hc012_ hc029_

save $easy\data\temp\sharew7_hc_a.dta, replace


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[15. Extract & Recode Variables from HO]---------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_ho.dta, clear
    keep mergeid ho001_
save $easy\data\temp\sharew1_ho_a.dta, replace

*>> w2:

use $easy\data\release\sharew2_rel7-0-0_ho.dta, clear
    keep mergeid ho001_
save $easy\data\temp\sharew2_ho_a.dta, replace

*>> w4:

use $easy\data\release\sharew4_rel7-0-0_ho.dta, clear
    keep mergeid ho001_
save $easy\data\temp\sharew4_ho_a.dta, replace

*>> w5:

use $easy\data\release\sharew5_rel7-0-0_ho.dta, clear
    keep mergeid ho001_
save $easy\data\temp\sharew5_ho_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_ho.dta, clear
    keep mergeid ho001_
save $easy\data\temp\sharew6_ho_a.dta, replace
```

```
*>> w7:

use $easy\data\release\sharew7_rel7-0-0_ho.dta, clear
    keep mergeid ho001_
save $easy\data\temp\sharew7_ho_a.dta, replace


*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[16. Extract & Recode Variables from IV]---------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_iv.dta, clear
    keep mergeid iv009_
save $easy\data\temp\sharew1_iv_a.dta, replace

*>> w2:

use $easy\data\release\sharew2_rel7-0-0_iv.dta, clear
    keep mergeid iv009_
save $easy\data\temp\sharew2_iv_a.dta, replace

*>> w4:

use $easy\data\release\sharew4_rel7-0-0_iv.dta, clear
    keep mergeid iv009_
save $easy\data\temp\sharew4_iv_a.dta, replace

*>> w5:

use $easy\data\release\sharew5_rel7-0-0_iv.dta, clear
    keep mergeid iv009_
save $easy\data\temp\sharew5_iv_a.dta, replace

*>> w6:

use $easy\data\release\sharew6_rel7-0-0_iv.dta, clear
    keep mergeid iv009_
save $easy\data\temp\sharew6_iv_a.dta, replace

*>> w7:

use $easy\data\release\sharew7_rel7-0-0_iv.dta, clear
    keep mergeid iv009_
save $easy\data\temp\sharew7_iv_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[17. Extract & Recode Variables from MC ]--------------------------------


use $easy\data\release\sharew5_rel7-0-0_mc.dta, clear

    lab def lblbooks -2 "refusal" -1 "don't know"                        ///
                      1 "none or very few (0-10 books)"                  ///
                      2 "enough to fill one shelf (11-25 books)"         ///
                      3 "enough to fill one bookcase (26-100 books)"     ///
                      4 "enough to fill two bookcases (101-200 books)"   ///
                      5 "enough to fill two or more bookcases (more than 200)"
    lab val mc005_ lblbooks
    rename mc005_ books_age10


    recode mc006_ 9=-9
    lab def lblmaths -2 "refusal" -1 "don't know"                       ///
                     -9 "filtered: did not go to school"  ///
                      1 "much better" 2 "better" 3 "about the same"     ///
                      4 "worse" 5 "much worse"
    lab val mc006_ lblmaths
    rename mc006_ maths_age10


    recode mc007_ .=-9 if maths_age10==-9
    lab def lbllanguage -2 "refusal" -1 "don't know"                    ///
                        -9 "filtered: did not go to school"             ///
                         1 "much better" 2 "better" 3 "about the same" ///
                         4 "worse" 5 "much worse"
    lab val mc007_ lbllanguage
    rename mc007_ language_age10


    lab def lblchhealth -2 "refusal" -1 "don't know"                    ///
                         1 "Excellent" 2 "Very Good" 3 "Good"           ///
                         4 "Fair" 5 "Poor"                              ///
                         6 "Health varied a great deal"
    lab val mc010_ lblchhealth
    rename mc010_ childhood_health


    lab def lblvaccination -2 "refusal" -1 "don't know"                   ///
                            1 "Yes" 5 "No"
    lab val mc015_ lblvaccination
    rename mc015_ vaccinated

    keep mergeid books_age10 maths_age10 language_age10 childhood_health vaccinated

save $easy\data\temp\sharew5_mc_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[18. Extract & Recode Variables from PH ]--------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_ph.dta, clear

    mvdecode ph048* ph049* ph006d? ph006d?? , mv(-1=.a \ -2=.b)

    gen     mobilityind = ph048d1 + ph049d2 + ph048d4 + ph048d5
    lab var mobilityind "Mobility index (high: has difficulties)"

    gen     lgmuscle = ph048d2 + ph048d3 + ph048d6 + ph048d8
    lab var lgmuscle "Large muscle index (high: has difficulties)"

    gen     adlwa = ph049d1 + ph049d3 + ph049d4
    lab var adlwa ///
     "Activities of daily living w&h index (high: has difficulties)"

    gen     adla = ph049d1 + ph049d3 + ph049d4 + ph049d2 + ph049d5
    lab var adla "Activities of daily living index (high: has difficulties)"

    gen     grossmotor = ph048d1 + ph049d2 + ph048d5 + ph049d3
    lab var grossmotor "Gross motor skills index (high: has difficulties)"

    gen     finemotor = ph048d10 + ph049d4 + ph049d1
    lab var finemotor "Fine motor skills index (high: has difficulties)"

    gen     iadla = ph049d10 + ph049d11 + ph049d13
    lab var iadla ///
     "Instrumental activities of daily living index (high: has difficulties)"

    gen     iadlza= ph049d10 + ph049d11 + ph049d13 + ph049d9 + ph049d8
    lab var iadlza ///
     "Instrumental activities of daily living index 2 (high: has difficulties)"

    egen    chronic_mod = rowtotal(ph006d1  ph006d2  ph006d3  ph006d4  ///
                                   ph006d5  ph006d6  ph006d10 ph006d11 ///
                                   ph006d12 ph006d13 ph006d14)
    lab var chronic_mod "Number of chronic diseases (easySHARE version)"

    mvencode ph048* ph049* ph006d? ph006d??, mv(.a=-1 \ .b=-2)

    replace mobilityind = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace lgmuscle    = -12 if ph048d1==-1 | ph048d1==-2
    replace adlwa       = -12 if ph049d1==-1 | ph049d1==-2
    replace adla        = -12 if ph049d1==-1 | ph049d1==-2
    replace grossmotor  = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace finemotor   = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace iadla       = -12 if ph049d1==-1 | ph049d1==-2
    replace iadlza      = -12 if ph049d1==-1 | ph049d1==-2
    replace chronic_mod = -12 if ph006d1==-1 | ph006d1==-2
    replace chronic_mod =   . if ph006d1==  .

    keep mergeid mobilityind lgmuscle adlwa adla grossmotor finemotor iadla ///
         iadlza ph012 ph013 chronic_mod

save $easy\data\temp\sharew1_ph_a.dta, replace
```

```
*>> w2:

use $easy\data\release\sharew2_rel7-0-0_ph.dta, clear

    mvdecode ph048* ph049* ph006d? ph006d??, mv(-1=.a \ -2=.b)

    gen     mobilityind = ph048d1 + ph049d2 + ph048d4 + ph048d5
    lab var mobilityind "Mobility Index (high: has difficulties)"

    gen lgmuscle = ph048d2 + ph048d3 + ph048d6 + ph048d8
    lab var lgmuscle "Large Muscle Index (high: has difficulties)"

    gen     adlwa = ph049d1 + ph049d3 + ph049d4
    lab var adlwa ///
     "Actvities of Daily Living W&H Index (high: has difficulties)"

    gen     adla= ph049d1 + ph049d3 + ph049d4 + ph049d2 + ph049d5
    lab var adla "Actvities of Daily Living Index (high: has difficulties)"

    gen     grossmotor = ph048d1 + ph049d2 + ph048d5 + ph049d3
    lab var grossmotor "Gross Motor Skills Index (high: has difficulties)"

    gen     finemotor = ph048d10 + ph049d4 + ph049d1
    lab var finemotor "Fine Motor Skills Index (high: has difficulties)"

    gen     iadla = ph049d10 + ph049d11 + ph049d13
    lab var iadla ///
     "Instrumental activities of Daily Living Index (high: has difficulties)"

    gen     iadlza = ph049d10 + ph049d11 + ph049d13 + ph049d9 + ph049d8
    lab var iadlza ///
     "Instrumental activities of Daily Living Index 2 (high: has difficulties)"

    egen    chronic_mod = rowtotal(ph006d1  ph006d2  ph006d3  ph006d4  ///
                                   ph006d5  ph006d6  ph006d10 ph006d11 ///
                                   ph006d12 ph006d13 ph006d14)
    lab var chronic_mod "Number of chronic diseases (easySHARE version)"

    mvencode ph048* ph049* ph006d? ph006d??, mv(.a=-1 \ .b=-2)

    replace mobilityind = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace lgmuscle    = -12 if ph048d1==-1 | ph048d1==-2
    replace adlwa       = -12 if ph049d1==-1 | ph049d1==-2
    replace adla        = -12 if ph049d1==-1 | ph049d1==-2
    replace grossmotor  = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace finemotor   = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace iadla       = -12 if ph049d1==-1 | ph049d1==-2
    replace iadlza      = -12 if ph049d1==-1 | ph049d1==-2
    replace chronic_mod = -12 if ph006d1==-1 | ph006d1==-2
    replace chronic_mod =   . if ph006d1==  .  // egen ignores sysmis

    keep mergeid mobilityind lgmuscle adlwa adla grossmotor finemotor iadla ///
         iadlza ph012 ph013 chronic_mod

save $easy\data\temp\sharew2_ph_a.dta, replace
```

```
*>> w4:

use $easy\data\release\sharew4_rel7-0-0_ph.dta, clear

    mvdecode ph048* ph049*, mv(-1=.a \ -2=.b)

    gen     mobilityind = ph048d1 + ph049d2 + ph048d4 + ph048d5
    lab var mobilityind "Mobility Index (high: has difficulties)"

    gen     lgmuscle = ph048d2 + ph048d3 + ph048d6 + ph048d8
    lab var lgmuscle "Large Muscle Index (high: has difficulties)"

    gen     adlwa = ph049d1 + ph049d3 + ph049d4
    lab var adlwa ///
     "Activities of Daily Living W&H Index (high: has difficulties)"

    gen     adla = ph049d1 + ph049d3 + ph049d4 + ph049d2 + ph049d5
    lab var adla "Activities of Daily Living Index (high: has difficulties)"

    gen     grossmotor = ph048d1 + ph049d2 + ph048d5 + ph049d3
    lab var grossmotor "Gross Motor Skills Index (high: has difficulties)"

    gen     finemotor = ph048d10 + ph049d4 + ph049d1
    lab var finemotor "Fine Motor Skills Index (high: has difficulties)"

    gen     iadla= ph049d10 + ph049d11 + ph049d13
    lab var iadla ///
     "Instrumental acivities of Daily Living Index (high: has difficulties)"

    gen     iadlza= ph049d10 + ph049d11 + ph049d13 + ph049d9 + ph049d8
    lab var iadlza ///
     "Instrumental activities of Daily Living Index 2 (high: has difficulties)"

    egen    chronic_mod = rowtotal(ph006d1  ph006d2  ph006d3  ph006d4  ///
                                   ph006d5  ph006d6  ph006d10 ph006d11 ///
                                   ph006d12 ph006d13 ph006d14)
    lab var chronic_mod "Number of chronic diseases (easySHARE version)"

    mvencode ph048* ph049* ph048* ph049*, mv(.a=-1 \ .b=-2)

    replace mobilityind = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace lgmuscle    = -12 if ph048d1==-1 | ph048d1==-2
    replace adlwa       = -12 if ph049d1==-1 | ph049d1==-2
    replace adla        = -12 if ph049d1==-1 | ph049d1==-2
    replace grossmotor  = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace finemotor   = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace iadla       = -12 if ph049d1==-1 | ph049d1==-2
    replace iadlza      = -12 if ph049d1==-1 | ph049d1==-2
    replace chronic_mod = -12 if ph006d1==-1 | ph006d1==-2
    replace chronic_mod =   . if ph006d1==  .  // egen ignores sysmis

    keep mergeid mobilityind lgmuscle adlwa adla grossmotor finemotor iadla ///
         iadlza ph012 ph013 chronic_mod

save $easy\data\temp\sharew4_ph_a.dta, replace
```

```
*>> w5:

use $easy\data\release\sharew5_rel7-0-0_ph.dta, clear

    mvdecode ph048* ph049*, mv(-1=.a \ -2=.b)

    gen     mobilityind = ph048d1 + ph049d2 + ph048d4 + ph048d5
    lab var mobilityind "Mobility Index (high: has difficulties)"

    gen     lgmuscle = ph048d2 + ph048d3 + ph048d6 + ph048d8
    lab var lgmuscle "large Muscle Index (high: has difficulties)"

    gen     adlwa = ph049d1 + ph049d3 + ph049d4
    lab var adlwa ///
     "Activities of Daily Living W&H Index (high: has difficulties)"

    gen     adla = ph049d1 + ph049d3 + ph049d4 + ph049d2 + ph049d5
    lab var adla "Activities of Daily Living Index (high: has difficulties)"

    gen     grossmotor = ph048d1 + ph049d2 + ph048d5 + ph049d3
    lab var grossmotor "Gross Motor Skills Index (high: has difficulties)"

    gen     finemotor = ph048d10 + ph049d4 + ph049d1
    lab var finemotor "Fine Motor Skills Index (high: has difficulties)"

    gen     iadla= ph049d10 + ph049d11 + ph049d13
    lab var iadla ///
     "Instrumental acivities of Daily Living Index (high: has difficulties)"

    gen     iadlza= ph049d10 + ph049d11 + ph049d13 + ph049d9 + ph049d8
    lab var iadlza ///
     "Instrumental activities of Daily Living Index 2 (high: has difficulties)"

    egen    chronic_mod = rowtotal(ph006d1  ph006d2  ph006d3  ph006d4  ///
                                   ph006d5  ph006d6  ph006d10 ph006d11 ///
                                   ph006d12 ph006d13 ph006d14)
    lab var chronic_mod "Number of chronic diseases (easySHARE version)"

    mvencode ph048* ph049* ph048* ph049*, mv(.a=-1 \ .b=-2)

    replace mobilityind = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace lgmuscle    = -12 if ph048d1==-1 | ph048d1==-2
    replace adlwa       = -12 if ph049d1==-1 | ph049d1==-2
    replace adla        = -12 if ph049d1==-1 | ph049d1==-2
    replace grossmotor  = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace finemotor   = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace iadla       = -12 if ph049d1==-1 | ph049d1==-2
    replace iadlza      = -12 if ph049d1==-1 | ph049d1==-2
    replace chronic_mod = -12 if ph006d1==-1 | ph006d1==-2
    replace chronic_mod =   . if ph006d1==  .  // egen ignores sysmis

    keep mergeid mobilityind lgmuscle adlwa adla grossmotor finemotor iadla ///
         iadlza ph012 ph013 chronic_mod

save $easy\data\temp\sharew5_ph_a.dta, replace
```

```
*>> w6:

use $easy\data\release\sharew6_rel7-0-0_ph.dta, clear

    mvdecode ph048* ph049*, mv(-1=.a \ -2=.b)

    gen     mobilityind = ph048d1 + ph049d2 + ph048d4 + ph048d5
    lab var mobilityind "Mobility Index (high: has difficulties)"

    gen     lgmuscle = ph048d2 + ph048d3 + ph048d6 + ph048d8
    lab var lgmuscle "large Muscle Index (high: has difficulties)"

    gen     adlwa = ph049d1 + ph049d3 + ph049d4
    lab var adlwa ///
     "Activities of Daily Living W&H Index (high: has difficulties)"

    gen     adla = ph049d1 + ph049d3 + ph049d4 + ph049d2 + ph049d5
    lab var adla "Activities of Daily Living Index (high: has difficulties)"

    gen     grossmotor = ph048d1 + ph049d2 + ph048d5 + ph049d3
    lab var grossmotor "Gross Motor Skills Index (high: has difficulties)"

    gen     finemotor = ph048d10 + ph049d4 + ph049d1
    lab var finemotor "Fine Motor Skills Index (high: has difficulties)"

    gen     iadla= ph049d10 + ph049d11 + ph049d13
    lab var iadla ///
     "Instrumental acivities of Daily Living Index (high: has difficulties)"

    gen     iadlza= ph049d10 + ph049d11 + ph049d13 + ph049d9 + ph049d8
    lab var iadlza ///
     "Instrumental activities of Daily Living Index 2 (high: has difficulties)"

    egen    chronic_mod = rowtotal(ph006d1  ph006d2  ph006d3  ph006d4  ///
                                   ph006d5  ph006d6  ph006d10 ph006d11 ///
                                   ph006d12 ph006d13 ph006d14)
    lab var chronic_mod "Number of chronic diseases (easySHARE version)"

    mvencode ph048* ph049* ph048* ph049*, mv(.a=-1 \ .b=-2)

    replace mobilityind = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace lgmuscle    = -12 if ph048d1==-1 | ph048d1==-2
    replace adlwa       = -12 if ph049d1==-1 | ph049d1==-2
    replace adla        = -12 if ph049d1==-1 | ph049d1==-2
    replace grossmotor  = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace finemotor   = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace iadla       = -12 if ph049d1==-1 | ph049d1==-2
    replace iadlza      = -12 if ph049d1==-1 | ph049d1==-2
    replace chronic_mod = -12 if ph006d1==-1 | ph006d1==-2
    replace chronic_mod =   . if ph006d1==  .  // egen ignores sysmis

    keep mergeid mobilityind lgmuscle adlwa adla grossmotor finemotor iadla ///
         iadlza ph012 ph013 chronic_mod

save $easy\data\temp\sharew6_ph_a.dta, replace
```

```
*>> w7:

use $easy\data\release\sharew7_rel7-0-0_ph.dta, clear

    mvdecode ph048* ph049*, mv(-1=.a \ -2=.b)

    gen     mobilityind = ph048d1 + ph049d2 + ph048d4 + ph048d5
    lab var mobilityind "Mobility Index (high: has difficulties)"

    gen     lgmuscle = ph048d2 + ph048d3 + ph048d6 + ph048d8
    lab var lgmuscle "large Muscle Index (high: has difficulties)"

    gen     adlwa = ph049d1 + ph049d3 + ph049d4
    lab var adlwa ///
     "Activities of Daily Living W&H Index (high: has difficulties)"

    gen     adla = ph049d1 + ph049d3 + ph049d4 + ph049d2 + ph049d5
    lab var adla "Activities of Daily Living Index (high: has difficulties)"

    gen     grossmotor = ph048d1 + ph049d2 + ph048d5 + ph049d3
    lab var grossmotor "Gross Motor Skills Index (high: has difficulties)"

    gen     finemotor = ph048d10 + ph049d4 + ph049d1
    lab var finemotor "Fine Motor Skills Index (high: has difficulties)"

    gen     iadla= ph049d10 + ph049d11 + ph049d13
    lab var iadla ///
     "Instrumental acivities of Daily Living Index (high: has difficulties)"

    gen     iadlza= ph049d10 + ph049d11 + ph049d13 + ph049d9 + ph049d8
    lab var iadlza ///
     "Instrumental activities of Daily Living Index 2 (high: has difficulties)"

    egen    chronic_mod = rowtotal(ph006d1  ph006d2  ph006d3  ph006d4  ///
                                   ph006d5  ph006d6  ph006d10 ph006d11 ///
                                   ph006d12 ph006d13 ph006d14)
    lab var chronic_mod "Number of chronic diseases (easySHARE version)"

    mvencode ph048* ph049* ph048* ph049*, mv(.a=-1 \ .b=-2)

    replace mobilityind = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace lgmuscle    = -12 if ph048d1==-1 | ph048d1==-2
    replace adlwa       = -12 if ph049d1==-1 | ph049d1==-2
    replace adla        = -12 if ph049d1==-1 | ph049d1==-2
    replace grossmotor  = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace finemotor   = -12 if ph048d1==-1 | ph048d1==-2 ///
                               | ph049d2==-1 | ph049d2==-2
    replace iadla       = -12 if ph049d1==-1 | ph049d1==-2
    replace iadlza      = -12 if ph049d1==-1 | ph049d1==-2
    replace chronic_mod = -12 if ph006d1==-1 | ph006d1==-2
    replace chronic_mod =   . if ph006d1==  .  // egen ignores sysmis

    keep mergeid mobilityind lgmuscle adlwa adla grossmotor finemotor iadla ///
         iadlza ph012 ph013 chronic_mod

save $easy\data\temp\sharew7_ph_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[19. Extract & Recode Variables from SN]---------------------------------

*>> w4:

use $easy\data\release\sharew4_rel7-0-0_sn.dta, clear
    keep mergeid sn005_* sn006_*
save $easy\data\temp\sharew4_sn_a.dta, replace

*>> w6:

use $easy\data\release\sharew6_rel7-0-0_sn.dta, clear
    keep mergeid sn005_* sn006_*
save $easy\data\temp\sharew6_sn_a.dta, replace

*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[20. Extract & Recode Variables from SP]---------------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_sp.dta, clear
    keep mergeid sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3
save $easy\data\temp\sharew1_sp_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_sp.dta, clear
    keep mergeid sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3
save $easy\data\temp\sharew2_sp_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_sp.dta, clear

   rename sp003_1sp sp003_1
   rename sp003_2sp sp003_2
   rename sp003_3sp sp003_3
   rename sp009_1sp sp009_1
   rename sp009_2sp sp009_2
   rename sp009_3sp sp009_3

   keep mergeid sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3

save $easy\data\temp\sharew4_sp_a.dta, replace


*>> w5:

use $easy\data\release\sharew5_rel7-0-0_sp.dta, clear
    keep mergeid sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3
save $easy\data\temp\sharew5_sp_a.dta, replace
```

```
*>> w6:

use $easy\data\release\sharew6_rel7-0-0_sp.dta, clear
    keep mergeid sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3
save $easy\data\temp\sharew6_sp_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_sp.dta, clear

    keep mergeid sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3


save $easy\data\temp\sharew7_sp_a.dta, replace


*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[21. Extract & Recode Variables from HS]---------------------------

*>> w3:

use $easy\data\release\sharew3_rel7-0-0_hs.dta, clear

    lab def lblchhealth -2 "refusal" -1 "don't know"                    ///
                         1 "Excellent" 2 "Very Good" 3 "Good"           ///
                         4 "Fair" 5 "Poor"                              ///
                         6 "Health varied a great deal"
    lab val sl_hs003_ lblchhealth
    rename sl_hs003_  childhood_health

    rename sl_ph003_ sphus
    keep mergeid sphus childhood_health

save $easy\data\temp\sharew3_hs_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_hs.dta, clear

    lab def lblchhealth -2 "refusal" -1 "don't know"                    ///
                         1 "Excellent" 2 "Very Good" 3 "Good"           ///
                         4 "Fair" 5 "Poor"                              ///
                         6 "Health varied a great deal"
    lab val hs003_ lblchhealth
    rename  hs003_ childhood_health

    keep mergeid childhood_health

save $easy\data\temp\sharew7_hs_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[22. Extract & Recode Variables from HS]---------------------------

*>> w7:

use $easy\data\release\sharew7_rel7-0-0_rh.dta, clear

    lab def lblvaccination -2 "refusal" -1 "don't know" ///
                            1 "Yes" 5 "No"
    lab val rh002_ lblvaccination
    rename rh002_ vaccinated

    keep mergeid vaccinated

save $easy\data\temp\sharew7_rh_a.dta, replace


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[23. Extract & Recode Variables from GV_Health]---------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_gv_health.dta, clear

    rename numeracy numeracy_1
    lab var numeracy_1 "Score of first numeracy test (percentage)"
    keep mergeid sphus euro* orienti numeracy_1 maxgrip bmi bmi2

save $easy\data\temp\sharew1_gv_health_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_gv_health.dta, clear

    rename numeracy numeracy_1
    lab var numeracy_1 "Score of numeracy test (percentage)"
    keep mergeid sphus euro* orienti numeracy_1 maxgrip bmi bmi2

save $easy\data\temp\sharew2_gv_health_a.dta, replace

*>> w4:

use $easy\data\release\sharew4_rel7-0-0_gv_health.dta, clear

    gen recall_1=cf008tot
    gen recall_2=cf016tot
    lab var recall_1 "Recall of words, first trial (based on cf008tot)"
    lab var recall_2 "Recall of words, delayed (based on cf016tot)"
    rename numeracy numeracy_1
    rename numeracy2 numeracy_2
    lab var numeracy_1 "Score of numeracy test (percentage)"
    lab var numeracy_2 "Score of numeracy test (subtraction)"
    keep mergeid sphus euro* maxgrip orienti numeracy_1 recall* bmi bmi2

save $easy\data\temp\sharew4_gv_health_a.dta, replace
```

```
*>> w5:

use $easy\data\release\sharew5_rel7-0-0_gv_health.dta, clear

    gen recall_1=cf008tot
    gen recall_2=cf016tot
    lab var recall_1 "Recall of words, first trial (based on cf008tot)"
    lab var recall_2 "Recall of words, delayed (based on cf016tot)"
    rename numeracy numeracy_1
    rename numeracy2 numeracy_2
    lab var numeracy_1 "Score of numeracy test (percentage)"
    lab var numeracy_2 "Score of numeracy test (subtraction)"
    keep mergeid sphus euro* maxgrip orienti numeracy_1 numeracy_2 recall* bmi bmi2

save $easy\data\temp\sharew5_gv_health_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_gv_health.dta, clear

    gen recall_1=cf008tot
    gen recall_2=cf016tot
    lab var recall_1 "Recall of words, first trial (based on cf008tot)"
    lab var recall_2 "Recall of words, delayed (based on cf016tot)"
    rename numeracy numeracy_1
    rename numeracy2 numeracy_2
    lab var numeracy_1 "Score of numeracy test (percentage)"
    lab var numeracy_2 "Score of numeracy test (subtraction)"
    keep mergeid sphus euro* maxgrip orienti numeracy_1 numeracy_2 recall* bmi bmi2

save $easy\data\temp\sharew6_gv_health_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_gv_health.dta, clear

    gen recall_1=cf008tot
    gen recall_2=cf016tot
    lab var recall_1 "Recall of words, first trial (based on cf008tot)"
    lab var recall_2 "Recall of words, delayed (based on cf016tot)"
    rename numeracy  numeracy_1
    rename numeracy2 numeracy_2
    lab var numeracy_1 "Score of numeracy test (percentage)"
    lab var numeracy_2 "Score of numeracy test (subtraction)"
    keep mergeid sphus euro* maxgrip orienti numeracy_1 numeracy_2 recall* bmi bmi2

save $easy\data\temp\sharew7_gv_health_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[24. Extract & Recode Variables from GV_ISCED ]--------------------------

// We do not use the eduyear variable asked in wave 1, because it is coded
// based on the ISCED classification and hence something very different in
// comparison to the self-reported years of education collected in later waves.

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_gv_isced.dta, clear
    keep mergeid isced1997_r
save $easy\data\temp\sharew1_gv_isced_a.dta, replace

*>> w2: -> replace missings after appending the waves

use $easy\data\release\sharew2_rel7-0-0_gv_isced.dta, clear
    keep mergeid isced1997_r
save $easy\data\temp\sharew2_gv_isced_a.dta, replace

*>> w4: -> replace missings after appending the waves

use $easy\data\release\sharew4_rel7-0-0_gv_isced.dta, clear
    keep mergeid isced1997_r
save $easy\data\temp\sharew4_gv_isced_a.dta, replace

*>> w5: -> replace missings after appending the waves

use $easy\data\release\sharew5_rel7-0-0_gv_isced.dta, clear
    keep mergeid isced1997_r
save $easy\data\temp\sharew5_gv_isced_a.dta, replace

*>> w6: -> replace missings after appending the waves

use $easy\data\release\sharew6_rel7-0-0_gv_isced.dta, clear
    keep mergeid isced1997_r
save $easy\data\temp\sharew6_gv_isced_a.dta, replace

*>> w7: -> replace missings after appending the waves

use $easy\data\release\sharew7_rel7-0-0_gv_isced.dta, clear
    keep mergeid isced1997_r
save $easy\data\temp\sharew7_gv_isced_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[25. Extract & Recode Variables from GV_Imputations]-------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_gv_imputations.dta, clear

  * mergeing with cv_r because int_year is needed:
    merge m:1 mergeid using "$easy\data\release\sharew1_rel7-0-0_cv_r.dta"
    drop if _merge == 2
    drop _merge

  * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen thinc_mod = mean(thinc)
    replace thinc = thinc_mod

  * We only keep one out of the five implicats:
    keep if implicat==1

  * Income
  * thinc: household total net income (previous year)
    gen     thinc_m=thinc_mod*exrate         // euro amounts into
                                             // local currency
                // in w1 all values going into thinc were asked for 2003
                // even for interview in 2005; this changed for w2 onwards
    replace thinc_m=thinc_m/pppk2003 if country != 25
    replace thinc_m=thinc_m/pppk2004 if country == 25

  * country-specific income percentiles
    foreach c in 11 12 13 14 15 16 17 18 19 20 23 25 {
     pctile  p10_`c' = thinc_m if country==`c', nq(10)
     return list
     gen     income_pct_w1_`c'=.
     replace income_pct_w1_`c'=1 if thinc_m<=r(r1)                 ///
                                    & country==`c'
     replace income_pct_w1_`c'=2 if thinc_m<=r(r2) & thinc_m>r(r1) ///
                                    & country==`c'
     replace income_pct_w1_`c'=3 if thinc_m<=r(r3) & thinc_m>r(r2) ///
                                    & country==`c'
     replace income_pct_w1_`c'=4 if thinc_m<=r(r4) & thinc_m>r(r3) ///
                                    & country==`c'
     replace income_pct_w1_`c'=5 if thinc_m<=r(r5) & thinc_m>r(r4) ///
                                    & country==`c'
     replace income_pct_w1_`c'=6 if thinc_m<=r(r6) & thinc_m>r(r5) ///
                                    & country==`c'
     replace income_pct_w1_`c'=7 if thinc_m<=r(r7) & thinc_m>r(r6) ///
                                    & country==`c'
     replace income_pct_w1_`c'=8 if thinc_m<=r(r8) & thinc_m>r(r7) ///
                                    & country==`c'
     replace income_pct_w1_`c'=9 if thinc_m<=r(r9) & thinc_m>r(r8) ///
                                    & country==`c'
     replace income_pct_w1_`c'=10 if thinc_m>r(r9) & thinc_m<.     ///
                                    & country==`c'
     lab val income_pct_w1_`c' income_pct`c'
    }

    *one variable including all countries
```

```
    gen income_pct_w1 = .
    foreach c in 11 12 13 14 15 16 17 18 19 20 23 25 {
        replace income_pct_w1 =  1  if  income_pct_w1_`c'== 1
        replace income_pct_w1 =  2  if  income_pct_w1_`c'== 2
        replace income_pct_w1 =  3  if  income_pct_w1_`c'== 3
        replace income_pct_w1 =  4  if  income_pct_w1_`c'== 4
        replace income_pct_w1 =  5  if  income_pct_w1_`c'== 5
        replace income_pct_w1 =  6  if  income_pct_w1_`c'== 6
        replace income_pct_w1 =  7  if  income_pct_w1_`c'== 7
        replace income_pct_w1 =  8  if  income_pct_w1_`c'== 8
        replace income_pct_w1 =  9  if  income_pct_w1_`c'== 9
        replace income_pct_w1 = 10  if  income_pct_w1_`c'==10
    }

    lab var income_pct_w1 "Hhd income percentiles wave 1"
    lab val income_pct_w1 income_pct_w1

    keep mergeid income_pct_w1

save $easy\data\temp\sharew1_gv_imputations_a.dta, replace


*>> w2:

use $easy\data\release\sharew2_rel7-0-0_gv_imputations.dta, clear

  * mergeing with cv_r because int_year is needed:
    merge m:1 mergeid using "$easy\data\release\sharew2_rel7-0-0_cv_r.dta"
    drop if interview ==0
    drop if deceased == 1

  * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen thinc_mod = mean(thinc)
    replace thinc = thinc_mod

  * We only keep one out of the five implicats:
    keep if implicat==1


  * Income
  * thinc: household total net income (previous year)
    gen     thinc_m=thinc_mod*exrate         // euro amounts into
                                             // local currency
              // the income measures going into thinc are mostly
              // based on the previous year
    replace thinc_m=thinc_m/pppk2005 if int_year==2006
    replace thinc_m=thinc_m/pppk2006 if int_year==2007
    replace thinc_m=thinc_m/pppk2007 if int_year==2008
    replace thinc_m=thinc_m/pppk2008 if int_year==2009
    replace thinc_m=thinc_m/pppk2009 if int_year==2010


  * country-specific income percentiles
    foreach c in 11 12 13 14 15 16 17 18 19 20 23 25 28 29 {
        pctile  p10_`c' = thinc_m if country==`c', nq(10)
        return list
        gen     income_pct_w2_`c'=.
        replace income_pct_w2_`c'= 1 if thinc_m<=r(r1)                  ///
                                      & country==`c'
        replace income_pct_w2_`c'= 2 if thinc_m<=r(r2) & thinc_m>r(r1) ///
```

```
                                    & country==`c'
        replace income_pct_w2_`c'= 3 if thinc_m<=r(r3) & thinc_m>r(r2) ///
                                    & country==`c'
        replace income_pct_w2_`c'= 4 if thinc_m<=r(r4) & thinc_m>r(r3) ///
                                    & country==`c'
        replace income_pct_w2_`c'= 5 if thinc_m<=r(r5) & thinc_m>r(r4) ///
                                    & country==`c'
        replace income_pct_w2_`c'= 6 if thinc_m<=r(r6) & thinc_m>r(r5) ///
                                    & country==`c'
        replace income_pct_w2_`c'= 7 if thinc_m<=r(r7) & thinc_m>r(r6) ///
                                    & country==`c'
        replace income_pct_w2_`c'= 8 if thinc_m<=r(r8) & thinc_m>r(r7) ///
                                    & country==`c'
        replace income_pct_w2_`c'= 9 if thinc_m<=r(r9) & thinc_m>r(r8) ///
                                    & country==`c'
        replace income_pct_w2_`c'=10 if thinc_m> r(r9) & thinc_m<.     ///
                                    & country==`c'
        lab val income_pct_w2_`c' income_pct`c'
    }

  * one variable including all countries
    gen income_pct_w2 = -7 if country==. | country==30
    foreach c in 11 12 13 14 15 16 17 18 19 20 23 25 28 29 {
        replace income_pct_w2 =  1  if  income_pct_w2_`c'== 1
        replace income_pct_w2 =  2  if  income_pct_w2_`c'== 2
        replace income_pct_w2 =  3  if  income_pct_w2_`c'== 3
        replace income_pct_w2 =  4  if  income_pct_w2_`c'== 4
        replace income_pct_w2 =  5  if  income_pct_w2_`c'== 5
        replace income_pct_w2 =  6  if  income_pct_w2_`c'== 6
        replace income_pct_w2 =  7  if  income_pct_w2_`c'== 7
        replace income_pct_w2 =  8  if  income_pct_w2_`c'== 8
        replace income_pct_w2 =  9  if  income_pct_w2_`c'== 9
        replace income_pct_w2 = 10  if  income_pct_w2_`c'==10
    }

    lab var income_pct_w2 "Hhd income percentiles wave 2"
    lab val income_pct_w2 income_pct_w2
    lab def income_pct_w2 -7 "not yet coded", add

    keep mergeid income_pct_w2 thinc_m

save $easy\data\temp\sharew2_gv_imputations_a.dta, replace


*>> w4:

use $easy\data\release\sharew4_rel7-0-0_gv_imputations.dta, clear

  * mergeing with cv_r because int_year is needed:
    merge m:1 mergeid using "$easy\data\release\sharew4_rel7-0-0_cv_r.dta"
    drop if interview ==0
    drop if deceased == 1

  * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen thinc_mod = mean(thinc)
    replace thinc = thinc_mod

  * We only keep one out of the five implicats:
    keep if implicat==1
```

```
  * Income
  * thinc: household total net income (previous year)
    gen     thinc_m=thinc_mod*exrate         // euro amounts into
                                             // local currency
                // even though Estonia changed to Euro in 2011
                // the income measures going into thinc are mostly
                // based on the previous year, i.e. a non-Euro year
    replace thinc_m=thinc_m/pppk2010 if  int_year==2010
    replace thinc_m=thinc_m/pppk2010 if  int_year==2011
    replace thinc_m=thinc_m/pppk2011 if  int_year==2012

    *country-specific income percentiles
    foreach c in 11 12 13 14 15 16 17 18 20 23 28 29 32 33 34 35 {
        pctile  p10_`c' = thinc_m if country==`c', nq(10)
        return list
        gen     income_pct_w4_`c'= 1 if thinc_m<=r(r1)                  ///
                                        & country==`c'
        replace income_pct_w4_`c'= 2 if thinc_m<=r(r2) & thinc_m>r(r1) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 3 if thinc_m<=r(r3) & thinc_m>r(r2) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 4 if thinc_m<=r(r4) & thinc_m>r(r3) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 5 if thinc_m<=r(r5) & thinc_m>r(r4) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 6 if thinc_m<=r(r6) & thinc_m>r(r5) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 7 if thinc_m<=r(r7) & thinc_m>r(r6) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 8 if thinc_m<=r(r8) & thinc_m>r(r7) ///
                                        & country==`c'
        replace income_pct_w4_`c'= 9 if thinc_m<=r(r9) & thinc_m>r(r8) ///
                                        & country==`c'
        replace income_pct_w4_`c'=10 if thinc_m>r(r9)                   ///
                                        & country==`c'
        lab val income_pct_w4_`c' income_pct`c'
    }

    *one variable including all countries
    gen income_pct_w4 =.
    foreach c in 11 12 13 14 15 16 17 18 20 23 28 29 32 33 34 35 {
        replace income_pct_w4 =  1  if  income_pct_w4_`c'== 1
        replace income_pct_w4 =  2  if  income_pct_w4_`c'== 2
        replace income_pct_w4 =  3  if  income_pct_w4_`c'== 3
        replace income_pct_w4 =  4  if  income_pct_w4_`c'== 4
        replace income_pct_w4 =  5  if  income_pct_w4_`c'== 5
        replace income_pct_w4 =  6  if  income_pct_w4_`c'== 6
        replace income_pct_w4 =  7  if  income_pct_w4_`c'== 7
        replace income_pct_w4 =  8  if  income_pct_w4_`c'== 8
        replace income_pct_w4 =  9  if  income_pct_w4_`c'== 9
        replace income_pct_w4 = 10  if  income_pct_w4_`c'==10
    }

    lab var income_pct_w4 "Hhd income percentiles wave 4"
    lab val income_pct_w4 income_pct_w4
    label def income_pct_w4 -13 "not asked in this wave"
    format income_pct_w4 %38.0g

    keep mergeid income_pct_w4 thinc_m

save $easy\data\temp\sharew4_gv_imputations_a.dta, replace
```

```
*>> w5:

use $easy\data\release\sharew5_rel7-0-0_gv_imputations.dta, clear

  * mergeing with cv_r because int_year is needed:
    merge m:1 mergeid using "$easy\data\release\sharew5_rel7-0-0_cv_r.dta"
    drop if interview ==0
    drop if deceased == 1

  * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen thinc_mod = mean(thinc)
    replace thinc = thinc_mod

  * We only keep one out of the five implicats:
    keep if implicat==1

    * Income
    * thinc: household total net income (previous year)
    gen     thinc_m=thinc_mod*exrate
    replace thinc_m=thinc_m/pppk2012 if  int_year==2013

    *country-specific income percentiles
    foreach c in 11 12 13 14 15 16 17 18 20 23 25 28 31 34 35 {
        pctile  p10_`c' = thinc_m if country==`c', nq(10)
        return list
        gen     income_pct_w5_`c'= 1 if thinc_m<=r(r1)                 ///
                                      & country==`c'
        replace income_pct_w5_`c'= 2 if thinc_m<=r(r2) & thinc_m>r(r1) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 3 if thinc_m<=r(r3) & thinc_m>r(r2) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 4 if thinc_m<=r(r4) & thinc_m>r(r3) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 5 if thinc_m<=r(r5) & thinc_m>r(r4) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 6 if thinc_m<=r(r6) & thinc_m>r(r5) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 7 if thinc_m<=r(r7) & thinc_m>r(r6) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 8 if thinc_m<=r(r8) & thinc_m>r(r7) ///
                                      & country==`c'
        replace income_pct_w5_`c'= 9 if thinc_m<=r(r9) & thinc_m>r(r8) ///
                                      & country==`c'
        replace income_pct_w5_`c'=10 if thinc_m>r(r9)                  ///
                                      & country==`c'
        lab val income_pct_w5_`c' income_pct`c'
    }

    *one variable including all countries
    gen income_pct_w5 =.
    foreach c in 11 12 13 14 15 16 17 18 20 23 25 28 31 34 35 {
        replace income_pct_w5 =  1  if  income_pct_w5_`c'== 1
        replace income_pct_w5 =  2  if  income_pct_w5_`c'== 2
        replace income_pct_w5 =  3  if  income_pct_w5_`c'== 3
        replace income_pct_w5 =  4  if  income_pct_w5_`c'== 4
        replace income_pct_w5 =  5  if  income_pct_w5_`c'== 5
        replace income_pct_w5 =  6  if  income_pct_w5_`c'== 6
        replace income_pct_w5 =  7  if  income_pct_w5_`c'== 7
        replace income_pct_w5 =  8  if  income_pct_w5_`c'== 8
        replace income_pct_w5 =  9  if  income_pct_w5_`c'== 9
```

```
        replace income_pct_w5 = 10  if  income_pct_w5_`c'==10
    }

    lab var income_pct_w5 "Hhd income percentiles wave 5"
    lab val income_pct_w5 income_pct_w5
    label def income_pct_w5 -13 "not asked in this wave"
    format income_pct_w5 %38.0g

    keep mergeid income_pct_w5 thinc_m

save $easy\data\temp\sharew5_gv_imputations_a.dta, replace


*>> w6:

use $easy\data\release\sharew6_rel7-0-0_gv_imputations.dta, clear

  * mergeing with cv_r because int_year is needed:
    merge m:1 mergeid using "$easy\data\release\sharew6_rel7-0-0_cv_r.dta"
    drop if interview ==0
    drop if deceased == 1

  * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen thinc_mod = mean(thinc)
    replace thinc = thinc_mod

  * We only keep one out of the five implicats:
    keep if implicat==1

    * Income
    * thinc: household total net income (previous year)
    gen     thinc_m=thinc_mod*exrate
    replace thinc_m=thinc_m/pppk2014 if  int_year==2015

    *country-specific income percentiles
    foreach c in 11 12 13 15 16 17 18 19 20 23 25 28 29 31 33 34 35 47 {
        pctile  p10_`c' = thinc_m if country==`c', nq(10)
        return list
        gen     income_pct_w6_`c'= 1 if thinc_m<=r(r1)                 ///
                                      & country==`c'
        replace income_pct_w6_`c'= 2 if thinc_m<=r(r2) & thinc_m>r(r1) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 3 if thinc_m<=r(r3) & thinc_m>r(r2) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 4 if thinc_m<=r(r4) & thinc_m>r(r3) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 5 if thinc_m<=r(r5) & thinc_m>r(r4) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 6 if thinc_m<=r(r6) & thinc_m>r(r5) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 7 if thinc_m<=r(r7) & thinc_m>r(r6) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 8 if thinc_m<=r(r8) & thinc_m>r(r7) ///
                                      & country==`c'
        replace income_pct_w6_`c'= 9 if thinc_m<=r(r9) & thinc_m>r(r8) ///
                                      & country==`c'
        replace income_pct_w6_`c'=10 if thinc_m>r(r9)                  ///
                                      & country==`c'
        lab val income_pct_w6_`c' income_pct`c'
    }
```

```
    *one variable including all countries
    gen income_pct_w6 =.
    foreach c in 11 12 13 15 16 17 18 19 20 23 25 28 29 31 33 34 35 47 {
        replace income_pct_w6 =  1  if  income_pct_w6_`c'== 1
        replace income_pct_w6 =  2  if  income_pct_w6_`c'== 2
        replace income_pct_w6 =  3  if  income_pct_w6_`c'== 3
        replace income_pct_w6 =  4  if  income_pct_w6_`c'== 4
        replace income_pct_w6 =  5  if  income_pct_w6_`c'== 5
        replace income_pct_w6 =  6  if  income_pct_w6_`c'== 6
        replace income_pct_w6 =  7  if  income_pct_w6_`c'== 7
        replace income_pct_w6 =  8  if  income_pct_w6_`c'== 8
        replace income_pct_w6 =  9  if  income_pct_w6_`c'== 9
        replace income_pct_w6 = 10  if  income_pct_w6_`c'==10
    }

    lab var income_pct_w6 "Hhd income percentiles wave 6"
    lab val income_pct_w6 income_pct_w6
    label def income_pct_w6 -13 "not asked in this wave"
    format income_pct_w6 %38.0g

    keep mergeid income_pct_w6 thinc_m

save $easy\data\temp\sharew6_gv_imputations_a.dta, replace


*>> w7:

use $easy\data\release\sharew7_rel7-0-0_gv_imputations.dta, clear

  * mergeing with cv_r and technical_variables module is needed:
    merge m:1 mergeid using "$easy\data\release\sharew7_rel7-0-0_cv_r.dta"
    drop if interview ==0
    drop if deceased == 1
    merge m:1 mergeid using ///
          "$easy\data\release\sharew7_rel7-0-0_technical_variables.dta", nogen


    * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen thinc_mod = mean(thinc)
    replace thinc = thinc_mod


  * We only keep one out of the five implicats:
    keep if implicat==1

    * We only keep panel interviews
    *keep if mn103_==0

    * Income
    * thinc: household total net income (previous year)
    gen     thinc_m=thinc_mod*exrate
    replace thinc_m=thinc_m/pppk2016 if  int_year==2017
    replace thinc_m=thinc_m/pppk2017 if  int_year==2018
    replace thinc_m=-10 if mn103_==1 // not asked to SHARELIFE respondents


    *country-specific income percentiles; only non-SHARELIFE countries included
    foreach c in 11 12 13 15 16 17 18 19 20 23 28 29  {
        pctile  p10_`c' = thinc_m if country==`c', nq(10)
```

```
        return list
        gen     income_pct_w7_`c'= 1 if thinc_m<=r(r1)                  ///
                                        & country==`c'
        replace income_pct_w7_`c'= 2 if thinc_m<=r(r2) & thinc_m>r(r1) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 3 if thinc_m<=r(r3) & thinc_m>r(r2) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 4 if thinc_m<=r(r4) & thinc_m>r(r3) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 5 if thinc_m<=r(r5) & thinc_m>r(r4) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 6 if thinc_m<=r(r6) & thinc_m>r(r5) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 7 if thinc_m<=r(r7) & thinc_m>r(r6) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 8 if thinc_m<=r(r8) & thinc_m>r(r7) ///
                                        & country==`c'
        replace income_pct_w7_`c'= 9 if thinc_m<=r(r9) & thinc_m>r(r8) ///
                                        & country==`c'
        replace income_pct_w7_`c'=10 if thinc_m>r(r9) & thinc_m!=.     ///
                                        & country==`c'
        lab val income_pct_w7_`c' income_pct`c'
    }

    *one variable including all countries
    gen income_pct_w7 =.
    foreach c in 11 12 13 15 16 17 18 19 20 23 28 29 {
        replace income_pct_w7 =  1  if  income_pct_w7_`c'== 1
        replace income_pct_w7 =  2  if  income_pct_w7_`c'== 2
        replace income_pct_w7 =  3  if  income_pct_w7_`c'== 3
        replace income_pct_w7 =  4  if  income_pct_w7_`c'== 4
        replace income_pct_w7 =  5  if  income_pct_w7_`c'== 5
        replace income_pct_w7 =  6  if  income_pct_w7_`c'== 6
        replace income_pct_w7 =  7  if  income_pct_w7_`c'== 7
        replace income_pct_w7 =  8  if  income_pct_w7_`c'== 8
        replace income_pct_w7 =  9  if  income_pct_w7_`c'== 9
        replace income_pct_w7 = 10  if  income_pct_w7_`c'==10
    }

    lab var income_pct_w7 "Hhd income percentiles wave 7"
    lab val income_pct_w7 income_pct_w7
    label def income_pct_w7 -13 "not asked in this wave"

    format income_pct_w7 %38.0g

    keep mergeid income_pct_w7 thinc_m

save $easy\data\temp\sharew7_gv_imputations_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[26. Extract & Recode Variables from GV_Grossnett]-----------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_gv_grossnet.dta, clear
    merge 1:m mergeid using $easy\data\release\sharew1_rel7-0-0_gv_imputations.dta

  * Mean value for HH-income over the 5 implicates
    bysort mergeid: egen hhytotn_mod = mean(hhytotn)
    replace hhytotn = hhytotn_mod

  * We only keep one out of the five implicats:
    keep if implicat==1

  * Income
  * thinc: household total net income (previous year)
    gen     thinc_m=hhytotn_mod*exrate
    replace thinc_m=thinc_m/pppk2003 if country != 25
    replace thinc_m=thinc_m/pppk2004 if country == 25

    keep mergeid thinc_m

save $easy\data\temp\sharew1_gv_grossnet_a.dta, replace



*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[27. Extract & Recode Variables from DROPOFF]----------------------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_dropoff.dta, clear

    mvdecode q4* q2*, mv(-5=.c)

  * Quality of Life Score(CASP-12)

    gen     q2_d_inv=.
    replace q2_d_inv=1 if q2_d==4
    replace q2_d_inv=2 if q2_d==3
    replace q2_d_inv=3 if q2_d==2
    replace q2_d_inv=4 if q2_d==1

    gen     q2_g_inv=.
    replace q2_g_inv=1 if q2_g==4
    replace q2_g_inv=2 if q2_g==3
    replace q2_g_inv=3 if q2_g==2
    replace q2_g_inv=4 if q2_g==1

    gen     q2_h_inv=.
    replace q2_h_inv=1 if q2_h==4
    replace q2_h_inv=2 if q2_h==3
    replace q2_h_inv=3 if q2_h==2
```

```
    replace q2_h_inv=4 if q2_h==1

    gen     q2_i_inv=.
    replace q2_i_inv=1 if q2_i==4
    replace q2_i_inv=2 if q2_i==3
    replace q2_i_inv=3 if q2_i==2
    replace q2_i_inv=4 if q2_i==1

    gen     q2_j_inv=.
    replace q2_j_inv=1 if q2_j==4
    replace q2_j_inv=2 if q2_j==3
    replace q2_j_inv=3 if q2_j==2
    replace q2_j_inv=4 if q2_j==1

    gen     q2_k_inv=.
    replace q2_k_inv=1 if q2_k==4
    replace q2_k_inv=2 if q2_k==3
    replace q2_k_inv=3 if q2_k==2
    replace q2_k_inv=4 if q2_k==1

    gen     q2_l_inv=.
    replace q2_l_inv=1 if q2_l==4
    replace q2_l_inv=2 if q2_l==3
    replace q2_l_inv=3 if q2_l==2
    replace q2_l_inv=4 if q2_l==1

    // Subscale Control
    gen con= q2_a + q2_b + q2_c
    alpha    q2_a   q2_b   q2_c

    // Subscale Autonomy
    gen aut= q2_d_inv + q2_e + q2_f
    alpha    q2_d_inv q2_e q2_f

    // Subscale Pleasure
    gen ple= q2_g_inv + q2_h_inv + q2_i_inv
    alpha    q2_g_inv   q2_h_inv   q2_i_inv

    // Subscale Self-Realisation
    gen sel= q2_j_inv + q2_k_inv + q2_l_inv
    alpha    q2_j_inv   q2_k_inv   q2_l_inv

    gen     casp = con + aut + ple + sel
    lab var casp  "CASP quality of life (high is high quality)"

    mvencode q4* q2*, mv(.c=-5)

    keep mergeid casp q34_re

save $easy\data\temp\sharew1_dropoff_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[28. Extract & Recode Variables from Technical Variables]-----------------

*>> w1:

use $easy\data\release\sharew1_rel7-0-0_technical_variables.dta, clear
save $easy\data\temp\sharew1_technical_variables_a.dta, replace

*>> w2:

use $easy\data\release\sharew2_rel7-0-0_technical_variables.dta, clear
save $easy\data\temp\sharew2_technical_variables_a.dta, replace

*>> w4:

use $easy\data\release\sharew4_rel7-0-0_technical_variables.dta, clear
save $easy\data\temp\sharew4_technical_variables_a.dta, replace

*>> w5:

use $easy\data\release\sharew5_rel7-0-0_technical_variables.dta, clear
save $easy\data\temp\sharew5_technical_variables_a.dta, replace

*>> w6:

use $easy\data\release\sharew6_rel7-0-0_technical_variables.dta, clear
save $easy\data\temp\sharew6_technical_variables_a.dta, replace

*>> w7:

use $easy\data\release\sharew7_rel7-0-0_technical_variables.dta, clear
save $easy\data\temp\sharew7_technical_variables_a.dta, replace


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[29. Merge modules per wave ]---------------------------------------------

// We use the CV_R modules as master and then merge the other modules.


*>> w1:

use $easy\data\temp\sharew1_cv_r_a.dta, clear

  merge 1:1 mergeid using $easy\data\temp\sharew1_dn_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_ac_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_br_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_cf_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_ch_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_co_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_ep_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_hc_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_ho_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_ph_a.dta,assert(  3) nogen
```

```
  merge 1:1 mergeid using $easy\data\temp\sharew1_sp_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_gv_health_a.dta        ///
                                                          ,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_gv_isced_a.dta         ///
                                                          ,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_dropoff_a.dta          ///
                                    ,  gen(merge_w1_dropoff) assert(1 3)
                                  // not every respondent has linked dropooff
  merge 1:1 mergeid using $easy\data\temp\sharew1_gv_imputations_a.dta   ///
                                                          ,assert(1 3) nogen
    // specific imputated variables we want for easySHARE are not available
    // for IL (i.e. not included in "imputations_ilextra" module)
  merge 1:1 mergeid using $easy\data\temp\sharew1_iv_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew1_gv_grossnet_a.dta      ///
                                                          ,assert(1 3) nogen

  merge 1:1 mergeid using $easy\data\temp\sharew1_technical_variables_a.dta ///
                                                          ,assert(  3) nogen


save $easy\data\temp\sharew1_merged_a.dta, replace


*>> w2:

use $easy\data\temp\sharew2_cv_r_a.dta, clear

  merge 1:1 mergeid using $easy\data\temp\sharew2_dn_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_ac_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_br_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_cf_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_ch_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_co_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_ep_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_hc_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_ho_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_ph_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_sp_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_gv_health_a.dta        ///
                                                          ,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_gv_isced_a.dta         ///
                                                          ,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_gv_imputations_a.dta      ///
                                                          ,assert(1 3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_iv_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew2_technical_variables_a.dta ///
                                                          ,assert(  3) nogen

save $easy\data\temp\sharew2_merged_a.dta, replace


*>> w3:

use $easy\data\temp\sharew3_cv_r_a.dta, clear

   merge 1:1 mergeid using $easy\data\temp\sharew3_st_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew3_rp_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew3_cs_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew3_hc_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew3_hs_a.dta,assert(3) nogen
```

```
   merge 1:1 mergeid using $easy\data\temp\sharew3_gs_a.dta,assert(3) nogen

save $easy\data\temp\sharew3_merged_a.dta, replace


*>> w4:

use $easy\data\temp\sharew4_cv_r_a.dta, clear

  merge 1:1 mergeid using $easy\data\temp\sharew4_dn_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_ac_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_br_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_cf_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_ch_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_co_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_ep_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_hc_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_ho_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_ph_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_sn_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_sp_a.dta,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_gv_health_a.dta   ///
                                                          ,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_gv_isced_a.dta    ///
                                                          ,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_gv_imputations_a.dta ///
                                                          ,assert(3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_iv_a.dta,assert(  3) nogen
  merge 1:1 mergeid using $easy\data\temp\sharew4_technical_variables_a.dta ///
                                                          ,assert(  3) nogen

save $easy\data\temp\sharew4_merged_a.dta, replace


*>> w5:

use $easy\data\temp\sharew5_cv_r_a.dta, clear

   merge 1:1 mergeid using $easy\data\temp\sharew5_dn_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_ac_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_br_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_cf_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_ch_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_co_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_ep_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_hc_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_ho_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_mc_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_ph_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_sp_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_gv_health_a.dta   ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_gv_isced_a.dta    ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_gv_imputations_a.dta ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_iv_a.dta,assert(  3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew5_technical_variables_a.dta ///
                                                          ,assert(  3) nogen
```

```
save $easy\data\temp\sharew5_merged_a.dta, replace


*>> w6:

use $easy\data\temp\sharew6_cv_r_a.dta, clear

   merge 1:1 mergeid using $easy\data\temp\sharew6_dn_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_ac_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_br_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_cf_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_ch_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_co_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_ep_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_hc_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_ho_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_ph_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_sn_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_sp_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_gv_health_a.dta   ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_gv_isced_a.dta    ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_gv_imputations_a.dta ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_iv_a.dta,assert(  3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew6_technical_variables_a.dta ///
                                                          ,assert(  3) nogen

save $easy\data\temp\sharew6_merged_a.dta, replace

*>> w7:

use $easy\data\temp\sharew7_cv_r_a.dta, clear

   merge 1:1 mergeid using $easy\data\temp\sharew7_dn_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_ac_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_br_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_cc_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_cf_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_ch_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_co_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_ep_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_hc_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_ho_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_hs_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_ph_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_rh_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_sp_a.dta,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_gv_health_a.dta   ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_gv_isced_a.dta    ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_gv_imputations_a.dta ///
                                                          ,assert(3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_iv_a.dta,assert(  3) nogen
   merge 1:1 mergeid using $easy\data\temp\sharew7_technical_variables_a.dta ///
                                                          ,assert(  3) nogen

save $easy\data\temp\sharew7_merged_a.dta, replace
```

```
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[30. Other recodes per wave ]---------------------------------------------

*>> w1:

use $easy\data\temp\sharew1_merged_a.dta, clear

*>> Assign (grand)children information to partner of family respondent
    gsort coupleid1 -fam_resp
    replace ch001_  = ch001_[_n-1]       ///
                                if coupleid1==coupleid1[_n-1] & coupleid1!=""
    replace ch007_hh= ch007_hh[_n-1]     ///
                                if coupleid1==coupleid1[_n-1] & coupleid1!=""
    replace ch007_km= ch007_km[_n-1]     ///
                                if coupleid1==coupleid1[_n-1] & coupleid1!=""
    replace ch021_mod  = ch021_mod[_n-1] ///
                                if coupleid1==coupleid1[_n-1] & coupleid1!=""

*>> Assign variables on social support (received and given) to partner
    foreach var in sp002_ sp003_1 sp003_2 sp003_3 {
        replace `var' = `var'[_n-1] if coupleid1==coupleid1[_n-1] ///
                                    & coupleid1!="" & fam_resp==0 ///
                                    & fam_resp[_n-1]==1 & `var'==.
        replace `var' = `var'[_n-1] if coupleid1==coupleid1[_n-1] ///
                                    & coupleid1!="" & fam_resp==0 ///
                                    & `var'==. & `var'[_n-1]!=.
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

    foreach var in sp003_1 sp003_2 sp003_3 {
        replace `var' = -9 if sp002_== 5
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }
    lab def relative 10 "child" -9 "filtered: no help received/given", modify

    foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        rename `var' `var'_mod
    }

    foreach var in sp002_ {
        rename `var' `var'mod
    }

*>> Household able to make ends meet
    gen     co007a   =      co007_
    replace co007a   = . if co007a < 0
    egen    co007_sd =   sd(co007_), by(hhid1)
    replace co007a   = . if co007_sd > 0  &  co007_sd < .
    egen    co007b   =  min(co007a),by(hhid1)
    replace co007_   = co007b if co007b != .
    drop co007a co007b co007_sd

*>> Assign information on area to other household members
```

```
    gsort hhid1 -hou_resp
    gen iv009_mod     = iv009_
    replace iv009_mod = -9 if iv009_==. & ho001 == 5
    replace iv009_mod = iv009_[_n-1] if iv009_mod==. & hhid1==hhid1[_n-1]
    lab var iv009_mod "Area of location"
    lab val iv009_mod arealocated
    lab def arealocated -9 "filtered: interview not at home", modify

save $easy\data\temp\sharew1_merged_b.dta, replace



*>> w2:

use $easy\data\temp\sharew2_merged_a.dta, clear

*>> Assign (grand)children information to partner of family respondent
    gsort coupleid2 -fam_resp
    replace ch001_  =ch001_[_n-1]       ///
                                if coupleid2==coupleid2[_n-1] & coupleid2!=""
    replace ch007_hh=ch007_hh[_n-1]     ///
                                if coupleid2==coupleid2[_n-1] & coupleid2!=""
    replace ch007_km=ch007_km[_n-1]     ///
                                if coupleid2==coupleid2[_n-1] & coupleid2!=""
    replace ch021_mod=ch021_mod[_n-1]   ///
                                if coupleid2==coupleid2[_n-1] & coupleid2!=""

*>> Assign variables on received social support to partner
    foreach var in sp002_ sp003_1 sp003_2 sp003_3 {
        replace `var' = `var'[_n-1] if coupleid2==coupleid2[_n-1] ///
                                    & coupleid2!="" & fam_resp==0 ///
                                    & fam_resp[_n-1]==1 & `var'==.
        replace `var' = `var'[_n-1] if coupleid2==coupleid2[_n-1] ///
                                    & coupleid2!="" & fam_resp==0  ///
                                    & `var'==. & `var'[_n-1]!=.
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

    foreach var in sp003_1 sp003_2 sp003_3 {
        replace `var' = -9 if sp002_== 5
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }
    lab def relative 10 "child" -9 "filtered: no help received/given", modify

    foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        rename `var' `var'_mod
    }

    foreach var in sp002_ {
        rename `var' `var'mod
    }

*>> Household able to make ends meet
    gen     co007a   = co007_
    replace co007a   = . if co007a < 0
    egen    co007_sd = sd(co007_), by(hhid2)
```

```
    replace co007a   = . if co007_sd > 0  &  co007_sd < .
    egen    co007b   = min(co007a),by(hhid2)
    replace co007_   = co007b if co007b != .
    drop co007a co007b co007_sd


*>> Assign information on area to other household members
    gsort hhid2 -hou_resp
    gen iv009_mod     = iv009_
    replace iv009_mod = -9 if iv009_==. & [ho001 == 5 | mn024 ==2]
    replace iv009_mod = iv009_[_n-1] if iv009_mod==. & hhid2==hhid2[_n-1]
    lab var iv009_mod "Area of location"
    lab val iv009_mod arealocated
    lab def arealocated -9 "filtered: interview not at home", modify


save $easy\data\temp\sharew2_merged_b.dta, replace



*>> w3:

use $easy\data\temp\sharew3_merged_a.dta, clear

*>> Fix gender missing data problems in w3/SHARELIFE
    // The problem comes form 66 missing lines in the cv_r dataset
    replace female = 0 if female==. & sl_st011_ == 1
    replace female = 1 if female==. & sl_st011_ == 2

save $easy\data\temp\sharew3_merged_b.dta, replace



*>> w4:

use $easy\data\temp\sharew4_merged_a.dta, clear

*>> Fill up ch007_hh & ch007_km with infiormation from sn006_

    foreach n in 1 2 3 4 5 6 7 {
        replace ch007_hh = 1 if [ch007_hh == 0 | ch007_hh==.] &    ///
         [sn005_`n'>= 10 & sn005_`n' <= 13] & [sn006_`n' == 1 | sn006_`n' ==2]
        replace ch007_hh = 0 if ch007_hh==. &        ///
         [sn005_`n'>= 10 & sn005_`n' <= 13]  & sn006_`n' > 2 & sn006_`n' <.
    }

    foreach n in 1 2 3 4 5 6 7 {
        replace ch007_km = 1 if [ch007_km == 0 | ch007_km==.] &    ///
         [sn005_`n'>= 10 & sn005_`n' <= 13] & sn006_`n' == 3
        replace ch007_km = 0 if ch007_km==. &        ///
         [sn005_`n'>= 10 & sn005_`n' <= 13] & sn006_`n' > 3 & sn006_`n' <.
    }
        replace ch007_km = 1 if ch007_hh == 1


*>> Assign (grand)children information to partner of family respondent
    gsort   coupleid4 -fam_resp
    replace ch001_  =ch001_[_n-1]     ///
                              if coupleid4==coupleid4[_n-1] & coupleid4!=""
    replace ch007_hh=ch007_hh[_n-1]   ///
```

```
                                if coupleid4==coupleid4[_n-1] & coupleid4!=""
    replace ch007_km=ch007_km[_n-1]   ///
                                if coupleid4==coupleid4[_n-1] & coupleid4!=""
    replace ch021_mod=ch021_mod[_n-1] ///
                                if coupleid4==coupleid4[_n-1] & coupleid4!=""

*>> Assign information on received social support to partner
    foreach var in sp002_ sp003_1 sp003_2 sp003_3 {
        replace `var' = `var'[_n-1] if coupleid4==coupleid4[_n-1] ///
                                    & coupleid4!="" & fam_resp==0 ///
                                    & fam_resp[_n-1]==1 & `var'==.
        replace `var' = `var'[_n-1] if coupleid4==coupleid4[_n-1] ///
                                    & coupleid4!="" & fam_resp==0 ///
                                    & `var'==. & `var'[_n-1]!=.
    }


*>> Adapt the values in sn005_1 - sn005_7 to the codes in sp003* & sp009*
     foreach var in sn005_1 sn005_2 sn005_3 sn005_4 sn005_5 sn005_6 sn005_7 {
        replace `var' = 10 if `var' == 11
        replace `var' = 20 if `var' == 12
        replace `var' = 21 if `var' == 13
        replace `var' = 22 if `var' == 14
        replace `var' = 23 if `var' == 15
        replace `var' = 24 if `var' == 16
        replace `var' = 25 if `var' == 17
        replace `var' = 26 if `var' == 18
        replace `var' = 27 if `var' == 19
        replace `var' = 28 if `var' == 20
        replace `var' = 29 if `var' == 21
        replace `var' = 30 if `var' == 22
        replace `var' = 31 if `var' == 23
        replace `var' = 32 if `var' == 24
        replace `var' = 33 if `var' > 24 & `var' <=96
        }

*>> Include information from the list of sn members in sp003* and sp009*
         foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        replace `var' = sn005_1 if `var'== 101
        replace `var' = sn005_2 if `var'== 102
        replace `var' = sn005_3 if `var'== 103
        replace `var' = sn005_4 if `var'== 104
        replace `var' = sn005_5 if `var'== 105
        replace `var' = sn005_6 if `var'== 106
        replace `var' = sn005_7 if `var'== 107
        replace `var' = 10              if `var'==  19
        replace `var' = 33              if `var'>   33 & `var'< 101
    }

    foreach var in sp003_1 sp003_2 sp003_3 {
        replace `var' = -9 if sp002_== 5
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

        lab def relative 10 "child" 33 "other acquaintance" ///
        -9 "filtered: no help received/given", modify
```

```
    foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        rename `var' `var'_mod
    }

    foreach var in sp002_ {
        rename `var' `var'mod
    }

*>> Household able to make ends meet
    gen     co007a   = co007_
    replace co007a   = . if co007a < 0
    egen    co007_sd = sd(co007_), by(hhid4)
    replace co007a   = . if co007_sd > 0  &  co007_sd < .
    egen    co007b   = min(co007a),by(hhid4)
    replace co007_   = co007b if co007b != .
    drop    co007a co007b co007_sd

*>> Assign information on area to other household members
      gsort hhid4 -hou_resp
      gen iv009_mod         = iv009_
      replace iv009_mod = -9 if iv009_==. & [ho001 == 5 | mn024 ==2]
      replace iv009_mod = iv009_[_n-1] if iv009_mod==. & hhid4==hhid4[_n-1]
    lab var iv009_mod "Area of location"
      lab val iv009_mod arealocated
      lab def arealocated -9 "filtered: interview not at home", modify

save $easy\data\temp\sharew4_merged_b.dta, replace



*>> w5:

use $easy\data\temp\sharew5_merged_a.dta, clear

*>> Assign (grand)children information to partner of family respondent
    gsort   coupleid5 -fam_resp
    replace ch001_  =ch001_[_n-1]     ///
                              if coupleid5==coupleid5[_n-1] & coupleid5!=""
    replace ch007_hh=-9 if ch001==0 & coupleid5==coupleid5[_n-1] & coupleid5!=""
    replace ch007_hh=ch007_hh[_n-1]   ///
                              if coupleid5==coupleid5[_n-1] & coupleid5!=""
    replace ch007_hh=-9 if ch001==0 & coupleid5==coupleid5[_n-1] & coupleid5!=""
    replace ch007_km=ch007_km[_n-1]   ///
                              if coupleid5==coupleid5[_n-1] & coupleid5!=""
    replace ch021_mod=ch021_mod[_n-1] ///
                              if coupleid5==coupleid5[_n-1] & coupleid5!=""

*>> Assign variables on received social support to partner
    foreach var in sp002_ sp003_1 sp003_2 sp003_3 {
        replace `var' = `var'[_n-1] if coupleid5==coupleid5[_n-1] ///
                                    & coupleid5!="" & fam_resp==0 ///
                                    & fam_resp[_n-1]==1 & `var'==.
        replace `var' = `var'[_n-1] if coupleid5==coupleid5[_n-1] ///
                                    & coupleid5!="" & fam_resp==0  ///
                                    & `var'==. & `var'[_n-1]!=.
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }
```

```
    foreach var in sp003_1 sp003_2 sp003_3 {
        replace `var' = -9 if sp002_== 5
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

    foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        rename `var' `var'_mod
    }

    foreach var in sp002_ {
        rename `var' `var'mod
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }
    lab def relative 10 "child" -9 "filtered: no help received/given", modify

*>> Household able to make ends meet
    gen     co007a   = co007_
    replace co007a   = . if co007a < 0
    egen    co007_sd = sd(co007_), by(hhid5)
    replace co007a   = . if co007_sd > 0  &  co007_sd < .
    egen    co007b   = min(co007a),by(hhid5)
    replace co007_   = co007b if co007b != .
    drop    co007a co007b co007_sd

*>> Assign information on area to other household members
    gsort hhid5 -hou_resp
    gen iv009_mod     = iv009_
    replace iv009_mod = -9 if iv009_==. & [ho001 == 5 | mn024 ==2]
    replace iv009_mod = iv009_[_n-1] if iv009_mod==. & hhid5==hhid5[_n-1]
    lab var iv009_mod "Area of location"
    lab val iv009_mod arealocated
    lab def arealocated -9 "filtered: interview not at home", modify

save $easy\data\temp\sharew5_merged_b.dta, replace



*>> w6:

use $easy\data\temp\sharew6_merged_a.dta, clear

*>> Fill up ch007_hh & ch007_km with infiormation from sn006_

    foreach n in 1 2 3 4 5 6 7 {
        replace ch007_hh = 1 if [ch007_hh == 0 | ch007_hh==.] &    ///
         [sn005_`n'>= 10 & sn005_`n' <= 13] & [sn006_`n' == 1 | sn006_`n' ==2]
        replace ch007_hh = 0 if ch007_hh==. &        ///
         [sn005_`n'>= 10 & sn005_`n' <= 13]  & sn006_`n' > 2 & sn006_`n' <.
    }

    foreach n in 1 2 3 4 5 6 7 {
        replace ch007_km = 1 if [ch007_km == 0 | ch007_km==.] &    ///
```

```
         [sn005_`n'>= 10 & sn005_`n' <= 13] & sn006_`n' == 3
        replace ch007_km = 0 if ch007_km==. &        ///
         [sn005_`n'>= 10 & sn005_`n' <= 13] & sn006_`n' > 3 & sn006_`n' <.
    }
        replace ch007_km = 1 if ch007_hh == 1


*>> Assign (grand)children information to partner of family respondent
    gsort   coupleid6 -fam_resp
    replace ch001_  =ch001_[_n-1]     ///
                              if coupleid6==coupleid6[_n-1] & coupleid6!=""
    replace ch007_hh=ch007_hh[_n-1]   ///
                              if coupleid6==coupleid6[_n-1] & coupleid6!=""
    replace ch007_km=ch007_km[_n-1]   ///
                              if coupleid6==coupleid6[_n-1] & coupleid6!=""
    replace ch021_mod=ch021_mod[_n-1] ///
                              if coupleid6==coupleid6[_n-1] & coupleid6!=""

*>> Assign information on received social support to partner
    foreach var in sp002_ sp003_1 sp003_2 sp003_3 {
        replace `var' = `var'[_n-1] if coupleid6==coupleid6[_n-1] ///
                                    & coupleid6!="" & fam_resp==0 ///
                                    & fam_resp[_n-1]==1 & `var'==.
        replace `var' = `var'[_n-1] if coupleid6==coupleid6[_n-1] ///
                                    & coupleid6!="" & fam_resp==0 ///
                                    & `var'==. & `var'[_n-1]!=.
    }


*>> Adapt the values in sn005_1 - sn005_7 to the codes in sp003* & sp009*
     foreach var in sn005_1 sn005_2 sn005_3 sn005_4 sn005_5 sn005_6 sn005_7 {
        replace `var' = 10 if `var' == 11
        replace `var' = 20 if `var' == 12
        replace `var' = 21 if `var' == 13
        replace `var' = 22 if `var' == 14
        replace `var' = 23 if `var' == 15
        replace `var' = 24 if `var' == 16
        replace `var' = 25 if `var' == 17
        replace `var' = 26 if `var' == 18
        replace `var' = 27 if `var' == 19
        replace `var' = 28 if `var' == 20
        replace `var' = 29 if `var' == 21
        replace `var' = 30 if `var' == 22
        replace `var' = 31 if `var' == 23
        replace `var' = 32 if `var' == 24
        replace `var' = 33 if `var' > 24 & `var' <=96
    }

*>> Include information from the list of sn members in sp003* and sp009*
     foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        replace `var' = sn005_1 if `var'== 101
        replace `var' = sn005_2 if `var'== 102
        replace `var' = sn005_3 if `var'== 103
        replace `var' = sn005_4 if `var'== 104
        replace `var' = sn005_5 if `var'== 105
        replace `var' = sn005_6 if `var'== 106
        replace `var' = sn005_7 if `var'== 107
        replace `var' = 10             if `var'==  19
        replace `var' = 33             if `var'>   33 & `var'< 101
    }
```

```
    foreach var in sp003_1 sp003_2 sp003_3 {
        replace `var' = -9 if sp002_== 5
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

        lab def relative 10 "child" 33 "other acquaintance" ///
        -9 "filtered: no help received/given", modify


    foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        rename `var' `var'_mod
    }

    foreach var in sp002_ {
        rename `var' `var'mod
    }

*>> Household able to make ends meet
    gen     co007a   = co007_
    replace co007a   = . if co007a < 0
    egen    co007_sd = sd(co007_), by(hhid6)
    replace co007a   = . if co007_sd > 0  &  co007_sd < .
    egen    co007b   = min(co007a),by(hhid6)
    replace co007_   = co007b if co007b != .
    drop    co007a co007b co007_sd

*>> Assign information on area to other household members
    gsort hhid6 -hou_resp
    gen iv009_mod     = iv009_
    replace iv009_mod = -9 if iv009_==. & [ho001 == 5 | mn024 ==2]
    replace iv009_mod = iv009_[_n-1] if iv009_mod==. & hhid6==hhid6[_n-1]
    lab var iv009_mod "Area of location"
    lab val iv009_mod arealocated
    lab def arealocated -9 "filtered: interview not at home", modify


save $easy\data\temp\sharew6_merged_b.dta, replace


*>> w7:

use $easy\data\temp\sharew7_merged_a.dta, clear

*>> Assign (grand)children information to partner of family respondent
    gsort   coupleid7 -fam_resp
    replace ch001_  =ch001_[_n-1]     ///
                                if coupleid7==coupleid7[_n-1] & coupleid7!=""
    replace ch007_hh=-9 if ch001==0
    replace ch007_hh=ch007_hh[_n-1]   ///
                                if coupleid7==coupleid7[_n-1] & coupleid7!=""
    replace ch007_km=ch007_km[_n-1]   ///
                                if coupleid7==coupleid7[_n-1] & coupleid7!=""
    replace ch021_mod=ch021_mod[_n-1] ///
                                if coupleid7==coupleid7[_n-1] & coupleid7!=""
```

```
*>> Assign variables on received social support to partner
    foreach var in sp002_ sp003_1 sp003_2 sp003_3 {
        replace `var' = `var'[_n-1] if coupleid7==coupleid7[_n-1] ///
                                    & coupleid7!="" & fam_resp==0 ///
                                    & fam_resp[_n-1]==1 & `var'==.
        replace `var' = `var'[_n-1] if coupleid7==coupleid7[_n-1] ///
                                    & coupleid7!="" & fam_resp==0  ///
                                    & `var'==. & `var'[_n-1]!=.
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

    foreach var in sp003_1 sp003_2 sp003_3 {
        replace `var' = -9 if sp002_== 5
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }

    foreach var in sp003_1 sp003_2 sp003_3 sp009_1 sp009_2 sp009_3 {
        rename `var' `var'_mod
    }

    foreach var in sp002_ {
        rename `var' `var'mod
    }

    foreach var in sp009_1 sp009_2 sp009_3 {
        replace `var' = -9 if sp008_ == 5
        replace `var' = 10 if `var' >= 10 & `var' < 20
    }
    lab def relative 10 "child" -9 "filtered: no help received/given", modify

*>> Household able to make ends meet
    gen     co007a   = co007_
    replace co007a   = . if co007a < 0
    egen    co007_sd = sd(co007_), by(hhid7)
    replace co007a   = . if co007_sd > 0  &  co007_sd < .
    egen    co007b   = min(co007a),by(hhid7)
    replace co007_   = co007b if co007b != .
    drop    co007a co007b co007_sd

*>> Assign information on area to other household members
    gsort hhid7 -hou_resp
    gen iv009_mod     = iv009_
    replace iv009_mod = -9 if iv009_==. & [ho001 == 5 | mn024 ==2]
    replace iv009_mod = iv009_[_n-1] if iv009_mod==. & hhid7==hhid7[_n-1]
    lab var iv009_mod "Area of location"
    lab val iv009_mod arealocated
    lab def arealocated -9 "filtered: interview not at home", modify


*>> Assign missing value for SHARELIFE interviews

    foreach var in ch001_ ch021_mod ch023_ ch524_ ch007_hh ch007_km br001_     ///
                   br002_ br015_ numeracy_1 numeracy_2 ep009_mod ep011_mod     ///
                   ep013_mod ep026_mod ep036_mod sp002_mod sp003_1_mod         ///
```

```
                    sp003_2_mod sp003_3_mod sp008_ sp009_1_mod sp009_2_mod      ///
                    sp009_3_mod euro1 euro2 euro3 euro4 euro5 euro6 euro7       ///
                    euro8 euro9 euro10 euro11 euro12 eurod eurodcat orienti     ///
                    income_pct_w7 {
        replace `var' = -10 if mn103_ == 1
    }

    foreach var in childhood_health vaccinated books_age10 maths_age10          ///
                    language_age10 {
        replace `var' = -11 if mn103_ == 0
      }


save $easy\data\temp\sharew7_merged_b.dta, replace



*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[31. Append waves to panel long format & integrate "long" variables]------

*>> Append single wave files to one long file:

    use           $easy\data\temp\sharew1_merged_b.dta, clear
    append using $easy\data\temp\sharew2_merged_b.dta
    append using $easy\data\temp\sharew3_merged_b.dta
    append using $easy\data\temp\sharew4_merged_b.dta
    append using $easy\data\temp\sharew5_merged_b.dta
    append using $easy\data\temp\sharew6_merged_b.dta
    append using $easy\data\temp\sharew7_merged_b.dta


*>> Integrate hhid? into a "long" format hidd variable
    gen      hhid = ""
    foreach w in $w {
        replace hhid = hhid`w' if wave==`w'
    }
    drop     hhid?
    lab var  hhid "Household identifier in respective wave - see var. wave"


*>> Integrate coupleID? into a "long" format coupleid variable
    gen      coupleid = ""
    foreach w in $w {
     replace coupleid = coupleid`w' if wave==`w'
    }
    drop     coupleid?
    lab var  coupleid "Couple identifier in respective wave - see var. wave"


*>> Generate wave participation overview variable
    foreach w in $w {
        gen temp1_`w' = `w' if wave==`w'
       egen temp2_`w' = max(temp1_`w'), by(mergeid)
    }
    gen      wavepart = ""
    foreach w in $w {
     replace wavepart = wavepart + string(temp2_`w')  if string(temp2_`w')!="."
```

```
    }
    destring wavepart, replace
    lab var  wavepart "Wave participation pattern"

    drop temp1_* temp2_*


*>> Generate interview version variable (baseline or panel interview)
    clonevar int_version = mn101_
    replace int_version  = 0 if mn101_==. & wave  ==1
    replace int_version  = 0 if mn101_==. & dn044_==. // dn044 only asked in panel
    replace int_version  = 1 if mn101_==. & dn044 !=.
    lab var int_version "Interview version (baseline or longitudional)"

*>> Generate SHARELIFE interview
    clonevar SHARELIFE_int = mn103_
    replace  SHARELIFE_int = 1 if wave == 3
    lab var  SHARELIFE_int "SHARELIFE Interview"

*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[32. Fix date intv., year/month birth, gender, & partnervars & gen age]---

*   Using information of other waves and simple mode imputations for date of
*   interview and month of birth (if year is missing in all waves available,
*   we do not impute)

*>> Impute date of interview if missing (SHARELIFE known issue)
*   by (minimum) mode values of wave/country:
    gen     int_date = ym(int_year, int_month) // %tm format
    replace int_date =dofm(int_date)           // changes to date format

    egen   int_date_mode = mode(int_date),by(wave country_mod) minmode
    format int_date_mode int_date %d

    replace int_month = month(int_date_mode)  if int_month==.
    replace int_year  =  year(int_date_mode)  if int_year ==.

    drop int_date int_date_mode


*>> Check for deviations within gender (known issue in IL wave 1 vs 2):
    // if gender deviates between waves, one information must be wrong
    // as there is no way to know which is the wrong information, both
    // are set to -3 "implausible value/suspected wrong"
    egen    female_sd = sd(female), by(mergeid)
    replace female = -3 if female_sd > 0 & female_sd < .
    drop                   female_sd

*>> Check for deviations within year of birth (known issue in IL wave 1 vs 2):
    // same as with gender:
    gen     dn003_mod  = dn003_ if dn003_ > 0

    egen    dn003_mod_sd = sd(dn003_mod), by(mergeid)
    replace dn003_mod    = -3 if dn003_mod_sd > 0 & dn003_mod_sd < .
    replace dn003_mod    = -3 if dn003_mod == 2011
    drop                   dn003_mod_sd
```

```
    lab var dn003_mod "Year of birth"
    lab def dn003_mod -15 "no information"                    ///
                      -3 "implausible value/suspected wrong"
    lab val dn003_mod dn003_mod

*>> Generate new month of birth variable, taking the minimum mode of all info:
    // here we are less strict as with year of birth and gender
    // i.e. we do not set the self-report to missing if it deviates between
    // waves, instead we take the minimum modus answer of the self-reported
    // dn002_ variable per person:

    gen     dn002_mod = dn002_ if dn002_ > 0
    egen    dn002_mod_modus = mode(dn002_mod), by(mergeid) minmode
    replace dn002_mod = dn002_mod_modus if dn002_mod != dn002_mod_modus
    drop                dn002_mod_modus

    lab val dn002_mod month
    lab var dn002_mod "Month of birth"

*>> Now we can replace missing information from another wave if available:
    foreach var in female dn003_mod dn002_mod dn004_ dn006_ {
      di "`var'"
        // Here we start to copy the information to all lines of
        // the respondent; i.e. isced1997_r from wave 1 is written into
        // all lines of the same respondent.
        sort mergeid
        foreach i in $w {
           di "`var'"
            replace `var' = `var'[_n+`i'] if mergeid==mergeid[_n+`i'] & ///
                                             `var'== . & `var'[_n+`i'] !=.
            replace `var' = `var'[_n-`i'] if mergeid==mergeid[_n-`i'] & ///
                                             `var'== . & `var'[_n-`i'] !=.
        }
    }

       rename dn004_ dn004_mod // because we transfer information to wave 3


*>> Finally we can generate age of respondent:
    * for the few respondents with available year of birth (dn003) but missing
    * month of birth (dn002), we assume they are born in June:

    gen     age = ( (int_year * 12  + int_month) -                    ///
                    (dn003_mod * 12 + dn002_mod) ) / 12 if dn003_mod != -3

    replace age = ( (int_year * 12  + int_month) -                    ///
                    (dn003_mod * 12 + 6        ) ) / 12 if dn003_mod != -3 ///
                                                    & dn002_mod ==  .

    lab var  age "Age at interview (in years)"


*>> Generate Age of partner
    gen age_partner = .
    // if we have a self report of the repective wave specific partner
    // we take it:
    sort wave coupleid
    replace age_partner = age[_n-1] if coupleid==coupleid[_n-1] & coupleid!=""
    replace age_partner = age[_n+1] if coupleid==coupleid[_n+1] & coupleid!=""
```

```
    // otherwise we generate age based on the cv_r information:
    replace age_partner = ( (int_year * 12        + int_month      ) -       ///
                            (yrbirthp * 12 + mobirthp) ) / 12  ///
                           if age_partner == .        /// no self-report
                            & coupleid != ""          /// has partner
                            & yrbirthp > 0 & yrbirthp < .      ///
                            & mobirthp > 0 & mobirthp < .

    // finally, as with age we assume the partner is born in June if only
    // the month of birth of partner is missing, but the year is available:
    replace age_partner = ( (int_year * 12        + int_month      ) -       ///
                            (yrbirthp * 12 + 6              ) ) / 12  ///
                           if age_partner == .        /// no self-report
                            & coupleid != ""          /// has partner
                            & yrbirthp > 0 & yrbirthp < .

    drop yrbirthp mobirthp

    assert age_partner == . if coupleid=="" // just to make sure
    assert age_partner >= 0

    replace age_partner =  -9 if    coupleid==""
    replace age_partner = -15 if age_partner== .

    lab var age_partner "Age at interview of partner"

    lab def age_partner -15 "no information"                          ///
                         -9 "filtered: single or no partner in hh"    ///
                         -3 "implausible value/suspected wrong"
    lab val age_partner age_partner

*>> Take out very weird values in age & age_partner
    replace age         = -3 if age         < 14
    replace age_partner = -3 if age_partner < 14 & age_partner >=0

*>> After having transferred age to age_partner in couples set no info codes
*   in dn003_mod dn002_mod and age:

    replace dn003_mod = -15 if dn003_mod == .
    replace dn002_mod = -15 if dn002_mod == .

    replace age =   -3 if dn003_mod == -3
    replace age = - 15 if       age == .

    lab def age -15 "no information"                          ///
                -3 "implausible value/suspected wrong"
    lab val age age


*>> Change storage format of age and age_partner to only one decimal
    // it is only month exact

    replace age         = round(age        ,0.1)
    replace age_partner = round(age_partner,0.1)
    format  age         %9,1f
    format  age_partner %9,1f


*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
```

```
*-------------------------------------------------------------------------------

*----[33. Transfer information collected once (in baseline interviews)]--------
*         to the next waves of the respondent
*         (done for isced, eduyears, country of birth, citizenship, height)

    // replace birth_country with counry of interview if dn004_mod = 1(yes)
    replace birth_country = country_mod if dn004_mod ==1 & birth_country ==.

foreach var in eduyears birth_country citizenship {
    // isced1997_r is already transfered in gv_isced of rel 6.0.0
  di "`var'"
    // First we check for deviations within person and set the
    // variable to missing if we have contradictory information.
    // This occurs rarely and there is no way to know which
    // information is correct.
    gen     `var'c   = `var'
    replace `var'c   = .  if `var'c < 0
    egen    `var'c_sd =   sd(`var'c), by(mergeid)
    replace `var'c    = . if `var'c_sd > 0  &  `var'c_sd < .

    // Here we start to copy the information to all lines of
    // the respondent; i.e. isced1997_r from wave 1 is written into
    // all lines of the same respondent.
    sort mergeid wave
    foreach i in $w {
       di "`var'"
        replace `var'c = `var'c[_n+`i'] if mergeid==mergeid[_n+`i'] & ///
                                           `var'c== . & `var'c[_n+`i'] !=.
        replace `var'c = `var'c[_n-`i'] if mergeid==mergeid[_n-`i'] & ///
                                           `var'c== . & `var'c[_n-`i'] !=.
    }
    // Here we copy back the missing codes, only into the line of the wave
    // the missing code occurred, if we did not find a valid answer in any
    // other wave.
    replace `var'c = `var' if `var'c == . & `var' < 0
    rename         `var' `var'_orig // to preserve the label stored in `var'
    rename  `var'c `var'
}

lab val isced1997_r isced
lab var isced1997_r "Education of respondent in ISCED-97 Coding"

lab val eduyears eduyears_mod
lab var eduyears "Years of education"
rename  eduyears eduyears_mod // We ignored the coded eduyears from wave 1
                              // in this variable, this is why we add the
                              // "modified" indicator here.

drop eduyears_orig eduyearsc_sd

foreach var in isced1997_r eduyears_mod {
    lab def `var'  95 "still in school", add
    lab def `var'  97 "other", add
}

replace eduyears_mod = -3 if eduyears_mod>=100 & eduyears_mod!=.

lab val citizenship     citizenship     // labelname of dn008c
lab var citizenship     "Citizenship of respondent (ISO coded)"
```

```
lab val birth_country   countryofbirth  // labelname of dn005c
lab var birth_country   "Country of birth (ISO coded)"

drop dn008c dn005c // Now we don't need these two variables anymore.
                   // They were only kept to have their label info available.



*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[34. Pass on information to next wave that may have changed/not changed]--


*>> Assign w3 information on childhood conditions to w5
    replace wave=44 if wave==4 // Change wave variable to 44 for w4
                               // This allows sorting w3 and w5 in order
                               // and to easier omitt wave 4.
    sort mergeid wave

    foreach var in books_age10 maths_age10 language_age10 {
        replace `var' = `var'[_n-1] if `var'==. & `var'[_n-1] !=. ///
        & wave == 5 & wave[_n-1] == 3 & mergeid == mergeid[_n-1]
    }

    replace wave=4 if wave==44


*>> Assign marital status if longitudinal
    sort mergeid wave

    replace dn014_ = 6 if mergeid==mergeid[_n-1] & dn014_==. & ///
             (pdeath_last >= int_year[_n-1]) & (pdeath_last <= int_year) ///
            & dn014_[_n-1]!=4 & dn014_[_n-1]!=5
            // include widowhood from w3

    replace dn014_ = 5 if mergeid==mergeid[_n-1] & dn014_==. & ///
             (pdivorce_last >= int_year[_n-1]) & (pdivorce_last <= int_year) ///
            & dn014_[_n-1]!=4 & dn014_[_n-1]!=5 & dn014_[_n-1]!=6
            // include divorces from w3

    // transfer onwards if marital status did not change
    replace dn014_ = dn014_[_n-1] if mergeid==mergeid[_n-1] & dn014_>=. & ///
             (dn044_==5 | dn044_ >=.)
    replace dn014_ = dn014_[_n-2] if mergeid==mergeid[_n-2] & dn014_>=. & ///
             (dn044_==5 | dn044_ >=.)
    replace dn014_ = dn014_[_n-3] if mergeid==mergeid[_n-3] & dn014_>=. & ///
             (dn044_==5 | dn044_ >=.)
    replace dn014_ = dn014_[_n-4] if mergeid==mergeid[_n-4] & dn014_>=. & ///
             (dn044_==5 | dn044_ >=.)
    replace dn014_ = dn014_[_n-5] if mergeid==mergeid[_n-5] & dn014_>=. & ///
             (dn044_==5 | dn044_ >=.)

    // transfer backwards if baseline conducted in later wave + no previous info
    replace dn014_ = dn014_[_n+1] if mergeid==mergeid[_n+1] & dn044_ ==  5 ///
            & dn014_>=. & int_version[_n+1]==0
    replace dn014_ = dn014_[_n+2] if mergeid==mergeid[_n+2] & dn044_ ==  5 ///
            & dn014_>=. & int_version[_n+2]==0
```

```
    replace dn014_ = dn014_[_n+3] if mergeid==mergeid[_n+3] & dn044_ ==  5 ///
            & dn014_>=. & int_version[_n+3]==0
    replace dn014_ = dn014_[_n+4] if mergeid==mergeid[_n+4] & dn044_ ==  5 ///
            & dn014_>=. & int_version[_n+4]==0
    replace dn014_ = dn014_[_n+5] if mergeid==mergeid[_n+5] & dn044_ ==  5 ///
            & dn014_>=. & int_version[_n+5]==0

    rename  dn014_ mar_stat


replace wave=33 if wave==3 // Change wave variable to 33 for SHARELIFE
                           // This allows sorting w1, w2 and w4 in order
                           // and to easier omitt wave 3.

*>> Assign siblings and parents alive:
    sort mergeid wave
    replace siblings_alive = -9                     ///
                    if siblings_alive == .          ///
                     & mergeid == mergeid[_n-1]     ///
                     & (siblings_alive[_n-1] == 0   ///
                     | siblings_alive[_n-1] == -9)  ///
                     & wave != 33
    replace siblings_alive = - 3 if siblings_alive==99

    replace dn026_1=5 if mergeid==mergeid[_n-1]                         ///
                       & (dn026_1==. | dn026_1<0) & dn026_1[_n-1]==5    ///
                       & wave!=33
    replace dn026_2=5 if mergeid==mergeid[_n-1]                         ///
                       & (dn026_2==. | dn026_1<0) & dn026_2[_n-1]==5    ///
                       & wave!=33
    replace dn026_1=1 if mergeid == mergeid[_n+1]                       ///
                       & dn026_1==. & dn026_1[_n+1] ==1                 ///
                       & wave!=33
    replace dn026_2=1 if mergeid == mergeid[_n+1]                       ///
                       & dn026_2==. & dn026_2[_n+1] ==1                 ///
                       & wave!=33

    rename  dn026_1 mother_alive
    rename  dn026_2 father_alive


*>> Assign ever smoked and currently smoking
    sort mergeid wave
    replace br002_=5 if br001_==5 & br002_==.
    replace br001_=5 if mergeid==mergeid[_n-1] & br001_[_n-1]==5 ///
                      & br001_==. & wave!=33
    replace br001_=1 if mergeid==mergeid[_n-1] & br001_[_n-1]==1 ///
                      & br001_==. & wave!=33

    rename  br001_ ever_smoked
    rename  br002_ smoking
    label   def br002 5 "no", modify


*>> Assign children living in same hh for w4 and w5
    sort mergeid wave
    replace ch007_hh=ch007_hh[_n-1] if mergeid==mergeid[_n-1] & ch524==5 ///
                          & ch007_hh[_n-1] !=. & ch007_hh ==. & wave !=33
    replace ch007_hh=ch007_hh[_n-2] if mergeid==mergeid[_n-2] & ch524==5 ///
                          & ch007_hh[_n-1] ==. & ch007_hh[_n-2] !=.      ///
```

```
                        & ch007_hh ==. & wave !=33
    gsort coupleid -fam_resp
    replace ch007_hh=ch007_hh[_n-1] if fam_resp == 0                  ///
                         & coupleid==coupleid[_n-1] & coupleid!="" ///
                         & [wave ==4 | wave ==5]
    replace ch007_hh =. if ch001>0 & ch001!=. & ch007_hh ==-9

*>> Assign ch007_km for w4 and w5
    sort mergeid wave
    replace ch007_km = 1 if ch007_hh == 1
    replace ch007_km=ch007_km[_n-1] if mergeid==mergeid[_n-1] & ch524==5 ///
                        & ch007_km[_n-1] !=. & ch007_km ==. & wave !=33
    replace ch007_km=ch007_km[_n-2] if mergeid==mergeid[_n-2] & ch524==5 ///
                        & ch007_km[_n-1] ==. & ch007_km[_n-2] !=.      ///
                        & ch007_km ==. & wave !=33
    gsort coupleid -fam_resp
    replace ch007_km=ch007_km[_n-1] if fam_resp == 0                  ///
                         & coupleid==coupleid[_n-1] & coupleid!="" ///
                         & [wave ==4 | wave ==5]
    replace ch007_km =. if ch001>0 & ch001!=. & ch007_km ==-9

replace wave=3 if wave==33 // Change wave variable back to 3 for SHARELIFE


*------------------------------------------------------------------------------
*------------------------------------------------------------------------------
*------------------------------------------------------------------------------

*----[35. Fix & re-generate variables, labels, etc.]---------------------------


*>> Use information from social networks module to reduce missing values
*   in mother-/father_alive in waves 4 and 6
    foreach var in sn005_1 sn005_2 sn005_3 sn005_4 sn005_5 sn005_6 sn005_7 {
        replace mother_alive = 1 if `var' ==2
    }

    foreach var in sn005_1 sn005_2 sn005_3 sn005_4 sn005_5 sn005_6 sn005_7 {
        replace father_alive = 1 if `var' ==3
    }

*>> Re-generate yes/no values (1=yes, 5=no) for consistency reasons
    replace ch007_hh = 5 if ch007_hh ==0
    lab def lblch007_hh 5 "no", modify

    replace ch007_km = 5 if ch007_km ==0
    lab def lblch007_km 5 "no", modify

*>> Add labels for additional wave 7 answer options for sp003# and sp009#
    lab def relative 35 "Minister, priest, or other clergy"      ///
                     36 "Therapist or other professional helper" ///
                     37 "Housekeeper/Home health care provider"  ///
                     96 "None of these", add

*>> Add, re-assign or complete missing labels
    lab var casp "CASP: quality of life and well-being index"
    lab val casp casp

    lab val country_mod country_mod7
```

```
    lab val partnerinhh partnerinhh
    lab def partnerinhh 1 "living with a spouse/partner in household"       ///
                        3 "living without a spouse/partner in household"    ///
                       97 "other"

    lab var partnerinhh "Living with spouse/partner"
    lab var thinc_m "Household net income, imputed"
    lab var bmi     "Body mass index"
    lab var bmi2    "Body mass index categories"


    lab val hhsize         hhsize
    lab val recall_1       recall_1
    lab val recall_2       recall_2
    lab val chronic_mod    chronic_mod
    lab val mobilityind    mobilityind
    lab val lgmuscle              lgmuscle
    lab val adlwa                 adlwa
    lab val adla                  adla
    lab val grossmotor            grossmotor
    lab val finemotor             finemotor
    lab val iadla          iadla
    lab val iadlza         iadlza
    lab val maxgrip        maxgrip

    lab def hhsize         -15 "no information"
    lab def recall_1       -15 "no information"
    lab def recall_2       -15 "no information"
    lab def chronic_mod    -15 "no information"
    lab def mobilityind    -15 "no information"
    lab def lgmuscle       -15 "no information"
    lab def adlwa          -15 "no information"
    lab def adla           -15 "no information"
    lab def grossmotor     -15 "no information"
    lab def finemotor      -15 "no information"
    lab def iadla          -15 "no information"
    lab def iadlza         -15 "no information"
    lab def income_pct_w1   -13 "not asked in this wave", modify

*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[36. Implement/complete wave/country skip patterns]-----------------------

    foreach var in mar_stat mother_alive father_alive siblings_alive ///
                   dn004_mod thinc_m int_version isced1997_r {
        replace `var' = -13 if wave==3
    }

    foreach var in co007_ ever_smoked smoking br015_ ///
                   recall_1 recall_2 iv009_mod {
        replace `var' = -13 if wave==3
    }

    foreach var in ch001_ ch021_mod bmi bmi2 chronic_mod {
        replace `var' = -13 if wave==3
    }

    foreach var in ep005_ ep009_ ep011_mod ep013_mod ep026_mod ep036_mod {
```

```
        replace `var' = -13 if wave==3
    }

    foreach var in hc002_mod hc012_ hc029_ {
        replace `var' = -13 if wave==3
    }

    foreach var in sp002_  sp003_1 sp003_2 sp003_3 sp009_1 sp009_2      ///
                   sp009_3 sp008_ {
        replace `var' = -13 if wave==3
    }

    foreach var in casp euro1 euro2 euro3 euro4 euro5 euro6 euro7 euro8 ///
                   euro9 euro10 euro11 euro12 eurod {
        replace `var' = -13 if wave==3
    }

    foreach var in mobilityind lgmuscle adlwa adla grossmotor finemotor ///
                   iadla iadlza orienti numeracy_1 numeracy_2 {
        replace `var' = -13 if wave==3
    }

    foreach var in br010_mod  {
        replace `var' = -13 if wave==3 | wave==6 | wave==7
    }

    foreach var in ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6      ///
                   ac002d7 ac002dno {
        replace `var' = -13 if wave==3 | wave==4 | wave==5 | wave==6 | wave==7
    }

    foreach var in ch007_hh ch007_km {
        replace `var' = -13 if wave==3
    }

    foreach var in q34_re {
        replace `var' = -13 if wave==2 | wave==3 | wave==4 | wave==5 | wave==6 |
wave==7
    }

    replace numeracy_2 = -13 if wave == 1 | wave == 2

    replace income_pct_w1 = -13 if wave !=1
    replace income_pct_w2 = -7  if country ==30
    replace income_pct_w2 = -13 if wave !=2
    replace income_pct_w4 = -13 if wave !=4
    replace income_pct_w5 = -13 if wave !=5
    replace income_pct_w6 = -13 if wave !=6
    replace income_pct_w7 = -13 if wave !=7

*>> Variables only included in SHARELIFE & wave5
    foreach var in books_age10 maths_age10 language_age10 childhood_health
vaccinated {
        replace `var' = -13 if wave==1 | wave==2 | wave==4 | wave==6
    }

*>> q34_re : religion not asked in France drop-off wave 1:
    replace q34_re =-14 if substr(mergeid,1,2)=="FR" & wave==1
```

```
*>> Set additional missing code if information is not available
*   because wave 1 dropoff is not available / could not be linked

    foreach var in casp q34_re {
        replace `var' = -16 if merge_w1_dropoff == 1 & wave==1
    }

    foreach label in casp q34_re {
      lab def `label' -16                                        ///
        "no drop-off (information in drop-off in this wave)"     ///
      , add modify
    }

    drop merge_w1_dropoff

*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[37. Integrate DK/RF and implement no information missing code]-----------


*>> Integrate "don't know" and "refusal" for remaining variables
    foreach var in                                                          ///
        female age partnerinhh dn002_ dn003_ dn004_mod dn006_ gender_partner ///
        age_partner hhsize country country_mod birth_country citizenship    ///
        iv009_mod sphus books_age10 maths_age10 language_age10              ///
        vaccinated childhood_health q34_re                                  ///
        mar_stat mother_alive father_alive siblings_alive                   ///
        income_pct_w1 income_pct_w2 income_pct_w4 income_pct_w5             ///
        mobilityind lgmuscle adlwa adla grossmotor finemotor                ///
        iadla iadlza orienti numeracy_1 numeracy_2 maxgrip eurod            ///
        isced1997_r eduyears_mod                                            ///
        co007_ ever_smoked smoking br010_mod br015_ recall_1 recall_2       ///
        ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6 ac002d7             ///
        ac002dno                                                            ///
        sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3       ///
        ch001_ ch021_mod ch007_hh ch007_km bmi bmi2 chronic_mod             ///
        ch007_hh ch007_km                                                   ///
        ep005_ ep009_ ep011_mod ep013_mod ep026_mod ep036_mod               ///
        hc002_mod hc012_ hc029_                                             ///
        casp euro1 euro2 euro3 euro4 euro5 euro6 euro7 euro8                ///
        euro9 euro10 euro11 euro12 eurod                                 {

            replace `var' = -12 if `var'==-1 | `var'==-2
    }

*>> Set "no information" missing code, when no other reason is left:
    foreach var in                                                          ///
            female age dn002_ dn003_ dn004_mod dn006_ partnerinhh           ///
            gender_partner age_partner int_partner                          ///
            hhsize country country_mod birth_country citizenship iv009_mod  ///
            vaccinated childhood_health  sphus q34_re                       ///
            books_age10 maths_age10 language_age10                          ///
            mar_stat mother_alive father_alive siblings                     ///
            income_pct_w1 income_pct_w2 income_pct_w4 income_pct_w5          ///
            mobilityind lgmuscle adlwa adla grossmotor finemotor            ///
            iadla iadlza orienti numeracy_1 numeracy_2 maxgrip eurod         ///
            isced1997_r eduyears_mod                                        ///
            co007_ ever_smoked smoking br010_mod br015_ recall_1 recall_2   ///
```

```
            ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6 ac002d7          ///
            ac002dno                                                         ///
            sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3    ///
            ch001_ ch021_mod bmi bmi2 chronic_mod                            ///
            ch007_hh ch007_km                                                ///
            ep005_ ep009_ ep011_mod ep013_mod ep026_mod ep036_mod            ///
            hc002_mod hc012_ hc029_                                          ///
            casp euro1 euro2 euro3 euro4 euro5 euro6 euro7 euro8             ///
            euro9 euro10 euro11 euro12 eurod thinc_m                        {

                replace `var' = -15 if `var'==.
    }


*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------
*-------------------------------------------------------------------------------

*----[38. Keep, add easy missing codes & labels, order, data labels & save]----

keep  mergeid hhid coupleid wave wavepart int_version                        ///
      int_year int_month country country_mod language                        ///
      female dn002_mod dn003_mod dn004_mod age birth_country citizenship     ///
      iv009_mod q34_re                                                       ///
      isced1997_r eduyears_mod mar_stat hhsize partnerinhh int_partner       ///
      age_partner gender_partner  mother_alive father_alive siblings_alive   ///
      ch001_ ch021_mod ch007_hh ch007_km                                     ///
      ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6 ac002d7 ac002dno       ///
      sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3          ///
      books_age10 maths_age10 language_age10 vaccinated childhood_health     ///
      sphus chronic_mod casp euro1 euro2 euro3 euro4 euro5 euro6 euro7       ///
      euro8 euro9 euro10 euro11 euro12 eurod                                 ///
      hc002_mod hc012_ hc029_                                                ///
      maxgrip adlwa adla iadla iadlza mobilityind lgmuscle                   ///
      grossmotor finemotor recall* orienti numeracy_1 numeracy_2             ///
      bmi bmi2 smoking ever_smoked br010_mod br015_                          ///
      ep005_ ep009_ ep011_mod ep013_mod ep026_mod ep036_mod co007_ thinc_m   ///
      income_pct_w1 income_pct_w2 income_pct_w4 income_pct_w5 income_pct_w6  ///
      income_pct_w7


*>> Add additional easy labels to all remaining labelsets:
    label dir
    global labellist "`r(names)'"
    foreach label in $labellist {
        lab def `label'  -3 "implausible value/suspected wrong"        , add modify
        lab def `label'  -7 "not yet coded"                             , add modify
        lab def `label' -10 "SHARELIFE interview"                       , add modify
        lab def `label' -11 "regular interview"                         , add modify
        lab def `label' -12 "don't know / refusal"                      , add modify
        lab def `label' -13 "not asked in this wave"                    , add modify
        lab def `label' -14 "not asked in this country"                 , add modify
        lab def `label' -15 "no information"                            , add modify
    }

    replace maxgrip = -15 if maxgrip < 0
    lab def maxgrip -15 "no/no valid measure"
    lab val maxgrip maxgrip
```

```
*>> Order
order mergeid hhid coupleid wave wavepart int_version                          ///
      int_year int_month country country_mod language                          ///
      female dn002_mod dn003_mod dn004_mod age birth_country citizenship       ///
      iv009_mod q34_re                                                         ///
      isced1997_r eduyears_mod mar_stat hhsize partnerinhh int_partner         ///
      age_partner gender_partner  mother_alive father_alive siblings_alive     ///
      ch001_ ch021_mod ch007_hh ch007_km                                       ///
      ac002d1 ac002d2 ac002d3 ac002d4 ac002d5 ac002d6 ac002d7 ac002dno         ///
      sp002_ sp003_1 sp003_2 sp003_3 sp008_ sp009_1 sp009_2 sp009_3            ///
      books_age10 maths_age10 language_age10 vaccinated childhood_health       ///
      sphus chronic_mod casp euro1 euro2 euro3 euro4 euro5 euro6 euro7         ///
      euro8 euro9 euro10 euro11 euro12 eurod                                   ///
      hc002_mod hc012_ hc029_                                                  ///
      maxgrip adlwa adla iadla iadlza mobilityind lgmuscle                     ///
      grossmotor finemotor recall* orienti numeracy_1 numeracy_2               ///
      bmi bmi2 smoking ever_smoked br010_mod br015_                            ///
      ep005_ ep009_ ep011_mod ep013_mod ep026_mod ep036_mod co007_ thinc_m     ///
      income_pct_w1 income_pct_w2 income_pct_w4 income_pct_w5 income_pct_w6    ///
      income_pct_w7

    sort mergeid wave

*>> Label & Notes for dataset
    label data ""
    notes drop _dta
    note: {it: easy}SHARE release {cmd: 6.1.0} waves {cmd: $w} ///
          {it:doi } {cmd: 10.6103/SHARE.easy.610}
    label data "easySHARE release 7.0.0  waves $w  doi 10.6113/SHARE.easy.700"

compress

saveold $easy\data\easySHARE_rel7-0-0.dta, version(12) replace
    // saveold is used to have the dataset readable in Stata 12
```