THE LWS USER GUIDE
2024 Template

Generic information

Structure of the LWS datasets

Variable standardisation

Generic missing values policy

Weights

Useful information on *LWS* household balance sheet

Aggregation of wealth variables

Pension assets availability and the concept of net worth

Breakdown of liabilities variables

The *LWS Database* focuses on the wealth and debt of households. The goal of *LWS* is to enhance studies on the understanding of households' financial stability through both the analyses of wealth distribution and other related dimensions of economic well-being. It draws upon the *LWS* project that started in 2007; that pilot version of the wealth database facilitated the exchange of expertise on wealth among scholars in that field, and as a consequence, the new framework of *LWS* has been introduced.

This document provides generic guidelines for using *LWS* data and its overall structure. Since the socio-demographic, labour market, income, and consumption variables are the same in *LIS* and *LWS* datasets, the users are encouraged to consult the *LIS User Guide* for more details.

More detailed information about the contents and coding of the *LWS* variables is available in the *METadata Information System (METIS)* under the *Variables definitions*. In addition, *METIS* provides dataset-specific information included as *Contents and Notes*; this field includes both warnings about specific situations as well as the contents of wealth, income, and consumption variables in each *LWS* dataset. Information about the original data source, such as the data provider, data collection information, and additional technical information, is also available in *METIS* in the *Dataset information* section. In addition, *Compare.It* provides a country-specific summary of data comparability-related issues.

Generic information

Naming of the LWS datasets - The *LWS Database* consists *ex post* harmonised datasets from different countries and years. Each dataset refers to one specific country at one point in time, and is named according to the 2-character country abbreviation coded according to the International Standard for country codes (*ISO-3166*) and the reference year. The reference year is the calendar year, which includes the data point to which wealth data refers. Please note that the reference year may differ from the year used by the data provider in the official documents. For the exact reference period of the values of wealth and income variables, please consult *Dataset Information* available for each specific dataset in *METIS* (accessible through the LIS website).

Structure of the LWS datasets - Each *LWS* dataset contains at least two files, namely the household-level file and the individual-level file; additionally for some datasets, there might also be a replicate weights file.

- *Household level file (LWS H-file)* – This file includes all household level variables (see technical, household characteristics, household balance sheet, and other wealth, behavioural, and flow variables reported under the heading H-file of the *LWS Variables List*). The unit of this file corresponds to the survey unit, which, for wealth surveys, typically is the household or the primary economic unit but may differ in some cases (e.g., tax unit, family unit, etc. – *Dataset Information* available in *METIS* for more information); irrespective of the exact definition of the survey unit, the unit of the file

is always referred to as "household." The *LWS* H-file contains at least one observation for each household of the sample. If the data provider conducted a multiple imputation procedure to impute missing values, all imputation replicates (implicates) are included in this file. Since the imputations are usually stored as five successive implicates of each record, the number of observations in the file is five times the actual number of households. This file is uniquely identified by the household identifier as well as the implicate number (variables *hid* and *inum*). If there was no imputation or a single imputation procedure applied by the data provider instead of multiple imputations, then the variance of variable *inum* is equal to 0.

- *Individual level file (LWS P-file)* – This file includes all individual level variables (see technical, socio-demographic, labour market, pension asset, behavioural, and flow variables reported under the heading P-file of the *LWS Variables List*). The unit of the file is the individual, and the P-file includes at least one record for each individual belonging to the household (even if the data were originally not reported for all household members). If the data provider conducted a multiple imputation procedure to impute missing values, all implicates are included in this file. This file is uniquely identified by the household identifier, the person identifier, and the implicate number (variables *hid*, *pid*, and *inum*). If no imputation or a single imputation procedure was applied by the data provider instead of multiple imputation, then the variance of variable *inum* is equal to 0.
- *Replicate weights file (LWS R-file)* – This file contains as many replicate weights as the data producer has created (up to 1,000), and it includes precisely one observation for each household of the sample. By using the replicate weights, users obtain more precise confidence intervals and significance tests; in other words, the replicate weights generate more informed empirically derived standard error estimates by simulating multiple samples from a single sample. The file is uniquely identified by the household identifier (*hid*). To perform analyses using household or person-level variables, users need to merge the R-file with the H-file or P-file using the *hid* variable as a key. For detailed guidance on utilizing replicate weights, refer to the online tutorial.

Note that all *LWS* H-files include the same variables in all datasets (i.e. all variables are actually present even if they contain only the missing or zero values). The same holds for all *LWS* P-files.

Multiple imputations in LWS datasets – Some LWS datasets are multiply imputed by the data producer, which requires special attention during statistical analysis, particularly when using Stata. Users performing analyses in Stata are encouraged to refer to the online tutorial or Appendix 1 at the end of this document for detailed guidance. The R users are encouraged to refer to Appendix 2. Users of SPSS and SAS are advised to consult the respective software documentation for handling multiple imputations.

Variable standardisation - The *LWS* variables are standardised in terms of conceptual content (the variables are as comparable as possible across datasets in terms of concepts/definitions, see *LWS Variables Definitions* in METIS) as well as in terms of coding structure:

- All continuous variables are standardised and report information expressed in the same unit across different datasets. *LWS* wealth and flow variables are reported in annual amounts and in units of the national currency in force at the time of the data collection (see variable *currency*).
- Most categorical variables are standardised and report information expressed with the same value codes and labels.

There are also some categorical variables that are not standardised (variables denoted by a "*_c*" suffix). While the variable name is the same across all datasets, the variable label may differ to indicate the actual (dataset-specific) contents for the dataset in question. Both the exact contents and the coding structure will differ across datasets (see the *LWS Codebooks* in METIS).

The standardisation of some categorical variables follows a two- or multi-digit coding structure, where codes starting with the same digit belong to the same overall category. For example, in the variable BASB (saving behaviour), codes in the 10s indicate that a person does not save. Within this category, code 11 means "does not save: expenses exceed income," and code 12 means "does not save: expenses about the same as income." It is important to note that observations can be coded either at the overall category level or at more detailed subcategory levels. For instance, some individuals may be coded as 11 or 12, while others are coded directly as 10 ("does not save") if the specific reason for not saving could not be determined. As a result, selecting only the subcategories (codes 11 and 12) does not necessarily capture all individuals in the overall category. To ensure all "does not save" cases are included, the higher-level code 10 must also be selected.

Generic missing values policy - In the *LWS* database, the system missing value ('dot') represents all cases of observations for which the information is not available. This includes both, the cases where the information is not applicable (e.g., the person does not work and hence cannot have an industry), and the cases where the information is applicable, but not available. The former applies only to the categorical variables, but not to the continuous variables such as wealth, income, and consumption (see also below discussion on "0" values). The latter case includes the following situations:

- the information is applicable, but has not been collected by the data provider for a given subset of the sample or at all (e.g., only heads of households have been asked questions about financial literacy);
- the information is applicable and has been collected by the data provider, but the respondent did not answer (don't know/refusal); in most of the *LWS* datasets, these kinds of missing values have been imputed by the data providers.

The number of "applicable missings" (as described in the two situations above) can vary substantially across datasets. This is due to the different *ex ante* reasons, such as collection practices and household's willingness to respond as well as *ex post* imputation practices adopted by the data producers. The *LWS* data include imputed values only if they were available in the original data provided by the data producer.

A special attention needs to be given to "0" values in the *LWS* datasets. Besides the cases where zero represents a value (e.g., a household has no consumer debts), the continuous variables from the sections of wealth, income and consumption have zeros in the following scenarios:

- the information has not been collected at all (e.g., expenditures on health have all "0" values if the question concerning this expenditure has not been asked in the survey);
- the information has been collected but is not available at the level of detail necessary for the *LWS* variable in question (e.g., the vehicle loans - variable *hlncv* - has all "0" values if all consumer goods loans are collected together).
- the collection of a specific item is not applicable for a subgroup (e. g. wage income is set to "0" values for persons who do not receive wages).

Weights - Survey weights are created to correct coverage, sampling, and non-sampling errors, including biases from unit and item non-response. Their purpose is to ensure that, while used, the sample accurately represents the entire target population. Since an unweighted sample reflects only itself and not the broader population, calibrated weighting factors, which align sample estimates with known population totals, are essential for statistical analyses if a researcher intends to make valid inferences about the target population or specific subpopulations.

All *LWS* files include weight variables calculated by the data producers (at the household level in the H- and R-files and at the individual level in the P-file) that are needed to make the sample representative of the overall population. All of them correct for errors and biases, but do not necessarily inflate to the covered population (please refer to *Data Information* in METIS for more information on weights). Suppose the weight initially provided by the data producer does not inflate to the covered population. In such a few cases, an inflation of the weight to the covered population is conducted by LIS team using the following formula:

$$w_i' = w_i \frac{p}{\sum_{i=1}^{n} w_i}$$

where w_i' is the adjusted weight for each observation so the total matches the given population size, w_i is the original weight for each observation, p is the total population size, and $\sum_{i=1}^{n} w_i$ is the sum of the original weights across all observations. This formula rescales each weight proportionally, ensuring consistency with the population while preserving the relative contribution of each observation.

The variables *hpopwgt* and *ppopwgt* are household and person-level weights normalized to population size and available for all *LWS* datasets. Note that when only the household-level weight was provided, the individual-level weight is set equal to the household-level weight for all household members.

Suppose a researcher pools (append) datasets together for cross-country analysis. Among many, there are two countries whose populations differ significantly, e.g., Luxembourg and the United States. Using the weights that inflate to the population size of each country (*hpopwgt* or *ppopwgt*) in the regression analysis would make the results biased toward the United States because of its population size. Therefore, the weights need to be normalized so a country with a larger population size (United States) does not take over the results from a country with a smaller population size (Luxembourg). One way to normalize survey weights across different samples for cross-country analysis (adopted by LIS team) is to adjust the weights so they sum up to 1 by dividing each observation's weight by the total sum of weights, ensuring proportionality across observations:

$$w' = \left(\frac{w_i}{\sum_{i=1} w_i}\right) \cdot 10{,}000$$

Scaling the weights with multiplication by 10,000 is done for practical use in statistical analyses while maintaining relative proportions. The normalized weights are scaled so that their sum is approximately 10,000. Normalized and scaled weights as described above are represented in variables *hwgt* and *pwgt*, which represent relative proportions within each country and preserve the relative importance of observations within a dataset, independent of the country's population size or sample size. Normalizing and scaling weights ensures that each country's dataset contributes proportionally in both descriptive and inferential analyses.

Finally, LWS also provides two additional weight variables (*hwgta* and *pwgta*) if some questions in the original survey were asked only to a subsample of individuals/ households, and subsample weights were provided in the original data.

In addition, if the data producer provides replicate weights, those are accessible in the additional *LWS* R-file (see *Structure of LWS datasets* above). The replicate weights are crucial for ensuring accurate variance, standard errors, and confidence intervals and, as a result, reducing the bias in statistical inference calculations in complex survey designs. Users are strongly encouraged to use sample weights and replicate weights (if available) in their analysis to correct for bias due to the sampling and data collection processes; this is particularly important for *LWS* datasets, as wealth surveys often oversample the relatively wealthy.

Useful information on *LWS* household balance sheet

Aggregation of wealth variables – The guiding principle for the *LWS* household balance sheet variables is that each original variable is recorded in the asset/liability variable where it fits best. *LWS* wealth variables are constructed over several aggregation stages: total assets is the

sum of non-financial assets, financial assets, and pension assets and other long-term savings. These three sub-components, in their turn, are aggregates from further sub-components. For instance, non-financial assets are the sum of real estate and non-housing assets, whereas real estate is the sum of principal residence and other real estate. The upper-level variables do not always constitute the sum of the lower-level variables due to the lack of sufficient detail for a specific asset/liability. For instance, the amount of financial investments might not always be equal to the sum of its subcomponents such as bonds, stocks, and investment funds/alternative investments, because some of the financial investments collected by the data provider could be a mixture of these subcomponents; therefore, those amounts would be coded directly in financial investments. The calculation of the amount of financial investments that could not be allocated to specific subcategories due to insufficient detail is formally represented as

$$U = F - \sum_{i=1}^{n} C_i$$

This formula expresses *U* as the difference between the total financial investments (*F*) and the sum of its explicitly reported subcategories of financial investments such as bonds, stocks, and investment funds/alternative investments ($\sum_{i=1}^{n} C_i$).

Pension assets availability and the concept of net worth **-** Some surveys do not yet collect all pension assets. In general, this is mostly the case for social security and/or occupational pension entitlements, where sometimes additional computation techniques are needed. In order to separate differences in the measurement of pension assets and to distinguish these practices, LIS has adopted a breakdown of defined-benefit vs. defined-contribution schemes in addition to the breakdown in individual, occupational, and social security pensions. Some data providers just collect the current value of the pension accounts (defined-contribution schemes), whereas they do not calculate the current value of future cash flows, as required in defined-benefit schemes. As soon as some pension assets are not collected or computed, variables pension assets and long-term savings (*has*) and total assets (*ha*) contain only zero values. Thus, for comparability reasons, four measures of net worth are offered in the *LWS* datasets:

1) *disposable net worth* (variable *dnw*) – this variable has defined values if non-financial assets, financial assets (excluding pensions) as well as real estate and non-housing liabilities are available; it excludes the totality of pension assets and other long-term savings (variable *has*);
2) *adjusted disposable net worth* (variable *anw*) – this variable has only defined values if life insurance and individual pension assets (variable *hasi*) are available;
3) *integrated net worth* (variable *inw*) – this variable has defined values when the total value of occupational pensions (*haso*) or its subcomponents, such as defined benefits occupational pensions (*hasodb*) and/or defined contribution occupational pensions (*hasodc*), are available. In rare cases, integrated net worth has defined values if other net worth variables cannot be constructed due to missing information about assets or liabilities, particularly when a collection of assets or liabilities was not exhaustive.

Additionally, in very few instances, integrated net worth includes the total value of pension assets and other long-term savings (*has*), making it equal to total net worth (variable *tnw*).

4) *total net worth* (variable *tnw*) – this variable has only defined values when all pension assets and other long-term savings (variable *has*) or its subcomponents that are relevant in that country (because of institutional settings for pension assets) are available.

***Breakdown of liabilities variables* -** Liabilities are mainly disaggregated following the purpose for which the obligation was made (e.g., real estate, investments, consumer goods, education). An additional breakdown of liabilities by security is also available. Three notes of caution should be highlighted for users of this additional set of liabilities:

- these are additional variables, which overlap with the other liabilities variables, i.e. most amounts included in those variables were also included at some level of disaggregation in the main set of liabilities variables;
- in some cases, the secured loans include business debts while they are not present in the liabilities by purpose due to the fact that all business debts are deducted from business assets and reported on the asset side as business equity;
- the liabilities included in this additional set are NOT always exhaustive; this might happen when all liabilities cannot be categorised as secured or non-secured loans.

Appendix 1

Using LWS Multiply Imputed Data with Stata

```
/*********************************
Using LWS multiply imputed data
Example of the United States (US22)
*********************************/

lissyuse, lws cc(us22) pvars(inum pid hid relation educ) ///
hvars(inum hid dhi dnw haf hpopwgt own )

*replicate weights file
merge m:1 hid using $us22r, assert(match master) keep(match)
*here only reference person for purpose of analysis below
keep if relation==1000

/*****************************
Register multiple-imputed data
*****************************/

/*****************************************************
Creation of the zero implicate.
Stata's mi routines expect implicate
number 0 to contain missing values.
We create this implicate, and set as all
missing values in implicate 0 that vary over
implicates. Implicate 0 is not used in the calculations.
*****************************************************/

expand 2 if inum==1, gen(missing_flag)
replace inum=0 if missing_flag==1
drop missing_flag

global IMPVAR ""
foreach var of varlist relation educ dhi dnw haf own {
 capture confirm numeric variable `var'
  if !_rc {
  tempvar sd count
  quietly bysort hid : egen `sd'=sd(`var')
  quietly bysort hid : egen `count'=count(`var')
  count if ( (`sd'>0 & `sd' <. )  ) & inum==0 & `var'!=.
  if r(N)>0 global IMPVAR "$IMPVAR `var'"
  replace `var'=. if ( (`sd'>0 & `sd' <. )  ) & inum==0
  drop `sd' `count'
  }
}
* Import as multiply imputed data
mi import flong, m(inum) id(hid) clear
```

```
* Register imputed variables
mi register imputed $IMPVAR
* Check if other variables are varying
mi varying
* Summarize to check which variables were imputed
sum $IMPVAR
* Register NOT imputed variables (see above)
mi register regular(relation)
* Declare mi data to be svy with replicate weights
mi svyset, clear
* all 1000 replicate weights, but could be hrwgt1-hrwgt50
mi svyset hid [pw=hpopwgt],bsrweight(hrwgt*) vce(linearized)
* Run simple OLS regression
mi esti, cmdok vceok esampvaryok: svy: regress dnw dhi i.own
```

Appendix 2

Using LWS Multiply Imputed Data with R

```r
# --- Using LWS multiply imputed data ----
# --- Example of the United States (US22) ---

# Load libraries
library(lissyrtools)
library(tidyverse)
library(haven)
library(survey)
library(mitools)

# Alter the "print.MIresult" function in the mitools library for LISSY
print.MIresult <- function(x, ...){
  cat("Multiple imputation results:\n")
  lapply(x$call, function(a) {cat("      "); print(a)})
  out <- data.frame(results = coef(x), se = sqrt(diag(vcov(x))))
  out <- as_tibble(out, rownames = "variable")
  print(out)
}

# Load data using lissyuse()
lissyuse(
  data = c("us22"),
  vars = c("inum", "did", "dhi", "dnw", "haf", "hpopwgt",
           "own", "relation", "age", "sex", "edyrs", "pid")
)
class(lws_datasets) # Note: This is a list of data.frames
names(lws_datasets)

# Create a data.frame
us_data <- bind_rows(lws_datasets, .id = "dataset")

# --- Loading replicate weight files into LISSY ---
file_r <- read.LIS('us22r')

# Merge replicate weights
us_data <- us_data %>%
  left_join(file_r, by = "hid") %>%
  filter(relation == 1000) # Keep reference person for analysis

# --- Analysis using multiple-imputed data ---
imp_list <- purrr::map(1:5, function(x) filter(us_data, inum == x))
models <- with(imputationList(imp_list), glm(dnw ~ dhi + edyrs))
MIcombine(models)


```

```r
# --- Analysis using multiply imputed data and replicate weights ---
us_data_na_zero <- us_data %>%
  mutate(
    across(hrwgt1:hrwgt999, function(x) coalesce(x, 0))
  )
imp_list_na_zero <- purrr::map(1:5, function(x) filter(us_data_na_zero,
    inum == x))
us_designrep <- svrepdesign(type = "bootstrap",
                            repweights = "hrwgt[1-999]",
                            weights = ~hpopwgt,
                            data = imputationList(imp_list_na_zero))
models1_rep <- with(us_designrep, svyglm(dnw ~ dhi + edyrs))
MIcombine(models1_rep)
```