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 16.0.0

LIfBi
LEIBNIZ INSTITUTE FOR
EDUCATIONAL TRAJECTORIES

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).

Contact

E-mail: fdz@lifbi.de
Web: https://www.lifbi.de/FDZ

Bibliographic Data

FDZ-LIfBi. (2025). *Data Manual NEPS Starting Cohort 6 – Adults, Adult Education and Lifelong Learning, Scientific Use File Version 16.0.0*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

This data manual for Starting Cohort 6 – Adults "Adult Education and Lifelong Learning" has been prepared by the staff of the Research Data Center at the Leibniz Institute for Educational Trajectories (Forschungsdatenzentrum, FDZ-LIfBi). It represents a major collaborative effort. The contribution of the following persons is particularly acknowledged:

Daniel Fuß
Tobias Koberg
Benno Schönberger
Gregor Lampel
Dietmar Angerer
Katja Vogel

For his support in writing this manual, special thanks go to Ralf Künster

Section 5 Special Issues has been contributed by the following colleagues:

Agnieszka Althaber, Teresa S. Friedrich, Alexander Helbig, Marie-Christine Laible, Josefine C. Matysiak, Ralf Künster, Benjamin Schulz, Annette Trahms, Basha Vicari

Leibniz Institute for Educational Trajectories (LIfBi)
Wilhelmsplatz 3, 96047 Bamberg
Director: Prof. Dr. Cordula Artelt
Administrative Director: Dr. Stefan Echinger
Bamberg; November 21, 2025

Contents

1 Introduction

1.1 About this manual

This manual facilitates your work with data of the NEPS Starting Cohort 6 – Adults. It serves both as a first guide for getting started with the complex data and as a reference book. The primary emphasis is on aspects such as sampling and sample development, conventions of data preparation, data structure, and merging of information. The manual is neither complete nor exhaustive, but several links to other resources are provided in the respective paragraphs. 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 – this manual is regularly updated and revised.

The first chapter refers to further documentation material, requirements for data access, instructions for data citation, some general rules and recommendations, and selected services provided by the FDZ-LIfBi (Forschungsdatenzentrum, Research Data Center). In the second chapter, the fundamental objectives of Starting Cohort 6 (SC6) and its sampling strategy are briefly introduced. The main part of this chapter describes the sample development across the waves including field times, realized case numbers, survey modes, and the measurement of competence domains. The general principles of Scientific Use File data-editing processes as well as the applied conventions for naming the data files and variables are introduced 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 the relevant data types, identifiers, and short portraits of all available datasets in the Scientific Use File. These portraits also include syntax examples for merging variables of this dataset with variables from other datasets. The last chapter addresses some specific issues that should be considered when working with data of Starting Cohort 6. In the manuals for Starting Cohorts 3 and 6 this section provides very detailed explanations of how the biographical life history data were collected and how they are stored in the various spell datasets in the Scientific Use File.

The contents of the first chapter as well as large parts of the third and fourth chapters apply to the Scientific Use Files of all NEPS starting cohorts. It is not mandatory that the examples mentioned there explicitly refer to Starting Cohort 6, but they are transferable accordingly.

1.2 Further documentation

The manual does not address all aspects of data documentation in detail. Therefore, a comprehensive set of reports and additional materials with background information (see Figure 1) can be downloaded from our website:

→ www.neps-data.de > Data and Documentation > Starting Cohort 6 > Documentation

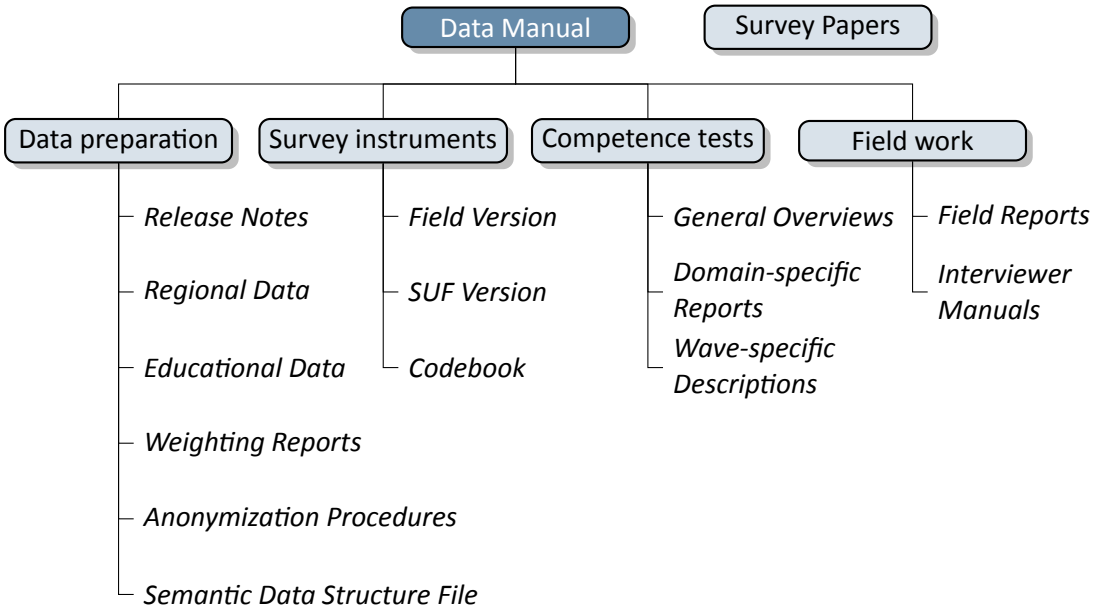

Figure 1: NEPS supplementary data documentation

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

Regional Data Fine-grained regional indicators from commercial providers (microm, RegioInfas) are available in the On-site environment (*not for Starting Cohort 8*). The reports describe the regional levels covered, the content, and how to merge it to the survey data.

Educational Data The report gives an overview of the generation of the derived educational variables ISCED, CASMIN and Years of Education.

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 field versions and Scientific Use File (SUF) versions. While the field versions consist of the origi-

nally 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 aligned 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. Usually, for each domain there is a brief description of the construct with sample items as well as 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. They exemplify the interview process and the content of each of the questionnaire modules. They are available in German only (*not for Starting Cohort 1*).

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 from the LIfBi website at:

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

Additional documentation material might be available for this Starting Cohort. Please visit the documentation website mentioned at the beginning of this chapter 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. **Therefore, it is strongly recommended to use the most current release of a Scientific Use File.**

File Format

All Scientific Use Files are provided in Stata and SPSS format with bilingual variable 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]
```

Versioning and Digital Object Identifier

With each new release of a Scientific Use File, the existing data files 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 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 (see Table 1).

Every release of a NEPS Scientific Use File is registered at `DataCite` and clearly labeled with a unique *Digital Object Identifier* (DOI, see Wenzig, 2012). This DOI has two main functions: On the one hand, it enables researchers to cite the used NEPS data in an easy and precise way (see Section 1.5), which is a fundamental prerequisite 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:SC6:16.0.0`.

Table 1: Release history of SUF in Starting Cohort 6

SUF Version	DOI	Date of release
16.0.0 (current)	doi:10.5157/NEPS:SC6:16.0.0	**2025-11-26**
15.0.0	doi:10.5157/NEPS:SC6:15.0.0	2024-10-22
14.0.0	doi:10.5157/NEPS:SC6:14.0.0	2023-08-25
13.0.0	doi:10.5157/NEPS:SC6:13.0.0	2022-07-11
12.1.0	doi:10.5157/NEPS:SC6:12.1.0	2021-12-09
12.0.1	doi:10.5157/NEPS:SC6:12.0.1	2021-09-08
12.0.0	doi:10.5157/NEPS:SC6:12.0.0	2021-07-15
11.1.0	doi:10.5157/NEPS:SC6:11.1.0	2020-12-02
11.0.0	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
9.0.1	doi:10.5157/NEPS:SC6:9.0.1	2018-12-11
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:SC6:3.0.1	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 to members of the scientific community. It is granted upon 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 all persons included 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 or mail. Further instructions and the relevant forms are provided on our website at:

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

- After approval by the Research Data Center, each registered NEPS data user receives an individual 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 and Documentation > NEPS Data Portfolio

- There are two other modes of access to more sensitive NEPS data (see below); each demanding a Supplemental Agreement in addition to the basic Data Use Agreement.

- A separate form is available to state changes to the Data Use Agreement, such as the later addition of project participants or an extension of the original 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 regarding data utility while complying with the national and international standards of confidentiality protection. Each mode corresponds to a Scientific Use File version that is different in terms of accessibility of sensitive information.

- *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 requires a guest stay at the LIfBi in Bamberg. More details about the access modes can be found at:

→ www.neps-data.de > 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 of a Scientific Use File, e. g., fine-grained regional indicators or open text entries. For more details see:

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

This concept of *nested 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 access level are also included in the corresponding 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" (see Section 1.2).

1.5 Publications with NEPS data

Referencing the use of data from the National Educational Panel Study 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 specify the version of the data used by citing its DOI as follows:

> NEPS Network. (2025). *National Educational Panel Study, Scientific Use File of Starting Cohort 6 – Adults*. Leibniz Institute for Educational Trajectories (LIfBi), Bamberg. https://doi.org/10.5157/NEPS:SC6:16.0.0

In addition, the NEPS study must be referenced at an appropriate point within the publication:

> This paper uses data from the National Educational Panel Study (NEPS; see Blossfeld and Roßbach, 2019). The NEPS is carried out by the Leibniz Institute for Educational Trajectories (LIfBi, Germany) in cooperation with a nationwide network.

Finally, the reference article must be listed in the bibliography:

> Blossfeld, H.-P., & Roßbach, H.-G. (Eds.). (2019). *Education as a lifelong process: The German National Educational Panel Study (NEPS). Edition ZfE* (2nd ed.). Springer VS. https://doi.org/10.1007/978-3-658-23162-0

Authors of any kind of publications based on the NEPS data are requested to notify the FDZ-LIfBi about their articles by sending an e-mail with the bibliographic details to fdz@lifbi.de. All known publications are listed in the NEPS Bibliography on our website at:

→ www.neps-data.de > Publications

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. (2025). *Data Manual NEPS Starting Cohort 6 – Adults, Adult Education and Lifelong Learning, Scientific Use File Version 16.0.0*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

> NEPS Network. (2025). *Starting Cohort 6 – Adults, Wave 16, Questionnaires (SUF Version 16.0.0)*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

If a document is not part of this series, please cite the author and title according to the following example of a field report by one of the survey institutes:

> Malina, A., Wefelmeyer, D. M., Ruland, M., & Aust, F. (2023). *Feld- und Methodenbericht. NEPS-Startkohorte 6 (Erwachsene) – Haupterhebung 2022/2023, Teilstudie B158*. Bonn, Germany: infas Institut für angewandte Sozialwissenschaft GmbH.

1.6 Rules and recommendations

Working with NEPS data is bound to a couple of rules that are defined in the Data Use Agreement. Each data user has to confirm these rules through their signature on the agreement. The already mentioned obligation to cite the NEPS 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 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 such a re-identification. 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 applicant and further project members that are mentioned by name in the agreement). 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!

Please note that a 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 obligation applies in case of encountering any issues concerning data quality or any security leaks with regard to NEPS data protection.

Recommendations

In addition to the aforementioned rules, there are some recommendations for using the 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 research data we cannot guarantee absolute correctness. Notices on problems or errors in the datasets are welcome at any time at the Research Data Center.
- *Enhanced understanding of the data:* Consult the documentation and survey instruments before starting the analyses. The work with complex data requires a precise idea of how the information were collected and processed. All relevant material is available online.
- *Facilitated handling of the data:* Use the tools that are offered. Several user services are provided to support NEPS data analyses – from specific Stata commands (e. g., for an easy recoding of missing values) to a meta search engine (e. g., for an interactive exploration of all instruments) and an online discussion forum (e. g., for asking specific 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 the NEPS data. First and foremost, the Research Data Center maintains a regularly updated and enhanced website with detailed information on all Scientific Use Files, a complete list of registered NEPS analysis projects, a reference to NEPS-related events, and a newsletter. All subsequently introduced services can be reached via this website:

→ www.neps-data.de > NEPS

NEPScomp

The additional *NEPScomp* data offering is designed particularly for the use in university courses and in student theses. NEPScomp features a number of simplifications with regard to data structure and data handling compared to the regular Scientific Use File of Starting Cohort 6. The "comp" in the title – short for "compenedium" – refers to the explicit claim to provide a compact and easy-to-understand NEPS data package that still enables empirically valid findings on a wide range of questions in the field of educational research. Access to this data also requires a valid Data Use Agreement (see Section 1.4).

→ www.neps-data.de > NEPScomp

Variable Search

The *Variable Search* facilitates an interactive and quick full text search through all instruments of released NEPS surveys, including competence variables. The tool is particularly suitable for getting a first idea of the availability of constructs, items, and variables in the datasets. It is based on both keyword search with several filtering options and hierarchical topic search. It also offers some helpful functions such as displaying the occurence of the selected variable in all NEPS starting cohorts, its answering scheme, relevant references, and more. As a web application, the service relies on the most up-to-date information; any correction in the metadata is thus instantly visible.

→ www.neps-data.de > Variable Search

Online Forum

The so-called *Forum4MICA – Making Information Commonly Available* is an open discussion platform for data users as well as for persons who are just searching for relevant information. The forum is joined by various Research Data Centers with their data collections, including the FDZ-LIfBi with the NEPS data. It offers the opportunity to directly exchange with NEPS staff members and with other researchers in a transparent dialogue. We recommend browsing the content first when struggling with NEPS issues or whenever help is needed with specific data matters. If there is no solution available, please share your question by posting it in the forum. Active participation is highly encouraged and requires no more than a one-time registration. The NEPS user community (and beyond) benefits from a broad participation.

→ https://forum.lifbi.de

Data Trainings

The Research Data Center offers a series of regular NEPS data trainings, conducted as online courses. Participation is free of charge. The courses consist of different modules, whereby single modules can be attended separately. While the *Basic Modules* provide knowledge on the general framework of the NEPS study and on how to access and work with the NEPS data plus documentation, the *Advanced Modules* address selected topics such as the handling of competence data, episode data, linked NEPS-ADIAB data, weights, etc. A schedule of current training courses together with information for registration can be found at our website.

→ www.neps-data.de > Data Trainings

NEPS Tutorials

The *Video Tutorials* provided at the website serve as visual support and supplement to the NEPS data trainings and the data documentation materials. They focus on general information about the NEPS and its cohorts or the data structure of the Scientific Use Files, but also on specific topics such as the combination of information from different datasets (“merging”) or the survey data linked to administrative data (“NEPS-ADIAB”). The tutorials are currently only available in German.

→ www.neps-data.de > NEPS Tutorials

NEPStools

NEPStools is a free to use collection of specific Stata commands. 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. *NEPStools* can be 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 respective website.

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

NEPSscaling

Plausible Values 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 (PV) is suitable for more precise inferential statistical tests in correlation and mean value analyses. The R package *NEPSscaling* enables data users to generate own PV with a background model adapted to the specific research question. The package is able to handle missing values in the background model and has additional features.

→ www.neps-data.de > Overview and Assistance > NEPSscaling

1.8 Contacting the Research Data Center

The Research Data Center at the Leibniz Institute for Educational Trajectories (FDZ-LIfBi) accounts for large parts of the NEPS data preparation and documentation, for the data dissemination, and for the user support including individual consultation. We appreciate any feedback in order to further improve our services.

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

E-mail: fdz@lifbi.de
Web: → www.neps-data.de > Research Data Center
Forum: → www.neps-data.de > Online Forum

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.

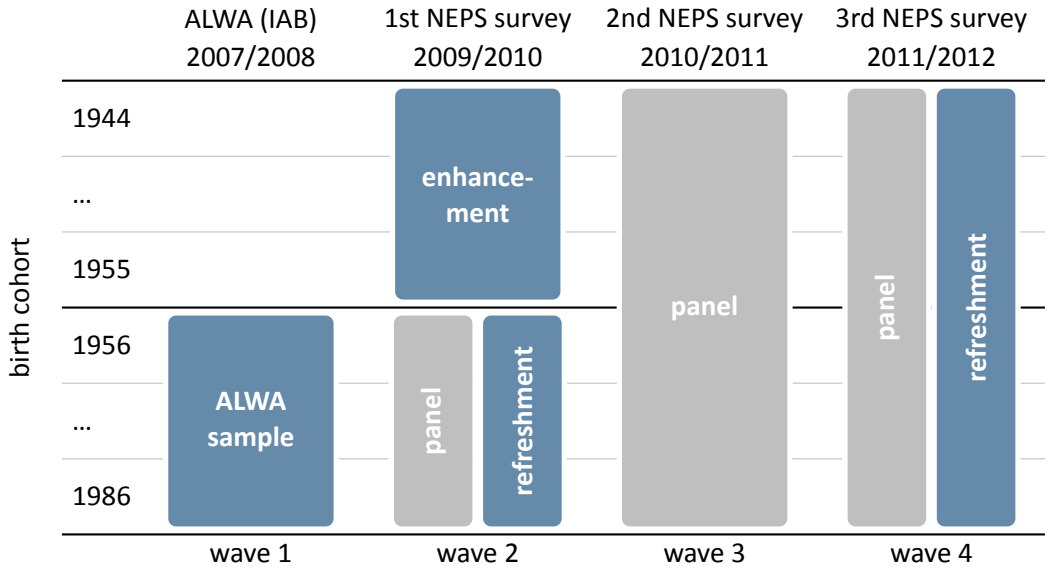

Figure 2: Longitudinal sampling design of Starting Cohort 6

The sampling design and its consequences for the derivation of sampling weights are described in Hammon et al., 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 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 – for some cohorts – *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 generated test scores such as the sum or mean of correct answers. Detailed descriptions on how these scores were estimated can be found in separate Technical Reports for the respective competence domains, available on the corresponding data documentation website (see Section 1.2). The individual and generated scores for students at *regular schools* are compiled in the dataset `xTargetCompetencies`.[1]

All competence data are structured in the so-called WIDE format, that is, all responses of a single respondent are placed in one row of the data matrix (see Section 4). As a consequence, variable names for competence scores follow a specific nomenclature. These conventions not only allow for the identification of the respective domain, the target group, the testing modus, and the kind of scoring – they also indicate the repeated administration of a test item in a different wave or starting cohort (see Section 3.2.2). Table 2 shows the schedule of competence measures in Starting Cohort 6 with domains by waves and test modes.

It should be noted that the table reflects the current planning status for testing the various competence domains in future survey waves. However, these plans may change over time. A list of all possible competence domains together with the respective abbreviation can be found in Table 5.

1 The Scientific Use File contains another competence dataset called `xPlausibleValues`, which contains exemplary variables with Plausible Values that were generated using the freely available R package *NEPSscaling* (see Scharl and Zink, 2022 and Section 1.7).

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]	2021/22 **Wave 14** (35-75 y.)	2024/25 **Wave 17** (38-75 y.)
Domain-General Competencies							
DGCF: Cognitive Basic Skills	dg	—	—	C	—	—	—
Domain-Specific Competencies							
Reading Competence[1]	re	P	P	—	C	—	C
Reading Speed	rs	P	P	—	—	—	—
Vocabulary: Listening Comprehension at Word Level[1]	vo	—	—	C	—	—	—
Mathematical Competence[1]	ma	P	—	—	C	—	C
Scientific Competence[1]	sc	—	P	—	—	C	—
Metacompetencies							
ICT Literacy[1]	ic	—	P	—	—	C	—

[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 2011* were tested in their reading competencies (re, rs); the target persons of the *initial sample plus refreshment and enhancement sample 2009* were tested in their scientific and ICT literacy competencies (sc, ic).
[4] **Wave 9**: The target persons of the *refreshment sample 2011* were tested in their reading competencies (re) only, while the target persons of the *initial sample plus refreshment and enhancement sample 2009* 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 in the current Scientific Use File, there is a short characterization in terms of field time, groups of respondents, number of realized cases, survey modes, and the survey institute(s) responsible for collecting the data. A more detailed insight into all aspects of the field work can be found in the wave-specific *Field Reports*, which are available on the website (in German only) as part of the data documentation.

→ www.neps-data.de > Data and Documentation > Starting Cohort 6 > Documentation

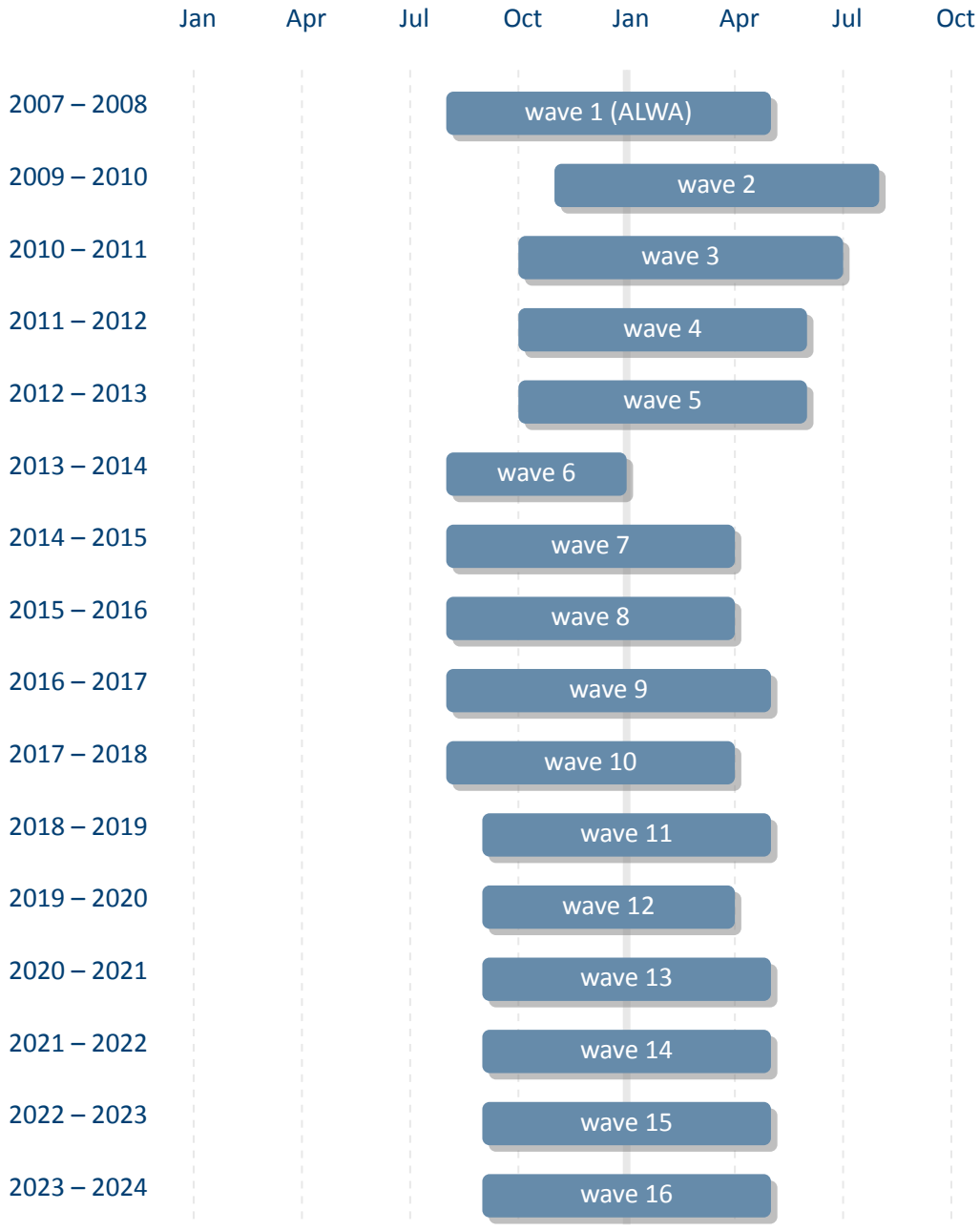

Figure 3: Panel progress of Starting Cohort 6

2.4.1 Wave 1: 2007/2008 (ALWA)

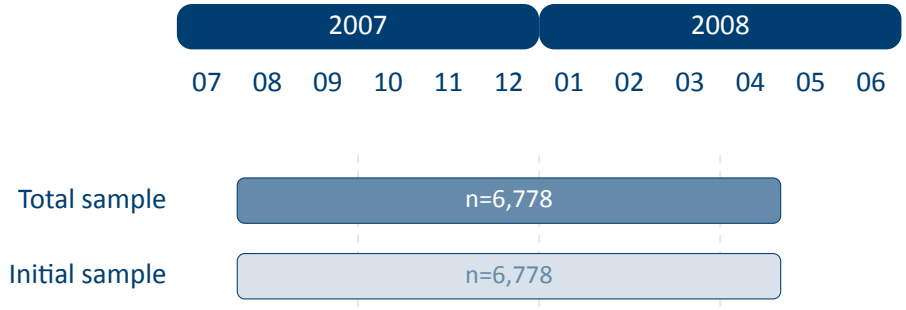

Figure 4: Field times and realized case numbers in wave 1

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Sampling 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 main survey)

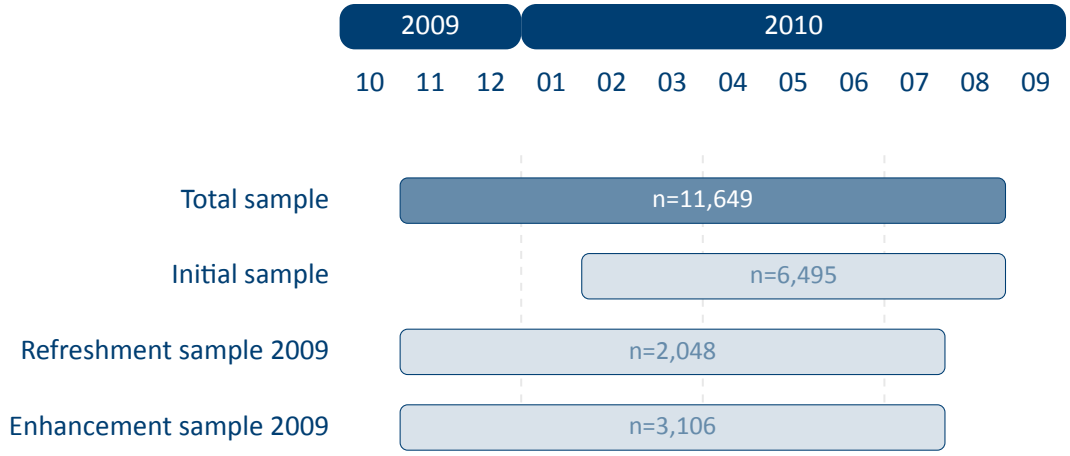

Figure 5: Field times and realized case numbers in wave 2

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Sampling 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

  Sampling 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 main survey)

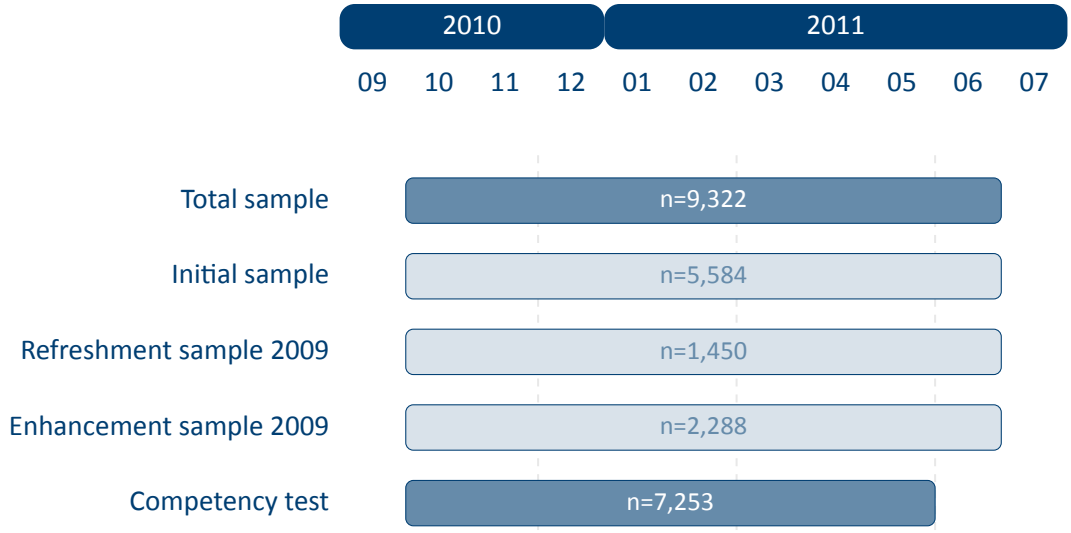

Figure 6: Field times and realized case numbers in wave 3

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

- **Competence testing** Randomized allocation of reading and mathematics competence tests to split sample (50% with three domains: reading competence, reading speed, mathematics | 50% with two domains: reading speed, mathematics OR reading speed, reading competence)
- **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 main survey)

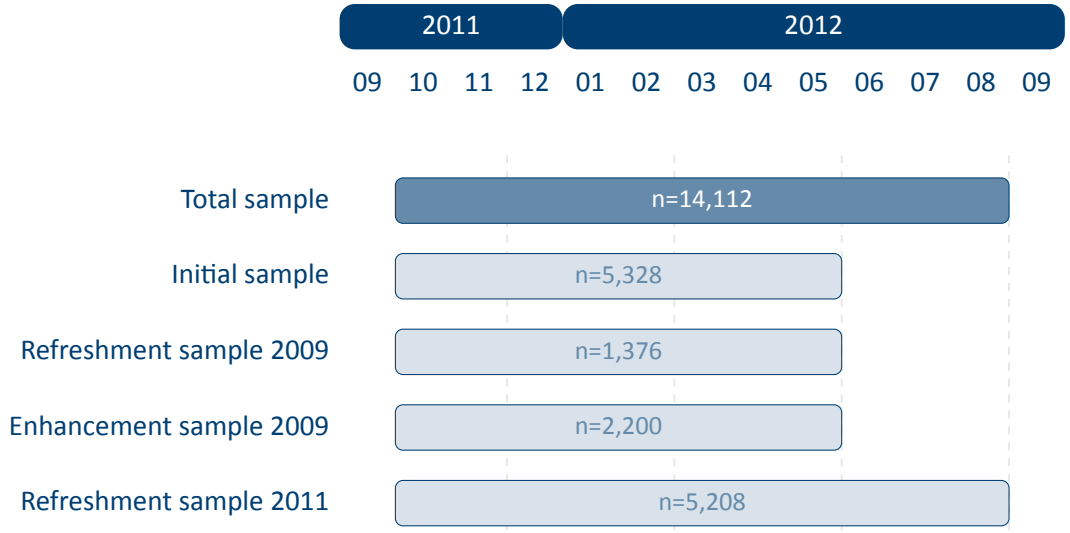

Figure 7: Field times and realized case numbers in wave 4

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

  Sampling 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 main survey)

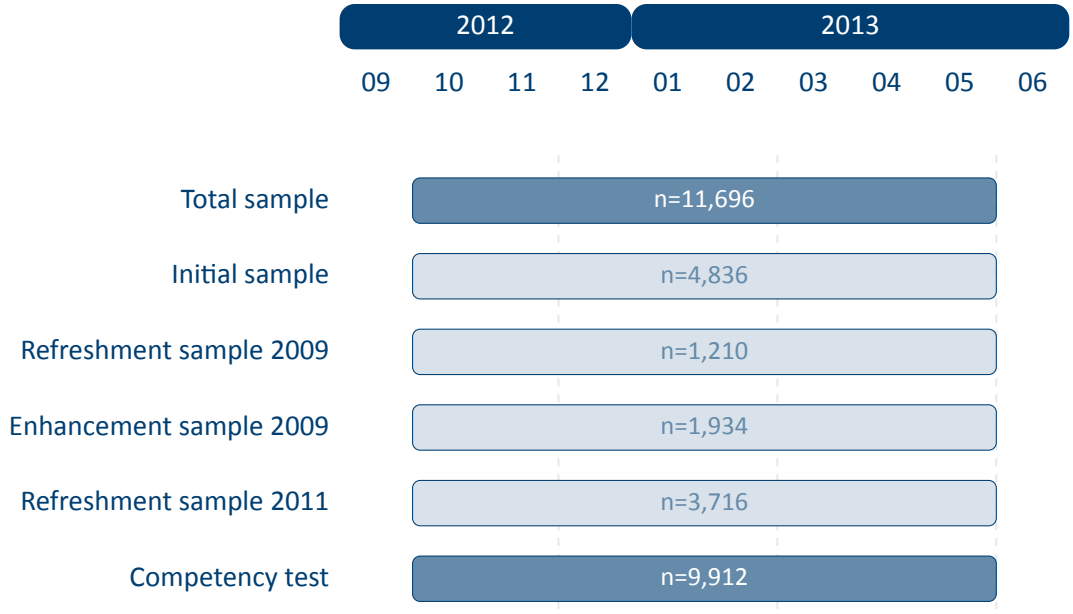

Figure 8: Field times and realized case numbers in wave 5

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **Competence testing** Respondents of the *refreshment sample 2011* were tested in reading competencies and reading speed; respondents of the three *other samples* were tested in scientific competencies and ICT literacy competencies
- **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 main survey)

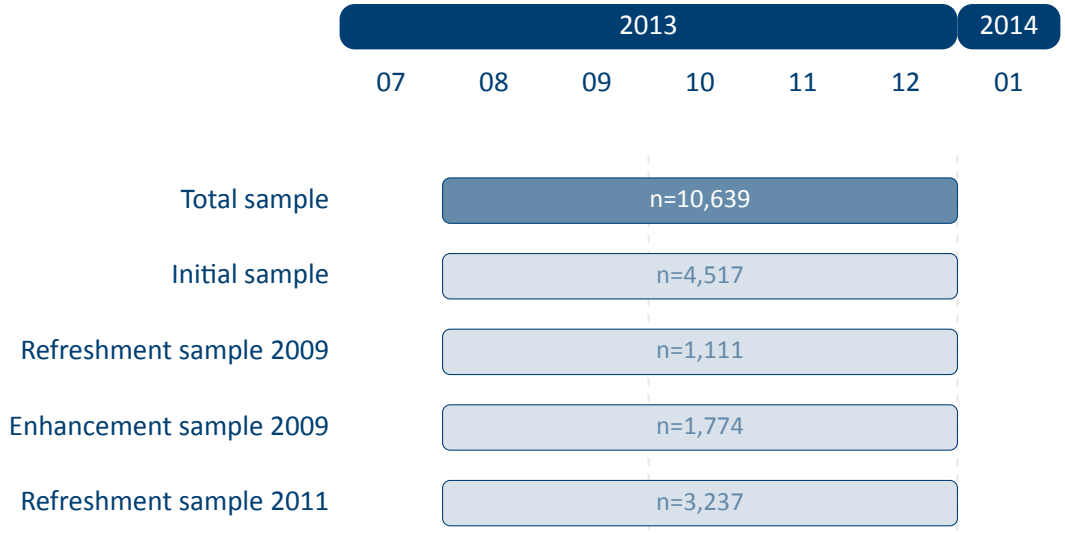

Figure 9: Field times and realized case numbers in wave 6

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **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 main survey)

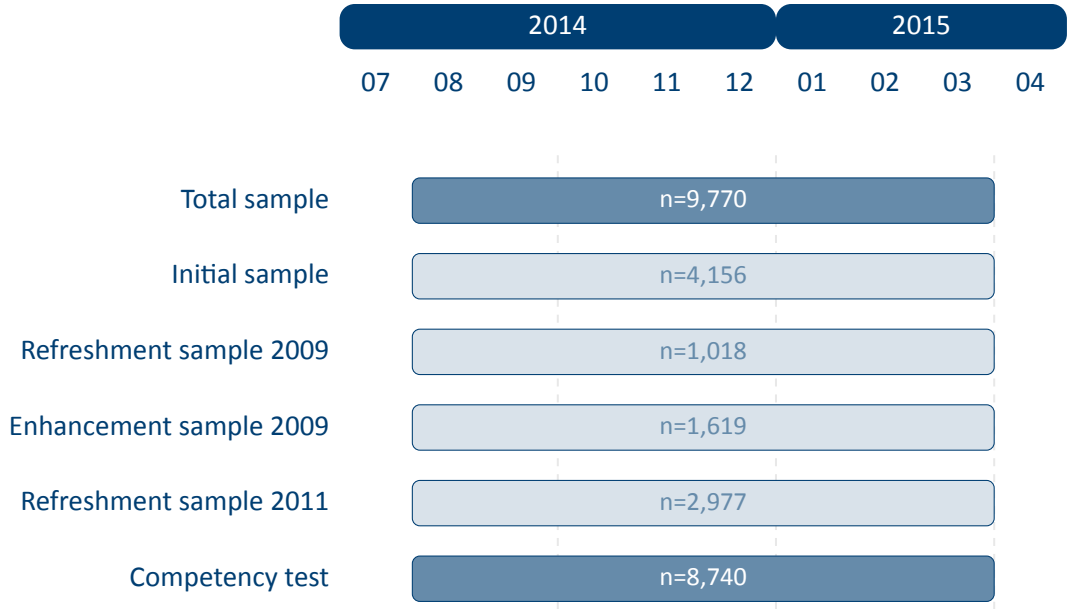

Figure 10: Field times and realized case numbers in wave 7

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **Competence testing** Respondents of all samples were tested in cognitive basic skills (DGCF) and vocabulary
- **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 main survey)

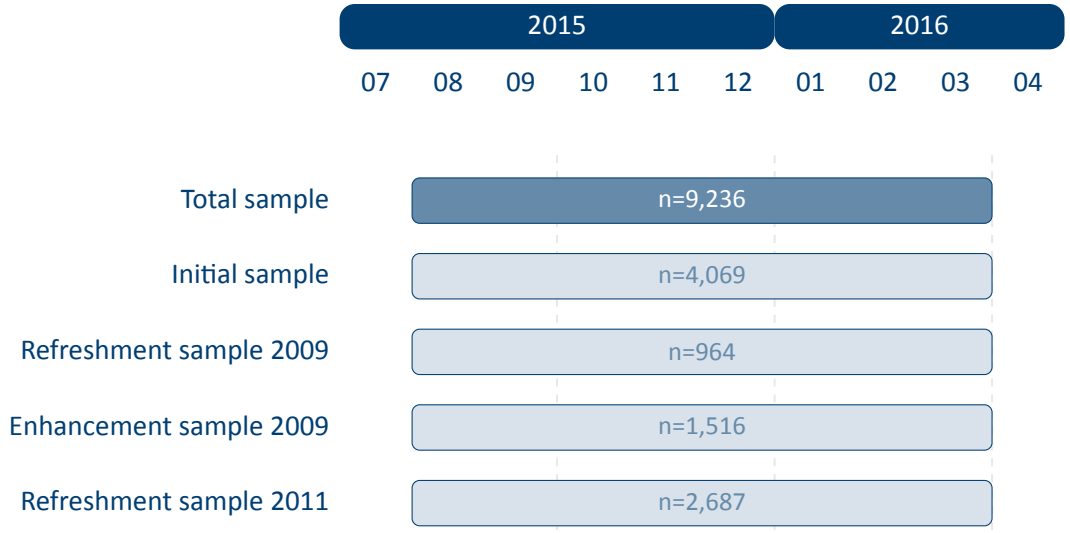

Figure 11: Field times and realized case numbers in wave 8

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **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 main survey)

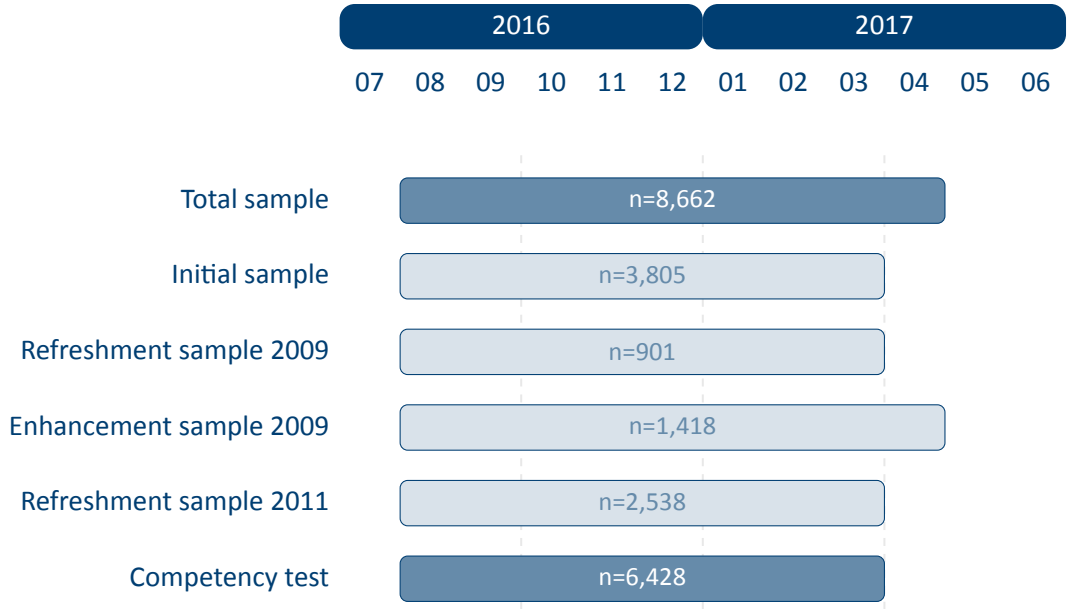

Figure 12: Field times and realized case numbers in wave 9

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **Competence testing** Respondents of the *refreshment sample 2011* were tested in reading competencies only; respondents of the three *other samples* were tested in reading competencies and mathematics
- **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 main survey)

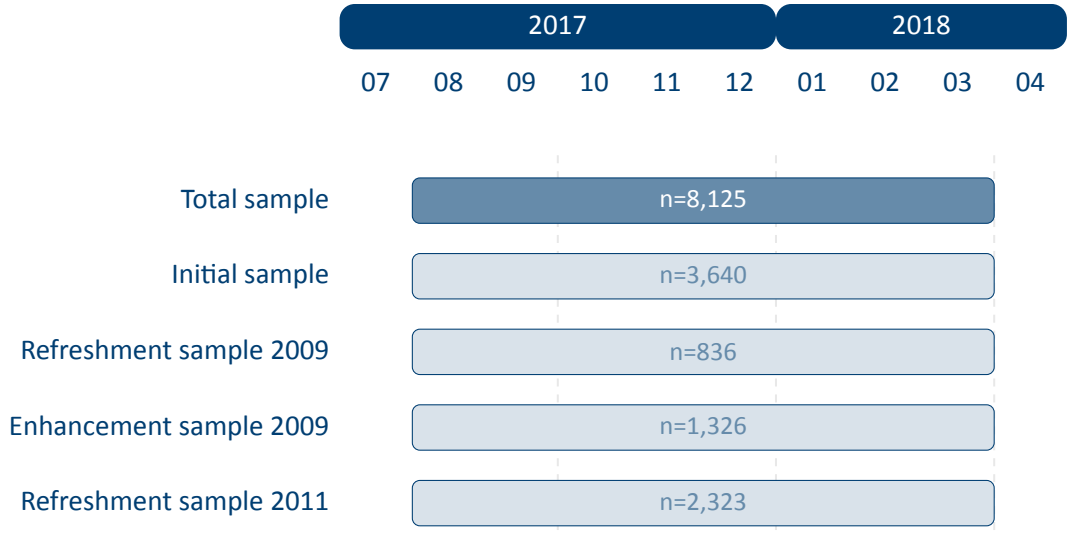

Figure 13: Field times and realized case numbers in wave 10

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **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 main survey)

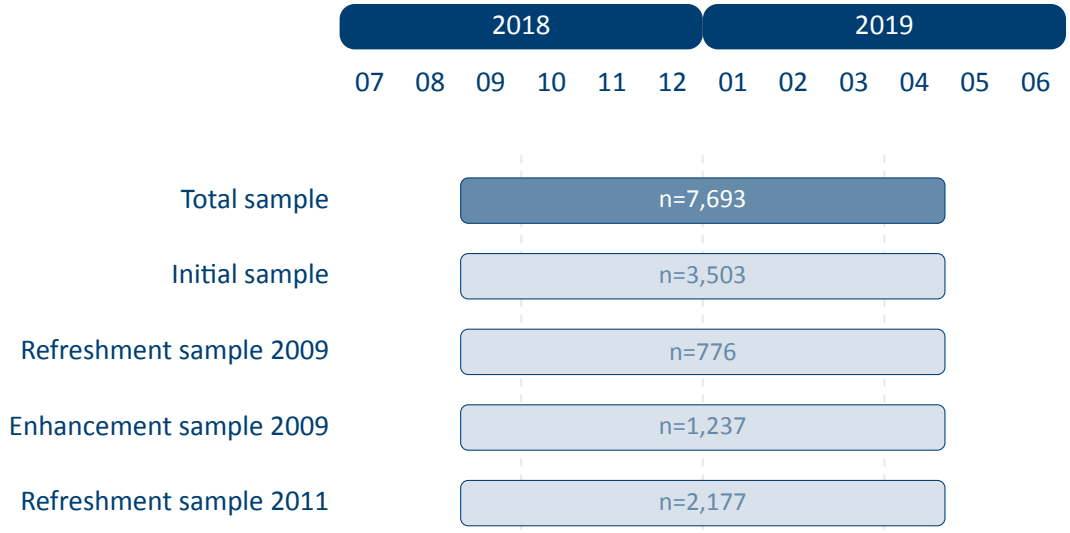

Figure 14: Field times and realized case numbers in wave 11

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **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

2.4.12 Wave 12: 2019/2020 (11th NEPS main survey)

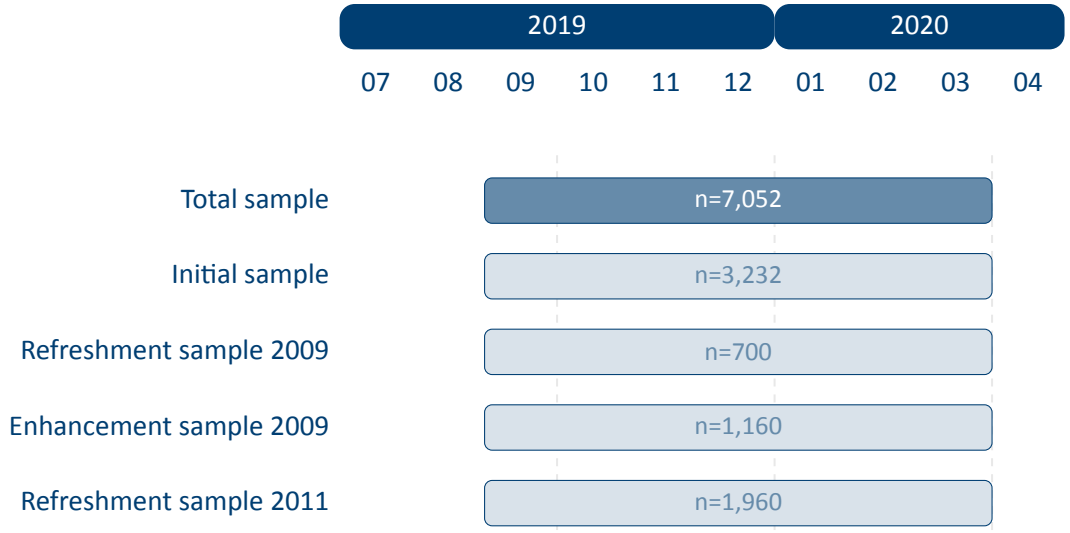

Figure 15: Field times and realized case numbers in wave 12

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **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

2.4.13 Wave 13: 2020/2021 (12th NEPS main survey)

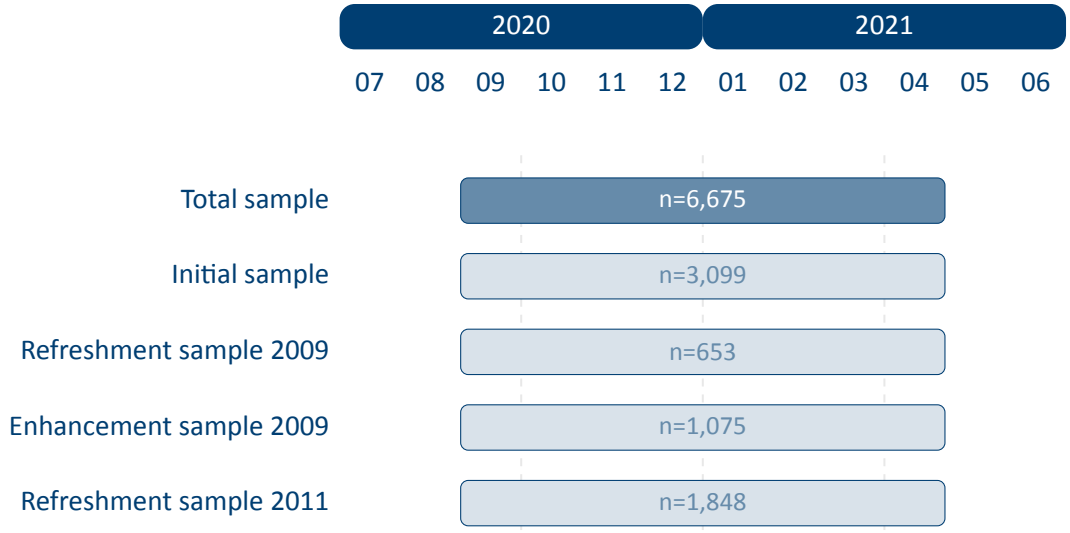

Figure 16: Field times and realized case numbers in wave 13

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **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

2.4.14 Wave 14: 2021/2022 (13th NEPS main survey)

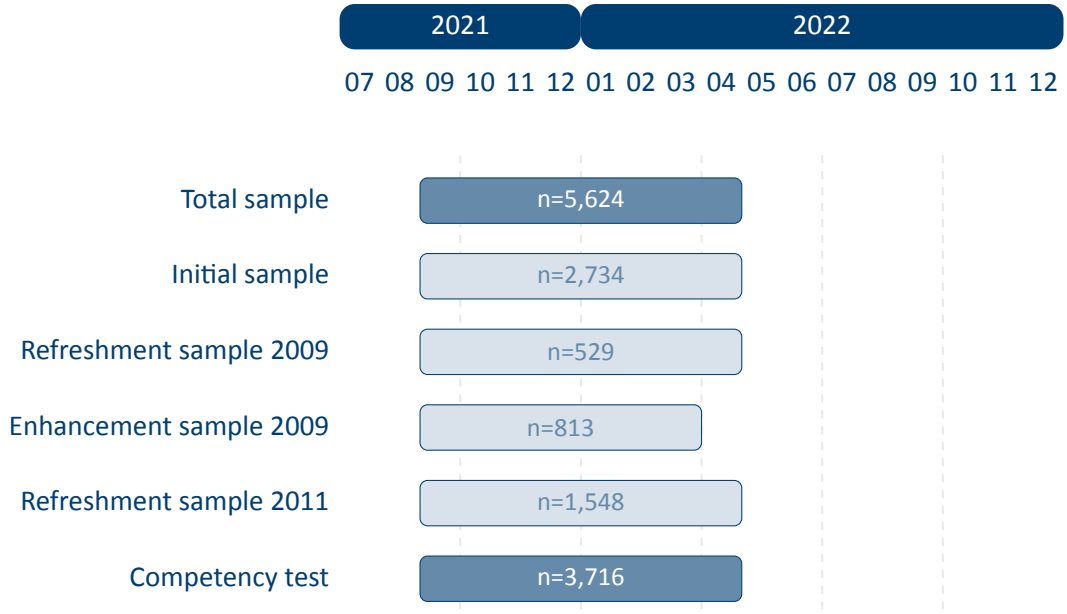

Figure 17: Field times and realized case numbers in wave 14

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

- **Competence testing** Respondents of all samples were tested in scientific competencies and ICT literacy competencies
- **Mode of survey** Computer-assisted telephone interviews (CATI) for the life history survey, additional computer-assisted personal interviews (CAPI) for competency testing; and additional online survey (CAWI)
- **Data collection** infas – Institute for Applied Social Sciences, Bonn

2.4.15 Wave 15: 2022/2023 (14th NEPS main survey)

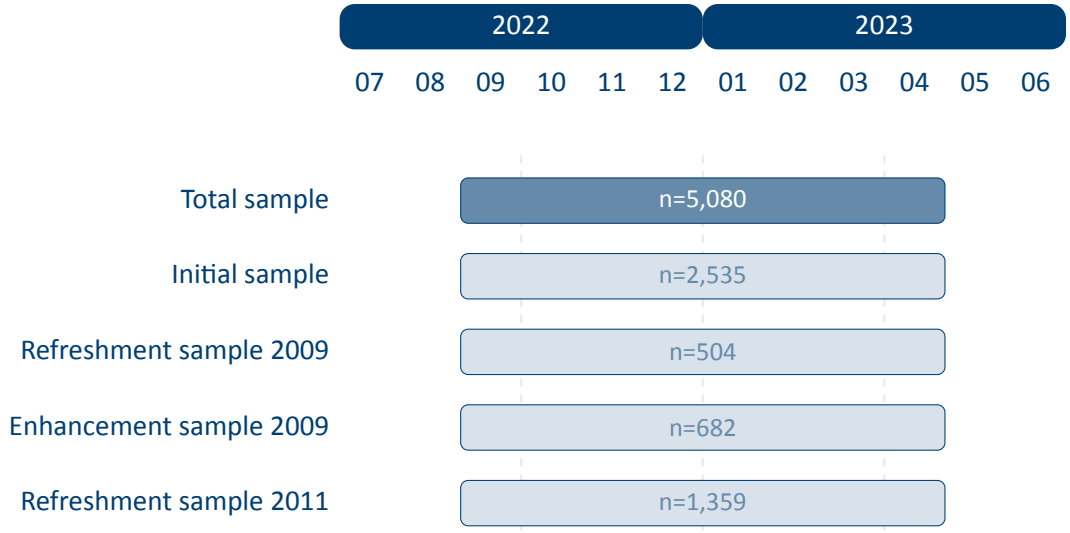

Figure 18: Field times and realized case numbers in wave 15

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

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

2.4.16 Wave 16: 2023/2024 (15th NEPS main survey)

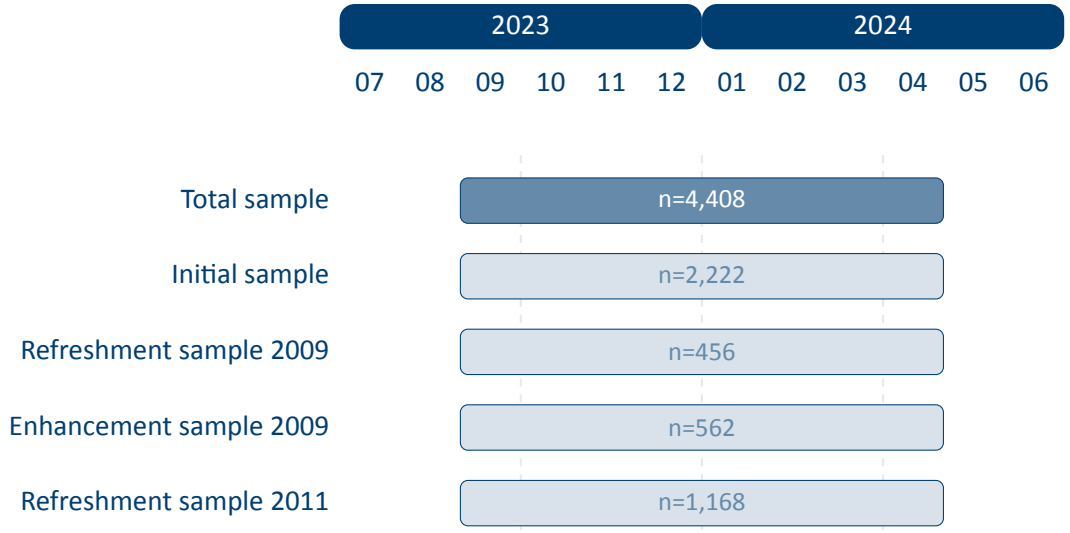

Figure 19: Field times and realized case numbers in wave 16

- **Respondent groups**

  Initial sample Panel start 2007/08 (ALWA)

  Target persons Adults of birth cohorts 1956 to 1986

  Refreshment sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1956 to 1986

  Enhancement sample 2009 Panel start 2009/10

  Target persons Adults of birth cohorts 1944 to 1955

  Refreshment sample 2011 Panel start 2011/12

  Target persons Adults of birth cohorts 1944 to 1986

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

3 General Conventions

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

1. **The first paradigm is that of unaltered data.** Wherever possible, the content of the original data is neither changed nor modified for the Scientific Use File. This paradigm is the basis for preserving the full research potential of the data collected. Therefore, no corrections are made during data preparation in order to "establish" any content validity. This means that the Scientific Use File may contain implausible values unless appropriate checks were already implemented in the survey instrument. Only in rare cases, in which the responsible developers of a variable request the removal of clearly implausible information in the data, these values are replaced by the special missing code "implausible value removed" (-52, see Table 6). The only systematic exception to this paradigm concerns the recoding of open-ended responses that can be subsequently assigned to a closed response category for the respective question (see Section 3.4 for details). The NEPS Scientific Use Files are provided with a special dataset `EditionBackups` that contains backup information for all content that has been modified by such recoding procedures (see Section 4.5.5 for details).
2. **The second paradigm is to integrate the data as much as possible without compromising the usability of the Scientific Use File.** For this purpose, the original data – some of which comprise over a hundred individual datasets – are combined into a few dozen panel and episode datasets (see Section 4.3 and Section 4.4 for details). This strategy is based on the assumption that it is far more convenient for the vast majority of data users to reduce already integrated data for a specific analysis than to correctly merge the information relevant for the analysis from scattered source data themselves.

There are additional conventions for the data structure of all NEPS Scientific Use Files. The aim of this overall structuring is to ensure a maximum of consistency between the data of all NEPS cohorts. Thus, a researcher who is familiar with the data logic of a particular cohort should be able to immediately recognize this structure when starting to work with data from another cohort. The conventions described in the following sections apply equally to Starting Cohort 6, although some of the examples refer to other NEPS cohorts.

3.1 File names

The naming of the data files in the NEPS Scientific Use Files is determined by a few rules that are summarized in Table 3. The four different elements of a dataset name are each separated by an underscore (_).

Table 3: Naming conventions for NEPS data files

Element	Definition
SC[1–8]	**Indicator for the starting cohort and launch year of the panel** 1 = Newborns (2012) 2 = Kindergarten (2011) 3 = Fifth-grade students (2010) 4 = Ninth-grade students (2010) 5 = First-year university students (2010) 6 = Adults (2007) 8 = Fifth-grade students (2022)
[filename]	**Meaning of the file name** *Prefix*: x = cross-sectional file; sp = spell file; p = panel file *Keyword*: indicates the content of the file (e. g., pTarget contains panel data with regard to the target persons; 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
[#]-[#]-[#]	**Indicator for the release version** *First digit*: the main release number is incremented with every further survey wave available; e. g., the first digit "10" implies that data of the first ten waves are included in the Scientific Use File *Second digit*: the major update number is incremented with every bigger change to the Scientific Use File; major updates affect the data structure (updating of analysis syntax 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 or labels (syntax updating not necessary)

For instance, SC6_CohortProfile_D_16.0.0.dta refers to the generated *CohortProfile* dataset of *Starting Cohort 6* in its *Download* version of the Scientific Use File release *16.0.0*.

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 20. It has to be noted that a separate nomenclature is used for variables from competence measurements.

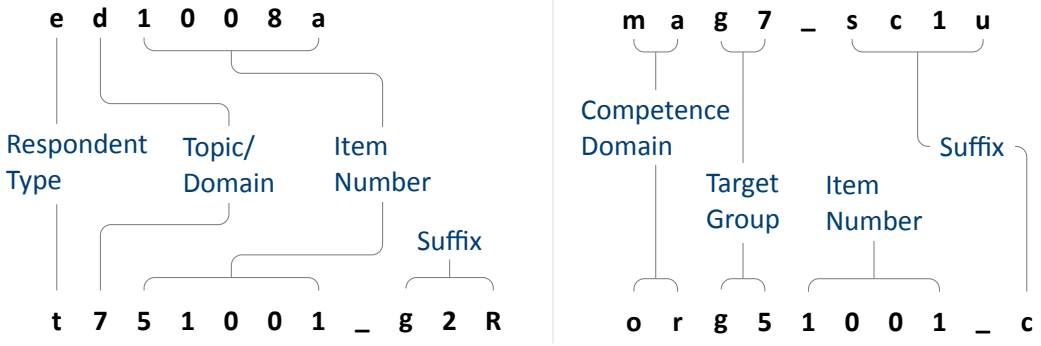

Figure 20: General variable naming (left) and competence variable naming (right)

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/legal guardian of target person `c` = Cohabiting partner of the target child's parent `e` = Educator/childminder/teacher `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 m = Corona variables 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. 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 (e. g., `_g1`). However, there are three types of generated variables that assign specific meanings to digits, namely regional classifications, occupational variables, and education variables.

The **regional classifications** are based on the Nomenclature of Territorial Units for Statistics (NUTS) and refer to units within Germany:

- 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 **occupation and prestige indices** are (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)

Generated variables for the **classification of highest educational qualification** are:

- g1: ISCED-97 (Int. Standard Classification of Education 1997, *not in Starting Cohort 8*)
- g2: CASMIN (Comparative Analysis of Social Mobility in Industrial Nations)
- g3: Years of Education (derived from CASMIN)
- g4: ISCED-2011 (International Standard Classification of Education 2011)

- *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 `_v1` for the data before the question was modified for the first time, `_v2` for the data before the question was modified for a second time, and so on.

- *Harmonized variables:* The suffix _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 _w1, _w2, …, _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 (t*_R) or in combination with other suffixes (t*_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 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 measurements 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 all specifications of a competence variable name.[2]

2 The variables generated from the competence data in the additional dataset xPlausibleValues follow the same naming logic – with a uniform suffix _pv# after the first two parts of the naming convention. Please note, that this dataset will be added to the Scientific Use File of Starting Cohort 8 at a later date.

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
cl	Civic Literacy
dc	Digital competence
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: Attention abilities
gk	General knowledge
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/discourse level
lk	Early knowledge of letters
ma	Mathematical competence
mb	Mathematical competence from IQB-BT
md	Declarative metacognition
mp	Procedural metacognition
ni	Nonverbal reasoning
nr/nt	Native language Russian/Turkish: Listening comprehension
on	Blending of onset and rimes: Phonological awareness
or	Orthography
rb	Reading competence from IQB-BT
re	Reading competence
ri	Rimes: Phonological awareness
rs	Reading speed
rx	Early reading competence
sc	Scientific competence
st	Scientific thinking: Science propaedeutics
vi	Verbal reasoning
vo	Vocabulary: Listening comprehension at word level

(...)

Table 5: (continued)

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

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 most competence domains, these item numbers only indicate 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) [a] [b]
_sc2	Standard error for the WLE [b]
_sc3	Sum score
_sc4	Mean score
_sc5	Difference score (for procedural metacognition)
_sc6	Proportion correct score (for procedural metacognition)
_sc8	Test stop
_sc9	Basal/Ceiling set (for vocabulary), number of practice items (for digit span)
_p	Maximum value for an item (only in xDirectMeasures of Starting Cohort 1)
_b	Minimum value for an item (only in xDirectMeasures of Starting Cohort 1)
_m	Mean value for an item (only in xDirectMeasures of Starting Cohort 1)
_s	Sum value for an item (only in xDirectMeasures of Starting Cohort 1)
_n	Number value for an item (only in xDirectMeasures of Starting Cohort 1)

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

[b] 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).

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 single 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 in competence variable names and their meanings.

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 (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 time of current item administration.

The following example illustrates this logic: The competence variable `vok10067_sc2g1_c` is a vocabulary item (`vo`) initially measured during the first kindergarten survey wave of Starting Cohort 2 (`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 B.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, associated filter conditions, as well as other meta information. All extended features can be accessed directly within the 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.[3]

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: **The meta information available in a dataset always corresponds to the most recent 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 the meta information of such a variable in the dataset is requested, 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 original 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, one has to 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. The three main types of missing codes are summarized in Table 6 and described below.

3 Some variables are based on different versions of a question formulation, depending on previous filter settings (e.g., stay abroad of at least one month for training/ for study/ for doctorate etc. [ts15223 in SC3, SC4, SC6] according to the current type of training of the respondent [ts15223]). Only the first version is provided in the extended characteristics, but an additional note is displayed: "ATTENTION: There are several question formulations for this variable – see the Variable Search or Survey Instruments for detailed information!"

Table 6: Overview of missing codes

Code	Meaning	Note
Item nonresponse		
–94	not reached	only relevant for instruments with time restrictions (e. g., competence test measures)
–95	implausible value	assigned by the survey agency (e. g., multiple answers to a one-answer question in PAPI)
–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	e. g., question not answered, empty field (PAPI) or missing information despite soft response constraint (CAWI/CASI)
–91	survey aborted	question not reached due to early termination of the instrument (CAWI/CASI)
–92	question erroneously not asked	question not asked by mistake (CAWI/CATI)
–93	does not apply	as default answer option to the question
–99	filtered	filtered out question (other than CATI/CAPI)
.	*system*	filtered out question (CATI/CAPI)
Edition missings (recoded into missing)		
–52	implausible value removed	only in exceptional cases (at the request of 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)
–56	not participated	in case of unit nonresponse (only used in certain datasets)

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

- Typical cases of item nonresponse are the "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).
- If either the respondent 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).
- If a respondent has terminated the survey in an online mode (CAWI/CASI) before reaching the end of the instrument or if the survey session has been aborted by timeout, the missing code "survey aborted" (–91) is assigned to all questions that were not answered at the end of the questionnaire.
- 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 or in CAWI/CASI interviews when a respondent has refused to answer a question despite soft response constraint.

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 requested 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 the sense that target persons for whom no survey data at all are available 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-ended 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. Therefore, the recoding 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 working units 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 scheme 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 21 shows the derivation paths for several **occupational scales and schemes** provided in the NEPS.

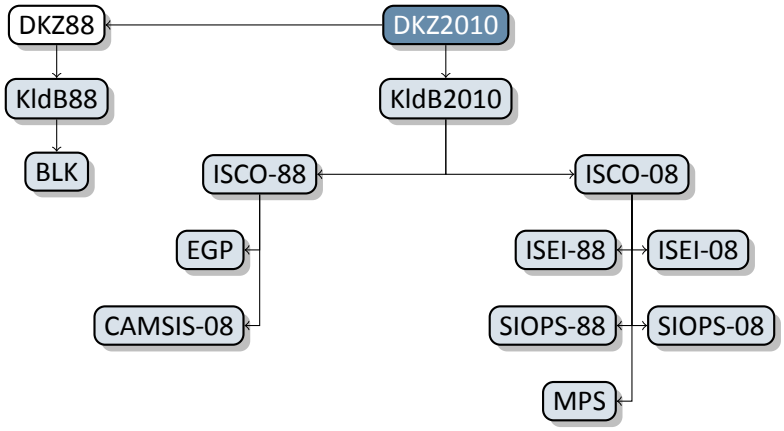

Figure 21: Derivation paths for several occupational scales and schemes provided in the NEPS

A detailed description of the standard derivations for **educational attainment** (ISCED, CASMIN and Years of Education) can be found in the corresponding documentation report by Pelz, 2025.

4 Data Structure

4.1 Overview

The longitudinal NEPS study is a complex research database. It is the result of extensive data edition processes with the aim of organizing the information in a well-structured, reproducible and user-friendly way, while at the same time preserving a maximum level of detail in 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. Data files containing panel information from several waves are denoted with a ***p*** at the beginning of the file name. For example, the `pTarget` file contains information from the target persons' interviews with one row in the dataset representing the information of one individual in one wave (see Section 4.3).

This convention does not apply to all longitudinal information in the Scientific Use File. There are competence measurements that were repeatedly carried out with the same target persons. Since the content of competence tests varies over time, the corresponding data is structured in *WIDE format* (see Section 3.2.2). Such cross-sectionally structured data files with one row representing information of one individual from all waves are marked with an ***x***.

Another type of longitudinal data structuring refers to episode or spell data. For the information collected prospectively and retrospectively by using iterative question sets, the Scientific Use File provides life area-specific spell datasets. They are marked by a preceding ***sp***. An example is `spEmp`, informing about current and former episodes of employment (see Section 4.4).

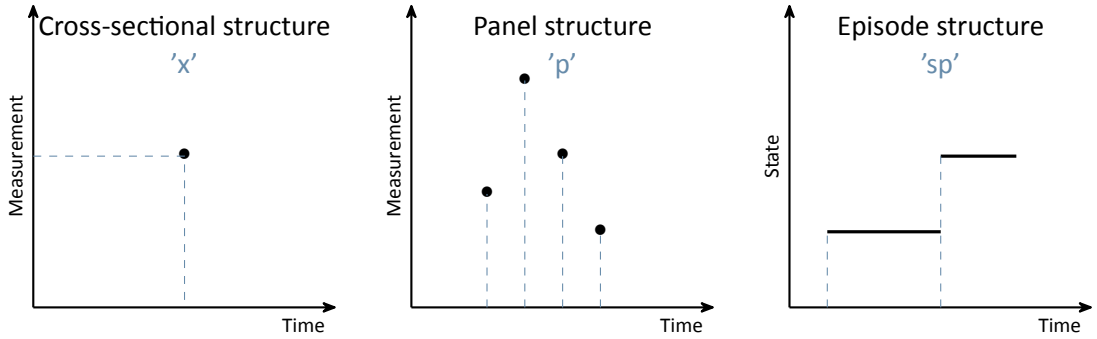

Figure 22: Different types of data structures

In addition to the interview and competence and episode data surveyed from the respondents, there are so-called paradata and derived information available. The respective data files can be identified by the leading capital letter in the name (e. g., `CohortProfile`, `TargetMethods` or `Weights`; see Figure 24).

4.2 Identifiers

The multi-level and multi-informant design of the NEPS – together with the provision of information in several data files – requires the use of multiple identifiers, especially for merging information from different datasets. The syntax examples in Section 4.5 demonstrate typical applications for linking information from the respective file to information from other files. The following identifier variables are particularly relevant in Starting Cohort 6:

ID_t identifies a target person persistently. The variable `ID_t` is unique across waves and samples; it is also used uniquely in each of the NEPS starting cohorts.

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

splink uniquely identifies episodes/spells across all datasets within each person. It is used to link biographical data from generated or single episode datasets.

There are further identifier variables available, for example to indicate a target person's membership in a particular test group (`ID_tg` in `CohortProfile`, not applicable to all starting cohorts) or to indicate the interviewer who conducted the respective interview (`ID_int` in the `Methods` datasets). These identifiers are less relevant for the merging of information from different datasets and negligible for most empirical applications.

4.3 Panel data

In general, all information from the latest survey wave is appended to the already existing information from previous waves (as far as possible). This kind of data preparation 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 NEPS panel data, the following points are important to be considered:

- A row in the dataset contains the information of one respondent from one survey wave.
- More than one variable is needed to identify a single row for uniquely selecting and merging information from different datasets. Usually, **`ID_t` and `wave`** are the relevant identifiers.
- Although not all questions were administered in each survey wave, the data structure contains cells for all variables and waves. If no data is available, e. g., because a question was not asked in a wave, the corresponding cells are filled with a missing code (see Section 3.3).
- If information about a variable has been repeatedly surveyed from one individual across multiple waves, the corresponding data is stored in multiple rows in the dataset.

The *LONG format* is usually the preferred data structure for the analysis of panel information. However, cross-sectional information is often required as well in analyses, e. g., because it depicts time-invariant characteristics or was collected only once for other reasons. In most scenarios, the relevant set of variables might not have been measured in a single wave. Therefore, the 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. The two typical procedures are:

- 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-specifically identifiable. The result is a dataset with a cross-sectional structure in which the information of one 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 of a variable are copied into the unobserved cells of this variable. For example, if the place of birth was only surveyed in the first wave, the corresponding value can be copied into the respective cells of the respondent's other waves. This method is particularly useful for time-invariant variables (e. g., country of birth, language of origin), that are usually collected only once in a panel study.

4.4 Episode or spell data

A major focus of the NEPS is on recording biographical trajectories as completely as possible. Depending on the NEPS cohort, different areas of the life course are surveyed as so-called ***episodes***. These areas range from school history, education and employment history to household-related histories (e. g., partnership, siblings, children). The retrospective collection of biographical information – What has happened in a certain area of life since time X or since the last interview? When did an episode start and when did it end? What are the characteristics of this episode? – is very demanding and the resulting data material is rather complex. Episode or spell data are therefore a particular challenge for the analysis. The following explanations help to better understand this data format and its processing in order to handle it in a meaningful and appropriate way. The information applies equally to all NEPS cohorts, even if the specific data material differs from starting cohort to starting cohort according to the surveyed biographical areas. Information on how to work with the spell data can also be found in the video tutorials offered and in the online forum (see Section 1.2). The "Special Issues" section in Section 5 provides a comprehensive insight into the episode modules implemented.

In episode data, there is one row for each episode that was captured during an interview. Usually, a start and an end date describe the duration of the episode. The remaining variables in spell datasets provide additional information about that episode. These "descriptors" are related to the particular episode and fill it with content, so to speak. It means (especially for time-variant variables like education or occupation or employment) that the respective values indicate the status *at the time of the episode*, which is not necessarily the current status valid nowadays (or at the time of the interview). To give an example, in the dataset `spEmp` there is a period of time for a particular respondent during which she or he worked in a particular job without interruption. If this person changed to a new job, this defines a new episode stored in a new data row. Further changes in this context may also lead to new episodes, e. g., a change of the employer or the conclusion of a new employment contract – but not if the salary, working hours or other characteristics (possible descriptors) of the respective job change. Episodes can be understood as the smallest possible units of one's life history, in this case the employment biography. Several relevant changes in such a biographical area are reflected in several new data rows.

To make this clear: The number of episodes is per se independent of the survey wave. During an interview (one wave) there might be a number of episodes recorded (several rows) or no episode at all (no row). The dates given for an episode relate to that episode, whereas the wave indicator relates to the interview date. The two can overlap, but do not have to. Data users should consider both entities – `spell` and `wave` – to be independent of each other. In exceptional cases, it might be important to know when the information about an episode was collected. Beyond that, however, the variable `wave` can be ignored in the episode data. In particular, the `wave` variable should **not** be used to merge episode data with panel data in the *LONG format*. Since episode data may contain multiple (or no) rows per survey wave and target ID, and panel data contain exactly one row for each survey wave and target ID, such a merge will result in converting the panel data to an episode structure. The result of this kind of transformation is no longer analyzable in a meaningful way. A better approach is to aggregate the episode data to one piece of information either for each interview date (e.g., number of jobs since the last interview) or for the entire life course (e.g., highest educational attainment), so that only one row per survey wave and respondent is left for the merging process.

In addition to (time-dependent) episode data such as jobs, which we call *duration spells*, there are two other types of episode spells in the NEPS data:

- Occurring events or the transition from one state to another (e. g., change of marital status, change of educational level) are recorded in *event spells* with one row describing one state.
- The existence of children, partners, etc., is recorded in *entity spells* with one row per entity.

Regardless of the type of episode, at least two variables are necessary to identify a single row in the data file, namely the respondents' identifier `ID_t` and an numerator for the episode, event or entity such as `spell` or `child`. More detailed information on the available identifier variables can be found on the respective data file descriptions in Section 4.5.

4.4.1 Edition of the life course

The life course data in all NEPS starting cohorts mainly consists of information on episodes of school attendance, participation in vocational preparation measures and vocational training, university education, as well as of compulsory or voluntary services, employment and unemployment, and parental leave. We refer to these activities as *main activities*. The episodes are grouped by type and recorded in separate modules. The aim of this recording is to capture chronologically complete life histories across key biographical areas of the respondents. This goal is supported by two data-guided measures:

Data edition during the interview

The first step takes place during the interview. The episodes reported by the respondent are summarized by the instrument and put into a chronological order. They are then checked for gaps and overlaps. Their clarification is made cooperatively by the interviewee and the interviewer with the help of the so-called *check module* (Hess et al., 2012).

If chronological *gaps* are identified, they are subsequently closed by recording additional episodes with regard to the above-mentioned main activities. If there is no suitable main activity for a gap, the respondent can close it with a "gap activity". Moreover, gaps can be filled by adjusting the start and end dates of the episodes between which the gap exists(see also Section 5.3.11).

Chronological *overlaps* of episodes are also reviewed together with the respondent. This may lead to an adjustment of the dates of the episodes involved in the overlap. For imprecise or missing date information, estimates are calculated where there is reasonable evidence. For example, the rather vague specification "summer" for the starting month of an episode is replaced by the value 7 for "July". This allows episodes with incomplete dates to be included in the plausibility test during the interview (Ruland et al., 2016; Matthes et al. 2005, 2007).

Data edition after the interview

Despite extensive review during the interview with largely complete and chronologically consistent life histories as a result, there might still be minor inaccuracies at the end. For example, one-month overlaps of episodes are not displayed or processed in the check module. The same applies to gaps of up to two months between consecutive episodes. Also, the review can be interrupted or skipped at the request of the respondent. Therefore, a second step of automated editing of biography information takes place after the end of the interview (Künster 2015a, 2015b). The results of this concern the `Biography` dataset only. In the spell datasets for the different life domains, the information provided by respondents during the interview with regard to the start and end dates of episodes remains unchanged.

- Firstly, one-month overlaps of episodes are removed. Such an overlap occurs when the end date of a previous episode is identical to the start date of the following episode, i.e. the same month was specified. In this case, the end date of the previous episode is shortened by one month. The condition for this is that the previous episode is longer than one month. If this

condition is not met, the start date of the following episode is shortened by one month. If both episodes have a duration of only one month, the dates remain unchanged.

- Secondly, one- and two-month gaps between consecutive episodes are closed. For a one-month gap, the end date of the previous episode is extended by one month. For a two-month gap, the start date of the following episode is additionally moved forward by one month.
- Finally, chronological gaps in the life history that are larger than two months are closed by inserting new episodes into the `Biography` file. These artificial episodes, labeled as "data edition gap" in the variable `sptype`, close larger gaps completely.

4.4.2 Revoked episodes

To make it easier for respondents to answer the life history modules and to minimize recall errors, information on episodes from previous interviews is preloaded. This information can be subsequently revoked during the current interview. The spell datasets also contain these revocations or contradictions (variables `disagint`, `disagwave`). The reasons for that are manifold; they primarily depend on the information presented to the respondent in order to recall an episode (the exact wording of the episode data collection can be seen in the questionnaires).

Subsequently revoked episodes are marked accordingly in the respective dataset. The information collected again in the current interview is additionally stored as a new episode in the corresponding (more recent) survey wave. That updated episode is **not** marked as a corrected spell. The identification of related spells – original information plus its correction in the subsequent survey wave – is up to the data user. It should be noted that practically all corrected episodes are *left-censored*. This is because it is technically not possible to specify a start date for an episode in the interview that precedes the last interview. The earliest start date is for episodes that began on the interview date of the last survey.

There is also the possibility of revoking a reported episode still during the interview. The *check module* (see Section 4.4.1) is also used for this purpose after all biographical information has been recorded. It ensures that the life course is captured as completely and consistently as possible. As part of the plausibility review within the interview, there is the option for respondents of correcting and revoking previously reported episodes. The identification of episodes that were revoked is possible by the variable `spms` "check module: type of event" and the value -20 "episode revoked in check module". The addition of new episodes in the check is indicated in the "episode mode" variable such as `ts23550=4` in `spEmp`).

4.4.3 Subspells and harmonization of episodes

When working with NEPS spell data, there is one important circumstance to consider: Biographical episode data are collected retrospectively. During an interview, respondents are asked about all episodes that have occurred since the last interview (or the first interview, since birth or a certain age). If an episode ended before the time of the current interview, the respondent provides an end date and the spell is completed. Challenges occur when the episode has not ended at the time of the interview, i.e., it is still ongoing.

Such an episode appears in the dataset as *right-censored*. In the next interview, this episode is then preloaded in the course of the "dependent interview" in a way that the respondent can report whether it has been finished in the meantime or whether it still continues. Technically, this results in multiple rows, which can be distinguished by the variable `subspell`:

- first data row with initial information about an episode (right-censored) reported in survey wave x (`subspell=0` if this is the only subspell for that episode, `subspell=1` if there are other subspells from later waves)
- second and further data rows for the continued episode, reported in subsequent survey waves x+ (`subspell=2`, `subspell=3`, etc.)

To make it easier for data users to work with these spread episode data, they are additionally summarized in a separate data line (record) according to defined rules. This data line reflects the most current or relevant information of the entire episode, depending on the harmonization rule applied (see below in this chapter). This ususally means, that for completed episodes the information valid at the end of the episode is selected and for episodes that were not yet completed at the time of the last interview, the information valid at the time of the last interview is selected. We call this process of summarizing information about an episode from different survey waves ***episode harmonization***. It is described in detail below.

An episode is defined by the assignment to a respondent (`ID_t`), by the type (e. g., training episode), by the episode identificator (`splink`, which typically consecutively numbers episodes of the same type for a case), and by the start and end date.

If an episode starts and ends within the retrospectively queried time period of a survey wave (spell 1 in interview A, see Figure 23), it can be assumed that this episode has been recorded completely with all information. In the corresponding spell dataset of the Scientific Use File, this episode appears in a single data row.

However, there are episodes that have not yet been finished at the time of the interview, but continue beyond that point. Such episodes are updated in the subsequent survey wave in which the respondent participates. That is, further information about the episode is collected in one or more subsequent waves until the episode is reported as finished (spell 2 in interview B and interview C, see Figure 23). In such cases, information about an episode is stored separately in one data row for each survey wave. Accordingly, the information is spread over several data

rows and a single data row contains only a subset of information for that episode. The respondent ID is identical in each data row for this episode, as well as the episode ID. The distinction is made by the variable `subspell`, in which the data rows belonging to an episode that was recorded over several survey waves are consecutively numbered (starting with the value 1).

Analogous to episodes that began and ended within the time period of a survey wave (spell 1), the variable `subspell` has a value of 0 also for episodes that were recorded for the first time in the current survey wave and were still ongoing at the day of the interview (spell 3 in interview C, see Figure 23).

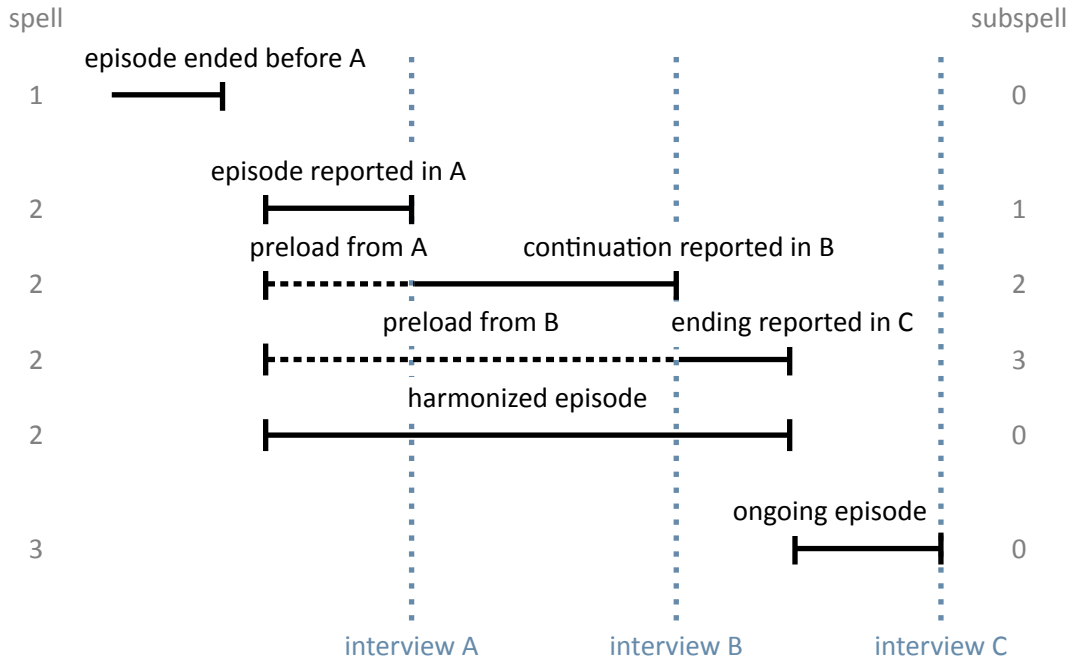

Figure 23: Logic of subspells

The sample episodes from Figure 23 correspond to the data structure presented in Table 7 *before* any episode harmonization.[4] There is only one data row for the first episode. It was completed before the data collection of wave 2, i.e. the information is completely recorded. The value of the variable `subspell` is 0. The second episode is spread over three data rows with information asked in the surveys waves 2 to 4. The values of the variable `subspell` are 1 to 3 according to the consecutive numbering of the sub-episodes. The third episode was recorded in the fourth survey wave. This episode continues, but since only part of the episode has been reported so far, `subspell` is also given the value 0. This value changes as soon as further information about this episode is added in a subsequent survey wave.

4 For the sake of convenience, the table only includes data from three consecutive survey waves, conducted in December 2009 (wave=2), 2010 (wave=3), and 2011 (wave=4).

Table 7: Data lines of the example case in the SUF before spell harmonization

ID_t	splink	wave	subspell	start_m	start_y	end_m	end_y	ongoing	var1	var2
1	300001	2	0	may	2005	april	2009	no	3	5
1	300002	2	1	june	2009	december	2009	yes	1	.
1	300002	3	2	june	2009	december	2010	yes	.	.
1	300002	4	3	june	2009	july	2011	no	.	8
1	300003	4	0	august	2011	december	2011	yes	2	4

For episodes that span over several survey waves, the same information is not collected in each survey wave. In the wave in which an episode is recorded for the first time, all unchanging core information about it is captured. In the example of training episodes, this includes the start date, the type of training (e. g., vocational training or study), the exact name of the training occupation and some other parameters that distinguish this training from others. In later survey waves, this information is no longer requested when updating this episode. However, additional characteristics, such as current pay, are recorded. Once the respondent indicates that the episode has been finished, information about the end is recorded. This is, for example, the achieved completion of a training and, of course, the end date of the episode. Thus, the information about an episode that lasts over several survey waves is divided among sub-episodes (subspells). The number of sub-episodes varies depending on the total duration of the episode or the number of interviews in the course of this duration. [5]

To ease the work with updated episodes, the information from the sub-spells of an episode is summarized in an additional data row. In addition to the data rows for the sub-episodes, there is one data row that provides a summary of the entire episode (up to the last interview). This data row represents the *harmonized episode*. Episode harmonization is only used if several subspells from different survey waves are available for the same episode.

The data row for the harmonized episode is simply added to the existing data rows for an episode. It is always identified by the value 0 in the variable `subspell`. In the example case, the additional data row concerns the second episode (`splink=30002`) as a summary of the three sub-episodes (see the highlighted row in Table 8). The other two episodes do not have multiple subspells across different survey waves, so harmonization is not necessary or possible.

Table 8: Data lines of the example case in the SUF after spell harmonization

ID_t	splink	wave	subspell	start_m	start_y	end_m	end_y	ongoing	var1	var2
1	300001	2	0	may	2005	april	2009	no	3	5
1	300002	2	1	june	2009	december	2009	yes	1	.
1	300002	3	2	june	2009	december	2010	yes	.	.
1	300002	4	3	june	2009	july	2011	no	.	8
1	300002	4	0	june	2009	july	2011	no	1	8
1	300003	4	0	august	2011	december	2011	yes	2	4

5 An update of episodes is only carried out in Starting Cohort 6 for the following datasets: `spChild`, `spChildCohab`, `spEmp`, `spGap`, `spMilitary`, `spParLeave`, `spPartner`, `spResidence`, `spSchool`, `spUnemp`, `spVocPrep`, `spVocTrain`.

Since the harmonized spell is a summary of all subspells of an episode, exactly one piece of information must be selected from these subspells for each variable to be transferred to the harmonized spell. There are six rules that are applied for selecting the relevant piece of information for the harmonized spell. Which of these rules is used for a variable depends on content-related criteria. Data users can identify the respective rule in the additional attributes or characteristics of each variable:

first_noedit For all variables that are filled only at the start of a new episode, i.e. when the episode is first reported, the information from the first sub-episode goes into the harmonized spell, since it can be found only there and is valid for the entire duration of the episode (see `var1` in Table 8). Missing values from -59 to -50 in the first subspell as well as the missing value -29 are **not** transferred to the harmonized spell.[6] In case that there are such missings in the first subspell, the next non-missing value from the subsequent subspells is taken instead.

last_noedit For information that is newly collected in each survey wave or that is only present in the last subspell of the episode, the information for the harmonized spell is taken from the last subspell (see `var2` in Table 8). Missing values from -59 to -50 as well as the missing value -29 in the last subspell are **not** transferred to the harmonized spell.[7] In case that there are such missings in the last subspell, the next non-missing value from the previous subspells is taken instead.

first_noeditnosys The harmonization of most variables follows either the *first_noedit* or the *last_noedit* selection rule. However, there are exceptions. One such exception is when a new question is introduced in the collection of episodes whose variable basically follows the *first_noedit* rule, but which is collected in the current survey wave for an episode that is already continuing. In such cases, the information is included in the data for an updated episode, however, not in the first subspell, but in a later subspell. In these cases, the first valid value found in any subspell of an episode is selected. Missing values from -59 to -50 as well as the missing value -29 and system missings (.) in the first subspell are **not** transferred to the harmonized spell.

last_noeditnosys A similar exception applies to variables that measure a changing state until a defined target state is reached. In the case of employment episodes, for example, this might be the change from a temporary position in a particular job to a permanent position. In cases where a position is temporary at the time of the first recording, the question about the temporary nature of that position is asked each time in subsequent survey waves. This continues until the employment either ends or the status changes to "permanent". Once this change has occurred, the question about a fixed term is no longer asked when the episode is updated later on.[8] Thus, the information about the fixed term of the episode is not necessarily in the first or in the last subspell. Here, the last valid value of a subspell of the episode is relevant. For this reason, the rule *last_noeditnosys*

6 If the missing code -53 (anonymized) is given in the first subspell, this value is copied to the harmonized spell.
7 If the missing code -53 (anonymized) is given in the last subspell, this value is copied to the harmonized spell.
8 A reverse change from permanent to temporary within the same job is not considered very realistic.

(last valid value found in the subspells of an episode) is used for harmonization. Missing values from -59 to -50 as well as the missing value -29 and system missings (.) in the last subspell are **not** transferred to the harmonized spell.

first_all This rule is identical to *first_noedit* with the exception that **all** missing codes from the first subspell are transferred to the harmonized spell.

last_all This rule is identical to *last_noedit* with the exception that **all** missing codes from the last subspell are transferred to the harmonized spell.

The Research Data Center at LIfBi protocols which harmonization rule was applied to which variable of life history episodes that have been updated over several survey waves. The information is stored in the datasets for each relevant variable in the additional attributes or characteristics. The harmonization can also be viewed upon specific request.

There is another special aspect regarding the harmonization of episodes: Respondents have the possibility to contradict the update of an episode in the current survey wave in the course of the review of the data in the check module (see Section 4.4.1 and Ruland et al., 2016). Only episode types included in this check during the interview are affected (from `spSchool`, `spVocPrep`, `spVocTrain`, `spMilitary`, `spEmp`, `spUnemp`, `spParLeave`, `spGap`). In the case of such a contradiction, the data edition assumes that the subspells recorded in previous waves of the survey contain correct information about this episode. This is simply because the inputs in the previous waves were also subjected to a joint review with the respondent – with no contradiction. Following this logic, it is only possible to contradict the part of the episode that was recorded in the current survey wave, not the entire episode. For the data structure, this means that the information already collected and stored in a data row for the current part of the episode (which was contradicted in the check module) is still in the dataset, but is marked in the variable `spms` with the code -20 as “episode revoked in check module”. With respect to harmonization, the contradiction is taken into account by filling the harmonized episode only with values from the subspells not marked as contradicted. This means, that only not contradicted subspells are included in the harmonized spell. The end date of the respective episode is set to the interview date of the survey wave in which the last uncontradicted information for this episode was recorded.

Last but not least: In the harmonized episodes, the occupational information is newly coded based on the summarized information. Therefore, it is possible that there are differences in the values of these generated variables between subspells and the harmonized episode. For example, it may happen that a self-employed activity is reported and additional questions are asked about it, such as the professional position, the presence of a management function, and so on. In subsequent waves, the professional episode of self-employment continues, but the function has changed with the hiring of a salaried employee. This current information is transferred to the harmonized spell. As a result, the first subspell shows a self-employed person without a leading function and the harmonized spell shows a self-employed person with a leading function. Accordingly, the occupational information is recoded in the harmonized spell.

Handling of harmonized episodes

Data users can and must decide for themselves whether to use the harmonized episodes for their data analysis or to consider the information from the separate subspells that reflect changes in the characteristics of an episode over time. Both pieces of information are available in the spell datasets.

If the harmonized episodes are to be used – including episodes that consist of only one subspell and therefore did not need to be harmonized – it is sufficient to select all data rows with the value 0 in the variable `subspell`.

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

After that, all episodes should be excluded that were contradicted in the check module (variable `spms=-20`) and do not belong to the harmonized episodes (variable `spext=0`).[9] As described above, this step is already included in the process of harmonizing episodes.

If, on the other hand, one does **not** want to use the harmonized episodes but the original subspells, then all data rows must be deleted where the variable `subspell` has the value 0 and at the same time the variable `spext` has the value 1. After that, all sub-episodes must be excluded as well, which were contradicted in the check module (variable `spms=-20`).

```
drop if subspell==0 & spext==1
drop if spms==-20
```

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

4.5 Data files

In the following section, every data file of Starting Cohort 6 is described in a subsection, including a data snapshot and a syntax example that often deals with the challenge of merging information from another file (in Stata). The syntax examples are written in an easily comprehensible way. There is no need to additional install any “ado files”, although it is highly advised to use the `NEPStools` (see Section 1.6).

To facilitate the understanding of the relationships between the data files in the Scientific Use File, an overview of all datasets is provided in Figure 24. The lines in this figure symbolize how a data file may be linked to other files. This is not meant to document every possible data link, but rather tries to give an idea on which data files relate most. By clicking on a box, one gets directed to the short description of this dataset.

For the Stata syntax examples in the subsequent dataset short portraits to work, the following globals must first be set. Just adapt and copy the lines below to the top of the syntax files or execute them in the Stata command line before running the syntax.

```
** Starting Cohort
global cohort SC6
** version of this Scientific Use File
global version 16-0-0
** path where the data can be found on your local computer
global datapath Z:/Data/${cohort}/${version}
```

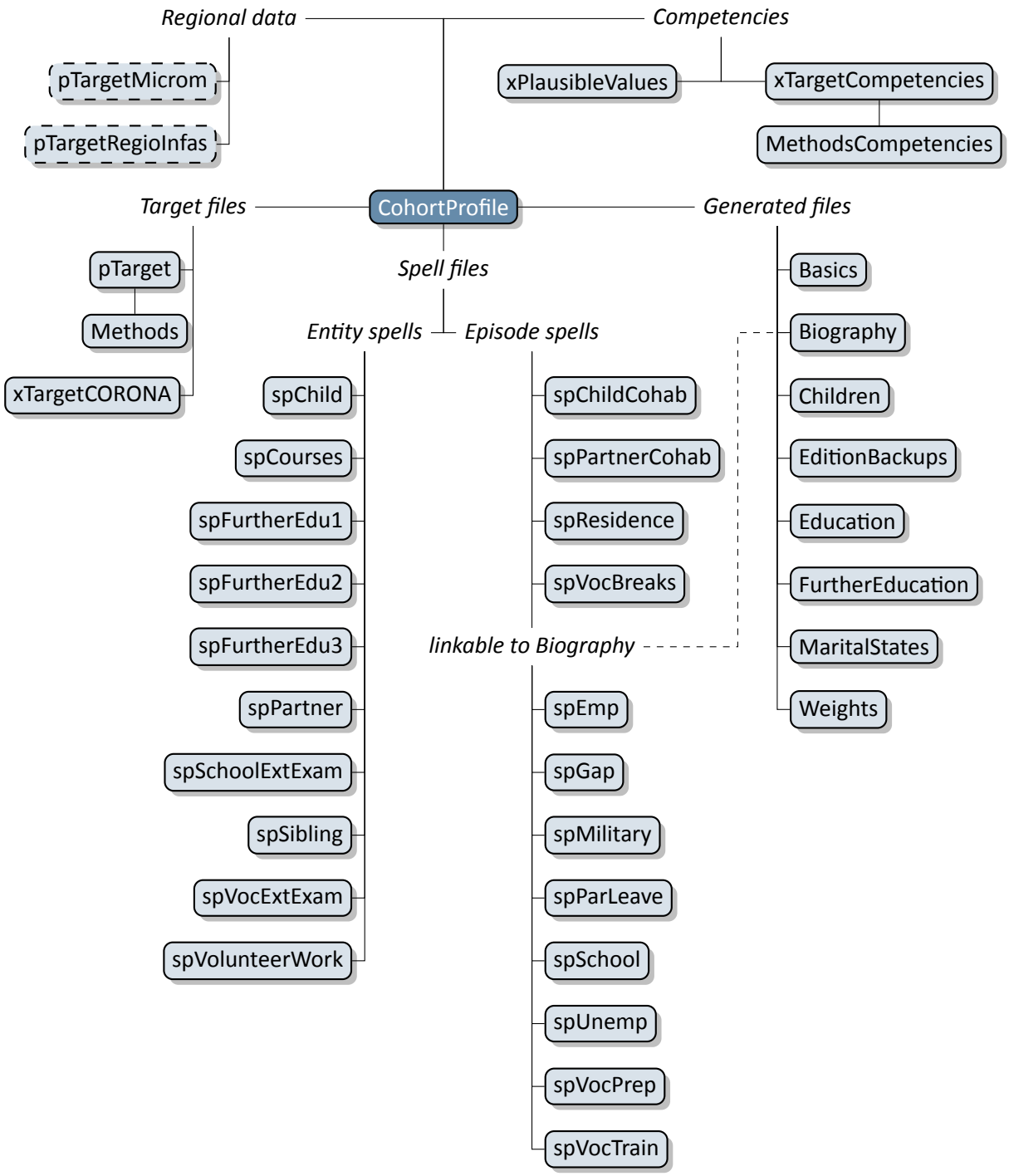

Figure 24: Graphical overview of all data files. Each box represents one data file. Relations are indicated by connecting lines. Files with a dashed border are not available in the Download version of the Scientific Use File. Click on a file to get more information.

4.5.1 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 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
tx29000	Age at interview month (years)
t70000m	Month of birth
t70000y	Year of birth
t700001	Gender
tx27000	Family status
tx29003	Mother tongue: German
tx29004	Citizenship: 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	male	yes	3	yes	5
8006178	32.92	female	yes	2	yes	2
8006267	55.58	male	yes	2	yes	2
8009232	47.83	female	yes	3	yes	5
8002141	63.92	female	yes	2	yes	5

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!**

Stata 1: 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!
```

4.5.2 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 / 227,514

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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
8000880	220004	2	22	1984	1990
8003037	260005	1	26	2003	2003
8005871	290101	1	29	1985	1986
8009315	220004	2	22	1972	1975
8013871	260001	1	26	1985	1985

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 4.4.2) 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`).
- Episodes with missing dates or negative durations are excluded
- Start 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.

Stata 2: 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
```

4.5.3 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 / 28,856

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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
8000387	2	Biological child	male	December	1976	1
8011783	2	Biological child	female	January	1979	1
8000420	2	Biological child	female	August	1985	1
8008144	2	Biological child	male	April	1993	1
8001443	2	Biological child	male	September	2005	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`).

Stata 3: 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
```

4.5.4 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

20 / 253,462

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
cohort	NEPS Starting Cohort
sample	Sample
tx80220	Participation/drop-out status
tx80521	Data available: survey target person
tx80522	Data available: competence test target person
tx8601y	Survey target person: survey year 1
tx8601m	Survey target person: survey month 1
tx80105	NEPS sub samples
tx80107	Sample: First participation in wave

Exemplary data snapshot

ID_t	wave	tx80220	tx80521	tx80522	tx8601y
8002353	7	Participation	yes	yes	2014
8003255	5	Participation	yes	yes	2013
8004643	3	Participation	yes	yes	2011
8010727	7	Participation	yes	yes	2014
8023325	5	Participation	yes	yes	2012

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 (`tx8600d/m/y`) and competency testing (`tx8610m/y`) was conducted.

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

Stata 4: 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
```

4.5.5 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 data file

ID variables needed to identify a single row

dataset varname ID_t ID_e ID_cc wave

Other ID variables useful for linkage

mergevars

Number of variables / number of rows in file

14 / 2,783

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
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)

Exemplary data snapshot

ID_t	wave	dataset	varname	mergevars	sourcevalue_num	editvalue_num
8002340	1	spVocExtExam	ts15304_v1	ID_t wave exam	22.00	2.00
8007013	2	spSchoolExtExam	ts11302	ID_t wave exam	7.00	4.00
8007577	2	spSchoolExtExam	ts11302	ID_t wave exam	7.00	4.00
8007985	2	spSchoolExtExam	ts11302	ID_t wave exam	7.00	4.00
8024242	4	pTarget	t731306	ID_t wave	5.00	3.00

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

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

```
** In this example, we want to restore the original
** values in variable t40607y (Year of naturalization) 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=="t40607y"

** 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 t40607y_source
rename editvalue_num t40607y_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 t40607y* if _merge==3

** replace the variable in the datafile with its original value
replace t40607y=t40607y_source if _merge==3
```

4.5.6 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

14 / 54,179

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
number	Sort number
datem	valid since (month)
datey	valid since (year)
tx28101	Recent CASMIN
tx28102	years of education = f(CASMIN)
tx28103	Recent ISCED-97
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
8011385	4	1978	6	15	6	220003	22
8001110	3	1973	5	13	3	220003	22
8004510	3	2007	6	15	8	240001	24
8000569	3	1990	2	12	4	240001	24
8020545	3	1998	8	18	9	240001	24

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)
```

Stata 6: 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
```

4.5.7 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 / 98,387

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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
8001156	4	260005	601	spCourses	2013	2013
8007436	2	260007	202	spCourses	2009	2010
8010046	1	260003	201	spCourses	2009	2010
8022646	21	260005	1001	spCourses	2017	2017
8023265	6	260002	502	spCourses	2012	2013

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

Stata 7: 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
```

4.5.8 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

8 / 18,988

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
number	Sort number
datem	Change since (month)
datey	Change since (year)
tx27000	Family status

Exemplary data snapshot

ID_t	number	datem	datey	tx27000
8009715	2	April	1996	divorced
8002317	5	September	2004	married/in registered civil partnership
8007623	2	September	1986	widowed
8001810	3	April	2016	married/in registered civil partnership
8009019	1	December	2003	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).

Stata 8: 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
```

4.5.9 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

43 / 188,365

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
tx80107	Sample: First participation in wave
tx80209	Interview: length of interview (minutes)
tx80200	Interview: number of all contact attempts
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	tx80209	ID_int	tx80301	tx80302
8003119	8	40.66	1115	female	50-65 years
8005797	15	85.01	1154	male	older than 65 years
8010594	16	41.00	2473	male	older than 65 years
8012441	4	46.85	1009	female	50-65 years
8021819	11	63.11	1074	female	50-65 years

This dataset provides a variety of information about data collection such as gender (`tx80301`) and age (`tx80302`) of the interviewer, the interview duration (`tx80209`), the number of all contact attempts (`tx80200`),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`.

Stata 9: 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 tx80207

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

4.5.10 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

96 / 37,034

Contains data from waves

1 2 **3** 4 **5** 6 **7** 8 **9** 10 11
12 13 **14** 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
tx76001	Split group
tx76004	Intro Numeracy
tx76005	Instruction Numeracy
tx76006	Start Timekeeping Numeracy
tx76007	Interview location
tx76008	Residential area: Density
tx76009	Residential area: Use residential buildings
tx76015	End Timekeeping Numeracy
tx76016	Start Metacognition Numeracy
tx76025	Start Timekeeping Reading
tx76044	Interruption Reading Speed
tx76054	Intro Testing
tx76089	Duration TBT-module in sec
tx76090	Duration TBT-intro in sec

Exemplary data snapshot

ID_t	wave	tx76004	tx76005	tx76007	tx76008	tx76016	tx76025	tx76089
8001772	3	1	1	1	2	1	.	-54
8002494	3	1	1	1	6	1	.	-54
8006804	3	1	1	1	2	1	.	-54
8010988	3	1	1	1	2	1	.	-54
8014286	3	1	1	1	2	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 (`tx76001`), interview location (`tx76007`) and different start and end markers for different modules (e. g., reading, ICT).

Stata 10: 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 tx76001 wave
```

4.5.11 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

th14599_g1 (see section 5.3.4)

Number of variables / number of rows in file

1,443 / 136,521

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
tx20000	Number of children in household
t700101_g1	Municipality of birth (west/east)
t34005a	Number books
t700001	Gender
t70000y	Year of birth
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
8000916	2	1	male	1953	1	9	6	10
8007805	2	1	female	1954	8	9	1	10
8020602	4	1	male	1970	7	4	1	9
8021622	4	2	female	1984	5	5	1	8
8021709	4	2	female	1958	8	5	1	9

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`.[10]

10 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).

Stata 11: 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
```

4.5.12 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

Exemplary variables

ID_t	Identifier for target person
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
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 euros

Exemplary data snapshot

ID_t	wave	regio	ID_regio	mso_k_ausland	mbe_k_haustyp	mpi_k_dichte
8006483	5	1	134580	5	2	9
8007800	6	1	149184	3	3	6
8009132	5	1	160072	1	2	8
8012691	6	1	122398	1	1	5
8021808	6	1	123450	1	2	8

The data file `pTargetMicrom` is only available **On-site**. You cannot work with this file having only access to the Download or Remote data version.

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.

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

```
** open Microm data file. 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
```

4.5.13 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 wave 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 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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
8000774	1	22049	17.45	2.37
8005723	3	17346	6.22	2.40
8007514	2	19153	8.60	2.18
8010482	1	20592	29.40	2.06
8006266	1	23031	39.87	2.36

The data file `pTargetRegioInfas` is only available **On-site**. You cannot work with this file having only access to the Download or Remote data version.

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.

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

```
** open data file. 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'
```

4.5.14 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

83 / 234,169

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
child	Child number
subspell	Number of subspell
wave	Wave
ts3320m	Month of birth of child
ts3320y	Year of birth of child
ts33203	Gender of the child
ts33204	Biological, adoptive or foster child
ts33209	Employment child
ts33301	School enrollment already completed
ts33216	Vocational training child
ts33202	Auxiliary variable: age of the child

Exemplary data snapshot

ID_t	child	subspell	wave	ts3320y	ts33203	ts33204
8009067	2	1	2	1988	female	biological child
8009122	1	1	2	2003	female	biological child
8010015	3	1	1	1996	female	biological child
8010567	1	1	2	1989	male	biological child
8021993	3	1	4	1983	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.

Stata 14: 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
```

4.5.15 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

22 / 137,691

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

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

Exemplary data snapshot

ID_t	child	spell	subspell	wave	ts3331y	ts3332y
8004314	2	201	1	1	1990	2007
8004391	1	104	1	14	2021	2021
8009790	2	201	4	5	1995	2012
8014100	1	101	9	14	2013	2022
8021395	2	201	2	5	1995	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.

Stata 15: 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
```

4.5.16 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

74 / 33,291

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
splink	Link for spell merging
sptype	Spell type
t271001	Total duration of training courses
course_w1	1st course number
t271011_w1	Duration of the course
course_w2	2nd course number
t271011_w2	Duration of the course
course_w3	3rd course number
t27800a	Start month episode
t27800b	Start year episode
t27800c	End month episode
t27800d	End year episode

Exemplary data snapshot

ID_t	wave	splink	sptype	course_w1	course_w2	course_w3
8003305	11	260005	26	1101	1102	1103
8021787	12	260003	26	1201	1202	1203
8021819	7	260001	26	701	702	703
8021951	4	260004	26	401	402	403
8023068	4	260004	26	401	402	403

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.

Stata 16: 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
```

4.5.17 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

146 / 214,583

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
ts23222	In partial retirement, (active phase)
ts2311y	Start year employment episode
ts2312y	End year employment episode
ts23410	Net income, open
ts23228	Type of required 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
8004892	1	5	2005	2007	800	6	40.00
8020532	1	11	2012	2013	1200	1	48.00
8020614	1	10	2016	2016	450	1	5.00
8005505	1	2	1988	2007	1700	6	40.00
8005982	1	2	1984	2007	1325	3	40.00

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.

Stata 17: 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
```

4.5.18 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

20 / 28,502

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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	Course discontinued
t272000_R	Course content
t272000_g13	Course content other
	course (course ID)
t271043	Duration of course

Exemplary data snapshot

ID_t	wave	course	t271045	t271046	t271048
8002253	2	201	2009	2009	no
8002949	2	202	2009	2009	no
8006848	2	202	2008	2009	no
8007960	2	202	2008	2010	yes, course is ongoing
8011864	2	202	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.

Stata 18: 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
```

4.5.19 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 / 57,384

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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: patient instructor
t279047	Course costs employment agency

Exemplary data snapshot

ID_t	wave	course	t279046	t279041	t272043
8001795	4	401	fully	little effort	1
8005000	5	501	fully	a lot of effort	1
8012813	7	702	fully	little effort	1
8020344	8	801	fully	a lot of effort	1
8021217	9	902	fully	no effort at all	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`.

Stata 19: 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)
```

4.5.20 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 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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
8008747	201	2	May	2005	November	2005
8009657	201	2	April	1991	January	1992
8022345	401	4	February	1986	July	1986
8023198	401	4	October	1974	November	1974
8023726	402	4	September	1978	December	1978

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.

Stata 20: 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)
```

4.5.21 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

36 / 50,486

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
splink	Link for spell merging
spell	Spell number
subspell	Number of subspell
wave	Wave
spms	Check module: spell type
ts29901	Auxiliary variable Current gap
ts29300	Episode mode
ts2911m	Start month gap
ts2911y	Start year gap
ts2912m	End month gap
ts2912y	End year gap
ts2912c	Ongoing of gap episode
ts29201	Training course during gap
ts29101	Type of gap

Exemplary data snapshot

ID_t	spell	subspell	wave	ts29901	ts2911y	ts2912y
8000438	1	1	2	1	2001	2009
8001272	3	2	2	1	1990	2010
8005733	1	2	3	1	2008	2011
8020412	4	1	4	1	2009	2011
8022871	2	1	4	1	2010	2012

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`.

Stata 21: 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
```

4.5.22 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

26 / 6,596

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
wave	Wave
ts21201	Type of military service episode
ts2111m	Start month military service episode
ts2111y	Start year military service episode
ts2112m	End month military service episode
ts2112y	End year military service episode
ts21202	Attendance of training courses/courses during military service

Exemplary data snapshot

ID_t	splink	subspell	spell	wave	ts2111y	ts2112y
8004831	250001	1	1	1	2007	2007
8006303	250001	1	1	1	2002	2007
8020025	250001	2	1	5	2012	2012
8020559	250001	6	1	9	2000	2016
8020559	250001	9	1	12	2000	2020

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`.

Stata 22: 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
```

4.5.23 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

41 / 13,388

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
child	Child number
spell	Spell number
subspell	Number of subspell
wave	Wave
ts2711m	Start month parental leave
ts2711y	Start year parental leave
ts2712m	End month parental leave
ts2712y	End year parental leave
ts2712c	Ongoing of parental leave
th27100	Employment after parental leave

Exemplary data snapshot

ID_t	child	spell	subspell	wave	ts2711y	ts2712y
8001967	2	202	3	3	2007	2010
8002891	3	301	2	2	2007	2008
8005460	1	101	2	3	2009	2011
8007508	1	101	1	2	2009	2010
8023263	2	201	4	7	2012	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.

Stata 23: 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
```

4.5.24 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

108 / 128,738

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
partner	Partner number
subspell	Number of subspell
ts31204	Partner born in Germany/abroad
ts31211	Partner German
ts31203	Gender partner
ts3131y	Year of moving in with partner
ts3141m	Month of marriage
ts3141y	Year of marriage
ts3120m	Month of birth partner
ts3120y	Year of birth partner
ts31206	Age at immigration Partner
ts31207	Place of birth Father Partner
ts31209	Place of birth Mother Partner

Exemplary data snapshot

ID_t	partner	subspell	ts31203	ts3120m	ts3120y
8012342	1	1	female	June	1987
8006750	1	1	female	December	1962
8007405	1	1	male	April	1953
8019923	2	1	female	February	1963
8002902	1	1	female	March	1972

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.

Stata 24: 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)
```

4.5.25 spPartnerCohab

« go back to overview

Description

Spell data on all episodes of cohabitation with a partner

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

partner wave

Number of variables / number of rows in file

25 / 20,546

Contains data from waves

Exemplary variables

ID_t	Identifier for target person
partner	Partner number
spell	Spell number
subspell	Number of subspell
ts31203	Gender partner
ts3120m	Month of birth partner
ts3120y	Year of birth partner
th2804m	Start month living together partner
th2804y	Start year living together partner
th2805m	End month living together partner
th2805y	End year living together partner

Exemplary data snapshot

ID_t	partner	spell	ts31203	ts3120y	th2804m	th2804y	th2805m	th2805y
8002953	2	201	male	1964	6	2020	1	2021
8020493	3	301	male	1986	12	2019	9	2020
8011580	4	401	male	1949	10	2019	12	2023
8003109	1	101	female	1981	10	2021	12	2023
8010289	7	702	male	1992	4	2020	7	2020

If a respondent lives together with a partner in a household, the duration of the cohabitation is registered in `spPartnerCohab`. Cohabitation episodes are connected to the respective partner via the number in the variable `partner`, and thus can be linked to datafile `spPartner` for more information. For convenience, some basic information about the partner are also available in this file, e.g., her or his gender (`ts31203`) or birthdate (`ts3120m/y`).

Stata 25: Working with spPartnerCohab

```
** open the data file
use ${datapath}/SC6_spPartnerCohab_D_${version}.dta, clear

** switch to english language
label language en

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

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

4.5.26 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 / 83,861

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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 year residence episode
th21108	End year residence episode
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
8002062	1	3	1	Germany	1985	2007	North Rhine-Westphalia
8002682	1	2	4	Germany	2011	2011	North Rhine-Westphalia
8004590	1	1	1	Germany	1982	2007	Rhineland-Palatinate
8005519	1	2	1	Germany	1961	2007	Berlin (total)
8006504	1	3	4	Germany	2011	2011	Baden-Wuerttemberg

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.

Stata 26: 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
```

4.5.27 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

65 / 38,489

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
wave	Wave
ts11204	Type of school
ts1111m	Start date School episode, month
ts1111y	Start date School episode, year
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
8002610	220003	1	3	1	2005	2007
8007465	220004	4	4	9	2013	2016
8010223	220006	1	6	8	2015	2015
8011801	220004	2	4	3	2010	2010
8012974	220004	3	4	5	2009	2012

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.

Stata 27: 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
```

4.5.28 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

16 / 972

Contains data from waves

Exemplary variables

ID_t	Identifier for target person
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	Final grade Leaving certificate (awarded qualification)
ts11301_g1R	Country of awarded school-leaving qualification
ts11301_g2	Country of awarded school-leaving qualification (categorized)

Exemplary data snapshot

ID_t	wave	exam	ts11300	ts1130y	ts11302	ts11300_g1
8007727	2	1	1	2005	5	1
8008131	2	1	1	2006	3	1
8008142	2	1	1	1965	3	1
8012296	1	1	1	2003	1	1
8012438	1	1	1	1986	4	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`).

Stata 28: 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)
```

4.5.29 spSibling

« go back to overview

Description

Spell data on siblings

File structure

entity format: 1 row = 1 sibling of 1 respondent

ID variables needed to identify a single row

ID_t t732105

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

21 / 7,250

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
t732105	Sibling number
t732109	Sibling still alive
t732220	Gender sibling
t73221m	Date of birth: month
t73221y	Date of birth: year
t732210	Twin sibling

Exemplary data snapshot

ID_t	wave	t732105	t732109	t732220	t73221m
8002496	15	1	yes	female	June
8009720	15	1	yes	male	October
8010489	15	1	yes	male	November
8024083	15	2	no	male	April
8024489	15	2	yes	male	September

The module in the dataset `spSibling` surveyed multiple data about the respondents siblings, like gender, date of birth, or qualifications. Each row in the datafile corresponds to one sibling of a respondent. Please note that this module was only introduced in the CAWI survey of wave 15, so only people still participating here are covered. You can not conclude from the fact that someone is not present in this file that they have no siblings. To conduct this information, you will have to rely on other files, e.g. pTarget (`t732100`).

Stata 29: Working with spSibling

```
** In this example, we just want to know whether a person has as many sisters as
  brothers

** open the data file
use ${datapath}/SC6_spSibling_D_${version}.dta, clear

** switch to english language
label language en

* tabulating the gender variable and note the coding
fre t732220

** for a more convenient edition later, we recode 2 (female) to 0
recode t732220 (2=0 "female") (1=1 "male"), gen(sibling)
tab t732220  sibling

** keep only the necessary items
keep ID_t t732105 sibling

** also, drop missings
drop if sibling<0

** making the datafile more "visual", by transfering the information
** from rows to columns. You now have one row for each respondent only
reshape wide sibling, i(ID_t) j(t732105)

* we now create a "gender rate", where 0 means respondent has only female
* siblings, 1 means respondent has only male siblings.
egen rate=rowmean(sibling*)

** a rate at .5 means both genders are equally present
browse if rate==.5
```

4.5.30 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 / 30,804

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
subspell	Number of subspell
spell	Spell number
wave	Wave
ts2511m	Start month unemployment episode
ts2511y	Start year unemployment episode
ts2512m	End month unemployment episode
ts2512y	End year unemployment episode
ts25202	Receiving unemployment benefits or assistance at the beginning
ts25203	Registered unemployment (current/former)
ts25205	Number of job applications
ts25206	Invitation to job interviews
ts25207	Number job interviews

Exemplary data snapshot

ID_t	subspell	spell	wave	ts2511m	ts2511y	ts2512m	ts2512y
8004425	2	2	13	4	2019	10	2020
8007227	3	2	10	10	2015	3	2017
8008941	1	1	2	3	2002	3	2010
8010748	1	2	2	11	1973	5	2010
8011903	4	2	16	1	2020	12	2022

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.

Stata 30: 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
```

4.5.31 spVocBreaks

« go back to overview

Description

breaks during vocational training

File structure

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

ID variables needed to identify a single row

ID_t splink break

Other ID variables useful for linkage

none

Number of variables / number of rows in file

14 / 899

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
splink	Link for spell merging
break	Training interruption number
ts1531y	Start year training episode
ts1531m	Start month training episode
ts1532y	End year Training episode
ts1532m	End month Training episode

Exemplary data snapshot

ID_t	splink	break	ts1531y	ts1532y
8024752	240002	1	2015	2016
8009937	240005	1	2011	2011
8021492	240001	1	1993	1994
8021347	240002	1	2014	2014
8000858	240002	2	1981	1982

This module covers all breaks of further trainings, vocational and/or academic, that a respondent ever attended. Information on vocational breaks were part of `spVocTrain` in prior data releases. Since release 13-0-0 break episodes are being extracted and edited to `spVocBreaks`. The data structure of breaks has been transformed from wide format to long format. In this dataset …

- different types of breaks (break semesters, de-registrations, non-formal breaks) are included.
- includes several breaks within a single episode of a person.
- closes gaps between succeeding breaks (< 3 months) and combines overlapping breaks to continuous breaks.
- breaks within breaks were deleted.
- dates of beginnings and endings were corrected and stored as variables with `_g1`–suffixes.

- every break is in a separate row.
- `splink` helps you to merge data to `spVocTrain` and `Biography` as well.

Stata 31: Working with spVocBreaks

```
** example 1: merge spVocBreaks and spVocTrain

** open the vocational breaks
use ${datapath}/SC6_spVocBreaks_D_${version}.dta , clear

** reshape vocational breaks to wide format to match data with spVocTrain; first add
  _w-suffix to variables
foreach var of varlist ts15310_g1 ts15310_g2 ts1531y ts1531m ts1531m_g1 ts1531y_g1
  ts1532c ts1532y ts1532m ts1532y_g1 ts1532m_g1 {
        rename `var' `var'_w
}
reshape wide *_w, i(ID_t splink) j(break)

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

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

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

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

** you now have put together information of breaks and the vocational track to
  analyze the respondents with breaks. The number of total episodes with breaks
  reduces the amount of rows of the combined dataset.


** example 2: merge spVocBreaks and Biography (further data preparation to analyze
  data is recommended)

** open the vocational breaks
use ${datapath}/SC6_spVocBreaks_D_${version}.dta , clear

*merge breaks with biography data
merge m:1 ID_t splink using ${datapath}/SC6_Biography_D_${version}.dta

** now you could cut those vocational episodes using dates of episodes and breaks to
  re-define vocational episodes
************************
```

4.5.32 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

25 / 898

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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 external examination (month)
ts1530y	Date 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
8005104	15	2	11	2022	19
8009957	13	1	12	2019	18
8019953	4	1	3	2000	28
8020567	4	1	6	1985	19
8022569	4	1	7	2009	19

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`).

Stata 32: 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)
```

4.5.33 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

33 / 1,457

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
splink	Link for spell merging
spell	Spell number
subspell	Number of subspell
spgen	Generated spell
wave	Wave
ts13103	Type of measure
ts1311m	Start month vocational preparation
ts1311y	Start year vocational preparation
ts1312m	End month vocational preparation
ts1312y	End year vocational preparation
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
8002156	2	2	7	8	2013	7	2014
8007754	1	1	3	10	2010	1	2011
8008472	1	2	8	9	2013	6	2015
8012437	1	1	3	10	2010	1	2011

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.

Stata 33: 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
```

4.5.34 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

80 / 45,747

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
spell	Spell number
subspell	Number of subspell
ts15201	Type of vocational training
ts1511m	Start date Vocational training episode (month)
ts1511y	Start date Vocational training episode (year)
ts1512m	End date Vocational training episode (month)
ts1512y	End date Vocational 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
8012447	1	3	10	2005	1	2011
8011631	2	1	7	2012	11	2012
8001317	9	3	10	2021	3	2024
8010459	2	5	4	2010	8	2013
8000846	3	1	1	2012	10	2012

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.

Stata 34: 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
```

4.5.35 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

38 / 14,753

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
volunteer	Volunteering number
t262950_g1	Volunteering - societal areas
t262951_g1	Volunteering - main contents
t261902	Duration 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
8001479	10	1001	Sport and exercise	once a week
8002337	10	1001	Emergency and ambulance service	several times a month
8005579	10	1001	Church, religion	several times a month
8007339	6	601	Sport and exercise	several times a month
8024969	10	1001	Emergency and ambulance service	several times a week

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.

Stata 35: 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
```

4.5.36 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

52 / 17,140

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
sample	Sample
subsample	Subsample
psu	Sample: Primary Sampling Unit (Point number)
stratum	Sample: stratification variable (implicit stratum)
w_t2_cal	Cross-sectional weight for TP particip. in wave 2, calibr. (microcensus 2009)
w_t3_cal	Cross-sectional weight for TP particip. in wave 3, calibr. (microcensus 2010)
w_t9_cal	Cross-sectional weight for TP particip. in wave 9, calibr. (microcensus 2016)

Exemplary data snapshot

ID_t	sample	subsample	psu	stratum
8000668	2009 (NEPS)	2009 (enhancement)	18	510
8004933	2007 (ALWA)	2007 (ALWA)	164	807
8004582	2007 (ALWA)	2007 (ALWA)	9	908
8010866	2009 (NEPS)	2009 (enhancement)	29	1204
8019983	2010 (NEPS)	2010 (NEPS)	104	2020

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.

Stata 36: 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_t4toC

** create a "panel" logic, i.e., clone each row
expand 15

** 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_t4toC!=0
```

4.5.37 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
12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
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
8001674	1	0.7792	0.8636	0.0144	0.5660
8001809	1	1.8051	1.6215	1.4645	2.8603
8002187	1	1.1544	0.5590	0.9452	1.4867
8002202	1	0.4956	0.8420	0.8183	0.7675
8001669	1	1.2394	0.3836	1.3407	1.5351

Plausible Values (PV) are a way of describing individual competencies at group level. They enable (unbiased) estimates of effects at the group level that are adjusted for measurement errors. PV 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. Hypothesis tests for the specific question of interest are calculated for each of these values and combined into an overall result (Scharl et al., 2020).

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

Stata 37: Working with xPlausibleValues (find R example here)

```
** 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!
```

4.5.38 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

541 / 12,474

Contains data from waves

1 2 **3** 4 **5** 6 **7** 8 **9** 10 11
12 13 **14** 15 16

Exemplary variables

ID_t	Identifier for target person
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: standard error of WLE (corrected)
rea3_sc1	Reading competence: WLE (corrected)
rea3_sc2	Reading competence: standard error of WLE (corrected)
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
8005555	1	1	2.89600	0.91838	4.31540	2.17796
8004449	1	1	0.70769	0.57473	2.37318	1.02652
8002196	1	1	1.81645	0.65275	2.86147	1.31344
8005823	1	1	1.44370	0.60012	1.41304	0.78730
8011639	1	1	1.40822	0.68237	1.80728	0.89174

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.

Stata 38: 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
```

4.5.39 xTargetCORONA

« go back to overview

Description

Data collected in May 2020 regarding the impact of the corona pandemic on respondents life

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

Number of variables / number of rows in file

172 / 2,678

Contains data from waves

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Exemplary variables

ID_t	Identifier for target person
wave	Wave
t514001	Satisfaction with life
tm00001	Impact: coronavirus infection - no
tm00007	Impact: quarantine - no
tm00013	Employment status before coronavirus pandemic
tm00015	Systemically important profession
tm00016	Change working time
tm00017	Change work place

Exemplary data snapshot

ID_t	wave	t514001	tm00013	tm00015
8002220	.	9	I was employed	no
8004745	.	7	I was employed	yes
8007701	.	7	I was employed	no
8022416	.	8	I was employed	no
8024571	.	4	I was employed	yes

This data have been established to investigate the medium and long-term effects of the corona pandemic on skills development and educational pathways over the life course. The following questions are in particular:

- How do learning environments change and which potentials and risks become clear through the beginning digitalization of learning?
- Are there effects on upcoming educational decisions and are there medium and long-term effects on social educational inequality
- What are the effects on educational outcomes, such as income, but also non-monetary returns, e. g., health and labor market participation

Data is collected by means of a cross-cohort questionnaire program adapted to the current situation of the respective participants. In order to collect this data in a timely manner, the

first questions were administered via online survey in the NEPS Starting Cohorts 2 to 6 in May 2020. As this time span did not overlap with regular survey waves, data from this survey is marked with a missing wave (`wave==.`), and is contained in this data file. The corresponding questions have then been integrated in an additional module on the Corona pandemic, which is part of the regular main surveys in all starting cohorts afterwards. You find these data in the file `pTarget`.

Stata 39: Working with xTargetCORONA (find R example here)

```
** open the file
use ${datapath}/${cohort}_xTargetCORONA_D_${version}.dta, clear
label language en


** note that the wave is missing,
** as this reflects the pre-wave survey in may 2020
tab wave


** but rows can be uniquely identified by ID_t and wave
isid ID_t wave
```

5 Special Issues

5.1 Introduction and life course concept

A key characteristic of Starting Cohort 6 data is its rich life course information. Starting in 2007, Starting Cohort 6 was the first NEPS cohort to implement a modularized life course measurement, which still makes for a key advantage of the data (for the general conception of Starting Cohort 6, see Allmendinger et al., 2011, 2019, and also section 2.1). Modularized life course measurement means that longitudinal information on the respondent's biography is collected through customized self-reports within predefined domains of life. These domains cover different kinds of activities, such as vocational training, partnership, employment or further training. For each domain, spells are collected one after another in the life course interview. Thus, full personal biographies are recorded domain by domain. We will often refer to these life-domain-specific parts of the life course interview as life course modules throughout this section.

The modularized life course measurement is a remarkable improvement in the collection of life course data as it implements key insights from cognitive psychology and neuroscientific research into survey research. The approach benefits the empirical analysis because it leads to more accurate and complete life course data (Ruland et al., 2016). Interviewee burden is reduced by pre-structuring life courses by separating them into life domains and thereby giving interviewees (more) easily accessible stimuli that strengthen their mental recalling and reporting of biographical events. As Drasch and Matthes (2013) show, the modularized measurement leads, among others, to higher data quality, e.g., more reported unemployment episodes. In contrast to less structured calendar measurements that essentially ask what happened first, what next, what next, what next, the modularized approach reduces the risk that respondents forget or omit episodes, for instance, overlapping, parallel, or unpleasant ones – thus, it reduces streamlining of life courses and underreporting of life events (Ruland et al., 2016). Thereby, survey researchers not only get more complete life course data but also more precise information, especially with regard to dates and durations of certain activities. This is a great deal for longitudinal data collection and analysis because it reduces the measurement error and the confounding bias. Having more precise life course information on an independent variable, for instance, leads to less biased estimates of that variable as data is less noisy. Having more precise life course information in a dependent variable (e.g., in sequence or event history analyses), decreases the variance of the error terms – a fact that likewise strengthens causal analysis.

However, the life-domain-centered (i.e., modularized) approach has its downside, too. Above all – as the flip side of gaining more complete and less stream-lined biographies – it leads to the reporting of more overlapping events across life domains. A respondent might, for instance, report having an employment of 32 hours per week while attending full-time vocational training at the same time or being on parental leave while being unemployed. In a second step of the

life course interview – processed by software in the computer-assisted interview – the domain-specific life histories are therefore compiled into a full, cross-domain life course. This merging step includes coherence checks across life domains. For instance, when a respondent reports full-time employment parallel with full-time schooling, he or she is asked to sort this overlap out. In order to keep the individual checking of coherence in the life course data manageable and time-efficient, this verification is only applied to the occupational (esp., employment and unemployment) and the educational modules (esp. vocational training and further training) but not to the further life course modules like those on partnerships and children.

How life course information is compiled and checked across life domains (modules) in the Starting Cohort 6 life course interview is set forth in Ruland et al. (2016). Adding to them and the conceptual overview given in Allmendinger et al. (2019), we provide an in-depth information about the Starting Cohort 6 life course measurement in this chapter (also see section 4.4 for more general information about episode data). In particular, we provide detailed information about the various modules of the life course interview with a special focus on key definitions and changes across waves, respectively (cf., section 5.3 and table 9 for an overview).

Table 9 gives an overview on module names, numbers, and their main content. Module names and numbers refer to the structuring of the questionnaire in the programming template. Please note that besides in this chapter, module numbers are only to be found in the field version of the survey instrument and in field reports. No reference to these numbers is used in other documentation, such as SUF instruments, codebooks, NEPSplorer, or SUF data. For more information about the available documentation, see section 1.2.

Table 9: List of life course modules in Starting Cohort 6

Module	Number	SUF-files	Main contents
Vocational Training	24 AB	spVocTrain	The module captures all vocational or academic educational spells.
Military	25 WD	spMilitary	The module records all episodes of military, civilian and voluntary services.
Employment	26 ET	spEmp	The module captures information on all employment episodes that respondents report on gainful employment, e.g., all activities leading to income.
		pTarget	Additional panel information on different topics are available for selected employment episodes.

(...)

Table 9: (continued)

Module	Number	SUF-files	Main contents
Job Tasks	26b ET	pTarget	The job task module collects various job task types that describe the main employment spell.
Unemployment	27 AL	spUnemp	The module records all current and past periods during which participants were unemployed, independently of any registration with the Federal Employment Agency.
Further Education Activities and Informal Learning	35 KU	spCourses	The course module collects further training activities in the life course modules.
	31 WB	spFurtherEdu1 spFurtherEdu2	The further training modules capture all further training activities that were not reported in the previous modules.
Partnerships	28 PA	spPartner	Information on partners are collected, such as gender, date of birth, migration background, nationality, highest educational degree, current employment status and current profession.
Children	29 KI	Children spChild spChildCohab	The modules capture information on respondents' children and …
Parental Leave	29 EZ	spParLeave	… parental leave episodes.
Retirement	38 RE	pTarget	The module captures different types of retirement, the individual experience of retiring as well as reasons for employment alongside retirement.
Residence History	21 WG	spResidence	In this module, episodes of place of residence are collected and updated over the life course (ALWA study)[11].

(...)

11 ALWA: *Working and Learning in a Changing World*. Respondents of the ALWA study were already surveyed by the IAB in 2007/2008 and later transferred to the NEPS. For more information, see section 2.1

Table 9: (continued)

Module	Number	SUF-files	Main contents
Gap	50 LU	spGap	The gap module covers all temporal gaps between the main life course activities and collects the activities practiced within these gap periods.

5.2 Differences between initial survey and panel survey

In order to ensure that the retrospective record of the educational trajectory and the employment history is precise and complete, the survey is structured by life domains. The life history is split into different survey modules. Each of them covers the topic associated with that domain and captures corresponding activities, for example the (monthly) duration of school attendance.

In the initial survey (usually in the year of the first wave of a particular subsample) the entire biography of an interviewee is recorded retrospectively. Those initial surveys took place for the ALWA subsample in 2007/2008 and for the NEPS subsamples in the waves of 2009/2010 and 2011/2012. In order to collect the biographical data, the activities within each module are recorded, starting with the first activity and ending with the current activities (the ongoing activities at the date of the interview, if applicable). An exception to this approach is the partner module, as its starting point is the current partner followed by information about former partners.

Once the biography was initially recorded, the participant's biography is updated in each consecutive panel wave. Hence, the data from previous waves is used to adapt the questionnaire. Firstly, follow-up questions concerning activities recorded in the previous interview are asked. The interviewee can object preloaded information in case it was recorded incorrectly in the earlier interview, otherwise the respective episode continues. Secondly, new activities are recorded that have started (and ended) since the last interview. Those new activities are also recorded chronologically per wave until the date of the current interview. Thereby, biography data are completed wave by wave, in each case referring to information from the previous wave.

5.3 Further information on data files

5.3.1 Vocational Training

SUF file spVocTrain

Module 24 AB

The vocational training module captures vocational or academic education for example vocational training, college education or post-graduate degrees. Even if the educational activity is not completed, it is recorded in this module. If an individual participates in multiple educational activities at the same time, all activities are recorded as individual episodes. If an episode is no longer ongoing, the episode's end is the day of graduation or the day of dropout.

Amongst others, the vocational training module enquires about the type of vocational training taken; the location and duration of the training; the type of contract including salary; the degree obtained in addition to the vocational training degree and the grade; as well as the satisfaction with the vocational training and for dropouts, the reasons for dropping out of the educational activity.

For college degrees, a new episode is captured if the major subject or the type of degree to be acquired changes. Changes in minors or changes in colleges are disregarded. For post-graduate degrees, characteristics of the degree are recorded.

Respondents are being asked if there have been any interruptions during their vocational training. If this is the case, respondents can report up to three so called *interruption episodes*. These are stored in wide format and only include the start and end date of the interruption. This should be kept in mind when working with the harmonized spell data.

In general, further training activities are captured in the further training module or the course module (see section 5.3.6). In some cases, trainings are recorded in this vocational training module, for example if they lead to a license in case of particular IHK[12] courses.

After general vocational training, the module enquires about further educational episodes made in the context of external examinations (*Externenprüfungen*).

Special issues

Grades

- No grades were recorded in the first wave for the vocational training degree or the Ph.D.

12 Industrie- und Handelskammer (Chamber of Commerce and Industry)

- The second and third wave recorded grades only in the cross-section, i.e., for the last college degree or the last Ph.D. Grades for degrees other than college or Ph.D. were not surveyed.
- In wave 4, grades of degrees were integrated into the longitudinal survey for vocational training of any kind and Ph.D.s.

ALWA

ALWA (`wave==1`) does not provide information on the location of the vocational degree provider. Data users can apply a *best-guess* approach by using the data file `spResidence`, which contains the history of a respondent's places of residence.

Wave-specific

For wave 11, licensed courses and IHK courses are not captured in the vocational training module as it was before. The idea was that respondents report them directly in the further training module (see section 5.3.6) to save overall survey time. For wave 11 these courses are stored in the data file `spFurtherEdu1`. Unfortunately, it then turned out that this change leads to an underreporting of such courses. Thus, this process has been changed again for wave 12 and beyond to the following:

- IHK courses are again reported in the vocational training module and are hence stored in the `spVocTrain` data file.
- Licensed courses are now captured in the course module (see section 5.3.6) and are therefore now stored in the data file `spCourses`.

5.3.2 Military

SUF file spMilitary

Module 25 WD

Changes over time

Wave 1 - 3

In the first survey waves the military module has been used to collect episodes of basic military services, community services and alternative services, as well as episodes of voluntary social years, ecological years or European voluntary services.

Wave 4

The categories were expanded to also include federal voluntary service and voluntary military service.

Wave 6

Basic military service, community service and alternative service were removed due to the decision of the federal cabinet in 2011 to abolish these compulsory services. Since this wave, the episode types in this module consist exclusively of voluntary services. Also, an additional category, international youth voluntary service, was added.

5.3.3 Employment

SUF files spEmp, pTarget

Modules 26 ET

In Starting Cohort 6 the longitudinal information on employment has two components: The information on employment episodes is covered by the annual employment module of the questionnaire (26 ET) and the data is stored in the `spEmp` data file.

Additional information on employment is gathered at larger intervals as supplementary employment modules in the questionnaire (26a ET - 26g ET). This additional information is stored in the `pTarget` data file.

Content in spEmp

The data file `spEmp` captures information on all employment episodes that respondents reported on gainful employment, e.g., all activities leading to income. This includes regular employment, but also traineeships and secondary jobs. Vacation jobs, volunteering, and paid or unpaid internships are not covered by the `spEmp` file.

Due to the modular recording of the life course, the collected information on the employment situation is not restricted to one job at the same time but also contains information on parallel jobs. There is no restriction to a specific number of parallel jobs.

In the questionnaire of the employment module, the introduction statements give specific anchors for respondents to report regular employment, traineeships and secondary jobs. However, note that these different employment types can be reported anytime in the course of the module. It is also important to note that the `spEmp` data file does not contain a clear information from the respondent whether a job represents a main or a secondary employment or whether a job is a main or secondary activity in comparison to activities reported in the other life course modules. Since such a decision highly depends on the research question of interest, we strongly recommend a conception-based and thorough edition of the employment episodes together with the life course data of Starting Cohort 6 that suits the underlying research purpose. As a starting point Rompczyk and Kleinert (2017) provide an instruction on how to edit the life course data of Starting Cohort 6. You can also find more information about this in section 4.4 of this data manual.

The employment episodes in the spEmp data file cover information on the following topics:

- Occupation coded in different German and international classifications
- General employment type and detailed information about the employment type, e.g., supervision and management tasks, temporary employment, whether the reported employment is situated in the subsidized labor market, whether the reported job is contract work or seasonal employment
- Working hours and for respondents at the age of 55 years or older, whether they take part in a partial retirement scheme
- Company characteristics and conditions for the participation in further training, courses and seminars
- Gross and net earnings for employees as well as the profit before and after taxes for self-employed

Changes over time in spEmp

Wave 16

`ts23249` Variable on telework from the Covid-19-module (`th18090`) was adopted and included in the ET-module

Wave 14

`ts23228` "Master Professional" included for item on required educational qualifications

Wave 13

`ts23217` Seasonal work is no longer recorded as one continuous episode at a time, but each episode of seasonal work separately.

Wave 11

New variables:

- `ts23208` Mini-jobs
- `ts23247` Termination of the job (dismissal/quitted)
- `ts23248` Chain contracts for fixed-term contracts

Other changes:

- `ts2333m` and `ts2333y` Items deleted in questionnaire
- `ts23320`, `ts2332m`, `ts2332y`, `ts23244`, `ts23245`, `ts23246` Comprehensive filter adjustments in the questionnaire[13]

13 complex filters in the questionnaire were simplified and made clearer e.g., by filtering all respondents into a time stamp variable and then defining new groups for filtering into the next items. The adjustments also include corrections of filters for some of the mentioned variables.

- ts23552 No longer asked whether subsequent employment with the same employer was already reported
- ts23229, ts23230, ts23231, ts23232, ts23233, ts23234, ts23243 Yearly updates introduced
- ts23410–ts23546 In addition to the yearly income updates, for all completed episodes, the complete income information is collected at the end of the employment spell.

Wave 10

ts23215 Value 1 "ABM jobs [labor market measure jobs]" deleted in questionnaire.

Wave 8

New variables:

- ts23256 Student job or other job
- ts23257 Relation/relevance to studies (student employment)

Wave 7

New variables: ts23553 Contractual working time currently/at the end.

Wave 5

ts2312m, ts2312y, ts23219, ts23221, ts23223 Additional interviewer references added, which imply a change in the definition of employment episodes and working time (due to the introduction of items on partial retirement schemes, cf. th32218 in pTarget).

Wave 4

New variables:

- ts23251 Type of employment
- ts23237 Additional Items in the questionnaire to differentiate for the place of work in Berlin Berlin Mitte/Berlin Pankow/Berlin Lichtenberg
- ts23239 Country of working location (open text)
- ts23552 Subsequent employment with the same employer already reported
- ts23531 Special payment: 13th month salary
- ts23541 Special payment: 13th month salary (gross)
- ts23532 Special payment: 14th month salary
- ts23542 Special payment: 14th month salary (gross)
- ts23533 Special payment: Christmas bonus
- ts23543 Special payment: Christmas bonus (gross)

- ts23534 Special payment: Holiday pay
- ts23544 Special payment: Holiday pay (gross)
- ts23535 Special payment: bonus, profit-sharing, gratification
- ts23545 Special payment: Bonus, share of profits, gratuity (gross)
- ts23536 Special payment: other
- ts23546 Special payment: other (gross)

Content in pTarget

A special feature of collecting information on employment in Starting Cohort 6 is the additional panel information that goes beyond the episode data of the spEmp data file and is stored in the pTarget data file. However, this additional information is not available for every topic on employment in every wave. Table 10 shows all additional topics related to employment ever covered and in which waves the topics were covered. For example, in the first, second and third waves, only the main questions on employment were covered, whereas the job tasks were covered in the fourth, eighth and twelfth waves. It is important to note that this additional panel information is only collected for the main employment episode if there are several parallel employment episodes at one time. It is up to the respondent to decide which episode this is.

Changes over time in pTarget

To allow for longitudinal analyses, the questionnaire holds the phrasing of the questions as constant as possible. However, the phrasing of some of the questions had to be adjusted over the years. Furthermore, the module does not ask all questions within a particular topic in every wave. Thus, the number of variables in each topic can vary across the waves. You can find further information on documentation and changes within the questionnaire in additional documentation (see section 1.2 for more information).

NEPS-ADIAB: employment data linked to administrative data of the IAB

As an option to add more information on the employment reported in the NEPS survey, e.g., on income or on the company, the Research Data Center of the Institute for Employment Research (FDZ-IAB) offers administrative data containing additional labor market information of the NEPS respondents. The linkage of these data is conditional on the consent of the survey respondents. Furthermore, the administrative data only cover the following employment status: employment liable to social security (since 1975), marginal employment (since 1999), receipt of benefits under the SGB III legal system (since 1975), or SGB II (since 2005), registered as a jobseeker at the Federal Employment Agency (since 2000), or (planned) participation in labor market policy measure (since 2000). Therefore, the administrative data do not cover all types of employment, such as self-employment or civil service. For a detailed data report as well as information of labels and frequencies, see Bachbauer and Wolf (2020).

Table 10: Items on employment by wave

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Annual questionnaire	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Job Tasks				x				x			x					x	
Occupational Change						x		x		x							
Health burden of work						x				x							
Social capital and work climate				x		x				x			x				
Language use with colleagues						x				x					x		
Social capital & labor market resources				x			x				x			x			
Work-Life-Conflict											x		x			x	
Job characteristics											x	x	x	x	x	x	x
Time and performance pressure											x		x				
Digitalization of work												x	x	x	x	x	
Digital Assistance Systems																x	

5.3.4 Job Tasks

SUF file pTarget

Modules 26b ET

Starting Cohort 6 contains further information about job tasks performed in an employment episode that is stored in `pTarget`. In case a respondent reports multiple parallel employment episodes, the job task information is collected only for the main employment. In this case, the respondent him- or herself decides which episode spell (stored in `spEmp`) is the main gainful activity. The data user can merge the data on job tasks from `pTarget` to the main employment spell by using the variables `th14599_g1` from `pTarget` and `splink` from `spEmp`.

Content

The job task data (variables: `th343000` to `th34395`) contains information on how much respondents read or do math in their jobs, whether they do calculations, work with money, measure something, operate a computer and what ICT skills they need for this. Furthermore, it contains information about whether respondents solve difficult problems, often learn new things or whether their work involves different routines. The data further contains information on how autonomously they can perform their work, whether they are exposed to physical exposures during work and whether they interact with others. For further information see Matthes et al. (2014).

Changes over time

So far, the job task module has been used every 4 years in wave 4 (2011/12), wave 8 (2015/16),

wave 12 (2019/20), and wave 16 (2023/24).

Wave 16

- Wording adjustments to standardize the catalog of questions that belong together (th34303, th34313, th34390).
- Adaptation to technological progress when listing working tools, e.g. writing on a computer and tablet instead of a typewriter (th34310).
- Deleting an example for programming languages (th34335).
- Updating the definition of computers in the interviewer instructions (th34330).

Wave 12

- Updating the definition of computers in the interviewer instructions (th34330).

5.3.5 Unemployment

SUF files spUnemp, pTarget

Module 27 AL

The unemployment module captures all periods during which participants were unemployed. Participants are considered unemployed if they are registered as unemployed or if they are not working, but actively seeking work.

Content

When a respondent participates in the NEPS survey for the first time, the module records current and past unemployment episodes retrospectively over the entire life course. In the subsequent waves, the module collects all current and past unemployment episodes since the last interview. In addition, the data provides further information on the unemployment episode, the application process and on further training during unemployment. The data file contains information on the following topics:

- Registration of unemployment, receipt of unemployment benefit
- Number of job applications and job interviews
- Courses/further training during unemployment, financed by the employment agency
- Job search efforts in the last four weeks (file pTarget)
- Possibilities to start a new job within two weeks (file pTarget)

Special issues

The module does not distinguish between different types of unemployment. Therefore, no new unemployment episode starts when changes occur (e.g., in registration of unemployment or benefit). Therefore, there are no consecutive unemployment episodes, as the module records the entire period of unemployment in one piece. Furthermore, the module does not record unemployment episodes for periods immediately before the end of training or employment, even if the respondent is already registered as a job seeker because of the three-month registration deadline in German employment agencies.

Changes over time

In the first and third wave, variables on job search efforts (`th09211`) and possibilities to start a new job (`th09212`) are not available.

5.3.6 Further Education Activities and Informal Learning

SUF files pTarget, spCourses, FurtherEducation, spFurtherEdu1, spFurtherEdu2

Modules 35 KU, 31 WB

Further training activities are captured throughout the questionnaire mainly in two particular modules: First, the course module (35 KU) collects further training in the life course modules, such that respondents recall context-specific further training activities, for example courses taken during employment or during parental leave episodes. Second, the further training module (31 WB) captures all further training activities of all respondents in Starting Cohort 6, which have not been reported in earlier modules.

In addition, the vocational training module (see section 5.3.1) captures licensed courses and other vocational trainings which were identified as further training courses.

Content: Course module

When respondents state having participated in further training within an episode, this statement triggers the course module. For up to five courses per episode, the course module records the content, duration and motivation (occupational vs. private reasons) for the course. Further

training activities are not recorded exactly to the date, but rather the respondents recall all training activities since the last interview or since the beginning of an episode.

Content: Further training module

Towards the end of the interview, the further training module captures all further training activities (classes, courses and seminars) in which the respondents participated since the last interview and have not yet reported. The module records all types of further training including classes taken out of personal interest, such as cooking or yoga classes.

At the beginning of the module, the interviewer reads all further training activities collected through the course module to the respondent and asks whether the respondent has participated in any other further training activities since the last interview. The further training activity's name is recorded in the further training module along with its content, duration and completion. Additionally, information on the motivation (private vs. occupational), whether the class was mandatory and whether the respondent received a certificate is captured.

Further information on randomly chosen courses are also collected within this module, for example (among others) financial support for the course, provider of the training and evaluation of the course.

Irrespective of having participated in a further training activity, the module asks about informal learning in five questions, i.e., whether the respondent has participated in any informal learning activity such as reading scientific literature, attending a conference or learning with online apps or programs.

Special issues and changes over time

Assignment of further training activities across waves

Further training activities that were not completed at the time of the interview are not incorporated in the next wave as a preload, which means that they might be reported again. It is not evident to assign a further training activity from the last wave to the next:

- The respondent would have to phrase the name of the further training activity exactly the same way in both waves.
- A respondent can report two different further training activities within the same field, even if the content and names of the further training activities are the same, for example two yoga classes.
- For the download SUF: Only the categorical variable for further training course content (`tx28202_g13`) allows the assignment of a training activity from the previous wave to the next. However, due to data protection reasons, the course content is aggregated and categorized in this variable, therefore identically categorized courses have a high likelihood of actually being different further training activities.

Assignment of vocational training to further training

Vocational training courses are being integrated into the data file `FurtherEducation` when it pertains to licensed courses (`ts15201_v1 == 13 OR ts15201 == 14`) or vocational trainings which were identified as further training during the data edition process (`ts15291_g12 == 2`).

Double recording of IHK courses

In some rare cases, IHK courses are recorded in the vocational training module, but are then not assigned to the list of courses that is read to the respondent at the beginning of the further training module. Therefore, it is possible that the respondent reports this course a second time. However, it is not evident to identify these doubly recorded courses. This error was fixed in wave 12.

Additional information on further training activities

Starting in wave 11, additional information on further training activities is no longer only surveyed for randomly chosen courses, but for all courses. Therefore, these information are now stored in `spFurtherEdu1` instead of `spFurtherEdu2`. This applies to the following items:

- Private vs. occupational reasons for further training participation
- Motivation for participating
- Whether the course was mandatory and who made it mandatory
- Certificates[14]

Randomly chosen further training activities

- Out of all further training activities collected throughout the interview, one is chosen randomly. For the randomly chosen further training activity, additional questions are asked, for example on the learning atmosphere, the courses' structure and its difficulty. Before wave 11, two further training activities were chosen randomly.
- Further training activities, randomly chosen for additional questions, were meant to only include further training activities that had already been completed. However, until and including wave 12, further trainings from the course module were also chosen when they were not yet completed at the time of the interview. This error was corrected while wave 12 was in the field but the SUF data file `spFurtherEdu2` still contains additional information on ongoing further training activities from the course module. No mistakes were made for the further training module and the vocational training module. Thus only completed further training activities were chosen from these two modules.

14 Please note that in wave 12 there were changes on the value scale of the certification variable in the vocational training module and the further training module. These changes made it necessary to create two variable versions. The old version of the variable can be identified by the suffix `_v1` added to its variable name.

Merging FurtherEdu2 with FurtherEducation

- The variable course, which is important for the merging process between FurtherEducation and FurtherEdu2, has many missings for courses which were reported in the vocational training module. This has two reasons:

  1. Only licensed courses have a value assigned in the course variable because only these courses are taken into consideration in the random selection process for the additional detailed questions.

  2. Licensed courses which were completed a long time ago (more than 12 months to last interview) are also not considered in the random selection process and thus have a missing value in the course variable. In order to merge the two data files, courses with missings have to be dropped.

- When merging the data files FurtherEdu2 with FurtherEducation, five further training episodes from the vocational training module cannot be merged. This is due to small errors in recording further training activities and ensuing difficulties in assigning the additional information collected in the further education module for randomly chosen further training to the further training episode captured in the vocational training module. Therefore, these five cases have to be excluded in the analyses.

Licensed and IHK courses

Licensed courses and IHK courses were no longer captured in the vocational training module (see section 5.3.1) starting in wave 11. The idea was that respondents report them directly in the further training module and hence save survey time overall. For wave 11 these courses are stored in the data file spFurtherEdu1. Unfortunately, it then turned out that this change lead to an underreporting of such courses. Thus, this process has been changed again for wave 12 and beyond to the following:

- IHK courses are again reported in the vocational training module and are hence stored in the spVocTrain data file.

- Licensed courses are now captured in the course module. Therefore, new courses of this type are now stored in the data file spCourses.

Additional information on informal learning activity - missings in wave 15

In a few rare cases in wave 15 some further questions on the informal learning activity "Informal media - digital" (t271805) have not been posed due to filtering issues. Missing Values in the following Variables were marked with an appropriate errorcode: t272800_g1, t272805_0, t279840_w4, t272802_w4. This issue was fixed during field time in wave 15.

5.3.7 Partnerships

SUF file spPartner

Module 28 PA

The NEPS uses the following partnership definition:

> A partnership is a fixed relationship of two people living together or apart – independently of the legal status (married, married and living apart, divorced, widowed, registered partnership).

For participants entering the survey for the first time, the module records the current partnership at the time of the interview and asks for preceding partnerships. After the first survey participation, the module records all partnerships since the last interview that correspond to the definition of partnership. For subsequent interviews, if the partner did not change since the last interview, the interviewer asks whether there has been a change in legal status of the partnership, if the partner acquired an additional degree or professional qualification, and the current employment situation of the partner.

In case there has been more than one partnership since the last interview, the interviewer starts with the first and ends with the current partnership. The module also asks for additional information on the partner such as gender, date of birth, migration background, nationality, highest degree, current employment status, and current profession – whereas the last two points are only asked for the current partnership.

The module does *not* record multiple overlapping partnerships.

Changes over time

Wave 11

- The module takes the legal introduction of the *marriage for all* in 2017 into account. Starting with wave 11 registered same-sex partnerships are only continued or annulled. The module asks respondents living in a same-sex partnership whether they married their partner. If this applies, the survey handles the same-sex partnerships like other marriages. Therefore, same-sex marriages are only identifiable by comparing the sex of the partners.

Wave 13

- *Living Apart Together* (LAT) partnerships are couples having an intimate relationship but live at separate addresses. Up to wave 13 these partnerships were not treated in the partner module but in a cross-sectional module. As a result, these partnerships were not continued but treated as new partnerships in each cross-section in which the respondent participated. Consequently, respondents had to answers all questions on the LAT partnerships and partner in every wave, irrespective of the continuation of the

same partnership. As of wave 13, the partner module captures all new and continuing partnerships, irrespective of living together or apart. For existing partnerships, the module asks whether the partnership continues. For marriages, if respondents answer this question in the affirmative, the module assumes that also the marriage continues. If partnerships are discontinued, the module asks whether the marriage still persists and records the divorce date, if applicable. For unmarried partnerships, which already existed in the last wave, the module records whether a marriage took place, followed by the question whether the partnership is prolonged. If the partnership is discontinued, or was discontinued at the time of the last interview, the module captures whether the marriage (or registered partnership) is also discontinued by now.

- The survey now records the death of a partner at three points in the partner module. If the partner is deceased, the module records the date of death. The module asks no further questions on the deceased partner.
- The introduction of the LAT concept in the partner module resulted in changes in the way the module captures the history of cohabitation. Up to wave 13, the survey assumed that partners either live together or apart but did not cover discontinuity in cohabitation. Now, the module covers cases where partners do not steadily cohabit.
- Starting with wave 13 the survey of the partners' education, training, and employment changed. The module does no longer provide the possibility to disagree with information given in earlier waves on this topic, as the respondents seldom used this possibility. The information on the partners education, surveyed previously, is now used to filter question on the highest school-leaving qualification of the partner (as *obtaining a high school diploma* makes questions on the highest school-leaving qualification redundant).
- Starting with wave 13, the survey records the contact frequency for all partners captured in the partner module.

5.3.8 Children and Parental Leave

SUF files Children, spChild, spChildCohab, spGap, spParLeave

Modules 29 KI, 29 EZ

Information on respondents' children and parental leave episodes are captured throughout the questionnaire in two modules: First, the children module (29 KI) collects data on respondents' children and related living conditions. Second, the parental leave module (29 EZ) captures information on parental leave episodes as part of the life course.

Content: Children module

The children module is queried for all respondents of the study. Respondents without children

are only asked about their further care activities. Respondents with children pass through the whole module. Information on all adopted, foster, biological and children living in the same household are collected.
There is an item loop for every child. It consists of items on sociodemographic information, episodes of living together in one household and episodes of parental leave.

If the respondent reports an episode of parental leave, a redirection to the parental leave module (see below) follows, as well as a redirection to the course module (see section 5.3.6) in case of further training activities during this episode. Back in the children module, child-specific information on the childcare situation, educational aspirations, the current activity status, and educational and vocational certificates are recorded.
In the end of the module, there are some general cross-sectional questions about the respondent's engagement in childcare and further care activities.

Content: Parental Leave

The parental leave module was created in wave 13 as a decoupling of items from the children module. Respondents are redirected to the parental leave module if they indicate an episode of parental leave in the children module or in the data revision module. The episode dates are recorded in the original module. Information on administrative issues, and the re-entry into employment are part of the parental leave module. As the parental leave module can be accessed from both the children module and the data revision module, both the `spParLeave` SUF file and the `spGap` file contain some cases.

Since a parental leave is recorded in form of a life course episode, parental leave episodes are considered during the life course check in the data revision module. Often such an episode runs parallel to e.g., an employment episode, even if the respondent was not working during the parental leave. This is due to the fact that in the interview all other life course episodes are collected before the recording of parental leave.

Changes over time

Wave 12

- Items on the care situation of every child (as part of the child loop) are added to the module.

Wave 13

- The items on the employment re-entry after parental leave are decoupled as a new module. Thereby, it is possible to collect the information although a parental leave episode is added to the life course during the data revision module.
- The definition of parental leave has changed. Previously, the respondents were asked to indicate parental leave only if they had a legal right to this parental leave and did not work more than 30 hours per week. As of wave 13, the definition of a parental leave is up to the respondents.

5.3.9 Retirement

SUF file pTarget

Module 38 RE

Variables regarding retirement are part of the `pTarget` data file. The module captures different types of retirement, the individual experience of retiring, as well as reasons for employment alongside retirement.

Content

The module captures whether respondents currently receive pension payments, such as a statutory pension or state pension, a widower's pension, or a disability or invalidity pension. Furthermore, it records whether the respondents receive private/corporate pension funds or basic income support. For the first pension payment, the module records the month and the year as well as several information on the individual experience considering the entrance in retirement.

Retired respondents can update annually whether they currently work alongside the retirement or if they plan to do so. The module also captures possible reasons for working alongside retirement. If respondents are in partial retirement, the module asks about the time of the active phase respectively. Considering partial retirement and the date of partial retirement, it is important to distinguish which partial retirement model the respondents attended (block model or part-time model).

Target group

Besides respondents who were retired in earlier interviews, respondents who are 55 years and older at the time of the interview automatically enter this module. Irrespective of the age, respondents having a gap in their life course enter the data revision module and can declare a retirement episode.

Changes over time

The module on retirement was introduced in wave 5, to face the maturing panel and the resulting increase in the share of retired individuals. Since then, the NEPS survey records information on retirement annually without substantive changes.

5.3.10 Residence History

SUF file spResidence

Module 21 WG

The residence history is only collected for respondents who have been sampled for the ALWA study. Respondents who joined Starting Cohort 6 with the NEPS survey only indicate their current place of residence at the time of the interview (stored in data file `pTarget`).

In this module, episodes of place of residence are collected and updated over the life course. First, it is recorded whether the place of residence is in Germany. If it is in Germany, the community is identified; if the place of residence is abroad, the country is identified. A list of communities and countries, which is stored in the module, helps to enter the respondent's information.

The aim is to collect all places (local communities) a respondent resided in since birth.

In addition to changes of residence due to moves, we are also interested in relocations of at least one month's duration for professional or educational reasons (as well as au pair activities).

Episodes cover the period between moving in and leaving the residence at the respective location. If the respondent stays in more than one location at the same time, the data records two individual episodes (that simply overlap in time).

Capturing parallel residence episodes

- The stay in the two places of residence is regular and important
- The interviewee resides in each residence for at least three days per week
- The module captures the location where the respondent lives and the location where the respondent works. If these are not fixed addresses, the item *"changing locations"* is available.
- If an interviewee is registered with his/her parents (for example as a student), but only stays there once a month, the study location will be captured as place of residence instead.

No new residence episode starts if the respondent

- moves within the same location or community
- commutes daily between two locations
- is on vacation (less than three months)

5.3.11 Gap

SUF file spGap

Module 50 LU

Immediately after collecting the main activities of the life course of a respondent with the modules school, vocational preparation, vocational training, military, employment, unemployment and parental leave, the data revision module checks, among other things, whether there are any chronological gaps between these main activities. If this is the case, these chronological gaps are closed in collaboration with the interviewee, either by specifying an additional main activity by the interviewee that closes the gap, or by specifying an activity that is not covered by the main activities, a so-called gap activity.

In the first case, the survey instrument branches from the data revision module back to the corresponding module for the main activity. There, a new episode with a main activity will be collected to close the chronological gap in the life course. Then, the survey instrument filters back into the data revision module.

In the second case, i.e., if no main activities were exercised in the chronological gap, the gap module will be activated. Here, other activities can be specified to fill the chronological gap, such as *housewife/househusband*, *sick/unable to work*, *retirement* et cetera. In this respect, unlike the main activity modules, the gap module is only used within the data revision module to fill in chronological gaps.

One exception to the closing of chronological gaps in the data revision module is the main activity *parental leave*. Since the recording of parental leave episodes does not take place in a standalone module, but is embedded in the child module and is done there specifically for each child separately, a direct return from the data revision module to the corresponding main activity module is not possible. Instead, an episode of parental leave recorded in the data revision module is treated as a gap activity. This means that instead of the main activity module, the gap activity module will be activated and the parental leave is recorded there, but without specific reference to a child and only regarding information to start and end date of the episode. As of survey wave 13, additional information on the employment of the interviewed person during this parental leave is collected in the gap module in the same way as in the parental leave module (see section 5.3.8).

Although the gap module is only used to fill chronological gaps, it is possible that gap activities and main activities overlap chronologically. On the one hand, this can be the case if after collecting a gap activity, other activities are collected in the data revision module and these activities overlap chronologically confirmed by the interviewed person. On the other hand, there may be chronological overlaps if a gap activity persisted at the time of the interview and was pursued further in the subsequent survey wave. In this case, the gap episode is continued regardless of the existence of a gap for this period in the life course.

5.4 Adjustments to items from the Corona module

In the Scientific Use Files of versions 13, 14 and 15, some changes were made to names of variables from the Corona module in the dataset `pTarget`:

Firstly, the variables listed below were renamed in version 15.0.0 to standardize them with the other variable names from the Corona module in accordance with the general NEPS conventions.

- `tm00055` -> `th18020`
- `tm00056_0` -> `th18021_0`
- `tm00090` -> `th18040`
- `tm00091` -> `th18041`
- `tm00092` -> `th18042`
- `tm00093` -> `th18043`
- `tm00094` -> `th18044`
- `tm00095` -> `th18045`
- `tm00114` -> `th18050`
- `tm00115_0` -> `th18051_0`

Secondly, in version 15.0.0 of the Scientific Use File, the original suffix `_w1` was renamed to `_v1` in accordance with the NEPS conventions. It indicates the variant of a variable with the same name. The reason for the need of variants concerning variables from the Corona module is the different time reference in the question texts. They came into use in two separate modules on the Corona pandemic in survey wave 13 – module 36a for “temporary dropouts” (=respondents who did not participate in the last survey wave) and module 36b for “repeaters” (=respondents who participated in the last survey wave). This distinction is reflected in `pTarget` by two variants of each variable. Information from the “temporary dropouts” is identified by the suffix `_v1` (formerly `_w1`); information from the “repeaters” is contained in the corresponding variables without suffix.

A References

Allmendinger, J., Kleinert, C., Antoni, M., Christoph, B., Drasch, K., Janik, F., Leuze, K., Matthes, B., Pollak, R., & Ruland, M. (2011). Adult education and lifelong learning. In H.-P. Blossfeld, H.-G. Roßbach, & J. von Maurice (Eds.), *Education as a lifelong process: The German National Educational Panel Study (NEPS)* (pp. 283–299, Vol. 14). VS Verlag für Sozialwissenschaften.

Allmendinger, J., Kleinert, C., Pollak, R., Vicari, B., Wölfel, O., Althaber, A., Antoni, M., Christoph, B., Drasch, K., Janik, F., Künster, R., Laible, M.-C., Leuze, K., Matthes, B., Ruland, M., Schulz, B., & Trahms, A. (2019). Adult education and lifelong learning. In H.-P. Blossfeld & H.-G. Roßbach (Eds.), *Education as a lifelong process: The German National Educational Panel Study (NEPS)* (2nd ed., pp. 325–346, Vol. 3). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-658-23162-0

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).

Bachbauer, N., & Wolf, C. (2020). *NEPS-SC6 survey data linked to administrative data of the IAB (NEPS-SC6-ADIAB)* (FDZ-Datenreport No. 04/2020 (en)). Institute for Employment Research (IAB). Nürnberg, Germany. https://doi.org/10.5164/IAB.FDZD.2004.en.v1

Blossfeld, H.-P., & Roßbach, H.-G. (Eds.). (2019). *Education as a lifelong process: The German National Educational Panel Study (NEPS). Edition ZfE* (2nd ed.). Springer VS. https://doi.org/10.1007/978-3-658-23162-0

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*.

Drasch, K., & Matthes, B. (2013). Improving retrospective life course data by combining modularized self-reports and event history calendars: Experiences from a large scale survey. *Quality & Quantity*, *47 (2)*, 817–838. https://doi.org/10.1007/s11135-011-9568-0

FDZ-LIfBi. (2025). *Data Manual NEPS Starting Cohort 6 – Adults, Adult Education and Lifelong Learning, Scientific Use File Version 16.0.0*. 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.

Hess, D., Steinwede, A., & Schneider, B. (2012). *Erhebung von retrospektiven Längsschnittdaten - Prüfmodul*. Bonn, infas Institut für angewandte Sozialwissenschaft GmbH.

Künster, R. (2015a). *Startkohorte 6: Erwachsene (SC6) Datenversion 5.0.0. Technical Report 1: Edition und Korrektur der Lebensverlaufsdaten*. Leibniz-Institut für Bildungsverläufe, Nationales Bildungspanel. Bamberg, Germany.

Künster, R. (2015b). *Startkohorte 6: Erwachsene (SC6) Datenversion 5.1.0. Technical Report: Korrektur der Lebensverlaufsdaten*. Leibniz-Institut für Bildungsverläufe, Nationales Bildungspanel. Bamberg, Germany.

Malina, A., Wefelmeyer, D. M., Ruland, M., & Aust, F. (2023). *Feld- und Methodenbericht. NEPS-Startkohorte 6 (Erwachsene) – Haupterhebung 2022/2023, Teilstudie B158*. Bonn, Germany: infas Institut für angewandte Sozialwissenschaft GmbH.

Matthes, B., Christoph, B., Janik, F., & Ruland, M. (2014). Collecting information on job tasks—an instrument to measure tasks required at the workplace in a multi-topic survey. *Journal for Labour Market Research*, *47*(4), 273–297. https://doi.org/10.1007/s12651-014-0155-4

Matthes, B., Reimer, M., & Künster, R. (2005). TrueTales – ein neues Instrument zur Erhebung von Längsschnittdaten. In *Arbeitsbericht 2 des Projektes „Frühe Karrieren und Familiengründung: Lebensverläufe der Geburtskohorte 1971 in Ost- und Westdeutschland“*.

Matthes, B., Reimer, M., & Künster, R. (2007). Techniken und Werkzeuge zur Unterstützung der Erinnerungsarbeit bei der computergestützten Erhebung retrospektiver Längsschnittdaten. *Methoden, Daten, Analysen – Zeitschrift für Empirische Sozialforschung*, *1*(1), 69–92.

NEPS Network. (2025-a). *National Educational Panel Study, Scientific Use File of Starting Cohort 6 – Adults*. Leibniz Institute for Educational Trajectories (LIfBi), Bamberg. https://doi.org/10.5157/NEPS:SC6:16.0.0.

NEPS Network. (2025-b). *Starting Cohort 6 – Adults, Wave 16, Questionnaires (SUF Version 16.0.0)*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Pelz, S. (2025). *NEPS Technical Report: Implementation of the ISCED-2011, CASMIN and Years of Education Classification Schemes in SUF Starting Cohort 6*. 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.

Rompczyk, K., & Kleinert, C. (2017). *Episode-split biography data in NEPS starting cohort 6: structure and editing process* (NEPS Survey Paper 28). Leibniz-Institut für Bildungsverläufe, Nationales Bildungspanel. Bamberg, Germany. https://doi.org/10.5157/NEPS:SP28:1.0

Ruland, M., Drasch, K., Künster, R., Matthes, B., & Steinwede, A. (2016). Data-Revision Module - A Beneficial Tool to Support Autobiographical Memory in Life-Course Studies. In H.-P. Blossfeld, J. von Maurice, M. Bayer, & J. Skopek (Eds.), *Methodological Issues of Longitudinal Surveys. The Example of the National Educational Panel Study* (pp. 367–384). Springer VS.

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. 10). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Scharl, A., & Zink, E. (2022). NEPSscaling: plausible value estimation for competence tests administered in the German National Educational Panel Study. *Large-scale Assessments in Education*, *10*(28). https://doi.org/10.1186/s40536-022-00145-5

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

B Appendix

B.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. Just like there, the examples become more adaptable if some variables are defined beforehand:

```
# Starting Cohort
cohort  <- "6"

# version of this Scientific Use File
version <- "16-0-0"
```

To further ease the readability and shorten the examples, we also define a function `read.neps()`. Please note that you also need the libraries `readstata13` and (optionally) `Hmisc` for this to work. If you do not have those libraries installed on your computer, you can easily do so by executing the command `install.packages("readstata13")` from inside R.

R 40: read.neps()

```
library(readstata13)
library(Hmisc)

## convenient wrapper function to 'read.dta13()'. Example of usage:
## cp <- read.neps("CohortProfile")
##
read.neps <- function(token,path="Z:/SUF/Download"){

  # absolute path to the file. Might need some adaption in your setting!
  # the current definition refers to
  # "Z:/SUF/Download/<cohort>/<cohort>_<version>/Stata14/
  #  <cohort>_<token>_<version>.dta"
  file <- paste0(
            path,"/",
            cohort,"/",
            cohort,"_",
            version,
            "/Stata14/",
            cohort,"_",
            token,"_",
            version,
            ".dta"
          )

  # read the data
  data <- read.dta13(file, convert.factors = F)

  # set the language to english (comment this out if you work in german)
  data <- suppressWarnings(set.lang(data, "en"))

  # The following step is not absolutely necessary.
  # However, it is recommended if you find it convenient to have the variable
  # labels handy during your analysis. After importing the dataset,
  # you can display an overview of all variable labels by running the command
  # 'varlabel(data)'. However, this command does not work anymore after modifying
  # the data, e.g., by deleting or merging variables, since the variable labels
  # are attached to the data frame, and not the single variable.
  # For this line to work, you need library(Hmisc) loaded.
  # Afterwards, you are able to show the label using the command 'label(..)'
  for(i in seq_along(data)){
    label(data[,i]) = attr(data,"var.labels")[i]
  }

  return(data)
}
```

R 41: 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))
```

R 42: Working with Biography

```
# import the data file
Biography <- read.neps("Biography")

# 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
```

R 43: 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))
```

R 44: 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)))
```

R 45: 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]
  }
}
```

R 46: 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)))
```

R 47: 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)
```

R 48: 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)
```

R 49: 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)
```

R 50: 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)))
```

R 51: 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))
```

R 52: Working with pTargetMicrom

```
# open pTargetMicrom datafile. Note that this data file is only available OnSite!
Microm <- read.neps("pTargetMicrom")

# additionally to ID_t and wave, line identification in this file is done
# via variable regio, denoting the regional level of information
anyDuplicated(Microm[,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(Microm$wave, Microm$regio))

# only keep housing level
Microm <- subset(Microm, Microm$regio ==  1)
```

```
# now you can enhance CohortProfile with regional data
CohortProfile <- read.neps("CohortProfile")
Microm <- merge(CohortProfile, Microm, by = c("ID_t", "wave"), all = TRUE)
```

R 53: Working with pTargetRegioInfas

```
# open RegioInfas datafile. Note that this data file is only available OnSite!
RegioInfas <- read.neps("pTargetRegioInfas")

# 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.neps("CohortProfile")
RegioInfas <- merge(CohortProfile, RegioInfas, by = c("ID_t","wave"), all = TRUE)
```

R 54: 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
```

R 55: 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),]
```

R 56: 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'
```

R 57: 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))
```

R 58: 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'
```

R 59: 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
'------------------------------------------------------------'
```

R 60: 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
'------------------------------------------------------------'
```

R 61: 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))
```

R 62: 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))
```

R 63: 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))
```

R 64: 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)
```

R 65: 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)
```

R 66: 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))
```

R 67: 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
```

R 68: 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))
```

R 69: 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
```

R 70: 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))
```

R 71: 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))
```

R 72: 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
```

R 73: 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)))
```

R 74: Working with xPlausibleValues

```
# open datafile.
xPlausibleValues <- read.neps("xPlausibleValues")

# 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(xPlausibleValues[,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(xPlausibleValues$wave_w1)
```

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

R 75: 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)
```

R 76: Working with xTargetCORONA

```
# open the file
```

```
xTargetCORONA <- read.neps("xTargetCORONA")

# note that the wave is missing,
# as this reflects the pre-wave survey in may 2020
table(xTargetCORONA$wave)

# but rows can be uniquely identified by ID_t and wave
anyDuplicated(xTargetCORONA[,c("ID_t","wave")])
# returns "0" if there are no duplicates.
```

B.2 Release notes

Below you can find the *Release Notes* for the current Scientific Use File. They contain information on relevant data edition issues compared to the previous version of the Scientific Use File as well as information on data-related specifics and known problems at the time of the data publication. The *Release Notes* can also be downloaded from the documentation website of Starting Cohort 6 – as a text file with the complete history of data edition information on all Scientific Use File versions so far (see Section 1.2).

```
================================================
**
** NEPS STARTING COHORT 6 - RELEASE NOTES a.k.a CHANGE LOG
** changes and updates for release NEPS SC6 16.0.0
** (doi:10.5157/NEPS:SC6:15.0.0)
**
================================================


================================================
* Changes introduced to NEPS:SC6 by version 16.0.0 *
================================================

General:
        - minor changes to several value and variable labels due to a general
          modification of the gendering guidelines

        - new variables coded according to the ISCED2011 classification have been
          added to datasets containing educational qualifications

        - the "extended characteristics" with additional information on each variable,
          including the question text (in Stata accessible via the NEPStools command
          *infoquery var*, in SPSS displayed at the end of the line in the *Variable
          View* window), have been supplemented with a note for those variables for
          which different versions of a question formulation were used due to filter
          settings; for these variables, a complete list of all question formulations
          is available in the online *Variable Search* at the website and in the
          *Survey Instruments* as part of the data documentation

pTarget:
        - in wave 14, the response options for the variables of the IILS-II scale
          (t66207*) were reversed compared to previous implementations in waves 4
          and 7; accordingly, the generated scale indices for waves 4 and 7 were
          also coded in reverse (higher values = lower interest); in the current
          SUF version, this has been corrected in line with all other NEPS starting
          cohorts and waves (higher values = greater interest)

xTargetCompetencies:
        - in the two previous SUF versions (14.0.0, 15.0.0), the variable for the
          linked WLE for sciences (sca14_sc1u, *Scientific literacy: WLE
          (uncorrected)*) was incorrect or identical to the unlinked WLE variable;
          this error has been corrected with the current SUF version

        - in the previous SUF version (15.0.0), the names for the two variables for
          procedural metacognition regarding the science test (mpa14sc_s5, mpa14sc_sc6)
          were swapped; in the current SUF version, the variable names are correctly
          assigned (mpa14sc_sc5 = *difference score*, mpa14sc_sc6 = *proportion
          correct*)
```

- in the previous SUF version (15.0.0), the names for the two variables for procedural metacognition regarding the static items of the ICT test (mpa14ic_s5a, mpa14ic_sc6a) were swapped; in the current SUF version, the variable names are correctly assigned (mpa14ic_sc5a = *difference score for static items*, mpa14ic_sc6a = *proportion correct for static items*)
- the same applies to the variables for procedural metacognition regarding the interactive items of the ICT test (mpa14ic_s5b, mpa14ic_sc6b); in the current SUF version, the variable names are correctly assigned (mpa14ic_sc5b = *difference score for interactive items*, mpa14ic_sc6b = *proportion correct for interactive items*)