NEPS
National Educational Panel Study

Research Data

Data Documentation: Imputed Data File of Starting Cohort 6

Ariane Würbach, Angelina Hammon, Ferdinand Geissler, and Solange Goßmann

edited by NEPS-Methods Group

A STUDY BY

LIfBi
LEIBNIZ INSTITUTE FOR EDUCATIONAL TRAJECTORIES

Leibniz Institute for Educational Trajectories (LIfBi)
Wilhelmsplatz 3, 96047 Bamberg
Director: Prof. Dr. Hans-Günther Roßbach
Executive Director of Research: Dr. Jutta von Maurice
Executive Director of Administration: Dr. Robert Polgar
Bamberg, 2014

Research Data Papers
at the NEPS Data Center, Bamberg

The NEPS Research Data Paper series presents documentation resources prepared to support the work with data from the National Educational Panel Study (NEPS).

Full citation of this document:
Ariane Würbach, Angelina Hammon, Ferdinand Geissler, and Solange Goßmann. *Data Documentation: Imputed Data File of Starting Cohort 6.* Ed. by NEPS-Methods Group. Bamberg, 2014

This imputed data file of scientific use data from the NEPS Starting Cohort 6 – "Adult Education and Lifelong Learning" was prepared by the staff of the operational unit Methods, Weighting, and Imputation in collaboration with colleagues of the operational unit Returns to Education Across the Life Course and staff from the Data Center. It represents a major collective effort. The contribution of the following staff members of the NEPS is gratefully acknowledged:

Data preparation, editing and imputation
Ariane Würbach (editing of data, imputation, documentation)
Angelina Hammon (editing of data, examples)
Ferdinand Geißler (selection of variables, data processing)
Solange Goßmann (initialization, imputation)
Christian Aßmann (imputation)
Anika Biedermann (selection of variables)
Mihaela Tudose (management and editing of metadata)
Tobias Koberg (documentation)

Data documentation
Ariane Würbach
Angelina Hammon
Ferdinand Geißler
Solange Goßmann

Leibniz Institute for Educational Trajectories (LIfBi)
National Educational Panel Study (NEPS)
Data Center and Method Development
Wilhelmsplatz 3
96047 Bamberg, Germany

E-mail: fdz@lifbi.de
Web: https://www.neps-data.de/en-us/datacenter
Phone: +49 951 863 3511

This document has been typeset with XeLaTeX 0.9999 using the free font Linux Libertine from the Libertine Open Fonts Project.

Contents

List of syntax examples

List of Figures

List of Tables

1 Introduction

This manual is intended to assist your work with the imputed data of the NEPS Starting Cohort 6 – *Adult Education and Lifelong Learning* (SC6_Imputation_1-0-0) based on the first release: http://dx.doi.org/10.5157/NEPS:SC6:1.0.0. We aim at providing a guide on how to use these data for your research. Therefore, our focus is on practical aspects of data usage such as the data set structure, refinements comparing to the preceding version SC6_D-1.0.0, and examples for application.

This manual is not an all-embracing documentation resource. Please consult our website for background information on the studies, survey instruments, a structured documentation, and many more resources, especially on the original data set: https://www.neps-data.de/en-us/datacenter.

We aim at keeping this manual as short and simple as possible. At several places, we refer to supplementary documents presenting additional information that we consider essential for working with our data. The following supplements you can find attached:

- Supplement A – List of new and modified variables
- Supplement B – Filter structure

Additionally, there is a NEPS working paper available in which the procedure of imputation is described comprehensively (cf. Aßmann, Würbach, Goßmann, Geissler, and Biedermann [1]). You can download this document here:

→ www.neps-data.de > Project overview > Publications > NEPS Working Papers

We welcome feedback from our users that will help us improve the quality of this manual and our data for future releases. Please report any feedback to:

- fdz@lifbi.de

1.1 Obtaining the data

There are three simple steps to obtain the data of this file:

- Sign the data use contract and mail it to us:

  → www.neps-data.de > Data Center > Data Access > Data Use Agreements

- After approval, sign in as a registered NEPS user at the login at https://www.neps-data.de
- Access the data via Download from our website (after login).

This data file will be provided in a download version. Note that files offered for download include data with the highest level of anynomization due to sensitivity of the data. These data are available to registered users from the web portal via a secure connection. For details see the Data Manual of SC6_1.0.0 [12]. Also, find additional information on our website at:

→ www.neps-data.de > Data Center > Data Access

For a quick reference on which steps of anonymization are undertaken and on which level you will find your desired information, please see Supplement E in the Data Manual of SC6_1.0.0.

File Format

All files from SC6_D-1.0.0 are available in Stata and SPSS format. The imputed data file is available only in Stata format, with English and German labels as well.

1.2 General conventions

The *naming of the data file* follows a number of conventions which are summarized in the Data Manual for Starting Cohort 6 – Version SC6_D-1.0.0 [12]. The physical file `SC6_Imputation_1-0-0.dta` refers to the *Download*-version (*D*) for the imputed data file of *SC6 – Adult Education and Lifelong Learning* of data release *1.0.0*.

The *variable naming* conventions are aimed at ensuring the consistency of variable names across panel waves. They reflect the panel structure of the NEPS data and allow users to conveniently identify variables across waves. In this release we follow strictly the conventions for the formalia of the original data used for imputation [12].

1.3 Publications with NEPS data

If you publish with NEPS data, it is mandatory to quote the following reference:

> Blossfeld, H.-P., Roßbach, H.-G., & von Maurice, J. (Eds.). (2011). Education as a lifelong process: The German National Educational Panel Study (NEPS) [Special issue]. *Zeitschrift für Erziehungswissenschaft, 14*.

In addition, publications using data from this data file must include the following acknowledgement:

This paper uses data from the National Educational Panel Study (NEPS): Starting Cohort 6 – Adults, doi 10.5157/NEPS:SC6:1.0.0. From 2008 to 2013, NEPS data were collected as part of the Framework Programme for the Promotion of Empirical Educational Research funded by the German Federal Ministry of Education and Research (BMBF). As of 2014, the NEPS survey is carried out by the Leibniz Institute for Educational Trajectories (LIfBi) at the University of Bamberg in cooperation with a nationwide network.

A digital object identifier (DOI) uniquely identifies each release of NEPS data (cf. Wenzig [30]). The DOI of this data file redirects to a landing page providing basic information on the data:

http://dx.doi.org/10.5157/NEPS:SC6:1.0.0

2 Preliminaries for imputation

2.1 Why imputation?

Survey data is usually accompanied by incomplete information due to nonresponse – either unit-nonresponse, or item-nonresponse. Resulting missing values have to be taken into account when analyses are performed. Unit-nonresponse, the drop out of an individual, is typically addressed by means of weighting procedures. Item-nonresponse arises when respondents are not able or willing to give a valid answer, usually due to difficult or highly sensitive questions. Especially when asking for income data, relatively high rates of item-nonresponse can be observed, see e.g. Riphahn and Serfling [20]. These missing values are usually handled by imputation. Most straightforward by multiple imputation, where missings are routinely replaced by several plausible sets of values which yields repeated-imputation inferences, see Rubin [22, p. 75], thus taking the uncertainty due to missing information into account.

This manual and the concurrent NEPS working paper (cf. Aßmann, Würbach, Goßmann, Geissler, and Biedermann [1]) document the applied multiple imputation technique which was performed in the context of household net income data. Please keep in mind, that several processes preceding and during imputation were led by this main topic. Hence, the selection of variables was a subject matter consideration. Variables assumed to be influential in prediction of household net income, in prediction of missingness as well as accommodating the structure, for example interactions, are chosen. Also the whole data editing process was aimed at comprising the data to meet the demands of the applied multiple imputation technique with respect to household income data. Last but not least, the multiple imputation algorithm was adapted correspondingly to capture all the special features necessary for imputation of income data within a complex data structure with possible nonlinear relationships, and many skewed and categorical variables. In addition to that skip patterns and constraints have to be considered ensuring that no invalid values are imputed. Also bracketed income information is available and included as restrictions for imputation of the open income questions, though imputed passively themselves, see van Buuren [28, p. 130].

2.2 Data structure and data sets

The NEPS surveys engender extensive and complex data. For usability the scientific use files are structured in a user-friendly way and a number of additional data sets from one or more of the original files have been already generated by the *Research Data Center LIfBi* to ease the preparation and analysis of life course data. This also applies to the data from the second wave of the NEPS Starting Cohort 6 (adults). For the imputed data set presented here further steps have been made to create a suitable framework for analysis: one single data set for all respondents adjusted for missing values, comprehensive episode information, and useful generated variables is produced. So, neither matching of data sets nor further data edition is necessary. The exact procedure of constructing this multiply imputed data set containing valuable merged and prepared data is described in the following section.

Overall, the number of respondents is 11,649: 5,154 are first-time respondents in NEPS and 6,495 are re-interviewed (first-time respondents in ALWA, see the Data Manual of SC6_1.0.0 [12] for more details). Starting from 22 subfiles, of which the main release consists, four subfiles are taken into account for imputation purpose: The panel file (`pTarget`), the employment module (`spEmp`) and two generated files (`Basics`, `Methods`) are regarded as being essential for forecasting income. In total there are 1,125 variables, from which 243 are chosen by the operational unit *Returns to Education Across the Life Course* in accordance to their direct or indirect relation to income data, or for prediction of missingsness respectively. After data processing 213 variables remain and are entered into the model (respondent identifier `ID_t` and imputation identifier `impit` not counted), see Supplement B section A.2 on page 31.

2.3 Data processing

For a detailed description of the original data sets we would like to refer to Leopold, Raab, and Skopek [12]. The questionnaire starts and ends with cross-sectional modules. In the course of those, panel respondents are often only asked for updates. All cross-sectional information collected at both panel waves is merged into one single subfile (`pTarget`). Starting from this, `Basics` has been generated by the *Research Data Center LIfBi* and comprehends current sociodemographic as well as summarized biographical information on the respondent. Between the cross-sectional modules, retrospective information on the respondents' life-course is collected within ten separate longitudinal modules. A check module identifies inconsistencies in the sequence of episodes. Resulting corrections of time durations or objection after rejection are passed to the `Biography` subfile which contains all episode information from all longitudinal modules with respect to their starting and ending time as well as their continuing. Therefore, spells and specified time within the longitudinal modules are proven by means of `Biography` before enlisting for imputation (linking variables: respondent identifier `ID_t` and spell identifier `splink`).

Data is originally stored in long-format, one row representing one respondent at different times (wave identifier `wave` in `pTarget`), or different episodes of a respondents area of live respectively (episode identifier `spell` in `spEmp`). One of the prerequisites of current imputation methods is the availability of individual information given in one single row. For this, information from different waves was *integrated*, i.e. information from the current wave is selected if the question has been repeatedly questioned, or information from the preceding wave otherwise. From longitudinal data harmonized spells (`subspell=0`), either complete or right-censored are selected in a first step. In a second step information from episodes is selected or aggregated. *Selection* criteria for episodes of the employment module (`spEmp`) is: the information of the last episode is selected, independent if the episode is continuing or not. Current information from main occupation and from side job or activity with training character is regarded separately. Information from side job or activity with training character is reduced to their duration, amount of income and their continuing, and incorporated as additional variables. For *aggregation* three modes are applied: information is dichotomized (e.g. an indicator variable for main job vs. side job or job with training character), summarized (e.g. total duration of employment) or indices are built (e.g. number of special payments).

Keeping the usability and the *paradigm of original information esteem* in mind, only few alterations are actually done to the data set. These modifications always account for the fact that information may never be lost completely. Though, information is aggregated into coarse categories, new variables or selection of spell data per each respondent. Please note that all original information is still available in version SC6_1.0.0 and only the imputed data set is constraint in this matter. In fact, 45 variables are modified in some way – which is about 19 percent of the imputed data set and about four percent of the whole original volume. In the Appendix an explanatory overview of all modifications as well as new built variables is given (see Supplement A section A.1 on page 28). The variables are listed in their order of appearance.

Finally, all four subfiles are merged into one single data set.

2.4 Skip patterns and constraints

The complex structure of skip patterns and the hierarchy of variables require special handling. Manifold types of skip patterns determine the sequence of variables to be imputed and constraints restrict the ranges of admissible values for imputation at an individual level. The admissible ranges can vary considerably between the respondents. Skip patterns need to be differentiated from true missing values, because they are *not relevant* and have to be excluded from imputation. First, filtered variables within a module need to be regarded, e.g. if a person stated, that no special payment was received, the follow-up question according to the amount of special payments was *not relevant*. Second, filtered variables across modules have to be taken into account, e.g. year of birth defines a filter for retirement. Third, whole module can be skipped because they did not apply, e.g. when a respondent has not have any employment episode there has no employment history module to be imputed. The first two types of skip

patterns are incorporated as restrictions, e.g. when the respondent is filtered the answer concerning the follow-up question will be automatically set to a residual class labelled as *does not apply* or *not determinable* in `Basics`, respectively. The third skip pattern is taken into account by running different imputations for respondents with merely information from `Basics` and `pTarget`, and respondents with additional information from `spEmp`. Furthermore, empirical implausible values are considered. Inconsistencies, e.g. if the reported individual net income exceeds the reported gross income, are explored and treated as missing, thus subject to imputation. Whereas skip patterns determine the sequence of variables subject to imputation, dependencies among the variables have to be considered as well when setting up the sequence of imputation models. An example for such a dependency in form of a logical constraint is given for net and gross income. Net income defines the lower bound for imputation of gross income and gross income defines the upper bound for imputation of net income. The number of missing values thereby determines the sequence of mutually dependent variables for imputation. The variable showing fewer missings is imputed first, and then, used for imputation of the dependent variable. Variables restricting the admissible range of other variables in a unidirectional way are imputed in the first place as well. An example is given with age and age at first employment where age defines a lower bound for age at first employment but not vice versa. Bracketed income information represents a special case of such an unidirectional dependency. Income brackets define lower or upper bounds for the imputation of exact income, though are only subject to passive imputation, see van Buuren [28]. However, passively imputed values are disregarded for imputation of exact income in the next iteration.

2.5 Missing values

The original data sets of SC6_D-1.0.0 provide different missing codes for different situations of missing values. In general, we distinguish between missing codes indicating sorts of item nonresponse, not applicable missings, and edition missings. The Data Manual of SC6_1.0.0 [12] gives an overview of missing codes you will encounter in the NEPS data.

As part of the editing process for imputation missing values are inserted if there was any information from a follow-up question available which could be used to logically infer to the filter question, e.g. a respondent gave an answer on how many hours of overtime he had last month, so the preceding question if he had overtime last month was set to *yes* if it was missing. *Not determinable* denotes missing data that occurs because the item does not apply to a person. This category comprises three kinds of missings: design missings (due to sample-specific questionnaires), skip patterns (the question does not apply to a person), and missings that occur for unkown reasons (system missings). For convenience, the code *–99* is set for not applicable information. In the generated modules `Basics` and `Biography` the corresponding code is *-55* for *not determinable*. Any remaining missing values due to item nonresponse – differentiated in the original data set into: *refused*, *do not know* or *implausible values removed* – are summarized and imputed. So, for distinction of the type of missing the researcher is encouraged to consult the original SC6_D-1.0.0 data sets.

3 Generating multiply imputed data

3.1 Imputation method

The implemented multiple imputation method is based on a nonparametric tree-based sequential classification and regression approach combining the partition algorithm CART (Classification and Regression Trees), see Breiman, Friedman, Olshen, and Stone [5], and the imputation technique MICE (Multivariate Imputation by Chained Equations), see van Buuren, Brand, Groothuis-Oudshoorn, and Rubin [29] and van Buuren [28]. In a first step, the missing values are initialized by draws from the unconditional empirical distribution, for variables with skip patterns and constraints taking the sequence of the full conditional distributions into account. In a second step, the missing values are replaced by draws from the partitions identified by CART as approximations of the full conditional distributions. The second step is performed $L = 10$ times to mitigate the effect of initialization, and further $M = 100$ iterations are stored from which 20 imputations are provided in an imputed data set.

For a more detailed description of the imputation method see Aßmann, Würbach, Goßmann, Geissler, and Biedermann [1]. This multiple imputation approach assures that standard errors can be estimated correctly, allows for nonlinear relationships among the surveyed variables, and is capable with several types of variables (nominal, ordinal, count, and continuous). Imputation is performed for a large set of variables (213). We considered such a large frame to ensure inclusion of all possible predictors of the variables with missing values, thus making the imputation operationally and statistically more efficient (cf. Raghunathan and Bondarenko [18]). Special features of the imputation method applied for this study are: it preserves logical consistencies among variables. Hence, skip patterns and constraints are considered correctly at the same time. This is done by implementation of a $N \times K$-matrix for $i = 1,\ldots,N$ observations and $j = 1,\ldots,K$ variables, which contains lists of all restrictions and admissible ranges meshing with a highly flexible scheme containing the hierarchy of questions, both uniting the complex filter structures in itself. Since the focus of imputation lies on household income data, some features of the applied imputation technique directly address the income variables. This means, additional information from the classified income variables is used in our model. First, system missings for bracketed income information are inserted as follows: respondents are classified into the rough and fine brackets according to their answer in the open income question. Second, for respondents with merely classified income data, these are considered as admissible ranges for imputation of the open income query.

All variables with missing values are imputed in the context of each other, simultaneously as in the manner of chained equations. Since only a selection of variables is included in the imputation models, imputations will not reflect potential relationships with variables excluded from the models. Because of this, we strongly recommend not to use the imputed data set to analyse relationships between variables in this data set and non-imputed variables merged from another subfile of the original data.

3.2 Data file

The imputed data set assembles cross-sectional and aggregated information for a selection of subfiles and variables. It is no longer necessary for a researcher to merge the data sets by himself. A list containing all selected variables as well as an overview concerning all filters and their retracement is given in Supplement B in section A.2 on page 31.

In order to provide a most convenient data structure, the different imputation iterations are already merged in the respective file. The imputed data file is stored in long-format containing an identifier variable `impit` that indicates each single imputation iteration. `impit=0` represents the data line before imputation, per each respondent. Thus, the researcher can either run complete cases analyses, or use indicator variables to account for missingness in predictor variables for prediction of missingness in the outcome variable, as proposed by Cohen and Cohen [9]. On completion we produced 20 imputed data sets containing 213 variables and 11,649 observations (upon request more imputed data sets are available). Each imputation iteration is indicated by `impit=1` up to `20`. For investigation purposes researches can run analyses 20 times and compare the results. We recommend researchers to do analyses on combined estimates, parameters and variances, resulting from all imputation iterations while applying Rubin's Combining Rules to ensure correct inferences (cf. Rubin [22] and Rubin and Schenker [23]). Examples for computation with R and Stata are given in section 4 on page 12.

Data preparation, imputation and analyses are performed using `R version 3.0.1`, required packages are `foreign` [16], `tree` [21], `lattice` [24], `VIM` [27], `plyr` [32], `survey` [13], `psych` [19], `lme4` [2], `Amelia` [11], and `mitools` [14]. Our code is an adaption of the basic implementation of `treeMI` [7] from Burgette and Reiter [6], which is available at `http://www.burgette.org/software.html`.

3.3 Data description (selection)

In this paragraph some descriptions of the data – especially the income variables – are given. Different visualization techniques are employed to explore incomplete or imputed data. All graphs can be reproduced easily using the `R code` from example 3 on page 25 in this documentation. At first, the kernel densities for household net income as well as individual net income are displayed. A typical unimodal distribution can be seen which is skewed to the right, see figure 1 on the next page. In the figure, densities before imputation (blue line), and the first three imputation iterations (thin orange lines) are compared. The results of the imputations resemble the densities before imputation closely which indicates the applied imputation method to be highly plausible.

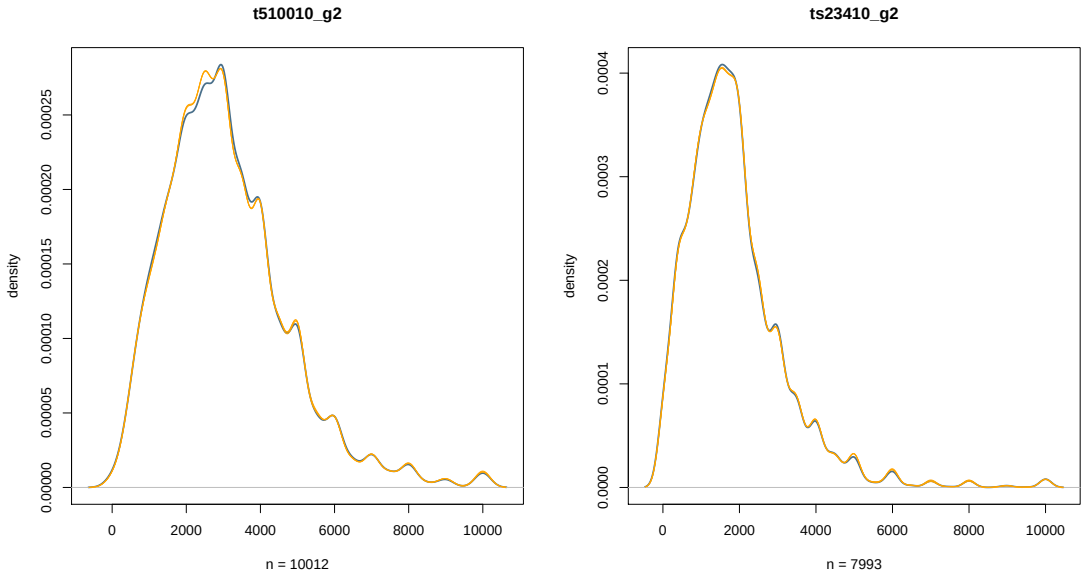

Figure 1: Kernel density estimates for the marginal distribution of the observed data (blue) and $m = 3$ densities calculated from the imputed data (thin orange lines) for household net income (left side) and individual net income (right side).

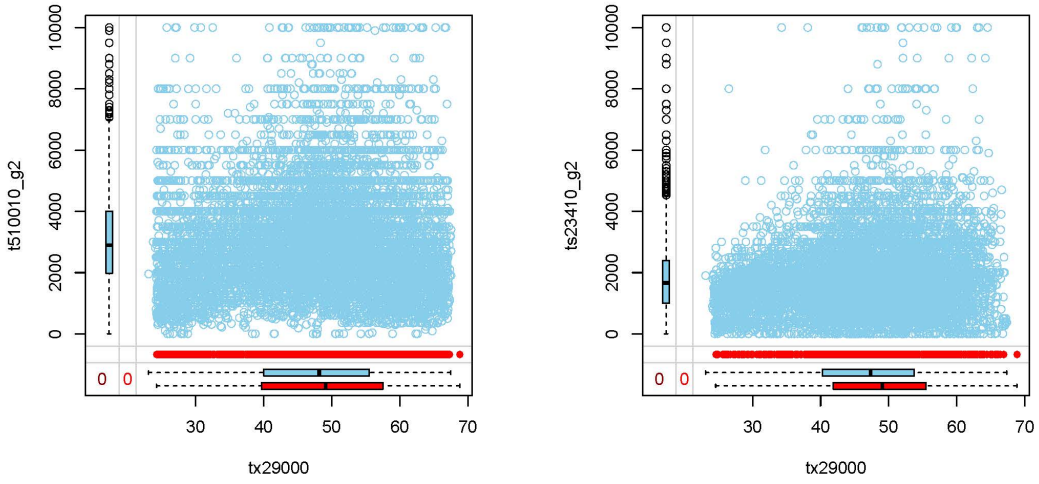

Figure 2: Marginal distribution of household net income (left side) and individual net income (right side) against age. The boxplots in the bottom line indicate the distributions of age for observed income data (blue) and missing income data (red).

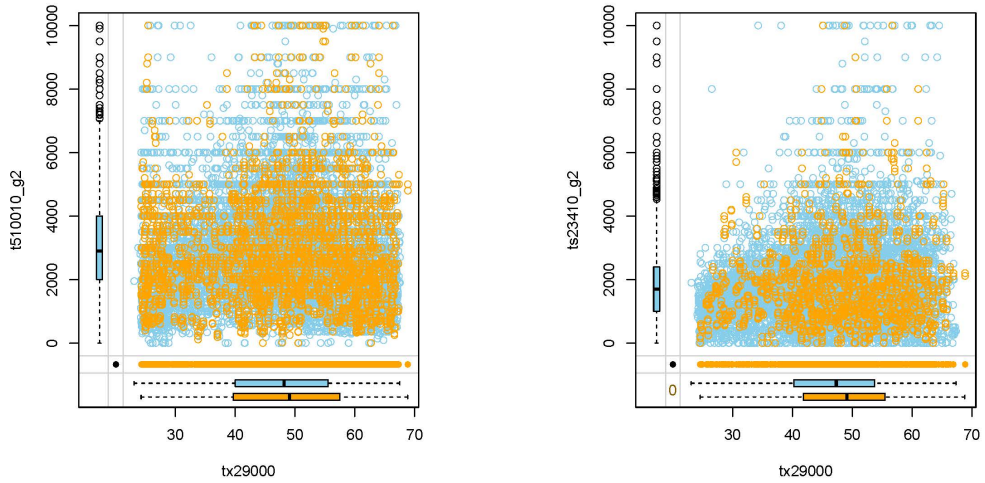

Figure 3: Marginal distribution of household net income (left side) and individual net income (right side) against age after imputation (observed data blue, and imputed data orange).

The function `marginplot` is used to illustrate the distribution of an independent variable for respondents with and without missing data in the variable of interest, according to Prantner [15]. Here, both income variables are plotted against age. In the `VIM` package, per default, the color-scheme for all graphical methods is blue for observed values, red for missing values and orange for imputed values. As indicated by the boxplots at the bottom, the mean age is somewhat higher for respondents with missing income data (red boxplot vs. blue boxplot), see figure 2 on page 9. At the ordinate only a blue boxplot is given for the corresponding income distribution, which means that age had no missing values to be imputed. The next figure 3 on the previous page shows the marginal distribution after applying the imputation algorithm. In figure 4 the rough income brackets for household and individual net income after imputation are shown separated by gender for each split (1 – *up to 1,500 Euro*, 2 – *1,500 up to 3,000 Euro*, 3 – *more than 3,000 Euro*). The proportions of imputes are given in orange, indicating the imputed values to be almost equally distributed across male and female respondents. In these figures, no departures from the MCAR (missing completely at random) assumption can be seen, neither for age nor gender.

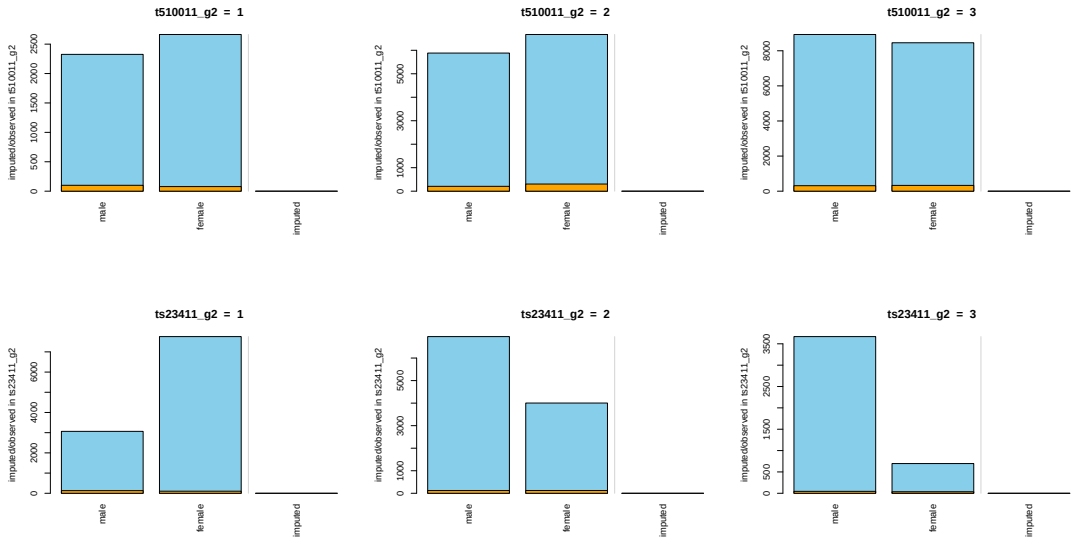

Figure 4: Proportion of imputes in household net income splits (above) and in individual net income splits (below) according to gender (observed data blue, and imputed data orange).

3.3 Notes

The NEPS invested a lot to ensure the integrity of these data. However, we strongly recommend you to examine the data critically when you work with this data file. Furthermore, you should always consult the questionnaire/s and data manual corresponding to the original data set on which the imputation builds up to obtain a precise understanding of how the data have been collected and edited.

4 Examples

This section gives some examples on how to perform proper analyses of a multiply imputed data set via Rubin's Combining Rules [22, 23]. The aim of the examples is to assure that estimates and standard errors are calculated correctly and not to answer substantial research questions. For this purpose the examples are kept as small as possible. Standard complete data analyses are performed on each imputed data set and the resultant estimators and their variances are then combined. The pooled estimator is purely the average of all single estimators. And for the variance, the average within-imputation variance and the between-imputation variance have to be aggregated. While the first two examples focus on estimators from either descriptives (mean), or regression analysis, the last example contains typical diagnostic plots to evaluate imputation, as proposed by Prantner [15] and Raghunathan and Bondarenko [18].

We provide the code to run the examples in R and Stata. Note that an operating version of at least `R 3.0.1`, or `Stata 12` respectively is needed, to execute the corresponding syntaxes.

4.1 Example 1 – Mean statistics with a multiply imputed data set

The first example illustrates the implementation of Rubin's Combining Rules [22, p. 76] in a simple descriptive analysis. Note that calibrated design weights are considered for the estimates before applying Rubin's Combining Rules [31, 34, 26]. The M repeated complete-data estimates and associated complete-data variances for the parameters of interest can be combined as follows. The degrees of freedom are calculated according to Rubin and Schenker [23]. For more details about the fractional increase in variance, and fraction of information missing due to nonresponse (γ), see Rubin [22, p. 77f.]. The fraction of missing data can be compared with the fraction of missing information. In many cases the fraction of missing values will exceed the fraction of missing information. This is due to the fact that the multiple imputation utilize information contained in inter-variable relationships for prediction of the missing values (cf. Schafer [25, p. 15]). The proportion of the variance attributable to the missing data (λ), is given as well, see van Buuren [28, p. 41]. Analyses are repeated in a survey design.

The replicate script of Damico [10] gives a good introduction for R in this context, and Carlin, Galati, and Royston [8] as well as Schimpl-Neimanns [26] provide a sufficient overview for application in Stata.

Note, the combining rules are given in detail in the R example and for Stata by means of the mim function. We used mim: reg instead of mi estimate since it enables us to consider robust estimates for standard errors as well.

R example 1: Example 1 in R

```
##############################################################################################
# Calculation of mean statistics
#
#
# 1. Open the datasets
# 2. Calculate the estimators for the mean of the household net income (t510010_g2)
#    for all respondents and for different demographic subgroups:
#     - separated by sex (t700001)
#     - separated by age groups (tx29000 -> "age"):
#       20-29 years
#       30-39 years
#       40-59 years
#       60+ years
#     - separated by sex & age groups (t700001 & age)
# 3. Apply weighted estimation procedures using the standardized calibrated weights
#    (weight_mc09_std)
# 4. Calculate the combined estimators and variances using Rubin's Combining Rules
# 5. Compare fraction of information missing with proportion of missing values
# 6. Run analyses as a multiply imputed survey design object
#
##############################################################################################


rm(list=ls())


if (!require("foreign")) install.packages("foreign")
if (!require("psych"))   install.packages("psych")
if (!require("plyr"))    install.packages("plyr")
if (!require("Amelia"))  install.packages("Amelia")
if (!require("mitools")) install.packages("mitools")
if (!require("survey"))  install.packages("survey")


library(foreign)
library(psych)
library(plyr)
library(Amelia)
library(mitools)
library(survey)
options(scipen=10)
```

```
#-------------------------------------------------------------------------------
weighted.var <- function(x, w, na.rm = FALSE) {
    if (na.rm) {
        w <- w[i <- !is.na(x)]
        x <- x[i]
    }
    sw <- sum(w)
    sw2 <- sum(w^2)
    mw <- sum(x * w) / sum(w)
    (sw / (sw^2 - sw2)) * sum(w * (x - mw)^2, na.rm = na.rm)
}
# usual formula
#-------------------------------------------------------------------------------


# 1.
#*******************************************************************************
# open the imputed dataset (long format) and take a subset:
Impdata <- read.dta("{path}\\SC6_Imputation_1-0-0.dta", convert.factors=FALSE)
Imp <- subset(Impdata, select=c(ID_t,impit,t700001,tx29000,tx20000,tx29060,
                                t510010_g2,t70000y,tx28101))

# create a new variable for age groups:
Imp$age <- ifelse(Imp$tx29000 >= 20 & Imp$tx29000 < 30, "20-29 years",
           ifelse(Imp$tx29000 >= 30 & Imp$tx29000 < 40, "30-39 years",
           ifelse(Imp$tx29000 >= 40 & Imp$tx29000 < 60, "40-59 years", "60+ years")))

# merge weights
Method <- read.dta("{path}\\SC6_Methods_D_1-0-0.dta", convert.factors=FALSE)
subsetMethod <- subset(Method, select=c(ID_t,wave,weight_mc09_std,psu,stratum))
subMeth <- subsetMethod[subsetMethod$wave==2,-2]
wdat <- merge(Imp, subMeth, by="ID_t", all=TRUE)


# 2./3.
#*******************************************************************************
# number of imputation iterations
m <- 20

# vectors and matrices for the different means of the household net income
mean_hh.al <- matrix(NA, nrow=m, ncol=1)
colnames(mean_hh.al) <- "all"

mean_hh.sex <- matrix(NA, nrow=m, ncol=2)
colnames(mean_hh.sex) <- c("male","female")

mean_hh.age <- matrix(NA, nrow=m, ncol=4)
colnames(mean_hh.age) <- c("20-29 years","30-39 years","40-59 years","60+ years")

mean_hh.sexage <- matrix(NA, nrow=m, ncol=8)
colnames(mean_hh.sexage) <- c("male 20-29 years","male 30-39 years",
                              "male 40-59 years","male 60+ years",
                              "female 20-29 years","female 30-39 years",
                              "female 40-59 years","female 60+ years")
```

```
mean_hh.est <- matrix(NA, nrow=m, ncol=15)

for(i in 1:m)
{
# exclude respondents which skipped survey (n=6)
app <- which(wdat$t510010_g2!=-99 & wdat$impit==i)
# calculation of the weighted mean m-times:
# for all respondents
mean_hh.est[i,1] <- mean_hh.al[i] <- weighted.mean(wdat$t510010_g2[app],
                                     w=wdat$weight_mc09_std[app])
# for different sex
sex <- ddply(wdat[app,], .(t700001), summarize,
             wm = weighted.mean(t510010_g2,weight_mc09_std))[,2]
mean_hh.est[i,c(2,3)] <- mean_hh.sex[i,] <- sex
# for different age groups
age <- ddply(wdat[app,], .(age), summarize,
             wm = weighted.mean(t510010_g2,weight_mc09_std))[,2]
mean_hh.est[i,c(4:7)] <- mean_hh.age[i,] <- age
# for different age groups separated by sex
sex.age <- ddply(wdat[app,], .(t700001, age), summarize,
                 wm = weighted.mean(t510010_g2,weight_mc09_std))[,3]
mean_hh.est[i,c(8:15)] <- mean_hh.sexage[i,] <- sex.age
}

# vectors and matrices for the different variances of the mean
# of the household net income
var_hh.al <- matrix(NA, nrow=m, ncol=1)
colnames(var_hh.al) <- "all"

var_hh.sex <- matrix(NA, nrow=m, ncol=2)
colnames(var_hh.sex) <- c("male","female")

var_hh.age <- matrix(NA, nrow=m, ncol=4)
colnames(var_hh.age) <- c("20-29 years","30-39 years","40-59 years","60+ years")

var_hh.sexage <- matrix(NA, nrow=m, ncol=8)
colnames(var_hh.sexage) <- c("male 20-29 years","male 30-39 years",
                             "male 40-59 years","male 60+ years",
                             "female 20-29 years","female 30-39 years",
                             "female 40-59 years","female 60+ years")

var_hh.est <- matrix(NA, nrow=m, ncol=15)

# case numbers per group
uni <- which(wdat$t510010_g2!=-99 & wdat$impit==1)
tsex <- table(wdat$t700001[uni])
tage <- table(wdat$age[uni])
tsexage <- table(wdat$t700001[uni],wdat$age[uni])

for(i in 1:m) {
# exclude respondents which skipped survey (n=6)
app <- which(wdat$t510010_g2!=-99 & wdat$impit==i)
# calculation of the variances (sigma^2/n) m-times:
```

```
# for all respondents
var_hh.est[i,1] <- var_hh.al[i,] <- weighted.var(wdat$t510010_g2[app],
                                    w=wdat$weight_mc09_std[app])/length(uni)
# for different sex
sex.v <- ddply(wdat[app,], .(t700001), summarize,
               wm = weighted.var(t510010_g2,weight_mc09_std))
var_hh.est[i,c(2,3)] <- var_hh.sex[i,] <- sex.v[,2]/tsex
# for different age groups
age.v <- ddply(wdat[app,], .(age), summarize,
               wm = weighted.var(t510010_g2,weight_mc09_std))
var_hh.est[i,c(4:7)] <- var_hh.age[i,] <- age.v[,2]/tage
# for different sex separated by age groups
sex.age.v <- ddply(wdat[app,], .(t700001, age), summarize,
                   wm = weighted.var(t510010_g2,weight_mc09_std))
var_hh.est[i,c(8:15)] <- var_hh.sexage[i,] <- sex.age.v[,3]/t(tsexage)
}

# calculate the standard deviation
sd_hh.est <- sqrt(var_hh.est)


# 4.
#*******************************************************************************
# run the function to combine estimated means and variances, calculate CI,
# fraction of missing information, lambda as well as proportion of missings

CombImpMean <- function(estmean, se, m)
{
if (ncol(estmean)!=ncol(se) | nrow(estmean)!=nrow(se)) {
   stop ("Dimensions of estimates and corresponding standard errors differ!") }

# average estimate
aveest <- colSums(estmean)/m
dif <- estmean - matrix(1, nrow=m, ncol=1) %*% aveest
bvar <- colSums(dif^2)/(m-1)            # between-variance
wvar <- colSums(se^2)/m                 # within-variane
tvar <- wvar + bvar * ((m+1)/m)         # total-variance
# degrees of freedom according to Rubin/Schenker (1986)
defr <- (m-1) * (1 + (m/(m+1)) * wvar/bvar)^2
# 95% confidence intervals
ci_l <- aveest - qt(0.975, defr) * sqrt(tvar)
ci_u <- aveest + qt(0.975, defr) * sqrt(tvar)
# fractional increase in variance due to nonresponse, see Rubin (1987)
k <- 1                                  # number of parameters
if (ncol(estmean)>1) {
   r <- (1 + 1/m) * tr(diag(bvar)*solve(diag(tvar)))/k
} else if (ncol(estmean)==1) {
   r <- (1 + 1/m) * (bvar*tvar)/k
} else stop ("Vector of estimates is empty!")

# fraction of information missing due to nonresponse (gamma), see Rubin (1987)
gamm <- (r + 2)/(defr + 3)/(r + 1)
# (lambda), see van Buuren (2012)
lamb <- (1 + 1/m) * bvar/tvar
```

```
# request output
out <- cbind(round(aveest,3), round(sqrt(tvar),3), round(ci_l,3), round(ci_u,3),
             round(defr,3), round(gamm,5), round(lamb,4))
colnames(out) <- c("estimate","standard.error","CI95.lower.bound",
                   "CI95.upper.bound","df","fmi","lambda")
return(out)
}

# pooled means and standard deviations for the household net income
CombImpMean(mean_hh.est, sd_hh.est, m)

# compare with function mi.meld from package 'Amelia'
mi.meld(q=mean_hh.est, se=sd_hh.est, byrow=TRUE)


# 5.
#*******************************************************************************
# Compare fraction of information missing with proportion of missing values
ndat <- wdat[wdat$impit==0,]
sel <- which(!is.na(ndat$t510010_g2) & ndat$t510010_g2==-99)
ndat$nmis <- ifelse(is.na(ndat$t510010_g2), 1, 0)

# for all respondents
weighted.mean(ndat$nmis[-sel], w=ndat$weight_mc09_std[-sel])
# for different sex
ddply(ndat[-sel,], .(t700001), summarize, wm = weighted.mean(nmis,weight_mc09_std))
# for different age groups
ddply(ndat[-sel,], .(age), summarize, wm = weighted.mean(nmis,weight_mc09_std))
# for different sex separated by age groups
ddply(ndat[-sel,], .(t700001, age), summarize,
      wm = weighted.mean(nmis,weight_mc09_std))


# 6.
#*******************************************************************************
# run analyses by a multiply imputed survey design object
# see: https://github.com/ajdamico/usgsd/blob/master/Consumer%20Expenditure%20Survey/
# 2011%20fmly%20intrvw%20-%20analysis%20examples.R

# create the survey design object
idat <- wdat[wdat$impit!=0,]
inc_design <- svydesign( ids=~psu, strata=~stratum, weights=~weight_mc09_std,
                         data=imputationList(split(idat, idat$impit)), nest=TRUE )

#exclude respondents which skipped survey (n=6)
inc_design <- subset( inc_design, t510010_g2 != -99   )
options(survey.lonely.psu = "adjust")
dim(inc_design)

# average household net income
MIcombine( with(inc_design, svymean( ~t510010_g2 )) )

# by age
MIcombine( with(inc_design, svyby( ~t510010_g2, ~age, svymean )) )
```

```
# gender should be treated as a factor:
inc_design <- update( t700001=factor(t700001), inc_design )

# by gender
MIcombine( with(inc_design, svyby( ~t510010_g2, ~t700001, svymean )) )
```

Stata example 1: Example 1 in Stata

```
/*******************************************************************************
 Calculation of mean statistics

  1. Open the datasets
  2. Calculate the estimators for the mean of the household net income (t510010_g2)
     for all respondents and for different demographic subgroups:
      - separated by sex (t700001)
      - separated by age groups (tx29000 -> "age"):
        20-29 years
        30-39 years
        40-59 years
        60+ years
      - separated by sex & age groups (t700001 & age)
  3. Apply weighted estimation procedures using the standardized calibrated weights
     (weight_mc09_std)
  4. Calculate the combined estimators and variances using Rubin's Combining Rules
  5. Compare fraction of information missing with proportion of missing values
  6. Run analyses as a multiply imputed survey design object

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

ssc install ice, replace
ssc install mim, replace


* 1.
*------------------------------------------------------------------------------
* open the imputed dataset (long format) and take a subset:
use ID_t impit t700001 tx29000 tx20000 tx29060 t510010_g2 t70000y tx28101 ///
    using "{path}\SC6_Imputation_1-0-0.dta",clear
sort ID_t
save {path}\data1, replace

* merge weights
use ID_t wave weight_mc09_std psu stratum ///
    using "{path}\SC6_Methods_D_1-0-0.dta",clear
sort ID_t
keep if wave==2
save {path}\data2, replace

use "{path}\data1.dta"
merge m:1 ID_t using "{path}\data2.dta"
```

```
* create a multiply imputed dataset for mitools
gen _mj=impit
gen _mi=ID_t

mim: check
mim: genmiss t510010_g2

gen sub = 1 if t510010_g2 != -99

* create a new variable for age groups:
generate age=1
replace age=2 if tx29000 >= 30 & tx29000 < 40
replace age=3 if tx29000 >= 40 & tx29000 < 60
replace age=4 if tx29000 >= 60

* set lables
label define age_lbl 1 ”20-29 years” 2 ”30-39 years” ///
                     3 ”40-59 years” 4 ”60+ years”

label values age age_lbl
label define sex_lbl 1 ”male” 2 ”female”
label values t700001 sex_lbl

* save data before analyzing
save {path}\data_imp, replace


* 2./3./4.
*-------------------------------------------------------------------------------
* calculate the mean and the variances m-times and combine the estimates
* for all respondents
mim: mean t510010_g2, over(sub)

* for different sex
mim: mean t510010_g2, over(sub t700001)

* for different age groups
mim: mean t510010_g2, over(sub age)
/*  _subpop_1: 20-29 years
    _subpop_2: 30-39 years
    _subpop_3: 40-59 years
    _subpop_4: 60+ years    */

* for different sex separated by age groups
mim: mean t510010_g2, over(sub t700001 age)
/*  _subpop_1: male 20-29 years
    _subpop_2: male 30-39 years
    _subpop_3: male 40-59 years
    _subpop_4: male 60+ years
    _subpop_5: female 20-29 years
    _subpop_6: female 30-39 years
    _subpop_7: female 40-59 years
    _subpop_8: female 60+ years  */
```

```
* 5.
*-------------------------------------------------------------------------------
* calculate proportion of missing data
egen int nmiss = rowmiss(t510010_g2) if _mj==0
tab nmiss


* 6.
*-------------------------------------------------------------------------------
*consider the hierarchical structure and build a survey design object
/*see more detailed: http://www.gesis.org/fileadmin/upload/forschung/publikationen/
gesis_reihen/gesis_methodenberichte/2009/gesis_mb_09_02.pdf*/
svyset psu [iw=weight_mc09_std], strata(stratum) singleunit(certainty)

mim: svy linearized, subpop(sub): mean t510010_g2
mim: svy linearized, subpop(sub): mean t510010_g2, over(age)
mim: svy linearized, subpop(sub): mean t510010_g2, over(t700001)
```

4.2 Example 2 – Linear regression with a multiply imputed data set

In the example shown below we want to illustrate the usage of a multiply imputed data set with regard to Rubin's combining rules in the context of a linear regression. The first approach considers robust standard errors, the second takes the two-stage sampling design into account via a mixed-effects model that includes besides the predictor variables, and the variable of interest those variables necessary for stratification and calibration. The third utilizes a design-based approach and a survey design object is build before estimation of regression.

In R package `Amelia` is available which calculates combined estimates and corrected standard errors. Note that the function `mi.meld` provides no confidence intervals, test statistics and fraction of missing information (γ), or λ at all! Another package that provides combined estimates and corrected standard errors is the function `MIcombine` from the package `mitools`. Note that, in order to calculate combined estimates by means of `MIcombine` the imputed data file has to be converted in an `imputationList` before.

R example 2: Example 2 in R

```
################################################################################
# Estimation of a linear regression
#
# 1. Open the datasets
# 2. Calculate the estimators for beta and the variances for every imputed dataset
#    (m=20) by fitting the linear regression model:
#    - dependent variable: household net income (t510010_g2)
#    - independent variables: age (tx29000),
#                             number of children in the household (tx20000)
#                             current employment (tx29060)
# 3. Calculate the combined estimators and variances using Rubin's Combining Rules
# 4. Compare results with Amelia and mitools
```

```
# 5. Run another regression with regard to the two-stage sampling design
#    and the variables used for calibration
# 6. Run analyses as a multiply imputed survey design object
#
################################################################################################

rm(list=ls())

if (!require("foreign"))   install.packages("foreign")
if (!require("survey"))    install.packages("survey")
if (!require("lme4"))      install.packages("lme4")
if (!require("Amelia"))    install.packages("Amelia")
if (!require("mitools"))   install.packages("mitools")
if (!require("psych"))     install.packages("psych")
if (!require("sandwich"))  install.packages("sandwich")

library(foreign)
library(survey)
library(lme4)
library(Amelia)
library(mitools)
library(psych)
library(sandwich)

# 1.
#*********************************************************************************************
# open the imputed dataset (long format) and take a subset:
Impdata <- read.dta("{path}\\SC6_Imputation_1-0-0.dta", convert.factors=FALSE)
Imp <- subset(Impdata, select=c(ID_t,impit,t700001,tx29000,tx20000,tx29060,
                                t510010_g2,t70000y,tx28101))

# merge stratification and weighting information
Method <- read.dta("{path}\\SC6_Methods_D_1-0-0.dta", convert.factors=FALSE)
subsetMethod <- subset(Method, select=c(ID_t,wave,weight_mc09_std,psu,stratum))
wdat <- merge(Imp, subsetMethod[subsetMethod$wave==2,-2], by="ID_t", all=TRUE)


# 2.
#*********************************************************************************************
# vector for the m estimates of the coefficients and the standard deviations
beta_est <- sd_est <- NULL
# fit the model to all imputed data sets:
m <- 20

for(i in 1:m) {
# restrict cases to applicables
app <- which(wdat$t510010_g2!=-99)
# variable to identify the number of the imputed data set: impit
lin_reg  <- lm(t510010_g2[app] ~ tx29000[app] + tx20000[app] + tx29060[app],
               data=wdat[wdat$impit==i,])
beta_est <- rbind(beta_est, lin_reg$coef)
# use robust estimates for SE -> White's (1980) estimators:
sd_est   <- rbind(sd_est, sqrt(diag(vcovHC(lin_reg, type="HC"))))
}
```

```
# 3.
#*******************************************************************************
# run the function to combine estimates and variances, and calculate CI

CombImpLinReg <- function(est, se, m)
{
if (ncol(est)!=ncol(se) | nrow(est)!=nrow(se)) {
   stop ("Dimensions of estimates and corresponding standard errors differ!") }

# average estimate
aveest <- colSums(est)/m
dif <- est - matrix(1, nrow=m, ncol=1) %*% aveest
bvar <- colSums(dif^2)/(m-1)             # between-variance
wvar <- colSums(se^2)/m                  # within-variane
tvar <- wvar + bvar * ((m+1)/m)          # total-variance
# degrees of freedom according to Rubin/Schenker (1986)
defr <- (m-1) * (1 + (m/(m+1)) * wvar/bvar)^2
# 95% confidence intervals
ci_l <- aveest - qt(0.975, defr) * sqrt(tvar)
ci_u <- aveest + qt(0.975, defr) * sqrt(tvar)
# t-test
t_value <- aveest / sqrt(tvar)
# p-value
p_value <- 2 * (1 - pt(abs(t_value), defr))
# fractional increase in variance due to nonresponse, see Rubin (1987)
k <- ncol(est)                           # number of parameters
if (ncol(est)>1) {
   r <- (1 + 1/m) * tr(diag(bvar)*solve(diag(tvar)))/k
} else if (ncol(est)==1) {
   r <- (1 + 1/m) * (bvar*tvar)/k
} else stop ("Vector of estimates is empty!")

# fraction of information missing due to nonresponse (gamma), see Rubin (1987)
gamm <- (r + 2)/(defr + 3)/(r + 1)
# (lambda), see van Buuren (2012)
lamb <- (1 + 1/m) * bvar/tvar

# request output
out <- cbind(as.numeric(round(aveest,3)), round(sqrt(tvar),3), round(ci_l,3),
             round(ci_u,3), round(defr,3), round(t_value,3), round(p_value,4),
             round(gamm,5), round(lamb,3))
colnames(out) <- c("estimate","standard.error","CI95.lower.bound",
                   "CI95.upper.bound","df","t.ratio","p.value","fmi","lambda")
return(out)
}

CombImpLinReg(beta_est, sd_est, m)


# 4.
#*******************************************************************************
# compare with function mi.meld from package 'Amelia'
mi.meld(q=beta_est, se=sd_est, byrow=TRUE)
```

```
# compare with function MIcombine from package 'mitools'
idat <- wdat[wdat$impit!=0,]
sel <- which(idat$t510010_g2 != -99)
ildat <- imputationList(split(idat[sel,], idat$impit[sel]))
models <- with(ildat, lm(t510010_g2 ~ tx29000 + tx20000 + tx29060))
summary(MIcombine(models))


# 5.
#*******************************************************************************
# do now consider the sampling design and nonresponse adjustment: add variables
# used for calibration: year of birth (t70000y) and highest CASMIN (tx28101)
beta_est <- sd_est <- NULL

for(i in 1:m)
{
# variable to identify the number of the imputed data set: impit
# restrict to subset of applicables
fit  <- lmer(t510010_g2 ~ 0+ tx29000 + tx20000 + tx29060 + t70000y + tx28101 + (1|psu)
             + (1|stratum), subset=idat$t510010_g2!=-99, data=idat[idat$impit==i,])
beta_est <- rbind(beta_est, fixef(fit))
se <- sqrt(diag(vcov(fit, useScale = FALSE)))
sd_est <- rbind(sd_est, se)
}
CombImpLinReg(beta_est, sd_est, m)


# 6.
#*******************************************************************************
# create the survey design object
inc_design <- svydesign( ids=~psu, strata=~stratum, weights=~weight_mc09_std,
                         data=imputationList(split(idat[sel,], idat$impit[sel])),
                         nest=TRUE )
inc_design <- update( t700001=factor(t700001), inc_design )
options(survey.lonely.psu = "adjust")

# do regression
summary( MIcombine( with(inc_design, svyglm( t510010_g2 ~ tx29000 + tx20000 + tx29060,
                    family=gaussian() )) ))
```

Stata example 2: Example 2 in Stata

```
/*******************************************************************************
 Estimation of a linear regression

 1. Open the datasets
 2. Calculate the estimators for beta and the variances for every imputed dataset
    (m=20) by fitting the linear regression model:
    - dependent variable: household net income (t510010_g2)
    - independent variables: age (tx29000),
                             number of children in the household (tx20000)
                             current employment (tx29060)
```

```
 3. Calculate the combined estimators and variances using Rubin's Combining Rules
 4. Run another regression with regard to the two-stage sampling design
    and the variables used for calibration
 5. Run analyses as a multiply imputed survey design object

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

ssc install ice, replace

* 1.
*-------------------------------------------------------------------------------
* open the imputed dataset (long format) and take a subset:
use ID_t impit t700001 tx29000 tx20000 tx29060 t510010_g2 t70000y tx28101 ///
    using "{path}\SC6_Imputation_1-0-0.dta",clear
sort ID_t
save {path}\data1, replace

* merge methods data:
use ID_t wave psu stratum ///
    using "{path}\SC6_Methods_D_1-0-0.dta",clear
sort ID_t
keep if wave==2
save {path}\data2, replace

use "{path}\data1.dta"
merge m:1 ID_t using "{path}\data2.dta"

* create a multiply imputed dataset for mitools
generate _mj=impit
generate _mi=ID_t
mim: check

generate sub = 1 if t510010_g2 != -99

* set labels
label values tx29000 age_lbl
label values t700001 sex_lbl

* save data before analyzing
save {path}\data_imp, replace


* 2./3.
*-------------------------------------------------------------------------------
* fit the model to all imputed datasets and combine the estimates and
* robust standard errors
micombine reg t510010_g2 tx29000 tx20000 tx29060 if sub==1, vce(robust)

* degrees of freedom
display e(df_m)

* fraction of missing information
matrix list e(gamma)
```

```
* 4.
*-------------------------------------------------------------------------------
* do now consider the sampling design and nonresponse adjustment:
* fit multiple group effects with multiple group effect terms and add variables
* used for calibration: year of birth (t70000y) and highest CASMIN (tx28101)

mim: xtmixed t510010_g2 tx29000 tx20000 tx29060 t70000y tx28101 ///
     if sub==1 || psu: , noconstant || stratum: , noconstant


* 5.
*-------------------------------------------------------------------------------
* consider the hierarchical structure and build a survey design object
svyset psu [iw=weight_mc09_std], strata(stratum) singleunit(certainty)

mim: svy linearized, subpop(sub): regress t510010_g2 tx29000 tx20000 tx29060
```

4.3 Example 3 – Check imputations by diagnostic plots

In the last example, we explore the imputed data and compare it with the complete data before imputation. This example is delivered only in R code due to availability of special graphical outputs with respect to multiple imputation analysis. Those diagnostic tools can help to check the reasonability of the created imputations (cf. Prantner [15] and Raghunathan and Bondarenko [18]). Exemplarily, the household and individual net income from the open question (continuous data) as well as the rough split (ordered data) will be illustrated.

R example 3: Example 3 in R

```
################################################################################
# Check imputations by diagnostic plots
#
# 1. Open the imputed dataset, keep imputation iteration zero (before imputation),
#    three imputation iterations, and select variables of interest
# 2. Graphical outputs for net income data (t510010_g2, ts23410_g2)
# 3. Graphical outputs for split income data (t510011_g2, ts23411_g2)
#
################################################################################

rm(list=ls())

if (!require("foreign"))   install.packages("foreign")
if (!require("lattice"))   install.packages("lattice")
if (!require("VIM"))       install.packages("VIM")

library(foreign)
library(lattice)
library(VIM)
options(scipen=10)
```

```
# 1.
#*******************************************************************************
# open the imputed dataset (long format) and keep the data before imputation:
Impdata <- read.dta("{path}\\SC6_Imputation_1-0-0.dta", convert.factors=FALSE)
Imp <- subset(Impdata[order(Impdata[,2]),], impit==0|impit==1|impit==2|impit==3,
              select=c("ID_t","impit","t700001","tx29000","t510010_g2","ts23410_g2",
                       "t510011_g2","ts23411_g2"))
BImp <- subset(Imp, impit==0)


# 2./3.
#*******************************************************************************
# diagnostic plots (see Raghunathan/Bondarenko 2007):
# plot densities before imputation
dens <- BImp[,5:6]
colnames(dens) <- c("t510010_g2","ts23410_g2")
Dens <- stack(dens)
densityplot(~ values[values >= 0 & values <= 10000] | ind, Dens, as.table=TRUE,
            xlab="", ylab="density")

# plot income data against age (continuous) - location of missings:
par(mfrow=c(1,2))
for(i in 5:6) {
marginplot(BImp[is.na(BImp[,i]) | BImp[,i] >= 0 & BImp[,i] <= 10000,c(4,i)])
}

# define delimiter for identification of imputes:
hhi <- ifelse(is.na(Imp[Imp[,2]==0,5]), TRUE, FALSE)
Imp$t510010_g2_imp <- rep(hhi,4)
ini <- ifelse(is.na(Imp[Imp[,2]==0,6]), TRUE, FALSE)
Imp$ts23410_g2_imp <- rep(ini,4)
shhi <- ifelse(is.na(Imp[Imp[,2]==0,7]), TRUE, FALSE)
Imp$t510011_g2_imp <- rep(shhi,4)
sini <- ifelse(is.na(Imp[Imp[,2]==0,8]), TRUE, FALSE)
Imp$ts23411_g2_imp <- rep(sini,4)
NImp <- Imp[-which(Imp$impit==0),]

# plot income data against age (continuous) - location of imputes:
par(mfrow=c(1,2))
for(i in 5:6) {
marginplot(NImp[NImp[,i] >= 0 & NImp[,i] <= 10000,c(4,i,i+4)], delimiter="_imp")
}

# plot densities before imputation and for three imputation iterations:
par(mfrow=c(1,2))
for(j in 5:6) {
plot(density(BImp[!is.na(BImp[,j]) & BImp[,j] >= 0 & BImp[,j] <= 10000,j]),
     xlab=paste("n =", length(BImp[!is.na(BImp[,j]) & BImp[,j] >= 0
                & BImp[,j] <= 10000,j])),
     ylab="density", main=paste(names(BImp)[j]), lwd=2, col="skyblue4")

for(i in 1:3) {
lines(density(Imp[Imp[,j] >= 0 & Imp[,j] <= 10000 & Imp[,2]==i,j]), col="orange")
}}
```

```
# plot split information before imputation:
par(mar=c(8.1,4.1,4.1,2.1), mfrow=c(1,2))
for(i in 7:8) {
barMiss(BImp[,i], ylab="frequency", xlab="", sub=NULL, main=paste(names(BImp)[i]),
        labels=c("filtered","< 1,500 Euro","1,500-3,000 Euro","> 3,000 Euro"))
}

# plot imputes of income splits against gender:
NImp <- Imp[-which(Imp$impit==0),]

par(mar=c(9.1,4.1,4.1,2.1), mfrow=c(2,3))
for(i in 7:8) {
for(j in 1:3) {
barMiss(NImp[NImp[,i]==j,c(3,i,i+4)], delimiter="_imp", xlab="", sub=NULL,
        main=paste(names(Imp)[i]," = ",j), interactive=FALSE,
        labels=c("male","female"))
}}
```

5 Further information

Please visit our web portal for further information and comprehensive documentation resources such as PAPI and CATI questionnaires, how-to guides, technical reports, and the codebook.

→ www.neps-data.de > Data Center > Research Data > Starting Cohort Adults

For further support, please contact the NEPS data center:

E-mail: fdz@lifbi.de
Web: → www.neps-data.de > Data Center > Contact the Data Center
Phone: +49 951 863 3511 (Mo-Fr 10:00-12:00 and 14:00-16:00)

Participation in the NEPS user trainings

Furthermore, the NEPS data center offers training courses on a regular basis. These courses introduce the research design of the NEPS, the structure of data sets, terms and conditions of data usage, issues of privacy and data protection, and so on. A central module of the courses consists of hands-on work with the NEPS data supervised by our staff. As skill levels, research interests, and methods vary greatly across users and disciplines, we will offer a comprehensive portfolio of seminars ranging from introductory topics on a rather general level to advanced methodological courses.

A Appendix

A.1 Supplement A – List of new and modified variables

Table 1: List of new and modified variables

Pos.	Name	Label	Origin	Modification
2	impit	imputation iteration	new	numeric: 0 – before imputation; 1:20 – imputation iteration
3	panel	panel respondent	wave	classified: 1 – panel respondent; 0 – first time respondent
13	t40503y_g1	years since immigration	inty (Biography), t40503y	numeric: inty-t40503y
20	t413000_g1D	first native language	t413000_D	classified: 1 – German; 2 – not German
21	t413010_g1D	second native language	t413010_D	classified: 1 – German; 2 – not German; 3 – no second native language
29	t725001_g1	number of class repetitions	t725001 – t725013, t725015	numeric: sum
39	t320307	working environment 2	t320307	summarized: -6 – I have no superiors/ I have no colleagues
74	t524204_g1	years since recognized disability	inty (Biography), t524204	numeric: inty-t524204
79	t413100_g1D	native language of the mother	t413100_D	classified: 1 – German; 2 – not German
88	t413120_g1D	native language of the father	t413120_D	classified: 1 – German; 2 – not German
100	t510011_g2	net household income, split	t510010/_v1,t510011/_v1	open income information inserted
101	t510012_g2	net household income, categories up to 1,500 Euro	t510010/_v1,t510012/_v1	open income information inserted
102	t510013_g2	net household income, categories between 1,500 and 3,000 Euro	t510010/_v1,t510013/_v1	open income information inserted
103	t510014_g2	net household income, categories over 3,000 Euro	t510010, t510014	open income information inserted
143	tx29070_g1	years since first employment	tx29000, tx29070	numeric: tx29000-tx29070
154	spell_g1	episode of main activity	spell	numeric: subspell=0; selected main activity
155	ts23102_g1	type of activity	ts23102 – ts23104	classified: 1 – main job; 2 – side job
163	startm_g1	duration of employment episode	startm, starty, endm, endy (Biography)	numeric: (endy-starty)*12+(endm-startm), for selected main activity
164	splast_g1	the employment persists	splast (Biography)	classified: 1 – yes; 2 – no; -55 – not determinable
186	ts23410_g2	net income main activity	ts23410	numeric: for selected main activity
187	ts23411_g2	net income main activity, 1. split	ts23410, ts23411/_v1	open income information inserted
188	ts23412_g1	net income main activity, 2. split, categories up to 1,500 Euro	ts23410, ts23412/_v1	open income information inserted
189	ts23413_g1	net income main activity, 2. split, categories between 1,500 and 3,000 Euro	ts23410, ts23413/_v1	open income information inserted
190	ts23414_g1	net income main activity, 2. split, categories for more than 3,000 Euro	ts23410, ts23414	open income information inserted
191	ts23510_g2	gross income main activity	ts23510	numeric: for selected main activity
192	ts23511_g2	gross income main activity, 1. split	ts23510, ts23511	open income information inserted

(...)

Table 1: (continued)

Pos.	Name	Label	Origin	Modification
193	ts23512_g1	gross income main activity, 2. split, categories up to 1,500 Euro	ts23510, ts23512	open income information inserted
194	ts23513_g1	gross income main activity, 2. split, categories between 1,500 and 3,000 Euro	ts23510, ts23513	open income information inserted
195	ts23514_g1	gross income main activity, 2. split, categories for more than 3,000 Euro	ts23510, ts23514	open income information inserted
198	ts23531_g1	special payments	ts23531 – ts23536	numeric: sum
199	ts23541_g1	amount of special payments	ts23541 – ts23546	numeric: sum
200	ts23102_g2	secondary activity	ts23102 – ts23104	classified: 1 – sideline job; 2 – training; 3 – no
201	spell_g2	episode of secondary activity	spell	numeric: selected secondary activity
202	splast_g2	secondary activity persists	splast (Biography)	classified: 1 – yes; 2 – no; -55 – not determinable
203	startm_g2	duration of secondary activity episode	startm, starty, endm, endy (Biography)	numeric: (endy-starty)*12+(endm-startm), for selected secondary activity
204	ts23410_g3	net income secondary activity	ts23410	numeric: for selected secondary activity
205	ts23411_g3	net income secondary activity, 1. split	ts23410, ts23411/_v1	open income information inserted
206	ts23412_g2	net income secondary activity, 2. split, categories up to 1,500 Euro	ts23410, ts23412/_v1	open income information inserted
207	ts23413_g2	net income secondary activity, 2. split, categories between 1,500 and 3,000 Euro	ts23410, ts23413/_v1	open income information inserted
208	ts23414_g2	net income secondary activity, 2. split, categories for more than 3,000 Euro	ts23410, ts23414	open income information inserted
209	ts23510_g3	gross income secondary activity	ts23510	numeric: for selected secondary activity
210	ts23511_g3	gross income secondary activity, 1. split	ts23510, ts23511	open income information inserted
211	ts23512_g2	gross income secondary activity, 2. split, categories up to 1,500 Euro	ts23510, ts23512	open income information inserted
212	ts23513_g2	gross income secondary activity, 2. split, categories between 1,500 and 3,000 Euro	ts23510, ts23513	open income information inserted
213	ts23514_g2	gross income secondary activity, 2. split, categories for more than 3,000 Euro	ts23510, ts23514	open income information inserted
214	startm_g3	duration of employment in total	startm, starty, endm, endy (Biography)	numeric: (endy-starty)*12+(endm-startm), in total
215	emp	employment	new	classified: 0 – no employment episode; 1 – at least one employment episode

A.2 Supplement B – Filter structure

Table 2: Filter structure

Module	Variable name	Variable label	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5
	ID	ID_t					
	imputation iteration	impit					
pTarget							
	panel respondent	panel					
	gender	t700001					
	year of birth	t70000y					
	contentment with life	t514001					
	contentment with financial situation	t514002					
	contentment with health	t514003					
	contentment with family life	t514004					
	contentment with circle of friends and acquaintances	t514005					
	contentment with work	t514009					
	born in Germany or abroad	t405000_ha					
	years since immigration	t40503y_g1	t405000_ha$= 2$				
	migrational status	t406000	t405000_ha$= 2$				
	German nationality	t406050					
	German nationality since birth	t406060_ha	t406050$= 1$				
	intention to naturalize	t406120	t406050$= 2$				
	residence permit	t406130	t406050$= 2$				
	work permit	t406140	t406050$= 2$	t406130$= 1$			
	first native language	t413000_g1D					
	second native language	t413010_g1D	**t413020_ag**$\neq$NA & $= -99 \rightarrow 2\vert 3$				
	subjective knowledge of German: comprehension	t41330c					
	subjective knowledge of German: speaking	t41330d					
	subjective knowledge of German: reading	t41330b					
	subjective knowledge of German: writing	t41330a					
	age at which respondent began learning German	t413020	t413000_g1D$= 2$ & t413010_g1D$= 2\vert 3$				
	kindergarten attendance	t712001					
	repeated school year	t725000					
	number of class repetitions	t725001_g1	t725000$= 1$				
	final grade highest school-leaving qualification	t724201	panel$= 1$				
	ego: career	t320005					
	ego: further education	t320006					
	military medical examination	t530001	t700001$= 1$				
	degree of physical fitness at first military medical examination	t530011	t700001$= 1$	t530001$= 1\vert 2$	t70000y≥ 1970		
	suitability for military service at first military medical examination	t530012	t700001$= 1$	t530001$= 1\vert 2$	t530011$= -99$		
	efforts to obtain particular position or stationing	t530020	t700001$= 1$	t530001$= 1\vert 2$	t70000y≥ 1970		
	application for extended military service	t530030	t700001$= 1$	t530001$= 1\vert 2$	t70000y≥ 1970		
	working environment 1	t321305					
	working environment 2	t320307	t321305$\neq -6$				
	working environment 3	t321306	t321305$\neq -5\vert -6$				
	global resources 1	t321301	t321305$\neq -5\vert -6$				
	global resources 2	t321302	t321305$\neq -5\vert -6$				
	global resources 3	t321303	t321305$\neq -5\vert -6$				

(...)

Table 2: (continued)

Module	Variable name	Variable label	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5
	person is currently employed/working	t779001					
	job search in the past 4 weeks	th09211					
	availability	th09212	th09211= 1	t779001= 0			
	information job	t324040	**t325020**≠NA | **t323030**≠NA →≠ −5				
	information job: number of people	t32404b	t324040= 1|2				
	information job: migrants	t32404d	t324040= 1|2				
	information job: degree	t32404e	t324040= 1|2				
	reference for job, given by somebody from environment	t325020	t324040= 1|2|3|4				
	reference job: number of people	t32502b	t325020= 1|2				
	reference job: migrants	t32502d	t325020= 1|2				
	reference job: degree	t32502e	t325020= 1|2				
	help with application	t323030	t324040= 1|2|3|4				
	cohabitation with a partner	t733002					
	living with a partner	t733003	panel= 0				
	living apart together	t733004	t733003= 2				
	contact frequency with partner	t733005	t733003= 2	t733004= 1			
	minor children in the household	t742003					
	time for childcare (hours per day)	t744001	t742003= 1				
	help with childcare	t744002	t742003= 1				
	informal congress visit	t271800					
	informal lectures	t271801					
	informal reading	t271802					
	self-learning programs	t271803					
	social circle: further education	t324570					
	social circle further education: number of people	t32457b	t324570= 1				
	social circle further education: number migration background	t32457d	t324570= 1				
	social circle further education: number with degree	t32457e	t324570= 1				
	state of health	t521000					
	recognized disability	t524200					
	degree of disability (percent)	t524205	t524200= 1				
	years since recognized disability	t524204_g1	t524200= 1				
	type of family situation up to age 14	t731101					
	number of siblings	t732101	panel= 0				
	number of older siblings	t732102	panel= 0	t732101> 0			
	birthplace of the mother	t405060_ha					
	native language of the mother	t413100_g1D	panel= 0				
	mother still alive	t731207					
	highest school-leaving qualification of the mother	t731301_ha					
	highest vocational education qualification of the mother	t731303_ha					
	employment of the mother	t731401	**t731402**≠NA & ≠ −99 →≠ 1				
	former employment of the mother	t731402	t731401= 2| − 5				
	management position of the mother	t731407	t731402= 1				
	person older than forty years	t700020					
	birthplace of the father	t405090_ha					
	native language of the father	t413120_g1D	panel= 0				
	father still alive	t731257					
	highest school-leaving qualification of the father	t731351_ha					

(...)

Table 2: (continued)

Module	Variable name	Variable label	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5
	highest vocational education qualification of the father	t731353_ha					
	employment of the father	t731451	**t731452**≠NA & ≠ −99 →≠ 1				
	former employment of the father	t731452	t731451= 2| − 5				
	management position of the father	t731457	t731452= 1				
	expectations father: achieve a lot professionally	t320435	t700020= 0	t731257= 1*			
	expectations father: continually learn new things	t320436	t700020= 0	t731257= 1*			
	expectations mother: achieve a lot professionally	t320475	t700020= 0	t731207= 1*			
	expectations mother: continually learn new things	t320476	t700020= 0	t731207= 1*			
	household net income	t510010_g2	**split variables!**				
	net household income, split	t510011_g2	t510010_g2=NA				
	net household income, categories up to 1,500 Euro	t510012_g2	t510010_g2=NA	t510011_g2= 1			
	net household income, categories between 1,500 and 3,000 Euro	t510013_g2	t510010_g2=NA	t510011_g2= 2			
	net household income, categories over 3,000 Euro	t510014_g2	t510010_g2=NA	t510011_g2= 3			
	household language	t413500_ha					
	care work last 12 months	t745001					
	frequency care work	t745002	t745001= 1				
	number of hours of care work	t745003	t745001= 1				
	care assistance social circle	t745004	t745001= 1				
	financial contributions last 12 months	th09241					
	sum of financial contributions last 12 months	th09242	th09241= 1				
	financial aid from social circle	th09243					
	global resource friends: women	t321101					
	global resource friends: migrants	t321102	t321101≠ −6				
	global resource friends: degree	t321103	t321101≠ −6	t321102≠ −6			
	expectations of friends: achieve success on a professional level	t320105	t321101≠ −6	t321102≠ −6	t321103≠ −6		
	expectations of friends: continually learn new things	t320106	t321101≠ −6	t321102≠ −6	t321103≠ −6		
	Burt check	t320910					
	Burt: number of people	t32091b	t320910= 1				
	position generator: nurse/carer	t32600a					
	position generator: engineer	t32600b					
	position generator: warehouse/transport worker	t32600c					
	position generator: social worker	t32600d					
	position generator: shop assistant	t32600e					
	position generator: policeman/policewoman	t32600f					
	position generator: doctor	t32600g					
	position generator: bank clerk	t32600h					
	position generator: motor mechanic	t32600k					
	position generator: judge	t32600l					
	position generator: optician	t32600m					
	position generator: translator	t32600n					
	position generator: teacher at elementary school, Hauptschule or Realschule	t32600o					
Basics							
	age at interview month	tx29000					
	family status	tx27000					

(...)

Table 2: (continued)

Module	Variable name	Variable label	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5
	German nationality	tx29004					
	birth in Germany	tx29005					
	household size	t741001					
	children in the household	tx20000					
	highest CASMIN	tx28101					
	currently employed	tx29060					
	duration of current employment (months)	tx29061	tx29060= 1				
	occupational status current employment	tx29062	tx29060= 1				
	ISEI of current employment	tx29063	tx29060= 1				
	years since first employment	tx29070_g1	emp= 1	tx29000-tx29070_g1≥ 14 years*			
	duration of first employment (months)	tx29071	emp= 1				
	occupational status first employment	tx29072	emp= 1				
	ISEI of first employment	tx29073	emp= 1				
	currently unemployed	tx29080					
	duration of current unemployment (months)	tx29081	tx29080= 1				
	registered as unemployed	tx29082	tx29080= 1				
	receipt of benefit in current unemployment	tx29083	tx29080= 1				
	occupational status of the mother	t731404					
	occupational status of the father	t731454					
Methods							
	BIK-10 municipality size (inhabitants)	tx80102					
spEmp							
	episode of main activity	spell_g1	ts23102_g1= 1				
	type of activity	ts23102_g1	ts23102_g1= 1				
	occupational status	ts23203	ts23102_g1= 1	**ts23212**≠NA | **ts23213**≠NA →≠ 7|8			
	management position	ts23212	ts23102_g1= 1	ts23203≠ 7|8*			
	management position: number of employees	ts23213	ts23102_g1= 1	ts23203≠ 7|8	ts23212= 1*		
	work within the second labour market	ts23215	ts23102_g1= 1	ts23203= 1|2*	**ts23216**= −99 → 1|2		
	temporary employment	ts23216	ts23102_g1= 1	ts23215= 3	ts23911= 1|3|4|6		
	seasonal work	ts23217	ts23102_g1= 1	ts23203= 1|7	ts23215= 3		
	type of employment	ts23911	ts23102_g1= 1				
	duration of employment episode	startm_g1	ts23102_g1= 1				
	the employment persists	splast_g1	ts23102_g1= 1				
	current employment	ts23901	ts23102_g1= 1				
	contractual working hours at the beginning	ts23219	ts23102_g1= 1				
	working hours of position at the end/today	ts23221	ts23102_g1= 1				
	in partial retirement	ts23222	ts23102_g1= 1	ts23901= 1	ts23911= 1|2	t70000y< 1955	
	actual working hours at the end	ts23223	ts23102_g1= 1				
	overtime	ts23224	ts23102_g1= 1	ts23203= 1|2|3	ts23901= 1|2	ts23911= 1|2|3	
	type of compensation for overtime	ts23225	ts23102_g1= 1	ts23224= 1	ts23901= 1|2	ts23911= 1|2|3	
	overtime last month	ts23226	ts23102_g1= 1	ts23224= 1	ts23901= 1	ts23911= 1|2|3	
	number of overtime hours last month	ts23227	ts23102_g1= 1	ts23224= 1	ts23226= 1	ts23901= 1	ts23911= 1|2|3
	further training measures in company: works agreement	ts23229	ts23102_g1= 1	ts23911= 1|2|3|5	ts23901= 1|2		

(...)

Table 2: (continued)

Module	Variable name	Variable label	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5
	further training measures in company: plan	ts23230	ts23102_g1= 1	ts23911= 1|2|3|5	ts23901= 1|2		
	further training measures in company: financing	ts23231	ts23102_g1= 1	ts23911= 1|2|3|5	ts23901= 1|2		
	person responsible for further training measures in company	ts23232	ts23102_g1= 1	ts23911= 1|2|3|5	ts23901= 1|2		
	further professional training: offer to be released from work	ts23233	ts23102_g1= 1	ts23911= 1|2|3|5|8			
	further professional training: offer of financial support	ts23234	ts23102_g1= 1	ts23911= 1|2|3|5|8			
	place of work in Germany/abroad	ts23236	ts23102_g1= 1				
	federal state in which workplace is located	ts23237_g1	ts23102_g1= 1	ts23236= 1*			
	public sector	ts23241	ts23102_g1= 1	ts23203≠ 5|6			
	size of company	ts23243	ts23102_g1= 1				
	temporary employment	ts23310	ts23102_g1= 1	ts23203≠ 5			
	conversion to permanent contract	ts23320	ts23102_g1= 1	ts23310= 1			
	net income main activity	ts23410_g2	ts23102_g1= 1	ts23901= 1	**split variables!**		
	net income main activity, 1. split	ts23411_g2	ts23102_g1= 1	ts23901= 1	ts23410_g2=NA		
	net income main activity, 2. split, categories up to 1,500 Euro	ts23412_g1	ts23102_g1= 1	ts23901= 1	ts23410_g2=NA	ts23411_g2= 1	
	net income main activity, 2. split, categories between 1,500 and 3,000 Euro	ts23413_g1	ts23102_g1= 1	ts23901= 1	ts23410_g2=NA	ts23411_g2= 2	
	net income main activity, 2. split, categories for more than 3,000 Euro	ts23414_g1	ts23102_g1= 1	ts23901= 1	ts23410_g2=NA	ts23411_g2= 3	
	gross income main activity	ts23510_g2	ts23102_g1= 1	ts23901= 1	**split variables!**		ts23510_g2>ts23410_g2*
	gross income main activity, 1. split	ts23511_g2	ts23102_g1= 1	ts23901= 1	ts23510_g2=NA		
	gross income main activity, 2. split, categories up to 1,500 Euro	ts23512_g1	ts23102_g1= 1	ts23901= 1	ts23510_g2=NA	ts23511_g2= 1	
	gross income main activity, 2. split, categories between 1,500 and 3,000 Euro	ts23513_g1	ts23102_g1= 1	ts23901= 1	ts23510_g2=NA	ts23511_g2= 2	
	gross income main activity, 2. split, categories for more than 3,000 Euro	ts23514_g1	ts23102_g1= 1	ts23901= 1	ts23510_g2=NA	ts23511_g2= 3	
	child benefit included in gross income	ts23521	ts23102_g1= 1	ts23901= 1			
	number of children receiving child benefits	ts23522	ts23102_g1= 1	ts23901= 1	ts23521= 1		
	special payments	ts23531_g1	ts23102_g1= 1	ts23901= 1	ts23911= 1|2|3	ts23541_g1= 0 → 0; ts23541_g1> 0 →> 0	
	amount of special payments	ts23541_g1	ts23102_g1= 1	ts23901= 1	ts23911= 1|2|3	ts23531_g1= 0 → 0; ts23531_g1> 0 → range(ts23541_g1)	
	secondary activity	ts23102_g2	ts23102_g2= 1|2				
	episode of secondary activity	spell_g2	ts23102_g2= 1|2				
	secondary activity persists	splast_g2	ts23102_g2= 1|2				
	duration of secondary activity episode	startm_g2	ts23102_g2= 1|2				
	net income secondary activity	ts23410_g3	ts23102_g2= 1|2	**split variables!**			
	net income secondary activity, 1. split	ts23411_g3	ts23102_g2= 1|2	ts23410_g3=NA			
	net income secondary activity, 2. split, categories up to 1,500 Euro	ts23412_g2	ts23102_g2= 1|2	ts23410_g3=NA	ts23411_g3= 1		
	net income secondary activity, 2. split, categories between 1,500 and 3,000 Euro	ts23413_g2	ts23102_g2= 1|2	ts23410_g3=NA	ts23411_g3= 2		
	net income secondary activity, 2. split, categories for more than 3,000 Euro	ts23414_g2	ts23102_g2= 1|2	ts23410_g3=NA	ts23411_g3= 3		
	gross income secondary activity	ts23510_g3	ts23102_g2= 1|2	**split variables!**			ts23510_g3>ts23410_g3*

(...)

Table 2: (continued)

Module	Variable name	Variable label	Filter 1	Filter 2	Filter 3	Filter 4	Filter 5
	gross income secondary activity, 1. split	ts23511_g3	ts23102_g2= 1|2	ts23510_g3=NA			
	gross income secondary activity, 2. split, categories up to 1,500 Euro	ts23512_g2	ts23102_g2= 1|2	ts23510_g3=NA	ts23511_g3= 1		
	gross income secondary activity, 2. split, categories between 1,500 and 3,000 Euro	ts23513_g2	ts23102_g2= 1|2	ts23510_g3=NA	ts23511_g3= 2		
	gross income secondary activity, 2. split, categories for more than 3,000 Euro	ts23514_g2	ts23102_g2= 1|2	ts23510_g3=NA	ts23511_g3= 3		
	duration of employment in total	startm_g3					
	employment	emp					

Notes:
The filters indicate which subgroups of the filter question became the follow-up question.
*Indicates filters that need to be regarded for initialization, but can have empirically observed values.
Bold are the restrictions for filter questions which can be logically derived by the follow-up question.

B References

[1] Christian Aßmann, Ariane Würbach, Solange Goßmann, Ferdinand Geissler, and Anika Biedermann. *A nonparametric multiple imputation approach for multilevel filtered questionnaires*. Bamberg, 2014.

[2] Douglas Bates, Martin Maechler, Ben Bolker, and Steven Walker. *lme4: Linear mixed-effects models using Eigen and S4*. R package version 1.1-5. 2014.

[3] Hans-Peter Blossfeld. *Bildungsexpansion und Berufschancen*. Frankfurt: Campus, 1985.

[4] Hans-Peter Blossfeld, Hans Günther Roßbach, and Jutta von Maurice. "Education as a Lifelong Process: The German National Educational Panel Study (NEPS)." In: Special Issue 14 (2011).

[5] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees*. New York: Chapman and Hall, 1984.

[6] Lane F. Burgette and Jerome P. Reiter. "Multiple Imputation for Missing Data via Sequential Regression Trees." In: *American Journal of Epidemiology* (2010).

[7] Lane F. Burgette and Jerome P. Reiter. *treeMI: R Package for Multiple Imputation via Sequential Regression Trees*. R package version 0.1-3. 2010.

[8] John B. Carlin, John C. Galati, and Patrick Royston. "A new framework for managing and analyzing multiply imputed data in Stata." In: *The Stata Journal* 8.1 (2008), pp. 49–67.

[9] Jacob Cohen and Patricia Cohen. *Applied multiple regression/correlation analysis for the behavioral sciences*. L. Erlbaum Associates, 1983.

[10] Anthony Damico. "Transitioning to R: Replicating SAS, Stata, and SUDAAN Analysis Techniques in Health Policy Data." In: *The R Journal* 1/2.December (2009), pp. 37–44.

[11] James Honaker, Gary King, and Matthew Blackwell. "Amelia II: A Program for Missing Data." In: *Journal of Statistical Software* 45.7 (2011), pp. 1–47.

[12] Thomas Leopold, Marcel Raab, and Jan Skopek. *Data Manual: Starting Cohort 6 – Adult Education and Lifelong Learning*. Ed. by NEPS Data Center. Bamberg, 2011.

[13] Thomas Lumley. "Analysis of complex survey samples." In: *Journal of Statistical Software* 9.1 (2004), pp. 1–19.

[14] Thomas Lumley. *mitools: Tools for multiple imputation of missing data*. R package version 2.2. 2012.

[15] Bernd Prantner. *Visualization of imputed values using the R-package VIM*. 2011.

[16] R Core Team. *foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...* R package version 0.8-60. 2014.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2014.

[18] Trivellore Raghunathan and Irina Bondarenko. *Diagnostics for Multiple Imputations*. 2007.

[19] William Revelle. *psych: Procedures for Psychological, Psychometric, and Personality Research*. R package version 1.3.2. Northwestern University. Evanston, Illinois, 2013.

[20] Regina T. Riphahn and Oliver Serfling. "Item Non-Response on Income and Wealth Questions." In: *Empirical Economics* 30.2 (2005), pp. 521–538.

[21] Brian Ripley. *tree: Classification and regression trees*. R package version 1.0-35. 2014.

[22] Donald B. Rubin. *Multiple imputation for nonresponse in surveys*. New York: Wiley, 1987.

[23] Donald B. Rubin and Nathaniel Schenker. "Multiple Imputation for Interval Estimation from Simple Random Samples With Ignorable Nonresponse." In: *Journal of the American Statistical Association* 81.394 (1986), pp. 366–374.

[24] Deepayan Sarkar. *Lattice: Multivariate Data Visualization with R*. ISBN 978-0-387-75968-5. New York: Springer, 2008.

[25] Joseph L. Schafer. *Analyzing the NHANES III Multiply Imputed Data Set: Methods and Examples*. National Center for Health Statistics, Hyattsville Maryland, 2001.

[26] Bernhard Schimpl-Neimanns. *Schätzung des Stichprobenfehlers im Mikrozensus mit Stata – Eine Einführung mit Beispielen zum Campus File Mikrozensus 2002*. Gesis, 2009.

[27] Matthias Templ, Andreas Alfons, Alexander Kowarik, and Bernd Prantner. *VIM: Visualization and Imputation of Missing Values*. R package version 4.0.0. 2013.

[28] Stef van Buuren. *Flexible Imputation of Missing Data*. CHAPMAN & HALL/CRC, 2012.

[29] Stef van Buuren, Jaap P. L. Brand, Catharina G. M. Groothuis-Oudshoorn, and Donald B. Rubin. "Fully conditional specification in multivariate imputation." In: *Journal of Statistical Computation and Simulation* 76.12 (2006), pp. 1049–1064.

[30] Knut Wenzig. *NEPS – Daten mit DOIs referenzieren*. Rat für Sozial- und Wirtschaftsdaten and Berlin, 2012.

[31] Halbert White. "A Heteroskedastic-Consistent Covariance MAtrix Estimator and a Direct Test for Heteroskedasticity." In: *Econometrica* 48.4 (1980), pp. 817–838.

[32] Hadley Wickham. "The Split-Apply-Combine Strategy for Data Analysis." In: *Journal of Statistical Software* 40.1 (2011), pp. 1–29.

[34] Achim Zeileis. "Econometric Computing with HC and HAC Covariance Matrix Estimators." In: *Journal of Statistical Software* 11.10 (2004), pp. 1–17.