NEPS
National Educational Panel Study

Research Data

FDZ-LIfBi

Data Manual

NEPS Starting Cohort 6—Adults

Adult Education and Lifelong Learning

Scientific Use File Version 11.0.0

LIfBi
LEIBNIZ INSTITUTE FOR
EDUCATIONAL TRAJECTORIES

Leibniz Institute for Educational Trajectories (LIfBi)
Wilhelmsplatz 3, 96047 Bamberg
Director: Prof. Dr. Cordula Artelt
Executive Director of Research: Dr. Jutta von Maurice
Executive Director of Administration: N.N.
Bamberg; July 9, 2020

Research Data Documentation

The *NEPS Research Data Documentation Series* presents resources prepared to support the work with data from the National Educational Panel Study (NEPS).

Full citation of this document:
FDZ-LIfBi. (2020). *Data Manual NEPS Starting Cohort 6– Adults, Adult Education and Lifelong Learning, Scientific Use File Version 11.0.0*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

This release of Scientific Use Data from Starting Cohort 6—Adults "Adult Education and Lifelong Learning" was prepared by the staff of the Research Data Center at Leibniz Institute for Educational Trajectories (Forschungsdatenzentrum, FDZ-LIfBi). It represents a major collaborative effort. *The contribution of the following persons is gratefully acknowledged:*

Eva Akins
Dietmar Angerer
Nadine Bachbauer
Pia Bechtloff
Daniel Bela
Hannes Götz
Daniel Fuß
Lydia Kleine
Tobias Koberg
Gregor Lampel
Sven Pelz
Benno Schönberger
Mihaela Tudose
Katja Vogel
Clara Wolf

For their support in writing this manual, special thanks go to:
Ralf Künster (WZB Berlin)

We also appreciate the work of the former colleagues at the Research Data Center:

Simon Dickopf, Thomas Leopold, Manuel Munz, Sebastian Pink, Marcel Raab, Jan Skopek, Knut Wenzig, Markus Zielonka

Leibniz Institute for Educational Trajectories (LIfBi)
Research Data Center (FDZ)
Wilhelmsplatz 3
96047 Bamberg, Germany

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

Contents

1 Introduction

1.1 About this manual

This manual is intended to facilitate your work with data of NEPS Starting Cohort 6—Adults (NEPS SC6). It serves both as a first guide for getting started with the complex data and as a reference book. The primary emphasis is on practical aspects such as sample development, data structure, and variable merging. The manual is neither complete nor exhaustive, but several links to other resources are provided in the respective paragraphs.

The first chapter refers to further documentation material, requirements for data access, instructions for data citation, some general rules and recommendations, and selected user services. In the second chapter, the fundamental objectives of Starting Cohort 6 and its sampling strategy are briefly introduced. The main part of this chapter is devoted to the sample development across the waves including field times, realized case numbers, survey modes, and the measurement of competency domains. The principles of Scientific Use File data-editing processes as well as conventions for naming the data files and variables are explained in the third chapter, supplemented by missing value definitions and an overview of additionally generated variables. The fourth chapter focuses on the data structure with information about data types, identifiers, and short portraits of all available datasets in the Scientific Use File. These short portraits include recommendations on how to use the dataset as well as syntax examples for merging variables of this dataset with variables from other files.

According to the cumulative release strategy—each new Scientific Use File contains the data of all previous survey waves plus the data of the currently prepared wave(s)—this manual will be regularly updated and revised. While the given information remain valid over time, at least the sample development has to be continuously complemented. In other words, the latest published manual replaces the previous ones. All relevant adjustments and extensions in future releases of this manual will be listed in a separate appendix.

1.2 Further documentation

The data manual cannot cover all issues in detail. Hence, a bunch of supplementary reports and other materials with background information on data preparation, survey instruments, competence tests, and field work is offered (see Figure 1). This frequently updated and enhanced data documentation can be downloaded from our website at:

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

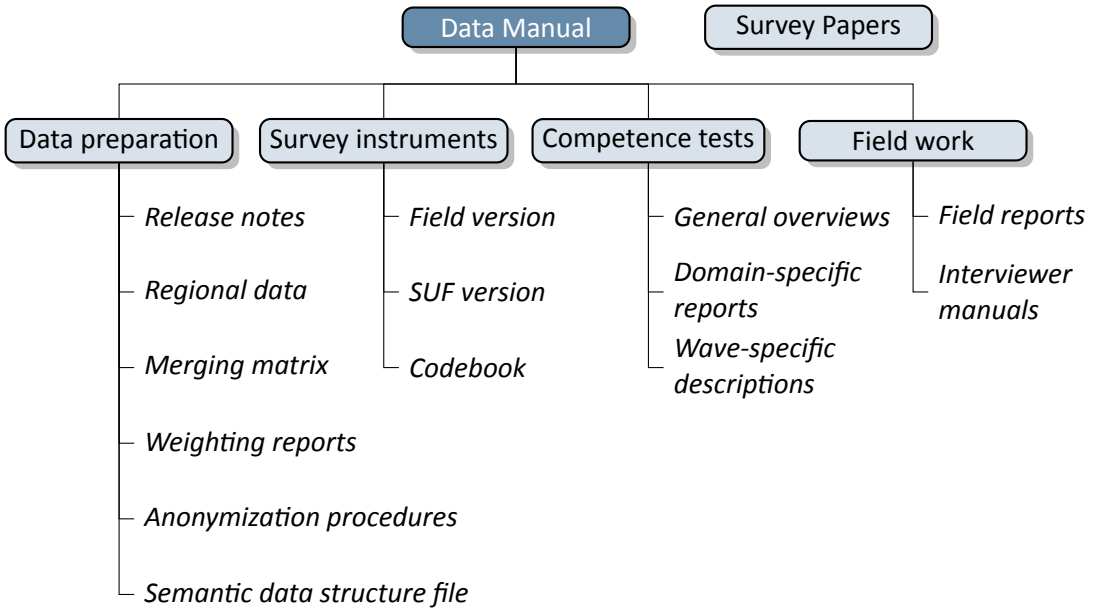

Figure 1: NEPS supplementary data documentation

Release notes All Scientific Use Files are accompanied by release notes that log changes in the data compared to prior versions and list bugs eliminated or at least known. For the latter, short syntax corrections are usually given. Please consult these notes when working with the data. See also section A.2 for a depiction of the current notes.

Regional data Fine-grained regional indicators from a commercial provider (microm) are available in our On-site environment. The report describes the regional levels covered by these indicators, their content, and how to merge them to the survey data.

Merging matrix This matrix provides an overview of how to link information from different datasets, taking into account the relevant identifier variables.

Weighting reports These reports entail information regarding the design principles of the sampling process and the creation of weights.

Anonymization procedures The document describes the anonymization measures carried out and provides an overview regarding the opportunity to access sensitive data.

Semantic data structure file This data package corresponds to the Scientific Use File but does not contain any observations (*purged datasets*). It provides all metadata including variable names, labels and answering scheme options to be used for exploring the data structure and for preparing analyses.

Survey instruments For each wave, the survey instruments are offered in the form of Scientific Use File (SUF) and field versions. While the field versions consist of the originally deployed instruments (in German only), the SUF versions are enriched by additional information

such as variable names and value labels used in the Scientific Use File. *Please note, that the competence test booklets are not publicly available*.

Codebook The codebook lists all variables and their corresponding labels plus the basic frequencies by waves in concordance with the datasets in the Scientific Use File.

Competence tests Information about competence testing is provided in various documentations, including general overviews and wave-specific descriptions; also, for each domain there is usually a brief description of the construct with sample items, a description of the data, and of the psychometric properties of the test.

Field reports The field reports document the overall data-collection process conducted by the survey institute. This information about survey preparation, interviewer deployment, respondent tracking, initial contacts, incentives, and sample realization is available in German only.

Interviewer manuals The interviewer manuals are a collection of instructions for the interviewers. In particular, they exemplify the interview process as well as the content of each of the questionnaire modules. They are available in German only.

NEPS Survey Papers Finally, there is a series of NEPS Survey Papers that address several topics of more general interest. These papers are listed for download on our website at:

→ www.neps-data.de > Data Center > Publications > NEPS Survey Papers

Additional documentation material might be available for specific cohorts and/or waves. Please visit the website above for further details.

1.3 Data release strategy

NEPS data are published in the form of Scientific Use Files. They are provided free of charge to the scientific community. Each Scientific Use File consists of multiple datasets, forming a complex data structure with cross-sectional, panel and episode or spell information (see section 4). The release of NEPS Scientific Use Files follows a cumulative strategy, i. e., the latest data release replaces all former data releases. Hence, we recommend to use the most current release of a Scientific Use File.

File Format

All Scientific Use Files are disseminated in Stata and SPSS format with bilingual variable labels and value labels in German and English. In the SPSS format, there are separate data files for both languages. Data stored in Stata format contain both languages within one file; the switch is induced by the following Stata command:

```
label language [de/en]
```

Due to the change of encoding to "Unicode" in Stata14 and the fact that older Stata versions are not able to open such data files, the NEPS Scientific Use Files contain two Stata formats, namely Stata14 and Stata12.

Versioning and Digital Object Identifier

Every time a new Scientific Use File is released, the data files existing up to now are either extended, usually by information from a new survey wave, or updated with changes due to larger or smaller corrections. The three digits of the version number inform about the number of waves integrated in the specific Scientific Use File, the frequency of major updates, and the frequency of minor updates. The version number is part of all relevant designations: that of the Scientific Use File, its data files (see Table 3), and the respective Digitial Object Identifier.

Every release of a NEPS Scientific Use File is registered at da|ra and clearly labeled with a unique Digital Object Identifier (DOI, cf. Wenzig, 2012). This DOI has two main functions. On the one hand, it enables researchers to cite the utilized NEPS data in an easy and precise way (see section 1.5). This in turn is a basic precondition for any replication analysis. On the other hand, the DOI directs to a landing page with further information about the Scientific Use File and the data access options. The DOI of the current release is doi:10.5157/NEPS:SC6:11.0.0. Other releases of Scientific Use Files for Starting Cohort 6 can be accessed by substituting the version number at the end of the DOI and the URL respectively (see Table 1).

Table 1: Release history of SUF in Starting Cohort 6

SUF Version	DOI	Date of release
11.0.0 (current)	doi:10.5157/NEPS:SC6:11.0.0	**2020-07-10**
10.0.1	doi:10.5157/NEPS:SC6:10.0.1	2019-10-24
10.0.0	doi:10.5157/NEPS:SC6:10.0.0	2019-09-02
9.0.0	doi:10.5157/NEPS:SC6:9.0.0	2018-10-31
8.0.0	doi:10.5157/NEPS:SC6:8.0.0	2017-10-13
7.0.0	doi:10.5157/NEPS:SC6:7.0.0	2016-12-22
6.0.1	doi:10.5157/NEPS:SC6:6.0.1	2016-07-13
6.0.0	doi:10.5157/NEPS:SC6:6.0.0	2016-05-13
5.1.0	doi:10.5157/NEPS:SC6:5.1.0	2015-07-16
5.0.0	doi:10.5157/NEPS:SC6:5.0.0	2015-03-27
3.0.1	doi:10.5157/NEPS:SC6:3.1.0	2013-08-06
3.0.0	doi:10.5157/NEPS:SC6:3.0.0	2013-06-06
1.0.0	doi:10.5157/NEPS:SC6:1.0.0	2011-12-22

1.4 Data access

Access to the NEPS data is free of charge but limited to the purpose of research and members of the scientific community. Granting the right to obtain the data requires the conclusion of a Data Use Agreement. The existence of a valid Data Use Agreement entitles to work with all NEPS Scientific Use Files, i. e., the full data portfolio is at the disposal of the data recipient and all further persons involved in the agreement.

Application for data access

- Fill in the online form for a NEPS Data Use Agreement either in German or in English. Enter a title, the duration, and a short description of the intended research project. Make sure that all project participants with NEPS data access are specified in the form and that these persons have signed the agreement. Submit one copy of the complete agreement by e-mail, fax, or post. Further instructions and the relevant forms are provided on our website at:

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

- After approval by the Research Data Center, the registered NEPS data user receives a user name and a password to log in to our website.
- The basic Data Use Agreement permits the download of all available Scientific Use Files from our website at:

  → www.neps-data.de > Data Center > Data and Documentation > NEPS Data Portfolio

- There are two other modes of access to the NEPS data (see below); each demanding a supplemental agreement in addition to the basic Data Use Agreement.
- Another form is provided to state changes of the Data Use Agreement regarding further project participants or a prolonged project duration.

Modes of data access

Three modes of accessing the NEPS Scientific Use Files are available. They are designed to support the full range of researchers' interests and maximize data utility while complying with national and international standards of confidentiality protection. Each modus corresponds to a data version that is different with regard to the accessibility of sensitive information as the three versions of a Scientific Use File vary according to their level of data anonymization.

- *Download* from the website = highest level of anonymization
- *RemoteNEPS* as browser-based remote desktop access = medium level of anonymization
- *On-site* access at secure working stations at LIfBi = lowest level of anonymization

While working with RemoteNEPS requires a biometrical authentication and Internet access, the On-site use of NEPS data necessitates a guest stay at LIfBi in Bamberg. More details about the three access modes and their implications for application and utilization are given on our website at:

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

Sensitive information

The download version of a Scientific Use File contains the least amount of information. Indicators of a certain sensitivity are modified in the download data, such as aggregated categories for countries of citizenship or languages of origin. A few datasets and variables are exclusively accessible in the On-site version, e.g. the fine-grained regional indicators or open text entries. For a full picture of the availability of sensitive information, please refer to the overview on our website at:

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

The hierarchical concept of data dissemination translates into an onion-shaped model of datasets. The most sensitive on-site level represents the outer layer with the remote and download levels being subsets of these data. That is, any data contained within a less sensitive level are also included in the higher level(s). A detailed list of variables offered at the different levels together with notes on the generation of the three data versions can be found for each release of a Scientific Use File in the respective report on ”Anonymization Procedures”.

1.5 Publications with NEPS data

Referencing the use of data from the National Educational Panel Study (NEPS) is essential for a good scientific practice as well as for revealing the scientific value of this study. The following citation rules apply to all publications based on NEPS data of Starting Cohort 6.

It is obligatory to acknowledge the NEPS study in general and to indicate the utilized data version by including a phrase like this in your publication:

> This paper uses data from the National Educational Panel Study (NEPS): Starting Cohort 6—Adults, doi:10.5157/NEPS:SC6:11.0.0. From 2008 to 2013, NEPS data was collected as part of the Framework Program for the Promotion of Empirical Educational Research funded by the German Federal Ministry of Education and Research (BMBF). As of 2014, NEPS is carried out by the Leibniz Institute for Educational Trajectories (LIfBi) at the University of Bamberg in cooperation with a nationwide network.

Please also add these bibliographic details to your list of references:

> 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*.

Authors of any kind of publications based on NEPS data are requested to notify the Research Data Center about their articles and to provide an electronic version or a special print or a copy. All reported publications are listed in the NEPS Bibliography on our website at:

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

Citing documentation

To refer to any of the documentation material published in the *NEPS Research Data Documentation Series* (e. g. this manual), please make use of the following citation templates:

> FDZ-LIfBi. (2020). *Data Manual NEPS Starting Cohort 6– Adults, Adult Education and Lifelong Learning, Scientific Use File Version 11.0.0*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

Or another example:

> Schönberger, K., & Koberg, T. (2017). *Regional Data: Microm*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

If no author is given, please take a universal *NEPS* instead:

> NEPS (Ed.). (2020). *Starting Cohort 6: Adults (SC6), Wave 11, Questionnaires (SUF Version 11.0.0)*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

If a document has not been published in this series, please refer to the author and the title as in the following citation of a field report by one of our survey institutes:

> Steinwede, J., & Aust, F. (2012). *Methodenbericht, NEPS Startkohorte 5 – CATI-Haupterhebung Herbst 2010, B52*. Bonn, Germany: infas

1.6 Rules and recommendations

Working with NEPS data is bound to a couple of rules that are codified in the Data Use Agreement. Each data user has to confirm these rules by his or her signature. The already mentioned obligation to cite the study and to indicate any kind of publication resulting from the use of NEPS data (see section 1.5) are just two examples. The major part of rules refers to issues of data privacy and the requirements of a careful data handling.

Rules

- *Avoidance of re-identification:* Any action aimed at and suitable for re-identifying persons, households, or institutions is strictly forbidden. This also includes the combination of NEPS data with other data that allow for a re-identification of persons. In case of any accidental re-identification, the Research Data Center has to be informed immediately and all individual data gained therefrom have to be kept secret.
- *Avoidance of data disclosure:* NEPS data are exclusively provided on the basis of a valid Data Use Agreement—for a defined purpose (research project) and to a defined group of persons (data recipient and further project members that are involved in the contract). Any use for commercial or other economic purposes is not permitted just as any transfer of the data to third parties. Please handle the provided NEPS data with strict confidentiality!
- *Regulations on using the Federal State label:* For NEPS data collected in connection with schools or higher education institutions it is not allowed to use Federal-State-related information directly or indirectly contained in the data for analyses aiming at direct comparisons of the German Federal States (*Bundesländer*), or aiming at direct conclusions to be drawn about a Federal State, or aiming at a reconstruction of the concrete Federal State affiliation of persons, households, and institutions. Any kind of ranking between the Federal States based on NEPS data is prohibited (see **??**).

Please note that violation of these rules may lead to severe penalties as stated in the NEPS Data Use Agreement. If there is any doubt or question regarding the given regulations, please contact the Research Data Center (see section 1.8). The same applies in case of encountering any deficiencies in data quality or any security leaks with regard to NEPS data protection and data security.

Recommendations

In addition to the binding rules, there are some recommendations for the use of NEPS data:

- *As a matter of course:* Always be critical when working with empirical data! Although a big effort is being made to ensure the integrity of the provided data we cannot guarantee absolute correctness. Notices on problems or errors in the data are welcome at any time at the Research Data Center.
- *Enhanced understanding of the data:* Consult the documentation and survey instruments! The analysis of complex data necessitates a precise idea of how the information were collected and edited. All relevant material is available online (see section 1.2).
- *Facilitated handling of the data:* Utilize the tools that are offered! Several user services are provided to support NEPS data analyses—reaching from specific Stata commands (e. g., for an easy and adequate recoding of missing values) to a meta search engine (e. g., for an interactive exploration of all instruments) to a discussion forum (e. g., for the clarification of questions). These tools are also available online, see section 1.7 for more details.

1.7 User services

In addition to a comprehensive data documentation there are several user services to support researchers working with NEPS data. First and foremost, the Research Data Center maintains a regularly updated and enhanced website with detailed information on all available Scientific Use Files, a complete list of NEPS projects, a NEPS bibliography, a reference to NEPS events, and a NEPS newsletter. All subsequently introduced services and tools can also be reached via this website:

→ www.neps-data.de > NEPS

NEPSforum

The *NEPSforum* is an open online discussion platform for experienced users as well as for persons who are searching for NEPS related information. It offers the opportunity to exchange with NEPS staff members and with other researchers in a transparent dialogue. That way, the forum will become a rich archive of knowledge with practical solutions for numerous problems and questions. We highly encourage you to browse the forum first when struggling with NEPS issues or when help is needed with specific data matters. If there is no available solution, please take the opportunity to share your question by posting it to the forum. Active participation requires no more than a one-time registration. The entire NEPS user community will benefit from a broad participation. You can find the *NEPSforum* at:

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

NEPSplorer

The *NEPSplorer* facilitates an interactive and quick full text search through all instruments of released NEPS surveys, with the exception of competence tests. The tool is particularly suitable for getting a first idea of the availability of constructs, items, and variables in the datasets. It is currently based on keyword search with several filtering options, but a hierarchical construct search will be added soon. The *NEPSplorer* offers some helpful functions such as displaying univariate statistics, listing relevant metadata, and enabling registered users to create their own personal watch list of interesting items. As a web application—a mobile version aligned for smartphone usage is also available—the *NEPSplorer* relies on the most up-to-date information; any correction in the metadata is thus instantly visible. Start the tool here:

→ www.neps-data.de > Data Center > Overview and Assistance > NEPSplorer

NEPStools

NEPStools is a free to use collection of Stata commands that is created and supplied by the Research Data Center. The package includes some programs (“ado files”) that make NEPS data handling easier. As an example, the `nepsmiss` command automatically recodes all of the numeric missing values (-97, -98, etc.) into Stata’s “Extended Missings” (.a, .b, etc.) with correctly recoded value labels. Another example ist the `infoquery` command that displays additional attributes of the variable such as the question text and the initial variable name in the instrument. The *NEPStools* set can be easily installed from our repository through Stata’s built-in installation mechanism:

```
net install nepstools, from(http://nocrypt.neps-data.de/stata)
```

A description of the programs and further information are given on the website at:

→ www.neps-data.de > Data Center > Overview and Assistance > Stata Tools

User trainings

The Research Data Center offers a series of regular user training courses at the Leibniz Institute for Educational Trajectories in Bamberg. The standard 2-day courses are free of charge. On the first day, there is a general introduction to the design of the NEPS study, the structure of NEPS Scientific Use Files, the terms and conditions of data access and data usage, and the handling of documentation materials. The second day is more focused on data of a certain starting cohort and on selected methodological and/or theoretical concepts. Both parts come along with guided hands-on sessions. A crucial aspect of all user trainings is the sensitization of participants to issues of privacy and data protection. In this context, participation is obligatory for those who want to enroll in the biometric authentication system in order to gain access to the NEPS remote or On-site environment. A schedule of all training dates together with information on how to register for a course can be retrieved from our website at:

→ www.neps-data.de > Data Center > User Training

1.8 Contacting the Research Data Center

The Research Data Center at the Leibniz Institute for Educational Trajectories (Forschungsdatenzentrum, FDZ-LIfBi) accounts for large parts of the NEPS data preparation and documentation process, for the data dissemination, and for the user support including individual advice. We welcome your feedback at any time to further improve our products and services. This particularly applies to this manual as the guiding document to facilitate your work with NEPS data of Starting Cohort 6.

Please contact us with your questions, comments, requests, and suggestions:

E-mail: fdz@lifbi.de
Web: → www.neps-data.de > Data Center > Contact Data Center
Phone: +49 951 863 3511

2 Sampling and Survey Overview

2.1 Adult education and lifelong learning

As part of this NEPS substudy, data on educational and professional careers as well as on competence acquisition across adult life courses are being collected.

In order to be able to study adult education, the entire spectrum of educational activities and learning processes (formal, nonformal, and informal), and decisions resulting in their participation, as well as the respondents' previous life course (especially the course of education and occupation, relationships, and children) are recorded in detail. Similar to the lack of knowledge concerning adult education in Germany, very little information is available on competencies and their changes after school. This is why this substudy collects data on competencies in reading, mathematics, sciences, and ICT literacy as well as data on noncognitive skills (such as personality, motivation, and social skills). The data should enable researchers to:

- trace the aquisition of education across the adult life course and to follow the course of education and employment of younger cohorts after their job entry;
- study why individuals decide to participate or not to participate in formal or nonformal learning activities after their initial vocational training;
- describe the competencies of different groups of adults in Germany and to explain competence development in adulthood as well as the importance of the employment situation in this context;
- analyze the impact of specific educational contexts in adult life, especially that of the employment situation and family constellation, on educational choices and participation in further training;
- estimate the returns of formal qualifications, competencies, and professional experience in terms of wages, occupational careers, and in other areas of life (e.g., well-being or volunteer work);
- generate empirical results on competencies of migrants, their resources, their participation in and returns from further training;
- identify opportunities and obstacles for learning processes and education in later adult life.

The field time of the adult survey already started in 2007, that is, prior to the foundation of the National Educational Panel Study. The adult survey 2007/08 was conducted by the Institute for Employment Research (IAB) under the name of *Working and Learning in a Changing World* (ALWA). After that, the data collection of the adult survey continued under the umbrella of the NEPS from November 2009 to June 2010 (see section 2.2 for details).

2.2 Sampling strategy

The target population of respondents in Starting Cohort 6 comprises all persons born between 1944 and 1986 who live in private households in Germany, irrespective of the language they speak, their nationality or their employment status. Persons living in shared facilities (old people's homes, prisons, etc.) are excluded. The sample is made up of four subsamples which, taken together, provide a representative picture of the adult population in Germany:

- **ALWA**: The data of the first wave of the Scientific Use File come from the survey *Working and Learning in a Changing World* (Arbeiten und Lernen im Wandel, ALWA, see Antoni et al., 2011) conducted in 2007 by the Institute for Employment Research (IAB). The ALWA subsample includes all respondents of this survey from the birth cohorts 1956 to 1986 who agreed to participate in a panel study. These respondents were transferred to the actual NEPS study.
- **Refreshment 2009**: With the start of the actual NEPS survey in 2009, which corresponds to the second wave in the Scientific Use File, the initial sample became refreshed with additionally sampled persons of the birth cohorts 1956 to 1986.
- **Enhancement 2009**: Parallel to this first refreshment, the sample was also enhanced to include persons born between 1944 and 1955.
- **Refreshment 2011**: A second sample refreshment took place two years later in the 2011 survey, which corresponds to the fourth wave in the Scientific Use File. This refreshment covers persons of the entire age spectrum of the Starting Cohort 6 sample, i.e. the birth cohorts 1944 to 1986.

The individual subsamples were drawn in 2007, 2009 and 2011 based on a two-stage selection process with municipalities (Gemeinden) as primary sampling units (PSU) and addresses of target persons as secondary sampling units (SSU). The selection of municipalities at the first stage was made only once in the context of the ALWA sampling. All later samplings refer to these municipalities, that is the enhancement subsample and the refreshment subsamples were drawn from within the same communities as the ALWA subsample.

- **Selection of municipalities (PSU)**: On the basis of population data provided by the German Federal Statistical Office and the statistical offices of the German Laender, all German communities were initially stratified according to Federal States, administrative districts, and degree of urbanization (BIK categorization). Within each stratum, municipalities were then randomly selected with a probability proportional to the extrapolated size of the target resident population. In the end, 281 sample points were drawn representing 250 municipalities. Due to the proportional sampling design, larger cities are included in the sample more than once, i.e. they are represented by two or more sample points.
- **Selection of addresses (SSU)**: For each selected sample point, an equal number of personal addresses of the target population was then drawn from the registers of the residents' registration offices. The selection was again made at random with a randomly chosen address as the starting point and a systematic inclusion of further addresses at a given interval.

For the additional subsamples in 2009 respective 2011, the number of cooperating municipalities that provided addresses decreased from 250 to 240 respective 242 (corresponding to 271 respective 273 sample points). In addition, the target population for the enhancement subsample 2009 and the refreshment subsample 2011 was adjusted to also include persons born between 1944 and 1955.

For each sample point, 152 addresses were selected for the ALWA subsample, 24 addresses for the refreshment subsample of 2009, 43 addresses for the enhancement subsample of 2009, and 63 addresses for the refreshment sample of 2011. The resulting gross samples consist of 22,656 individuals (ALWA; of the total of 42,712 addresses, only persons with identifiable telephone numbers were considered for field work), 6,547 individuals (Refreshment 2009), 11,465 individuals (Enhancement 2009), and 17,111 individuals (Refreshment 2011). It should be noted that 8,997 persons who participated in the first ALWA survey, who agreed to be contacted again, and who belonged to the birth cohorts 1956 to 1986 have been integrated into the NEPS gross sample for the second wave.

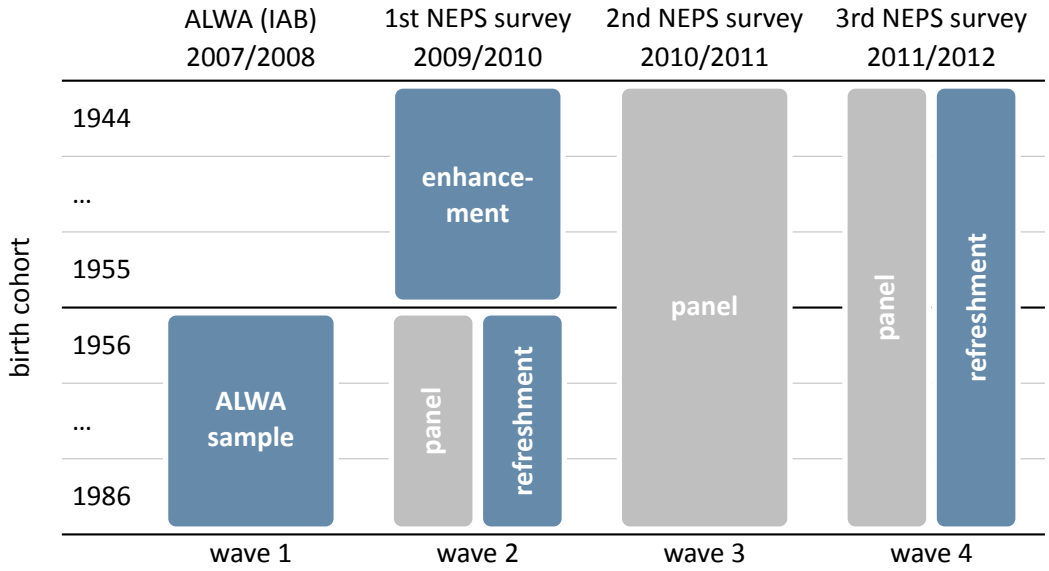

Figure 2: Longitudinal sampling design of Starting Cohort 6

The sampling design and its consequences for the derivation of sampling weights are described in Hammon, Zinn, Aßmann, and Würbach, 2016. Detailed remarks on the recruiting process are given in the NEPS field reports of survey waves 2 and 4 (in German only). All documents are available on our website at:

→ www.neps-data.de > Data Center > Data and Documenation
> Starting Cohort Adults > Documentation

2.3 Competence measures

The collection and provision of data on the development of competencies and skills throughout the life course is a key element of the German National Educational Study (NEPS). Competence measurements are carried out across different waves in all NEPS starting cohorts covering domain-general and domain-specific cognitive competencies as well as metacompetencies and stage-specific competencies.

Data from the competence tests pass through an editing process before they get integrated into the Scientific Use File. This data preparation enables users to work with scored items and test scores such as the sum or mean of correct answers. Detailed descriptions on how these scores were estimated can be found in separate reports for the respective competence domains (see section 1.2). The scores are compiled in a dataset named `xTargetCompetencies`. This dataset is structured in the so-called wide format, that is, all responses of a single respondent are represented in one row of the data matrix. As a consequence, variable names for competence scores follow a specific nomenclature. It not only allows for the identification of the respective domain, the target group, the testing modus, and the kind of scoring, but also informs about the repeated administration of a test item in a different wave or starting cohort (see section 3.2.2).

The next table shows the schedule of competence measures in Starting Cohort 6 with domains by waves including test modus. The overview contains released data as well as data that is not yet published.

Table 2: Schedule of competence measures. P = Paper-Based Test (proctored), C = Computer-Based Test (proctored)

		2010/11 **Wave 3** (24-67 y.)[2]	2012/13 **Wave 5** (26-69 y.)[3]	2014/15 **Wave 7** (28-71 y.)	2016/17 **Wave 9** (30-73 y.)[4]
Domain-General Competencies					
DGCF: Cognitive Basic Skills	dg	—	—	C	—
Domain-Specific Competencies					
Reading Competence[1]	re	P	P	—	C
Reading Speed	rs	P	P	—	—
Vocabulary: Listening Comprehension at Word Level[1]	vo	—	—	C	—
Mathematical Competence[1]	ma	P	—	—	C
Scientific Competence[1]	sc	—	P	—	—
Metacompetencies					
ICT Literacy[1]	ic	—	P	—	—

[1] Subsequent to the respective competence test the target persons had to assess their own test performance (Procedural Metacognition, mp).
[2] Wave 3: Randomized allocation of reading and mathematics competence tests to split sample (50% with three domains: re, rs, ma / 50% with two domains: rs, ma or rs, re).
[3] Wave 5: The first-surveyed target persons of the refreshment sample were tested in their reading competencies (re, rs); the target persons of the initial sample were tested in their scientific and ICT literacy competencies (sc, ic).
[4] Wave 9: The target persons of the refreshment sample were tested in their reading competencies (re) only, while the target persons of the initial sample were tested in their reading and mathematics competencies (re, ma).

2.4 Survey overview and sample development

This section informs about the progress of the Starting Cohort 6 sample. For each survey wave included in the current Scientific Use File there is a short characterization in terms of field time, number of realized cases, relevant subsamples and domains of competence testing (if appropriate), survey modus, and the institution(s) responsible for collecting the data. A more detailed insight into all relevant field work issues is provided by the *Field Reports* of the survey institutes, which are available on the website (in German only) as part of the data documentation for each (sub-)study:

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

Figure 3 starts with an overview illustrating the panel progress of Starting Cohort 6 in terms of field times and survey modes from wave 1 to 11.

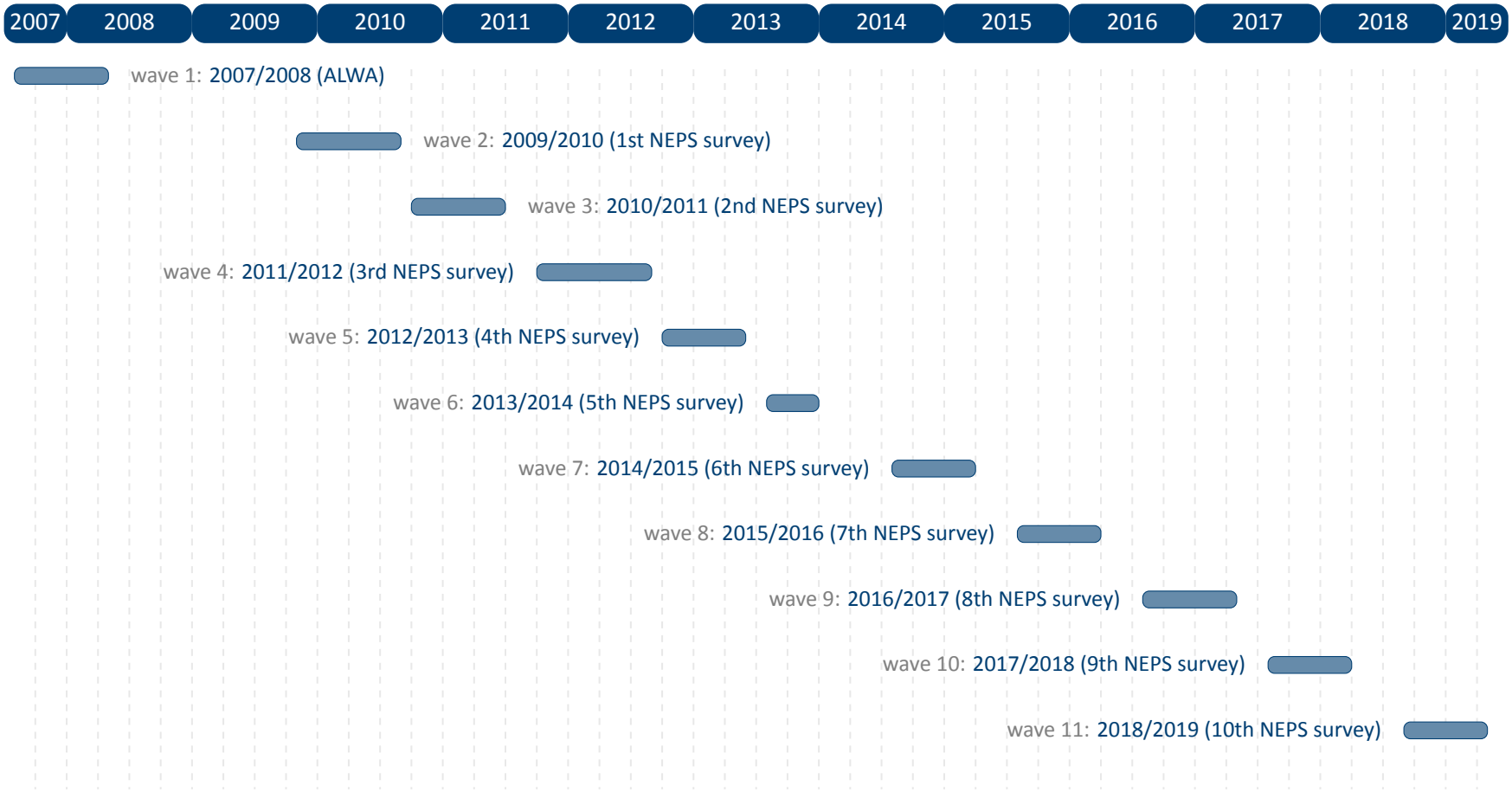

Figure 3: Survey progress of Starting Cohort 6 (waves 1 to 11)

2.4.1 Wave 1: 2007/2008 (ALWA)

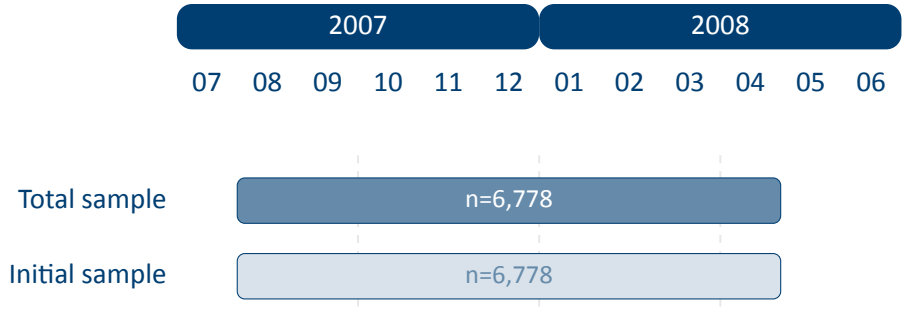

Figure 4: Field times and realized case numbers in wave 1

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Sample from the population register (with the selection stages municipalities and individuals); random selection of individuals from the resident population in Germany, independent of employment status, nationality and German language skills (see section 2.2)

- **Mode of survey** Computer-assisted telephone interviews (CATI)
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.2 Wave 2: 2009/2010 (1st NEPS survey)

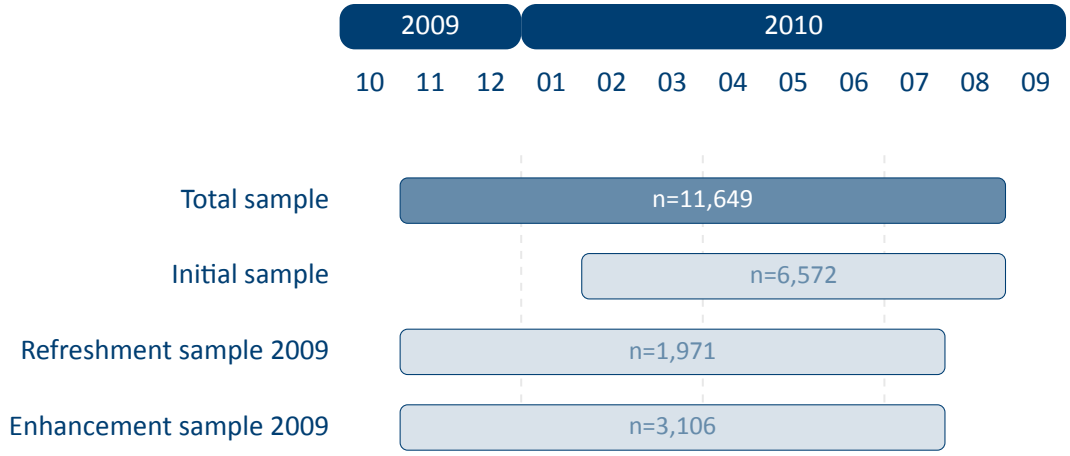

Figure 5: Field times and realized case numbers in wave 2

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Sample from the population register, corresponding to the ALWA sampling strategy (see section 2.2)

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Sample from the population register, corresponding to the ALWA sampling strategy, but focussed on the birth cohorts 1944 to 1955 (see section 2.2)

- **Mode of survey** Computer-assisted telephone interviews (CATI); computer-assisted personal interviews (CAPI) for those who could not be reached by telephone or insisted on a personal interview

- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.3 Wave 3: 2010/2011 (2nd NEPS survey)

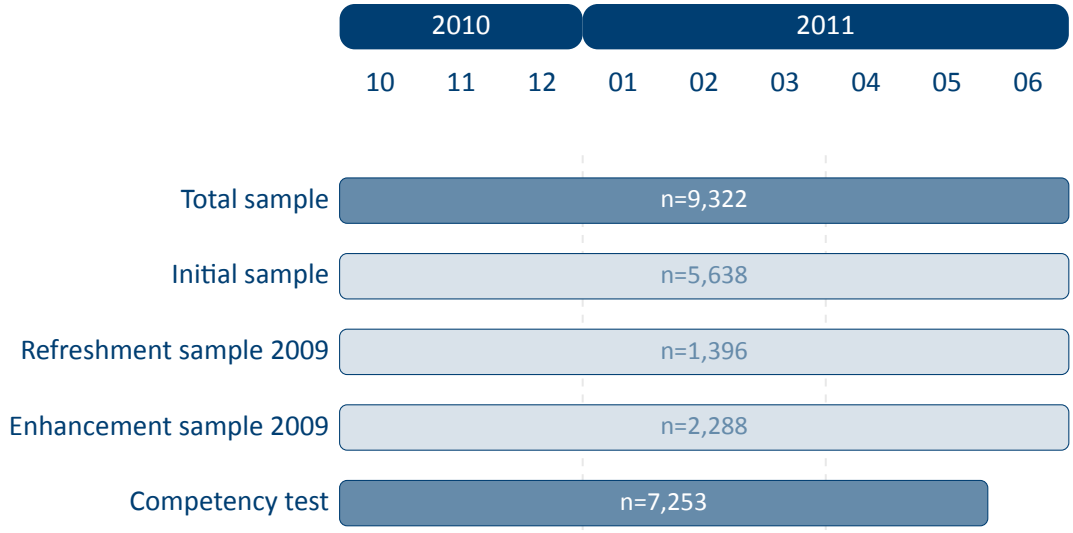

Figure 6: Field times and realized case numbers in wave 3

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted personal interviews (CAPI) including paper-based competency tests (PAPI); computer-assisted telephone interviews (CATI) for those who could not be interviewed in person or insisted on a telephone interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.4 Wave 4: 2011/2012 (3rd NEPS survey)

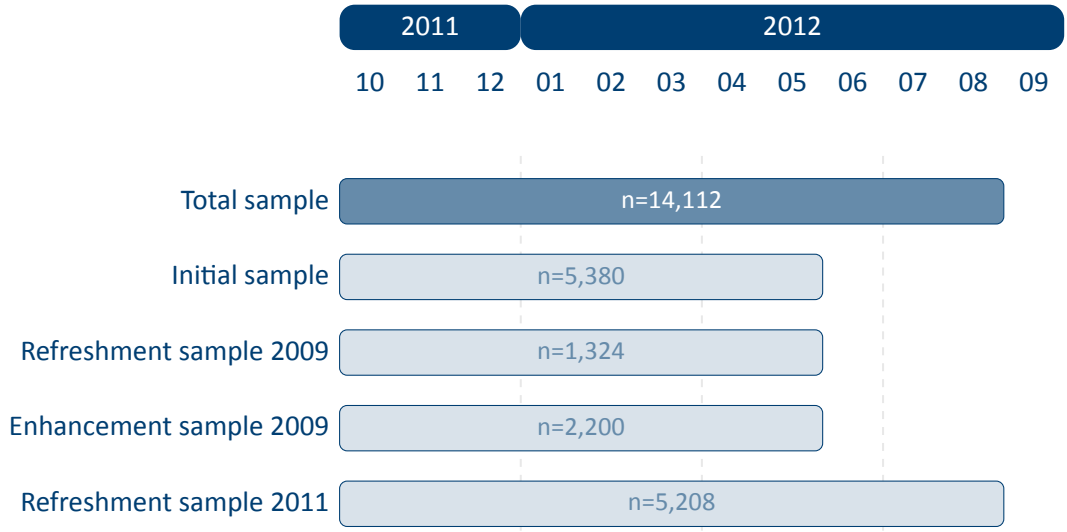

Figure 7: Field times and realized case numbers in wave 4

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Sample from the population register, corresponding to the ALWA sampling strategy, including all birth cohorts from 1944 to 1986 (see section 2.2)

- **Mode of survey** Computer-assisted telephone interviews (CATI); computer-assisted personal interviews (CAPI) for those who could not be reached by telephone or insisted on a personal interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.5 Wave 5: 2012/2013 (4th NEPS survey)

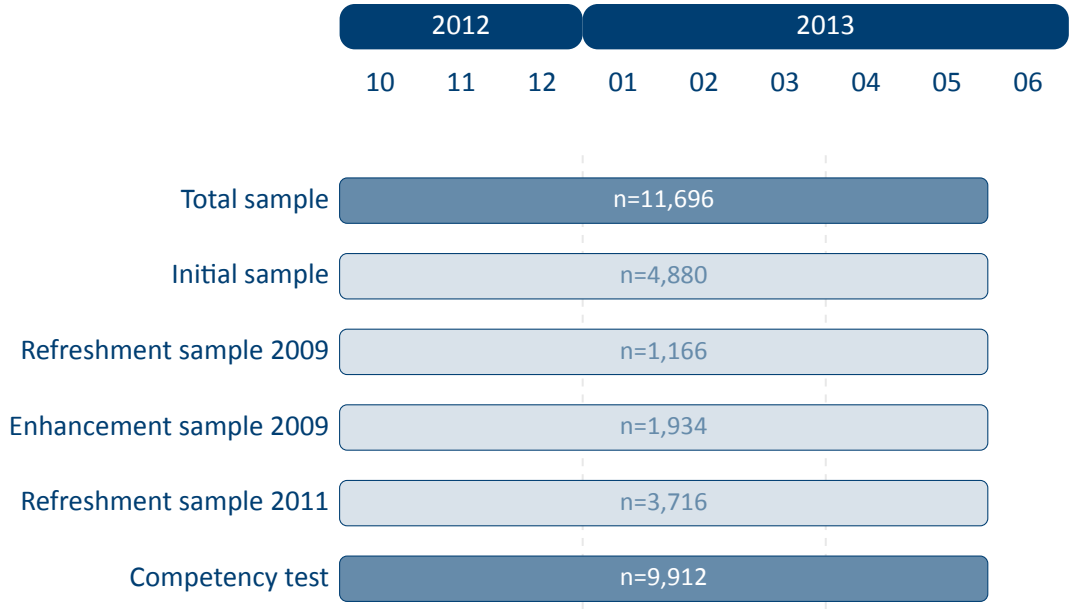

Figure 8: Field times and realized case numbers in wave 5

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted personal interviews (CAPI) including paper-based competency tests (PAPI); computer-assisted telephone interviews (CATI) for those who could not be interviewed in person or insisted on a telephone interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.6 Wave 6: 2013/2014 (5th NEPS survey)

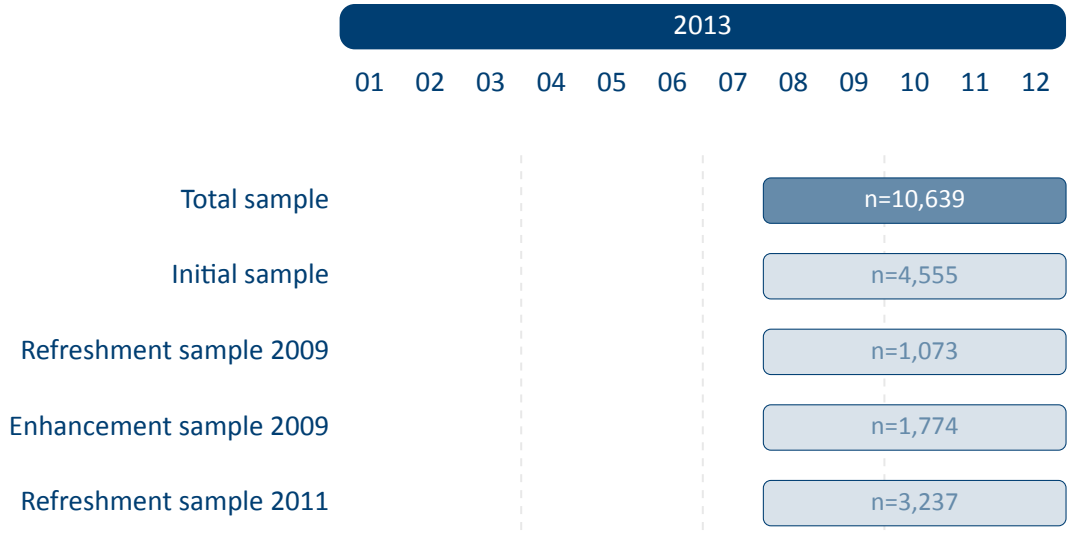

Figure 9: Field times and realized case numbers in wave 6

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted telephone interviews (CATI); computer-assisted personal interviews (CAPI) for those who could not be reached by telephone or insisted on a personal interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.7 Wave 7: 2014/2015 (6th NEPS survey)

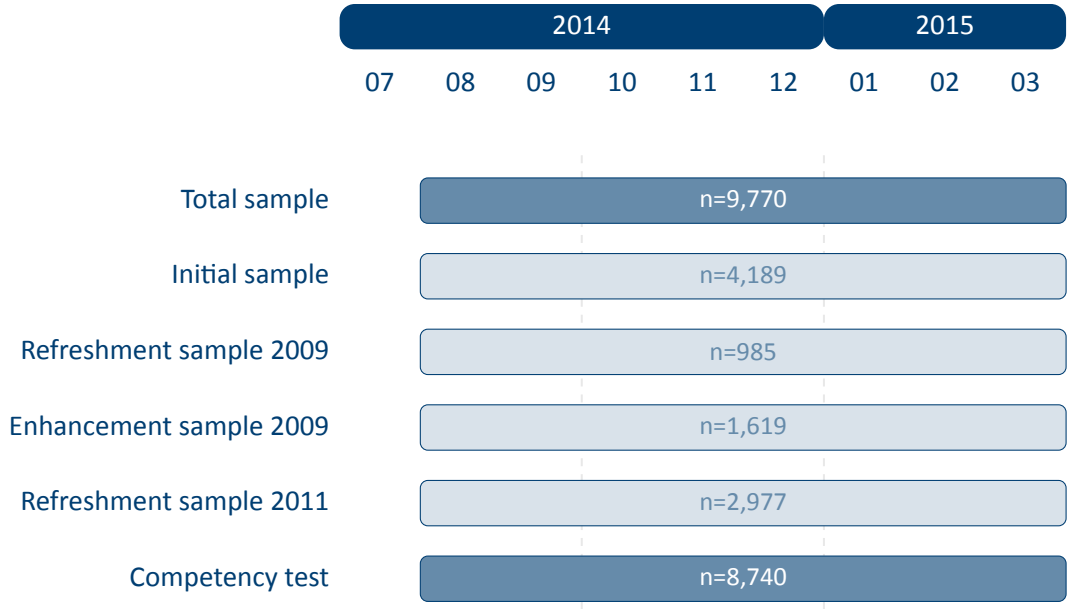

Figure 10: Field times and realized case numbers in wave 7

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted personal interviews (CAPI) including computer-based competency tests (CBA); computer-assisted telephone interviews (CATI) for those who could not be interviewed in person or insisted on a telephone interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.8 Wave 8: 2015/2016 (7th NEPS survey)

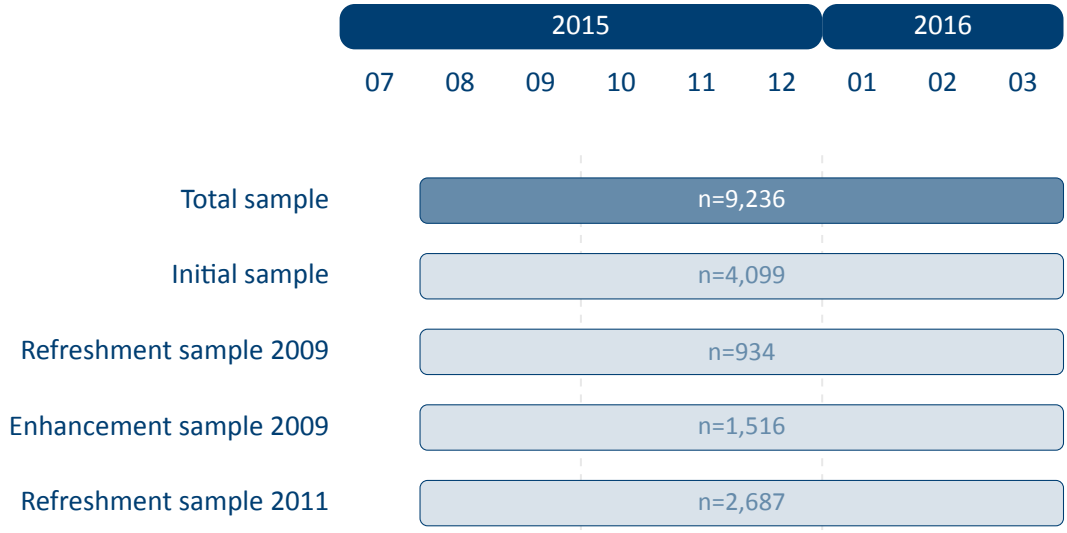

Figure 11: Field times and realized case numbers in wave 8

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted telephone interviews (CATI); computer-assisted personal interviews (CAPI) for those who could not be reached by telephone or insisted on a personal interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.9 Wave 9: 2016/2017 (8th NEPS survey)

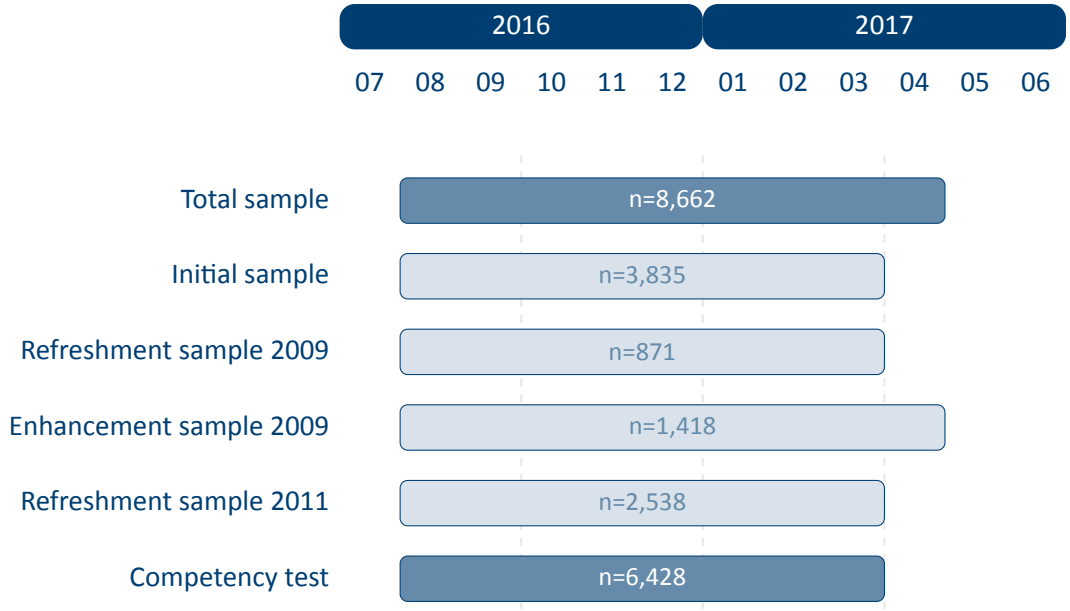

Figure 12: Field times and realized case numbers in wave 9

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted personal interviews (CAPI) including computer-based competency tests (CBA); computer-assisted telephone interviews (CATI) for those who were not designated for testing or insisted on a telephone interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.10 Wave 10: 2017/2018 (9th NEPS survey)

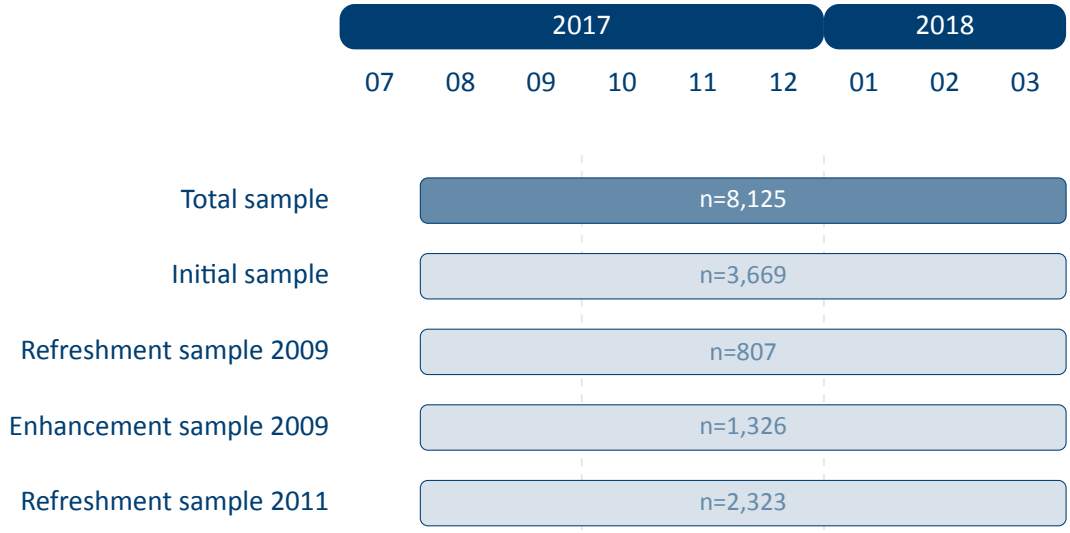

Figure 13: Field times and realized case numbers in wave 10

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted personal interviews (CAPI); computer-assisted telephone interviews (CATI) for those who were not designated for testing or insisted on a telephone interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.11 Wave 11: 2018/2019 (10th NEPS survey)

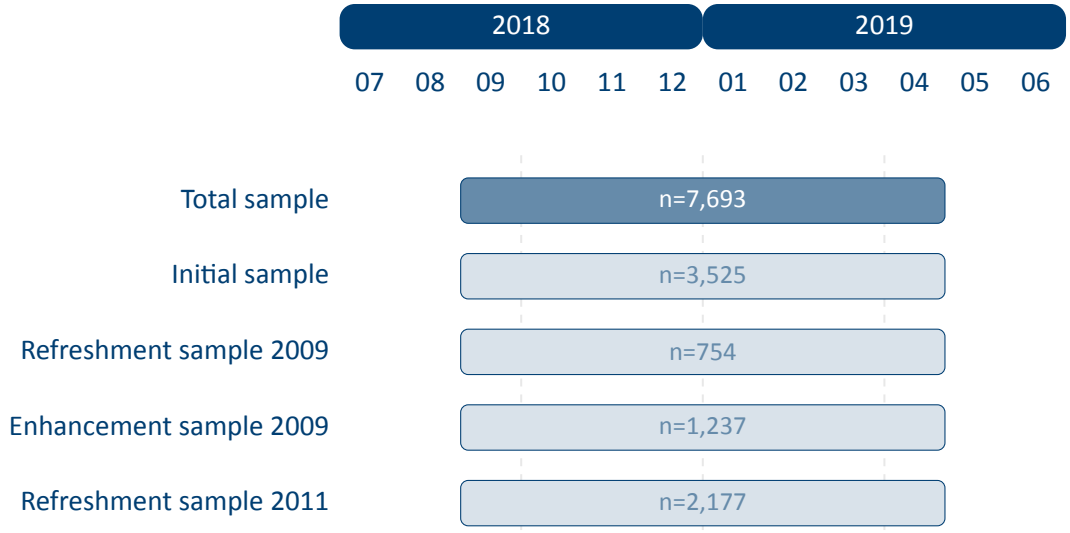

Figure 14: Field times and realized case numbers in wave 11

- **Samples**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Sample Follow-up survey with respondents willing to participate in the panel

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sample Follow-up survey with respondents willing to participate in the panel

- **Mode of survey** Computer-assisted personal interviews (CAPI); computer-assisted telephone interviews (CATI) for those who were not designated for testing or insisted on a telephone interview
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

3 General Conventions

The compilation of NEPS Scientific Use Files follows two general paradigms on how to edit the source data (i. e., the data that is delivered to the LIfBi Research Data Center by the survey agencies). There may be exceptions to these principles that are explicitly noted in the respective documentation material.

The first and foremost paradigm in creating NEPS Scientific Use Files is the one of unaltered data. Wherever possible, the data editing procedures do neither change nor destruct the content of the original data. We consider this to be the basis for preserving the full research potential of the collected data. For this reason, no corrections are made during the entire data editing process to ensure the content validity of the source data. As a consequence, this means that the data in the Scientific Use File may contain implausible values, unless corresponding controls were already provided in the survey instrument. Only in rare cases, in which the responsible developers of a variable require the removal of clearly implausible information, these values are replaced by the special missing code *implausible value removed* (–52, see Table 6). The most prominent (and only systematic) exception to this general paradigm concerns the recoding of open responses that could originally have been recorded directly as closed responses (see section 3.4 for details). NEPS Scientific Use Files are equipped with a dataset `EditionBackups` that contains backup information for all content that has been modified by such recoding procedures (see section 4.2.5 for details).

The second paradigm is to integrate the data as much as possible without compromising the usability of the Scientific Use File. The underlying assumption is that for a vast majority of data users it is far more comfortable to reduce already integrated data for a specific analysis as opposed to correctly compile the relevant information from scattered source data themselves. In the end, each Scientific Use File contains only a few dozen integrated panel and spell datasets according to a general structure (see section 4.1.2 and section 4.1.3 for details), even if the compilation is based on several hundred separate source dataset files.

In addition to these two basic principles of data editing, there are several conventions for the data structure of all NEPS Scientific Use Files. The aim of this structuring is to ensure a maximum of consistency between the data of the different starting cohorts. In other words, a researcher who is familiar with the data logic of a particular NEPS cohort should be able to immediately recognize this structure when starting to work with data from another NEPS cohort. These conventions are explained in more detail in the following sections.

3.1 File names

The naming of the data files in NEPS Scientific Use Files follows a series of rules that are summarized in Table 3. The different elements are concatenated with an underscore (_) to generate the complete file name.

Table 3: Naming conventions for NEPS file names

Element	Definition
`SC[1-6]`	**Indicator for the starting cohort** 1 = Newborns 2 = Kindergarten 3 = Fifth-grade students 4 = Ninth-grade students 5 = First-year university students 6 = Adults
`[filename]`	**Meaning of the file name** *Prefix*: `x` = cross-sectional file; `sp` = spell file; `p` = panel file *Keyword*: indicates the content of the corresponding file (e. g., data file `xTarget` contains cross-sectional data from the target questionnaire; `spSchool` contains spell data from the school history) File names of generated datasets do not have a prefix and always start with a capital letter (e. g., `CohortProfile`, `Weights`)
`[D,R,O]`	**Indicator for the confidentiality level** D = Download version R = Remote access version O = On-site access version
`[#]-[#]-[#](_beta)`	**Indicator for the release version** *First digit*: the main release number is incremented with every further wave in the Scientific Use File; e. g., the first digit 5 implies that data of the first five survey waves are included in the release *Second digit*: the major update number is incremented with every bigger change to the Scientific Use File; major updates affect the data structure, so updating the syntax files may be necessary *Third digit*: the minor update number is incremented with every smaller change to the Scientific Use File; minor updates affect the content of cells, so updating the syntax files is not necessary `_beta`: this suffix indicates a preliminary Scientific Use File release which allows users to test the data before the main release; the beta release is no longer available after the main release

For instance, the file `SC6_CohortProfile_D_11.0.0.dta` refers to the *CohortProfile* data of *Starting Cohort 6* in its *Download* version of the Scientific Use File release *11.0.0*.

3.2 Variables

The naming conventions for variables in NEPS Scientific Use Files aim to ensure maximum consistency both between the panel waves and between the starting cohorts. The names also refer to different characteristics and thus allow the data user an orientation regarding the contents of the variables. The principles of these naming conventions are exemplified in Figure 15. It has to be noted that a separate nomenclature is used for variables from competence measurements. Section 3.2.1 offers a detailed description of the general naming conventions for NEPS variables; the logic of naming competence variables is introduced in section 3.2.2.

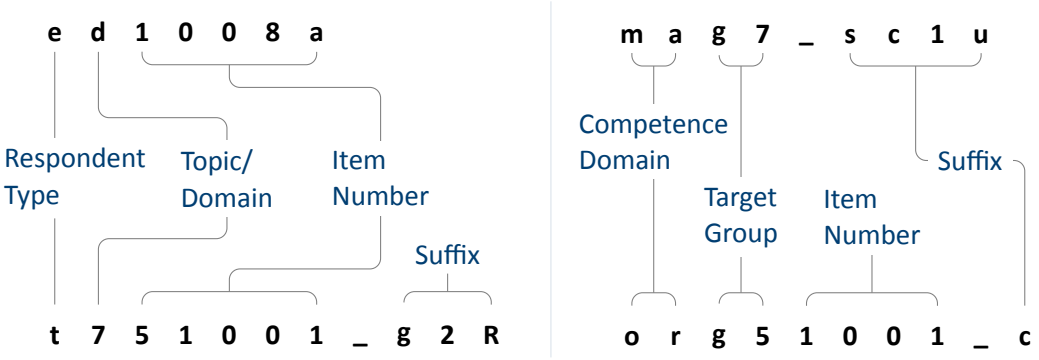

Figure 15: General variable naming (left) and competence variable naming (right)

3.2.1 Conventions for general variable naming

A variable name consists of up to four elements: the respondent type, the domain of information, an item number, and an optional suffix providing further information.

Table 4: Conventions for variable names

Digit	Description
1	**Respondent type** Indicator to which group of respondents the variable refers; note that variables related to the target person start with `t` even if the target person was not the actual informant (e. g., generated variables, list data from schools/kindergartens) `t` = Target person `p` = Parent of target person `e` = Educator/childminder `h` = Head/manager of institution (information about school/kindergarten)

(...)

Table 4: (continued)

Digit	Description
2	**Topic/domain** Indicator to which theoretical dimension or educational stage the variable refers 1 = Competence development 2 = Learning environments 3 = Educational decisions 4 = Migration background 5 = Returns to education 6 = Interest, self-concept and motivation 7 = Socio-demographic information a = Newborns and early childhood education b = From kindergarten to elementary school c = From elementary school to lower secondary school d = From lower to upper secondary school e = From upper secondary school to higher ed./occ. training/labor market f = From vocational training to the labor market g = From higher education to the labor market h = Adult education and lifelong learning s = Basic program x = Generated variables
3–7	**Item number** Indicator for the item number which typically consists of four numeric characters plus one alphanumeric character
8–11	**Suffixes** (optional, see below) Indicator for several types of variables; separated from the previous characters by an underscore

Suffixes

- *Generated variables:* The _g# suffix indicates a generated variable; the running number after _g is in most cases a simple enumerator (e. g., _g1). Since scale indices are generated by a set of other variables, they are also identified by a _g# suffix. Note that scale indices are named after the first of the set of variables from which they were generated. In this case, numbering is only relevant if the first variable is identical for several scale indices. The number after _g is in most cases a simple enumerator. However, there are two types of generated variables that assign specific meanings to digits, namely regional and occupational variables. The former are based on the Nomenclature of Territorial Units for Statistics (NUTS):

- g1: Indicator for East or West Germany
- g2: NUTS level 1 (federal state/Bundesland)
- g3: NUTS level 2 (government region/Regierungsbezirk)
- g4: NUTS level 3 (district/Kreis)

Generated variables for occupational classifications and prestige indices (see also section 3.4):

- g1: KldB 1988 (German Classification of Occupations 1988)
- g2: KldB 2010 (German Classification of Occupations 2010)
- g3: ISCO-88 (International Standard Classification of Occupations 1988)
- g4: ISCO-08 (International Standard Classification of Occupations 2008)
- g5: ISEI-88 (International Socio-Economic Index of Occupational Status 1988)
- g6: SIOPS-88 (Standard International Occupational Prestige Scale 1988)
- g7: MPS (Magnitude Prestige Scale)
- g8: EGP (Erikson, Goldthorpe, and Portocarero's class categories)
- g9: BLK (Blossfeld's Occupational Classification)
- g14: ISEI-08 (International Socio-Economic Index of Occupational Status 2008)
- g15: CAMSIS (Social Interaction and Stratification Scale)
- g16: SIOPS-08 (Standard International Occupational Prestige Scale 2008)

- *Versions of variables:* If question formulations, interviewer instructions, etc. change between panel waves to such an extent that sufficient meaning equivalence is no longer guaranteed, the answers to these questions are stored in different versions of a variable. The data for the latest and most current version of a question are provided under the variable name without any version suffix. Previous item versions are identified by `var_v1` for the data before the question was modified for the first time, `var_v2` for the data before the question was modified for a second time, and so on.
- *Harmonized variables:* The suffix `var_ha` indicates a harmonized variable in which common information from different versions of a variable is integrated. This is often done by aggregating detailed value characteristics into common superordinate categories. In other words, a harmonized variable reflects the lowest common denominator of information from a variable and its version(s).

- *Wide format variables:* The _w# suffix indicates variables that are stored in wide format. Note that this suffix does not necessarily imply a wave logic. The presence of a set of variables var_w1, var_w2, …, var_w10 may mean that there are up to 10 values for this variable per person or episode. This is the case, for example, if the corresponding item in the survey instrument was repeatedly measured in a loop. Another example concerns the date of the competence measurement within a survey wave if it took place on two different days.
- *Confidentiality level:* The _D, _R, or _O suffix indicates variables that have been modified during the anonymization process (see section 1.4). The suffix _O signalizes that data in this variable is only available via on-site acces; _R refers to variables where access to detailed information is only possible via RemoteNEPS and on-site stay; and _D means that data in this variable has been extracted from the corresponding _O or _R variable to make at least some information available in the download version of the Scientific Use File. The confidentiality suffixes stand either alone (e. g., country of birth: t405010_R) or in combination with other suffixes (e. g., district of place of birth: t700101_g3R).

3.2.2 Conventions for competence variable naming

The naming of variables from competence measurements and direct measures follows an alternative logic. In contrast to other data files, the competence datasets (xTargetCompetencies and xDirectMeasures in Starting Cohort 1) are structured in wide format; that is, all values for a single respondent are represented in one row of the data matrix. Thus, the integration of information from several competence domains collected across several survey waves requires specific conventions for variable naming. Competence variables are characterized by three name components and supplementing suffixes. The first component indicates the competence domain of the measurement (two characters, e. g., vo for vocabulary). The second part identifies the target group and the survey wave or class level in which the measurement was first used (two or three characters, e. g., k1 for kindergarten children during the first wave). The target group identification does not necessarily indicate the cohort or testing wave of the measurement. Please refer to the explanations in the next section for the special features of repeatedly used test items. Some competence measurement are not designed for specific age groups, but are implemented unmodified in different cohorts and testing waves. In these cases the target group is defined as ci (cohort invariant). The third component denotes the item number. Table 5 contains a list of all possible specifications of the three parts of a competence variable name.

The additional suffixes inform about the mode of test execution if more than one survey modus has been applied for a measurement and about the sort of item score and overall competence score. There is a distinction between scored items named [varname]_c and scored partial credit-items named [varname]s_c. The latter is relevant if more than one correct solution is possible (e. g., value 0 = 0 out of two points, value 1 = 1 out of two points, value 2 = 2 out of two points), whereas the former is applied for dichotomous solutions (value 0 = not solved, value 1 = solved). In addition to the item scores, several aggregated

scores are provided for competence measurements. They are indicated by `_sc[number]` and a few special suffixes for Starting Cohort 1. A letter appended to the suffix indicates that more than one aggregated score for a competence measurement is available (e. g., `_sc3a`, `_sc3b` for different sum scores of any test). Detailed descriptions on how the aggregated competence scores were estimated can be found in the domain-specific documentation reports. The last part of Table 5 shows all possible suffixes and their meanings.

Table 5: Conventions for competence variable names

Part I: Competence Domain (2 chars)

`ba`	Business administration and economics
`bd`	Backwards digit span: Phonological working memory
`ca`	Categorization: SON-R subtest
`cd`	Cognitive development: Sensorimotor development
`de`	Delayed gratification: Executive control
`dg`	Domain-general cognitive functions (DGCF): Cognitive basic skills
`ds`	Digit span: Phonological working memory
`ec`	Flanker task: Executive control
`ef`	English foreign language: English reading competence
`fa`	FAIR: Concentration abilities
`gr`	Grammar: Listening comprehension at sentence level
`hd`	Habituation-dishabituation paradigm
`ic`	Information and communication technology literacy (ICT)
`ih`	Interaction at home: Parent-child interaction
`ip`	Identification of phonemes: Phonological awareness
`li`	Listening: Listening comprehension at text/ciscourse level
`lk`	Early knowledge of letters
`ma`	Mathematical competence
`md`	Declarative metacognition
`mp`	Procedural metacognition
`nr/nt`	Native language Russian/Turkish: Listening comprehension
`on`	Blending of onset and rimes: Phonological awareness
`or`	Orthography
`re`	Reading competence
`ri`	Rimes: Phonological awareness
`rs`	Reading speed
`rx`	Early reading competence
`sc`	Scientific competence
`st`	Scientific thinking: Science propaedeutics
`vo`	Vocabulary: Listening comprehension at word level

Part II: Target Group (1 char)**, followed by wave or grade** (1-2 digits)

(...)

Table 5: (continued)

n#	Newborns in wave #
k#	Kindergarten children in wave #
g#	Students at school in grade #
s#	University students in wave #
a#	Adults in wave #
ci	Cohort invariant (for instruments administered unchanged in all cohorts)

Part III: Item number (3-4 chars)

For some competence domains, these item numbers follow a certain scheme, but for most competence domains they only indicate the different items

Part IV: Suffixes (starting with an underscore)

_pb	Paper-based test modus (proctored)
_cb	Computer-based test modus (proctored)
_wb	Web/Internet-based test modus (unproctored)
_c	Scored item variable (s_c for partial credit-items)
_sc1	Weighted likelihood estimate (WLE)[1][2]
_sc2	Standard error for the WLE[2]
_sc3	Sum score
_sc4	Mean score
_sc5	Difference score (for procedural metacognition)
_sc6	Proportion correct score (for procedural metacognition)
_p	Maximum value for an item (only in Starting Cohort 1)
_b	Minimum value for an item (only in Starting Cohort 1)
_m	Mean value for an item (only in Starting Cohort 1)
_s	Sum value for an item (only in Starting Cohort 1)
_n	Number value for an item (only in Starting Cohort 1)

Identification of repeated test items

In some competence measurements identical items are implemented in different testing waves (e. g., mathematics). Identifying repeatedly measured test items in NEPS data can be easily done by looking for competence variables with an identical word stem. If the same test item is surveyed in different survey waves or starting cohorts, the variable name is equiped with an additional suffix. It is important to know that the two or three characters for the target group

1 WLEs and their standard errors are estimated in tests that are scaled based on models of Item Response Theory (cf. Pohl and Carstensen, 2012).

2 WLEs and their standard errors are corrected for test position; uncorrected WLEs and standard errors are indicated by an additional u in the suffix (_sc1u, _sc2u).

(second part of the variable name) always indicate the wave or cohort in which the item was initially used. The word stem is then fixed and does not change when the item is used again in later waves or other cohorts. If the variable name does not contain a suffix for repeated use, then the second part of the word stem refers to the target group of the realized measurement. However, if the variable name includes a suffix for repeated use, then the values of the variable do not refer to the target group according to the word stem, but to the target group according to the suffix. The suffix that points to the repeated use consists of two parts: The first element indicates the starting cohort of current item administration and the second element indicates the cohort or testing wave of current item administration.

The following example illustrates this logic: The competence variable `vok10067_sc2g1_c` is a vocabulary item (`vo`) that was initially measured during the first kindergarten survey wave (`k1`). However, the values in this variable reflect the scored measurements of this item´s repeated use among the target persons of Starting Cohort 2 in the course of the survey wave in grade 1 (`_sc2g1`), and thus two years after the first measurement.

3.2.3 Labels

As a rule, the seven-digit variable names are not sufficient to uniquely identify the respective contents of the variables and to differentiate sufficiently between items. All variables therefore have *variable labels* for more detailed description. In addition, most variables contain *value labels* for the respective value characteristics. All information is available in German and English and is typically displayed directly in the editor of the statistics program, e.g. for frequency calculation or when searching the data (applies to SPSS and Stata, see also section 1.3). For users of R, see section A.1 for hints on this.

In addition to the variable and value labels, the datasets also contain extended characteristics for variables. These include the question text from the survey instrument, any associated interviewer instructions and filter conditions, as well as other meta information. All extended features can be accessed directly within data files. Stata users apply the `infoquery` command for this, which is part of the *NEPStools* package (see section 1.7). SPSS users will find the additional meta information in the “Variable View” at the end of each variable line.

As explained in more detail in section 4, NEPS data from different waves are integrated as much as possible. For panel data, this primarily means that many variables contain information from multiple waves. In most cases of such a data integration, the meta information between the waves does not change. However, if there are changes to the meta information of a repeatedly measured item, and if these changes are not significant enough to store the information in separate variables, the assignment of meta information follows a general rule: **All meta information available in a dataset always corresponds to the most recently instrument in which the respective item was used.**

A concrete example is the adaptation of interviewer instructions or question texts from the informal salutation (“Du”) to the formal salutation (“Sie”). Since these changes are not expected

to have any effect on how a question is answered, the corresponding values across multiple waves get integrated into one variable. If you request the meta information of such a variable in the dataset, the wording of the latest item formulation will be displayed (in the given example with the formal salutation "Sie"). In case of uncertainties regarding the continuity of meta information of a variable across different waves, we recommend to consult the respective *survey instruments* for the individual waves.

3.3 Missing values

The NEPS data contain various missing codes to differentiate between various types of missing values. All missing codes have negative values or are defined as system missing. Depending on the statistics program used, you must ensure that these codes are processed correctly. In the offered SPSS datasets, the missing codes are already defined as missing values. When using Stata, the missing codes must first be excluded from the analyses by the user as missing values. For this purpose the command `nepsmiss` is available in the *NEPStools* package (see section 1.7). The general recommendation is to always carefully check the frequency distributions of the relevant variables before running an analysis.

We distinguish between three types of missing codes, which are summarized in Table 6 and described in more detail below.

Item nonresponse: The first type of missing codes occurs when a person has not (validly) replied to a question.

- The most common cases of item nonresponse are *refused* (–97) answers and *don't know* (–98) answers.
- Missing values specified by the survey agency due to an incorrect use of the instrument are coded as *implausible value* (–95).
- Within the competence data, there is a special missing code indicating that a question or test item was *not reached* (–94) due to time constraints or other test setting restrictions. It usually signals that the respondent had to quit the test somewhere before this point.
- Other missing codes refer to various categories of *item-specific nonresponse* (–20, …,–29) such as –20 for "*stateless*" in the citizenship variable `p407050_D`.

Not applicable: The second type of missing codes occurs when an item does not apply to a respondent.

- The code *missing by design* (–54) is assigned when respondents in a (sub)sample have not been asked the respective questions. This is usually the case if the administered survey instrument contains (sub)sample-specific questionnaire modules. The code is also used for the more general case where values of a variable are not available due to the design of the survey (e. g., measurement rotation with either easier or heavier test tasks).

Table 6: Overview of missing codes

Code	Meaning	Note
Item nonresponse		
–94	not reached	only relevant for instruments with time restrictions (e. g., competency test measures)
–95	implausible value	assigned by the survey agency (e. g., multiple answers to a one-answer question in PAPI mode)
–97	refused	as default answer option to the question
–98	don't know	as default answer option to the question
–20,...,–29	*various*	item-specific missing with informative value label (e. g., "*no grade received*" for question about school grades)
Not applicable		
–54	missing by design	question not included in (sub)sample-specific instrument (e. g., not asked in all waves)
–90	unspecific missing	in PAPI mode (e. g., question not answered, empty field)
–93	does not apply	as default answer option to the question
–99	filtered	filtered out question, in other than CATI/CAPI mode
.	*system*	filtered out question, in CATI/CAPI mode
Edition missings (recoded into missing)		
–52	implausible value removed	only at the request of the responsible item developers
–53	anonymized	sensitive information removed (e. g., country of birth of parents in the download version)
–55	not determinable	not sufficient information to generate the variable value (e. g., net household income `t510010_g1`)
–56	not participated	in case of unit nonresponse, only used in certain datasets

- If the respondent him-/herself or the interviewer indicates that a particular question is not applicable to the person, the missing value is coded as *does not apply* (–93). If, on the other hand, filtering takes places automatically via the survey instrument, the coding of the filtered out questions depends on the survey mode: in CATI and CAPI interviews, a system missing value (`.`) is assigned for this; in all other modes the respective code is *filtered* (–99).
- Missing values that cannot be assigned to any of the above categories are coded as *unspecific missing* (–90). This missing code usually occurs in PAPI questionnaires when a respondent has not answered a question for unknown reasons.

Edition missings: The third type of missing codes is defined in the process of data preparation for the Scientific Use File.

- If in the data edition process certain values which are not considered to be meaningful are re-

quested to be removed, the missing code *implausible value removed* (–52) is assigned in their place. As a rule, however, all values from the field instruments are included in the Scientific Use File without further plausibility checks (see section 3). Only in exceptional cases, when the responsible item developers explicitly recommend a removal of implausible answers, this missing coding is done.

- Sensitive information that is only available via remote and/or on-site access is encoded in the more anonymized data access option as *anonymized* (–53).
- In general, coding schemes are used to generate variables (e. g., occupational coding; see section 3.4). However, if the information from the original data is not sufficient to generate a suitable value, the missing code *not determinable* (–55) is used instead.
- If a person was not present during the interview or did not complete a questionnaire at all, even though it was administered to the person, the concerning variables receive the code *not participated* (–56). This missing code is special in that target persons without survey data for a certain wave (e. g., due to illness) are usually not included in the corresponding datasets. This missing code is only used in the special cases of datasets that integrate several waves in wide format (e. g., `xTargetCompetencies`) or that also contain observations for non-participating persons in a wave (e. g., `CohortProfile`).

3.4 Generated variables

Coding and recoding of open responses

At various points in the NEPS survey instruments there are so-called open questions where respondents can or should enter their answers as text. A typical example is information about occupation.

The open text format allows respondents to specify anything they want. A practical way to deal with the resulting string information is to code and recode the information for further processing and later analyses. In general, coding describes the process of assigning one or more codes from selected category schemes to the string information, e. g. the classification of occupational data according to DKZ (database of documentation codes, *Datenbank der Dokumentationskennziffern*) or WZ (classification of economy branches, *Klassifikation der Wirtschaftszweige*).

The term “recoding” is used here to describe the process of assigning a code from an already presented closed answer scheme. This usually applies to semi-open question formats where respondents enter a text under the category “other”, but which can be assigned ad hoc to one of the given closed answer categories. The recoding therefore does not define any new codes; the presented answer scheme of the respective question is not extended.

The most common and comprehensive coding scenarios in the fields of occupation, education, branches, courses, and regional information are processed by the Research Data Center (FDZ-LIfBi) itself. Other coding tasks are distributed among the responsible departments at the LIfBi in Bamberg and the partners in the NEPS consortium.

Derived scales and classifications

The (re-)coding of open answers or string entries into primary classifications (such as DKZ2010 or WZ08) is a first and essential step towards making this information available within the NEPS Scientific Use Files in a user-friendly and analyzable way. The standardized derivation of further classifications or scales, especially in the area of educational qualifications and occupational titles, is a second and no less important step. At least three types and objectives of derivations can be distinguished:

- Derivations from primary classifications (and originated from string entries/open answers) into other classifications that function as a standard schema in other studies or international comparisons, e. g. ISCO instead of KldB in the field of occupations
- Derivations from primarily closed response schemes into general classifications and schemes using auxiliary information, e. g. ISCED or CASMIN from school certificate and training data plus additional information on the type of school/training
- Combination of the two types, e. g. EGP class scheme via derived ISCO classification plus information on self-employment and supervisory status

Figure 16 shows the derivation paths for several occupational scales and schemes provided in the NEPS. A detailed description of the standard derivations for educational attainment (ISCED, CASMIN and Years of Education) can be found in the corresponding documetation report by Zielonka and Pelz, 2015.

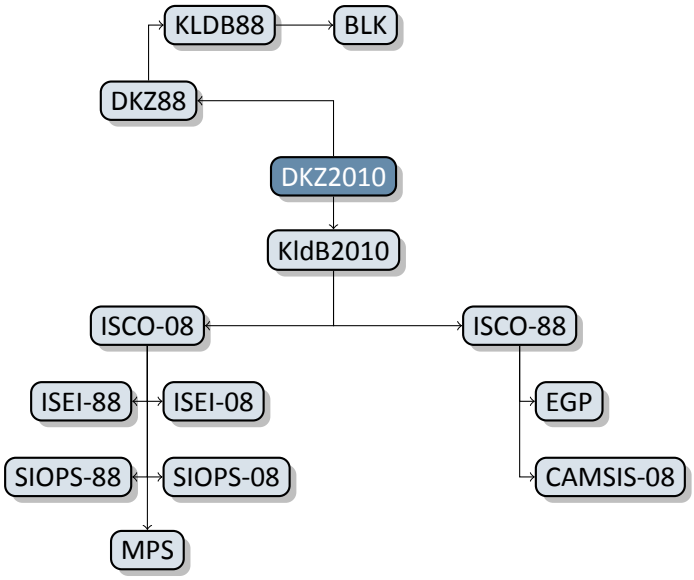

Figure 16: Derivation paths for several occupational scales and schemes provided in the NEPS

4 Data Structure

4.1 Overview

The broad objectives and the large size of the longitudinal NEPS surveys inevitably lead to a complex database. The crucial task is to organize this data in a well-structured, reproducible and user-friendly way, while at the same time preserving a maximum level of detail in the data. To meet this challenge, a number of additionally generated variables and datasets is included in the Scientific Use File to facilitate the preparation and analysis of the data.

In principle, all information collected in the course of a panel wave is appended to the information from previous waves in the corresponding data file, together with the required identifiers. Data files containing longitudinal information from several waves are denoted with a *p* in the file name. For example, the `pTarget` file(s) contain(s) information from the target persons' interviews with one row in the dataset representing the information of one target in one wave.

This convention does not apply to all longitudinal data. For example, there are competence measurements that were repeatedly carried out with the same target persons. However, since the instruments, i.e. the content of competence tests, vary over time, the corresponding information is structured in wide format (for more details, see section 3.2.2 or section 4.2.35). Such cross-sectionally structured data files with one line representing information of a respondent from all waves are marked with a *x*.

Another type of data structuring refers to episode data. For the information collected prospectively and retrospectively using iterative question sets, the Scientific Use File provides life area-specific spell datasets. These datasets are marked by a preceding *sp*. An example is the file `spEmp`, which informs about current and former episodes of employment.

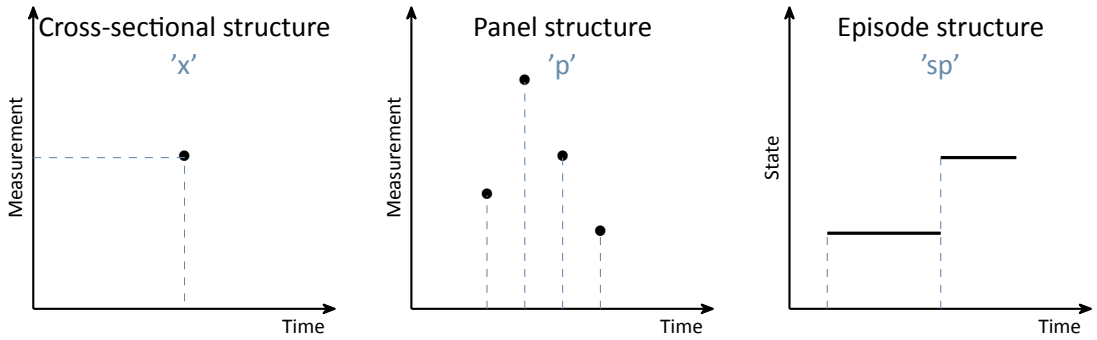

Figure 17: Different types of data structures

In addition to interview and test data provided by the respondents as well as episode data, there are also so-called paradata or derived information. These data files can be identified by

the leading capital letter in the name (e. g. `Weights` or `CohortProfile`). In most cases, these datasets correspond to the panel structure.

4.1.1 Identifiers

The multi-level and multi-informant design of the NEPS and the distribution of survey information across different datasets requires the use of multiple identifiers. The following identifier variables are relevant in this Starting Cohort for linking data:

ID_t identifies a target person. The variable `ID_t` is unique across waves and samples (and also starting cohorts).

wave indicates the sample wave in which the data was collected.

In addition, there are other identifier variables to indicate a target person's membership in a particular test group (`ID_tg` in `CohortProfile`, not applicable to all starting cohorts) and to indicate the interviewer who conducted the respective interview (`ID_int` in `Methods` datasets). However, these identifiers are not relevant for the merging of information from different datasets and are negligible for most empirical applications.

4.1.2 Panel data

As mentioned above, all information from subsequent survey waves are appended to the already existing data files (as far as possible). This method of data processing generates *integrated panel data* files in a long format as opposed to providing one separate file per wave (where each file contains only the information from a single wave). When working with the integrated panel data in the NEPS Scientific Use Files, the following points should be considered:

- A row in the dataset contains the information of one respondent from one survey wave.
- This means that more than one identifier variable is needed to identify a single row for uniquely selecting and merging information from different datasets. These are usually `ID_t` and `wave`.
- It also means that although not all variables were administered in each survey wave, the integrated structure of the dataset contains cells for all variables of all waves. If no data is available, e.g. because a variable was not queried in a particular wave, the corresponding cells are filled with a missing code (see section 3.3).
- Once information about a variable has been surveyed from one individual across multiple waves, the corresponding data is distributed across multiple rows in the dataset.

This long format is usually the preferred data structure for the analysis of panel items with information from several waves. However, cross-sectional information is often also required, e. g. because it depicts time-invariant characteristics or was collected only once for other reasons. In most analysis scenarios, the combined set of relevant variables is not measured in a single wave. Therefore, the corresponding data cannot be analyzed together straightaway because it is stored in *different rows* of the dataset. Cross-tabulating these variables in their current state results in an L-shaped table in which all observations of one variable fall into the missing category of the other variable and vice versa. The best way to deal with this issue depends very much on the intended analysis and the methods used. Two typical procedures are:

- First, the integrated panel data file is split into wave-specific subfiles so that each dataset contains only information from one wave. The relevant information from these subfiles is then merged together by using only the respondent's identifier (`ID_t`) as key variable. The `wave` variable is not needed here and remains neglected. Before this step, variables may need to be renamed to make them wave-specific identifiable. The result is a dataset with a cross-sectional structure in which the information of a respondent is summarized in one single row (wide format). Stata's *reshape* command (and similar tools in other software packages) basically follow this strategy.
- Alternatively, the panel structure is retained and the values from observed cells are copied into the unobserved cells. If, for example, the place of birth was only surveyed in the first wave, the corresponding value can be transferred to the respective cells of the other waves of the respondent. This method is particularly useful for time-invariant variables (e. g. country of birth, language of origin), which are usually collected only once in a panel study.

4.1.3 Episode or spell data

Handling cross-sectional data is usually not a problem. Most data users also know how to work with and analyze panel data. Episode or spell data, on the other hand, present a particular challenge for understanding data processing. The following explanations should help to deal with this data format in a meaningful and appropriate way.

In the episode (or spell) data, there is one row for each episode that was captured. Note that the number of episodes per se is independent of the survey wave. This means that several episodes (=several rows) can be recorded in a single wave. Usually, a start date and an end date describe the duration of an episode. The remaining variables in such spell datasets contain additional information about that episode. These characteristics are chronologically linked to the episode. In other words: Especially for time-variant variables (e. g. ISEI, CASMIN) it is important to know that the respective values indicate the status of the respondent **at the time of the episode** and not necessarily the current status.

To give an example: In the spell dataset `spEmp` there is a period of time for a certain respondent in which he or she worked without interruption in a particular job. If this person changes to a new job, this marks a new episode which is stored in a new data row. Further changes in

this context may also lead to new episodes, e. g. a change of employer or the conclusion of a new employment contract (but not if the salary, working hours or other characteristics of the respective job change). Episodes can therefore be understood as the smallest possible units of one's life history, in this case the employment biography. As soon as there are several relevant changes in such a biography between two consecutive interview dates, this is reflected in several data rows per survey wave.

In addition to these (time) episode data, which we call *duration spells*, there are two other types of episode data: Occurring events or the transition from one state to another (e. g., change of marital status, change of educational level) are recorded in so-called *event spells*; the existence of children, partners, etc. is recorded in so-called *entity spells* with one row per entity. Regardless of the type of episode, two variables are usually necessary to identify a single row in the data file, namely the respondents' identifier `ID_t` and an episode, event or entity numerator such as `spell` that identifies a duration spell. More detailed information on the required identifier variables can be found on the respective data file pages in in section 4.2.

There is one important circumstance to consider when working with NEPS spell data. This concerns *subspells*. Biographical episode data are collected retrospectively. During an interview the respondents are asked about all episodes that have occurred since the last interview (in the first interview it is since birth or a certain age). If an episode is finished at the time of the interview, the respondent reports a corresponding end date and the spell is completed. Difficulties arise when the episode is not yet finished at the time of the interview, i.e. it is ongoing. Such an episode appears as right-censored in the dataset. In the next interview, this episode is then queried using preloads in the course of "dependent interviewing" in such a way that the respondent can report whether it has been finished in the meantime or whether it continues. Technically this leads to several rows in the data structure, which can be distinguished by the variable `subspell`:

- original (right-censored) episode reported in initial wave (`subspell=1`)
- continued episode reported in next wave(s) (`subspell=2`, `subspell=3`, etc.)

Normally, attention is paid to the last subspell, as it contains the most up-to-date information about an episode. However, the most recently captured information for an episode may contain missing values, or the value from the last-mentioned sub-episode may have been transferred to it. To facilitate the handling of spell datasets, *harmonized rows* with the (presumed) most relevant information for all episodes are generated from the summary of the corresponding subspells. This information often corresponds to the last (non-missing) reported entry, but sometimes also to the first (non-missing) reported entry of an episode. The harmonized rows are selectable by the filter condition `subspell=0`. The same applies to all episodes reported as completed without any subspells.[1] If there is no particular interest in subspell information, it is recommended to use only the harmonized data rows for analysis:

```
keep if subspell==0
```

1 The variable `spgen` indicates whether an episode was originally reported as finished (`spgen=0`) or whether it is a harmonized (generated) episode (`spgen=1`).

Note that the information cumulated in the harmonized spells (last valid available) may originate from different waves. Please be also aware that the selection of harmonized spells should **not** be used when working with information stored in wide format (e. g., interruption episodes of vocational training spells in `spVocTrain`).

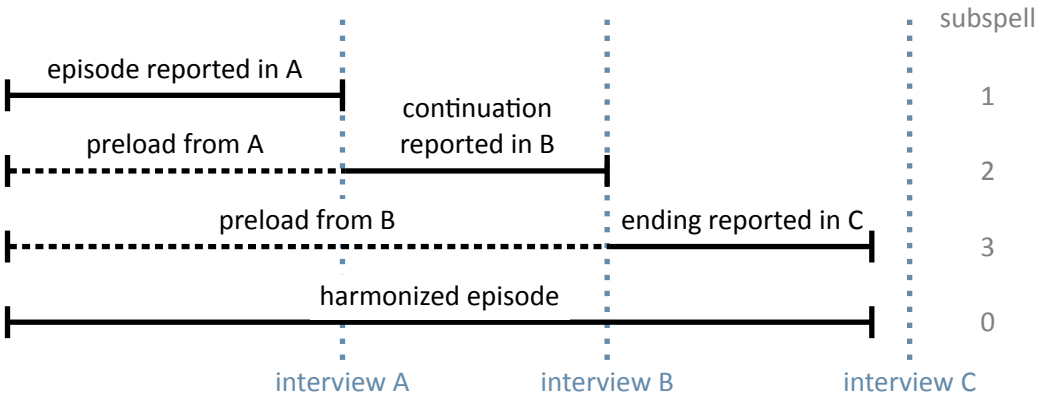

Figure 18: Logic of subspells

4.1.4 Revoked episodes

In order to reduce seem bias, spell data are preloaded by prior wave information. This information from previous waves can be revoked by the respondent during the current interview. Spell datasets therefore also contain information about revocations (variables `disagint`, `disagwave`). The reasons for a revocation or contradiction are manifold; they depend mainly on the information that is presented to the respondent to remember the episode (see the questionnaires for the exact wording of the episode data collection).

If an episode is later revoked by the respondent, this episode is marked accordingly in the dataset. The respective information is collected again in the current interview and saved as a new episode in the actual data collection wave. The updated spell is not flagged as a corrected spell. The identification of related spells (=previously given information plus their correction in the following wave) is up to the data user. Please note: Since it is technically impossible to specify a start date for an episode prior to the last interview date, virtually all corrected spell episodes are left-censored. The only exception are episodes that started on the interview date of the last wave.

In addition to the possibility of revoking an episode in the course of the subsequent survey wave, there is also the possibility of revoking an episode during the interview. For this purpose, a *check module* is used after the biographical information has been recorded. It ensures that the life course is captured as completely as possible. The biographical episodes asked in the thematically structured questionnaire modules are already examined in the interview for their chronological plausibility. To verify the temporal consistency of the events across the questionnaire modules, a complete overview of all types of events is created. For this purpose, all

recorded biographical episodes are displayed in tabular form in the check module. If gaps or overlaps are indicated, the respondent will be asked again. He or she can then make corrections, add new episodes, or revoke already recorded episodes. The identification of episodes revoked in the check module is possible in the spell datasets by the variable "Biography: Type of event (edited)" (`spms=20`); the addition of new episodes in the check module is indicated in the variable "Episode mode" (`ts23550=4` in `spEmp`). A detailed description of the functionality of the check module for reported life courses (in German language) can be found on the website in the section "Data Documentation":

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

4.2 Data files

In the following section, every data file of this Starting Cohort is explained in a subsection, including a data snapshot and an example of data usage (in Stata). The examples are written so that everyone knowing Stata should easily understand it. You also do not need additional ado files installed, although you are highly advised to use the `nepstools` (see section 1.6).

To ease your understanding of the relationship of those files, Figure 19 provides an overview. The edges in this graph symbolize how a data file may be linked to other files. This is not meant to document every possible data link you could do but rather tries to give you an idea which data files relate most. By clicking on a node, you get directed to this data file's explanatory page.

You need to set the following globals for the Stata examples to work. Just adapt and copy the lines below to the top of the syntax files or execute them in your Stata command line before running the syntax:

```
** Starting Cohort
global cohort SC6
** version of this Scientific Use File
global version 11-0-0
** path where the data can be found on your local machine
global datapath Z:/Data/${cohort}/${version}
```

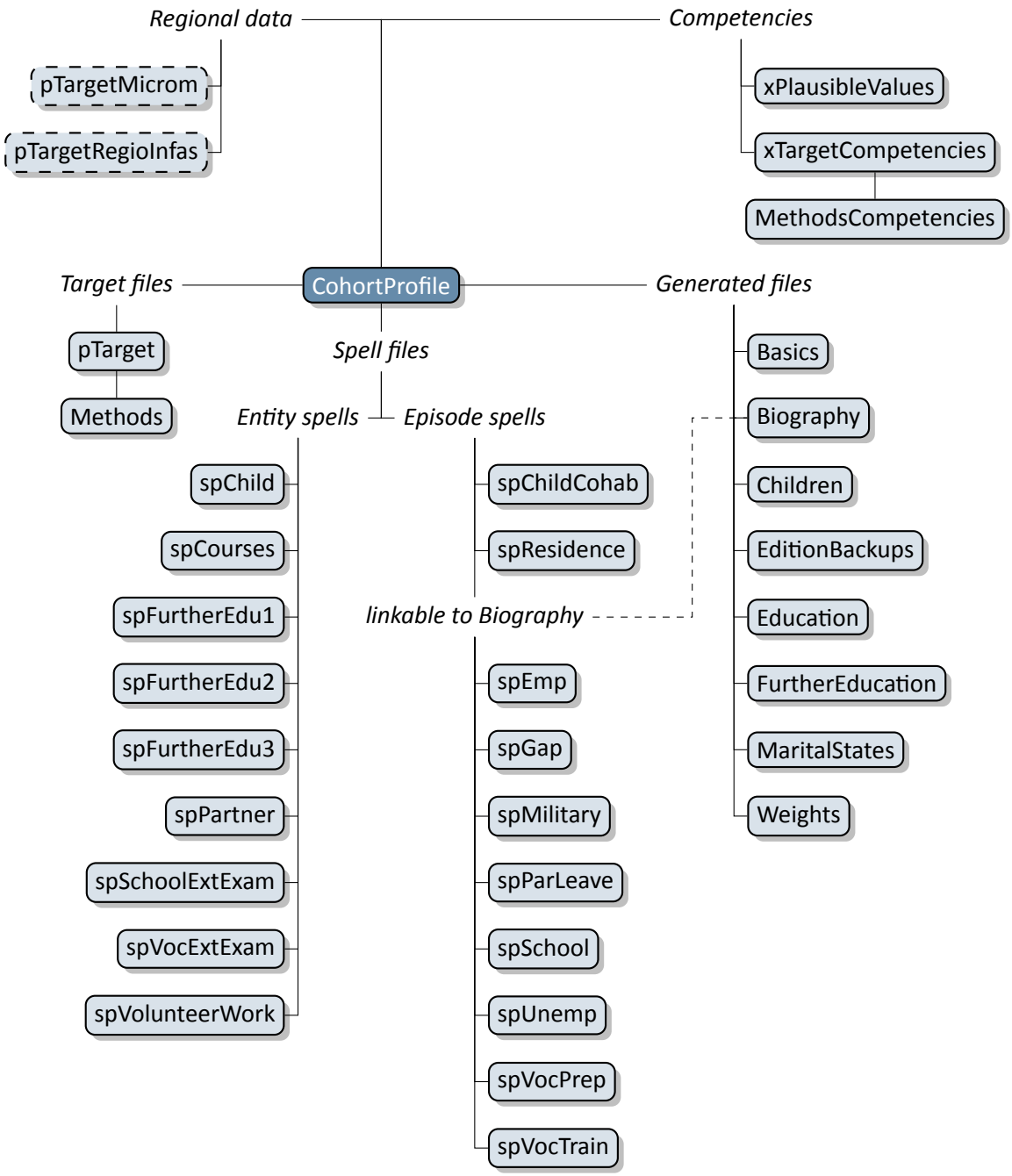

Figure 19: Graphical overview of all data files. Each node represents one data file. Relations are indicated by connection lines. Files with a dashed border are not available in the Download version of the Scientific Use File. Click on a data file to get more information.

4.2.1 Basics

« go back to overview

Description

Compilation of up-to-date information about respondents in a simple data format

File structure

wide format: 1 row = 1 respondent

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

none

Number of variables / number of rows in file

99 / 17,140

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
tx29000	Age at interview month (years)
t70000m	Date of birth: month
t70000y	Date of birth: year
t700001	Gender
tx27000	Family status
tx29003	Mother tongue: German
tx29004	Nationality: German
tx29005	Born in Germany
t741001	Size of Household (persons)
tx20000	Children in household
tx29060	currently employed
tx29904	Main spells of type 'Emp' (number)
tx29007	Age at migration to Germany
t400500_g1	Generation status
t400500_g3	Group of origin

Exemplary data snapshot

ID_t	tx29000	t700001	tx29005	t741001	tx29060	tx29904
8004732	52.67	[m] male	yes	3	yes	5
8006178	32.92	[w] female	yes	2	yes	2
8006267	55.58	[m] male	yes	2	yes	2
8009232	47.83	[w] female	yes	3	yes	5
8010970	68.25	[w] female	yes	2	yes	8

This file contains up-to-date basic information about the respondents, including sociodemographic variables such as age at the time of the interview (`tx29000`), gender (`t700001`), place of birth in Germany (`tx29005`), number of persons in the household (`t741001`), current employment status (`tx29060`), etc. The dataset also contains meta information about certain biographical episodes such as the number of main employment spells (`tx29904`). All information is generated from the `pTarget` file and various spell files. The `Basics` dataset is updated prospectively with each new release of a Scientific Use File. The data structure is cross-sectional reflecting the latest information available on the respondents (which can originate from different survey waves). This simplified structure is intended to give a first impression of the data. However, it should be used with caution as it may not contain the most appropriate information about the respondent. **The main purpose of this file is to get an overview of the data. For analyses, the original panel or spell files should be used!**

Example 1 (Stata): Working with Basics (find R example here)

```
** open the data file
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear

** merge the data from Basics, enhancing every entry in CohortProfile
** (i.e. every wave, this is why m:1 merge is needed)
** with information from Basics
merge m:1 ID_t using ${datapath}/SC6_Basics_D_${version}.dta

** change language to english (defaults to german)
label language en

** tabulate gender by wave
tab wave t700001

** please note that now, you have the most recent information known about respondents
** in every wave. This does not have to be equal to the information actually surveyed
** in that wave!
** Proceed at your own risk!
```

4.2.2 Biography

« go back to overview

Description

Integrated and edited life course data

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t splink

Other ID variables useful for linkage

wave sptype

Number of variables / number of rows in file

10 / 222,046

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
splink	Link for spell merging
wave	Wave
sptype	Spell type
startm	Episode start (month)
starty	Episode start (year)
endm	Episode end (month)
endy	Episode end (year)
spms	Type of event
splast	Episode is ongoing

Exemplary data snapshot

ID_t	splink	wave	sptype	starty	endy
8003125	300002	1	30	1991	1992
8006026	240002	1	24	2003	2007
8009552	260001	2	26	1975	1978
8019916	260002	4	26	2007	2010
8021535	240002	5	24	2012	2012

The file `Biography` serves to facilitate the analysis of complex life course data collected both retrospectively and prospectively. The dataset puts together harmonized episodes with educational and employment relevance from the following duration spell files: `spSchool`, `spVocPrep`, `spVocTrain`, `spMilitary`, `spEmp`, `spUnemp`, `spParLeave`, and `spGap`. The variable `sptype` is provided to identify the source of each episode.

In contrast to the "raw" biographical data from each of the module-specific spell modules, the `Biography` file provides more consistent life course data that has been additionally checked and edited. In particular, inconsistencies in the individual life course data were identified and corrected during the interview with the help of a "check module". Corrected times are stored in the duration spell files as `_g1` variables. For example, the variable `ts2311y_g1` in `spEmp` contains the starting date of an employment episode which was corrected within the check module. Such corrected times form the basis for further adjustments that are implemented in the data editing process for `Biography`. Essentially, the following measures are taken to ensure the integrity of the life course data in this file:

- All subspells have been removed, i. e., `Biography` contains only completed, harmonized, or right-censored episodes (`subspell=0`).
- Episodes revoked by respondents during the interview or in the subsequent survey wave (see section 4.1.4) are deleted, unless the episode was re-recorded in the current wave. Revoked episodes are included in the original spell files and can be identified there with the corresponding marker variables (`spms` or `disagint`).
- Starting and end dates of episodes are smoothed and corrected, i. e., overlaps of one month and more between adjacent episodes have been resolved.
- Gaps between adjacent episodes that do not exceed two months are closed; gaps of more than two months are defined as specific gap episodes (edition gaps) within the `Biography` file.

Due to the additional editing steps and the compilation of spells from different biographical modules, it is recommended to use the `Biography` dataset as a starting point for life course analyses.

Example 2 (Stata): Working with Biography (find R example here)

```
** open the data file
use ${datapath}/${cohort}_Biography_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** check out which spell modules you can merge to this file
tab sptype

** check that you will need splink to merge information
** from other modules to this file
** (command gives no result, which means approval)
isid ID_t splink
```

4.2.3 Children

« go back to overview

Description

Generated dataset regarding the child(ren) of the respondents

File structure

entity format: 1 row = 1 child of 1 respondent

ID variables needed to identify a single row

ID_t child

Other ID variables useful for linkage

none

Number of variables / number of rows in file

9 / 28,375

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
child	Child number
tx27100	Child (biological/adopted/foster)
tx27101	Child: gender
tx2710m	Child: Date of birth (month)
tx2710y	Child: Date of birth (year)
tx27130	lived with a child
tx27131	lives with child (last interview)
tx27199	Child deceased

Exemplary data snapshot

ID_t	child	tx27100	tx27101	tx2710m	tx2710y	tx27130
8004604	2	Biological child	[w] female	May	1991	1
8022093	1	Biological child	[m] male	August	1988	1
8004667	2	Biological child	[w] female	June	1999	1
8011475	1	Biological child	[w] female	September	1978	1
8005716	2	Biological child	[w] female	October	2006	1

The file `Children` simplifies the information available in `spChild`, supplemented by data from `spChildCohab` (cohabitation status). The dataset mainly contains information on the number of children (`child`), the sex of the children (`tx27101`), their date of birth (`tx2710m/y`), and their cohabitation status (`tx27130`). All biological, step, foster and adopted children as well as other children with whom the respondent has ever cohabited are taken into account (see `tx27100`).

Example 3 (Stata): Working with Children (find R example here)

```
** open the data file
use ${datapath}/SC6_Children_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** verify that you will need ID_t and child (child number)
```

```
** to merge information from other modules to this file
** (command gives no result, which means approval)
isid ID_t child

** check distribution of variable child as a child counter
tab child
```

4.2.4 CohortProfile

« go back to overview

Description

Paradata on the cohort's panel sample

File structure

long format: 1 row = 1 respondent in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

15 / 167,762

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
cohort	NEPS Starting Cohort
sample	Sample
tx80220	Participation/drop-out status
tx80521	Data available: Survey Target person
tx80522	Data available: Competency test Target person
inty	Interview date (year)
intm	Interview date (month)
tx80105	NEPS sub samples
tx80107	Sample: First participation in wave

Exemplary data snapshot

ID_t	wave	tx80220	tx80521	tx80522	inty
8000634	3	Participation	yes	yes	2011
8005606	3	Participation	yes	yes	2011
8011813	7	Participation	yes	yes	2014
8022013	5	Participation	yes	yes	2012
8022439	5	Participation	yes	yes	2013

The `CohortProfile` dataset includes all target persons of the panel sample. It applies to all study participants with an initial agreement to take part in the survey. For each respondent in each wave, the `CohortProfile` contains basic information on participation status (`tx80220`), the availability of survey data (`tx80521`), or the availability of competence data (`tx80522`). In addition, there are variables available that indicate when the interview (`intm/y`) and competency testing (`testm/y`) was conducted.

It is strongly recommended to use this data file as a starting point for any analysis!

Example 4 (Stata): Working with CohortProfile (find R example here)

```
** open the data file
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear

** change language to english (defaults to german)
label language en
```

```
** how many different respondents are there?
distinct ID_t

** as you can see, in this file there is an entry for every
** respondent in each wave
tab wave

** check participation status by wave
tab wave tx80220
```

4.2.5 EditionBackups

« go back to overview

Description

Backup of original data that were modified during the data edition process

File structure

long format: 1 row = 1 changed value of a variable in a datafile

ID variables needed to identify a single row

dataset varname ID_t wave splink subspell partner child

Other ID variables useful for linkage

mergevars

Number of variables / number of rows in file

14 / 1,145

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

dataset	Dataset name
varname	Variable name
mergevars	ID-Variables for merging
sourcevalue_num	Original value (if numeric)
editvalue_num	New value (if numeric)
sourcevalue_str	Original value (if string)
editvalue_str	New value (if string)
ID_t	ID target
wave	Wave

Exemplary data snapshot

dataset	varname	mergevars	sourcevalue_num	editvalue_num	ID_t	wave
spSchoolExtExam	ts11302	ID_t wave exam	7.00	4.00	8000511	2
spSchoolExtExam	ts11302	ID_t wave exam	7.00	3.00	8008604	2
spSchoolExtExam	ts11302	ID_t wave exam	7.00	3.00	8011381	2
spVocExtExam	ts15304_v1	ID_t wave exam	22.00	1.00	8012759	1
pTarget	t731306	ID_t wave	5.00	3.00	8022294	4

The dataset `EditionBackups` consists of single values that have been changed or modified in the data edition process. These single values can potentially originate from all other datasets. `EditionBackups` contains both the original and the changed value of a particular variable in a particular data file (i. e., one change or edition per row). The following variables are provided for each change:

- `varname` and `dataset` specify the name of the variable affected by an edition and the respective data file
- `mergevars` lists the identifier variables that are required to merge the information back to the respective data file

- `sourcevalue_[num/str]` contains the original, unaltered value; variables with the suffix `_num` refer to values from numeric variables and variables with the suffix `_str` refer to values from string variables (if the variable is numeric, `_str` is used to store the value label for this value instead)
- `editvalue_[num/str]` contains the result of the modification, i. e. the value into which the original value was changed; these values correspond exactly to the values in the respective data file (again, there is a version for both numeric and string variables - or the label).
- `ID_t`, `wave`, ... are the different identifier variables needed to merge the orginal values to the respective data files

Example 5 (Stata): Working with EditionBackups (find R example here)

```
** In this example, we want to restore the original
** values in variable t520003 (weight in kg) in datafile pTarget

** open the datafile
use ${datapath}/${cohort}_EditionBackups_D_${version}.dta, clear

** only keep rows containing data of the aforesaid variable
keep if dataset=="pTarget" & varname=="t520003"

** check which variables we need for merging
tab mergevars

** then keep the merging variables and the variable with
** the original values (for cross-checking, we also keep the
** variable editvalue, which contains the values found in pTarget)
keep ID_t wave sourcevalue_num editvalue_num

** rename the variables to emphasize affiliation
rename sourcevalue_num t520003_source
rename editvalue_num t520003_edit

** temporary save this data extract
tempfile edition
save `edition'

** open pTarget
use ${datapath}/${cohort}_pTarget_D_${version}.dta, clear

** add the above data
merge 1:1 ID_t wave using `edition', keep(master match)

** check all edition made
list ID_t wave t520003* if _merge==3

** replace the variable in the datafile with its original value
replace t520003=t520003_source if _merge==3
```

4.2.6 Education

« go back to overview

Description

Generated dataset for transitions in educational trajectories

File structure

spell format: 1 row = 1 event (episode) of 1 respondent

ID variables needed to identify a single row

ID_t number

Other ID variables useful for linkage

splink exam tx28100

Number of variables / number of rows in file

11 / 53,454

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
number	Sort number
datem	Valid since (month)
datey	Valid since (year)
tx28101	Recent CASMIN
tx28102	Years of education = f(CASMIN)
tx28103	Recent ISCED-97
tx28109	Change in educational classification
splink	Link for spell merging
exam	Exam number
tx28100	Source of information of educational qualification

Exemplary data snapshot

ID_t	number	datey	tx28101	tx28102	tx28103	splink	tx28100
8004508	2	1978	3	10	2	220002	22
8009350	3	1975	8	18	9	240001	24
8009990	3	2011	8	18	9	240001	24
8007220	2	1989	1	9	1	220002	22
8008112	2	1972	3	10	2	220002	22

The data file Education provides longitudinal information on transitions in the educational careers of respondents. It contains only persons who have completed lower secondary education or higher. To generate the dataset, information on the educational attainment from spSchool (Hauptschule, Realschule, (Fach-)Abitur), spVocPrep (participation in vocational preparation measures) and spVocTrain (all successfully completed trainings) is taken into account. In addition, data from spVocExtExam and spSchoolExtExam were integrated. A total of three measures of educational attainment are available: CASMIN (tx28101), years of education (tx28102, derived from CASMIN), and ISCED-97 (tx28103). The variables splink, exam and tx28100 can be used to merge information from the original spells.

In the Education file, the transitions are stored in a long event time format. This means that each row represents a transition in at least one classification (CASMIN and/or ISCED-97). Since ISCED-97 and CASMIN follow different concepts, some educational transitions are covered by

only one of these two classifications. Months and years for the transition dates are contained in the variables `datem` and `datey`. As a rule, the transitions over time reflect upwards transitions at CASMIN level or up- and sidewards transitions at ISCED-97 level (CASMIN is ordinal, while ISCED-97 has some nominal elements). However, it can also happen that a transition from a higher to a lower degree takes place over time (e. g., by completing a training course after university graduation). In order to determine the highest educational attainment for all respondents, the maximum entry must be selected for each person, for CASMIN in Stata for example by the command:

```
bysort ID_t: egen [varname] = max(tx28101)
```

Example 6 (Stata): Working with Education (find R example here)

```
** we want to merge the school type from spSchool to this datafile.
** For this to work, we first have to prepare spSchool and keep only
** harmonized episodes (subspell==0)
use ${datapath}/SC6_spSchool_D_${version}.dta, clear
label language en
keep if subspell==0
tempfile temp
save `temp'

** now, open the Education data file
use ${datapath}/SC6_Education_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** check out which spell modules you can merge to this file
tab tx28100

** only keep school episodes
keep if tx28100==22

** check that you will need splink to merge information
** from other modules to this file
** (command gives no result, which means approval)
isid ID_t splink, miss

** merge the previously generated temporary data file
merge 1:1 ID_t splink using `temp', keep(master match) keepusing(ts11204)

** see that this only added information to the rows corresponding to spSchool
tab tx28100 _merge
```

4.2.7 FurtherEducation

« go back to overview

Description

Generated dataset for all training courses and further education

File structure

spell format: 1 row = 1 course of 1 respondent

ID variables needed to identify a single row

ID_t number

Other ID variables useful for linkage

wave splink course

Number of variables / number of rows in file

16 / 80,978

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
number	Sort number
wave	Wave
splink	Link for spell merging
course	Course number
tx28200	Origin course information
tx2821y	Course attendance (date/interval) Start (year)
tx2822y	Course attendance (date/interval) End (year)
tx2822c	Course attendance is ongoing
tx28201	Source of course dating
tx28202_R	Course content
tx28204	Course discontinued
tx28203	Course duration in total (hours)

Exemplary data snapshot

ID_t	number	splink	course	tx28200	tx2821y	tx2822y
8004690	4	260007	701	spCourses	2013	2014
8007637	2	260002	202	spCourses	2009	2010
8008327	14	260008	1101	spCourses	2017	2018
8008767	13	260003	1001	spCourses	2016	2017
8024777	10	260008	904	spCourses	2015	2017

Information about the respondents' participation in further education measures is spread across several spell files. The generated file `FurtherEducation` integrates data on courses from the specific datasets `spCourses`, `spFurtherEdu1`, and `spVocTrain` into a consolidated format. These courses are stored there as duration spells in long format. Start and end dates of courses were imputed if the available information was not precise (e.g., *spring*) or missing. Since the third wave (2nd NEPS survey 2010/11), the start and end dates for further courses (`spFurtherEdu1`) are no longer collected. Instead, respondents are asked if they have attended any courses since the last interview. In these cases, the date of the last interview was coded as the start date and the date of the current interview as the end date. This means that the start and end dates here only indicate the time interval in which the course was attended. The variable `tx28201` can be used to see whether the course dates have been asked directly or whether they are derived from interview or episode dates. Information on the content of the courses is available as open answers and in coded form using a classification of the Federal Employment Agency (*Kompetenzkatalog der Bundesagentur für Arbeit*).

All respondents who reported at least one participation in further education are included in FurtherEducation. It should be noted that this file, in contrast to spCourses and spFurtherEdu1, does not only contain course participations from the last year, but also from the previous life course. The latter originate from spVocTrain and are vocational trainings, which can be classified as courses and trainings related to further education. The variable course (course number) allows to link the courses with the original files spCourses, spFurtherEdu1 and spVocTrain. For a subset of courses that have a course number, additional information from spFurtherEdu2 can be added. There is also a second subset of courses that can be linked to spells from spVocTrain or spEmp because they have been reported within the context of these spells or (in case of spells from spVocTrain) because they are derived directly from them. The variables ID_t, course, and splink make it possible to match these original spell data to FurtherEducation. The following overview shows which courses are included in FurtherEducation and with which spells they can be linked in the original files.

- course=valid & splink=missing: episode of further education reported in the further education module; stored in spFurtherEdu1; the spell is right-censored or was completed within the last 12 months
- course=missing & splink=24.... (Vocational Training): episode of vocational training related to further education and participation; stored in spVocTrain; the spell was completed more than 12 months ago
- course=valid & splink=24.... (Vocational Training): episode of vocational training related to further education and participation; stored in spVocTrain; the spell is right-censored or was completed within the last 12 months
- course=valid & splink=25.... (Military/Civilian Service): episode of further education reported in the course module; triggered by spells in spMilitary; courses are stored in spCourses; the triggering spell is right-censored or was completed within the last 12 months
- course=valid & splink=26.... (Employment): episode of further education reported in the course module; triggered by spells in spEmp; courses are stored in spCourses; the triggering spell is right-censored or was completed within the last 12 months
- course=valid & splink=27.... (Unemployment): episode of further education reported in the course module; triggered by spells in spUnemp; courses are stored in spCourses; the triggering spell is right-censored or was completed within the last 12 months
- course=valid & splink=29.... (Parental Leave): episode of further education reported in the course module; triggered by spells in spParLeave; courses are stored in spCourses; the triggering spell is right-censored or was completed within the last 12 months
- course=valid & splink=30.... (Gap): episode of further education reported in the course module; triggered by spells in spGap; courses are stored in spCourses; the triggering spell is right-censored or was completed within the last 12 months

Example 7 (Stata): Working with FurtherEducation (find R example here)

```
** open the data file
use ${datapath}/SC6_FurtherEducation_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** Check the source module of contained courses
tab tx28200
```

4.2.8 MaritalStates

« go back to overview

Description

Generated dataset for all marital states

File structure

entity format: 1 row = 1 marital state of 1 respondent

ID variables needed to identify a single row

ID_t number

Other ID variables useful for linkage

none

Number of variables / number of rows in file

6 / 19,122

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
number	Sort number
datem	Change since (month)
datey	Change since (year)
tx27000	Family status
problem	Problematic case (information)

Exemplary data snapshot

ID_t	number	datem	datey	tx27000
8023645	2	October	2007	divorced
8004340	1	February	1992	married/in registered civil partnership
8002686	1	December	2008	married/in registered civil partnership
8024816	4	August	2010	widowed
8022698	1	June	2002	married/in registered civil partnership

The generated file `MaritalStates` is derived from information in `spPartner` and lists all marital states with their entry date. Only persons who are or were married are included in this file. There is an auxiliary variable `problem` that marks and documents problematic cases (e. g., when a divorce is reported before marriage).

Example 8 (Stata): Working with MaritalStates (find R example here)

```
** open the data file
use ${datapath}/SC6_MaritalStates_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** Look at the distribution of family status
tab tx27000
```

4.2.9 Methods

« go back to overview

Description

Paradata from the CATI/CAPI interviews of the target persons

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

ID_int

Number of variables / number of rows in file

49 / 127,443

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
inty	Interview date - year
intm	Interview date - month
tx80107	Sample: First participation in wave
tx80209	Interview: Length of interview (minutes)
tx80210	Interview: Incentive (euro)
ID_int	Interviewer: ID
tx80301	Interviewer: Gender
tx80302	Interviewer: Age group
tx80400	Willingness: Panel participation
tx80207	Interview: Response code differentiated (final outcome)
tx80101	Sample: Federal state

Exemplary data snapshot

ID_t	wave	inty	intm	tx80209	ID_int	tx80301	tx80302
8001190	7	2014	12	50.30	2284	[m] male	50-65 years
8006576	2	2010	3	41.65	1210	[w] female	50-65 years
8009457	4	2011	10	31.47	1117	[w] female	30-49 years
8011225	3	2010	11	93.63	1321	[m] male	older than 65 years
8020830	4	2011	11	60.35	1172	[m] male	older than 65 years

This dataset provides a variety of information about data collection such as gender (`tx80301`) and age (`tx80302`) of the interviewer, the interview date (`intm`, `inty`), the interview duration (`tx80209`), and the individual survey participation status (`tx80220`).

It should be noted that `Methods` contains all respondents contacted, regardless of whether an interview was conducted or not (see variable `tx80207` for more details). For this reason, the data file `Methods` consists of more cases than the file `pTarget`.

Example 9 (Stata): Working with Methods (find R example here)

```
** open the data file
use ${datapath}/SC6_Methods_D_${version}.dta, clear

** change language to english (defaults to german)
label language en
```

```
** check out participation status by wave
tab wave tx80220

** how many different interviewers did CATI surveys?
distinct ID_int

** create one single variable containing the interview date
generate intdate=mdy(intm,intd,inty)
format intdate %td
list intd intm inty intdate in 1/10
```

4.2.10 MethodsCompetencies

« go back to overview

Description

Paradata of the realization of competence measures

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

89 / 33,288

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
splitgr	Split group
iiw	Interview location
tlesz1_2	Start Timekeeping Reading
tlesz2_2	End Timekeeping Reading
tmlintro_2	Start Metacognition Reading
tmlz2_2	End Metacognition Reading
tmatz1_2	Start Timekeeping Numeracy
tmatz2_2	End Timekeeping Numeracy
tbtdauer	Duration TBT-module in sec
tictstart	Start Timekeeping Task booklet ICT
tictend	End Timekeeping Task booklet ICT
tbt_skip	Participation in test refused

Exemplary data snapshot

ID_t	wave	splitgr	iiw	tlesz2_2	tbtdauer
8002914	3	only booklet Reading	.	1	-54
8003127	3	only booklet Numeracy	1	.	-54
8004316	3	both tests, booklet Numeracy first	2	.	-54
8009224	3	both tests, booklet Reading first	.	1	-54
8011044	3	both tests, booklet Reading first	.	1	-54

Analogous to other method files, this dataset also contains paradata about the interview situation, in particular about the realization of the competence tests. Available variables include sample splits (`splitgr`), interview location (`iiw`) and different start and end markers for different modules (e. g., reading, ICT).

Example 10 (Stata): Working with MethodsCompetencies (find R example here)

```
** open the data file
use ${datapath}/SC6_MethodsCompetencies_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** look at the distribution of split groups
** note that this has only been conducted in wave 3
tab splitgr wave
```

4.2.11 pTarget

« go back to overview

Description

Data from respondents CATI/CAPI interviews

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

splink

Number of variables / number of rows in file

1,126 / 107,682

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
tx20000	Number of children in household
t700101_g1	Municipality of birth (West-/East)
t34005a	Number of books
t700001	Gender
t70000y	Date of birth: year
t405010_g2	Country of birthplace (categorized)
tx20000	Number of children in household
t514001	Satisfaction with life
t733001	Current family status
t400500_g1	Generation status
t514004	Satisfaction with family life
t66003a_g1	Global self-esteem
t66800a_g1	Big Five: Extraversion
t66800b_g1	Big Five: Agreeableness

Exemplary data snapshot

ID_t	wave	tx20000	t700001	t70000y	t405010_g2	t514001	t400500_g1	t514004
8008309	2	4	[w] female	1970	9	9	1	8
8020378	4	2	[w] female	1983	3	6	2	10
8020905	4	2	[w] female	1978	1	9	3	10
8024234	4	1	[m] male	1960	3	7	1	3
8024638	4	1	[m] male	1976	6	10	1	10

The data in the file `pTarget` comes from computer-assisted telephone (CATI) or personal (CAPI) interviews. Since several questions are asked repeatedly over different survey waves, data integration takes place in a long format. This means that for each new survey wave there is an additional row for each target participating in that wave. Target persons can be uniquely identified by the variable `ID_t`, but rows can only be identified by the combination of the variables `ID_t` and `wave`. Since rows exist only for those respondents for whom answers from the respective survey wave are available, there are fewer rows in `pTarget` than in the `CohortProfile`.[2]

2 The `CohortProfile` contains all respondents in the panel sample, regardless of their participation in any wave.

The dataset pTarget provides hundreds of variables and thus contains most of the information collected. Some of the variables describe sociodemographic characteristics such as gender (t700001), year of birth (t70000y), country of birth (t405010_g2), or generation status (t400500_g1). Other variables contain information on the household context such as the number of children (tx20000) or subjective assessments such as satisfaction with life (t514001) or family life (t514004).

Example 11 (Stata): Working with pTarget (find R example here)

```
** open the CohortProfile
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear

** merge some variable from pTarget
merge 1:1 ID_t wave using ${datapath}/SC6_pTarget_D_${version}.dta, ///
  keepusing(t400500_g1 t733001) nogen // assert(master match)

** note that this information in now available only in waves which have
** surveyed the topic
tab wave t400500_g1

** if it makes sense, you can copy this information to cells of other waves.
** This copies information downwards (i.e., to later waves), unless a new
** value has been reported (which is usually what you want in a panel study)
bysort ID_t (wave): replace t400500_g1=t400500_g1[_n-1] ///
        if t400500_g1==-54 | missing(t400500_g1)

tab wave t400500_g1
```

4.2.12 pTargetMicrom

« go back to overview

Description

Small-scale regional indicators on respondents' place of residence

File structure

panel format: 1 row = 1 regional level in 1 wave of 1 respondent

ID variables needed to identify a single row

ID_t wave regio

Other ID variables useful for linkage

ID_regio

Number of variables / number of rows in file

189 / 146,682

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
regio	Indicator for enrichment level
ID_regio	System-free ID of enrichment level
mso_k_ausland	Share foreigners
mso_k_familie	Family structure
mbe_k_haustyp	Type of house
mgm_k_dom	Dominant microm geo milieu®
mgs_k_dom	Dominant geo-submilieu
mmo_k_volumen	Move volume
mpi_k_dichte	Car density
mas_k_berufsuv	Occupational disability insurance
mas_k_krankzuv	Additional health insurance
mlt_k_primlt	Primary Limbic Type
kkr_w_summe	Total purchasing power in euro

Exemplary data snapshot

ID_t	wave	regio	ID_regio	mso_k_ausland	mbe_k_haustyp	mpi_k_dichte
8012602	6	1	147356	6	4	4
8012603	6	1	159404	1	4	6
8021446	6	1	142897	1	2	9
8021511	5	1	154963	5	3	7
8022624	6	1	147968	8	1	7

The data file `pTargetMicrom` is only available via **On-site** access. The file is not included in the Download and Remote versions of the Scientific Use File.

The data include details about the respondent's residence at five different regional levels, distinguishable by the variable `regio`: house area, street section, postal code, postal code 8, municipality. All these levels are available for each respondent and each wave. Numerous regional indicators are provided, e. g. the percentage of foreigners, unemployment rate, family and age structure, milieu types, car type density, distribution of insurances, etc. To clarify, this information does **not** refer to individuals, but to regional units to which respondents belong via their

place of residence. Accordingly, the unemployment rate, for example, indicates the proportion of unemployed people in the population of a given region.

Please note that a separate documentation exists for this data file on the website (see section 1.2), which not only lists all variables, but also explains the background of the data.

Example 12 (Stata): Working with pTargetMicrom (find R example here)

```
** open Microm datafile. Note that this data file is only available OnSite!
use ${datapath}/${cohort}_pTargetMicrom_O_${version}.dta, clear
label language en

** additionally to ID_t and wave, line identification in this file is done
** via variable regio, denoting the regional level of information
isid ID_t wave regio

** tabulating wave against regio shows availability of all levels
** in wave 5 and 7, but only the most detailled level available
** in wave 1 and 3 (usually housing level)
tab wave regio

** only keep housing level
keep if regio==1

** now you can enhance CohortProfile with regional data
merge 1:1 ID_t wave using ${datapath}/${cohort}_CohortProfile_O_${version}.dta
```

4.2.13 pTargetRegioInfas

« go back to overview

Description

Small-scale regional indicators on respondents' place of residence

File structure

panel format: 1 row = 1 regional level of 1 respondent

ID variables needed to identify a single row

ID_t regio

Other ID variables useful for linkage

none

Number of variables / number of rows in file

67 / 46,596

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
regio	Regional level
tx44288	Share residents 0-14 years (in %)
tx44289	Share residents 15-24 years (in %)
tx44294	Purchasing power per resident (EUR)
tx44298	Companies in total per km² (trade indicator)
tx44302	Share retail (in %)
tx44001	Residents per household
tx44318	Share single-person households (in %)
tx44242	Type of residential area
tx44312	Share agriculture (in %)
tx44354	Share residential type post-war apartment complex (in %)

Exemplary data snapshot

ID_t	regio	tx44294	tx44298	tx44001
8008720	3	14543	2.80	2.65
8011308	2	15472	3.18	2.36
8012615	2	22022	34.95	2.04
8005465	2	14746	4.35	2.53
8011150	2	19212	173.81	1.78

The data file pTargetRegioInfas is only available via **On-site** access. The file is not included in the Download and Remote versions of the Scientific Use File.

The data include details about the respondent's residence at four different regional levels, distinguishable by the variable regio: street section, quarter, postal code, and municipality. Information on all these levels is only available for the second wave (1st NEPS survey, 2009/2010). The regional indicators available in this file include the purchasing power per resident in EUR (tx44294), the total number of companies per km^2 (tx44298), the average number of residents per household (tx44001), and so on. As in pTargetMicrom these data do **not** refer to the respondents themselves, but to the regional levels in which the respondents live (i. e., the unemployment rate, for example, indicates the proportion of unemployed people in the population of a given region such as the municipiality).

Please note that a separate documentation exists for this data file on the website (see section 1.2), which not only lists all variables, but also explains the background of the data.

Example 13 (Stata): Working with pTargetRegioInfas (find R example here)

```
** open datafile. Note that this data file is only available OnSite!
use ${datapath}/${cohort}_pTargetRegioInfas_0_${version}.dta, clear
label language en

** identification in this file is done
** via variable regio, denoting the regional level of information
isid ID_t regio

** existing regional levels are:
tab regio

** only keep housing level
keep if regio==1


** save to temporary file
tempfile regio
save `regio'

** now you can enhance CohortProfile with regional data
use ${datapath}/${cohort}_CohortProfile_0_${version}.dta, clear
label language en

merge 1:1 ID_t wave using `regio'
```

4.2.14 spChild

« go back to overview

Description

Spell data on all children of the respondents

File structure

entity format: 1 row = 1 child of 1 respondent

ID variables needed to identify a single row

ID_t child subspell

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

53 / 186,863

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
child	Child number
subspell	Number of subspell
wave	Wave
ts3320m	Date of birth of child (month)
ts3320y	Date of birth of child (year)
ts33203	Gender of the child
ts33204	Biological, adoptive or foster child
ts33209	Employment Child
ts33301	School enrollment already completed
ts33216	Vocational training Child
ts33101	Own children
ts33306	Children in household
ts33202	Auxiliary variable: Age of the child

Exemplary data snapshot

ID_t	child	subspell	wave	ts3320y	ts33203	ts33204
8010985	3	1	2	1993	[m] male	Biological child
8012784	2	1	1	1998	[m] male	Biological child
8020445	1	1	4	1967	[m] male	Biological child
8021000	1	1	4	1979	[m] male	Biological child
8022403	3	1	4	2002	[w] female	Biological child

The data set spChild informs about all biological, foster and adopted children of a respondent as well as about every other child that currently lives or has lived with the respondent (e. g., children of former and current partners). For the latter, episodes were only recorded if the interviewee and the child lived in the same household. The variable ts33204 can be used to distinguish the child type. In the case of twins and higher orders of multiple births, separate episodes are generated for each child. The variable child counts up the children per respondent. Note that a child episode was skipped in the interview when the respondent reported that the child was deceased. In addition to sociodemographic characteristics such as year of birth (ts3320y) and gender (ts33203), the data mainly contain educational and employment-related information on the children.

Spell data on living with children can be found in the file spChildCohab; spell data on parental leave related to the children are stored in the file spParLeave.

Example 14 (Stata): Working with spChild (find R example here)

```
** open the data file
use ${datapath}/SC6_spChild_D_${version}.dta, clear

** switch to english language
label language en

** only keep full or harmonized episodes
keep if subspell==0

** generate the total count of children for each respondent
** you can do this either by taking the maximum child number:
bysort ID_t: egen children=max(child)
** or counting the number of rows:
bysort ID_t: gen children2=_N
** which both computes the same result
assert children==children2

** recode rough values (e.g. end of year) to real months
replace ts3320m=ts3320m-20 if ts3320m>20

** compute the age of one's children today
** first, create a Stata monthly date (months since 1960m1) of the birth variables
generate birth_ym =ym(ts3320y,ts3320m)
** then, create the same for the current date
gen now_ym=mofd(date(c(current_date), "DMY"))
** the age is then easily computed
gen age=(now_ym-birth_ym)/12

summarize age
```

4.2.15 spChildCohab

« go back to overview

Description

Spell data on all episodes of cohabitation with children

File structure

spell format: 1 row = 1 cohabitation episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

child wave

Number of variables / number of rows in file

23 / 118,785

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
child	Child number
spell	Spell number cohabitation with child
subspell	Number of subspell
wave	Wave
ts33203	Gender of the child
ts3331m	Start date Living together Child (month)
ts3331y	Start date Living together Child (year)
ts3332m	End date Living together Child (month)
ts3332y	End date Living together Child (year)
ts3332c	Currently living together with child
ts33308	Episode update Living together with child

Exemplary data snapshot

ID_t	child	spell	subspell	wave	ts3331y	ts3332y
8002099	5	508	4	11	2015	2018
8002969	1	102	7	7	1995	2013
8008140	3	303	1	6	2013	2013
8022555	2	203	3	11	2015	2018
8024020	3	302	1	4	2001	2012

If a respondent lives together with one or more children in a household, the duration of the cohabitation is registered in `spChildCohab`. Cohabitation episodes are connected to the respective child via the number in the variable `child`. Please note that the periods of cohabitation from the year of the beginning (`ts3331y`) to the year of the end (`ts3332y`) do not necessarily coincide with the dates of birth and death; for direct information on the children rather consult the `spChild` dataset.

Example 15 (Stata): Working with spChildCohab (find R example here)

```
** open the data file
use ${datapath}/SC6_spChildCohab_D_${version}.dta, clear
```

```
** switch to english language
label language en

** only keep full or harmonized episodes
keep if subspell==0

** recode rough values (e.g. end of year) to real months
replace ts3331m=ts3331m-20 if ts3331m>20
replace ts3332m=ts3332m-20 if ts3332m>20

** generate the following durations in months:
* a) the total duration of a cohabitation episode
gen cohab_duration = ym(ts3332y,ts3332m) - ym( ts3331y, ts3331m)
* b) the total duration a respondent lived together with specific child
bysort ID_t child (spell): egen total_duration_per_child  = total(cohab_duration)
* c) the total duration a respondent lived together with any child
bysort ID_t (child spell): egen total_duration_per_target = total(cohab_duration)

** to work with the latter information in other files, you could do
** which gives you a cross-sectional display of cohabitation time for every
  respondent
keep ID_t total_duration_per_target
duplicates drop
```

4.2.16 spCourses

« go back to overview

Description

Spell data on courses and trainings

File structure

entity format: 1 row = 1 episode with one or more courses of 1 respondent

ID variables needed to identify a single row

ID_t wave splink

Other ID variables useful for linkage

sptype course_w*

Number of variables / number of rows in file

69 / 29,409

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
splink	Link for spell merging
sptype	Spell type
t271001	Total duration of training courses
course_w1	1st course number
t271011_w1	Course duration
course_w2	2nd course number
t271011_w2	Course duration
course_w3	3rd course number
t27800a	Start date episode (month)
t27800b	Start date episode (end)
t27800c	End date Episode (month)
t27800d	End date Episode (year)

Exemplary data snapshot

ID_t	wave	splink	sptype	course_w1	course_w2	course_w3
8001797	11	260003	26	1101	1102	1103
8004840	7	260001	26	701	702	703
8005278	10	260008	26	1001	1002	1003
8020955	7	260002	26	701	702	703
8021423	11	260003	26	1101	1102	1103

The file spCourses indicates courses and trainings attended since the last interview (or within the last 12 months to the first interview) during episodes of employment (spEmp), unemployment (spUnemp), parental leave (spParLeave), military or civilian service (spMilitary), and episodes from the spGap module. The start and end dates of the spells correspond to the original episodes from the modules just mentioned, in which a course was taken. For each of these episodes, information on up to five (up to three until wave 9) courses is included in a wide data format. The dataset covers all course spells that were recorded in these modules (see sptype for identification). Spells may also be included if no course was taken during this episode. The only criterion for inclusion in spCourses is that a person has provided information about at least one course. Note that the course numbers in this dataset are stored in wide format (course_w1, ..., course_w5), while in the other course files (spFurtherEdu1, FurtherspEdu2) there is only one single enumerator (course).

Basic information from this dataset is integrated into the generated file FurtherEducation. If

you are not necessarily interested in the details from `spCourses`, we recommend using `FurtherEducation` instead.

Example 16 (Stata): Working with spCourses (find R example here)

```
** open the data file
use ${datapath}/SC6_spCourses_D_${version}.dta, clear

** check which modules provided course information
tab sptype

** only keep courses from employment spells
keep if sptype==26

** save this datafile for later usage
tempfile courses
save `courses'

** open the employment module
use ${datapath}/SC6_spEmp_D_${version}.dta, clear

** add the temporary datafile from above;
** note that this is an m:1 merge, as there are still subspells in spEmp
merge m:1 ID_t wave splink using `courses', assert(master match) nogenerate

** you now have the spEmp datafile, enhanced with information from spCourses,
** and can proceed with this in the usual way
```

4.2.17 spEmp

« go back to overview

Description

Spell data on employment episodes

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

splink wave

Number of variables / number of rows in file

145 / 185,028

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
wave	Wave
ts23222	In partial retirement, (active phase)
ts2311y	Start Employment episode (year)
ts2312y	End Employment episode (year)
ts23410	Net income, open
ts23228	Type of required vocational training
ts23223_g1	Actual weekly working time, currently/at the end
ts23201_g1	Professional title (KldB 1988)
ts23201_g2	Professional title (KldB 2010)
ts23201_g3	Professional title (ISCO-88)

Exemplary data snapshot

ID_t	subspell	spell	ts2311y	ts2312y	ts23410	ts23228	ts23223_g1
8001651	4	2	1996	2011	389	1	11.00
8005894	1	5	2010	2010	2000	2	38.00
8006510	3	2	2008	2011	1795	6	60.00
8019836	1	1	1979	2012	2000	4	42.00
8024601	1	3	2011	2012	390	1	3.00

The comprehensive dataset spEmp covers all episodes of the respondents' regular employment, also traineeships. Information on second jobs is only collected for activities that are ongoing at the date of the interview. Vacation jobs, volunteering, and internships are not included. New episodes are created at the following events:

- Change of employer
- Change of occupation
- Interruption of employment (e. g., due to unemployment or military service)

The file provides information about the start and end dates of each episode (`ts2311y`, `ts2312y`), as well as net income (`ts23410`), type of required vocational training (`ts23228`), actual working time per week (`ts23223_g1`), and so on.

Example 17 (Stata): Working with spEmp (find R example here)

```
** open the data file
use ${datapath}/SC6_spEmp_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.18 spFurtherEdu1

« go back to overview

Description

Spell data on additional courses

File structure

entity format: 1 row = 1 course of 1 respondent

ID variables needed to identify a single row

ID_t course

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

19 / 23,504

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
nepswave	NEPS wave
course	Course number
t271044	Start of course (month)
t271045	Start of course (year)
t271047	End of course (month)
t271046	End of course (year)
t271048	Course is ongoing
t271049	Termination Course
t272000_R	Course content other course
t272000_g13	Course content other course (course ID)
t271043	Duration of course

Exemplary data snapshot

ID_t	wave	course	t271045	t271046	t271048
8002574	2	204	2009	2010	yes, course is ongoing
8005087	2	201	2009	2009	no
8006537	2	203	2009	2009	no
8009677	2	201	2008	2009	no
8012191	2	201	2009	2009	no

The data set `spFurtherEdu1` contains information about other courses that the respondent has attended since the last interview (in the first interview within the last 12 months) and has not reported in `spCourses` or `spVocTrain`. This includes both professional trainings (similar to `spCourses`) as well as courses for private purposes (e. g., cooking course, yoga course, NLP coaching). In addition to the content of the respective course, the start date (`t271045`) and end date (`t271046`) as well as the current status (`t271048`) are available.

Information from this dataset is integrated into the generated file `FurtherEducation`. If you are not necessarily interested in the details from `spFurtherEdu1`, we recommend using `FurtherEducation` instead.

Example 18 (Stata): Working with spFurtherEdu1 (find R example here)

```
** open the datafile
use ${datapath}/SC6_spFurtherEdu1_D_${version}.dta, clear

** One row contains information for one course. The only possibility to use
```

```
** this file is to merge it to the data for this respondents wave (we use the
** CohortProfile). We have to reshape the file so one row contains one wave.
bysort ID_t wave (course): gen course_nr=_n
reshape wide course t*, i(ID_t wave) j(course_nr)

** create a temporary datafile for later merge
tempfile spfurther
save `spfurther'

** open CohortProfile
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear

** merge data
merge 1:1 ID_t wave using `spfurther', assert(master match) nogen

** Please note that you now have multiple variables added to CohortProfile,
** one set of variables for each course reported in spFurtherEdu1
```

4.2.19 spFurtherEdu2

« go back to overview

Description

Spell data with additional information on selected courses

File structure

entity format: 1 row = 1 course of 1 respondent

ID variables needed to identify a single row

ID_t course

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

40 / 50,099

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
course	Course number
t323520	Care support through social capital
t279040	Professional/private reasons
t279046	Course costs Employer
t272040	Provider
t279041	Motivation for course attendance
t272043	Certificate
t272003	Course assessment: learned new things
t274022	Course assessment: Instructor patient
t279047	Course costs Employment agency

Exemplary data snapshot

ID_t	wave	course	t279046	t279041	t272043
8004415	4	401	fully	some effort	1
8009414	2	202	fully	a lot of effort	1
8010677	5	501	fully	a lot of effort	1
8015512	8	801	fully	a lot of effort	3
8024794	8	803	not at all	a lot of effort	1

Using the survey instrument, two courses from the modules `spVocTrain`, `spCourses` and `spFurtherEdu1` are randomly selected. For both courses the respondent is asked to provide additional information such as costs incurred by the employer (`t279046`), motivation for course attendance (`t279041`) and certificates (`t272043`). This information is contained in the dataset `spFurtherEdu2`.

Example 19 (Stata): Working with spFurtherEdu2 (find R example here)

```
** Two possibilities to use spFurtherEdu2

** A) Merge data to spCourses

** open spCourses datafile
use ${datapath}/SC6_spCourses_D_${version}.dta, clear
```

```
** one row contains information for up to three courses.
** To make merging possible, you first have to reshape the datafile
** so one row contains only one course
reshape long course_w, i(ID_t wave splink) j(course_nr)
rename course_w course

** merge spFurtherEdu2 using ID_t and course
merge m:1 ID_t course using ${datapath}/SC6_spFurtherEdu2_D_${version}.dta, keep(
  master match)

** ----
** B) merge to spFurtherEdu1

** open spFurtherEdu1 datafile
use ${datapath}/SC6_spFurtherEdu1_D_${version}.dta, clear

** merge spFurtherEdu2 using ID_t and course
merge 1:1 ID_t course using ${datapath}/SC6_spFurtherEdu2_D_${version}.dta, keep(
  master match)
```

4.2.20 spFurtherEdu3

« go back to overview

Description

Spell data on German language courses

File structure

entity format: 1 row = 1 German course of 1 respondent

ID variables needed to identify a single row

ID_t gcourse

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

11 / 512

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
gcourse	Course number (german course)
wave	Wave
t416510	Duration of course
t41652m	Start of German course (month)
t41652y	Start of German course (year)
t41653m	End of German course (month)
t41653y	End of German course (year)
t416540	Termination Course
nepswave	NEPS wave
t416500	Attended German course 1

Exemplary data snapshot

ID_t	gcourse	wave	t41652m	t41652y	t41653m	t41653y
8001378	201	2	December	1992	October	1993
8001873	201	2	February	1997	August	1997
8020360	401	4	October	2011	December	2011
8023996	401	4	November	1990	November	1991
8024809	401	4	October	2000	July	2001

Information on courses in German as a foreign language is only collected for migrants. The dataset `spFurtherEdu3` lists the start date (`t41652m/y`), the end date (`t41653m/y`) and the duration of German courses attended by respondents with migration background.

Example 20 (Stata): Working with spFurtherEdu3 (find R example here)

```
** Two possibilities to use spFurtherEdu3

** A) Merge data to spCourses

** open spCourses datafile
use ${datapath}/SC6_spCourses_D_${version}.dta, clear

** one row contains information for up to three courses.
** To make merging possible, you first have to reshape the datafile
** so one row contains only one course
```

```
reshape long course_w, i(ID_t wave splink) j(course_nr)
rename course_w gcourse

** merge spFurtherEdu3 using ID_t and gcourse
merge m:1 ID_t gcourse using ${datapath}/SC6_spFurtherEdu3_D_${version}.dta, keep(
  master match)

** ----
** B) merge to spFurtherEdu1

** open spFurtherEdu1 datafile
use ${datapath}/SC6_spFurtherEdu1_D_${version}.dta, clear

** rename course variable to match variable name in spFurtherEdu3
rename course gcourse

** merge spFurtherEdu3 using ID_t and course
merge 1:1 ID_t gcourse using ${datapath}/SC6_spFurtherEdu3_D_${version}.dta, keep(
  master match)
```

4.2.21 spGap

« go back to overview

Description

Spell data on reported gap episodes

File structure

spell format: 1 row = 1 gap of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

29 / 39,850

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
splink	Link for spell merging
spell	Spell number
subspell	Number of subspell
wave	Wave
spms	Check module: Type of event (edited)
ts29901	Auxiliary variable Current gap
ts29300	Episode mode
ts2911m	Start date Gap (month)
ts2911y	Start date Gap (year)
ts2912m	Start date gap (month)
ts2912y	Start date gap (year)
ts2912c	Gap ongoing
ts29201	Training course during gap
ts29101	Type of gap

Exemplary data snapshot

ID_t	spell	subspell	wave	ts29901	ts2911y	ts2912y
8000346	2	2	3	1	1978	2010
8000881	2	2	3	1	1976	2011
8004518	1	1	2	1	2008	2010
8009405	1	1	2	1	2008	2010
8024969	2	1	4	1	2009	2011

Gaps in the individual life histories are identified by a “check module”. Such gap episodes are contained in the file `spGap` with start dates (`ts2911m/y`) and end dates (`ts2912m/y`). The spells here refer to different types of gaps that are indicated by the variable `ts29101`.

Example 21 (Stata): Working with spGap (find R example here)

```
** open the data file
use ${datapath}/SC6_spGap_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
```

```
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.22 spMilitary

« go back to overview

Description

Spell data on military or civilian service and years of voluntary work

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

25 / 6,590

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
wave	Wave
ts21201	Type of military service episode
ts2111m	Start Military service episode - Month
ts2111y	Start Military service episode - Year
ts2112m	End Military service episode - month
ts2112y	End Military service episode - year
ts21202	Attendance of training courses/courses during military service
ts2111y_g1	Check module: start date (year, edited)
ts2112y_g1	Check module: end date (year, edited)

Exemplary data snapshot

ID_t	splink	subspell	spell	wave	ts2111y	ts2112y
8001275	250001	1	1	2	2009	2010
8001799	250001	2	1	2	2007	2008
8010065	250002	3	2	10	2015	2017
8019886	250002	6	2	9	1995	2016
8020559	250001	5	1	8	2000	2015

The dataset `spMilitary` contains episodes of military or civilian service as well as years used for voluntary work in the social or environmental sector with respective start dates (`ts2111m/y`) and end dates (`ts2112m/y`). Regular or professional soldiers are regarded as employed and are therefore more likely to be found in the employment file `spEmp`.

Example 22 (Stata): Working with spMilitary (find R example here)

```
** open the data file
use ${datapath}/SC6_spMilitary_D_${version}.dta, clear
```

```
** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.23 spParLeave

« go back to overview

Description

Spell data on periodes of parental leave

File structure

spell format: 1 row = 1 parental leave episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave child splink

Number of variables / number of rows in file

31 / 12,548

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
child	Child number
spell	Spell number
subspell	Number of subspell
wave	Wave
ts2711m	Start Parental leave (month)
ts2711y	Start Parental leave (year)
ts2712m	End Parental leave (month)
ts2712y	End Parental leave (year)
ts2712c	Ongoing of parental leave
th27100	Employment after parental leave

Exemplary data snapshot

ID_t	child	spell	subspell	wave	ts2711y	ts2712y
8006625	2	202	1	5	2011	2012
8006930	1	102	3	4	1996	2011
8006935	2	204	2	5	2011	2012
8020873	2	202	1	8	2015	2015
8022970	2	201	3	6	2011	2013

For each child (except for deceased children, see `spChild`), information is collected on whether the respondent has taken parental leave. Each parental leave episode adds one row to the dataset `spParLeave`, including information on the beginning of the leave (`ts2711m/y`) and its end (`ts2712m/y`). According to the study design, periods of maternity leave do not count as parental leave. These periods are usually added to the respective employment episode. This means that an employment spell is not interrupted if the mother only takes maternity leave without additional parental leave.

Example 23 (Stata): Working with spParLeave (find R example here)

```
** open the data file
use ${datapath}/SC6_spParLeave_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0
```

```
** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.24 spPartner

« go back to overview

Description

Spell data on partners in the household

File structure

entity format: 1 row = 1 partner of 1 respondent

ID variables needed to identify a single row

ID_t partner subspell

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

100 / 104,147

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
partner	Partner number
subspell	Number of subspell
ts31204	Partner born in Germany/abroad
ts31211	Partner German
ts31203	Gender of partner
ts3131y	Date when couple moved in together (year)
ts3141m	Marriage date (month)
ts3141y	Marriage date (year)
ts3120m	Date of birth Partner (month)
ts3120y	Date of birth Partner (year)
ts31206	Age at moving Partner
ts31207	Place of birth Father Partner
ts31209	Place of birth Mother Partner

Exemplary data snapshot

ID_t	partner	subspell	ts31203	ts3120m	ts3120y
8009364	1	1	[w] female	September	1962
8023604	1	1	[m] male	August	1984
8019993	2	1	[w] female	May	1947
8005961	1	1	[w] female	December	1975
8006454	1	1	[w] female	September	1970

The dataset `spPartner` covers the respondent's partnership history. The subjective reports of the respondents define whether they live in a relationship and whether they cohabit with their partner or not. A comprehensive set of additional questions refers to the current partner, including gender (`ts31203`) and date of birth (`ts3120m/y`). For former partners, only information about the year of birth and education is available. Information about the current partner is collected regardless of the status of cohabitation, while former partners are only included in the survey if they have lived together with the respondent. The enumerator variable `partner` identifies partners *within* respondents. This variable is coded with 1 for the first partner and counts up to the last (current) partner.

Example 24 (Stata): Working with spPartner (find R example here)

```
** open the data file
use ${datapath}/SC6_spPartner_D_${version}.dta, clear

** switch to english language
```

```
label language en

** only keep full or harmonized episodes
keep if subspell==0

** to find out if a respondent is or was ever been married,
** check out if the indicating variable ever stated a marriage
bysort ID_t: egen married = max(ts31410==1)

** look at the data
list ID_t partner ts31410 married in 1/20, sepby(ID_t)

** reduce the datafile, so you have one single row for each respondent
keep ID_t married
duplicates drop

** you now can save this datafile and merge it to, e.g., CohortProfile
tempfile married
save `married'
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear
merge m:1 ID_t using `married', nogen keep(master match)
```

4.2.25 spResidence

« go back to overview

Description

Spell data on the residential history

File structure

spell format: 1 row = 1 residential episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave th21111_*

Number of variables / number of rows in file

25 / 69,217

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
subspell	Number of subspell
spell	Spell number
spgen	Generated spell
wave	Wave
th21102	Place of residence in Germany/ Abroad
th21103_g2	Country of place of residence (categorized)
th21106	Start date place of residence episode (year)
th21108	End date place of residence episode (year)
th21111_g2	Place of residence (RS Federal State)
th21109	Current place of residence
th21111_g1	Place of residence (RS West-/East)

Exemplary data snapshot

ID_t	subspell	spell	wave	th21103_g2	th21106	th21108	th21111_g2
8002690	1	1	1	Germany	1957	2007	Schleswig-Holstein
8002876	1	4	10	Germany	2016	2018	Lower Saxony
8006300	1	1	1	Germany	1970	2007	North Rhine-Westphalia
8006419	1	2	1	Germany	1980	2007	Brandenburg
8012120	1	3	1	Germany	1981	2007	Hesse

The dataset `spResidence` shows the retrospectively surveyed places of residence of the respondents. The data not only reflect the current residence (at the time of the interview), but also the individual relocation history with start (e. g., `th21106`) and end date (e. g., `th21108`) for each episode. For data protection reasons, the places of residence are only accessible at the federal state level (`th21111_g2`, in the Download version) and the administrative district level (`th21111_g3R`, in the RemoteNEPS version). For foreign places of residence, the respective country is indicated (`th21103_g2`).

Please note that this residential history is **not** collected for the entire sample, but only for a small subpopulation. Only respondents who have already participated in the ALWA study (wave 1, see section 2.2) are asked questions about their previous places of residence.

Example 25 (Stata): Working with spResidence (find R example here)

```
** open the data file
use ${datapath}/SC6_spResidence_D_${version}.dta, clear

** switch to english language
label language en

** only keep full or harmonized episodes
keep if subspell==0

** find all persons who live or ever lived in Bremen
bysort ID_t: egen bremen = max(th21111_g2==4)

** reduce the datafile, so you have one single row for each respondent
keep ID_t bremen
duplicates drop

** you now can save this datafile ...
tempfile bremen
save `bremen'

** .. and merge it to, e.g., CohortProfile
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear
label language en
merge m:1 ID_t using `bremen', nogen keep(master match)

** please note that data in spResidence is only available for the ALWA-sample!
tab tx80105 bremen, miss
```

4.2.26 spSchool

« go back to overview

Description

Spell data on general schooling history

File structure

spell format: 1 row = 1 schooling episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

64 / 38,482

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
wave	Wave
ts11204	Type of school
ts1111m	Start date month School episode
ts1111y	Start date year School episode
ts1112m	End month School episode
ts1112y	End year School episode
ts11209	School-leaving qualification
ts11214	Intended school-leaving qualification
ts11218	Final grade Leaving certificate
t724801	1st 'Abitur' subject
t724802	2nd Abitur subject

Exemplary data snapshot

ID_t	splink	subspell	spell	wave	ts1111y	ts1112y
8001138	220004	2	4	3	2009	2010
8002438	220002	1	2	1	1997	2007
8004580	220004	1	4	1	2006	2008
8006504	220004	2	4	2	2007	2008
8012336	220004	2	4	2	2005	2008

The file `spSchool` covers the general educational history of each respondent from school entry to (expected) completion, including

- periods of primary schooling,
- completed secondary school episodes leading to a school leaving certificate, and
- incomplete schooling episodes that would have led to a school leaving certificate if they had been completed.

Usually, a new episode with start date (`ts1111m/y`) and end date (`ts1112m/y`) is generated when the school type changes. This means that a change from one *Gymnasium* to another is **not** recorded here. As a result, a single schooling episode can take place at more than one location. In such cases, only information about the last location is considered. A new episode is created each time a school type is changed, even if both schools offer the same certificate.

Example 26 (Stata): Working with spSchool (find R example here)

```
** open the data file
use ${datapath}/SC6_spSchool_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.27 spSchoolExtExam

« go back to overview

Description

Spell data on school exam certificates acquired outside the regular German education system

File structure

entity format: 1 row = 1 external school exam of 1 respondent

ID variables needed to identify a single row

ID_t exam

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

32 / 970

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
exam	Exam number
ts11300	Awarded qualification in Germany?
ts1130m	Date: month qualification was awarded
ts1130y	Date: year qualification was awarded
ts11302	Awarded school-leaving qualification
ts11300_g1	Awarded qualification in Germany? (edited)
ts11308	Overall grade on school-leaving certificate
ts11301_g1R	Country of school-leaving qualification
ts11301_g2	Country of awarded school-leaving qualification (categorized)

Exemplary data snapshot

ID_t	wave	exam	ts11300	ts1130y	ts11302	ts11300_g1
8006518	1	1	1	1998	4	1
8011946	1	1	1	1996	3	1
8012713	1	2	1	2002	4	1
8023392	4	1	1	1973	3	1
8024300	4	1	1	1989	2	1

The file `spSchoolExtExam` contains information about school exam certificates which were not acquired through “regular” schooling in the German educational system. This could be:

- certificates obtained abroad and recognized by German authorities,
- certificates obtained at a German school as an external examinee (i. e., without attending class lessons), or
- certificates that are automatically awarded by skipping class levels into upper secondary education.

The dataset, for instance, informs whether the school exam certificate was awarded in Germany (`ts11300/_g1`), in which month and year the certificate was obtained (`ts1130m/y`), and what type of certificate was acquired (`ts11302`).

Example 27 (Stata): Working with spSchoolExtExam (find R example here)

```
** aim of this example is to evaluate the age of the respondent
** at the exam

** first, we have to get the birth date of the respondent
use ${datapath}/SC6_pTarget_D_${version}.dta, clear
keep if wave==1  // only first wave as this data is time-invariant
keep ID_t t70000m t70000y
label language en
tempfile temp
save `temp'

** now, open the data file
use ${datapath}/SC6_spSchoolExtExam_D_${version}.dta, clear
label language en

** merge the previously extracted birth dates
merge m:1 ID_t using `temp', keep(master match) nogenerate

** recode the two date variables (year, month) into one:
gen exam_date=ym(ts1130y,ts1130m)
gen birth_date=ym(t70000y,t70000m)
format *_date %tm

** calculate the age (in years)
gen age=(exam_date-birth_date)/12

** recode missings to .a, b.,... (not necessarily needed)
nepsmiss ts11302

** show some deviation
tabulate ts11302, summarize(age)
```

4.2.28 spUnemp

« go back to overview

Description

Spell data on unemployment episodes

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

33 / 28,701

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
subspell	Number of subspell
spell	Spell number
wave	Wave
ts2511m	Start Unemployment episode (month)
ts2511y	Start Unemployment episode (year)
ts2512m	End Unemployment episode
ts2512y	End Unemployment episode
ts25202	Receiving unemployment benefits or assistance at start
ts25203	registered unemployment at present/at end
ts25205	Number Job applications
ts25206	Invitation to job interviews
ts25207	Number Interviews

Exemplary data snapshot

ID_t	subspell	spell	wave	ts2511m	ts2511y	ts2512m	ts2512y
8002837	2	8	8	8	2014	1	2015
8006313	2	3	9	9	2015	2	2016
8007168	3	1	4	8	1991	12	2011
8010757	2	1	3	8	2001	12	2010
8019934	2	1	5	1	2011	10	2012

The dataset `spUnemp` contains all episodes of unemployment, regardless of whether a person was registered as unemployed or not. Questions on unemployment registration and the receipt of social benefits refer to both the beginning (`ts2511m/y`) and the end (`ts2512m/y`) of an unemployment episode.

Example 28 (Stata): Working with spUnemp (find R example here)

```
** open the data file
use ${datapath}/SC6_spUnemp_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
```

```
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.29 spVocExtExam

« go back to overview

Description

Spell data on vocational training certificates acquired outside the regular German VET system

File structure

entity format: 1 row = 1 external vocational exam of 1 respondent

ID variables needed to identify a single row

ID_t exam

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

24 / 869

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
exam	Exam number
ts15301_g1	Professional/specialization title (KldB 1988)
ts15301_g4	Professional/specialization title (ISCO-08)
ts15301_g6	Professional/specialization title (SIOPS-88)
ts1530m	Date of external examination (month)
ts1530y	Date of external examination (year)
ts15304	external examination qualification
ts15302	External examination in Germany/abroad
th28370	External examination preparation done abroad for at least one month

Exemplary data snapshot

ID_t	wave	exam	ts1530m	ts1530y	ts15304
8004276	4	1	10	2011	17
8005478	3	2	1	2011	19
8008243	11	1	3	2018	1
8012041	3	1	7	2010	28
8022979	4	1	1	2011	28

The file `spVocExtExam` contains information on vocational training certificates acquired outside the "regular" German VET (*Vocational Education and Training*) system. This could be:

- certificates obtained abroad and recognized by German authorities, or
- certificates obtained in a German vocational training exam as an external examinee (i. e., without participation in lessons or courses registered with German authorities).

This includes in particular the second and third state examinations for graduates of medical and legal studies. Among other things, the dataset provides information on the respective examination date for the acquisition of the certificate (`ts1530m/y`) and the type of qualification acquired through the external examination (`ts15304`).

Example 29 (Stata): Working with spVocExtExam (find R example here)

```
** aim of this example is to evaluate the age of the respondent
** at the exam

** first, we have to get the birth date of the respondent
use ${datapath}/SC6_pTarget_D_${version}.dta, clear
keep if wave==1  // only first wave as this data is time-invariant
keep ID_t t70000m t70000y
label language en
tempfile temp
save `temp'

** now, open the data file
use ${datapath}/SC6_spVocExtExam_D_${version}.dta, clear
label language en

** merge the previously extracted birth dates
merge m:1 ID_t using `temp', keep(master match) nogenerate

** recode the two date variables (year, month) into one:
gen exam_date=ym(ts1530y,ts1530m)
gen birth_date=ym(t70000y,t70000m)
format *_date %tm

** calculate the age (in years)
gen age=(exam_date-birth_date)/12

** recode missings to .a, b.,... (not necessarily needed)
nepsmiss ts15304

** show some deviation
tabulate ts15304, summarize(age)
```

4.2.30 spVocPrep

« go back to overview

Description

Spell data on vocational preparation measures

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

28 / 1,457

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
splink	Link for spell merging
spell	Spell number
subspell	Number of subspell
spgen	Generated spell
wave	Wave
ts13103	Type of measure
ts1311m	Start Vocational preparation (month)
ts1311y	Start Vocational preparation (year)
ts1312m	End date vocational preparation (month)
ts1312y	End date vocational preparation (year)
ts1312c	Ongoing of the vocational preparatory year
ts13201	Termination Vocational preparation

Exemplary data snapshot

ID_t	spell	subspell	wave	ts1311m	ts1311y	ts1312m	ts1312y
8001634	1	1	9	7	2016	10	2016
8002040	1	2	4	11	2010	9	2011
8002156	2	2	7	8	2013	7	2014
8006001	1	1	2	11	2009	2	2010
8012406	2	1	8	1	2016	1	2016

The file `spVocPrep` describes episodes of vocational preparation after general schooling like

- pre-training courses,
- years of basic vocational training, and
- work preparation courses of the Federal Employment Agency (*Bundesagentur für Arbeit*).

Data were collected on the duration from the beginning (`ts1311m/y`) to the end (`ts1312m/y`) of a vocational preparation measure, including possible interruptions.

Example 30 (Stata): Working with spVocPrep (find R example here)

```
** open the data file
```

```
use ${datapath}/SC6_spVocPrep_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by information from
** the spell module. The number of total episodes did not change. Verify this
** by tabulating the spell type by the merging variable generated.
tab sptype _merge
```

4.2.31 spVocTrain

« go back to overview

Description

Spell data on the history of vocational education and training

File structure

spell format: 1 row = 1 vocational training episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

96 / 44,828

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
spell	Spell number
subspell	Number of subspell
ts15201	Type of vocational training
ts1511m	Start date (month/year) Vocational training episode (month)
ts1511y	Start date (month/year) Vocational training episode (year)
ts1512m	End date (month/year) Training episode (month)
ts1512y	End date (month/year) Training episode (year)
ts15215	Company size of training company
ts15219	Vocational qualification
ts15221	Intended vocational qualification

Exemplary data snapshot

ID_t	spell	subspell	ts1511m	ts1511y	ts1512m	ts1512y
8002641	9	8	4	2009	12	2016
8006322	3	2	10	2008	3	2011
8002319	4	1	5	2007	2	2008
8012254	1	1	10	2003	11	2007
8005878	1	3	10	2004	5	2011

The dataset `spVocTrain` comprises all further trainings, vocational and/or academic, with start dates (`ts1511m/y`) and end dates (`ts1512m/y`) that a respondent has ever attended. These include in detail:

- vocational training and retraining
- training at technical schools such as schools of public health, full-time vocational schools (without the years of basic vocational training), other vocational schools and master craftsmen's colleges
- training in specialized fields of medicine
- accredited training courses for obtaining licenses (only up to wave 9)
- doctorate or habilitation/postdoctoral thesis

- higher education at universities, universities of applied sciences, universities of applied sciences, universities of applied sciences for continuing vocational education and universities of applied sciences for administrative sciences and commerce. Note: Only the main subjects are surveyed. New episodes are generated in this context as soon as:
  - the main subject is changed during the course of study, or
  - the desired or attainable degree changes in the course of the study (e. g., from MA to teaching certification).

  On the other hand, episodes are continued when a location is changed, unless the main subject changes as well.

Trainings for licenses are comparable to courses in the files `spCourses`, `spFurtherEdu1` and `spFurtherEdu2` and can therefore be identified by the spell indicator `course`. This enumerator variable makes it possible to link information about the few courses contained in this dataset with the courses in the files just mentioned. Interruptions to vocational training, so-called interruption episodes, are stored in wide format; this should be noted when working with the harmonized spell data.

Example 31 (Stata): Working with spVocTrain (find R example here)

```
** open the data file
use ${datapath}/SC6_spVocTrain_D_${version}.dta, clear

** only keep full or harmonized episodes
keep if subspell==0

** save this file temporarily
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC6_Biography_D_${version}.dta, clear

** merge the previously created temporary data file to this
merge 1:1 ID_t splink using `tmp' , keep(master match)

** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e., the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.
tab sptype _merge
```

4.2.32 spVolunteerWork

« go back to overview

Description

Spell data on volunteer positions held by respondents

File structure

entity format: 1 row = 1 volunteer work activity of 1 respondent

ID variables needed to identify a single row

ID_t volunteer

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

26 / 11,717

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
wave	Wave
volunteer	Volunteering number
t262950_g1	Volunteering - societal areas
t262951_g1	Volunteering - main contents
t261902	Duration of activity
t261903	Volunteering - time expenditure
t262961	Volunteering - Requirement Organizational talent
t262962	Volunteering - Requirement Leadership competence
t262964	Volunteering - Requirement Specialist knowledge
t262967	Volunteering - Requirement Resilience
t517393	Share women (Volunteering)

Exemplary data snapshot

ID_t	wave	volunteer	t262950_g1	t261903
8000417	6	601	Judiciary and crime problems	less frequently
8001850	6	601	Leisure time and sociability	several times a month
8006002	6	602	Church, religion	once a month
8006489	6	601	Emergency and ambulance service	once a week
8022355	6	601	Leisure time and sociability	once a month

The data file `spVolunteerWork` contains up to three reported volunteer work activities per participant. In addition to the area of activity concerned (`t262950_g1`) and the time spent on it (`t261903`), the dataset also provides information on the requirements of the volunteer work activity and the proportion of women and persons with a migrant background in it.

Example 32 (Stata): Working with spVolunteerWork (find R example here)

```
** open the data file
use ${datapath}/SC6_spVolunteerWork_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** evaluate which ids are needed to identify single rows
isid ID_t volunteer
```

4.2.33 Weights

« go back to overview

Description

Sample weights for various applications

File structure

wide format: 1 row = 1 target

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

none

Number of variables / number of rows in file

34 / 17,140

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11

Exemplary variables

ID_t	ID target
sample	Sample
subsample	Subsample
psu	Sample: Primary Sampling Unit (Point number)
stratum	Sample: Stratum
w_t23_std	Weight for TP with participation in waves 2,3 (standardized)
w_t2_cal	Weight for TP with participation in wave 2 (calibrated at microcensus 2009)
w_t3_cal	Weight for TP with participation in wave 3 (calibrated at microcensus 2010)
w_t9_cal	Weight for TP with participation in wave 9 (calibrated at microcensus 2016)

Exemplary data snapshot

ID_t	sample	subsample	psu	stratum	w_t23_std
8012103	2007 (ALWA)	2007 (ALWA)	138	504	1.39
8010237	2007 (ALWA)	2007 (ALWA)	73	507	0.73
8001964	2007 (ALWA)	2007 (ALWA)	192	507	1.31
8012765	2007 (ALWA)	2007 (ALWA)	36	708	1.29
8006872	2007 (ALWA)	2007 (ALWA)	190	503	0.38

Weighting variables (starting with `w_`) are included in the `Weights` dataset. The dataset also contains identifiers for primary sampling units (`psu`) and stratification (`stratum`). Given the rather complex structure of the panel sample, there are no final recommendations or general rules for the use of design and adjusted weights. Detailed information on weight estimation can be found in Hammon et al., 2016 as well as in further reports at the documentation website (see section 1.2).

There are also no general rules on how the use of weights makes a possible analysis more stable. Weights may help to highlight important features of the analysis or at least serve as a robustness check for the analysis performed.

Example 33 (Stata): Working with Weights (find R example here)

```
** open Weights datafile
```

```
use ${datapath}/SC6_Weights_D_${version}.dta, clear

** note that this file is cross-sectional, although the weights
** seem to contain panel logic
d w_t*

** only keep weight corresponding to all waves
keep ID_t w_t23456789_std

** create a "panel" logic, i.e., clone each row
expand 9

** then create a wave variable
bysort ID_t: gen wave=_n

** save as temporary file
tempfile weights
save `weights', replace

** open CohortProfile
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear

** and merge weight
merge 1:1 ID_t wave using `weights', nogen

** note that this weight is only non-zero if respondents participated in
** all waves
tab wave tx80220 if w_t23456789_std!=0
```

4.2.34 xPlausibleValues

« go back to overview

Description

Plausible Values of competence data

File structure

wide format: 1 row = 1 respondent

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

wave_w*

Number of variables / number of rows in file

127 / 11,540

Contains data from waves

1 2 **3** 4 **5** 6 7 8 **9** 10 11

Exemplary variables

ID_t	ID target
wave_w3	Row contains data from wave 3 (2010/2011)
wave_w5	Row contains data from wave 5 (2012/2013)
wave_w9	Row contains data from wave 9 (2016/2017)
maa3_pv1	Math: cross-sectional plausible value 1
maa3_pv2	Math: cross-sectional plausible value 2
maa3_pv10	Math: cross-sectional plausible value 10
maa3_pv1u	Math: longitudinal plausible value 1
maa3_pv2u	Math: longitudinal plausible value 2
maa3_pv10u	Math: longitudinal plausible value 10

Exemplary data snapshot

ID_t	wave_w3	maa3_pv1	maa3_pv2	maa3_pv10	maa3_pv1u
8005513	1	1.3743	0.8021	0.5824	2.0241
8004483	1	0.4063	0.8010	0.9985	0.6181
8008012	1	0.3860	0.5629	0.6853	1.5241
8006801	1	2.0175	1.3144	1.3007	1.0854
8000513	1	0.7973	0.6850	0.4065	1.3702

Plausible Values (PV) are a way of describing the competencies of individuals at the group level. They allow (unbiased) estimates of effects at the population level that are adjusted for measurement errors. In contrast to point estimators such as Weighted Likelihood Estimates (WLE), the use of Plausible Values is suitable for more precise inferential statistical tests in correlation and mean value analyses.

Plausible Values are based on the individual answers in the competence tests and additional background characteristics (e. g. gender, age, socioeconomic status). For each person, the probability distribution of his or her competence is first determined and then several values are randomly drawn from it (hence *Plausible Values*). Hypothesis tests for the specific question of interest are calculated for each of these values and combined into an overall result.

Please find more information on Plausible Values in the corresponding NEPS Survey Paper (Scharl, Carstensen, and Gnambs, 2020) and on our website:

→ www.neps-data.de > Data Center > Overview and Assistance > Plausible Values

Example 34 (Stata): Working with xPlausibleValues

```
** open datafile.
use ${datapath}/${cohort}_xPlausibleValues_D_${version}.dta, clear
label language en

** as the 'x' in the filename indicates, this is a cross sectional file
** (no wave structure). You can verify this by asking if one row is
** solely identified by the respondents ID
isid ID_t

** note that competence testing has been conducted in multiple waves.
** An indicator marks if a row contains information for a specific wave.
tab1 wave_w*

** see more on how to work with this data in the Survey Paper mentioned above!
```

4.2.35 xTargetCompetencies

« go back to overview

Description

Competence data of respondents

File structure

wide format: 1 row = 1 target

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

wave_w*

Number of variables / number of rows in file

449 / 12,465

Contains data from waves

1 2 **3** 4 **5** 6 **7** 8 **9** 10 11

Exemplary variables

ID_t	ID target
wave_w3	Row contains data from wave 3 (2010/2011)
wave_w5	Row contains data from wave 5 (2012/2013)
maa3q071_c	Mathematical competence: Item 1
maa3d041_c	Mathematical competence: Item 8
maa3_sc1	Mathematical competence: WLE (corrected)
maa3_sc2	Mathematical competence: SE(WLE) (corrected)
rea3_sc1	Reading competence: WLE
rea3_sc2	Reading competence: SE of WLE
rea30110_c	Reading competence: Item 1
rea3012s_c	Reading competence: Item 2

Exemplary data snapshot

ID_t	wave_w3	wave_w5	maa3_sc1	maa3_sc2	rea3_sc1	rea3_sc2
8005619	1	1	1.11898	0.56477	4.31540	2.17796
8006640	1	1	0.72071	0.84802	1.08278	0.76560
8001608	1	1	0.56096	0.54279	0.85029	0.62809
8005175	1	1	1.81645	0.65275	2.86147	1.31344
8006394	1	1	0.82749	0.74116	0.90509	0.70644

The file `xTargetCompetencies` contains the data of the competence tests with the respondents. Currently, these are cognitive basic skills as domain-general competency as well as reading, listening comprehension, mathematics, and scientific competence as domain-specific competencies as well as ICT literacy as metacompetency. Scored item variables and aggregated scale variables are available in a cross-sectional wide format (for an overview of the timing of the competence measures see Table 2; for a description of the naming conventions see section 3.2.2).

Please note that **not** all respondents took part in the competence tests. Since the assessments could only be carried out in CAPI (personal) mode, there is no corresponding data available for persons interviewed in CATI (telephone) mode. In addition, those respondents who had severe

visual impairments or were even blind were excluded from the competence measurement. The variables `wave_w*` allow you to select those respondents for whom only data from a particular wave is available.

Example 35 (Stata): Working with xTargetCompetencies (find R example here)

```
** open datafile
use ${datapath}/SC6_xTargetCompetencies_D_${version}.dta, clear

** change language to english (defaults to german)
label language en

** as the 'x' in the filename indicates, this is a cross sectional file
** (no wave structure). You can verify this by asking if one row is
** solely identified by the respondents ID
isid ID_t

** note that competence testing has been conducted in multiple waves
** an indicator marks if a row contains information for a specific wave
tab1 wave_w*

** to work with competence data, you might want to merge it to CohortProfile.
** if you want to keep the panel logic (and not only add all competencies
** to every wave), you need a mergeable wave variable in xTargetCompetencies.
** in this example, we focus on math competencies which have been tested in wave 3.
generate wave=3

** now, remove cases which did not took part in the testing
drop if wave_w3==0

** and reduce the dataset to the relevant variables
keep ID_t wave maa3_sc1 maa3_sc2

** save a temporary datafile
tempfile tmp
save `tmp'

** and merge this to CohortProfile
use ${datapath}/SC6_CohortProfile_D_${version}.dta, clear
merge 1:1 ID_t wave using `tmp', nogen
```

5 References

Antoni, M., Drasch, K., Kleinert, C., Matthes, B., Ruland, M., & Trahms, A. (2011). Arbeiten und Lernen im Wandel. Teil I: Überblick über die Studie (2. aktualisierte Fassung). *FDZ Methodenreport*, *Institut für Arbeitsmarkt- und Berufsforschung (IAB)*(Nürnberg).

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*.

FDZ-LIfBi. (2020). *Data Manual NEPS Starting Cohort 6– Adults, Adult Education and Lifelong Learning, Scientific Use File Version 11.0.0*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Hammon, A., Zinn, S., Aßmann, C., & Würbach, A. (2016). *Samples, Weights, and Nonresponse: the Adult Cohort of the National Educational Panel Study (Wave 2 to 6)* (NEPS Survey Paper No. 7). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

NEPS (Ed.). (2020). *Starting Cohort 6: Adults (SC6), Wave 11, Questionnaires (SUF Version 11.0.0)*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Pohl, S., & Carstensen, C. H. (2012). *NEPS Technical Report – Scaling the Data of the Competence Tests* (NEPS Working Paper No. 14). German National Educational Panel Study (NEPS). Bamberg.

Scharl, A., Carstensen, C. H., & Gnambs, T. (2020). *Estimating Plausible Values with NEPS Data: An Example Using Reading Competence in Starting Cohort 6* (NEPS Survey Paper No. -). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Schönberger, K., & Koberg, T. (2017). *Regional Data: Microm*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Steinwede, J., & Aust, F. (2012). *Methodenbericht, NEPS Startkohorte 5 – CATI-Haupterhebung Herbst 2010, B52*. Bonn, Germany: infas.

Wenzig, K. (2012). *NEPS-Daten mit DOIs referenzieren*. RatSWD Working Paper Series. Rat für Sozial- und Wirtschaftsdaten, Berlin.

Zielonka, M., & Pelz, S. (2015). *Implementation of the ISCED-97, CASMIN and Years of Education. Classification Schemes in SUF Starting Cohort 6*. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

A Appendix

A.1 R examples

In this Appendix, you will find R usage examples that correspond to the Stata usage examples in the main body of the data manual. Before working with R, it is recommended to set a working directory and to install the package *readstata13*:

Example 36 (R): Setting working directory

```
setwd("C:/User/..../Desktop/R_examples")
#set working directory

install.packages("readstata13")
#install the package readstata13 that reads Stata files
library(readstata13)
#import the package readstata13 into library
```

If you'd like to work with the English version of the data, it is recommended to switch the language in Stata first, save the Stata file and then import it in R. The language can be switched by running the command `label language en` in Stata.

To import a data set, use:

Example 37 (R): Importing the data

```
'** here based on the example of the data set spEmp:'
spEmp = read.dta13("SC5_spEmp_D_version.dta", convert.factors = T)
#convert.factors = T converts value labels from Stata into factor label in R
#i.e. "1", "2" data class: integer becomes "yes", "no" data class: factor
```

The following step is not absolutely necessary. However it is recommended, if you attach great importance to keep the variable labels handy during your analysis. After importing the data set, you can display an overview over all variable labels by running the command `varlabel(spEmp)`. However, this command doesn't work anymore after modifying the data by e.g. deleting or merging variables, since the single variable labels aren't attached to the single variable names. To prevent that, following steps are necessary:

Example 38 (R): Assigning variable labels

```
'** here based on the example of the data set spEmp:'

#install and integrates the package "Hmisc"
install.packages("Hmisc")
library(Hmisc)
```

```
#First, create a dataframe with all variable names and labels for spEmp
spEmp_meta = data.frame(attr(spEmp,"names"),attr(spEmp,"var.labels"))

#renames the columns in "names" and "labels"
colnames(spEmp_meta) = c("names", "labels")

spEmp_meta_names = as.vector(spEmp_meta$names)
#extracts the column "names" as vector "spEmp_meta_names"

spEmp_meta_labels = as.vector(spEmp_meta$labels)
#extracts the column "labels"as vector "spEmp_meta_labels"

names(spEmp_meta_labels) = spEmp_meta_names
#assigns the names to the labels, so that the vector "spEmp_meta_labels" is now a
  named vector
#this procedure produces the same result as the following command:
#spEmp_meta_labels = c(ID_t = "Target-ID", splink = "Link für Spell-Merging",
  subspell = "Teilepisodennummer", ... for all variables)

for(i in seq_along(spEmp)){
  label(spEmp[,i]) = spEmp_meta_labels[i]
}
#assigns variable labels that are stored in spEmp_meta_labels to the single columns

label(spEmp)
label(spEmp$subspell)
#Now the variable labels are assigned to the single columns
```

Example 39 (R): Working with Basics

```
'** import the data files'
CohortProfile =
  read.dta13("SC6_CohortProfile_D_9-0-0.dta",
             convert.factors = T)

Basics =
  read.dta13("SC6_Basics_D_9-0-0.dta",
             convert.factors = T)

'** merge the data from Basics, enhancing every entry in CohortProfile'
CohortProfile = merge(CohortProfile, Basics, by = "ID_t", all = TRUE)
#The option all = TRUE makes sure that both, matched AND unmatched cases are kept
  during the merging process

'** tabulate gender by wave'
addmargins(table(CohortProfile$wave, CohortProfile$t700001))
```

Example 40 (R): Working with Biography

```
'** import the data file'
Biography =
```

```
  read.dta13("SC6_Biography_D_9-0-0.dta",
             convert.factors = T)

'** check out which spell modules you can merge to this file'
addmargins(table(Biography$sptype))

'** check that you will need splink to merge information
** from other modules to this file'
anyDuplicated(Biography[,c("ID_t","splink")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate
```

Example 41 (R): Working with Children

```
'** import the data file'
Children =
  read.dta13("SC6_Children_D_9-0-0.dta",
             convert.factors = T)


'** check that you will need ID_t and child (child number)
** to merge information from other modules to this file'
anyDuplicated(Children[,c("ID_t","child")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate


'** check distribution of variable child as a child counter'
addmargins(table(Children$child))
```

Example 42 (R): Working with CohortProfile

```
'** import the data file'
CohortProfile =
  read.dta13("SC6_CohortProfile_D_9-0-0.dta",
             convert.factors = T)

'** how many different respondents are there?'
length(unique(CohortProfile$ID_t))
#number of distinct ID_t

'** as you can see, in this file is an entry for every
** respondents in each wave'
cbind(addmargins(table(CohortProfile$wave)),
      addmargins(prop.table(table(CohortProfile$wave))))


'** check participation status by wave'
cbind(addmargins(table(CohortProfile$wave, CohortProfile$tx80220)))
```

Example 43 (R): Working with EditionBackups

```
'** In this example, we want to restore the original
** values in variable t520003 (weight in kg) in datafile pTarget'

'** import the data file'
EditionBackups =
  read.dta13("SC6_EditionBackups_D_9-0-0.dta",
             convert.factors = T)

'** only keep rows containing data of the variable mentioned above'
EditionBackups = subset(EditionBackups,
                        EditionBackups$dataset == "pTarget" &
                          EditionBackups$varname == "t520003")

'** check which variables we need for merging'
table(EditionBackups$mergevars)


'** then keep the merging variables  and the variable with
** the original values (for cross-checking, we also  keep the
** variable editvalue, which contains the values found in pTarget)'
EditionBackups = subset(EditionBackups,
                        select = c(ID_t, wave, sourcevalue_num, editvalue_num))


'** rename the variables to emphasize affiliation'
names(EditionBackups)[names(EditionBackups) == "sourcevalue_num"] = "t520003_source"
names(EditionBackups)[names(EditionBackups) == "editvalue_num"] = "t520003_edit"


'** open pTarget'
pTarget =
  read.dta13("SC6_pTarget_D_9-0-0.dta",
             convert.factors = T)


'** add the data above'
#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
pTarget = transform(merge(
  x = cbind(pTarget, source = "master"),
  #x contains the pTarget data set plus one extra column "source",
  #where source = "master"
  y = cbind(EditionBackups, source = "using"),
  #y contains the EditionBackups data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "wave")),
  #merges x and y by ID_t and wave
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
                  #in the merged dataset, source = "both" if the observations is in x
                    AND in y
                  ifelse(!is.na(source.x), "master", "using")),
```

```
  #otherwise, source = "master" if the obs. is only in x
  #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)


'** check all editions made'
View(subset(pTarget[c("ID_t", "wave", "t520003", "t520003_source", "t520003_edit")],
            pTarget$source == "both"))


'** replace the variable in the datafile with its original value'
for (i in 1:length(pTarget$t520003)) {
  if(pTarget$source[i] == "both"){
    pTarget$t520003[i] = pTarget$t520003_source[i]
  }
}
```

Example 44 (R): Working with Education

```
'** we want to merge the school type from spSchool to this datafile.
 ** For this to work, we first have to prepare spSchool and keep only
** harmonized episodes (subspell == 0)'
spSchool =
  read.dta13("SC6_spSchool_D_9-0-0.dta",
             convert.factors = T)

spSchool = subset(spSchool, spSchool$subspell ==  0)

'** open the Education data file'
Education =
  read.dta13("SC6_Education_D_9-0-0.dta",
             convert.factors = T)

'** check which spell modules you can merge to this file'
table(Education$tx28100)


'** only keep  school episodes'
Education = subset(Education, Education$tx28100 == "spSchool")


'** check that you will need splink to merge information
** from other modules to this file'
anyDuplicated(Education[,c("ID_t","splink")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate


'** merge spSchool to Education'
#After merging, Stata merge has one variable more than R, because in Stata
```

```
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Education = transform(merge(
  x = cbind(Education,source = "master"),
  #x contains the Education data set plus one extra column "source",
  #where source = "master"
  y = cbind(spSchool[,c("ID_t", "splink", "ts11204")],source = "using"),
  # y contains only the columns ID_t, splink and ts11204 from spSchool
  # plus one extra column "source" where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  # merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
                  # in the merged dataset, source = "both" if the observations is in
                    x AND in y
                  ifelse(!is.na(source.x), "master", "using")),
  #otherwise, source = "master" if the obs. is only in x
  #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  # the columns "source" in x and y are deleted
)


'** see that this only added information to the rows corresponding to spSchool'
cbind(addmargins(table(Education$tx28100, Education$source)))
```

Example 45 (R): Working with FurtherEducation

```
'** import the data file'
FurtherEducation =
  read.dta13("SC6_FurtherEducation_D_9-0-0.dta",
             convert.factors = T)

'** check the source module  of contained courses'
table(FurtherEducation$tx28200)
```

Example 46 (R): Working with MaritalStates

```
'** import the data file'
MaritalStates =
  read.dta13("SC6_MaritalStates_D_9-0-0.dta",
             convert.factors = T)

'** look at the distribution  of family status'
table(MaritalStates$tx27000)
```

Example 47 (R): Working with Methods

```
'** import the data file'
Methods =
```

```
  read.dta13("SC6_Methods_D_9-0-0.dta",
             convert.factors = T)

MethodsEng =
  read.dta13("SC6_Methods_D_9-0-0_eng.dta",
             convert.factors = T)

'** check out participation status by wave'
cbind(addmargins(table(Methods$wave, Methods$tx80220)))

'** how many different interviewers did CATI surveys?'
length(unique(Methods$ID_int))
#unique ID_ints INCL. NA (missing values)

length(unique(Methods$ID_int[!is.na(Methods$ID_int)]))
#unique ID_ints EXCL. NA (missing values)

'** create one single variable containing the interview date'
Sys.setlocale("LC_TIME", "English")
#use when you have English labels
Sys.setlocale("LC_TIME", "German")
#use when you have German labels

Methods$intdate  =
  as.Date(paste(Methods$intm, Methods$intd, Methods$inty, sep = '-'),
          "%B-%d-%Y")
#binds the three columns "intm", "intd" and "inty" into one new column "intdate"

head(Methods[c("intd", "intm", "inty", "intdate")], 10)
#displays first 10 rows of intd, intm, inty and intdate
```

Example 48 (R): Working with MethodsCompetencies

```
'** open the data file'
MethodsCompetencies =
  read.dta13("SC6_MethodsCompetencies_D_9-0-0.dta",
             convert.factors = T)

'** look at the distribution of split groups
 ** note that this has only been conducted in wave 3 2010/2011'
cbind(addmargins(table(MethodsCompetencies$splitgr, MethodsCompetencies$wave)))
```

Example 49 (R): Working with pTarget

```
'** open the data file'
CohortProfile =
  read.dta13("SC6_CohortProfile_D_9-0-0.dta",
             convert.factors = T)

'** merge some variable from pTarget'
pTarget =
  read.dta13("SC6_pTarget_D_9-0-0.dta",
```

```
            convert.factors = T)
#imports the pTarget dataset


CohortProfile =
  merge(x = CohortProfile,
        y = pTarget[,c("ID_t",  "wave", "t400500_g1", "t733001")],
        by = c("ID_t", "wave"), all = TRUE)
#merges only variables "t400500_g1" and "t733001" from pTarget to CohortProfile

'** note: this information is available only in waves which have surveyed the topic'
addmargins(table(CohortProfile$wave, CohortProfile$t400500_g1))

'** if it makes sense, you can copy this information to cells of other waves.
** This copies information downwards (i.e. to late waves), unless a new
** value has been reported (which is usually what you want in a panel study'
for (i in 2:length(CohortProfile$ID_t)) {
  if(CohortProfile$ID_t[i] ==  CohortProfile$ID_t[i-1]) {
    if(is.na(CohortProfile$t400500_g1[i]) |
       CohortProfile$t400500_g1[i] ==  "Missing by design") {
      CohortProfile$t400500_g1[i] = CohortProfile$t400500_g1[i-1]
    }
  }
}


addmargins(table(CohortProfile$wave, CohortProfile$t400500_g1))
```

Example 50 (R): Working with pTargetMicrom

```
'** open pTargetMicrom datafile. Note that this data file is only available OnSite!'
pTargetMicrom = read.dta13("SC6_pTargetMicrom_O_version.dta", convert.factors = T)


'** additionally to ID_t and wave, line identification in this file is done
** via variable regio, denoting the regional level of information'
anyDuplicated(pTargetMicrom[,c("ID_t", "wave" ,"regio")])
#returns 0 if there are no duplicates
#If there are duplicates this command returns the index of the first duplicate


'** tabulating wave against regio shows availability of all levels
** in wave 5 and 7, but only the most detailed level available
** in wave 1 and 3 (usually housing level)'
addmargins(table(pTargetMicrom$wave, pTargetMicrom$regio))

'** only keep housing level'
pTargetMicrom = subset(pTargetMicrom, pTargetMicrom$regio ==  1)

'** now you can enhance CohortProfile with regional data'
CohortProfile = read.dta13("SC6_CohortProfile_O_version.dta", convert.factors = T)
pTargetMicrom = merge(CohortProfile, pTargetMicrom, by = c("ID_t", "wave"), all =
  TRUE)
```

Example 51 (R): Working with pTargetRegioInfas

```
'** open RegioInfas datafile. Note that this data file is only available OnSite!'
RegioInfas = read.dta13("SC6_RegioInfas_O_version.dta", convert.factors = T)


'** identification in this file is done
** via variable regio, denoting the regional level of information'
anyDuplicated(RegioInfas[,c("ID_t", "regio")])
#returns 0 if there are no duplicates
#If there are duplicates this command returns the index of the first duplicate


'** existing regional levels are:'
table(RegioInfas$regio)


'** only keep housing level'
RegioInfas = subset(RegioInfas, RegioInfas$regio ==  1)


'** now you can enhance CohortProfile with regional data'
CohortProfile = read.dta13("SC6_CohortProfile_O_version.dta", convert.factors = T)
RegioInfas = merge(CohortProfile, RegioInfas, by = c("ID_t"), all = TRUE)
```

Example 52 (R): Working with spChild

```
'** open the data file'
spChild = read.dta13("SC6_spChild_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spChild = subset(spChild, spChild$subspell ==  0)

'** generate the total count of children for each respondent
** you can do this either by taking the maximum child number:'
spChild = within(spChild, {children = ave(child, ID_t, FUN = max)})

'** or counting the number of rows:'
spChild = within(spChild, {children2 = ave(ID_t, ID_t, FUN = length)})

'** which both computes the same result'
identical(spChild$children, spChild$children2)

'** recode rough values (e.g. end of year) to real months'
levels(spChild$ts3320m)[levels(spChild$ts3320m) == "Beginning of the year/winter"] =
  "January"
levels(spChild$ts3320m)[levels(spChild$ts3320m) == "Spring/Easter"] = "April"
levels(spChild$ts3320m)[levels(spChild$ts3320m) == "Mid-Year/Summer"] = "July"
levels(spChild$ts3320m)[levels(spChild$ts3320m) == "Fall"] = "October"
levels(spChild$ts3320m)[levels(spChild$ts3320m) == "End of year"] = "December"

'** compute the age of one`s children today
** first, create a date of the birth variables'
```

```
spChild$ts3320m = match(spChild$ts3320m, month.name)

install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

#transforms month names into month numbers
spChild$birth_ym = as.yearmon(paste(spChild$ts3320y, spChild$ts3320m), "%Y %m")

'** then, create the same for the current date'
spChild$today_ym = as.yearmon(rep(cut(Sys.Date(), "month"), length(spChild$ID_t)))

'** the age is then easily computed'
spChild$age = (spChild$today_ym - spChild$birth_ym)

summary(spChild$age)
# displays Min, Max and Mean of "age"
sd(spChild$age, na.rm = TRUE)
# displays Std.Dev. of "age"
length(spChild$age[!is.na(spChild$age)])
# displays the number of observations in "age" without NA
```

Example 53 (R): Working with spChildCohab

```
'** open the data file'
spChildCohab = read.dta13("SC6_spChildCohab_D_9-0-0.dta", convert.factors = T)


'** only keep full or harmonized episodes'
spChildCohab = subset(spChildCohab, spChildCohab$subspell ==  0)

'** recode rough values (e.g. end of year) to real months'
for (i in names(spChildCohab[c(16, 18)])){
  #run over the variables ts3331m and ts3332m in columns 16 and  18
  levels(spChildCohab[[i]])[levels(spChildCohab[[i]]) == "Beginning of the year/
    winter"] = "January"
  levels(spChildCohab[[i]])[levels(spChildCohab[[i]]) == "Spring/Easter"] = "April"
  levels(spChildCohab[[i]])[levels(spChildCohab[[i]]) == "Mid-Year/Summer"] = "July"
  levels(spChildCohab[[i]])[levels(spChildCohab[[i]]) == "Fall"] = "October"
  levels(spChildCohab[[i]])[levels(spChildCohab[[i]]) == "End of year"] = "December"
}

'** generate the following durations in months:
* a) the total duration of a cohabitation episode'
for (i in names(spChildCohab[c(16, 18)])) {
  spChildCohab[[i]] = match(spChildCohab[[i]], month.name)
  #transforms month names into month numbers
}

install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data
```

```
spChildCohab$cohab_start =
  as.yearmon(paste(spChildCohab$ts3331y, spChildCohab$ts3331m), "%Y %m")
spChildCohab$cohab_end =
  as.yearmon(paste(spChildCohab$ts3332y, spChildCohab$ts3332m), "%Y %m")

spChildCohab$cohab_duration =
  (spChildCohab$cohab_end - spChildCohab$cohab_start)*12

'* b) the total duration a respondent lived together with specific child'
spChildCohab = within(spChildCohab,
                      {total_duration_per_child =
                        ave(cohab_duration, ID_t, child, FUN =
                              function(x) round(sum(x, na.rm = TRUE)))})


'* c) the total duration a respondent lived together with any child'
spChildCohab = within(spChildCohab,
                      {total_duration_per_target =
                        ave(cohab_duration, ID_t, FUN =
                              function(x) round(sum(x, na.rm = TRUE)))})

'** to work with the latter information in other files, you could do
** which gives you a cross-sectional display of cohabitation time per respondent'
spChildCohab = subset(spChildCohab, select = c("ID_t", "total_duration_per_target"))
spChildCohab = spChildCohab[!duplicated(spChildCohab),]
```

Example 54 (R): Working with spCourses

```
'** open the data file'
spCourses = read.dta13("SC6_spCourses_D_9-0-0.dta", convert.factors = T)

'** check which modules provided course information'
cbind(addmargins(table(spCourses$sptype)))

'** only keep courses from employment spells'
spCourses = subset(spCourses, spCourses$sptype ==  "Emp")

'** open the employment module'
spEmp = read.dta13("SC6_spEmp_D_9-0-0.dta", convert.factors = T)

'** merge spCourses to spEmp
** note that this is an m:1 merge, as there are still subspells in spEmp'
#Since the variable nepswave is in both data sets spCourses AND spEmp
intersect(names(spCourses), names(spEmp))
#and since the variable is not one of the merging variables, both versions
#are contained in the new merged data set as nepswave.x and nepswave.y.

#To avoid that there are two possibilities:

#1. You can include the variable in the merging process by:
spEmp =
  merge(spEmp, spCourses, by = c("ID_t", "wave", "splink", "nepswave"), all.x = TRUE)
# In that case the version from the master data set, here spEmp, is kept
```

```
#OR

#2. If you'd like to compare the both versions first, you can merge the
#data sets as usual by:
spEmp =
  merge(spEmp, spCourses, by = c("ID_t", "wave", "splink"), all.x = TRUE)

#compare the two versions of the variable nepswave by:
addmargins(table(spEmp$nepswave.x, spEmp$nepswave.y))

#and then drop one of the variables by:
spEmp$nepswave.y = NULL

'** you now have the spEmp datafile, enhanced with information from spCourses,
** and can proceed with this in the usual way'
```

Example 55 (R): Working with spEmp

```
'** open the data file'
spEmp = read.dta13("SC6_spEmp_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spEmp = subset(spEmp, spEmp$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge the spEmp to Biography'
#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spEmp,source = "using"),
  #y contains the spEmp data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
                  #in the merged dataset, source = "both" if the observations is in x
                    AND in y
                  ifelse(!is.na(source.x), "master", "using")),
  #otherwise, source = "master" if the obs. is only in x
  #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)
```

```
#Since the variables wave and spms are in both data sets, Biography AND spEmp
#check before merging by: intersect(names(Biography), names(spEmp))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 56 (R): Working with spFurtherEdu1

```
'** open the datafile'
spFurtherEdu1 = read.dta13("SC6_spFurtherEdu1_D_9-0-0.dta", convert.factors = T)

'** one row contains information for one course.
** The only possibility to use this file is to merge it to the data for this
** respondents wave (we use CohortProfile). So first, we have to remodel
** the file so one row contains one wave.'
spFurtherEdu1$course_nr = ave(spFurtherEdu1$ID_t, spFurtherEdu1$ID_t,
                              spFurtherEdu1$wave, FUN = seq_along)

spFurtherEdu1 = reshape(data = spFurtherEdu1,
                        #data in long format
                        idvar = c("ID_t","wave"),
                        #idvar is/are the variable/s that need/s to be left unaltered
                        v.names = names(spFurtherEdu1[,4:13]),
                        #v.names contains names of variables in the long format that
                        #correspond to multiple variable in the wide format
                        timevar = "course_nr",
                        #timevar is/are the variable/s that need/s to be converted to
                        #wide format
                        direction = "wide")
#direction is to which format the data needs to be transformed


'** open CohortProfile'
CohortProfile = read.dta13("SC6_CohortProfile_D_9-0-0.dta", convert.factors = T)


'** merge the data'
CohortProfile =
  merge(CohortProfile, spFurtherEdu1, by = c("ID_t", "wave"), all.x = TRUE)
'** Please note that you now have multiple variables added to CohortProfile,'
'** one set of variables for each course reported in spFurtherEdu1'
```

Example 57 (R): Working with spFurtherEdu2

```
'** Two possibilities to use spFurtherEdu2'

'-----------------------------------------------------'
'** A) Merge data to spCourses'

'** open spCourses datafile'
spCourses = read.dta13("SC6_spCourses_D_9-0-0.dta", convert.factors = T)

'** one row contains information for up to five courses.
** To make merging possible, you first have to reshape the datafile
** so one row contains only one course'
spCourses = reshape(data = spCourses,
                    # data in wide format
                    idvar = c("ID_t","wave","splink"),
                    #idvar is/are the variable/s that need/s to be left unaltered
                    varying = c("course_w1","course_w2","course_w3","course_w4","
                      course_w5"),
                    #varying are the variables that need to be converted from
                    #wide to long
                    v.names = c("course"),
                    #v.names defines the name of the variable in that the in
                    #varying defined variables are summarized
                    times = c(1,2,3,4,5),
                    #new variable "time" is created with levels 1, 2, 3, 4 and 5
                    #for the five courses
                    new.row.names = 1:150000,
                    #sets row names as numeric
                    direction = "long"
                    ##direction is to which format the data needs to be transformed
)

names(spCourses)[names(spCourses) == "time"] <- "course_nr"
#renames the variable "time" to "course_nr"


'** merge spFurtherEdu2 using ID_t and course'
#open spFurtherEdu2 datafile
spFurtherEdu2 = read.dta13("SC6_spFurtherEdu2_D_9-0-0.dta", convert.factors = T)

intersect(names(spCourses), names(spFurtherEdu2))
#common variables in the both data sets are "ID_t", "wave", "nepswave" and "course"
#Since the variables "wave" and "nepswave" are not part of the merging process,
#both versions are contained in the new merged data set
#as wave.x/wave.y and nepswave.x/nepswave.y.

'**To avoid that, there are two merging options:'
#1. You can include the variables in the merging process by:
spCourses =
  merge(spCourses, spFurtherEdu2,
        by = c("ID_t", "course", "wave", "nepswave"), all.x = TRUE)
#In that case the versions from the master data set are kept (wave.x and nepswave.x)
```

```
#OR

#2. If you'd like to compare the both versions first,
#you can merge the data sets as usual by:
spCourses = merge(spCourses, spFurtherEdu2, by = c("ID_t", "course"), all.x = TRUE)

#compare the two versions of the variables by:
addmargins(table(spCourses$wave.x, spCourses$wave.y))
addmargins(table(spCourses$nepswave.x, spCourses$nepswave.y))

#and then drop one of the versions by:
spCourses$wave.y = NULL
spCourses$nepswave.y = NULL
'------------------------------------------------------------'


'------------------------------------------------------------'
'** B) merge to spFurtherEdu1'

'** open spFurtherEdu1 and FurtherEdu2 datafiles'
spFurtherEdu1 = read.dta13("SC6_spFurtherEdu1_D_9-0-0.dta", convert.factors = T)
spFurtherEdu2 = read.dta13("SC6_spFurtherEdu2_D_9-0-0.dta", convert.factors = T)

'** merge spFurtherEdu2 using ID_t and courses'

intersect(names(spFurtherEdu1), names(spFurtherEdu2))
#common variables in the both data sets are "ID_t", "wave", "course" and "nepswave"
#Since the variables "wave" and "nepswave" are not part of the merging process,
#both versions are contained in the new merged data set
#as wave.x/wave.y and nepswave.x/nepswave.y.

'**To avoid that, there are two merging options:'
#1. You can include the variables in the merging process by:
spFurtherEdu1 =
  merge(spFurtherEdu1, spFurtherEdu2,
        by = c("ID_t", "course", "wave", "nepswave"), all.x = TRUE)
#In that case the versions from the master data set are kept (wave.x and nepswave.x)

#OR

#2. If you'd like to compare the both versions first,
#you can merge the data sets as usual by:
spFurtherEdu1 =
  merge(spFurtherEdu1, spFurtherEdu2,
        by = c("ID_t", "course"), all.x = TRUE)

#compare the two versions of the variables by:
addmargins(table(spFurtherEdu1$wave.x, spFurtherEdu1$wave.y))
addmargins(table(spFurtherEdu1$nepswave.x, spFurtherEdu1$nepswave.y))

#and then drop one of the versions by:
spFurtherEdu1$wave.y = NULL
spFurtherEdu1$nepswave.y = NULL
'------------------------------------------------------------'
```

Example 58 (R): Working with spFurtherEdu3

```
'** Two possibilities to use spFurtherEdu3'

'-----------------------------------------------------------'
'** A) Merge data to spCourses'

'** open spCourses datafile'
spCourses = read.dta13("SC6_spCourses_D_9-0-0.dta", convert.factors = T)

'** one row contains information for up to five courses.
** To make merging possible, you first have to reshape the datafile
** so one row contains only one course'
spCourses = reshape(data = spCourses,
                    # data in wide format
                    idvar = c("ID_t","wave","splink"),
                    #idvar is/are the variable/s that need/s to be left unaltered
                    varying = c("course_w1","course_w2","course_w3","course_w4","
                      course_w5"),
                    #varying are the variables that need to be converted from
                    #wide to long
                    v.names = c("gcourse"),
                    #v.names defines the name of the variable in that the in
                    #varying defined variables are summarized
                    times = c(1,2,3,4,5),
                    #new variable "time" is created with levels 1, 2, 3, 4 and 5
                    #for the five courses
                    new.row.names = 1:150000,
                    #sets row names as numeric
                    direction = "long"
                    ##direction is to which format the data needs to be transformed
)

names(spCourses)[names(spCourses) == "time"] <- "course_nr"
#renames the variable "time" to "course_nr"


'** merge spFurtherEdu3 using ID_t and gcourse'
#open spFurtherEdu3 datafile
spFurtherEdu3 = read.dta13("SC6_spFurtherEdu3_D_9-0-0.dta", convert.factors = T)

intersect(names(spCourses), names(spFurtherEdu3))
#common variables in the both data sets are "ID_t", "wave", "nepswave" and "gcourse"
#Since the variables "wave" and "nepswave" are not part of the merging process,
#both versions are contained in the new merged data set
#as wave.x/wave.y and nepswave.x/nepswave.y.

'**To avoid that, there are two merging options:'
#1. You can include the variables in the merging process by:
spCourses =
  merge(spCourses, spFurtherEdu3,
```

```
        by = c("ID_t", "gcourse", "wave", "nepswave"), all.x = TRUE)
#In that case the versions from the master data set are kept (wave.x and nepswave.x)

#OR

#2. If you'd like to compare the both versions first,
#you can merge the data sets as usual by:
spCourses = merge(spCourses, spFurtherEdu3, by = c("ID_t", "gcourse"), all.x = TRUE)

#compare the two versions of the variables by:
addmargins(table(spCourses$wave.x, spCourses$wave.y))
addmargins(table(spCourses$nepswave.x, spCourses$nepswave.y))

#and then drop one of the versions by:
spCourses$wave.y = NULL
spCourses$nepswave.y = NULL
'-------------------------------------------------------'


'-------------------------------------------------------'
'** B) merge to spFurtherEdu1'

'** open spFurtherEdu1 and FurtherEdu2 datafiles'
spFurtherEdu1 = read.dta13("SC6_spFurtherEdu1_D_9-0-0.dta", convert.factors = T)

names(spFurtherEdu1)[names(spFurtherEdu1) == "course"] <- "gcourse"
#renames the variable "course" to "gcourse"

spFurtherEdu3 = read.dta13("SC6_spFurtherEdu3_D_9-0-0.dta", convert.factors = T)

'** merge spFurtherEdu3 using ID_t and gcourses'

intersect(names(spFurtherEdu1), names(spFurtherEdu3))
#common variables in the both data sets are "ID_t", "wave", "gcourse" and "nepswave"
#Since the variables "wave" and "nepswave" are not part of the merging process,
#both versions are contained in the new merged data set
#as wave.x/wave.y and nepswave.x/nepswave.y.

'**To avoid that, there are two merging options:'
#1. You can include the variables in the merging process by:
spFurtherEdu1 =
  merge(spFurtherEdu1, spFurtherEdu3,
        by = c("ID_t", "gcourse", "wave", "nepswave"), all.x = TRUE)
#In that case the versions from the master data set are kept (wave.x and nepswave.x)

#OR

#2. If you'd like to compare the both versions first,
#you can merge the data sets as usual by:
spFurtherEdu1 =
  merge(spFurtherEdu1, spFurtherEdu3,
        by = c("ID_t", "gcourse"), all.x = TRUE)

#compare the two versions of the variables by:
```

```
addmargins(table(spFurtherEdu1$wave.x, spFurtherEdu1$wave.y))
addmargins(table(spFurtherEdu1$nepswave.x, spFurtherEdu1$nepswave.y))

#and then drop one of the versions by:
spFurtherEdu1$wave.y = NULL
spFurtherEdu1$nepswave.y = NULL
'------------------------------------------------------------'
```

Example 59 (R): Working with spGap

```
'** open the data file'
spGap = read.dta13("SC6_spGap_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spGap = subset(spGap, spGap$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge the spGap to Biography'

#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spGap,source = "using"),
  #y contains the spGap data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
               #otherwise, source = "master" if the obs. is only in x
               #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spGap
#check before merging by: intersect(names(Biography), names(spGap))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
```

```
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 60 (R): Working with spMilitary

```
'** open the data file'
spMilitary = read.dta13("SC6_spMilitary_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spMilitary = subset(spMilitary, spMilitary$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge spMilitary to Biography'

#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spMilitary,source = "using"),
  #y contains the spMilitary data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spMilitary
#check before merging by: intersect(names(Biography), names(spMilitary))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
```

```
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 61 (R): Working with spParLeave

```
'** open the data file'
spParLeave = read.dta13("SC6_spParLeave_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spParLeave = subset(spParLeave, spParLeave$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge spParLeave to Biography'

#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spParLeave,source = "using"),
  #y contains the spParLeave data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
            ifelse(!is.na(source.x), "master", "using")),
            #otherwise, source = "master" if the obs. is only in x
            #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spParLeave
#check before merging by: intersect(names(Biography), names(spParLeave))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
```

```
addmargins(table(Biography$sptype, Biography$source))
```

Example 62 (R): Working with spPartner

```
'** open the data file'
spPartner = read.dta13("SC6_spPartner_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spPartner = subset(spPartner, spPartner$subspell ==  0)

'** to find out if a respondent is oder has ever been married,
 ** check out if the indicating variable ever stated a marriage
 ** you could
 ** ts31410 == "Yes" if respondent is married,'
spPartner$married =
  ifelse(!is.na(spPartner$ts31410) & spPartner$ts31410 == "ja", 1, 0)

spPartner = within(spPartner, {married = ave(married, ID_t, FUN = max)})
#for every ID_t with at least one married == 1, all other married observations
#are also replaced by 1 within this ID_t.


'** look at the data'
spPartner = spPartner[order(spPartner$ID_t),]
#sorts data by ID_t

head(spPartner[c("ID_t", "partner", "ts31410", "married")], 20)
#displays first 20 rows


'** reduce the datafile, so you have one single row for each respondent'
spPartner = subset(spPartner, select = c(ID_t, married))
spPartner = spPartner[!duplicated(spPartner),]


'** you now can merge this datafile to, e.g., CohortProfile'
CohortProfile = read.dta13("SC6_CohortProfile_D_9-0-0.dta", convert.factors = T)
CohortProfile = merge(CohortProfile, spPartner, by = "ID_t", all.x = TRUE)
```

Example 63 (R): Working with spResidence

```
'** open the data file'
spResidence = read.dta13("SC6_spResidence_D_9-0-0.dta", convert.factors = T)


'** only keep full or harmonized episodes'
spResidence = subset(spResidence, spResidence$subspell ==  0)


'** find all persons who live or ever lived in Bremen
** th21111_g2 == "Bremen" if respondent lives or lived in Bremen,'
spResidence$bremen =
```

```
  ifelse(!is.na(spResidence$th21111_g2) & spResidence$th21111_g2 == "Bremen", 1, 0)

spResidence = within(spResidence, {bremen = ave(bremen, ID_t, FUN = max)})
#for every ID_t with at least one bremen == 1, all other bremen observations
#are also replaced by 1 within this ID_t.


'** reduce the datafile, so you have one single row for each respondent'
spResidence = subset(spResidence, select = c(ID_t, bremen))
spResidence = spResidence[!duplicated(spResidence),]


'** you can now merge this datafile to, e.g., CohortProfile'
CohortProfile = read.dta13("SC6_CohortProfile_D_9-0-0.dta", convert.factors = T)
CohortProfile = merge(CohortProfile, spResidence, by = "ID_t", all.x = TRUE)


'** please note that data in spResidence is only available for the ALWA-sample!'
table(CohortProfile$tx80105, CohortProfile$bremen)
```

Example 64 (R): Working with spSchool

```
'** open the data file'
spSchool = read.dta13("SC6_spSchool_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spSchool = subset(spSchool, spSchool$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge spSchool to Biography'

#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spSchool,source = "using"),
  #y contains the spSchool data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
```

```
)

#Since the variables wave and spms are in both data sets, Biography AND spSchool
#check before merging by: intersect(names(Biography), names(spSchool))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 65 (R): Working with spSchoolExtExam

```
'** aim of this example is to evaluate the age of the respondent
 ** at the exam'

'** first, we have to get the birth date of the respondent'
#open pTargetCATI
pTarget = read.dta13("SC6_pTarget_D_9-0-0.dta", convert.factors = T)

#display value labels
levels(pTarget$wave)

#keep only the first wave as this data is time-invariant
pTarget =
        subset(pTarget, pTarget$wave ==  "2007/2008 (ALWA)")

#keep only ID_t, t70000m and t70000y from pTarget
pTarget =
        subset(pTarget, select = c("ID_t", "t70000m", "t70000y"))


'** now, open the data file spSchoolExtExam'
spSchoolExtExam =
  read.dta13("SC6_spSchoolExtExam_D_9-0-0.dta", convert.factors = T)


'** merge the previously extracted birth dates in pTarget to spSchoolExtExam'
spSchoolExtExam = merge(spSchoolExtExam, pTarget, by = c("ID_t"), all.x = TRUE)


'** recode the two date variables (year, month) into one:'

Sys.setlocale("LC_TIME", "English")
#use when you have English labels
Sys.setlocale("LC_TIME", "German")
```

```
#use when you have German labels

spSchoolExtExam$ts1130m = match(spSchoolExtExam$ts1130m, month.name)
spSchoolExtExam$t70000m = match(spSchoolExtExam$t70000m, month.name)
#transforms month names into month numbers

install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

spSchoolExtExam$exam_date =
        as.yearmon(paste(spSchoolExtExam$ts1130y, spSchoolExtExam$ts1130m), "%Y %m")
spSchoolExtExam$birth_date =
        as.yearmon(paste(spSchoolExtExam$t70000y, spSchoolExtExam$t70000m), "%Y %m")
#recode the two date variables (year, month) into one

'** calculate the age (in years)'
spSchoolExtExam$age = (spSchoolExtExam$exam_date - spSchoolExtExam$birth_date)


'** show some deviation'
aggregate(spSchoolExtExam$age, by = list(spSchoolExtExam$ts11302),
        FUN = function(x)
                c(mean = mean(x, na.rm = TRUE),
                sd = sd(x, na.rm = TRUE),freuquency = length(x[!is.na(x)])))
#displays mean and sd of age by school-leaving qualification

sum(!is.na(spSchoolExtExam$age))
#total number of observations without NA

summary(spSchoolExtExam$age)
#display mean of age in general

sd(spSchoolExtExam$age, na.rm = TRUE)
#display sd of age in general
```

Example 66 (R): Working with spUnemp

```
'** open the data file'
spUnemp = read.dta13("SC6_spUnemp_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spUnemp = subset(spUnemp, spUnemp$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge spUnemp to Biography'

#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
```

```
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spUnemp,source = "using"),
  #y contains the spUnemp data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spUnemp
#check before merging by: intersect(names(Biography), names(spUnemp))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 67 (R): Working with spVocExtExam

```
'** aim of this example is to evaluate the age of the respondent
 ** at the exam'

'** first, we have to get the birth date of the respondent'
#open pTarget
pTarget = read.dta13("SC6_pTarget_D_9-0-0.dta", convert.factors = T)

#display value labels
levels(pTarget$wave)

#keep only the first wave as this data is time-invariant
pTarget =
        subset(pTarget, pTarget$wave ==  "2007/2008 (ALWA)")

#keep only ID_t, t70000m and t70000y from pTarget
pTarget = subset(pTarget, select = c("ID_t", "t70000m", "t70000y"))
```

```
'** open the data file spVocExtExam'
spVocExtExam = read.dta13("SC6_spVocExtExam_D_9-0-0.dta", convert.factors = T)

'** merge the previously extracted birth dates in pTarget to spVocExtExam'
spVocExtExam = merge(spVocExtExam, pTarget, by = c("ID_t"), all.x = TRUE)

'** recode the two date variables (year, month) into one:'
Sys.setlocale("LC_TIME", "English")
#use when you have English labels

spVocExtExam$ts1530m = match(spVocExtExam$ts1530m, month.name)
spVocExtExam$t70000m = match(spVocExtExam$t70000m, month.name)
#transforms month names into month numbers

install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

spVocExtExam$exam_date =
        as.yearmon(paste(spVocExtExam$ts1530y, spVocExtExam$ts1530m), "%Y %m")
spVocExtExam$birth_date =
        as.yearmon(paste(spVocExtExam$t70000y, spVocExtExam$t70000m), "%Y %m")
#recode the two date variables (year, month) into one

'** calculate the age (in years)'
spVocExtExam$age = (spVocExtExam$exam_date - spVocExtExam$birth_date)

'** show some deviation'
aggregate(spVocExtExam$age, by = list(spVocExtExam$ts15304),
        FUN = function(x)
                c(mean = mean(x, na.rm = TRUE),
                sd = sd(x, na.rm = TRUE),freuquency = length(x[!is.na(x)])))
#displays mean and sd of age by school-leaving qualification


sum(!is.na(spVocExtExam$age))
#total number of observations without NA

summary(spVocExtExam$age)
#displays mean of age in general

sd(spVocExtExam$age, na.rm = TRUE)
#displays sd of age in general
```

Example 68 (R): Working with spVocPrep

```
'** open the data file'
spVocPrep = read.dta13("SC6_spVocPrep_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spVocPrep = subset(spVocPrep, spVocPrep$subspell ==  0)

'** open the Biography data file'
```

```
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge spVocPrep to Biography'
#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spVocPrep,source = "using"),
  #y contains the spVocPrep data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spVocPrep
#check before merging by: intersect(names(Biography), names(spVocPrep))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 69 (R): Working with spVocTrain

```
'** open the data file'
spVocTrain = read.dta13("SC6_spVocTrain_D_9-0-0.dta", convert.factors = T)

'** only keep full or harmonized episodes'
spVocTrain = subset(spVocTrain, spVocTrain$subspell ==  0)

'** open the Biography data file'
Biography = read.dta13("SC6_Biography_D_9-0-0.dta", convert.factors = T)

'** merge spVocTrain to Biography'
```

```
#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spVocTrain,source = "using"),
  #y contains the spVocTrain data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spVocTrain
#check before merging by: intersect(names(Biography), names(spVocTrain))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

'** you now have an enhanced version of Biography, enriched by
** information from the spell module. The number of total episodes
** (i.e. the amount of rows in the Biography file) did not change.
** Verify this by tabulating the spell type by the merging variable
** generated during the merge process.'
addmargins(table(Biography$sptype, Biography$source))
```

Example 70 (R): Working with spVolunteerWork

```
'** open the data file'
spVolunteerWork = read.dta13("SC6_spVolunteerWork_D_9-0-0.dta", convert.factors = T)

'** evaluate which ids are needed to identify single rows'
anyDuplicated(spVolunteerWork[,c("ID_t","volunteer")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate
```

Example 71 (R): Working with Weights

```
'** open the data file'
Weights = read.dta13("SC6_Weights_D_9-0-0.dta", convert.factors = T)

'** note that this file is cross-sectional,
 **although the weights seem to contain panel logic'
attr(Weights, "var.labels")

'** only keep weights corresponding to all waves'
Weights = subset(Weights, select = c(ID_t, w_t23456789_std))

'** create a "panel" logic, i.e. clone each row'
Weights = Weights[rep(seq_len(nrow(Weights)), each = 9),]

'** then create a wave variable'
Weights$wave = ave(Weights$ID_t, Weights$ID_t, FUN = seq_along)

'** open CohortProfile'
CohortProfile = read.dta13("SC6_CohortProfile_D_9-0-0.dta", convert.factors = T)

#value labels of "wave" in "CohortProfile" and "Weights"
#have to be synchronized before merging
levels((CohortProfile$wave))
levels(Weights$wave)

Weights$wave = as.factor(Weights$wave)
#sets "wave" in "Weights" as factor

for (i in 1:9) {
  levels(Weights$wave)[i] = levels(CohortProfile$wave)[i]
  #assigns the same value labels to "wave" in "Weights" as in "CohortProfile"
}

'** and merges Weights to CohortProfile'
CohortProfile = merge(CohortProfile, Weights, by = c("ID_t", "wave"), all = TRUE)

'** note that this weight is only nonzero if respondents participated in all waves'
with(subset(CohortProfile, w_t23456789_std != 0), addmargins(table(wave, tx80220)))
```

Example 72 (R): Working with xTargetCompetencies

```
'** open the data file xTargetCompetencies'
xTargetCompetencies =
        read.dta13("SC6_xTargetCompetencies_D_9-0-0.dta", convert.factors = T)


'** as the x in the filename indicates, this is a cross sectional file
 ** (no wave structure). You can verify this by asking if one row is
 ** solely identified by the respondents ID'
anyDuplicated(xTargetCompetencies[,c("ID_t")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate

'** note that competence testing has been conducted in multiple waves
```

```
 ** an indicator marks if a row contains information for a specific wave'
table(xTargetCompetencies$wave_w3)
table(xTargetCompetencies$wave_w5)
table(xTargetCompetencies$wave_w7)
table(xTargetCompetencies$wave_w9)

'** to work with competence data, you might want to merge it to CohortProfile.
 ** if you want to keep the panel logic (and not only add all competencies
 ** to every wave), you need a mergeable wave variable in xTargetCompetencies.
 ** here, we focus on math competencies, that have been tested in wave 3.'

#open the data file Cohort Profile
CohortProfile = read.dta13("SC6_CohortProfile_D_9-0-0.dta", convert.factors = T)

xTargetCompetencies$wave =
        rep(levels(CohortProfile$wave)[3],length(xTargetCompetencies$ID_t))
# take the label for wave 3 from CohortProfile, since the labels have to be equal for
   the later merge
xTargetCompetencies$wave = as.factor(xTargetCompetencies$wave)
# change the variable type of wave to factor

'** now, keep cases which did take part in the testing'
xTargetCompetencies = subset(xTargetCompetencies, wave_w3 ==  "ja")

'** and reduce the dataset to the relevant variables'
xTargetCompetencies =
        subset(xTargetCompetencies, select = c(ID_t, wave, maa3_sc1, maa3_sc2))

'** and merge this to CohortProfile'
CohortProfile =
  merge(CohortProfile, xTargetCompetencies, by =  c("ID_t", "wave"), all = TRUE)
```

A.2 Release notes

The following is the release note taken from the documentation page at the time this document has been computed:

```
================================================
**
** NEPS STARTING COHORT 6 - RELEASE NOTES a.k.a CHANGE LOG
** changes and updates for release NEPS SC6 10.0.1
** (doi:10.5157/NEPS:SC6:11.0.0)
**
================================================


================================================
* Changes introduced to NEPS:SC6 by version 11.0.0 *
================================================

General:
        - metadata for all variables have been revised and updated where necessary
        - data from the wave 11 interviews have now been integrated into the Scientific
            Use File

Weights:
        - in addition to providing the new weights for wave 11, the calibrated weights
            of wave 8 (w_t8_cal) have been subsequently adjusted
                to correct an error in the educational levels classification during
                    calibration
        - for wave 8, 9 and 10 the database for the nonresponse models and weighting
            procedures has been updated

xPlausibleValues:
        - new dataset since release 11-0-0: provides plausible values for competency
            data stored in xTargetCompetencies

Methods / CohortProfile:
        - indicator variables for linking the NEPS datsets with data from the NEPS-
            ADIAB project have been added

FurtherEducation:
        - course numbers from wave 2 suffered a coding error. This has been fixed.

pTarget:
        - variables t32454g and t32552g suffered a preload error in wave 7 and wave 11
            data. Values of these variables have been set to
                missing code -92 "(Question erroneously not asked)".

================================================
* Changes introduced to NEPS:SC6 by version 10.0.1 *
================================================

General:
        - the spell datasets spChild, spEmp, spPartner, spSchool and spVocTrain as well
            as the data sets Basics and FurtherEducation
               derived from these spell data sets had incorrect entries due to an
                   error in the data preparation process;
               values from the subspells were not correctly transferred to the
                   corresponding subspell 0 in SUF release 10.0.0;
               this error has now been fixed
```

```
================================================
* Changes introduced to NEPS:SC6 by version 10.0.0 *
================================================

General:
        - metadata for all variables have been revised and updated where necessary
        - data from the wave 10 interviews have now been integrated into the Scientific
            Use File

spPartner:
        - information of LAT (living apart together) partners from ALWA participants
            has been moved to the pTarget dataset

spResidence:
        - residence information of ALWA participants has been moved to the pTarget
            dataset

FurtherEducation:
        - for courses that originate from the dataset spFurtherEdu1 (tx28200====31) of
            wave 4, a data editing error in the previous
                SUF release led to missing values in the variables tx2821m [course
                    participation (date/interval) starting date (month)]
                and tx2821y [course participation (date/interval) starting date (year)
                    ]; this problem has now been solved

Weights:
        - in addition to providing the new weights for wave 10, the calibrated weights
            of wave 5 (w_t5_cal) have been subsequently
                adjusted to correct an error in the educational levels classification
                    during calibration


================================================
* Changes introduced to NEPS:SC6 by version 9.0.1 *
================================================

General:
        - renamed datafile RegioInfas to pTargetRegioInfas


CohortProfile:
        - in SC6 SUF 9.0.0 an error in the data processing routine led to incomplete
            data for wave 1 respondents;
                this bug has been fixed with the update


pTarget:
        - the set of generated variables on the "number of children in household" [
            tx20000, tx20001, tx20002, tx20003] contained
                an error in calculating the age of focus children in wave 3 and above
                    as soon as information on the children was
                continuously reported on waves; in these cases the children could
                    systematically not be correctly classified into
                age groups; this bug has been fixed with the update


================================================
* Changes introduced to NEPS:SC6 by version 9.0.0 *
================================================
```

```
General:
        – meta data for all variables have been revised and updated where appropriate
        – data from the interviews in wave 9 have been incorporated into the data


EditionBackups:
        – this new dataset is now available containing the original values of variables
             that have been recoded during the data preparation


==================================================
* Changes introduced to NEPS:SC6 by version 8.0.0 *
==================================================

General:
        – meta data for all variables have been revised and updated where appropriate
        – data from the interviews in wave 8 have been incorporated into the data



==================================================
* Changes introduced to NEPS:SC6 by version 7.0.0 *
==================================================

General:
        – meta data for all variables have been revised and updated where appropriate
        – data from the interviews in wave 7 have been incorporated into the data


spVolunteerWork:
        – in wave 6 interviews, an episode module regarding volunteer work had been
            surveyed;
                this module had been missing in versions 6.0.0 and 6.0.1; this has been
                     fixed


 ==================================================
* Changes introduced to NEPS:SC6 by version 6.0.1 *
==================================================

General:

        – meta data for all variables have been revised and updated where appropriate

spSchool:
        – variable "School–leaving certificate" [ts11209] suffered a coding error in
            version 6.0.0;
                this also led to erroneous codings in derived educational codes ("ISCED
                    –97" [tx28103],
                "CASMIN" [tx28101], "Years of Education[tx28102]) and misleading spell
                    structure of the generated Education
                file as well as the respective data provided in the Basics file; this
                    has been fixed


 ==================================================
* Changes introduced to NEPS:SC6 by version 6.0.0 *
==================================================

General:

        – meta data for all variables have been revised and updated where appropriate
```

```
- data from the interviews in wave 6 have been incorporated into the data
- in version 5.1.0, SPSS data sets erroneously were not equipped with MISSING
    VALUES definitions;
        this has been fixed; this can be corrected in version 5.1.0 using the
            following SPSS syntax:
        * ---------- BEGIN SPSS code ---------- *.
        SPSSINC SELECT VARIABLES MACRONAME="!numvarlist"
         /PROPERTIES TYPE=NUMERIC
        .
        MISSING VALUES !numvarlist (-99 THRU -5)
        .
        * ---------- End SPSS code ---------- *.
        This solution assumes that SPSS is installed including the Python
            integration plugin;
        if this is not the case, the macro '!numvarlist' has to be defined
            manually as a list of all numerical variables in the current data
            set

- up to wave 3 (NEPS main study 2), external exams have been recorded as part
    of the regular school or vocational training
        spell module (resulting in episodes as part of spSchool and spVocTrain,
             respectively);
        starting from wave 4 (NEPS main study 3), these exams are now recorded
            in a separate module
        each (resulting in events in spSchoolExtExam and spVocExtExam,
            respectively);
        with release 6.0.0 of NEPS:SC6, events from waves 1 through 3 have been
             moved to these separate datasets,
        and erased from spSchool and spVocTrain

- subspell-harmonization (filling variables in generated, harmonized subspells
    [spgen==1] in spell data sets) still had few issues in version 5.1.0;
        this has been fixed; in short, these issues could been describe as
            follows:
        variables that are automatically filled by the survey instrument with
            pre-loaded values from an earlier interview, but should
        contain the original value (from a preceeding wave) in the harmonized
            spells, erroneously contain the later (pre-loaded) value
        in the harmonized sub-spell; this can lead to erroneus values
            especially in generated variables, e.g. coded occupations in '
            spEmp';
        the following Stata syntax solves the problem
        (enter the desired list of variables into the local macro '
            correctvarlist';
        replace 'splink' with the corresponding spell identifier 'partner' or '
            child' in entity spell files):
        * ---------- Begin Stata code ---------- *
        local correctvarlist ts23201_g* // 'ts23201_g*' is an example for
            occupational variables in the spEmp data set
        local spellvar splink           // 'splink' is the correct identifier
            in all data sets besides 'spPartner', 'spChild' and 'spChildCohab'
        foreach var of varlist `correctvarlist' {
                bysort ID_t `spellvar' (subspell) : assert ID_t==ID_t[2] if (
                    spgen==1)
                bysort ID_t `spellvar' (subspell) : assert `spellvar'==`
                    spellvar'[2] if (spgen==1)
                bysort ID_t `spellvar' (subspell) : replace `var'=`var'[2] if (
                    spgen==1)
        }
        * ---------- End Stata code ---------- *

spVocTrain:
```

```
        - integration of variable "Type of vocational training program" [ts15201] from
          wave 1 (ALWA) into newer waves has been erroneous in versions up to 5.1.0;
           this has been fixed


spEmp:

        - in the spEmp dataset, variable "Time restriction" [ts23310] erroneously
          contained the unlabeled value "0" instead of the system missing value in
               (sub-)episodes; this has been fixed; in version 5.1.0, the following
                    Stata syntax can be used to fix the problem:
              * ---------- Begin Stata code ---------- *
              replace ts23310=. if ts23310==0
              * ---------- End Stata code ---------- *

spChild:

        - in version 5.1.0 and earlier, the dataset spChild erroneously contained
          information from wave 4 about 56 children of target persons with more than
           5 children that have been
              erroneously pre-loaded during field work; these children have been
                  correctly re-administered lateron in wave 5 and all subsequent
                  waves;
              thus, sub-spell information from wave 4 has been erased in the 6.0.0
                  release

spResidence:

        - in version 5.1.0 and earlier, the dataset spResidence erroneously contained
          1093 missing values for wave 1 participants that have left the panel
          before wave 3. This has been fixed.

pTarget:

        - in dataset pTarget, variables "Specialized fair/congress: professional/
          personal reasons" [t272802_w1] and
              "Specialized fair/congress: Learned something new" [t272802_w1,
                  t272802_v1w1] as well as the corresponding variables for "Lectures
                  "
              [t272802_w2,t272802_w2,t272802_v1w2] and "Self-instruction programs" [
                  t272802_w3,t272802_w3,t272802_v1w3] in version 5.1.0 and earlier
              erroneously were not filled for all interviewees reporting the specific
                   further education activity; this has been fixed

        - the concept of reflecting migrational background in NEPS SUFs has been
          improved in order to also represent migrants in 3.75th generation;
              thus, the older variables on migrational background [t400500_g1,
                  t400500_g2,t400500_g3] in the pTarget dataset have been renamed
                  using
              the "v1" suffix [t400500_g1v1,t400500_g2v1,t400500_g3v1], and the new
                  ones have been introduced

================================================
* Changes introduced to NEPS:SC6 by version 5.1.0 *
================================================

General:

        - meta data for all variables have been revised and updated where appropriate
        - subspell-harmonization (filling variables in generated, harmonized subspells
          [spgen==1] in spell data sets) erroneously filled system missing
```

```
                values in variables that should have been filled by variable
                    harmonization; this has been fixed
                as a consequence, all generated data sets that rely on these harmonized
                     information had to be consecutively updated;
                i.e. 'Biography', 'Education', 'Weights' and 'Basics'

spSchool:

        - variable 'School attendance in Germany?' [ts11103] had erroneously been
          flipped with variable 'Practical vocational instruction' [ts11237]
              in version 5.0.0; this has been fixed
        - variable 'Practical vocational instruction' [ts11237] had erroneously been
          omitted from dissemination in version 5.0.0; this has been fixed
        - variables 'School-leaving certificate' [ts11209] and 'Prospective school-
          leaving certificate' [ts11214] had erroneously not been filled
              for ALWA spells in version 5.0.0; this has been fixed

spFurtherEdu2:

        - variables 'Financial support through social capital' [t323510] and 'Care
          support through social capital' [t323520] had erroneously been
              omitted from dissemination in version 5.0.0; this has been fixed

spEmp:

        - variable 'Auxiliary variable: Type of employment' [ts23911_v1] had
          erroneously been omitted from dissemination in version 5.0.0; this has
          been fixed
        - variable 'Economic sector (WZ 2008)' [ts23240_g1] had erroneously been
          missing for observations from the ALWA survey; this has been fixed

pTarget:

        - variable 'Info job: personal environment 1' [t324540] had erroneously been
          omitted from dissemination in version 5.0.0; this has been fixed
        - variable 'Reference job' [t325520] had erroneously been omitted from
          dissemination in version 5.0.0; this has been fixed
        - variable 'Social circle further education: professional or personal reasons'
          [t32457a] had erroneously been omitted from dissemination
              in version 5.0.0; this has been fixed
        - variables '... learned something new' [t272802_v1w*] had erroneously been
          integrated into one variable in version 5.0.0; this has been fixed
        - concerning ISCED-97 for mother, father: ISCED-97-category '4A' added for
          cases who reported both:
              'Abitur' etc. and vocational training (the latter does not include
                  university degrees in this context);
        - distinction between ISCED-97 '3B' and '5B' now more precise for NEPS waves
          when 'Berufsfachschule' vs. 'Fachschule' was reported
              (this cannot be distinguished for ALWA, which were still coded '5B' for
                   such cases);


spPartner:

        - concerning ISCED-97 for partner: ISCED-97-category '4A' added for cases who
          reported both:
              'Abitur' etc. and vocational training (the latter does not include
                  university degrees in this context);
        - distinction between ISCED-97 '3B' and '5B' now more precise for NEPS waves
          when 'Berufsfachschule' vs. 'Fachschule' was reported
              (this cannot be distinguished for ALWA, which were still coded '5B' for
                   such cases);
```

```
Weights:

        - in all releases up to version 5.0.0, variable 'Primary sampling unit: point
          number' [psu] erroneously calculated distinct sampling points between the
              ALWA and NEPS samplings, even if persons were drawn from the same point
                   in the NEPS refreshment sample; the 'old' variable has been
                  renamed to
              [psu_v1], whilst a new sample point indicator, [psu], consistently
                  numbering all sample points across all samples, has been
                  incorporated

Education:

        - added additional vocational- and school exam information from spVocExtExam
          and spSchoolExtExam;
        - for the sake of linking information from a source spell data set, variables '
          Exam number' [exam] and
              'Source if information of educational qualification' [tx28100] have
                  been added
        - if the temporal order of reported events cannot be identified (same date of
          exams, certificates etc.) to distinguish between
              ISCED-97 '4A' (second cycle: voc. training first - then 'Abitur' etc.)
                  and '4B' (second cycle: 'Abitur' first - then voc. training),
              '4A' is used as a convention;
        -  a (new) technical report on educational variables has been published online,
           please refer to it for further details on educational coding:
              https://www.neps-data.de/Portals/0/NEPS/Datenzentrum/Forschungsdaten/
                  SC6/5-1-0/TR_Derived_Educational_Variables.pdf


================================================
* Changes introduced to NEPS:SC6 by version 5.0.0 *
================================================

General:

        - translation for all meta data (variable and value labels, question texts, etc
          ) have been revised and completed
        - meta data for all variables have been revised and updated where appropriate
        - additional waves 4 and 5 have been incorporated into the data, including
          observations from a sample refreshment in wave 4
        - all data editing scripts have been completely revised, and large parts have
          been completely rewritten to comply with NEPS' data editing
              standards from all Starting Cohorts; this may result in slight
                  differences on the observation level of many data sets,
              but not its overall structure or results of an analysis, when comparing
                   data from waves 1 through 3 with earlier release versions
        - all missing values that do not comply with the official NEPS editing
          standards (i.e. values -9 through -5) have been recoded to an
              equivalent value in the range -29 through -20

Biography:

        - programs checking for overlaps and gaps between episodes have been completely
           and consistently re-written; this will result in different start
              and end dates of episodes, and a different number of data edition gaps


Education:
```

- completed vocational- or school episodes but without any school-leaving qualifications or vocational degree are now classified in CASMIN as '1a' and in ISCED-97 as '0a/1A/1B', and included in the generate data set 'Education'
- vocational episodes with 'senior official' ('hoehere Beamte') vocational degree are now all classified in CASMIN as '3b' and in ISCED-97 as '5a'. (A university degree is the usual requirement to be a 'senior official' and therefore assumed.)
- the level of achieved school leaving qualification once reported cannot be decreased by future lower reports
- the level of vocational degree can be decreased by future lower reports (e.g. vocational training following university studies). A total 'loss' of any degree is excluded; instead the last known level of vocational degree will be considered to classify CASMIN and ISCED-97 in those cases.
- variable 'tx28102' (Years of Education), which is derived from CASMIN, now correctly classifies observations with '1a' in CASMIN level as '-20' ('no degree')
- in case of CASMIN / ISCED-97 related episodes with exactly the same end dates, only the spells leading to the highest CASMIN and ISCED-97 level are incorporated into the generated data set 'Education'

pTarget:

- variables 't731301_g3' and 't731351_g3' (Years of Education), which are derived from CASMIN, now correctly classify observations with '1a' in CASMIN level as '-20' ('no degree')

spPartner:

- ISCED-97 ('ts31212_g1') and CASMIN ('ts31212_g2') values are only generated for integrated spells (i.e. 'subspell==0')
- variable 'ts31212_g3' (Years of Education), which is derived from CASMIN, now correctly classifies observations with '1a' in CASMIN level as '-20' ('no degree')

spResidence:

- from wave 4 on, residential information is surveyed from the original ALWA population; this information, and all retrospective information from the ALWA study itself, are stored in the new data set 'spResidence'

spVocExtExam:

- new data from external vocational exams have been incorporated into this data set

spSchoolExtExam:

- new data from external school exams have been incorporated into this data set

xTargetCompetencies:

- data set 'xCompMethods' has been renamed to 'MethodsCompetencies' in order to comply with naming schemes in the other NEPS Starting Cohorts' Scientific Use data

Weights:

```
        - all variables containing weights and other sampling-specific information have
            been extracted from data set 'Methods'
                and moved to a new data set 'Weights' in order to comply with naming
                    schemes in the other NEPS Starting Cohorts' Scientific Use data

Basics:

        - variable 'Birth in Germany (W/E) or abroad (reconstructed)' [t405000_g2] has
            been rename from t405000_g1 to t405000_g2 in order to
                avoid overlap of variables names with other NEPS Starting Cohorts

================================================
* Changes introduced to NEPS:SC6 by version 3.1.0 *
================================================

spSchool/Education:

        - missing values in ts11209 were corrected for the ALWA wave (-5 instead of -88
            and -6 instead of -96);
        - some resulting classification errors in file Education have been corrected

spVocTrain:

        - recoding of missing values -88 (until today) to standardized code -5 (until
            today) in end dates of interruption episodes
                (ts1532m_w*, ts1532y_w*)

spCourses:

        - variable subspell removed; use ID_t, splink, and wave for merging with spell
            files

FurtherEducation:

        - wrong codes in variables containing estimated course dates (tx2821m, tx2821y,
            tx2822m, tx2822y) have been corrected
        - missing values in tx28201 resolved

Education:

        - episodes without any general or vocational degree deleted for CASMIN, ISCED
            -97 and "years of education"

Methods:

        - variable for day of interview (intd) added
        - variable "Interviewer: ID" [ID_int], known as [tx80300] in release version
            1.0.0, had been omitted; this has been corrected
                and the variable integrated into the data

pTarget:

        - variable for day of interview (intd) added


================================================
* Changes introduced to NEPS:SC6 by version 3.0.0 *
================================================

General:
```

- design weights for 283 respondents from the ALWA study that temporarily dropped out in NEPS main study 1 (wave 2) can not be calculated
- Wave update from Starting Cohort 6: new wave data from second NEPS main survey (2010/2011) has been fully integrated;
  - this scientific use file comprises now three waves (ALWA 2007/2008, NEPS 2009/2010, NEPS 2010/2010);
  - version number has been adjusted to resemble the number of cumulated wave data
- sample size increase up to 11,932 adults since 283 additional cases from the ALWA subsample participated in wave 3
- Selected highlights: updated and fully integrated life course data over three waves; additional concepts (like cultural capital);
  - data from competence assessment (introduced in 2010/2011); new regional data (microm)
- new datafiles available: spChildCohab, xTargetCompetencies (competences data, wave 3),
  - xCompMethods (para data on competence assessment, wave 3),
  - xTargetMicrom (regional data from microm database, accessible only on-site)
- all SPSS datasets now ship with NEPS missing values (range [-99;-5]) marked as MISSING
- preload data for interviews in wave 3 (second NEPS wave 2010/2011) have been added; this data is usually not needed, however,
  - it might help to understand the course of an interview
- metadata for all datasets has been revised and updated where appropriate
- in all spell data sets, variable spstat ('Most recent (sub-)spell status') has been revised; codes 91 and 92 have been removed
- in all spell data sets, for tracing the generation of edited times in Biopgraphy, original variables from the check module have been added.
  - This includes: starting and ending times [{variable_start_month}_g1, {variable_start_year}_g1, {variable_end_month}_g1, {variable_end_month}_g1]
  - as corrected by the check module, a variable marking right censoring of spell after checking routines [ts2312c_g1], and an
  - indicator variable 'type of event' [spms] from check module, discriminating between 'dominant' and 'side' spells
  - IMPORTANT NOTE: Those variables have been added for the sake of completeness and traceability. We strongly recommand to rely on fully edited
  - episodes times that can be found in file Biography.

pTarget:

- variables inty/intm (interview date) have been added for usability concerns (from file Methods)
- small coding corrections for variables t731454 and t731404
- variables fpmod, t733004, and t733005 were moved to file spPartner
- variable t731301_g2 ('Mother: CASMIN') now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable t731351_g2 ('Father: CASMIN') now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable t731301_g2 ('Mother: CASMIN') has been corrected, swapping contents of categories 7 [CASMIN 3a] and 8 [CASMIN 3b];
- variable t731301_g3 ('Mother: Years of education = f(CASMIN)') has been updated accordingly
- variable t731351_g2 ('Father: CASMIN') has been corrected, swapping contents of categories 7 [CASMIN 3a] and 8 [CASMIN 3b];
- variable t731351_g3 ('Father: Years of education = f(CASMIN)') has been updated accordingly
- variable t731301_g1 ('Mother: ISCED') now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]

- variable t731351_g1 ('Father: ISCED') now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable t731453_g2 ('Father's occupation (KldB 2010)') now adequately incorporates information about supervisory occupational tasks where needed
- variable t731453_g14 ('Father's occupation (ISEI-08)') has been added
- variable t731403_g2 ('Mother's occupation (KldB 2010)') now adequately incorporates information about supervisory occupational tasks where needed
- variable t731403_g14 ('Mother's occupation (ISEI-08)') has been added
- a new variable t405000_g1 ('Born in Germany or abroad (reconstructed)') including an categories for born in east or west Germany has been reconstructed from spell information for non wave 1 respondents, reflecting the scale of t405000_v1
- variables t413510_R/t413510_D ('Household: 1st foreign language') have been renamed to t413501_R/t413501_D
- multiple response variables ("Mehrfachnennungen") have been recoded for simplifying their usage: indicator variables for "refusal", "don't know", and "not in list" have been recoded to missing codes (-98,-97,-20) in the response variables and then removed.
  Following multiple response sets are affected:
  1) t725001-t725013: 'repeated school years'; missing indicators t725014 and t725015 removed
  2) t32404k-t32404s: 'Info job'; missing indicators t32404u and t32404v removed
  3) t32502k-t32502t: 'Reference job'; missing indicators t32502u and t32502v removed
  4) t32303k-t32303t: 'Help with application'; missing indicators t32303u and t32303v removed
  5) t32405k-t32405s: 'Information job-related course'; missing indicators t32405w, t32405u, and t32405v removed
  6) t32406k-t32406s: 'Information private course'; missing indicators t32406w, t32406u, and t32406v removed
  7) t32457k-t32457s: 'Social circle further education: who?'; missing indicators t32457s, t32457u, and t32457v removed
  8) t743021-t743031: 'Fellow occupant'; missing indicators t743032 and t743033 removed
  9) t32091k-t32091s: 'Burt'; missing indicators t32091u and t32091v removed
- variables t731403_g8, t731453_g8: EGP class scheme adjusted due to errors in the derivation syntax (particularly the classes IVc and V)

xTargetCompetencies:

- new file containing scored items and scaled values from competences assessment that was conducted at wave 3

xCompMethods:

- para data on competence assessment as generated by a specialized CAPI module at wave 3

spSchool:

- variable ts11218 was recoded to -54 for spells collected in ALWA survey (wave 1); was set to system missing previously
- variable marking right censoring of spell ts1112c has been adjusted (set to system missing for spells ending in the past)

spMilitary:

- variable marking right censoring of spell ts2112c has been adjusted (set to system missing for spells ending in the past)

spVocPrep:

- variable marking right censoring of spell ts1312c has been adjusted (set to system missing for spells ending in the past)

spVocTrain:

- variable ts15215 was incorrectly labeled for ALWA cases (wave 1); an additional variable ts15215_v1 containing values and labels as generated in the ALWA survey corrects for that
- variable ts15214_g1 had for some cases erroneously code –55 (not determinable) instead of system missings (no specification in open input abbras)
- variable marking right censoring of spell ts1512c has been adjusted (set to system missing for spells ending in the past)
- variable 'Description of profession/subject (ISCO–88)' ts15291_g3 now adequately incorporates information about large company size where needed
- variable 'Description of profession/subject (ISEI–08)' ts15291_g14 has been added
- variable 'Aspired vocational education qualification (reconstructed)' ts15221_v1 has been reconstructed, reporting contents for successfully completed episodes only
- variable 'Aspired vocational education qualification' ts15221_ha has been added
- variable 'vocational education qualification' ts15219_v1 no longer reports contents for episodes not successfully completed; these episodes are edited to the value –6 'no leaving certificate'

spEmp:

- ts23210_* was incorrectly labeled for ALWA cases (wave 1); an additional variable ts23210_v1 containing values and labels as generated in the ALWA survey corrects for that
- ts23243 was incorrectly labeled for ALWA cases (wave 1); an additional variable ts23243_v1 containing values and labels as generated in the ALWA survey corrects for that
- variable 'Most recent (sub–)spell status' spstat has been revised; codes 91 and 92 have been removed
- variable 'Episode updating' ts23101 is now correctly sorted before ts23102
- variable ts23201_g2 ('Job description (KldB 2010)') now adequately incorporates information about supervisory occupational tasks where needed
- variable ts23201_g3 ('Job description (ISCO–88)') now adequately incorporates information about large company size where needed
- variable ts23201_g14 ('Job description (ISEI–08)') has been added
- variable ts23221 ('Job volume at end of occupation (part–time/full–time, reconstructed)') has been reconstructed from spell information for wave 1 interviewees, reflecting the scale of ts23218_v1
- variable ts23201_g8: EGP class scheme adjusted due to errors in the derivation syntax (particularly the classes IVc and V)
- variable marking right censoring of spell ts2312c has been adjusted (set to system missing for spells ending in the past)

spUnemp:

- variable marking right censoring of spell ts2512c has been adjusted (set to system missing for spells ending in the past)

spParLeave:

- variable marking right censoring of spell ts2712c has been adjusted (set to system missing for spells ending in the past)

spGap:

- variable marking right censoring of spell ts2912c has been adjusted (set to system missing for spells ending in the past)
- value label for variable 'Kind of gap' ts29101_v1 has been corrected, swapping categories 6 and 7

spChild:

- to enhance usability when analyzing child cohabition spells, those spells – previously being stored in a wide data format – have
  been extracted into a new data file called spChildCohab

spChildCohab:

- new file containing spells of cohabitation with own or other children (the data was previously stored in wide format within spChild)
- the file contains cohabation spells which might be extended over panel waves
- hence, the file has a genuine spell data format involing a spell and a subspell variable
- harmonzised spells are identified by spgen=1 and subspell=0; thus, analougsly to the other spell files
  just select spells having subspell=0 (Stata: keep if subspell==0) for a plain and easy episode structure
- cohabitation spells are related to children in spChild via the identifier "child"
- variable marking right censoring of cohabitation spell ts3332c has been adjusted (set to system missing for spells ending in the past)

spPartner:

- variables fpmod (episode mode), t733004 (living apart together, current partner), and t733005 (frequency of contact, current partner)
  were moved from pTarget to spPartner
- variable ts31212_g2 ('Partner: highest educational achievement (CASMIN)') now correctly classifies observations 'other' eductional
  degree as 'not determinable' [–55]
- variable ts31212_g2 ('Partner: highest educational achievement (CASMIN)') has been corrected, swapping contents of categories
  7 [CASMIN 3a] and 8 [CASMIN 3b];
- variable ts31212_g3 ('Partner: highest educational achievement (years of education=f(CASMIN))') has been updated accordingly
- variable ts31212_g1 ('Partner: highest educational achievement (ISCED)') now correctly classifies observations 'other' eductional
  degree as 'not determinable' [–55]
- variable ts31226_g2 ('Partner: occupation (KldB 2010)') now adequately incorporates information about supervisory occupational tasks
  where needed
- variable ts31226_g14 ('Partner: occupation (ISEI –08)') has been added

- variable ts31226_g8: EGP class scheme adjusted due to errors in the derivation syntax (particularly the classes IVc and V)

spFurtherEdu1:

- variable marking right censoring of spell t271048 has been adjusted (set to system missing for spells ending in the past)

Methods:

- additional wave 1 & 2 rows for additional cases from ALWA
- new variable ALWAlatecomer marking the new cases who come from the ALWA sample but did not participate in wave 2
- additional weights (prob_w3/weight_isced_w3/weight_isced_w3_std) for wave 3
- new variable tx80220 (participation status) for indicating participation, temporary dropouts, and final dropouts

Basics:

- variable 'Highest CASMIN' [tx28101] now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable 'Highest ISCED' [tx28103] now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable 'Mother: CASMIN' [t731301_g2] now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable 'Father: CASMIN' [t731351_g2] now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable 'Mother: CASMIN' [t731301_g2] has been corrected, swapping contents of categories 7 [CASMIN 3a] and 8 [CASMIN 3b];
  - variable 'Mother: Years of education = f(CASMIN)' [t731301_g3] has been updated accordingly
- variable 'Father: CASMIN' [t731351_g2] has been corrected, swapping contents of categories 7 [CASMIN 3a] and 8 [CASMIN 3b];
  - variable 'Father: Years of education = f(CASMIN)' [t731351_g3] has been updated accordingly
- variable 'Mother: ISCED' [t731301_g1] now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable 'Father: ISCED' [t731351_g1] now correctly classifies observations with 'other' eductional degree as 'not determinable' [-55]
- variable 'Mother: SIOPS' [t731403_g6] has been added
- variable 'Mother: MPS' [t731403_g7] has been added
- variable 'Mother: ISEI-08' [t731403_g14] has been added
- variable 'Mother: BLK' [t731403_g9] has been added
- variable 'Father: SIOPS' [t731453_g6] has been added
- variable 'Father: MPS' [t731453_g7] has been added
- variable 'Father: ISEI-08' [t731453_g14] has been added
- variable 'Father: BLK' [t731453_g9] has been added
- variable 'Occupation of first employment (SIOPS)' [tx29075] has been added
- variable 'Occupation of first employment (MPS)' [tx29076] has been added
- variable 'Occupation of first employment (ISEI-08)' [tx29077] has been added
- variable 'Occupation of first employment (BLK)' [tx29078] has been added
- variable 'Current occupation (SIOPS)' [tx29065] has been added
- variable 'Current occupation (MPS)' [tx29066] has been added
- variable 'Current occupation (ISEI-08)' [tx29067] has been added
- variable 'Current occupation (BLK)' [tx29068] has been added
- variable 'Age at migration to Germany' [tx29007] has been added
- variable 'Born in Germany or abroad (reconstructed)' [t405000_g1] has been added

Education:

- variable 'Highest CASMIN' [tx28101] now correctly classifies observations with 'other' eductional degree as 'not determinable' [−55]
- variable 'Highest ISCED' [tx28103] now correctly classifies observations with 'other' eductional degree as 'not determinable' [−55]