NEPS
National Educational Panel Study

Research Data

FDZ-LIfBi

# Data Manual

## NEPS Starting Cohort 1—Newborns

*Education from the Very Beginning*

Scientific Use File Version 5.0.0

LIfBi
LEIBNIZ INSTITUTE FOR
EDUCATIONAL TRAJECTORIES

Leibniz Institute for Educational Trajectories (LIfBi)
Wilhelmsplatz 3, 96047 Bamberg
Director: Prof. Dr. Sabine Weinert
Executive Director of Research: Dr. Jutta von Maurice
Executive Director of Administration: Dr. Robert Polgar
Bamberg; October 30, 2018

**Research Data Documentation**

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

Full citation of this document:
FDZ-LIfBi. (2018). *Data Manual NEPS Starting Cohort 1– Newborns, Education from the Very Beginning, Scientific Use File Version 5.0.0*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

This release of Scientific Use Data from Starting Cohort 1—Newborns "Education from the Very Beginning" was prepared by the staff of the Research Data Center at Leibniz Institute for Educational Trajectories (Forschungsdatenzentrum, FDZ-LIfBi). It represents a major collaborative effort. *The contribution of the following persons is gratefully acknowledged:*

Dietmar Angerer
Nadine Bachbauer
Daniel Bela
Gregor Czerner
Simon Dickopf
Daniel Fuß
Lydia Kleine
Tobias Koberg
Sven Pelz
Benno Schönberger
Mihaela Tudose
Katja Vogel

*For their support in writing this manual, special thanks go to:*
Manja Attig, Jeong Eun Kim (LIfBi Bamberg), Annabell Barthel (University of Leipzig)

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

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

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

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

# Contents

# 1 Introduction

## 1.1 About this manual

This manual is intended to facilitate your work with data of NEPS Starting Cohort 1—Newborns (NEPS SC1). It serves both as a first guide for getting started with the complex data and as a reference book. The primary emphasis is on practical aspects such as sample development, data structure, and variable merging. The manual is neither complete nor exhaustive, but several links to other resources are provided in the respective paragraphs.

The first chapter refers to further documentation material, requirements for data access, instructions for data citation, some general rules and recommendations, and selected user services. In the second chapter, the fundamental objectives of Starting Cohort 1 and its sampling strategy are briefly introduced. The main part of this chapter is devoted to the sample development across the waves including field times, realized case numbers, survey modes, and the measurement of competency domains. The principles of Scientific Use File data-editing processes as well as conventions for naming the data files and variables are explained in the third chapter, supplemented by missing value definitions and an overview of additionally generated variables. The fourth chapter focuses on the data structure with information about data types, identifiers, and short portraits of all available datasets in the Scientific Use File. These short portraits include recommendations on how to use the dataset as well as syntax examples for merging variables of this dataset with variables from other files. The last chapter addresses some specific issues that should be noted when working with data of Starting Cohort 1.

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

## 1.2 Further documentation

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

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

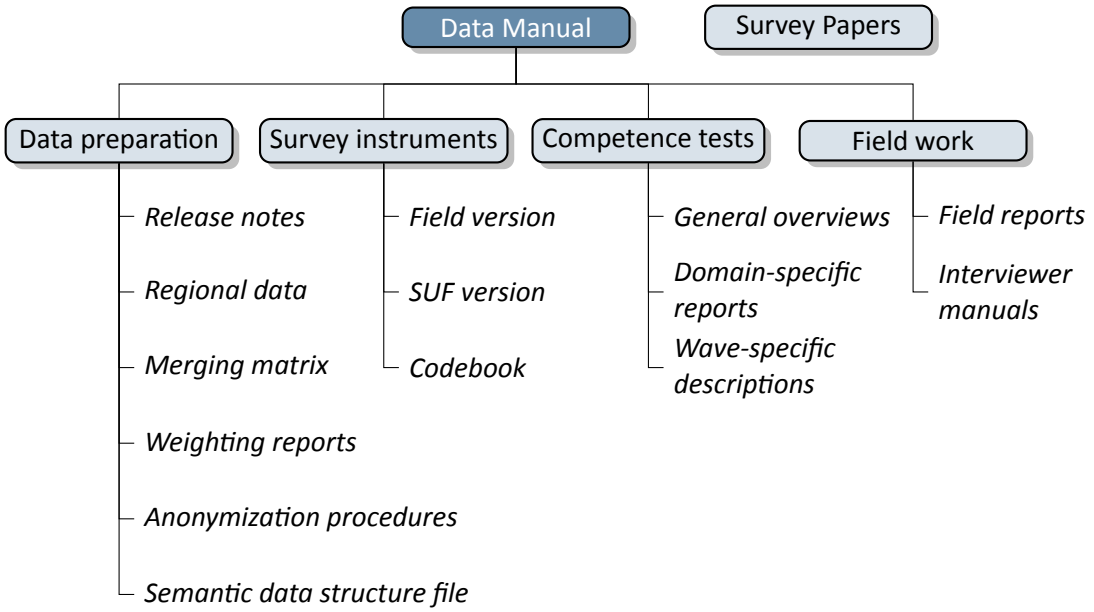



**Figure 1:** NEPS supplementary data documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 1.3 Data release strategy

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

### File Format

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

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

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

**Versioning and Digital Object Identifier**

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

Every release of a NEPS Scientific Use File is registered at da|ra and clearly labeled with a unique Digital Object Identifier (DOI, cf. Wenzig, 2012). This DOI has two main functions. On the one hand, it enables researchers to cite the utilized NEPS data in an easy and precise way (see section 1.5). This in turn is a basic precondition for any replication analysis. On the other hand, the DOI directs to a landing page with further information about the Scientific Use File and the data access options. The DOI of the current release is doi:10.5157/NEPS:SC1:5.0.0. Other releases of Scientific Use Files for Starting Cohort 1 can be accessed by substituting the version number at the end of the DOI and the URL respectively (see Table 1).

**Table 1:** Release history of SUF in Starting Cohort 1

| SUF Version | DOI | Date of release |
|---|---|---|
| **5.0.0** (current) | doi:10.5157/NEPS:SC1:5.0.0 | **2018-05-08** |
| 4.0.0 | doi:10.5157/NEPS:SC1:4.0.0 | 2017-08-10 |
| 3.0.0 | doi:10.5157/NEPS:SC1:3.0.0 | 2016-08-22 |
| 2.0.0 | doi:10.5157/NEPS:SC1:2.0.0 | 2015-11-24 |
| 1.0.0 | doi:10.5157/NEPS:SC1:1.0.0 | 2015-03-06 |

## 1.4 Data access

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

**Application for data access**

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

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

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

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

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

**Modes of data access**

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

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

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

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

**Sensitive information**

The download version of a Scientific Use File contains the least amount of information. For instance, institutional context data and the Federal State label (*Bundeslandkennung*, see section 1.7) are only available in the controlled environments of RemoteNEPS and our On-site data security rooms. Other indicators of a certain sensitivity are modified in the download data, such as aggregated categories for countries of citizenship or languages of origin. A few datasets and variables are exclusively accessible in the On-site version, e.g. the fine-grained regional indicators or open text entries. For a full picture of the availability of sensitive information, please refer to the overview on our website at:

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

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

## 1.5 Publications with NEPS data

Referencing the use of data from the National Educational Panel Study (NEPS) is essential for a good scientific practice as well as for revealing the scientific value of this study. The following citation rules apply to all publications based on NEPS data of Starting Cohort 1.

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

> This paper uses data from the National Educational Panel Study (NEPS): Starting Cohort 1—Newborns, doi:10.5157/NEPS:SC1:5.0.0. From 2008 to 2013, NEPS data was collected as part of the Framework Program for the Promotion of Empirical Educational Research funded by the German Federal Ministry of Education and Research (BMBF). As of 2014, NEPS is carried out by the Leibniz Institute for Educational Trajectories (LIfBi) at the University of Bamberg in cooperation with a nationwide network.

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

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

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

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

**Citing documentation**

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

> FDZ-LIfBi. (2018). *Data Manual NEPS Starting Cohort 1– Newborns, Education from the Very Beginning, Scientific Use File Version 5.0.0*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

Or another example:

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

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

> NEPS (Ed.). (2018). *Starting Cohort 1: Newborns (SC1), Wave 5, Questionnaires (SUF Version 5.0.0)*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

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

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

## 1.6 Rules and recommendations

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

**Rules**

- *Avoidance of re-identification:* Any action aimed at and suitable for re-identifying persons, households, or institutions is strictly forbidden. This also includes the combination of NEPS data with other data that allow for a re-identification of persons. In case of any accidental re-identification, the Research Data Center has to be informed immediately and all individual data gained therefrom have to be kept secret.

- *Avoidance of data disclosure:* NEPS data are exclusively provided on the basis of a valid Data Use Agreement—for a defined purpose (research project) and to a defined group of persons (data recipient and further project members that are involved in the contract). Any use for commercial or other economic purposes is not permitted just as any transfer of the data to third parties. Please handle the provided NEPS data with strict confidentiality!
- *Regulations on using the Federal State label:* For NEPS data collected in connection with schools or higher education institutions it is not allowed to use Federal-State-related information directly or indirectly contained in the data for analyses aiming at direct comparisons of the German Federal States (*Bundesländer*), or aiming at direct conclusions to be drawn about a Federal State, or aiming at a reconstruction of the concrete Federal State affiliation of persons, households, and institutions. Any kind of ranking between the Federal States based on NEPS data is prohibited (see section 1.7).

Please note that violation of these rules may lead to severe penalties as stated in the NEPS Data Use Agreement. If there is any doubt or question regarding the given regulations, please contact the Research Data Center (see section 1.9). The same applies in case of encountering any deficiencies in data quality or any security leaks with regard to NEPS data protection and data security.

**Recommendations**

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

- *As a matter of course:* Always be critical when working with empirical data! Although a big effort is being made to ensure the integrity of the provided data we cannot guarantee absolute correctness. Notices on problems or errors in the data are welcome at any time at the Research Data Center.
- *Enhanced understanding of the data:* Consult the documentation and survey instruments! The analysis of complex data necessitates a precise idea of how the information were collected and edited. All relevant material is available online (see section 1.2).
- *Facilitated handling of the data:* Utilize the tools that are offered! Several user services are provided to support NEPS data analyses—reaching from specific Stata commands (e. g., for an easy and adequate recoding of missing values) to a meta search engine (e. g., for an interactive exploration of all instruments) to a discussion forum (e. g., for the clarification of questions). These tools are also available online, see section 1.8 for more details.

## 1.7 On using the Federal State label *(Bundeslandkennung)*

In concurrence with the regulations of the Research Data Center at the Institute for Educational Quality Improvement (Institut zur Qualitätsentwicklung im Bildungswesen, IQB), using the Federal State label in conjunction with NEPS data collected in connection with schools or higher

education institutions is permitted in the context of exploring scientific research questions, if it is exclusively used for:

- control purposes in order to incorporate it as a covariate in the planned analysis; the identification of individual Federal States in the displayed results is not permitted
- incorporating contextual characteristics or other third-party variables; the identification of individual Federal States in the displayed results is not permitted
- comparing aggregated groups of Federal States where at least two states are combined to form a single meaningful group with regard to substantive issues; the identification of individual Federal States in the displayed results is not permitted
- for sample descriptions (e.g., the distribution of participants by state and by different types of schools within states)

When using data collected in connection with schools or higher education institutions, it is not allowed to use Federal-State-related information directly or indirectly contained in the data for analyses aiming at a direct Federal State comparison, direct conclusions to be drawn about a Federal State, or a reconstruction of the concrete Federal State affiliation of persons, households, and institutions.

The Federal State label in the starting cohorts of schools and higher education institutions is provided by LIfBi to the scientific community only via remote access (RemoteNEPS) and—depending on availability—via guest working stations in Bamberg (On-site). The respective analysis results are reviewed by LIfBi to ensure that this agreement has been observed before being passed on electronically to the researcher in a password-protected environment. The abovementioned restrictions concerning the use of the Federal State label do not apply to data collected in a nonschool context and/or in Federal-State-specific educational reform studies.

## 1.8 User services

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

→ www.neps-data.de > NEPS

### NEPSforum

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

→ www.neps-data.de > NEPSforum

### NEPSplorer

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

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

### NEPStools

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

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

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

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

**User trainings**

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

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

## 1.9 Contacting the Research Data Center

The Research Data Center at the Leibniz Institute for Educational Trajectories (Forschungsdatenzentrum, FDZ-LIfBi) accounts for large parts of the NEPS data preparation and documentation process, for the data dissemination, and for the user support including individual advice. We welcome your feedback at any time to further improve our products and services. This particularly applies to this manual as the guiding document to facilitate your work with NEPS data of Starting Cohort 1.

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

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

# 2 Sampling and Survey Overview

## 2.1 Education from the very beginning

The aim of this study is to generate a longitudinal cohort starting with infants in their first year of life. Therefore families are visited in their homes. Substantial, theory-driven surveys are conducted with the children (as target persons) and their parents as well as with external child care persons (institution manager of the day nursery or the kindergarten, educators and childminders; starting in wave 2). This database enables scientists to describe and analyze processes and courses of education as well as competence development.

The main research questions of this NEPS study include:

- How do children in early childhood develop early skills and abilities and in what ways are processes of development and education supported by settings of child care and education within and outside the family?
- How do intra-familial and extra-familial settings interact?
- From what age of the child do families make use of child care settings and education outside the family and to what extent does this depend on the development of the child and/or on the family background including the intra-familial learning environment, parental needs, and orientations?

## 2.2 Sampling strategy

The target population of Starting Cohort 1 is defined as all children born in Germany from February 2012 to July 2012 and their families. At the start of the panel survey, the target children had to be at least six months old, but not older than eight months, in order to ensure a valid measurement of infant development. This means that the time window for direct measurements with the newborns was fixed exactly according to the age of the child.

Access to this population was via a register-based sample of addresses available at the municipal level. Children living in an institution (e.g. children's home or parent-child home) and their legal guardians were not included in the survey. The random sample is based on a two-stage disproportional stratified sampling strategy with:

- municipalities as primary sampling units, proportionally stratified according to a classification of urbanization (BIK scale) and
- addresses of newborns as secondary sampling units, disproportionally stratified with more addresses in bigger municipalities.

The selection of 84 municipalities at the first stage was based on the distribution of births in the first half of 2009 according to the German Microcensus in three explicit strata (less than 50,000 inhabitants; 50,000 to 500,000 inhabitants; 500,000 and more inhabitants), whereby municipalities having less than ten births were excluded. At the second stage, addresses were then randomly selected from the municipalities' register data via systematic interval sampling, divided into two tranches (births from February to April; births from May to July[1]). In the end, a gross sample size of 8,483 addresses out of 90 sampling points in 84 municipalities turned out to be sufficient to achieve the planned sample size of approximately 3,000 newborns. With 3,481 participants in the first survey wave of Starting Cohort 1, the realized sample size has clearly exceeded this target, corresponding to a response rate of 41 percent.

In wave 2, parent interviews were conducted with all parents from wave 1 who gave their consent to be contacted again, but only a subsample of children was asked to take part in the direct measurements. A random sample of 34 municipalities has been drawn from the initial 84 municipalities for this purpose. In the third wave, all panel respondents—children and parents—were invited to be surveyed.

The sampling design and its consequences for the derivation of sampling weights are fully described in Würbach, Zinn, and Aßmann, 2016. Further remarks on the recruiting process are given in the CAPI field report of the first survey wave (in German only). Both documents are available on our website at:

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

## 2.3 Competence measures

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

Surveying early child characteristics and development is a particular challenge of NEPS Starting Cohort 1, taking into account the special situation of investigating infants and young children (no group testing, limited attentional skills, etc.). In the first three waves, so-called direct measures with the child were implemented. They involve measures of basic cognitive abilities as well as observational measures: habituation-dishabituation paradigm, parent-child interaction and sensorimotor development. All direct measures were administered in the households of the families, videotaped and coded afterwards.

1 Since the response rate in tranche 1 was unexpectedly high, those target persons born in July were not used, provided the exact month of birth was known. Only those born in May and June and children for whom no month of birth information was available were used in tranche 2.

Data from the direct measures and competence tests pass through an editing process before they get integrated into the Scientific Use File. This data preparation enables users to work with scored items and test scores such as the sum or mean of correct answers. Detailed descriptions on how these scores were estimated can be found in separate reports for the respective competence domains (see section 1.2).

The scores are compiled in two datasets named `xDirectMeasures` for the measurements of waves 1 to 3 and `xTargetCompetencies` for the measurements from wave 4 onwards. These datasets are structured in the so-called wide format, that is, all responses of a single respondent are represented in one row of the data matrix. As a consequence, variable names for competence scores follow a specific nomenclature. It not only allows for the identification of the respective domain, the target group, the testing modus, and the kind of scoring, but also informs about the repeated administration of a test item in a different wave or starting cohort (see section 3.2.2).

The next table shows the schedule of direct and competence measures in Starting Cohort 1 with domains by waves including test modus. The overview contains released data as well as data that is not yet published.

**Table 2:** Schedule of competence measures. OR = Observer Rating (based on videos), CBT = Computer-Based Test (proctored)

| | | 2012/13 **Wave 1** 6-8 months | 2013 **Wave 2**[1] 16-17 months | 2014 **Wave 3** 25-27 months | 2015 **Wave 4** 37-39 months | 2016 **Wave 5** 4 years |
|---|---|---|---|---|---|---|
| **Domain-Specific Competencies** | | | | | | |
| Vocabulary: Listening Comprehension at Word Level | vo | — | — | — | CBT | — |
| Mathematical Competence | ma | — | — | — | — | CBT |
| Scientific Competence | sc | — | — | — | — | — |
| **Stage-Specific Competencies** | | | | | | |
| Habituation-Dishabituation-Paradigm | hd | OR | OR | — | — | — |
| Interaction at Home: Parent-Child Interaction | ih | OR | OR | OR | — | — |
| Cognitive Development: Sensorimotor Development | cd | OR | — | — | — | — |
| Categorization: SON-R Subtest | ca | — | — | — | CBT | — |
| Delayed Gratification: Executive Control | de | — | — | — | CBT | — |
| Digit Span: Phonological Working Memory | ds | — | — | — | CBT | — |
| Flanker Task: Executive Control | ec | — | — | — | — | CBT |

[1] CAPI Subsample: Direct measures in wave 2 are available for a subsample of target persons only (simple random selection of 34 out of 84 initial municipalities)

## 2.4 Survey overview and sample development

This section informs about the progress of the Starting Cohort 1 sample. For each survey wave included in the current Scientific Use File there is a short characterization in terms of field time, number of realized cases, relevant subsamples and domains of competence testing (if appropriate), survey modus, and the institution(s) responsible for collecting the data. Figure 2 starts with an overview illustrating the field times and survey modes from wave 1 to 5.

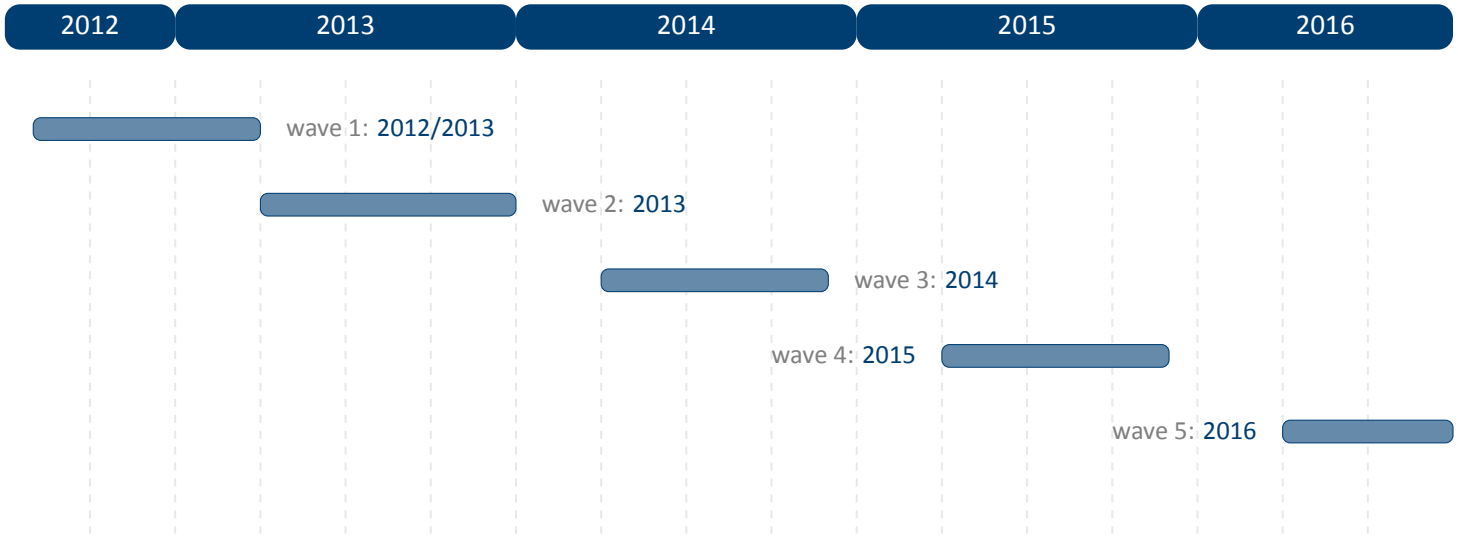



**Figure 2:** Survey progress of Starting Cohort 1 (waves 1 to 5)

### 2.4.1 Wave 1: 2012/2013


**Figure 3:** Field times and realized case numbers in wave 1

- Target persons

  **Current wave** 6-8 month-old infants

  **Initial sample** 6-8 month-old infants (panel entry 2012/2013)

  **Mode of survey** video-based survey of direct measures (parent-child interaction, sensori-motor development, and habituation-dishabituation paradigm)

- Context persons
  - Parents (esp. mothers)

    **Mode of survey** computer-assisted personal interviews (CAPI)

### 2.4.2 Wave 2: 2013

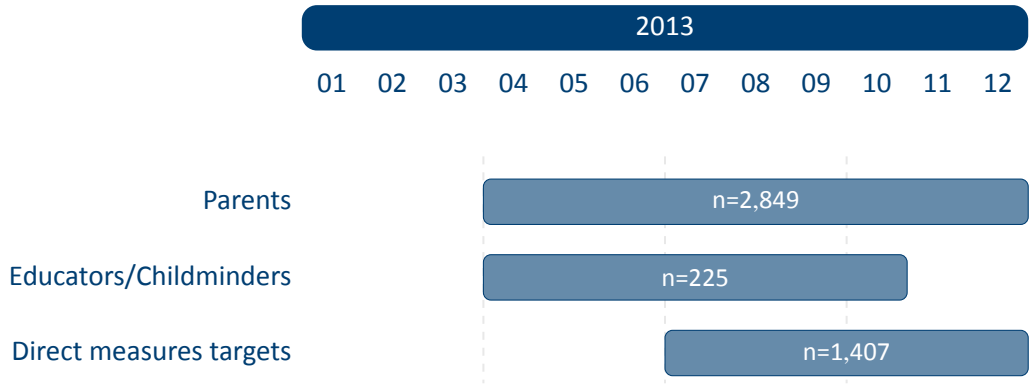



**Figure 4:** Field times and realized case numbers in wave 2

- Target persons

  **Current wave** Subsample of the initial sample; 16-17 month-old infants

  **Initial sample** 6-8 month-old infants (panel entry 2012/2013)

  **Mode of survey** video-based survey of direct measures (parent-child interaction and habituation-dishabituation paradigm)

- Context persons
  - Parents (esp. mothers)

    **Mode of survey** computer-assisted telephone interviews (CATI) for the parents; computer-assisted personal interviews (CAPI) for those parents who could not be reached via telephone and who belonged to the subsample of children with direct measures

  - External child care persons (educators/childminders)

    **Mode of survey** parents passed the written questionnaires (PAPI) to the external child care persons (=educators in kindergartens or day care childminders)

### 2.4.3 Wave 3: 2014

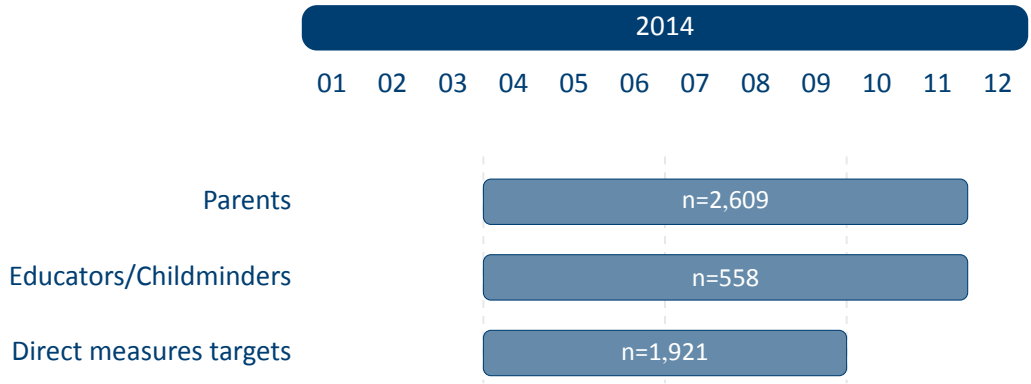



**Figure 5:** Field times and realized case numbers in wave 3

- Target persons

  **Current wave** 25-27 month-old infants

  **Initial sample** 6-8 month-old infants (panel entry 2012/2013)

  **Mode of survey** video-based survey of direct measures (parent-child interaction)

- Context persons
  - Parents (esp. mothers)

    **Mode of survey** computer-assisted personal interviews (CAPI); computer-assisted telephone interviews (CATI) for those parents who could not be reached at home; written questionnaire on the vocabulary of the child (PAPI)

  - External child care persons (educators/childminders)

    **Mode of survey** parents passed the written questionnaires (PAPI) to the external child care persons (=educators in kindergartens or day care childminders)

### 2.4.4 Wave 4: 2015

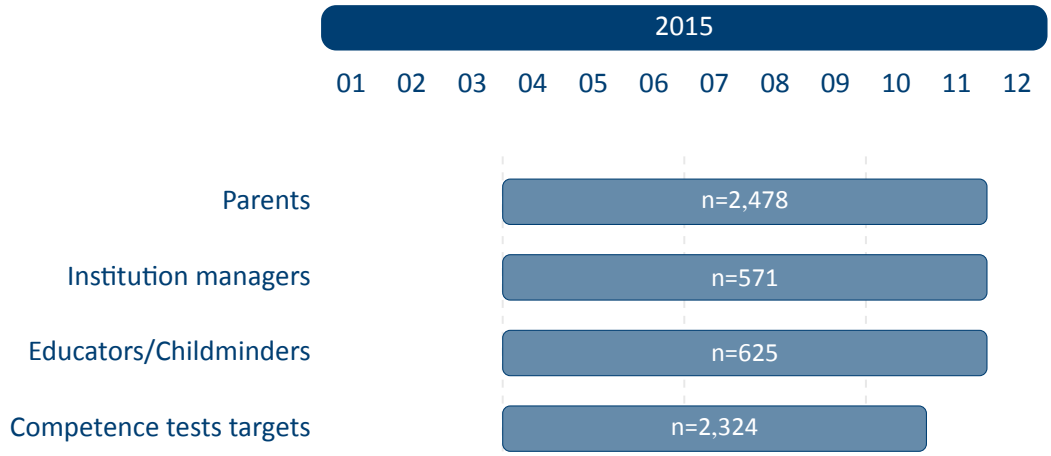



**Figure 6:** Field times and realized case numbers in wave 4

- Target persons

  **Current wave** 37-39 month-old infants

  **Initial sample** 6-8 month-old infants (panel entry 2012/2013)

  **Mode of survey** computer-based test (CBT/tablet) of direct measures (vocabulary, categorization, delayed gratification, digit span)

- Context persons
  - Parents (esp. mothers)

    **Mode of survey** computer-assisted personal interviews (CAPI); computer-assisted telephone interviews (CATI) for those parents who could not be reached at home

  - External child care persons (educators + institution managers)

    **Mode of survey** parents passed the written questionnaires (PAPI) to the external child care persons (=educators in kindergartens + institution managers of kindergartens)

### 2.4.5 Wave 5: 2016

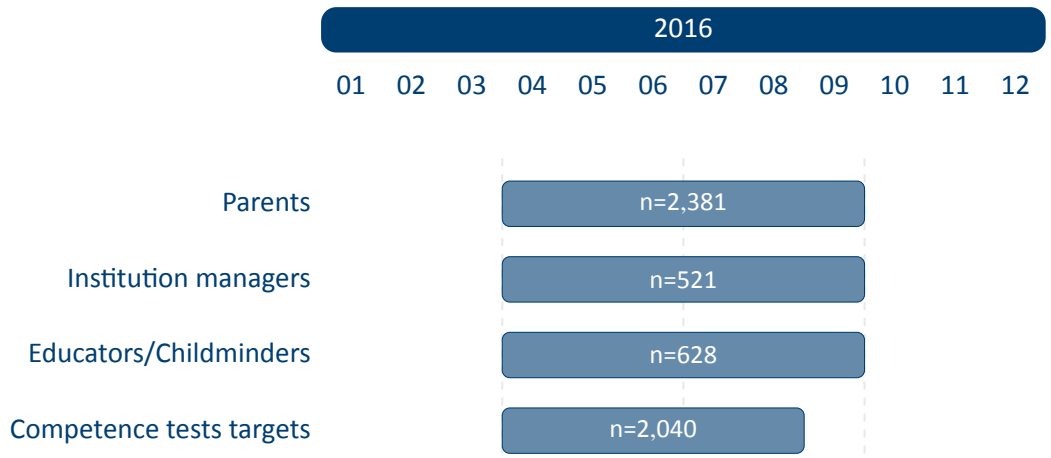



**Figure 7:** Field times and realized case numbers in wave 5

- Target persons

  **Current wave** approx. 48 month-old children

  **Initial sample** 6-8 month-old infants (panel entry 2012/2013)

  **Mode of survey** computer-based test (CBT/tablet) of direct measures (flanker task, mathematics)

- Context persons
  - Parents (esp. mothers)

    **Mode of survey** computer-assisted personal interviews (CAPI); computer-assisted telephone interviews (CATI) for those parents who could not be reached at home; written questionnaire on the child (PAPI)

  - External child care persons (educators + institution managers)

    **Mode of survey** parents passed the written questionnaires (PAPI) to the external child care persons (=educators in kindergartens + institution managers of kindergartens)

# 3 General Conventions

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

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

The second paradigm is to integrate the data as much as possible without compromising the usability of the Scientific Use File. The underlying assumption is that for a vast majority of data users it is far more comfortable to reduce already integrated data for a specific analysis as opposed to correctly compile the relevant information from scattered source data themselves. In the end, each Scientific Use File contains only a few dozen integrated panel and spell datasets according to a general structure (see section 4.1.1 and section 4.1.2 for details), even if the compilation is based on several hundred separate source dataset files.

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

## 3.1 File names

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

**Table 3:** Naming conventions for NEPS file names

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

For instance, the file SC1_CohortProfile_D_5.0.0.dta refers to the *CohortProfile* data of *Starting Cohort 1* in its *Download* version of the Scientific Use File release *5.0.0*.

## 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 8. It has to be noted that a separate nomenclature is used for variables from competence measurements. Section 3.2.1 offers a detailed description of the general naming conventions for NEPS variables; the logic of naming competence variables is introduced in section 3.2.2.

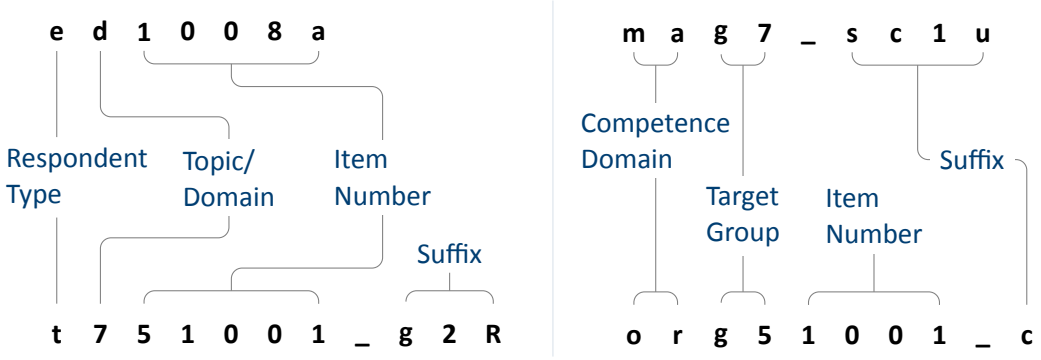



**Figure 8:** General variable naming (left) and competence variable naming (right)

### 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**<br>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)<br>`t` = Target person<br>`p` = Parent of target person<br>`e` = Educator/childminder<br>`h` = Head/manager of institution (information about school/kindergarten) |

(...)

**Table 4:** (continued)

| Digit | Description |
|---|---|
| 2 | **Topic/domain**<br>Indicator to which theoretical dimension or educational stage the variable refers<br>1 = Competence development<br>2 = Learning environments<br>3 = Educational decisions<br>4 = Migration background<br>5 = Returns to education<br>6 = Interest, self-concept and motivation<br>7 = Socio-demographic information<br>a = Newborns and early childhood education<br>b = From kindergarten to elementary school<br>c = From elementary school to lower secondary school<br>d = From lower to upper secondary school<br>e = From upper secondary school to higher ed./occ. training/labor market<br>f = From vocational training to the labor market<br>g = From higher education to the labor market<br>h = Adult education and lifelong learning<br>s = Basic program<br>x = Generated variables |
| 3–7 | **Item number**<br>Indicator for the item number which typically consists of four numeric characters plus one alphanumeric character |
| 8–11 | **Suffixes** (optional)<br>Indicator for three types of variables; separated from the previous characters by an underscore<br>*Suffixes for generated variables*: The `_g#` suffix indicates a generated variable; the running number after `_g` is in most cases a simple enumerator (e. g., `_g1`). Since scale indices are generated by a set of other variables, they are also identified by a `_g#` suffix. Note that scale indices are named after the first of the set of variables from which they were generated. In this case, numbering is only relevant if the first variable is identical for several scale indices. |

(…)

**Table 4:** (continued)

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

### 3.2.2 Conventions for competence variable naming

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

The additional suffixes inform about the mode of test execution if more than one survey modus has been applied for a measurement and about the sort of item score and overall competence score. There is a distinction between scored items named `[varname]_c` and scored partial

credit-items named `[varname]s_c`. The latter is relevant if more than one correct solution is possible (e. g., value 0 = `0 out of two points`, value 1 = `1 out of two points`, value 2 = `2 out of two points`), whereas the former is applied for dichotomous solutions (value 0 = `not solved`, value 1 = `solved`). In addition to the item scores, several aggregated scores are provided for competence measurements. They are indicated by `_sc[number]` and a few special suffixes for Starting Cohort 1. A letter appended to the suffix indicates that more than one aggregated score for a competence measurement is available (e. g., `_sc3a`, `_sc3b` for different sum scores of any test). Detailed descriptions on how the aggregated competence scores were estimated can be found in the domain-specific documentation reports. The last part of Table 5 shows all possible suffixes and their meanings.

**Table 5:** Conventions for competence variable names

**Part I: Competence Domain** (2 chars)

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

(...)

**Table 5:** (continued)

| | |
|---|---|
| st | Scientific thinking: Science propaedeutics |
| vo | Vocabulary: Listening comprehension at word level |

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

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

**Part III: Item number** (3-4 chars)

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

**Part IV: Suffixes** (starting with an underscore)

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

**Identification of repeated test items**

In some competence measurements identical items are implemented in different testing waves (e. g., mathematics). Identifying repeatedly measured test items in NEPS data can be easily

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

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

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 cohort or testing wave of current item administration.

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

### 3.2.3 Labels

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

In addition to the variable and value labels, the datasets also contain extended characteristics for variables. These include the question text from the survey instrument, any associated interviewer instructions and filter conditions, as well as other meta information. All extended features can be accessed directly within data files. Stata users apply the `infoquery` command for this, which is part of the *NEPStools* package (see section 1.8). SPSS users will find the additional meta information in the “Variable View” at the end of each variable line.

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

A concrete example is the adaptation of interviewer instructions or question texts from the informal salutation ("Du") to the formal salutation ("Sie"). Since these changes are not expected to have any effect on how a question is answered, the corresponding values across multiple waves get integrated into one variable. If you request the meta information of such a variable in the dataset, the wording of the latest item formulation will be displayed (in the given example with the formal salutation "Sie"). In case of uncertainties regarding the continuity of meta information of a variable across different waves, we recommend to consult the respective *survey instruments* for the individual waves.

## 3.3 Missing values

The NEPS data contain various missing codes to differentiate between various types of missing values. All missing codes have negative values or are defined as system missing. Depending on the statistics program used, you must ensure that these codes are processed correctly. In the offered SPSS datasets, the missing codes are already defined as missing values. When using Stata, the missing codes must first be excluded from the analyses by the user as missing values. For this purpose the command `nepsmiss` is available in the *NEPStools* package (see section 1.8). The general recommendation is to always carefully check the frequency distributions of the relevant variables before running an analysis.

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

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

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

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

- The code *missing by design* (–54) is assigned when respondents in a (sub)sample have not been asked the respective questions. This is usually the case if the administered survey instrument contains (sub)sample-specific questionnaire modules. The code is also used for the

more general case where values of a variable are not available due to the design of the survey (e. g., measurement rotation with either easier or heavier test tasks).

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

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

- If in the data edition process certain values which are not considered to be meaningful are 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 that target persons without survey data for a certain wave (e. g., due to illness) are usually not included in the corresponding datasets. This missing code is only used in the special cases of datasets that integrate several waves in wide format (e. g., `xTargetCompetencies`) or that also contain observations for non-participating persons in a wave (e. g., `CohortProfile`).

## 3.4 Generated variables

### Coding and recoding of open responses

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

**Table 6:** Overview of missing codes

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

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

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

The most common and comprehensive coding scenarios in the fields of occupation, education, branches, courses, and regional information are processed by the Research Data Center (FDZ-

LIfBi) itself. Other coding tasks are distributed among the responsible departments at the LIfBi in Bamberg and the partners in the NEPS consortium.

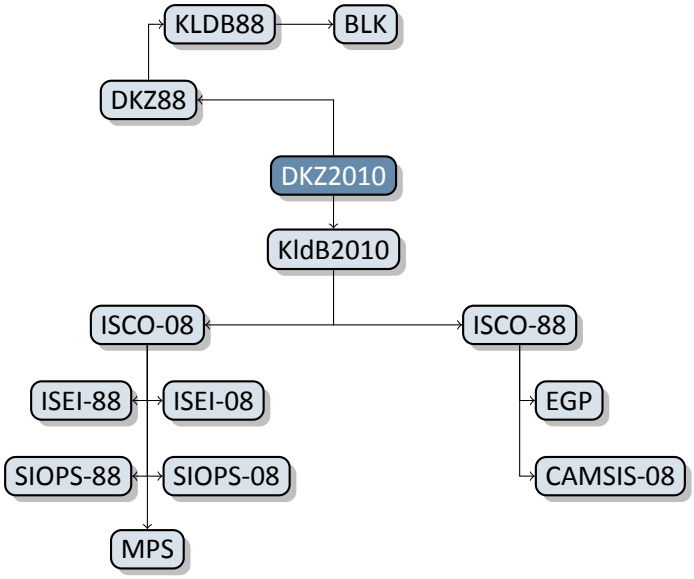



**Figure 9:** Derivation paths for several occupational scales and schemes provided in the NEPS

**Derived scales and classifications**

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

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

Figure 9 shows the derivation paths for several occupational scales and schemes provided in the NEPS. A detailed description of the standard derivations for educational attainment (ISCED, CASMIN and Years of Education) can be found in the corresponding documetation report by Zielonka and Pelz, 2015.

# 4 Data Structure

## 4.1 Overview

The aims and scope of the NEPS surveys inevitably create complex data. The idea was to organize these data in a well structured, traceable, and user-friendly way while preserving a high level of detail in the data. Occasionally, additional variables and datasets from one or more of the original files were generated to ease preparation and analysis of the data.

Usually, all information collected during a panel wave is appended to the corresponding data file from previous waves. Data files containing longitudinal information from multiple waves are denoted with a *p* in the filename. For instance, the file `pTargetCATI` records data from target's CATI questionnaire, while one row corresponds to one target at one wave. This convention does not fully apply to all panel moments. For example, competence testing has been conducted repeatedly. But because the content of competence tests differs to a large extent, their data structure is best represented in a wide format (see section 4.2.18 for a more detailed description). Such data files are denoted with an *x*, which shall indicate the cross-sectional design (one row represents all waves of one respondent).

For episode data, usually collected retrospectively using iterative sets of questions, we provided so called spell files that are prefixed by `sp`. An example is the file `spVocTrain` that contains a student's history of vocational training.

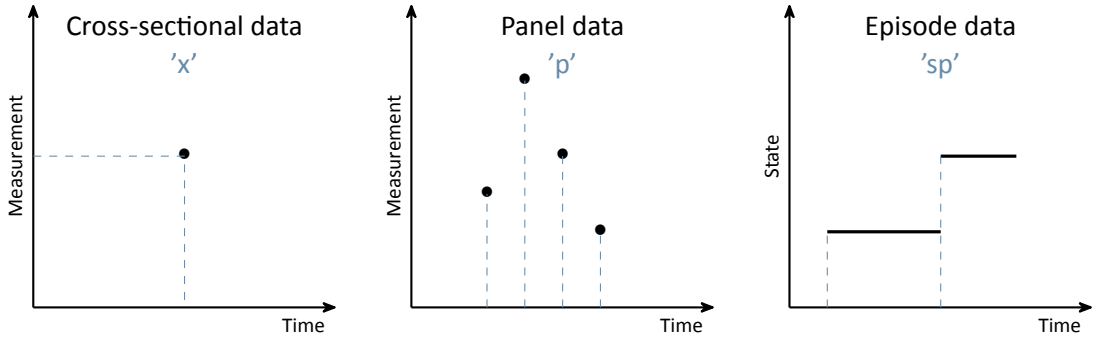



**Figure 10:** Different types of data structure

Besides questionnaire and test data provided by respondents, there is also paradata or derived information provided in the Scientific Use File. You may identify those by the leading uppercase letter (e. g. `Basics`).

Note that NEPS has a multi-level and multi-informant design; therefore there are identifiers of several units to be considered. In this Starting Cohort, this is:

**ID_t** Identifies a target person. `ID_t` is unique over waves and samples (and also Starting Cohorts).

**wave** Indicates the sample wave.

There are additional identifier variables for marking a target's membership in a test group (`ID_tg` in `CohortProfile`, not applicable to all starting cohorts) and for marking an interviewer (`ID_int` in `Methods` datasets). However, these IDs are not relevant for data merging and negligible for most empirical applications.

### 4.1.1 Panel data

As stated above, all data from subsequent waves are appended to the already existing data file (as far as possible). We call this method of data handling *integrated panel data*, in contrast to the method of releasing a single file for every wave (where every file only contains data from this unique wave). If you are working with integrated panel data files for the first time, you may find it helpful to pay attention to the following things:

One row contains data from one wave of one respondent. This means that

- you need more than one variable to identify a single row for selecting and merging. This is usually `ID_t` and `wave`.
- not all variables have been administered in every wave, but out of this integrated structure, all variables are *present* for every wave (and contain a missing code if no data is available).
- data from one individual is surveyed in multiple waves, and therefore is spread across multiple lines in the data file.

If your interest is primarily in analyzing panel items that have been surveyed in multiple waves, this is your preferred data structure. However, in many cases, you might need (e. g., time-invariant) cross-sectional information. Then those issues are crucial for your analysis. Usually, the combined set of variables of your interest will not be surveyed in a single wave. Thus, they can not be analyzed (e. g., cross-tabulated) together straightaway as they are stored in *different rows* of the data file! Cross-tabulating those variables in their current state will result in an L-shaped table, where all observations from one variable fall into the missing category of the other variable and vice versa. How to deal with this issue highly depends on your analysis and the applied methods, but here are some examples:

- you might split the data file into wave specific subfiles (each containing data from one wave). Then, merge them again, but only use the respondents identifier (`ID_t`), neglecting the wave variable (you might have to rename variables and make them wave specific). The result will be a cross-sectional file where every line is one respondent.
  Stata's *reshape* command (and similar tools in other software) basically does the same.

- you could stay with the panel structure and just copy values from observed cells to unobserved cells. For example, if place of birth has been surveyed only in wave one, you could copy this value to the cells of wave two, three, etc. This is especially useful for time-invariant variables such as gender, birth year, etc., which have been surveyed only once but are valid in every wave.

### 4.1.2 Episode or spell data

Most data users will know how to handle cross-sectional data. Many will also have an idea how to work with and analyze panel data. It is episode data which stresses your understanding of data edition. Hence, we spend some additional time on clarifying this data.

In episode (or spell) data, you find one row for every episode which has been recorded. At first, think of this as independent of respondents or survey waves. One row contains one episode. Usually, a start date and an end date describes this episode's duration. The rest of the variables in such a data file contain information about this time span. Note that this information corresponds to this episode chronologically! Especially for time variant variables (e. g., ISEI, CASMIN), this does not describe the status of the respondent but the status of the respondent *at that time*. Do not get confused about this issue.

To make an example, in the spell module `spEmp`, you might find as an episode a certain period of time where someone worked in a single job without any interruption. If this person changes to a new job, a new episode (i. e., a new row of data) is recorded. In fact, every other change in this setting also results in a new episode, e. g., the job is interrupted by parental leave, the respondent retires, or even if she/he starts an additional side job. So think of an episode as the smallest possible unit in one's life history.

Besides this kind of (time) episode data, which we call *duration spells*, there are also two other types of episode data: *event spells* that register occurring events or the transition from one state to another (e. g., change of marital state, change of educational status) and *entity spells* that contain one row for every entity that has been reported (e. g., children, partner).

To identify a single row in the data file, you usually need two variables: the respondents `ID_t`, and the episode-, event-, or entity-numerator (e. g., variable `spell` identifies one duration spell). See the data file pages in section 4.2 for the exact variables needed.

There is one extra circumstance you have to be aware of before working with our spell data. This is *subspells*. The data are collected retrospectively, i. e., during an interview, respondents are surveyed about all episodes which have occurred in the past since the last interview (in the first interview it is since birth). If an episode has been completed at the time of the interview, the respondent reports start and end dates and the episode is complete. Difficulties arise if the episode is not complete at the time of the interview. Then, the episode is right-censored but may be ongoing. In the next interview, this episode is then set up so respondents can report if it has ended in the meantime or if it is still ongoing. Technically, this results in multiple rows in the data file, which you can distinguish by variable `subspell`:

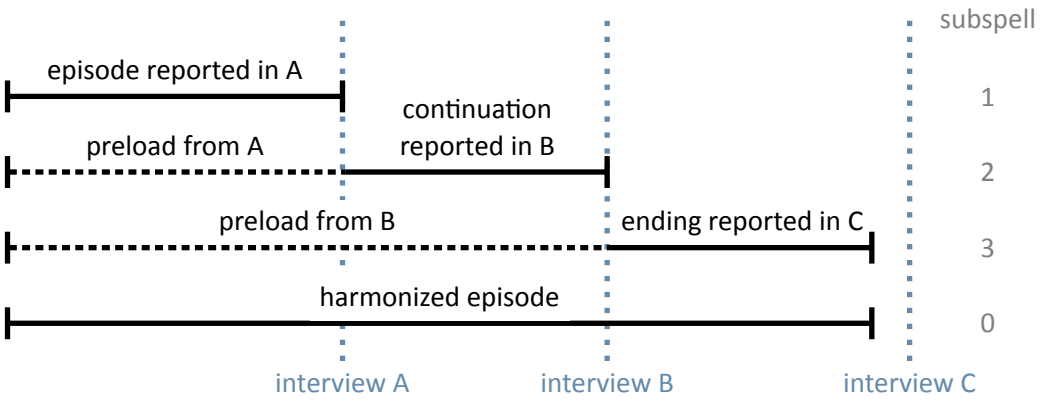



**Figure 11:** Logic of subspells

- original (right-censored) episode reported in first wave (`subspell=1`)
- continued episode reported in next wave (`subspell=2`)

Usually, you want the last subspell as it is the most recent information about this episode. To ease your work with the data, we already identified the latest subspell for you, and provide a harmonized episode with `subspell = 0`. Also, all episodes that have been reported completed in the first place do not have any subspells and are therefore initially marked with `subspell = 0` [2].

We generally recommend executing

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

at the start of your data preparation unless you are specifically interested in subspell information. However, be aware that data of harmonized spells may come from different waves because these spells always include the latest valid information available. There is another caveat: Do not use this selection if you work with information stored in wide format (like interruption episodes of vocational training spells stored in a wide format in spVocTrain).

### 4.1.3 Revoked episodes

In order to reduce seem bias, spell data are preloaded by prior wave information. This information from prior waves can be revoked by the respondent in the current wave. Spell data therefore contain information on revocation (see, for example, variable `disagint`). Reasons for revocation/ contradiction are manifold, they depend on the information given to the respondent to recall the episode (see questionnaire for the exact wording of episode data collection).

2 by variable `spgen`, you can detect if it is an episode originally reported complete (`spgen=0`) or a harmonized (generated) episode (`spgen=1`)

Whenever an episode is revoked by the respondent, the episode is marked as revoked/contradicted. The corresponding information is gathered anew and stored as a new episode in the current data collection wave. It is not actively marked as a corrected spell. Identification of homologue spells (previously given information and its correction in the subsequent wave) is up to the user. Please note: As it is technically impossible during data collection to indicate a start date prior to the last interview date, virtually all corrected spell episodes are left censored (exception: episodes that started on the last waves interview date).

## 4.2 Data files

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

To ease your understanding of the relationship of those files, Figure 12 provides an overview. The edges in this graph symbolize how a data file may be linked to other files. This is not meant to document every possible data link you could do but rather tries to give you an idea which data files relate most. By clicking on a node, you get directed to this data file's explanatory page.

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

```
** version of this Scientific Use File
global version 5-0-0
** path where the data can be found on your local machine
global datapath Z:/Data/SC1/5-0-0
```

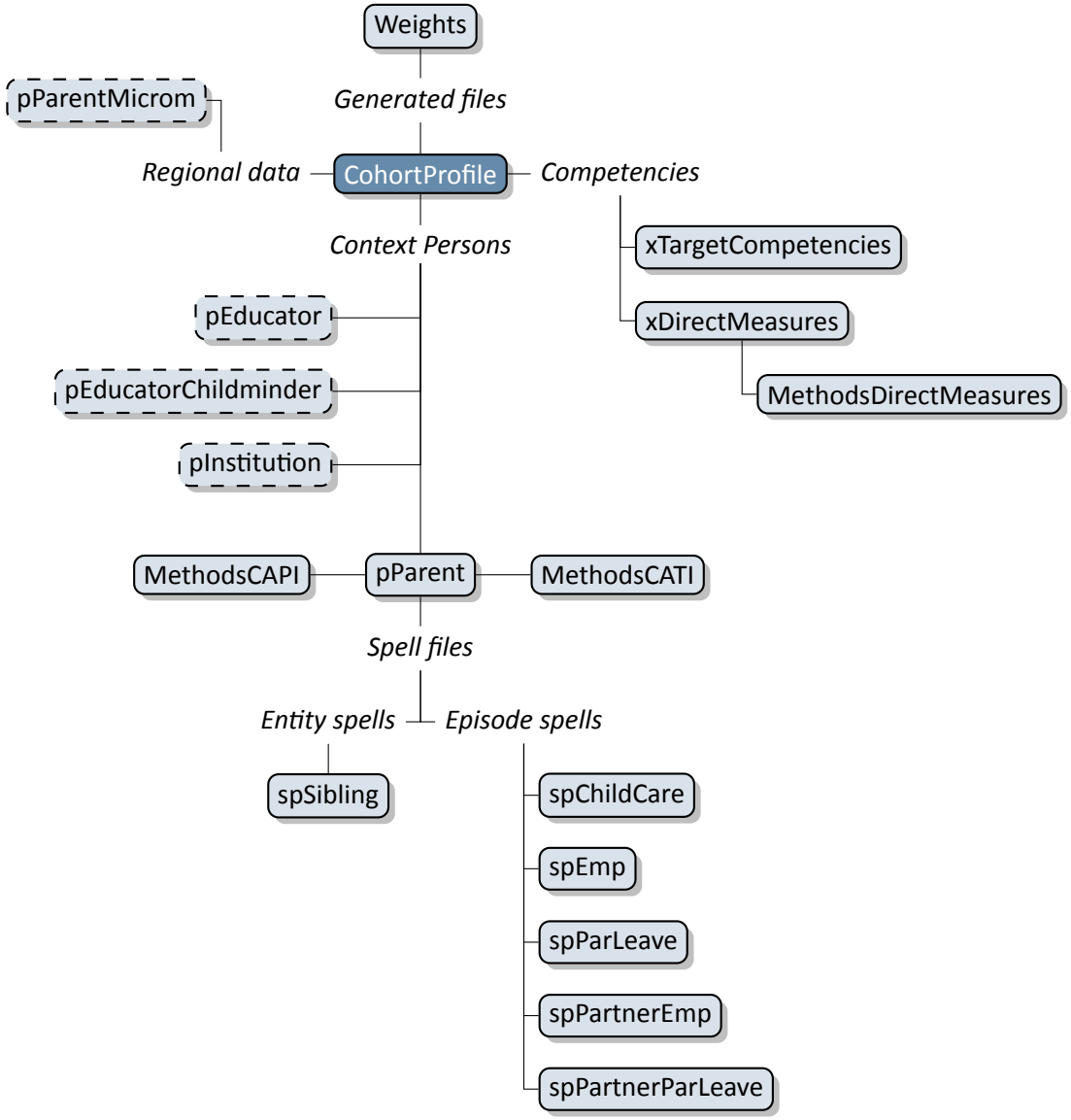
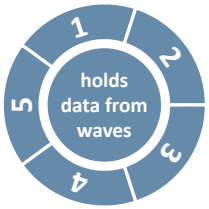



**Figure 12:** Graphical overview of all data files. Each node represents one data file. Relations are indicated by connection lines. Files with a dashed border are not available in the Download version of the Scientific Use File. Click on a data file to get more information.

### 4.2.1 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

No. of variables 15

Exemplary data snapshot

| ID_t | wave | tx80220 | tx80521 | tx80522 | tx80524 | inty | testy |
|---|---|---|---|---|---|---|---|
| 8054987 | 1 | Participation | Yes | -54 | 0 | 2012 | 2012 |
| 8054987 | 2 | Participation | Yes | -54 | 0 | 2013 | -54 |
| 8054987 | 3 | Participation | Yes | -20 | 0 | 2014 | 2014 |
| 8054987 | 4 | Temporary drop-out | No | 0 | 0 | -56 | -55 |
| 8054987 | 5 | Temporary drop-out | No | 0 | 0 | -56 | -55 |
| 8054996 | 1 | Participation | Yes | -54 | 0 | 2012 | 2012 |
| 8054996 | 2 | Participation | Yes | -54 | 0 | 2013 | -54 |
| 8054996 | 3 | Participation | Yes | -20 | 0 | 2014 | 2014 |
| 8054996 | 4 | Participation | Yes | 1 | 1 | 2015 | 2015 |
| 8054996 | 5 | Participation | Yes | 1 | 0 | 2016 | 2016 |

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

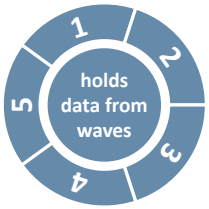
**It is strongly recommended to use this data file as a starting point for any analysis!**

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

```
** open the data file
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear

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

** how many different respondents are there?
distinct ID_t

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

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

### 4.2.2 MethodsCAPI

« go back to overview




Description

Paradata from the 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

No. of variables 95

Exemplary data snapshot

| ID_t | wave | ID_int | px80302 | px80301 | intm | inty | px80209 |
|---|---|---|---|---|---|---|---|
| 8055014 | 1 | 1576 | 30-49 years | 2 | 11 | 2012 | 90.61667 |
| 8055014 | 2 | 1576 | 30-49 years | 2 | 8 | 2013 | 56.75000 |
| 8055014 | 3 | 2324 | 50-65 years | 2 | 5 | 2014 | 98.56667 |
| 8055016 | 1 | 2065 | 50-65 years | 2 | 1 | 2013 | 77.63333 |
| 8055016 | 3 | 2326 | Up to 29 years | 2 | 5 | 2014 | 72.05000 |
| 8055022 | 1 | 1576 | 30-49 years | 2 | 11 | 2012 | 110.58334 |
| 8055022 | 2 | 1576 | 30-49 years | 2 | 8 | 2013 | 99.95000 |
| 8055022 | 3 | 2324 | 50-65 years | 2 | 5 | 2014 | 101.61667 |

This dataset provides a variety of information about data collection during the CAPI interview such as gender (px80301) and age (px80302) of the interviewer, the interview date (intm, inty) the interview duration (px80209), the use of incentives (px80210), and the individual survey participation status (px80220).

It should be noted that MethodsCAPI contains all respondents contacted, regardless of whether an interview was conducted or not (see variable px80207 for more details). For this reason, MethodsCAPI consists of more cases than the data file pParent.

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

```
** open the data file
use ${datapath}/SC1_MethodsCAPI_D_${version}.dta, clear

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

** check out participation status by wave
tab wave px80220

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

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

### 4.2.3 MethodsCATI

« go back to overview


Description

Paradata from the CATI 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

Exemplary data snapshot

No. of variables 30

| ID_t | wave | ID_int | px80302 | px80301 | intm | inty | px80209 |
|---|---|---|---|---|---|---|---|
| 8054982 | 2 | 2013 | Up to 29 years | 2 | 4 | 2013 | 29.93333 |
| 8054987 | 2 | 1117 | 30-49 years | 2 | 4 | 2013 | 30.01667 |
| 8054996 | 2 | 2000 | 30-49 years | 1 | 4 | 2013 | 23.98333 |
| 8055000 | 2 | 1003 | 50-65 years | 1 | 7 | 2013 | 27.31667 |
| 8055008 | 2 | 2037 | 30-49 years | 1 | 4 | 2013 | 33.63334 |
| 8055012 | 2 | 1232 | Older than 65 years | 1 | 4 | 2013 | 42.16667 |
| 8055014 | 2 | 1015 | 50-65 years | 2 | 4 | 2013 | 33.75000 |
| 8055016 | 2 | 2032 | Up to 29 years | 2 | 4 | 2013 | 28.58333 |

This dataset provides a variety of information about data collection during the CATI interview such as gender (px80301) and age (px80302) of the interviewer, the interview date (intm, inty), the interview duration (px80209), the use of incentives (px80210), and the individual survey participation status (px80220).

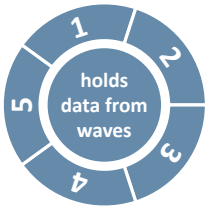
It should be noted that MethodsCATI contains all respondents contacted, regardless of whether an interview was conducted or not (see variable px80207 for more details). For this reason, MethodsCATI consists of more cases than the data file pParent.

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

```
** open the data file
use ${datapath}/SC1_MethodsCATI_D_${version}.dta, clear

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

** check out participation status by wave
tab wave px80220

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

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

### 4.2.4 MethodsDirectMeasures

« go back to overview




Description

Paradata of the realization of the direct measures

File structure

long format: 1 row = 1 target in 1 wave

Exemplary data snapshot

ID variables needed to identify a single row

ID_t wave

No. of variables 104

| ID_t | wave | px04021 | px04025 | px04011 |
|---|---|---|---|---|
| 8055356 | 1 | . | . | 0 |
| 8055356 | 2 | -54 | -54 | -54 |
| 8055356 | 3 | 0 | 0 | 0 |
| 8055356 | 5 | -54 | -54 | . |
| 8055867 | 1 | . | . | 0 |
| 8055867 | 2 | -54 | -54 | -54 |
| 8055867 | 3 | 0 | 0 | 0 |
| 8055867 | 4 | -54 | -54 | . |
| 8055867 | 5 | -54 | -54 | . |

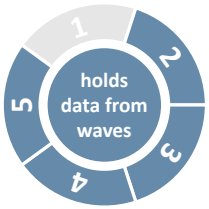
`MethodsDirectMeasures` contains various information on the data collection process for direct measures. These include variables on disturbances, performance issues, or implementation problems (e. g., `px04021`, `px04025`), but also on causes for missing consent (e. g., `px04011`).

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

```
** open the data file
use ${datapath}/SC1_MethodsDirectMeasures_D_${version}.dta, clear

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

** check out the different outcomes of  parent-child interaction.
** as you can see, 3 means test has been completed
tab px02002

** also, note that not all interactions have been measured
** between respondent (usually mother) and child. Some
** have been conducted together with the respondent's partner
tab px02003_v1
```

### 4.2.5 pEducator

« go back to overview




Description

Context data collected from child care persons in day-care institutions

File structure

long format: 1 row = 1 target in 1 wave

Exemplary data snapshot

ID variables needed to identify a single row

ID_t wave

No. of variables 391

| ID_t | wave | e40000a_g1 | e217515 | e209107 |
|---|---|---|---|---|
| 8056571 | 2 | Italy | -90 | Yes |
| 8056571 | 3 | Italy | 8 | Yes |
| 8056571 | 4 | . | 4 | Missing by design |
| 8062600 | 2 | Turkey | 3 | No |
| 8062600 | 3 | . | -90 | No |
| 8062600 | 5 | . | 4 | Missing by design |

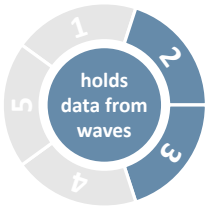
The responsible child care persons of target children attending day-care institutions (*Gruppenbetreuung Kindergarten*) were surveyed via PAPI questionnaires. This data is made available in the file `pEducator`. The dataset includes personal characteristics of the child care persons such as their country of origin (`e40000a_g1`) as well as information on the composition of the child group such as the number of children born in 2011 (`e217515`), but also data on the child care institution itself such as when it is organized on the initiative of parents (`e209107`).

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

```
** open the CohortProfile
use ${datapath}/SC1_CohortProfile_R_${version}.dta, clear

** merge sex and age of educator to CohortProfile.
** note that this datafile is directly linkable to
** the child (if you have been working with other SCs,
** you may have expected a variable ID_e)
merge 1:1 ID_t wave using ${datapath}/SC1_pEducator_R_${version}.dta, ///
  keepusing(e761110 e76112y) nogen assert(master match)

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

** now, compute the age of the educator at the date of the interview
nepsmiss inty  e76112y
generate ed_age = inty - e76112y

summarize ed_age
```

### 4.2.6 pEducatorChildminder

« go back to overview




Description

Context data collected from childminders

File structure

long format: 1 row = 1 target in 1 wave

Exemplary data snapshot

ID variables needed to identify a single row

ID_t wave

No. of variables 235

| ID_t | wave | e40000a_g1 | ea25020 | e208128 |
|---|---|---|---|---|
| 8063499 | 2 | . | 2 | 1 |
| 8063567 | 2 | . | 2 | 2 |
| 8065158 | 2 | Poland | 2 | 1 |
| 8067744 | 2 | Ukraine | 3 | 1 |

For children who do not attend a day care institution but are cared by a childminder (*Tagespflegepersonen*), a PAPI questionnaire corresponding to that used in `pEducator` was handed out to the childminders. The variables in the datafile `pEducatorChildminder` also provide information on personal characteristics of the child care person such as the country of origin (`e40000a_g1`) or number of own children (`ea25020`), but also on the group composition such as the number of children born in 2011 (`e208128`).

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

```
** open the CohortProfile
use ${datapath}/SC1_CohortProfile_R_${version}.dta, clear

** merge sex and age of childminder to CohortProfile.
** note that this datafile is directly linkable to
** the child (if you have been working with other SCs,
** you may have expected a variable ID_e)
merge 1:1 ID_t wave using ${datapath}/SC1_pEducatorChildminder_R_${version}.dta, ///
  keepusing(e767110 e76712y) nogen assert(master match)

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

** now, compute the age of the childminder at the date of the interview
nepsmiss inty  e76712y
generate cm_age = inty - e76712y

summarize cm_age
```

### 4.2.7 pInstitution

« go back to overview


Description

Context data collected from the institution head/manager

File structure

long format: 1 row = 1 target in 1 wave

Exemplary data snapshot

ID variables needed to identify a single row

ID_t wave

No. of variables 133

| ID_t | wave | h217001 | h217002 | h534010 | h219301 |
|---|---|---|---|---|---|
| 8055983 | 5 | 21 | 30 | 19 | 2 |
| 8056004 | 4 | 43 | 44 | 10 | 6 |
| 8056077 | 4 | 50 | 60 | 10 | 7 |
| 8056078 | 5 | 64 | 57 | 5 | 9 |
| 8056088 | 4 | 42 | 31 | 6 | 10 |
| 8056088 | 5 | 41 | 39 | 6 | 11 |

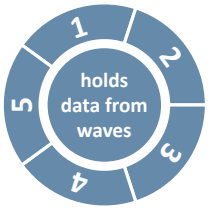
In order to provide more comprehensive context information about the day care institutions themselves, from the fourth wave onwards the heads or managers of the institutions were also surveyed in PAPI mode. These data are stored in the file `pInstitution` including key variables such as the number of registered girls (`h217001`) and boys (`h217002`), the number of kindergartens within a radius of 5 km (`h534010`), and the number of employees in the institution (`h219301`).

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

```
** open the CohortProfile
use ${datapath}/SC1_CohortProfile_R_${version}.dta, clear

** merge registered girls and boys to CohortProfile.
** note that this datafile is directly linkable to
** the child (if you have been working with other SCs,
** you may have expected a variable ID_i)
merge 1:1 ID_t wave using ${datapath}/SC1_pInstitution_R_${version}.dta, ///
  keepusing(h217001 h217002) nogen assert(master match)

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

** compute the total numnber of registered children
nepsmiss
generate total_reg=h217001+h217002

**cluster the children according to the quantiles of the institution size
xtile size = total_reg, nq(5)

tab size
```

### 4.2.8 pParent

« go back to overview




Description

Data surveyed from parents (usually mothers)

File structure

long format: 1 row = 1 target in 1 wave

Exemplary data snapshot

ID variables needed to identify a single row

ID_t wave

No. of variables 2,005

| ID_t | wave | p731905 | p731955 | p741001 | p102030 |
|---|---|---|---|---|---|
| 8057194 | 2 | 1 | 1 | 4 | 3 |
| 8057194 | 3 | . | . | 4 | -54 |
| 8057239 | 2 | 5 | 5 | 3 | 3 |
| 8065201 | 2 | 1 | 1 | 4 | 3 |
| 8065201 | 3 | . | . | 4 | -54 |
| 8066375 | 2 | 5 | 2 | 3 | 2 |
| 8066375 | 3 | . | . | 3 | -54 |

Parent data from both the CATI and the CAPI survey modes are available in the file `pParent`. The dataset covers different topics ranging from personal characteristics of the parent or partner, such as the respondent's occupational status (`p731905`) or that of the partner (`p731955`), to household specific matters, such as the size of the household (`p741001`), to topics directly related to the target child, such as the child's vocabulary size (`p102030`). Note that some information collected from the parents is in episode format, so it is not stored in the `pParent` data file, but in separate spell datasets.

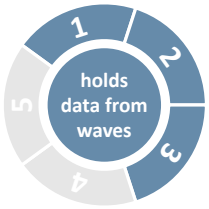
**Example 8 (Stata):** Working with pParent (find R example here)

```
** open the CohortProfile
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear

** merge week of pregnancy at birth and breastfeeding duration
** from pParent
merge 1:1 ID_t wave using ${datapath}/SC1_pParent_D_${version}.dta, ///
  keepusing(p529100 p526200 p526201) nogen assert(master match)

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

** recode missings
nepsmiss p529100 p526200 p526201

** note that the week of pregnancy at birth has only been surveyed
** once, in wave 1
tab p529100 wave

** thus, to work with this (static) information in other waves, you
** first have to carry over the values to other rows
bysort ID_t (wave): replace p529100=p529100[_n-1] if missing(p529100)
```

```
** generate one variable containing the total duration
** of breastfeeding in weeks (assuming 1 month == 4 weeks)
generate bfeed = p526200*4 + p526201

** check the correlation between week of pregnancy at birth and duration
** of breastfeeding
corr p529100 bfeed
```

### 4.2.9 pParentMicrom

« 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

No. of variables 188

Exemplary data snapshot

| ID_t | wave | regio | ID_regio | mso_k_ausland | mso_k_familie | mpi_k_dichte |
|---|---|---|---|---|---|---|
| 8054975 | 1 | 1 | 138818 | 5 | 7 | 5 |
| 8054975 | 1 | 2 | 233514 | 2 | 6 | 6 |
| 8054975 | 1 | 3 | 304210 | 8 | 5 | 4 |
| 8054975 | 1 | 4 | 424533 | 6 | 7 | . |
| 8054975 | 1 | 5 | 502390 | 8 | 4 | 4 |
| 8054975 | 2 | 1 | 133574 | 5 | 7 | 6 |
| 8054975 | 2 | 2 | 229496 | 2 | 6 | 7 |
| 8054975 | 2 | 3 | 304163 | 8 | 5 | 5 |
| 8054975 | 2 | 4 | 422787 | 6 | 7 | . |
| 8054975 | 2 | 5 | 502340 | 8 | 4 | 4 |

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

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

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

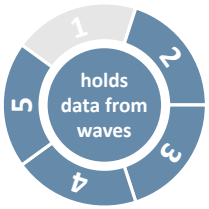
**Example 9 (Stata):** Working with pParentMicrom (find R example here)

```
** open Microm datafile. Note that this data file is only available OnSite!
use ${datapath}/SC1_pParentMicrom_0_${version}.dta, clear

** additional 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 all waves
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 ${datapath}/SC1_CohortProfile_0_${version}.dta
```

### 4.2.10 spChildCare

« go back to overview




Description

Spell data on child care episodes relating to the target child

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell sptype

Other ID variables useful for linkage

wave

Exemplary data snapshot

No. of variables 23

| ID_t | wave | sptype | spell | pa0112m | pa0112y | pa0113m | pa0113y | pa01510 |
|---|---|---|---|---|---|---|---|---|
| 8057265 | 2 | 2 | 201 | 11 | 2012 | 4 | 2013 | . |
| 8057265 | 2 | 5 | 201 | 11 | 2012 | 4 | 2013 | 1 |
| 8057265 | 2 | 5 | 202 | 11 | 2012 | 4 | 2013 | 2 |
| 8057265 | 3 | 2 | 301 | 4 | 2013 | 6 | 2014 | . |
| 8057265 | 3 | 5 | 301 | 4 | 2013 | 6 | 2014 | 1 |
| 8057265 | 4 | 1 | 401 | 4 | 2014 | 5 | 2015 | . |
| 8057265 | 5 | 1 | 501 | 5 | 2015 | 6 | 2016 | . |
| 8057265 | 5 | 5 | 501 | 5 | 2015 | 6 | 2016 | 1 |

The data file `spChildCare` contains all child care episodes relating to the target child, differentiated according to the carer (e. g., grandparent, nanny, childminder); see the variable `sptype`. Besides the start and end dates of the respective episodes (`pa0112m/y`, `pa0113m/y`), it essentially contains structural information such as an identification number of the caregivers (e. g. grandparents number `pa01510`).

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

```
** open the data file
use ${datapath}/SC1_spChildCare_D_${version}.dta, clear
label language en

** check who provided the child care
tab sptype

** only keep episodes where child care has been provided by au-pair
keep if sptype==4

** generate the total duration of the episode (in months)
generate ep_start=ym(pa0112y, pa0112m)
generate ep_end=ym(pa0113y, pa0113m)
generate duration=ep_end-ep_start+1

** check if this was correctly computed
list pa0112m pa0112y pa0113m pa0113y ep_start ep_end duration in 1/10

** display basic statistics for the duration of au-pair child care
summarize duration
```

### 4.2.11 spEmp

« go back to overview


Description

Spell data on parents' employment episodes (self-reported)

File structure

spell format: 1 row = 1 episode of 1 respondent

Exemplary data snapshot

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

No. of variables 24

| ID_t | wave | spell | p731504 | p731505 | p731509 |
|---|---|---|---|---|---|
| 8055062 | 2 | 1 | 2 | 20 | . |
| 8055062 | 4 | 1 | 2 | 20 | 30 |
| 8055062 | 4 | 1 | . | -54 | 30 |
| 8055062 | 4 | 2 | 2 | -54 | . |
| 8055092 | 2 | 1 | 2 | 40 | 0 |
| 8055092 | 2 | 2 | 5 | 50 | 0 |
| 8055229 | 2 | 1 | 5 | 40 | 10 |

The comprehensive dataset spEmp covers all episodes of regular employment of the responding parent. 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 comprises information such as the type of occupation (p731504), working hours 12 months prior to birth (p731505), or working hours upon respondent taking parental leave (p731509).

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

```
** open the data file
use ${datapath}/SC1_spEmp_D_${version}.dta, clear
label language en

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

** note that many respondents have more than one spell
** in this datafile. So you cannot merge this datafile
** to CohortProfile without any further editing
tab spell
```

```
** to check them out, we first create an additional variable
** containing the amount of spells for every respondent
egen max_spell=max(spell), by(ID_t)

** next, we have a look at those respondents with the most
** spells (more than 6 episodes)
list ID_t spell p73159m-p73158c if max_spell>6, sepby(ID_t)

** altering the above line by adding or removing variables
** and conditions, you will most likely get a feeling which
** data is most relevant for you and how you might aggregate
** the episode file to your needs.
** As a stub, we now only keep the first episode.
** You rather might want to aggregate the datafile in
** a more elaborate way such as keeping:
** - the last episode
** - the longest episode
** - the episode with the highest 'outcome' or any other specific episode
** - an aggregation of all (or a subset of) episodes
** - etc.
keep if spell==1

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

** open the CohortProfile data file
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear
label language en

** merge the previously created temporary data file to this
** note that this is wave independent, so your aggregated
** data matches to every row (every wave) of the respondent
merge m:1 ID_t using `tmp' , keep(master match)
```

### 4.2.12 spParLeave

« go back to overview


Description

Spell data on parents' parental leave episodes (self-reported)

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

Exemplary data snapshot

No. of variables 16

| ID_t | spell | subspell | wave | pa0403m | pa0403y | pa0404m | pa0404y |
|---|---|---|---|---|---|---|---|
| 8055141 | 1 | 0 | 4 | 4 | 2012 | 1 | 2015 |
| 8055141 | 1 | 1 | 2 | 4 | 2012 | 4 | 2013 |
| 8055141 | 1 | 2 | 4 | 4 | 2012 | 1 | 2015 |
| 8055144 | 1 | 0 | 2 | 6 | 2012 | 9 | 2012 |
| 8055144 | 2 | 0 | 2 | 9 | 2012 | 4 | 2013 |
| 8055145 | 1 | 0 | 4 | 6 | 2012 | 3 | 2015 |
| 8055145 | 1 | 1 | 2 | 6 | 2012 | 5 | 2013 |
| 8055145 | 1 | 2 | 4 | 6 | 2012 | 3 | 2015 |

The data file spParLeave essentially comprises all start and end dates of parental leave episodes (pa0403m/y, pa0404m/y) of the responding parent.

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

```
** open the data file
use ${datapath}/SC1_spParLeave_D_${version}.dta, clear
label language en

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

** generate a Stata variable for the start and end of the episode
generate ep_start=ym(pa0403y,pa0403m)
generate ep_end=ym(pa0404y,pa0404m)

** compute the duration of this episode in months
generate duration = ep_end - ep_start + 1

** sum up all durations of one respondent to give the total
** parental leave time in months
egen total_parleave = sum(duration), by(ID_t)

** only keep the relevant variables
keep ID_t total_parleave

** the total parleave has been added to every row (i.e., every episode)
** we just need it once, though, so we drop all duplicate entries
duplicates drop
```

```
** now you can see that the respondents ID is the sole identificator
isid ID_t

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

** now, open the CohortProfile
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear
label language en

** merge the previously computed total parleave time
** as this is a time-invariant information, we can merge
** it to every wave
merge m:1 ID_t using `temp', keep(master match) nogenerate
```

### 4.2.13 spPartnerEmp

« go back to overview


Description

Spell data on employment episodes of partners of responding parents (proxy information)

File structure

spell format: 1 row = 1 episode of 1 respondent

Exemplary data snapshot

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

No. of variables 24

| ID_t | wave | spell | p731604 | p731605 | p731609 |
|---|---|---|---|---|---|
| 8055062 | 2 | 1 | 2 | 40 | . |
| 8055062 | 4 | 1 | 2 | 40 | . |
| 8055062 | 4 | 1 | . | -54 | . |
| 8055092 | 2 | 1 | 2 | 40 | 0 |
| 8055092 | 2 | 2 | 5 | 5 | . |
| 8055229 | 2 | 1 | 5 | . | . |
| 8055229 | 2 | 2 | 2 | . | . |

Analog to spEmp, the dataset spPartnerEmp covers all episodes of regular employment of the **partner** of the responding parent. 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 comprises information such as the type of occupation (p731604), working hours 12 months prior to birth (p731605), or working hours upon respondent taking parental leave (p731609) of the partners of responding parents.

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

```
** open the data file
use ${datapath}/SC1_spPartnerEmp_D_${version}.dta, clear
label language en

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

** note that many respondents have more than one spell
** in this datafile. So you cannot merge this datafile
** to CohortProfile without any further editing
tab spell
```

```
** to check them out, we first create an additional variable
** containing the amount of spells for every respondent
egen max_spell=max(spell), by(ID_t)

** next, we have a look at those respondents with the most
** spells (more than 6 episodes)
list ID_t spell p73169m p73168c if max_spell>6, sepby(ID_t)

** altering the above line by adding or removing variables
** and conditions, you will most likely get a feeling which
** data is most relevant for you and how you might aggregate
** the episode file to your needs.
** As a stub, we now only keep the first episode.
** You rather might want to aggregate the datafile in
** a more elaborate way such as keeping:
** - the last episode
** - the longest episode
** - the episode with the highest 'outcome' or any other specific episode
** - an aggregation of all (or a subset of) episodes
** - etc.
keep if spell==1

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

** open the CohortProfile data file
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear
label language en

** merge the previously created temporary data file to this
** note that this is wave independent, so your aggregated
** data matches to every row (every wave) of the respondent
merge m:1 ID_t using `tmp' , keep(master match)
```

### 4.2.14 spPartnerParLeave

« go back to overview


Description

Spell data on parental leave episodes of partners of responding parents (proxy information)

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

Exemplary data snapshot

No. of variables 15

| ID_t | spell | subspell | wave | pa0503m | pa0503y | pa0504m | pa0504y |
|---|---|---|---|---|---|---|---|
| 8055016 | 1 | 0 | 2 | 9 | 2012 | 2 | 2013 |
| 8055016 | 2 | 0 | 4 | 4 | 2015 | 4 | 2015 |
| 8055274 | 1 | 0 | 2 | 2 | 2012 | 4 | 2012 |
| 8055277 | 1 | 0 | 2 | 3 | 2012 | 5 | 2012 |
| 8055277 | 2 | 0 | 2 | 2 | 2013 | 3 | 2013 |

Analog to spParLeave, the data file spPartnerParLeave essentially comprises all start and end dates of parental leave episodes (pa0503m/y, pa0504m/y) of the **partner** of the responding parent.

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

```
** open the data file
use ${datapath}/SC1_spPartnerParLeave_D_${version}.dta, clear
label language en

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

** generate a Stata variable for the start and end of the episode
generate ep_start=ym(pa0503y,pa0503m)
generate ep_end=ym(pa0504y,pa0504m)

** compute the duration of this episode in months
generate duration = ep_end - ep_start + 1

** sum up all durations of one respondent to give the total
** parental leave time in months
egen total_parleave_partner = sum(duration), by(ID_t)

** only keep the relevant variables
keep ID_t total_parleave_partner

** the total parleave has been added to every row (i.e., every episode)
** we just need it once, though, so we drop all duplicate entries
duplicates drop

** now you can see that the respondents ID is the sole identificator
```

```
isid ID_t

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

** now, open the CohortProfile
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear
label language en

** merge the previously computed total parleave time
** as this is a time-invariant information, we can merge
** it to every wave
merge m:1 ID_t using `temp', keep(master match) nogenerate
```

### 4.2.15 spSibling

« go back to overview


Description

Spell data on siblings of the respondent

File structure

entity format: 1 row = 1 sibling of 1 respondent

ID variables needed to identify a single row

ID_t p732105

Other ID variables useful for linkage

wave

Exemplary data snapshot

No. of variables 34

| ID_t | p732105 | wave | p73221m | p73221y | p732401 | p732313 |
|---|---|---|---|---|---|---|
| 8060949 | 1 | 4 | 1 | 2002 | . | . |
| 8060949 | 2 | 4 | 4 | 2010 | . | . |
| 8060949 | 3 | 4 | 8 | 1990 | 1 | 3 |
| 8064418 | 1 | 4 | 11 | 2013 | . | . |
| 8064418 | 2 | 4 | 3 | 1985 | 4 | 5 |
| 8064418 | 3 | 4 | 7 | 1988 | 1 | 4 |

The dataset `spSibling` informs about all reported siblings of the respondent. Each sibling is stored in a line with information about the date of birth (`p73221m/y`), the employment status (`p732401`), and highest school-leaving qualification (`p732313`) and so on.

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

```
** aim of this example is to evaluate the number of older and younger
** siblings of a respondent

** first, we have to get the birth date of the respondent
use ${datapath}/SC1_pParent_D_${version}.dta, clear
keep if wave==1  // only first wave as this data is time-invariant
keep ID_t p70012m p70012y
label language en
tempfile temp
save `temp'

** now, open the spSibling data file
use ${datapath}/SC1_spSibling_D_${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:
generate sibling_bdate=ym(p73221y,p73221m)
generate  target_bdate=ym(p70012y,p70012m)
format *_bdate %tm

** check the difference between the two
generate older=.
```

```
replace older=0 if sibling_bdate>target_bdate
replace older=1 if sibling_bdate<target_bdate
replace older=. if missing(sibling_bdate) | missing(target_bdate)

** care about twins. As we do not know the day (or even the hour),
** we can not know which is older. We set this for a missing thus.
replace older=. if (sibling_bdate==target_bdate)

** generate the total amount of older siblings
egen total_older=total(older), by(ID_t)
** generate the total amount of younger siblings
egen total_younger=total(1-older), by(ID_t)

** aggregate to a single line for each respondent.
** the file then is cross-sectional with ID_t the sole identificator
keep ID_t total*
duplicates drop
```

### 4.2.16 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

psu

Exemplary data snapshot

No. of variables 22

| ID_t | psu | stratum | w_t1 | w_t1comp | w_t2 | w_t2comp | w_t12comp |
|---|---|---|---|---|---|---|---|
| 8055118 | 80 | 3 | 0.60171 | 0.59675 | 0.54925 | 0.73222 | 0.74506 |
| 8055212 | 72 | 3 | 0.29649 | 0.28070 | 0.27934 | 0.32064 | 0.31240 |
| 8055420 | 15 | 2 | 0.49290 | 0.47579 | 0.46388 | 0.00000 | 0.00000 |
| 8055478 | 1 | 1 | 3.23837 | 3.07947 | 3.15385 | 0.00000 | 0.00000 |

Weighting variables (starting with `w_`) are included in the `Weights` dataset. The dataset also contains identifiers for stratification (`stratum`). Given the rather complex structure of the 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 Würbach et al., 2016 as well as in further reports regarding the use of weights at the documentation website (see section 1.2).

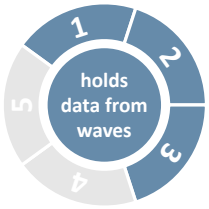
**Example 16 (Stata):** Working with Weights (find R example here)

```
** open Weights datafile
use ${datapath}/SC1_Weights_D_${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_t12345

** create a "panel" logic, i.e., clone each row
expand 5

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

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

** open CohortProfile
use ${datapath}/SC1_CohortProfile_D_${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_t12345!=0
```

### 4.2.17 xDirectMeasures

« go back to overview




Description

Direct measures conducted in the parental home

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*

No. of variables 344

Exemplary data snapshot

| ID_t | wave_w1 | wave_w2 | wave_w3 | ihn1p001_c | cdn1_sc1 | hdn1ah1t_p |
|---|---|---|---|---|---|---|
| 8055016 | 1 | 0 | 1 | -21 | -1.57 | 4700.33333 |
| 8055141 | 1 | 1 | 1 | 5 | -2.81 | 4000.33333 |
| 8055152 | 1 | 1 | 1 | 5 | -55.00 | -55.00000 |
| 8055240 | 1 | 1 | 0 | -55 | 0.48 | 5433.33333 |

This file provides the data from the direct measures conducted in the parental home. These measures – namely parent-child-interaction (starting with `ih*`), habituation-dishabituation paradigm (starting with `hd*`), and sensorimotor development (starting with `cd*`) – were decoded from videotaped observations. The data file contains one row per 'respondent' with the rated items for all three direct measures plus time stamps and coder id.

Further information on the process of coding the video-based material can be found on the website; see for example Sommer and Mann, 2015 for data generation on parent-child-interaction. Table 2 gives an overview of the content and timing of the direct measures.

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

```
** open datafile
use ${datapath}/SC1_xDirectMeasures_D_${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 direct measures have been conducted in multiple waves.
** an indicator marks if a row contains information for a specific wave
tab1 wave_w*

** to work with this data, you might want to merge it to CohortProfile.
** if you want to keep the panel logic (and not only add all rows of this file
** to every wave), you need a mergeable wave variable here.
** in this example, we focus on sensorimotor-development,
** which has been measured in wave 1.
```

```
generate wave=1

** now, remove rows which do not hold relevant information
drop if wave_w1==0

** and reduce the dataset to the relevant variables
keep ID_t wave cdn1_sc1 cdn1_sc2

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

** open CohortProfile
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear

** and merge the tempfile to this
merge 1:1 ID_t wave using `tmp', nogen
```

### 4.2.18 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*

No. of variables 276

Exemplary data snapshot

| ID_t | wave_w4 | wave_w5 | von40001 | dsn42101 | mak1z17s_c |
|---|---|---|---|---|---|
| 8055070 | 1 | 1 | 1 | 1 | 1 |
| 8055084 | 0 | 1 | -56 | -56 | 1 |
| 8055211 | 1 | 1 | 1 | 0 | 2 |
| 8055408 | 1 | 1 | 1 | 1 | -97 |

The file `xTargetCompetencies` contains the data of the competence tests with the respondents. These are currently the domain-specific competencies vocabulary and mathematical competence as well as the stage-specific competencies categorization, delayed gratification, digit span, and executive control. Scored item variables and aggregated scale variables are available in a cross-sectional format (see Table 2 for an overview of the content and timing of the competence measures; see section 3.2.2 for naming conventions). The variables `wave_w*` allow you to select those target persons for whom only data from a specific wave is available.

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

```
** open datafile
use ${datapath}/SC1_xTargetCompetencies_D_${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 5.
generate wave=5

** now, remove cases which did not took part in the testing
```

```
drop if wave_w5==0

** and reduce the dataset to the relevant variables
keep ID_t wave mak1_sc1 mak1_sc2

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

** open CohortProfile
use ${datapath}/SC1_CohortProfile_D_${version}.dta, clear

** and merge the tempfile to this
merge 1:1 ID_t wave using `tmp', nogen
```

# 5 Special Issues

## 5.1 On the use of data from direct and competence measures

**Wave 2:** Note that the sample size for the direct measures in this wave was reduced for design reasons. First, the entire sample participated in a telephone interview (CATI). Subsequently, a subset of target children (families) took part in the direct measures within a personal interview field (CAPI). For this purpose, a random subsample of 34 municipalities was drawn from the initial 84 municipalities (see also section 2.2 for the general sampling strategy and section 2.4 for wave-specific descriptions). Since the direct measures were age-sensitive (as in the wave before and after), a specific time frame was defined for each target child, so that it was between 16 and 17 months old at the time the direct measures were conducted.

**Vocabulary:** In wave 4 a vocabulary measure was used for the first time. When working with these data, please note that only the data of the test phase–but not of the training phase–are published in the Scientific Use File. Due to the stop criterion implemented in the instrument, there are children who have not reached the test phase and therefore have no data from the test phase. In the data the values for these children are not coded with 0, but with the special missing code –94 *not reached*. There are two variables in the data file `xTargetCompetencies`, namely `von4_sc3` and `von4_sc8`, which contain information about the training phase. Here the missing code –24 *training phase failed* means that only training items were conducted.

## 5.2 Change of interviewee or responding parent

The CAPI and CATI interviews were conducted with a parent or legal guardian of the target person (child). In general, the same person is interviewed in each wave. Nevertheless, in exceptional cases it is possible to change the interviewee if the new person fulfils the requirements (e. g. biological or social parents, the new person lives with the child). This possibility exists in all waves. In the data files there is only a child-specific ID, so that the interviewed parent cannot be traced back. For example, the mother of a target child participated in the first wave interview, the father was interviewed in the second wave and the mother again in the third wave. Using the variables `px80212` in the data files `MethodsCAPI` and `MethodsCATI` it is possible to identify the change of the interviewee from wave to wave in the data. However, it is **not** possible to recognize that the same person–in this case the mother–participated in the first and the third wave. The variable mentioned is therefore an indicator of the change of the interviewee, but **not** a person identifier for the responding parent.

## 5.3 Child care

Variables with child care information are contained in various data files: `pEducator` (PAPI), `pEducatorChildminder` (PAPI), `pInstitution` (PAPI), `pParent` (CAPI/CATI), `spChildCare` (generated from parent CAPI/CATI). Because Starting Cohort 1is based on an individual sample, the corresponding questionnaires (PAPI) were passed on from the parents to the educators or the childminders. This means that all information from educators, childminders and institution managers (for the first time in wave 4) is directly linked to the target child and there are no identifier variables available for educators, childminders or institutions. Therefore, external child care persons and institutions can only be connected to the child via `ID_t` and not themselves followed through the survey waves.

For further information, please refer to the presentation of the various data files in section 4.2. In contrast to other episode data files, `spChildCare` does not contain a harmonized subspell variable. The episodes may not be complete due to the survey instrument. A look at the instruments on the NEPS website should help to understand the structure of the panel and episode data:

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

## 5.4 Preloads

Preloads contain information from previous survey waves and make it possible to update this information in the current survey wave. In Starting Cohort 1, preloads were introduced for the first time in wave 3. Consequently, there is no follow-up information via preloads available in wave 2 (e. g. on socio-demographic or partnership characteristics).

# 6 References

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

FDZ-LIfBi. (2018). *Data Manual NEPS Starting Cohort 1– Newborns, Education from the Very Beginning, Scientific Use File Version 5.0.0*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

NEPS (Ed.). (2018). *Starting Cohort 1: Newborns (SC1), Wave 5, Questionnaires (SUF Version 5.0.0)*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

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

Sommer, A. & Mann, D. (2015). *Qualität elterlichen Interaktionsverhaltens* (NEPS Working Paper No. 56). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

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

Würbach, A., Zinn, S., & Aßmann, C. (2016). *Samples, Weights, and Nonresponse: the Early Childhood Cohort of the National Educational Panel Study (Wave 1 to 3)* (NEPS Survey Paper No. 8). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

# A Appendix

## A.1 R examples

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

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

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

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

To import a data set, use:

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

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

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

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

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

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

spEmp_meta_names = as.vector(spEmp_meta$names)
```

```
#extracts the column "names" as vector "spEmp_meta_names"

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

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

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

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

**Example 19 (R):** Working with CohortProfile

```
'** import the data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta",
        convert.factors = T)

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

'** respondents in each wave'
cbind(addmargins(table(CohortProfile$wave)),
      addmargins(prop.table(table(CohortProfile$wave))))


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

**Example 20 (R):** Working with MethodsCAPI

```
'** import the data file'
MethodsCAPI =
        read.dta13("SC1_MethodsCAPI_D_version_en.dta",
        convert.factors = T)

'** check out participation status by wave'
cbind(addmargins(table(MethodsCAPI$wave, MethodsCAPI$px80220)))

'** how many different interviewers did CAPI surveys?'
```

```
length(unique(MethodsCAPI$ID_int))
#number of distinct ID_int INCLUDING NA (Missing Values)

length(unique(MethodsCAPI$ID_int[!is.na(MethodsCAPI$ID_int)]))
#number of distinct ID_int EXCLUDING NA (Missing Values)

'** create one single variable containing the interview date'
install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

Sys.setlocale("LC_TIME", "C")
#turns off the location-specific language, such that the english month names are
  recognized as months.

MethodsCAPI$intdate=
        as.yearmon(paste(MethodsCAPI$inty, MethodsCAPI$intm, sep = '-'), "%Y-%B")
#bind the two columns "intm" and "inty" into one new column "intdate"

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

**Example 21 (R):** Working with MethodsCATI

```
'** import the data file'
MethodsCATI =
        read.dta13("SC1_MethodsCATI_D_version_en.dta",
        convert.factors = T)

'** check out participation status by wave'
cbind(addmargins(table(MethodsCATI$wave, MethodsCATI$px80220)))

'** how many different interviewers did CATI surveys?'
length(unique(MethodsCATI$ID_int))
#number of distinct ID_int INCLUDING NA (Missing Values)

length(unique(MethodsCATI$ID_int[!is.na(MethodsCATI$ID_int)]))
#number of distinct ID_int EXCLUDING NA (Missing Values)

'** create one single variable containing the interview date'
install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

Sys.setlocale("LC_TIME", "C")
#turns off the location-specific language, such that the english month names are
  recognized as months.

MethodsCATI$intdate=
        as.yearmon(paste(MethodsCATI$inty, MethodsCATI$intm, sep = '-'), "%Y-%B")
#bind the two columns "intm" and "inty" into one new column "intdate"

head(MethodsCATI[c("intm", "inty", "intdate")], 10)
```

```
#displays first 10 rows of variables intm, inty and intdate
```

**Example 22 (R):** Working with MethodsDirectMeasures

```
'** import the data file'
MethodsDirectMeasures =
        read.dta13("SC1_MethodsDirectMeasures_D_version_en.dta",
        convert.factors = T)

'** check out the different outcomes of parent-child interaction.'
cbind(table(MethodsDirectMeasures$px02002),
      prop.table(table(MethodsDirectMeasures$px02002)),
      cumsum(prop.table(table(MethodsDirectMeasures$px02002))))

'** also, note that not all interactions have been measured
 ** between respondent (usually mother) and child. Some
 ** have been conducted together with the respondents partner'
cbind(table(MethodsDirectMeasures$px02003_v1),
      prop.table(table(MethodsDirectMeasures$px02003_v1)),
      cumsum(prop.table(table(MethodsDirectMeasures$px02003_v1))))
```

**Example 23 (R):** Working with pEducator

```
'** import the data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_R_version_en.dta", convert.factors = T)
pEducator =
        read.dta13("SC1_pEducator_R_version_en.dta", convert.factors = T)


'** merge sex and year of birth of educator to CohortProfile.
** note that this datafile is directly linkable to
** the child (if you have been working with other SCs,
** you may have expected a variable ID_e)'
CohortProfile =
        merge(x = CohortProfile,
        y = pEducator[,c("ID_t", "wave", "e761110", "e76112y")],
        by = c("ID_t", "wave"), all.x = TRUE)
# merges only variables "e761110" and "e76112y" from pEducator to CohortProfile


'** now, compute the age of the educator at the date of the interview'
CohortProfile$inty[CohortProfile$inty < 0] = NA
# first, replace all negative values (nepsmissings) with NA
CohortProfile$e76112y[CohortProfile$e76112y < 0] = NA
# first, replace all negative values (nepsmissings) with NA

CohortProfile$ed_age = CohortProfile$inty - CohortProfile$e76112y
# create a new variable "ed_age" that ist the age of the educator

summary(CohortProfile$ed_age)
# displays Min, Max and Mean of "ed_age"
```

```
sd(CohortProfile$ed_age, na.rm = TRUE)
# displays Std.Dev. of "ed_age"
length(CohortProfile$ed_age[!is.na(CohortProfile$ed_age)])
# displays the number of observations in "ed_age" without NA
```

**Example 24 (R):** Working with pEducatorChildminder

```
'** import the data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_R_version_en.dta", convert.factors = T)
pEducatorChildminder =
        read.dta13("SC1_pEducatorChildminder_R_version_en.dta", convert.factors = T)


'** merge sex and year of birth of childminder to CohortProfile.
** note that this datafile is directly linkable to
** the child (if you have been working with other SCs,
** you may have expected a variable ID_e)'
CohortProfile =
        merge(x = CohortProfile,
        y = pEducatorChildminder[,c("ID_t", "wave", "e767110", "e76712y")],
        by = c("ID_t", "wave"), all.x = TRUE)
# merges only variables "e767110" and "e76712y" from pEducatorChildminder to
  CohortProfile


'** now, compute the age of the childminder at the date of the interview'
CohortProfile$inty[CohortProfile$inty<0] = NA
# first, replace all negative values (nepsmissings) with NA
CohortProfile$e76712y[CohortProfile$e76712y<0] = NA
# first, replace all negative values (nepsmissings) with NA

CohortProfile$cm_age = CohortProfile$inty - CohortProfile$e76712y
# create a new variable "cm_age" that ist the age of the childminder


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

**Example 25 (R):** Working with pInstitution

```
'** import the data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_R_version_en.dta", convert.factors = T)
pInstitution =
        read.dta13("SC1_pInstitution_R_version_en.dta", convert.factors = T)
```

```
'** merge registered girls and boys to CohortProfile.
** note that this datafile is directly linkable to
** the child (if you have been working with other SCs,
** you may have expected a variable ID_i)'
CohortProfile =
        merge(x = CohortProfile,
              y = pInstitution[,c("ID_t", "wave", "h217001", "h217002")],
              by = c("ID_t", "wave"), all.x = TRUE)
# merges only variables "h217001" and "h217002" from pInstitution to CohortProfile


'** compute the total number of registered children'
CohortProfile$h217001[CohortProfile$h217001<0] = NA
# first, replace all negative values (nepsmissings) with NA
CohortProfile$h217002[CohortProfile$h217002<0] = NA
# first, replace all negative values (nepsmissings) with NA

CohortProfile$total_reg = CohortProfile$h217001 + CohortProfile$h217002
# create a new variable "total_reg" that ist the total number of registered children


'**cluster the children according to the quantiles of the institution size'
CohortProfile =
        within(CohortProfile, {size = cut(total_reg,
                quantile(total_reg, probs=0:5/5),
                include.lowest=TRUE, labels=FALSE)})
# the quantile function calculates quantiles (here quintiles)
# probs denotes the thresholds in probabilities (here probs=0:5/5 equals probs=c(0,
  0.2, 0.4, 0.6, 0.8, 1))
# include.lowest = TRUE includes observations that equal to the lowest threshold
  value in the according category
# labels = FALSE returns integer codes for the new variable "size" instead of factor
  categories

cbind(addmargins(table(CohortProfile$size)),
      addmargins(prop.table(table(CohortProfile$size))))
```

**Example 26 (R):** Working with pParent

```
'** import the data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta", convert.factors = T)
pParent =
        read.dta13("SC1_pParent_D_version_en.dta", convert.factors = T)

'** merge week of pregnancy at birth and breastfeeding duration from pParent'
CohortProfile =
        merge(x = CohortProfile,
              y = pParent[,c("ID_t", "wave", "p529100", "p526200", "p526201")],
              by = c("ID_t", "wave"), all.x = TRUE)


'** recode missings'
```

```
for (i in names(CohortProfile[c("p529100", "p526200", "p526201")])) {
  CohortProfile[[i]][CohortProfile[[i]]<0] = NA
  #replace all negative values (nepsmissings) with NA
}


'** note that the week of pregnancy at birth has only been surveyed once, in wave 1'
cbind(addmargins(table(CohortProfile$p529100, CohortProfile$wave)))

'** thus, to work with this (static) information in other waves, you
** first have to carry over the values to other rows'
for (i in 2:length(CohortProfile$ID_t)) {
  if(CohortProfile$ID_t[i] == CohortProfile$ID_t[i-1]){
    if(is.na(CohortProfile$p529100[i])){
      CohortProfile$p529100[i] = CohortProfile$p529100[i-1]
    }
  }
}

cbind(addmargins(table(CohortProfile$p529100, CohortProfile$wave)))
```

**Example 27 (R):** Working with pParentMicrom

```
'** open Microm datafile. Note that this data file is only available OnSite!'
pParentMicrom =
        read.dta13("SC1_pParentMicrom_0_5-0-0.dta", convert.factors = T)


'** additionally to ID_t and wave, line identification in this file is done
 ** via variable regio, denoting the regional level of information'
anyDuplicated(pParentMicrom[,c(1,2,3)])
#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 all waves'
addmargins(table(pParentMicrom$wave, pParentMicrom$regio))

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

'** now you can enhance CohortProfile with regional data'
CohortProfile =
        read.dta13("SC1_CohortProfile_0_5-0-0.dta", convert.factors = T)

pParentMicrom = merge(CohortProfile, pParentMicrom, by = c("ID_t", "wave"), all=TRUE)
```

**Example 28 (R):** Working with spChildCare

```
'** open the data file'
spChildCare =
        read.dta13("SC1_spChildCare_D_version_en.dta", convert.factors = T)
```

```
'** check who provided the child care'
cbind(addmargins(table(spChildCare$sptype)),
      addmargins(prop.table(table(spChildCare$sptype))))


'** only keep episodes where child care has been provided by au-pair'
spChildCare =
        subset(spChildCare, spChildCare$sptype == "Child care provided by au-pair")

'** generate the total duration of the episode (in months)'
install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

Sys.setlocale("LC_TIME", "C")
#turns off the location-specific language, such that the english month names are
  recognized as months.

spChildCare$ep_start =
        as.yearmon(paste(spChildCare$pa0112y, spChildCare$pa0112m,
        sep = '-'), "%Y-%B")

spChildCare$ep_end =
        as.yearmon(paste(spChildCare$pa0113y, spChildCare$pa0113m,
        sep = '-'), "%Y-%B")

spChildCare$duration = (spChildCare$ep_end - spChildCare$ep_start)*12+1

'** check if this was correctly computed'
head(spChildCare[,c("pa0112m", "pa0112y", "pa0113m", "pa0113y", "ep_start", "ep_end",
   "duration")],10)

'** display basic statistics for the duration of au-pair child care'
summary(spChildCare$duration)
#displays Min, Max and Mean for "duration"
sd(spChildCare$duration, na.rm = TRUE)
#displays Std.Dev. for "duration"
length(spChildCare$duration[!is.na(spChildCare$duration)])
#displays the number of observations in "duration" without NA
```

**Example 29 (R):** Working with spEmp

```
'** open the data file'
spEmp =
        read.dta13("SC1_spEmp_D_version_en.dta", convert.factors = T)

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

'** note that many respondents have more than one spell
** in this datafile. So you cannot merge this datafile
```

```
** to CohortProfile without any further editing'
cbind(addmargins(table(spEmp$spell)), addmargins(prop.table(table(spEmp$spell))))

'** to check them out, we first create an additional variable
** containing the amount of spells for every respondent'
spEmp = within(spEmp, {max_spell = ave(spell, ID_t, FUN = max)})


'** next, we have a look at those respondents with the most
** spells (more than 6 episodes)'
View(subset(spEmp[,c(1, 2, 11:15)], spEmp$max_spell > 6))


'** altering the above line by adding or removing variables
** and conditions, you will most likely get a feeling which
** data is most relevant for you and how you might aggregate
** the episode file to your needs.
** As a stub, we now only keep the first episode.
** You rather might want to aggregate the datafile in
** a more elaborate way such as keeping:
** - the last episode
** - the longest episode
** - the episode with the highest outcome or any other specific episode
** - an aggregation of all (or a subset of) episodes etc.'
spEmp = subset(spEmp, spEmp$spell == 1)


'** open the CohortProfile data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta", convert.factors = T)


'** merge the data
** note that this is wave independent, so your aggregated
** data matches to every row (every wave) of the respondent'
CohortProfile = merge(CohortProfile, spEmp, by=c("ID_t"), all.x = TRUE)
```

**Example 30 (R):** Working with spParLeave

```
'** open the data file'
spParLeave =
        read.dta13("SC1_spParLeave_D_version_en.dta", convert.factors = T)

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

'** generate a variable for the start and end of the episode'
install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data

Sys.setlocale("LC_TIME", "C")
```

```
#turns off the location-specific language, such that the english month names are
  recognized as months.

spParLeave$ep_start =
        as.yearmon(paste(spParLeave$pa0403y, spParLeave$pa0403m, sep = '-'), "%Y-%B")
spParLeave$ep_end =
        as.yearmon(paste(spParLeave$pa0404y, spParLeave$pa0404m, sep = '-'), "%Y-%B")


'** compute the duration of this episode in months'
spParLeave$duration = (spParLeave$ep_end - spParLeave$ep_start)*12+1


'** sum up all durations of one respondent to give the total
** parental leave time in months'
spParLeave =
        within(spParLeave, {total_parleave =
        ave(duration, ID_t, FUN = function(x) sum(x, na.rm = TRUE))})

'** only keep the relevant variables'
spParLeave = subset(spParLeave[,c("ID_t", "total_parleave")])

'** the total parleave has been added to every row (i.e., every episode)
** we just need it once, though, so we drop all duplicate entries'
spParLeave = unique(spParLeave)

'** now you can see that the respondents ID is the sole identificator'
anyDuplicated(spParLeave[,c("ID_t")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate


'** open the CohortProfile data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta", convert.factors = T)


'** merge the previously computed total parleave time
** as this is a time-invariant information, we can merge
** it to every wave'
CohortProfile = merge(CohortProfile, spParLeave, by=c("ID_t"), all.x = TRUE)
```

**Example 31 (R):** Working with spPartnerEmp

```
'** open the data file'
spPartnerEmp =
        read.dta13("SC1_spPartnerEmp_D_version_en.dta", convert.factors = T)

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

'** note that many respondents have more than one spell
** in this datafile. So you cannot merge this datafile
```

```
** to CohortProfile without any further editing'
cbind(addmargins(table(spPartnerEmp$spell)),
      addmargins(prop.table(table(spPartnerEmp$spell))))

'** to check them out, we first create an additional variable
** containing the amount of spells for every respondent'
spPartnerEmp = within(spPartnerEmp, {max_spell = ave(spell, ID_t, FUN = max)})


'** next, we have a look at those respondents with the most
** spells (more than 6 episodes)'
View(subset(spPartnerEmp[,c("ID_t", "spell", "p73169m", "p73168c")],
            spPartnerEmp$max_spell > 6))


'** altering the above line by adding or removing variables
** and conditions, you will most likely get a feeling which
** data is most relevant for you and how you might aggregate
** the episode file to your needs.
** As a stub, we now only keep the first episode.
** You rather might want to aggregate the datafile in
** a more elaborate way such as keeping:
** - the last episode
** - the longest episode
** - the episode with the highest outcome or any other specific episode
** - an aggregation of all (or a subset of) episodes etc.'
spPartnerEmp = subset(spPartnerEmp, spPartnerEmp$spell == 1)


'** open the CohortProfile data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta", convert.factors = T)


'** merge the data
** note that this is wave independent, so your aggregated
** data matches to every row (every wave) of the respondent'
CohortProfile = merge(CohortProfile, spPartnerEmp, by=c("ID_t"), all.x = TRUE)
```

**Example 32 (R):** Working with spPartnerParLeave

```
'** open the data file'
spPartnerParLeave =
        read.dta13("SC1_spPartnerParLeave_D_version_en.dta", convert.factors = T)

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

'** generate a variable for the start and end of the episode'
install.packages("zoo")
library(zoo)
#the zoo package is needed to transform time data
```

```
Sys.setlocale("LC_TIME", "C")
#turns off the location-specific language, such that the english month names are
  recognized as months.

spPartnerParLeave$ep_start =
        as.yearmon(paste(spPartnerParLeave$pa0503y, spPartnerParLeave$pa0503m,
        sep = '-'), "%Y-%B")
spPartnerParLeave$ep_end =
        as.yearmon(paste(spPartnerParLeave$pa0504y, spPartnerParLeave$pa0504m,
        sep = '-'), "%Y-%B")


'** compute the duration of this episode in months'
spPartnerParLeave$duration =
        (spPartnerParLeave$ep_end - spPartnerParLeave$ep_start)*12+1


'** sum up all durations of one respondent to give the total
** parental leave time in months'
spPartnerParLeave =
        within(spPartnerParLeave, {total_parleave_partner =
        ave(duration, ID_t, FUN = function(x) sum(x, na.rm = TRUE))})

'** only keep the relevant variables'
spPartnerParLeave = subset(spPartnerParLeave[,c("ID_t", "total_parleave_partner")])

'** the total parleave has been added to every row (i.e., every episode)
** we just need it once, though, so we drop all duplicate entries'
spPartnerParLeave = unique(spPartnerParLeave)

'** now you can see that the respondents ID is the sole identificator'
anyDuplicated(spPartnerParLeave[,c("ID_t")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate


'** open the CohortProfile data file'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta", convert.factors = T)


'** merge the previously computed total parleave time
** as this is a time-invariant information, we can merge
** it to every wave'
CohortProfile = merge(CohortProfile, spPartnerParLeave, by=c("ID_t"), all.x = TRUE)
```

**Example 33 (R):** Working with spSibling

```
'** aim of this example is to evaluate the number of older and younger
 ** siblings of a respondent'


'** first, we have to get the birth date of the respondent'
```

```
#open pParent
pParent =
        read.dta13("SC1_pParent_D_version_en.dta", convert.factors = T)

#display value labels
levels(pParent$wave)

#keep only the first wave as this data is time-invariant
pParent = subset(pParent, pParent$wave == "2012/2013")

#keep only ID_t, p70012m and p70012y from pParent
pParent = subset(pParent, select = c("ID_t", "p70012m", "p70012y"))


'** now, open the data file spSibling'
spSibling =
        read.dta13("SC1_spSibling_D_version_en.dta", convert.factors = T)


'** merge the previously extracted birth dates in pTargetCATI to spSibling'
spSibling = merge(spSibling, pParent, by = c("ID_t"), all.x = TRUE)


'** recode the two date variables (year, month) into one:'
Sys.setlocale("LC_TIME", "C")
#turns off the location-specific language, such that the english month names are
  recognized as months.

View(spSibling[,c("p73221m", "p70012m")])

spSibling$p73221m = match(spSibling$p73221m, month.name)
#transforms month names into month numbers


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

spSibling$sibling_bdate =
        as.yearmon(paste(spSibling$p73221y, spSibling$p73221m), "%Y %m")

spSibling$target_bdate =
        as.yearmon(paste(spSibling$p70012y, spSibling$p70012m), "%Y %m")
#recode the two date variables (year, month) into one

'** check the difference between the two'

spSibling$older = rep(NA, times = length(spSibling$ID_t))
#create an empty variable "older"


#check the difference between the two bdates:
for (i in 1:length(spSibling$older)) {
  if(!is.na(spSibling$sibling_bdate[i]) & !is.na(spSibling$target_bdate[i]) &
```

```
            spSibling$sibling_bdate[i] > spSibling$target_bdate[i]) {
                spSibling$older[i] = 0
    } else {
      if (!is.na(spSibling$sibling_bdate[i]) & !is.na(spSibling$target_bdate[i]) &
                 spSibling$sibling_bdate[i] < spSibling$target_bdate[i]) {
                     spSibling$older[i] = 1
    } else {
      spSibling$older[i] = NA
    }
  }
}


'** generate the total amount of older siblings'
spSibling =
        within(spSibling, {total_older =
                ave(older, ID_t, FUN = function(x) sum(x, na.rm = TRUE))})

'** generate the total amount of younger siblings'
spSibling =
        within(spSibling, {total_younger =
                ave(older, ID_t, FUN = function(x) sum(1-x, na.rm = TRUE))})


'** aggregate to a single line for each respondent.
 ** the file then is cross-sectional with ID_t the sole identificator'

spSibling = subset(spSibling, select = c("ID_t", "total_older", "total_younger"))
#keep only the variables ID_t, total_older and total_younger

spSibling = unique(spSibling)
#drops duplicate rows from spSibling
```

**Example 34 (R):** Working with Weights

```
'** open the data file'
Weights =
        read.dta13("SC1_Weights_D_version_en.dta", convert.factors = T)
#imports the data file
class(Weights)

'** 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_t12345) )

'** create a "panel" logic, i.e. clone each row'
Weights = Weights[rep(seq_len(nrow(Weights)), each = 5),]
```

```
'** then create a wave variable'
Weights$wave = ave(Weights$ID_t, Weights$ID_t, FUN = seq_along)


'** open CohortProfile'
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.dta", convert.factors = T)


table(Weights$wave)
table(CohortProfile$wave)


#Problem: value labels of wave in CohortProfile and Weights are not the same
#the levels of "Wave" in "CohortProfile" and "Weights" have to be equalized
levels(CohortProfile$wave)
levels(Weights$wave)

Weights$wave = as.factor(Weights$wave)
#sets "wave" in "Weights" as factor

for (i in 1:5) {
  levels(Weights$wave)[i] = levels(CohortProfile$wave)[i]
  #assigns the same value labels to "wave" in "Weights" as in "CohortProfile"
}

'## and merge Weights to CohortProfile'
CohortProfile = merge(CohortProfile, Weights, by=c("ID_t", "wave"), all=TRUE)

'## note that this weight is only non-zero if respondents participated in all waves'
with(subset(CohortProfile, w_t12345 != 0), addmargins(table(wave, tx80220)))
```

**Example 35 (R):** Working with xDirectMeasures

```
'** open the data file'
xDirectMeasures =
        read.dta13("SC1_xDirectMeasures_D_version_en.dta", convert.factors = T)

#open the data file Cohort Profile
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.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(xDirectMeasures[,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 direct measures have been conducted in multiple waves.
** an indicator marks if a row contains information for a specific wave'
```

```
table(xDirectMeasures$wave_w1)
table(xDirectMeasures$wave_w2)
table(xDirectMeasures$wave_w3)

'** to work with this data, you might want to merge it to CohortProfile.
** if you want to keep the panel logic (and not only add all rows of this file
** to every wave), you need a mergeable wave variable here.
** in this example, we focus on sensorimotor-development,
** which has been measured in wave 1.'
levels(xDirectMeasures$wave_w1)
xDirectMeasures$wave =
        rep(levels(CohortProfile$wave)[1],length(xDirectMeasures$ID_t))
# take the label for wave 1 from CohortProfile, since the labels have to be equal for
   the later merge
xDirectMeasures$wave = as.factor(xDirectMeasures$wave)
# change the variable type of wave to factor
class(xDirectMeasures$wave)


'** now, remove rows which do not hold relevant information '
levels(xDirectMeasures$wave_w1)
xDirectMeasures = subset(xDirectMeasures, wave_w1 == "Yes")


'** and reduce the dataset to the relevant variables '
xDirectMeasures = subset(xDirectMeasures, select = c(ID_t, wave, cdn1_sc1, cdn1_sc2))


'** and merge the xDirectMeasures to CohortProfile'
levels(CohortProfile$wave)
levels(xDirectMeasures$wave)
CohortProfile =
        merge(CohortProfile, xDirectMeasures, by= c("ID_t", "wave"), all=TRUE)
```

**Example 36 (R):** Working with xTargetCompetencies

```
'** open the data file'
xTargetCompetencies =
        read.dta13("SC1_xTargetCompetencies_D_version_en.dta", convert.factors = T)

#open the data file Cohort Profile
CohortProfile =
        read.dta13("SC1_CohortProfile_D_version_en.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 direct measures have been conducted in multiple waves.
** an indicator marks if a row contains information for a specific wave'
table(xTargetCompetencies$wave_w4)
table(xTargetCompetencies$wave_w5)

'** 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 5.'
levels(xTargetCompetencies$wave_w5)
xTargetCompetencies$wave = rep(levels(CohortProfile$wave)[5],length(
  xTargetCompetencies$ID_t))
# take the label for wave 5 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
class(xTargetCompetencies$wave)


'** now, remove rows which do not hold relevant information '
levels(xTargetCompetencies$wave_w5)
xTargetCompetencies = subset(xTargetCompetencies, wave_w5 == "Yes")


'** and reduce the dataset to the relevant variables '
xTargetCompetencies =
        subset(xTargetCompetencies, select = c(ID_t, wave, mak1_sc1, mak1_sc2))


'** and merge the xDirectMeasures to CohortProfile'
levels(CohortProfile$wave)
levels(xTargetCompetencies$wave)
CohortProfile =
        merge(CohortProfile, xTargetCompetencies, by= c("ID_t", "wave"), all=TRUE)
```

## A.2 Release notes

The following is the release note taken from the documentation page at the time this document has been computed:

```
================================================
**
** NEPS STARTING COHORT 1 - RELEASE NOTES a.k.a CHANGE LOG
** changes and updates for release NEPS SC1 5.0.0
** (doi:10.5157/NEPS:SC1:5.0.0)
**
================================================

* Known Issues *

================================================
* Changes introduced to NEPS:SC1 by version 5.0.0 *
================================================

General:
        - meta data for all variables have been revised and updated where appropriate
        - additional wave 5 has been incorporated into the data

xTargetCompetencies:
        - new published dataset containing data from competency tests from wave 4 and
          later
        - methodical information on these competency tests have been integrated into
          the MethodsDirectMeasures dataset.


================================================
* Changes introduced to NEPS:SC1 by version 4.0.0 *
================================================

General:
        - meta data for all variables have been revised and updated where appropriate
        - additional wave 4 has been incorporated into the data


================================================
* Changes introduced to NEPS:SC1 by version 3.0.0 *
================================================

General:
        - meta data for all variables have been revised and updated where appropriate
        - additional wave 3 has been incorporated into the data

pParent:
        - the concept of reflecting migrational background in NEPS SUFs has been
          improved in order to also represent migrants in 3.75th generation;
              thus, the older variables on migrational background [p400500_g1,
                  p400500_g2,p400500_g3] in the pParent dataset have been renamed
                  using
              the "v1" suffix [p400500_g1v1,p400500_g2v1,p400500_g3v1], and the new
                  ones have been introduced

xDirectMeasures:
        - For 49 observations (27 from wave 1 and 22 from wave 2), no information in
          xDirectMeasures is available;
```

```
                in version 2.0.0, these cases had been coded 0 in all competency
                    variables and
                therefore remained in dataset xDirectMeasures; starting from version
                    3.0.0, these cases have
                consequently been erased from xDirectMeasures.
        - For wave 3, parent-child-interaction had been measured (again), but will not
            be published within this release.
                The parent-child-interaction-data will be likely available with release
                    4.0.0.

================================================
* Changes introduced to NEPS:SC1 by version 2.0.0 *
================================================

General:

        - translation for all meta data (variable and value labels, question texts, etc
            ) have been revised and completed
        - meta data for all variables have been revised and updated where appropriate
        - additional wave 2 has been incorporated into the data

pParent:

        - the variable set containing information from the multiple-response question "
            Birth complications" had been erroneously
                named [p529101] through [p529106] in version 1.0.0; this conflicts to
                    other variable names in NEPS Starting Cohorts 2 and 3;
                the battery has been renamed to [p529110] through [p529115]
```