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 9.0.0

LIfBi
LEIBNIZ INSTITUTE FOR
EDUCATIONAL TRAJECTORIES

Leibniz Institute for Educational Trajectories (LIfBi)
Wilhelmsplatz 3, 96047 Bamberg
Director: Prof. Dr. Cordula Artelt
Administrative Director: Dr. Stefan Echinger
Bamberg; February 25, 2022

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. (2022). *Data Manual NEPS Starting Cohort 1– Newborns, Education from the Very Beginning, Scientific Use File Version 9.0.0*. 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:*

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

For their support in writing this manual, special thanks go to:
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:

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

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

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

Contents

1 Introduction

1.1 About this manual

This manual is intended to facilitate your work with data of NEPS Starting Cohort 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

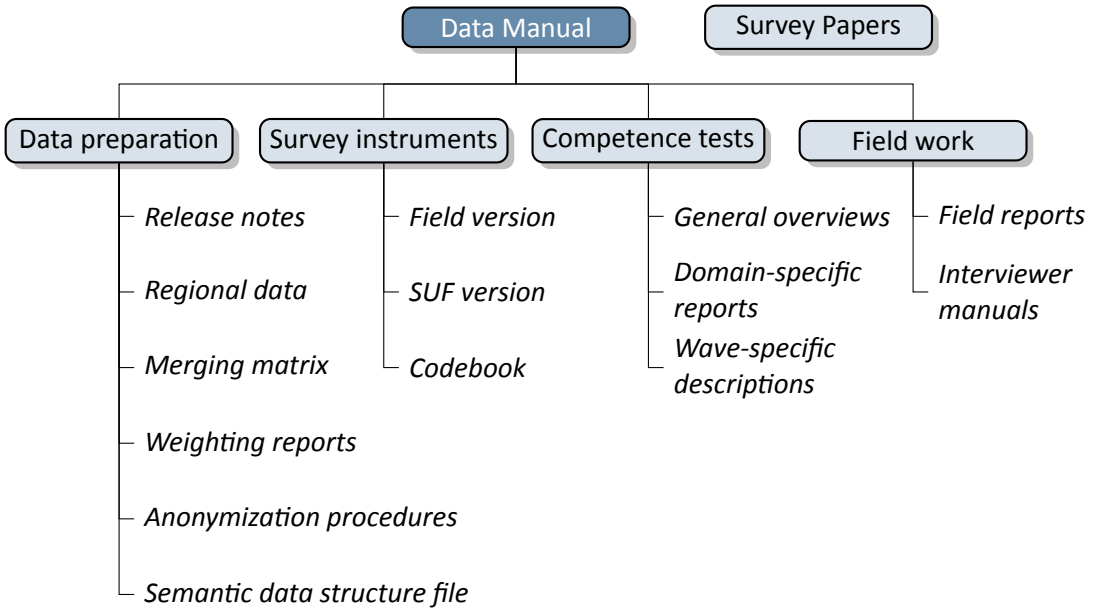

Figure 1: NEPS supplementary data documentation

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 B.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:SC1:9.0.0. 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
9.0.0 (current)	doi:10.5157/NEPS:SC1:9.0.0	**2022-02-25**
8.0.1	doi:10.5157/NEPS:SC1:8.0.1	2021-08-26
8.0.0	doi:10.5157/NEPS:SC1:8.0.0	2021-03-12
7.0.0	doi:10.5157/NEPS:SC1:7.0.0	2020-02-28
6.0.0	doi:10.5157/NEPS:SC1:6.0.0	2019-03-29
5.0.0	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.ndicators 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 citing the data version (DOI) as follows:

> NEPS Network. (2022). *National Educational Panel Study, Scientific Use File of Starting Cohort Newborns*. Leibniz Institute for Educational Trajectories (LIfBi), Bamberg. https://doi.org/10.5157/NEPS:SC1:9.0.0

In addition, the NEPS study is to be referred to at an appropriate place:

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

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

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

Authors of any kind of publications based on 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. (2022). *Data Manual NEPS Starting Cohort 1– Newborns, Education from the Very Beginning, Scientific Use File Version 9.0.0*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study

Or another example:

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

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

> NEPS Network. (2022). *Starting Cohort 1: Newborns (SC1), Wave 9, Questionnaires (SUF Version 9.0.0)*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study

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

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

1.6 Rules and recommendations

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

Rules

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

NEPSplorer

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

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

NEPStools

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

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

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

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

User trainings

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

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

1.9 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 et al., 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	2017 **Wave 6** 5 years	2018 **Wave 7** 6 years	2019 **Wave 8** 7 years	2020 **Wave 9** 8 years
Domain-General Competencies										
DGCF: Cognitive Basic Skills	dg	—	—	—	—	—	—	CBT	—	—
Domain-Specific Competencies										
Early Reading Competence	rx	—	—	—	—	—	—	—	—	CBT[2]
Reading Speed	rs	—	—	—	—	—	—	—	—	CBT
Vocabulary: Listening Comprehension at Word Level	vo	—	—	—	CBT	—	CBT	—	CBT[2]	—
Mathematical Competence	ma	—	—	—	—	CBT	—	CBT	—	CBT[2]
Scientific Competence	sc	—	—	—	—	—	CBT	—	CBT[2]	—
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	—	CBT	—	CBT	—
Digit Span: Phonological Working Memory	ds	—	—	—	CBT	—	—	CBT	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] Subsequent to the respective competence test the target persons had to assess their own test performance (Procedural Metacognition, mp)

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. A more detailed insight into all relevant field work issues is provided by the *Field Reports* of the survey institutes, which are available on the website (in German only) as part of the data documentation for each (sub-)study:

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

Figure 2 starts with an overview illustrating the panel progress of Starting Cohort 1 in terms of field times and survey modes from wave 1 to 9.

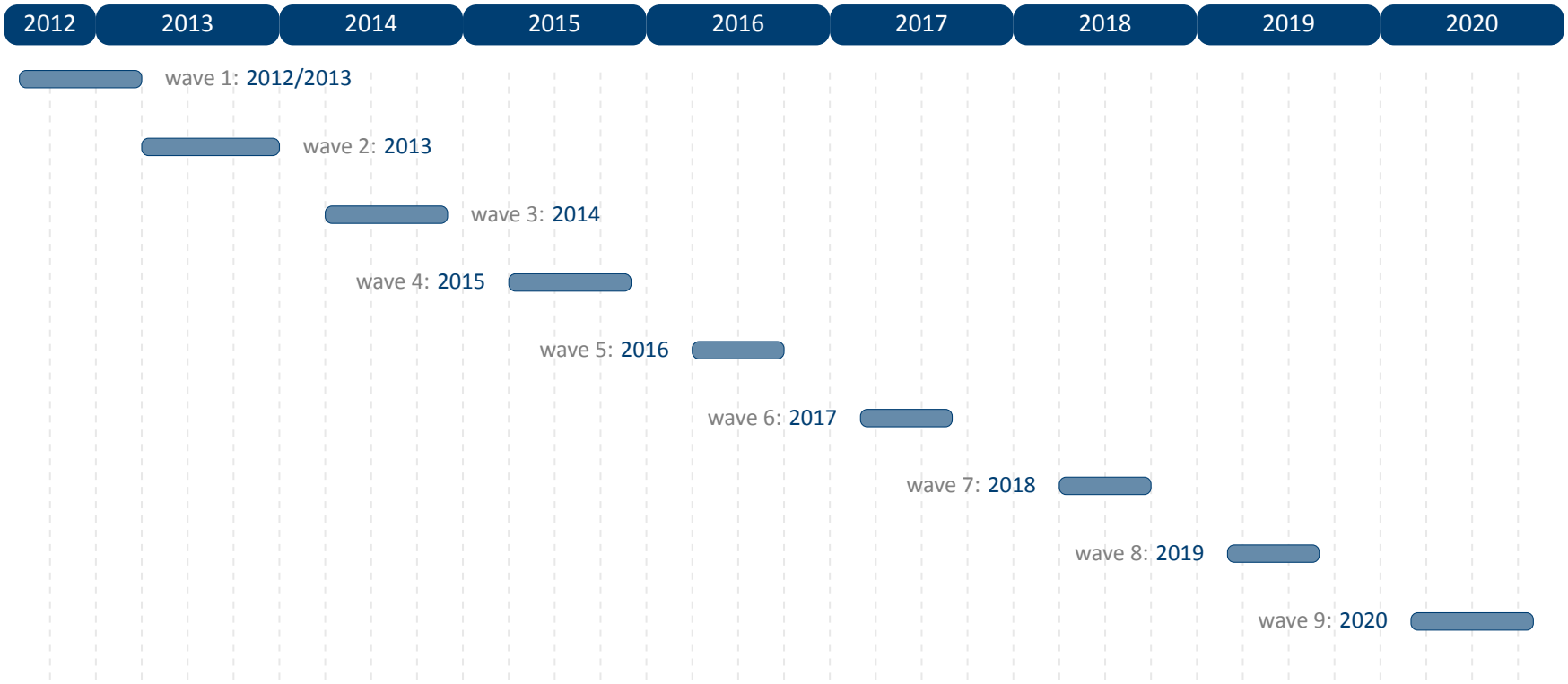

Figure 2: Survey progress of Starting Cohort 1 (waves 1 to 9)

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

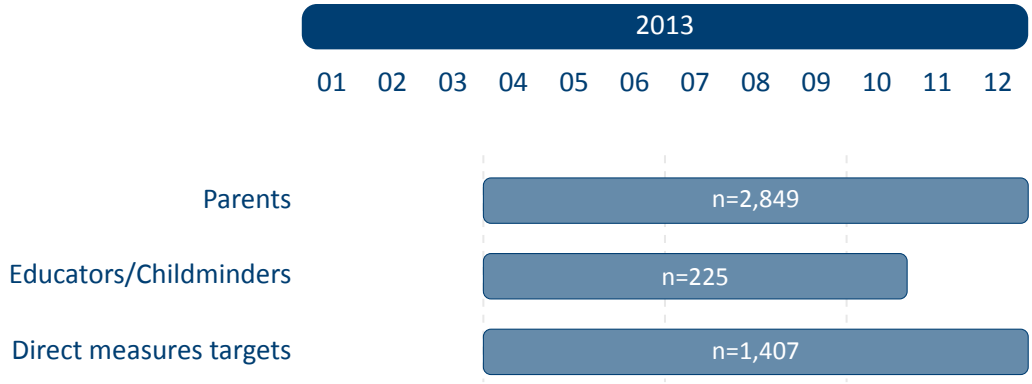

Figure 4: Field times and realized case numbers in wave 2

- 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

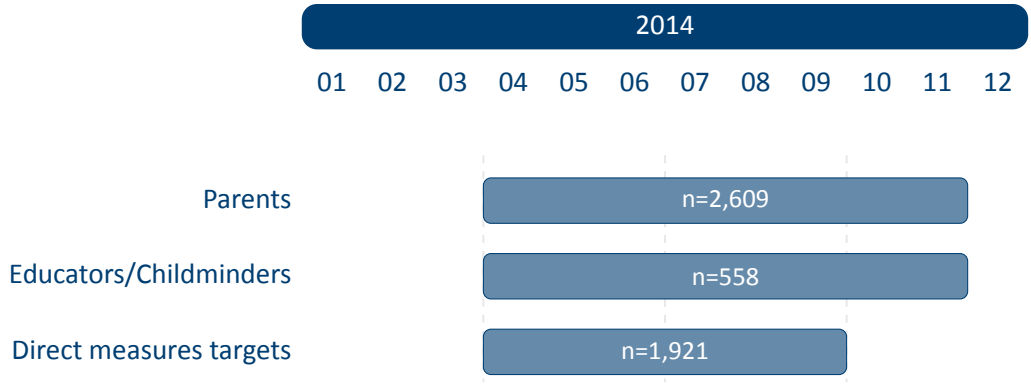

Figure 5: Field times and realized case numbers in wave 3

- 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

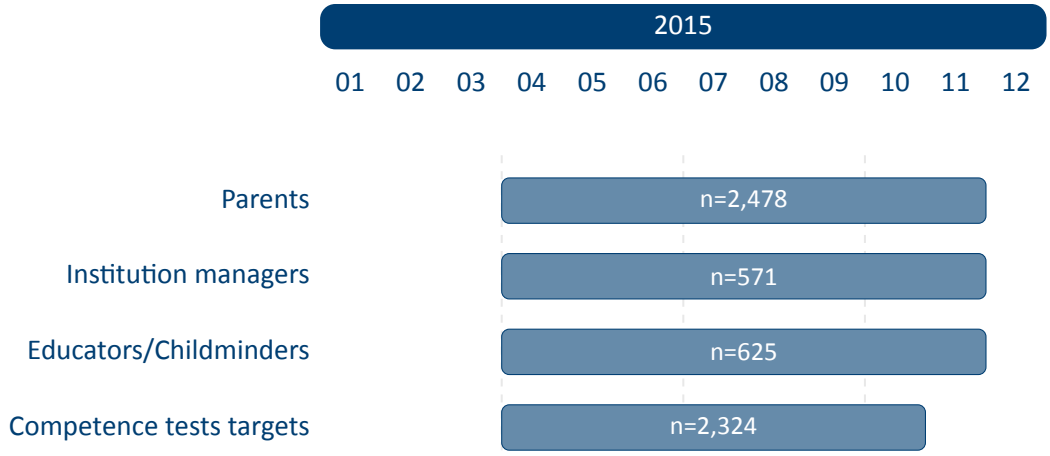

Figure 6: Field times and realized case numbers in wave 4

- 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 competence 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

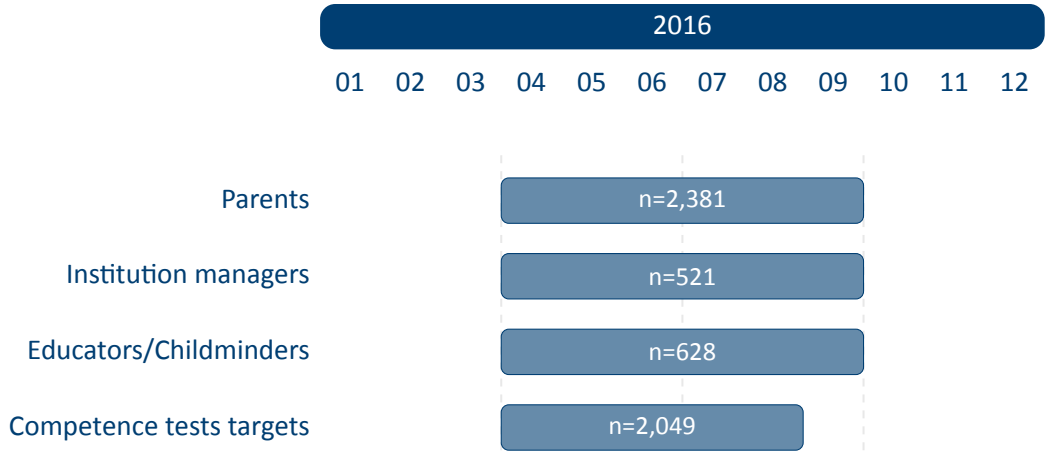

Figure 7: Field times and realized case numbers in wave 5

- 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 competence 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)

2.4.6 Wave 6: 2017

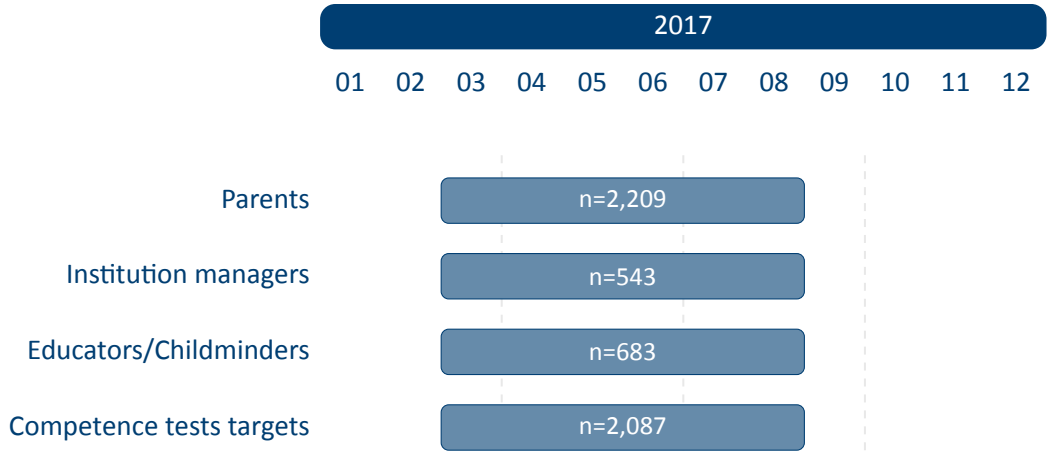

Figure 8: Field times and realized case numbers in wave 6

- Target persons

  Current wave approx. 5 year-old children

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

  Mode of survey computer-based test (CBT/tablet) of competence measures (vocabulary, scientific competence, delayed gratification)

- 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)

2.4.7 Wave 7: 2018

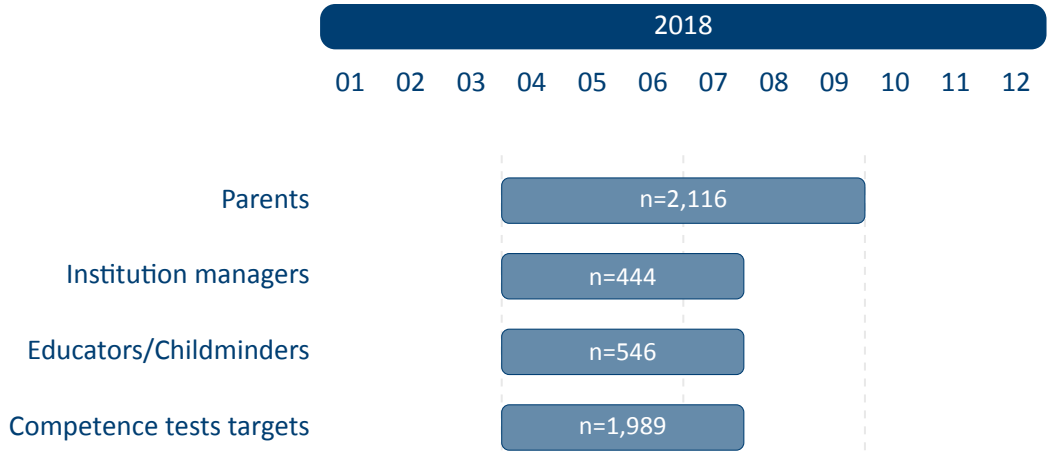

Figure 9: Field times and realized case numbers in wave 7

- Target persons

  Current wave approx. 6 year-old children

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

  Mode of survey computer-based test (CBT/tablet) of competence measures (DGCF: cognitive basic skills, mathematical competence, digit span: phonological working memory)

- 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)

2.4.8 Wave 8: 2019

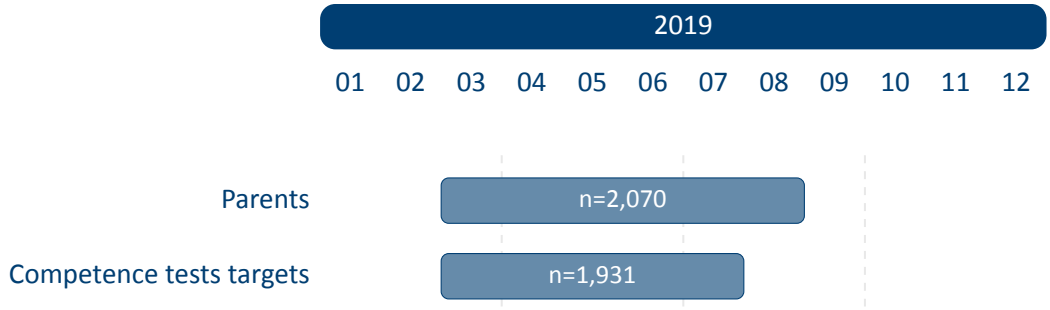

Figure 10: Field times and realized case numbers in wave 8

- Target persons

  Current wave approx. 7 year-old children

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

  Mode of survey computer-based test (CBT/tablet) of competence measures (vocabulary: listening comprehension at word level, scientific competence, delayed gratification: executive control, digit span: phonological working memory)

- 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)

2.4.9 Wave 9: 2020

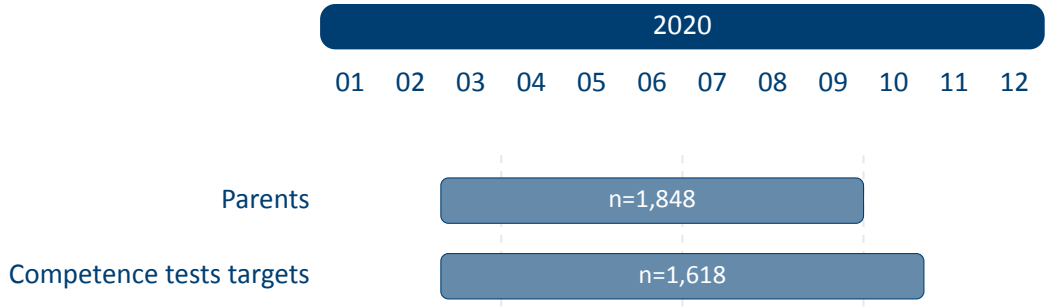

Figure 11: Field times and realized case numbers in wave 9

- Target persons

  Current wave approx. 8 year-old children

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

  Mode of survey computer-based test (CBT/tablet) of competence measures (reading speed, early reading competence, mathematical competence)

  Mode change during fieldwork Due to the Corona pandemic, the CAPI field had to be stopped after three weeks; the survey was restarted in June as CAPI-by-Phone. In this mode, the TBT test modules were converted into an online format in which the target child completed the tasks on a home tablet or computer and the interviewer actively accompanied the testing on the phone.

- Context persons
  - Parents (esp. mothers)

    Mode of survey computer-assisted personal interviews (CAPI) plus written questionnaires (PAPI) during competence testing of the target children

    Mode change during fieldwork Due to the Corona pandemic, the CAPI field had to be stopped after three weeks; the survey was restarted in June as CAPI-by-Phone. In this mode, the CAPI interviewers conducted the parents interviews by telephone from home. The originally planned CATI converting field was dropped.

3 General Conventions

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

The first and foremost paradigm in creating NEPS Scientific Use Files is the one of unaltered data. Wherever possible, the data editing procedures do neither change nor destruct the content of the original data. We consider this to be the basis for preserving the full research potential of the collected data. For this reason, no corrections are made during the entire data editing process to ensure the content validity of the source data. As a consequence, this means that the data in the Scientific Use File may contain implausible values, unless corresponding controls were already provided in the survey instrument. Only in rare cases, in which the responsible developers of a variable require the removal of clearly implausible information, these values are replaced by the special missing code *implausible value removed* (–52, see Table 6). The most prominent (and only systematic) exception to this general paradigm concerns the recoding of open responses that could originally have been recorded directly as closed responses (see section 3.4 for details). NEPS Scientific Use Files are equipped with a dataset `EditionBackups` that contains backup information for all content that has been modified by such recoding procedures (see section 4.5.2 for details).

The second paradigm is to integrate the data as much as possible without compromising the usability of the Scientific Use File. The underlying assumption is that for a vast majority of data users it is far more comfortable to reduce already integrated data for a specific analysis as opposed to correctly compile the relevant information from scattered source data themselves. In the end, each Scientific Use File contains only a few dozen integrated panel and spell datasets according to a general structure (see section 4.3 and section 4.4 for details), even if the compilation is based on several hundred separate source dataset files.

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

3.1 File names

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

Table 3: Naming conventions for NEPS file names

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

For instance, the file SC1_CohortProfile_D_9.0.0.dta refers to the *CohortProfile* data of *Starting Cohort 1* in its *Download* version of the Scientific Use File release *9.0.0*.

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 12. 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.

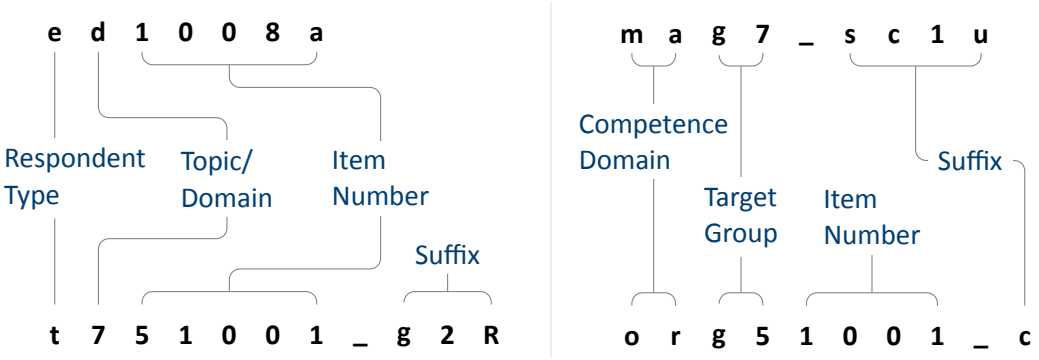

Figure 12: General variable naming (left) and competence variable naming (right)

3.2.1 Conventions for general variable naming

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

Table 4: Conventions for variable names

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

(...)

Table 4: (continued)

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

Suffixes

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

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

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

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

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

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

3.2.2 Conventions for competence variable naming

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

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

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

Table 5: Conventions for competence variable names

Part I: Competence Domain (2 chars)

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

(...)

Table 5: (continued)

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

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

Part III: Item number (3-4 chars)

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

Part IV: Suffixes (starting with an underscore)

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

Identification of repeated test items

In some competence measurements identical items are implemented in different testing waves (e. g., mathematics). Identifying repeatedly measured test items in NEPS data can be easily done by looking for competence variables with an identical word stem. If the same test item

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).

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 B.1 for hints on this.

In addition to the variable and value labels, the datasets also contain extended characteristics for variables. These include the question text from the survey instrument, 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 summarized in Table 6 and described in more detail below.

tab:MissingCodes

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.

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)
–91	survey aborted	respondent quit interview, in CAWI mode
–92	question erroneously not asked	question not asked by mistake, in CAWI and CATI
–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 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.

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

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

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

Derived scales and classifications

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

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

Figure 13 shows the derivation paths for several occupational scales and schemes provided in the NEPS. A detailed description of the standard derivations for educational attainment (ISCED, CASMIN and Years of Education) can be found in the corresponding documetation report by Zielonka and Pelz, 2015.

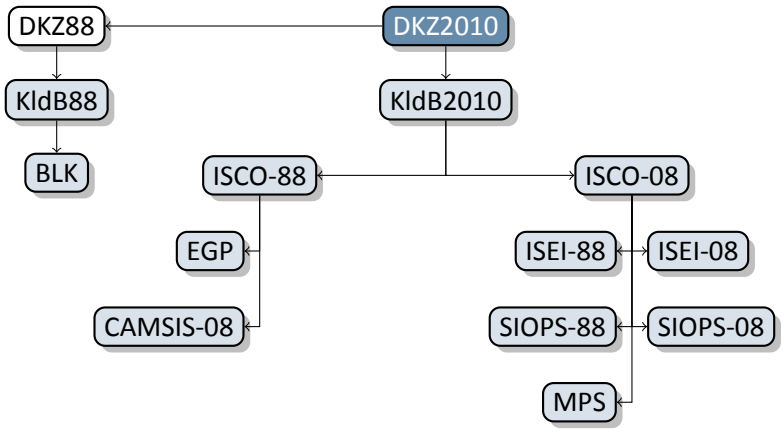

Figure 13: Derivation paths for several occupational scales and schemes provided in the NEPS

4 Data Structure

4.1 Overview

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

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

This convention does not apply to all longitudinal data. For example, there are competence measurements that were repeatedly carried out with the same target persons. However, since the instruments, i.e. the content of competence tests, vary over time, the corresponding information is structured in wide format (for more details, see section 3.2.2 or section 4.5.23). Such cross-sectionally structured data files with one line representing information of a respondent from all waves are marked with a *x*.

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

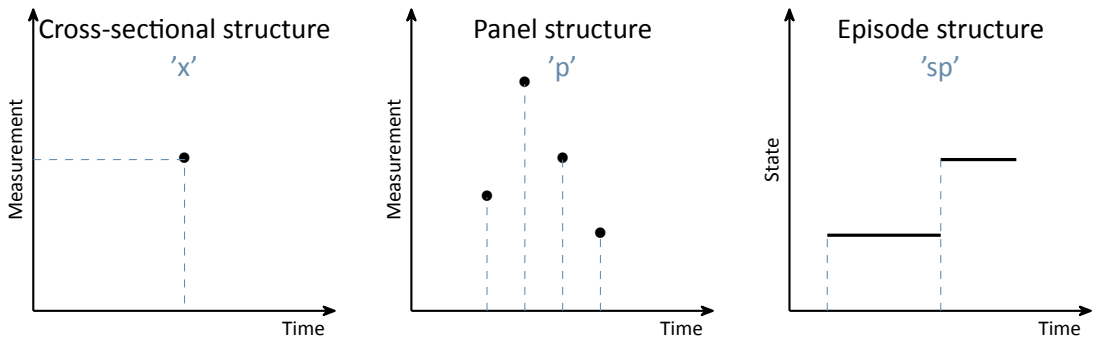

Figure 14: Different types of data structures

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

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

4.2 Identifiers

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

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

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

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

4.3 Panel data

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

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

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

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

4.4 Episode or spell data

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

In episode data, there is one row for each episode that was captured during the interview. Usually, a start date and an end date describe the duration of an episode. The remaining variables in such spell datasets contain additional information about that episode. These characteristics are chronologically linked to the episode. This means (especially for time-variant variables like ISEI or CASMIN) that the respective values indicate the status *at the given time of the episode*, and not necessarily the current status which is valid nowadays.

To give an example: In the spell dataset `spEmp` there is a period of time for a certain respondent in which he or she worked without interruption in a particular job. If this person changes to a new job, this marks a new episode which is stored in a new data row. Further changes in this context may also lead to new episodes, e. g., a change of employer or the conclusion of a new employment contract (but not if the salary, working hours or other characteristics of the

respective job change). Episodes can therefore be understood as the smallest possible units of one's life history, in this case the employment biography. As soon as there are several relevant changes in such a biography episode, this is reflected in a new data row.

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

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

Regardless of the type of episode, two variables are usually necessary to identify a single row in the data file, namely the respondents' identifier `ID_t` and an episode, event or entity numerator, such as `spell` or `child`. More detailed information on the required identifier variables can be found on the respective data file pages in in section 4.5. Please also note section 4.4.3 for a further complication of this matter.

One general remark: be aware that the number of episodes per se is independent of the survey wave. During one interview (one wave), there may be several episodes (several rows) recorded, or no episode at all. Also, the dates given in the episode relate to the time the episode was valid, whereas the wave relates to the interview date. They might not even overlap!
You should consider those two entities (`spell` and `wave`) as completely unrelated. Although there might be some situations where you have the need to know *when* the information of an episode has been collected, you are best advised to ignore the variable `wave` in episode data completely.

Do not try to use the variable `wave` to merge episode data to panel data. Although this might seem like the proper way to do this, episode data may contain multiple (or none) rows per wave and ID, while panel data contain exactly one row for every wave (of an ID). Such a merge results in the panel data obtaining an episode structure, which totally messes up the data.
A better approach seems prior to conduct such a merge, try to aggregate the episode data to *one information* for each interview date, or even just one information for the whole life course, so that in the end, you do not have more rows than waves (per respondent).

4.4.1 Edition of the life course

The life course data in the NEPS Starting Cohorts consists primarily of information on episodes of school attendance, participation in vocational preparation measures and vocational training or university education. Further it consists of information on exercise of compulsory or voluntary services (military module), employment and unemployment episodes, as well as spells of parental leave. We refer to these activities as *main activities*.

The episodes, grouped by episode type, are recorded independently in separate modules. The aim of recording these activities is to obtain chronologically complete life histories on the employment and training careers of our respondents. This requires two different edition steps of the data:

1. After the episodes have been collected in the longitudinal modules, the first step in the edition process of the life courses already takes place during the interview. The episodes are summarized in the data revision module and put into their chronological order. Subsequently, they are checked for chronological gaps and overlaps. This test is carried out by a cooperative clarification of chronological gaps between interviewee and interviewer.

   If chronological gaps are discovered in the data revision module, these are closed by subsequently recording additional episodes of the above-mentioned *main activities*. If there is no main activity for the examined period, the interviewee can close it with a so-called gap activity . In addition, gaps can be closed in the data revision module by adjusting the dates of the episodes between which a gap exists.

   Chronological overlaps of episodes are discussed in the data revision module together with the interviewee. This may lead to a change of the dates of the episodes involved in the overlap. For inaccurate or missing dates, estimates are calculated in addition to the original dates, as far as there are reasonable indications for good estimates. For example, the imprecise specification about the starting month of an episode “Summer” is replaced by the value 7 “July” and saved in the biography file. In this way, even episodes with incomplete date specifications can be included in the chronological test and checked for gaps or overlaps with temporally adjacent episodes in the overall context of the life course (for general and specific functionality of the data revision module, see Ruland et al., 2016 and Matthes et al. 2005, 2007).

   The result of this examination of the life courses during the interview are largely complete and time-consistent life courses.

2. Despite this meticulous examination during the interview, there are still minor inaccuracies in the consistency of life courses after the survey. For example, one-month overlaps of episodes are not edited in the data revision module. The same applies to gaps between successive episodes of up to two months. The test in the data revision module can also be interrupted or skipped at the request of the interviewee so that it was not carried out or not carried out completely.

   For these reasons, a second, automated step of processing the time data of the life courses takes place after the end of the interview during data edition. The results of these temporal adjustments are also saved in the biography file. The automated edition is divided into several successive edition steps.

   The first step is to remove one-month overlaps of episodes. A one-month overlap between two episodes is, in our definition, when the end date of a preceding episode is identical to the start date of the following episode. The procedure here is to shorten the end date of the previous episode by one month. The prerequisite is that the previous episode is longer than

one month, otherwise this one-month episode would be shortened to the duration of zero. If the duration of the previous episode is only one month, the start date of the following episode is shortened by one month. If both episodes have a duration of one month, the dates are not edited.

Subsequently, one to two-month gaps between successive episodes are automatically closed. If the gap has a duration of one month, the end date of the previous episode is extended by one month. If there is a two-month gap, the start date of the following episode is additionally brought forward by one month.

Finally, chronological gaps in the life course that are larger than two months are closed by inserting new episodes for these gaps in the biography file, which close these gaps completely. These episodes are marked as *data edition gap* in the `sptype` variable of the biography file.

All these changes of the time specifications described are exclusively made in the biography file. The respondents' original information on the start and end dates of the episodes remain in the data files of the longitudinal modules (also see Künster 2015a, 2015b).

4.4.2 Revoked episodes

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

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

4.4.3 Subspells and harmonization of episodes

There is one important circumstance to consider when working with NEPS spell data. Biographical episode data are collected retrospectively. During an interview, the respondents are asked about all episodes that have occurred since the last interview (in the first interview it is since

birth or a certain age). If an episode is finished at the time of the interview, the respondent reports a corresponding end date and the spell is completed. Difficulties arise when the episode is not yet finished at the time of the interview, i.e. it is still *ongoing*.

Such an episode appears as right-censored in the dataset. In the next interview, this episode is then queried using preloads in the course of *dependent interviewing* in such a way that the respondent can report whether it has been finished in the meantime or whether it continues. Technically this leads to several rows in the data structure, which can be distinguished by the variable `subspell`:

- first (right-censored) data row reported in initial wave (`subspell=0` if this is the only subspell for the episode, `subspell=1` if there are other subspells)
- continued episode reported in next wave(s) (`subspell=2`, `subspell=3`, etc.)

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

Episodes are defined by the assignment to a respondent (`ID_t`), by the type of episode (e.g., training episode), by an episode ID (`splink`, which typically consecutively numbers the episodes of the same type of episode of a case), and by the start and end date of the respective episode.

If an episode both begins and ends within the data collection period of a survey wave, then it can be assumed that this episode has been completely recorded with all the desired information (see figure 15, spell 1). In the SUF data of the corresponding longitudinal data file, there is a single data line for this episode, which contains the complete information.

However, there are many episodes that have not yet ended at the time of the interview of a survey wave, but are still ongoing at that time. Such persistent episodes are updated in the subsequent survey wave in which the respective person takes part. This means that further information on these episodes is recorded in the subsequent survey waves until the respondents report the episodes as finished (see figure 15, spell 2). In such cases, the information on an episode is stored separately in the SUF in one data line for each survey wave, so that the information on this episode is divided over several data lines and one data line contains only part of the information on this episode. Here, the person ID is identical in each data line of this episode, as well as the episode ID. There is, however, an additional variable `subspell`, which consecutively numbers the data lines that belong to one episode that was recorded over several survey waves (starting with value 1). For episodes that were recorded completely within one survey wave, i.e., those that began and ended during the period covered by the survey wave, the variable `subspell` contains the value 0. The same applies to episodes that were recorded for the first time in the current survey wave and that were still ongoing at the time of the interview (see figure 15, spell 3).

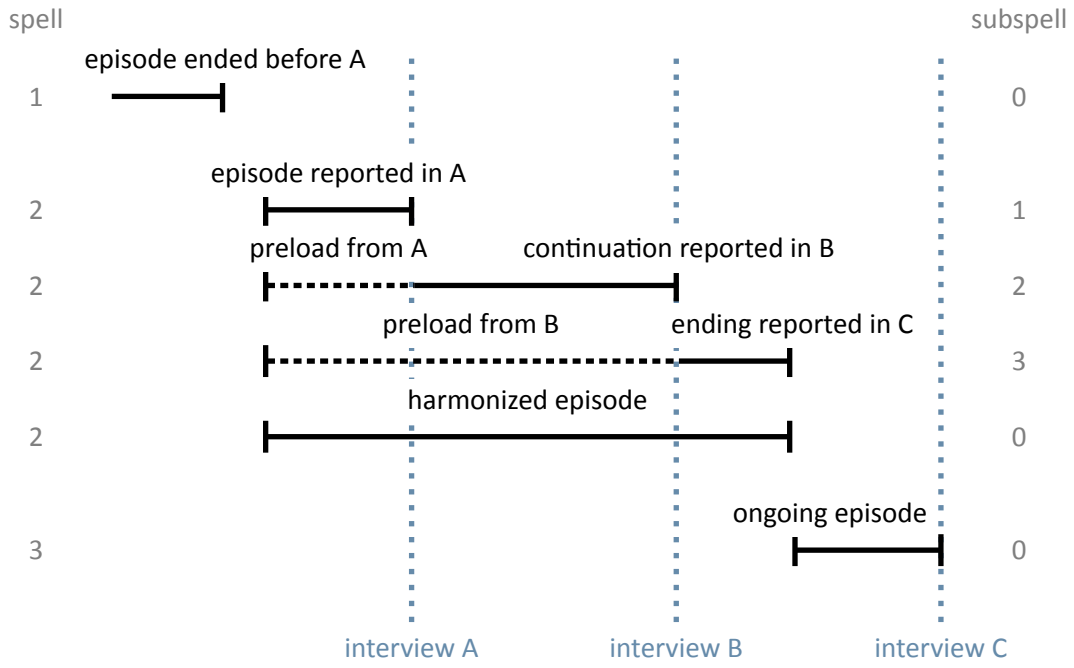

Figure 15: Logic of subspells

The episode file for the cases shown in figure 15 corresponds to the data structure listed in table 7 before an episode harmonization. For the sake of simplification, the table contains data from three consecutive surveys/waves, each conducted in december of the years 2009-2011.
There is only one row of data for the first episode of the example case, because it was completed before the survey time of wave 2, i.e., it was completely recorded in this wave. Accordingly, the value of `subspell` is 0.
For the second episode, there are three data lines with the information on this episode from waves 2-4. The `subspell` variable for the second episode numbers the partial episodes from 1-3. The end of the second episode was reported in the fourth survey wave.
The third episode was recorded in the fourth survey wave. This episode continues, but since only a part of the episode has been reported so far, `subspell` is also initially given the value 0. This does not change until further information for this episode is recorded in a subsequent survey wave.

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

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

For episodes that last over several survey waves, the NEPS does not collect the same informa-

tion in each survey wave. In the wave in which an episode is recorded for the first time, all unchangeable core information about this episode is collected. In the case of training episodes, this includes the type of training (e.g., vocational training or studies), the exact designation of the training occupation and some other parameters that distinguish this training from other training. This of course also includes the start date of the episode. This information will not be requested again when this episode is updated in later survey waves. Instead, additional characteristics of the episode, such as current pay, are recorded in these waves. As soon as the interviewee reports the episode as completed, information regarding the end is recorded. Such information is, for example, the achieved completion of a training episode and of course the end date of the episode. In this respect, the information on an episode, which was updated via different survey waves, is divided over the individual partial episodes (subspells) of this episode. The number of the partial episodes varies depending on the total duration of the episode.

In order to make it easier for data users to work with the data of updated episodes, the information from the partial spells of episodes is summarized in an additional data line. Therefore, besides the data lines for the partial episodes, there is also a data line that gives an overall overview of the updated episode and is referred to as the *harmonized episode*.
Thus, episode harmonization is only used if there are several partial spells for an updated episode from different survey waves.

The data line for the harmonized episode is added to the already available records in the longitudinal file. The variable `subspell` always has the value 0 for harmonized episodes. In our example case shown above, an additional data line would be added for the second episode as a summary of the three partial episodes of this episode in the longitudinal file (see table 8), since only the second episode has several partial spells in different survey waves.

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

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

Since a harmonized spell is a summary of all partial spells of an updated episode, exactly one piece of information must be selected from the partial spells for each variable, which is then transferred to the harmonized spell. In most cases the rule for selecting the relevant information that is transmitted is obvious. But if it is not, the following rules are applied:

first For all questions that are only asked when a new episode is entered, i.e., when the episode is reported for the first time, the information for the harmonized spell is taken from the first partial episode because it can only be found there and is valid for the complete duration of the episode (see `var1` in table 8).

last For information that is either updated in every survey wave or that can only be found in the last partial spell after the end of the episode, the information for the harmonized spell is

taken from the last partial episode (see `var2` in table 8).
There is an exception concerning the application of the harmonization rule *last*. If an already established question in the longitudinal modules is generally not asked in a certain survey wave, then the undetermined value of the associated variable is replaced with the value -54 *missing by design* during data edition. The reasons for not asking the question can be manifold. If this question follows the harmonization rule *last*, the value -54 is not stored into the harmonized episode. Instead, the existing partial episodes of the episode concerned are searched for a value that deviates from the value -54 and this value is stored in the harmonized episode. The same procedure is used with the value -55 *not applicable*. The idea is that the value determined in this way is a good estimate of the missing last information on this item of this episode.

first nonmissing The harmonization of most of the variables follows either the selection rule *first* or *last*. However, there are exceptions to this rule. An exception occurs, for example, if a new variable is introduced when recording episodes, which basically follows the *first* rule, but which should also be collected for episodes updated in the current survey wave. In such cases, the information on this variable is then also contained in the data for updated episodes, but is not in the first partial spell, but in a later partial episode. In these cases, the first valid value to be found in any partial spell of an episode is selected.

last nonmissing There is a similar exception for variables that measure a changing state until a target state is reached. In the case of employment episodes, this can be, for example, changing from a fixed-term position of a specific job to a permanent one. In cases in which an employment is temporary when it is first recorded, the question about the time limitation of the position is asked each time the episode extends over several survey waves. This continues until the employment either ends or the state of employment changes to *permanent*. Once this change from fixed-term to permanent job has been completed, the question of a time limitation is no longer asked when the episode is updated, since the reverse change from a permanent to a fixed-term job within the same job is hardly considered realistic. The information about the delimitation of the episode is therefore not necessarily in the first or last part of the spell. Here the last valid value of a partial spell of this episode is relevant. Therefore, in this case, the *last nonmissing* rule (last valid value to be found in the partial spells of an episode) is used for harmonization.

There is another exception in cases in which the continuation of an episode in the current survey wave is contradicted by the respondent during the life course assessment in the data revision module (see Ruland et al., 2016 for more information). This exception only affects episode types that are included in the life course assessment in the data revision module (episodes from the data files spSchool, spVocPrep, spVocTrain, spMilitary, spEmp, spUnemp, spParLeave, spGap). In such cases, we assume that the partial spells recorded in previous waves of the survey contain correct information on this episode up to the part of the episode that was contradicted, because they were subjected to a life course revision carried out together with the respondent in the previous waves of the survey. According to this logic only the part of the episode recorded in the current survey wave is contradicted by the respondent and not the complete episode. The information already collected and stored in a data line on the current partial spell (which was

contradicted in the data revision module) can still be found in the longitudinal file, but is marked in the variable spms with the code -20 as *episode canceled in the data revision module*. During the harmonization, this cancellation has been considered by only filling the harmonized episode with values from the partial spells that are not marked as canceled, i.e., all partial spells except for the contradicted partial spell of the current survey wave. The end date of this episode is set to the interview time of the survey wave in which the last, uncontradicted information on this episode was recorded.

Coded occupational information is recoded in the harmonized episodes based on the information available there. Therefore, there may be differences between the values of the partial episodes and the harmonized episodes for these generated variables.

The Research Data Center keeps track on which harmonization rule was applied to variables of the longitudinal data for which episodes were updated across survey waves. Those harmonization tables are currently not publicized, but you can obtain the rules for specific variables upon request.

Data users can decide whether they want to use the harmonized spells for data analysis or whether the information from the subspells that reflects the changes in characteristics of these episodes over time is important to them. Both information are available in the longitudinal data files.

If the harmonized episodes are to be used, including the episodes that only consist of a single partial spell and therefore did not have to be harmonized, then it is sufficient to select all records for which the value of the variable subspell is 0.

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

Thereafter, all episodes should be excluded that were contradicted in the data revision module (variable spms == -20) and which at the same time do not belong to the harmonized episodes (variable spext == 0)[2]. As described above, this step already has been included in the harmonization process for the harmonized episodes.

If, on the other hand, you do not want to use the harmonized episodes but the original partial spells of the episodes, then all records should be dropped where the variable subspell has the value 0 and simultaneously the variable spext has the value 1. Subsequently, it is also necessary to exclude all partial episodes that were contradicted in the data revision module (variable spms == -20).

4.5 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

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

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

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

```
** Starting Cohort
global cohort SC1
** version of this Scientific Use File
global version 9-0-0
** path where the data can be found on your local machine
global datapath Z:/Data/${cohort}/${version}
```

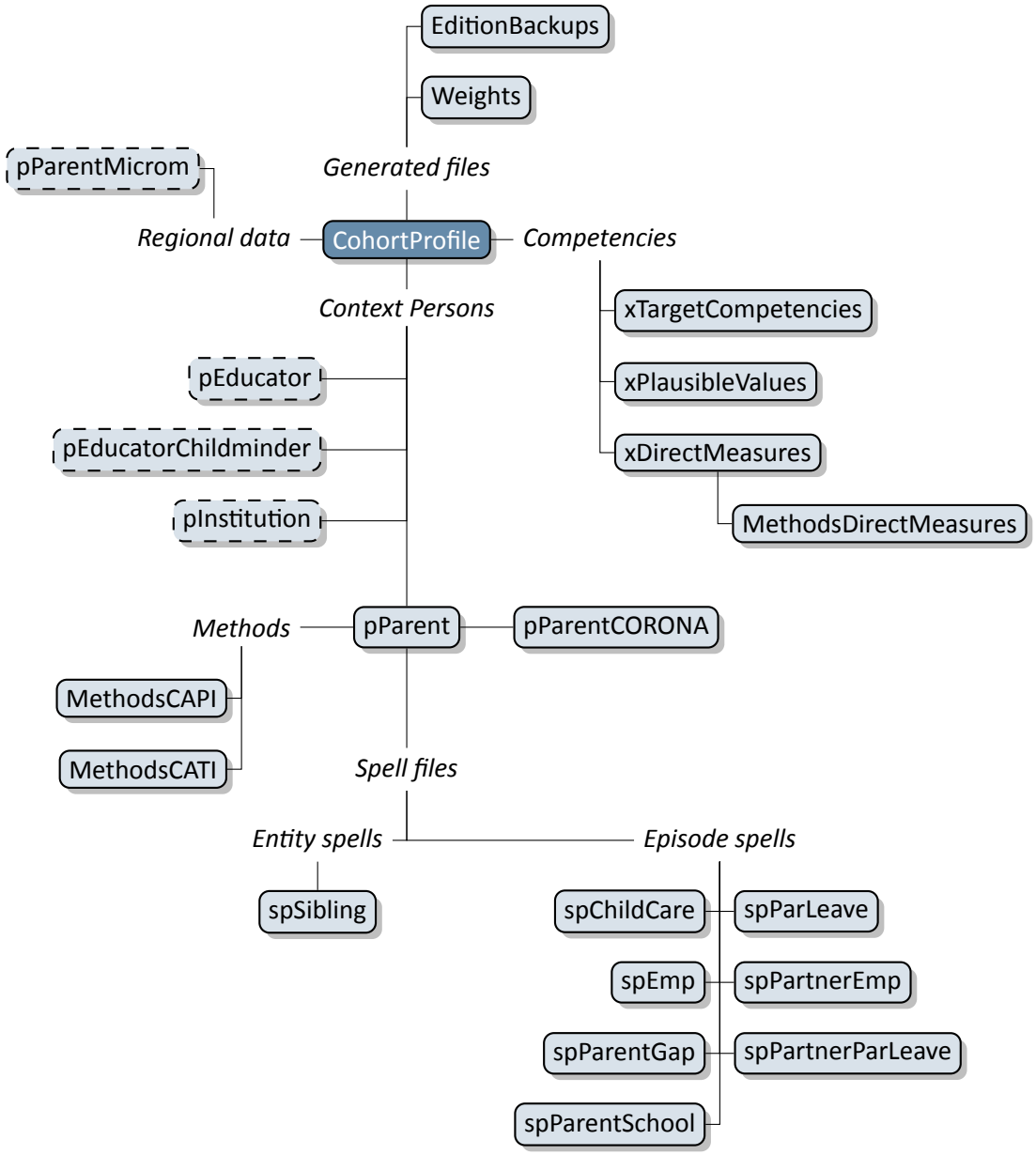

Figure 16: Graphical overview of all data files. Each node represents one data file. Relations are indicated by connection lines. Files with a dashed border are not available in the Download version of the Scientific Use File. Click on a data file to get more information.

4.5.1 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

Other ID variables useful for linkage

none

Number of variables / number of rows in file

17 / 31,329

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
cohort	NEPS Starting Cohort
tx80220	Participation/drop-out status
tx80521	Data available: survey target person
tx80522	Data available: competence test target person
tx80524	Data available: institution
tx80530	Data available: direct measure
tx80107	Sample: first participation in wave

Exemplary data snapshot

ID_t	wave	tx80220	tx80521	tx80522	tx80524
8055316	7	Participation	yes	yes	yes
8057289	5	Participation	yes	yes	yes
8060798	7	Participation	yes	yes	yes
8063788	6	Participation	yes	yes	yes
8066993	7	Participation	yes	yes	yes

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
```

4.5.2 EditionBackups

« go back to overview

Description

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

File structure

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

ID variables needed to identify a single row

dataset varname ID_t wave splink subspell partner child

Other ID variables useful for linkage

mergevars

Number of variables / number of rows in file

12 / 33,560

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

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

Exemplary data snapshot

ID_t	wave	dataset	varname	mergevars	sourcevalue_num	editvalue_num
8056430	2	pEducatorChildminder	ea2401d	ID_t wave	1.00	2.00
8058942	2	pEducatorChildminder	ea2401a	ID_t wave	1.00	2.00
8062978	2	pEducatorChildminder	ea2401c	ID_t wave	1.00	2.00
8062978	2	pEducatorChildminder	ea2401a	ID_t wave	1.00	2.00
8069316	2	pEducatorChildminder	ea2401a	ID_t wave	1.00	2.00

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

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

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

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

```
** In this example, we want to restore the original
** values in variable p731813 "(Highest) professional qualification respondent"

** open the datafile
use ${datapath}/SC1_EditionBackups_D_${version}.dta, clear

** only keep rows containing data of the aforesaid variable
keep if dataset=="pParent" & varname=="p731813"

** check which variables we need for merging
tab mergevars

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

** rename the variables to emphasize affiliation
rename sourcevalue_num p731813_source
rename editvalue_num p731813_edit

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

** open pParent
use ${datapath}/SC1_pParent_D_${version}.dta, clear

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

** check all edition made
list ID_t wave p731813* if _merge==3

** replace the variable in the datafile with its original value
replace p731813=p731813_source if _merge==3
```

4.5.3 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

Number of variables / number of rows in file

86 / 24,474

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
ID_int	Interviewer: ID
wave	Wave
px80305	Interviewer: own children
px80302	Interviewer: age group
px80209	Interview: length of interview (minutes)
px80222	Final outcome - repondent
px80400	Willingness: panel participation
px80210	Interview: incentive (euros)
px80301	Interviewer: gender
px80331	Interviewer: migrant background
px80321	Fatigue TP

Exemplary data snapshot

ID_t	ID_int	wave	px80302	px80209	px80301
8057010	1858	2	50-65 years	32.50000	2
8065709	1373	1	50-65 years	122.61667	2
8065817	2090	1	50-65 years	115.98333	2
8066922	2058	1	50-65 years	90.85000	2
8067506	2093	1	50-65 years	144.00000	2

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 3 (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
```

4.5.4 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

Number of variables / number of rows in file

26 / 3,431

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
ID_int	Interviewer: ID
wave	Wave
px80209	Interview: length of interview (minutes)
px80210	Interview: incentive (euros)
px80301	Interviewer: gender
px80302	Interviewer: age group
px80303	Interviewer: highest school-leaving qualification
px80400	Willingness: panel participation

Exemplary data snapshot

ID_t	ID_int	wave	px80209	px80301	px80302
8056286	1102	2	27.10000	2	30-49 years
8059612	1036	2	23.03333	1	30-49 years
8063092	2019	2	30.28333	1	30-49 years
8064230	1108	2	26.98333	2	up to 29 years
8066645	1846	2	40.40833	1	30-49 years

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 4 (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
```

4.5.5 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

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

182 / 20,264

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
px03001	Informed consent Sensomotoric development
px03002	Performance Sensomotoric development
px04021	No implementation of direct measure: data protection
px04025	No implementation of direct measure: child not in front of laptop
px05011	Informed consent Vocabulary
px05041	Informed consent Flanker
px05013	Selection Vocabulary
px04049	Video recording Tasks
px04071	Child experience with touch screen yes/no

Exemplary data snapshot

ID_t	wave	px03002	px04021	px04025
8062199	1	Competence test 3 Sensomotoric development not started	1	0
8064059	1	Competence test 3 Sensomotoric development not started	1	0
8066229	1	Competence test 3 Sensomotoric development not started	1	0
8067509	1	Competence test 3 Sensomotoric development not started	1	0
8067827	1	Competence test 3 Sensomotoric development not started	1	0

`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 5 (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
```

4.5.6 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

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

412 / 3,102

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
e209102	Daycare educator: Professional qualification
ea0301a	Daycare: sponsor
e209107	Daycare: parents' initiative
e209101	Daycare: free places
e217511	Daycare institution Group: 2015; number children, total
e400000	Migrant background of youth / childcare worker
e41100a_g1	Mother tongue of educator (number responses)
e66805a	Temperament - frustration
e66805b	Temperament - is concentrated

Exemplary data snapshot

ID_t	wave	e209107	e217511	e400000	e41100a_g1
8057892	2	no	-54	3	1
8060626	2	no	-54	3	1
8064672	4	Missing by design	25	3	1
8067951	4	Missing by design	10	3	1
8068227	3	no	-54	3	1

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 6 (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
```

4.5.7 pEducatorChildminder

« go back to overview

Description

Context data collected from childminders

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

236 / 183

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
e208128	Daycare: year of birth 2011; number of children
ea20010	Daycare: number of children cared for
e767110	Gender Caregiver
ea25020	Daycare: own children number
ea2701a	Daycare: further training
e40000a_g1	Country of origin of daycare worker
ea31010	Member daycare association
ea3001a	Daycare: supervision
e400000	Migration background of the respondent

Exemplary data snapshot

ID_t	wave	e208128	ea20010	e767110	ea25020	e40000a_g1
8056430	2	2	2	[w] female	3	Unspecific missing
8063499	2	1	2	[w] female	2	Unspecific missing
8063567	2	2	2	[w] female	2	Unspecific missing
8065742	2	2	2	[w] female	1	Unspecific missing
8066507	2	2	2	[w] female	2	Unspecific missing

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 7 (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
```

4.5.8 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

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

165 / 2,079

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
h217001	Institution: registered girls
h217002	Institution: registered boys
h451020	Institution: Number Children with migrant background
h217200	Daycare: number of children with a disability
h534010	Kindergartens within 5 km
h219001	Institution: free places
h219301	Institution: staff, educators, number of persons
hb10030	Existence of a special offer: language support offer
h539013	Admission criteria: proximity to the place of residence

Exemplary data snapshot

ID_t	wave	h217001	h217002	h534010	h219001	h219301
8055731	4	8	16	2	6	3
8056165	4	8	10	5	2	4
8064747	5	24	24	9	5	7
8066417	6	60	59	10	1	15
8066962	7	28	39	5	1	6

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 8 (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
```

4.5.9 pParent

« go back to overview

Description

Data surveyed from parents (usually mothers)

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

2,630 / 22,041

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
p406010_g1R	Country of birth Target child
p731905	Professional position Respondent
p731955	Professional position Partner
p731701	Relationship to target child
p741001	Household size
p510005	Monthly household income, open
p400500_g1	Generation status
p743040	TC in HH
p731905	Professional position Respondent
p34009d	Participation in high culture: theater
p73170y	Date of birth respondent: year
p401100	Citizenship Respondent
p731116	Gender partner

Exemplary data snapshot

ID_t	wave	p731905	p731955	p741001	p400500_g1	p743040
8056321	4	1	2	5	3	yes
8059728	4	2	1	5	3	yes
8063311	7	2	2	3	9	yes
8068680	7	2	2	4	3	yes
8068704	4	2	2	4	6	yes

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 9 (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
```

4.5.10 pParentCORONA

« go back to overview

Description

Data regarding the impact of the corona pandemic on respondents life

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

none

Number of variables / number of rows in file

86 / 1,814

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
p515051	Willingness to take risks in general
pm00015	Systemically important profession
pm00016	Change Working time
pm00017	Change Work place
pm00018	Change Status
pm00019	Support Employer
pm00020	Change Order situation
pm00051	Homeschooling, equipment

Exemplary data snapshot

ID_t	wave	p515051	pm00018	pm00019
8056215	9	8	Release from work duties with continued wage payment	very well supported
8063928	9	4	Instructed reduction of vacation/overtime	well supported
8066281	9	4	Instructed reduction of vacation/overtime	well supported
8066956	9	6	Release from work duties with continued wage payment	reasonably well supported
8068162	9	2	Release from work duties with continued wage payment	very well supported

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

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

Data is collected by means of a cross-cohort questionnaire program adapted to the current situation of the respective participants. In order to collect this data in a timely manner, the first questions were administered via online survey in Starting Cohorts 2-6 in May 2020. As this time span did not overlap with regular waves, data from this survey is marked with a missing wave

(`wave==.`). The integration of the corresponding questions is planned in an additional module on the corona pandemic for the forthcoming main surveys in all Starting Cohorts.

Example 10 (Stata): Working with pParentCORONA

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

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

4.5.11 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

Number of variables / number of rows in file

188 / 50,960

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

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

Exemplary data snapshot

ID_t	wave	regio	ID_regio	mso_k_ausland	mbe_k_haustyp	mpi_k_dichte
8055267	3	1	150636	6	1	3
8057067	2	1	118920	1	2	7
8062846	1	1	100053	4	3	3
8067293	2	1	100269	7	4	4
8068174	2	1	119795	6	5	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 11 (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
```

4.5.12 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

Number of variables / number of rows in file

28 / 21,361

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave	Wave
sptype	Childcare: Episode type
spell	Spell number
pa0112y	Childcare: start (year)
pa0113y	Childcare: end (year)
pa01270	Qualification childminder
p412600	Interaction language
	Grandparents - child
pa01510	Grandparents episode number
pa01140	Later institution care
pa01240	Childminder childcare later
pa01340	Nanny childcare later
pa01440	Au pair childcare later
pa01540	Later grandparents childcare
pa01640	Relatives childcare later

Exemplary data snapshot

ID_t	wave	sptype	spell	pa0112y	pa0113y	pa01510
8056326	5	5	501	2015	2016	1
8061337	2	5	201	2012	2013	1
8062361	4	5	401	2013	2015	1
8065138	6	5	601	2016	2017	1
8065919	4	5	401	2014	2015	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 12 (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
```

4.5.13 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

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

24 / 6,491

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
subspell	Number of subspell
spell	Spell number
p731504	Type of job
p73159y	Start employment episode: year
p73158y	End employment episode: year
p73158c	Duration of employment episode
p731505	Working hours 12 months prior to birth Respondent
p731506	Working hours at start Respondent
p731509	Working hours upon respondent taking parental leave
p731511	Working hours upon partner taking parental leave
p731512	other vocational activities

Exemplary data snapshot

ID_t	subspell	spell	p731504	p731505	p731509
8067951	1	1	2	28	20
8064484	1	1	3	40	30
8061236	1	1	2	40	4
8055120	1	3	5	10	15
8059730	1	1	1	40	20

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 13 (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)
```

4.5.14 spParentGap

« go back to overview

Description

gap episodes reported by the parents

File structure

spell format: 1 row = 1 gap of 1 respondent

ID variables needed to identify a single row

ID_t spell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

17 / 5

Contains data from waves

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

Exemplary variables

ID_t	ID target
splink	Link for spell merging
spell	Spell number
wave	Wave
ps29101	Type of gap episode
ps2911m	Start date Gap
ps2911y	Start date Gap
ps2912m	End date Gap
ps2912y	End date Gap
ps2912c	Ongoing of gap episode
ps2911y_g1	Check module: start date (year), corrected
spms	Check module: spell type

Exemplary data snapshot

ID_t	wave	ps2911m	ps2911y	ps2912m	ps2912y	ps2912c
8064281	9	7	2019	10	2019	.
8064318	9	6	2020	8	2020	1
8066386	9	2	2019	8	2019	.
8067260	9	7	2019	9	2019	.
8068603	9	3	2020	6	2020	.

The datafile `spParentGap` contains gaps in the individiual life courses of the target persons **reported by the parents during the parent CATI**. Note that this is not gaps in the lifecourse of the parent, but of the children! The spells in this file refer to different types of gaps that can be distinguished by the variable `ps29101` (Type of gap episode).

Example 14 (Stata): Working with spParentGap

```
** open the Gap data file
use ${datapath}/SC1_spParentGap_D_${version}.dta, clear

** get an overview about the type of gaps
tab ps29101
```

4.5.15 spParentSchool

« go back to overview

Description

general schooling history reported by the parents

File structure

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

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Number of variables / number of rows in file

35 / 5,641

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
splink	Link for spell merging
subspell	Number of subspell
spell	Spell number
wave	Wave
p723180	School authority
p72302m	End date School episode (month)
p72302y	End date School episode (year)
p723120	School: reason end school episode (child)
p723130	Reason Change of school
p723140	Reason Interruption of school

Exemplary data snapshot

ID_t	subspell	spell	wave	p72302m	p72302y
8055832	1	1	8	6	2019
8059769	1	1	8	5	2019
8061236	1	1	8	5	2019
8065048	1	1	8	5	2019
8065903	2	1	9	6	2020

This module covers each respondent's general education history from school entry until the date of (anticipated) completion, including

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

A new episode is generated only if the school type changes. That is, a change from one Gymnasium to another is not recorded. As a result, a single schooling episode may take place at more than one location. In such cases, only information on the last location is included. A new episode is generated at each school type change even if both schools offer the same certificate.

The data in this file is the school history reported by the parent during the parent CATI. See file `spSchool` for the school history reported by the target herself.

Example 15 (Stata): Working with spParentSchool

```
** open the data file
use ${datapath}/SC1_spParentSchool_D_${version}.dta, clear

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


** evaluate how many children have school episodes already
distinct ID_t

** check the distribution of the number of episodes per child
summarize spell

** generate an indicator if a child ever visited a public school (vs. church/private
  schools)
bysort ID_t: egen public =  max(p723180==1)

** create minimal dataset
keep ID_t public
duplicates drop
tempfile tmp
save `tmp'

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

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

** you now have an enhanced version of CohortProfile, enriched by
** information from the spell module.
```

4.5.16 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

Number of variables / number of rows in file

16 / 4,563

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
spell	Spell number
subspell	Number of subspell
wave	Wave
pa0403m	Respondent - Start parental leave: month
pa0403y	Respondent - Start parental leave: year
pa0404m	Respondent: End parental leave: month
pa0404y	Respondent: End parental leave: year
pa0404c	Duration of parental leave episode
pa04020	Respondent: Parental leave so far (yes/no)

Exemplary data snapshot

ID_t	spell	subspell	wave	pa0403m	pa0403y	pa0404m	pa0404y
8057736	1	2	4	7	2012	5	2014
8061175	1	1	2	8	2012	6	2013
8062876	1	2	4	4	2012	2	2015
8066018	1	1	2	4	2012	4	2013
8069649	1	2	4	3	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 16 (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
```

4.5.17 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

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

Number of variables / number of rows in file

24 / 7,428

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
spell	Spell number
subspell	Number of subspell
wave	Wave
p731604	Type of job partner
p73169y	Start employment episode partner: year
p73168y	End employment episode partner: year
p73168c	Duration of employment episode
p731605	Working hours Partner 12 months prior to birth
p731606	Working hours at start Partner
p731608	Change of partner's working hours upon respondent taking parental leave
p731607	Working hours at the end Partner
p731612	other employment partner

Exemplary data snapshot

ID_t	spell	wave	p731604	p73169y	p73168y	p731605
8064382	1	4	2	2011	2012	40
8067279	3	2	7	2008	2013	4
8067288	2	4	2	2011	2011	95
8067716	1	4	2	2007	2013	37
8068194	1	2	2	2008	2013	42

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 17 (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)
```

4.5.18 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

Number of variables / number of rows in file

15 / 2,095

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
spell	Spell number
subspell	Number of subspell
spgen	Generated spell
spext	Episode has subspells
spstat	last (sub-)spell status
wave	Wave
px80211	Survey/Test instrument
pa0503m	Partner - Start parental leave: month
pa0503y	Partner - Start parental leave: year
pa0504m	Partner: end parental leave: month
pa0504y	Partner: end parental leave: year
pa0504c	Duration of parental leave episode
disagint	Inconsistency in panel attachment
disagwave	Inconsistency in wave...

Exemplary data snapshot

ID_t	spell	subspell	wave	pa0503m	pa0503y	pa0504m	pa0504y
8062785	1	2	4	3	2013	5	2013
8064338	1	2	4	3	2013	5	2013
8064877	1	2	4	12	2012	8	2013
8068678	1	1	2	6	2012	6	2013
8069619	1	1	2	5	2013	6	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 18 (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
```

4.5.19 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

Number of variables / number of rows in file

34 / 5,367

Contains data from waves

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

Exemplary variables

ID_t	ID target
wave	Wave
p732107	Sibling lives with parents
p73221m	Date of birth sibling: month
p73221y	Date of birth sibling: year
p732220	Gender Sibling
p732230	Nature of relationship to siblings
p732313	Highest school-leaving qualification Sibling
p732314	Current training Sibling
p732315	Current civil servant training Sibling
p732316	Attended type of higher education institution Sibling
p732324	Doctorate Sibling
p732325	Type of civil servant training Sibling
p732401	Employment status Sibling
p732402	Unemployment Sibling

Exemplary data snapshot

ID_t	wave	p73221y	p732220	p732230
8058408	8	2008	1	half brother/half sister
8059154	8	2017	1	biological brother/biological sister
8060130	8	2010	2	biological brother/biological sister
8060430	8	2004	1	biological brother/biological sister
8068635	8	2013	2	biological brother/biological sister

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 19 (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
```

4.5.20 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

Number of variables / number of rows in file

41 / 3,481

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
w_t1	Cross-sectional weight participation in wave 1
w_t2	Cross-sectional weight participation in wave 2
w_t3	Cross-sectional weight participation in wave 3
w_t1comp	Cross-sectional weight participation in wave 1 (direct measures)
w_t2comp	Cross-sectional weight participation in wave 2 (direct measures)
w_t5comp	Cross-sectional weight participation in wave 5 (competences)

Exemplary data snapshot

ID_t	w_t1	w_t2	w_t1comp	w_t2comp
8061591	1.69605	1.58767	2.58779	2.80627
8068964	0.45873	0.40631	0.42539	0.38581
8061044	0.49640	0.46627	0.44520	0.41820
8064050	0.80874	0.80298	0.71392	0.80367
8058983	0.41765	0.41468	0.50550	0.56905

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 20 (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
```

4.5.21 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*

Number of variables / number of rows in file

344 / 3,203

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave_w1	Row contains data from wave 1
wave_w2	Row contains data from wave 2
wave_w3	Row contains data from wave 3
ihn1p001_c	Parent-child-interaction: sensitivity to distress (parent)
cdn1_sc1	Sensomotoric development: WLE
cdn1_sc2	Sensomotoric development: standard error (WLE)
cdn1c012_c	Sensorimotor development: thumb-finger grasp
hdn2c21t_p	Habituation: W2 DishabB2 - maximum fixation time on target 1
hdn2c22t_s	Habituation: W2 DishabB2 - total fixation time on target 2
hdn2c22t_n	Habituation: W2 DishabB2 - number of fixations on target 2

Exemplary data snapshot

ID_t	wave_w1	wave_w2	wave_w3	ihn1p001_c	cdn1_sc1	hdn2c21t_p
8056304	1	1	1	5	1.80	4467.33333
8066823	1	1	1	5	1.13	6500.33333
8065254	1	1	1	4	0.38	1733.33333
8067335	1	1	1	5	0.74	1966.33333
8062289	1	1	1	4	0.18	9767.33333

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 21 (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
```

4.5.22 xPlausibleValues

« go back to overview

Description

Plausible Values of competence data

File structure

wide format: 1 row = 1 respondent

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

wave_w*

Number of variables / number of rows in file

76 / 3,205

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

wave_w1	Row contains data from wave 1
wave_w5	Row contains data from wave 5
cdn1_pv1	Cognitive development: cross-sectional plausible value 1
man5_pv1	Math: cross-sectional plausible value 1
man5_pv1u	Math: longitudinal plausible value 1
scn6_pv1	Science: cross-sectional plausible value 1
man7_pv1	Math: cross-sectional plausible value 1
man7_pv1u	Math: longitudinal plausible value 1

Exemplary data snapshot

wave_w1	cdn1_pv1	man5_pv1	man5_pv1u	scn6_pv1	man7_pv1
1	0.88698	1.22118	1.29705	1.08322	0.08482
1	0.90822	1.34315	1.24878	0.96923	2.01000
1	0.23616	0.24945	0.90703	0.18893	0.34046
1	0.15279	1.76224	1.77097	1.40967	1.48146
1	2.66692	0.27684	0.46357	0.59333	1.06296

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

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

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

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

Example 22 (Stata): Working with xPlausibleValues

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

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

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

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

4.5.23 xTargetCompetencies

« go back to overview

Description

Competence data of respondents

File structure

wide format: 1 row = 1 target

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

wave_w*

Number of variables / number of rows in file

1,304 / 2,573

Contains data from waves

1 2 3 4 5 6 7 8 9

Exemplary variables

ID_t	ID target
wave_w4	Row contains data from wave 4
wave_w5	Row contains data from wave 5
den40002	Delay of gratification: waiting time
den40001_c	Delay of gratification: child has waited
can40001	SON categories Item 1
can40002	SON categories Item 2
von40001_c	Vocabulary: set 1, item 1
von40002_c	Vocabulary: set 1, item 2
von4_sc3	Vocabulary: sum
man5z17s_c	Mathematical competence Item 1
man5z021_c	Mathematical competence Item 2

Exemplary data snapshot

ID_t	wave_w4	wave_w5	den40002	man5z17s_c
8062112	1	1	181	3
8065223	1	1	181	3
8063003	1	1	181	1
8054996	1	1	181	3
8055120	1	1	181	3

The file `xTargetCompetencies` contains the data of the competence tests with the respondents. These are currently the domain-specific competencies vocabulary, mathematical competence, and scientific 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 23 (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 4.5. 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).

A References

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

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

FDZ-LIfBi. (2022). *Data Manual NEPS Starting Cohort 1– Newborns, Education from the Very Beginning, Scientific Use File Version 9.0.0*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

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

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

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

NEPS Network. (2022-a). *National Educational Panel Study, Scientific Use File of Starting Cohort Newborns*. Leibniz Institute for Educational Trajectories (LIfBi), Bamberg. https://doi.org/10.5157/NEPS:SC1:9.0.0

NEPS Network. (2022-b). *Starting Cohort 1: Newborns (SC1), Wave 9, Questionnaires (SUF Version 9.0.0)*. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

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

Scharl, A., Carstensen, C. H., & Gnambs, T. (2020). *Estimating Plausible Values with NEPS Data: An Example Using Reading Competence in Starting Cohort 6* (NEPS Survey Paper No. 10). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

B Appendix

B.1 R examples

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

Example 24 (R): Setting working directory

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

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

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

To import a data set, use:

Example 25 (R): Importing the data

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

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

Example 26 (R): Assigning variable labels

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

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

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

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

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

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

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

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

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

Example 27 (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 28 (R): Working with EditionBackups

```
'** In this example, we want to restore the original
** values in variable p731813 "(Highest) professional qualification respondent"'
```

```
'** import the data file'
EditionBackups =
  read.dta13("SC1_EditionBackups_D_6-0-0.dta",
             convert.factors = T)

'** only keep rows containing data of the variable mentioned above'
EditionBackups = subset(EditionBackups,
                        EditionBackups$dataset == "pParent" &
                          EditionBackups$varname == "p731813")

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


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


'** rename the variables to emphasize affiliation'
names(EditionBackups)[names(EditionBackups) == "sourcevalue_num"] = "p731813_source"
names(EditionBackups)[names(EditionBackups) == "editvalue_num"] = "p731813_edit"


'** open pParent'
pParent =
  read.dta13("SC1_pParent_D_6-0-0.dta",
             convert.factors = T)


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

```
  #the columns "source" in x and y are deleted
)


'** check all editions made'
View(subset(pParent[c("ID_t", "wave", "p731813", "p731813_source", "p731813_edit")],
            pParent$source == "both"))


'** replace the variable in the datafile with its original value'
for (i in 1:length(pParent$p731813)) {
  if(pParent$source[i] == "both"){
    pParentt520003[i] = pParent$p731813_source[i]
  }
}
```

Example 29 (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 30 (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 31 (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 32 (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 33 (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 34 (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 35 (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 36 (R): Working with pParentMicrom

```
'** open Microm datafile. Note that this data file is only available OnSite!'
pParentMicrom =
        read.dta13("SC1_pParentMicrom_O_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_O_5-0-0.dta", convert.factors = T)

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

Example 37 (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 38 (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 39 (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 40 (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 41 (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 42 (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 43 (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 44 (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 45 (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)
```

B.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 9.0.0
** (doi:10.5157/NEPS:SC1:9.0.0)
**
================================================


================================================
* Changes introduced to NEPS:SC1 by version 9.0.0 *
================================================

General:
        - new data from wave 9 have been incorporated into the Scientific Use File
        - meta information for all variables have been revised and updated where
            appropriate

pParent:
        - variables p731802_g1 "Highest educational qualification Respondent (ISCED)"
            and p731802_g2 "Highest
                educational qualification Respondent (CASMIN)" are suffering a minor
                    coding error.
                This has been fixed in the current release.


================================================
* Changes introduced to NEPS:SC1 by version 8.0.1 *
================================================

General:
        - meta information for all variables have been revised and updated where
            appropriate


pParent:
        - value labels for variable "Gender Target child" [p700010] had been flipped in
             version 8.0.0; this has been fixed
                (thanks to J. Lettau for reporting the issue)


================================================
* Changes introduced to NEPS:SC1 by version 8.0.0 *
================================================

General:
        - new data from wave 8 have been incorporated into the Scientific Use File
        - meta information for all variables have been revised and updated where
            appropriate
        - all variables relating to the dates of data collection (e.g. when the
            competency tests and cati-questionnaires took place)
                were updated and now stored centrally in the CohortProfile dataset (
                    variables tx86***). Variables intm and inty have
                been removed from all other datasets.
```

```
pParentCORONA:
        - a new dataset with information from an additional CAWI survey (May 2020) on
          Corona related topics has been incorporated
              in this SUF release.

xTargetCompetencies:
        - competency data from wave 7 have been updated to correct some minor errors
          and variable names

MethodsDirectMeasures:
        - variable names have been partially revised in order to better reflect
          similarities across waves


================================================
* Changes introduced to NEPS:SC1 by version 7.0.0 *
================================================

General:
        - new data from Wave 7 have been incorporated into the Scientific Use File
        - an updated Data Manual is published on the website
        - meta information for all variables have been revised and updated where
          appropriate

pParent:
        - information on the federal state (Bundesland) of the place of residence (
          Wohnort) is now available in the
              Download SUF; more fine-grained information on the administrative
                  district (Regierungsbezirk) and
              the county (Kreis) of the place of residence is available in the Onsite
                   SUF

spParentSchool:
        - information on the federal state (Bundesland) of the municipality of school (
          Gemeinde der Schule) is now
              available in the Download SUF; more fine-grained information on the
                  administrative district
              (Regierungsbezirk) and the county (Kreis) of the municipality of school
                   is available in the Onsite SUF

Weights:
        - information on the federal state (Bundesland) of the sample frame is now
          available in the Download SUF

xTargetCompetencies:
        - in the previous SUF version 6.0.0 there were erroneous names for the
          variables dsn4201 to dsn4001_sc9
              (Zahlenspanne / Digit span); this has been fixed

pEducator:
        - the labeling error in SUF version 6.0.0 for the generated variables e539020_g
          [1-6] has been fixed


================================================
* Changes introduced to NEPS:SC1 by version 6.0.0 *
================================================

General:
        - new data from Wave 6 have been incorporated into the Scientific Use File
```

```
        - an updated Data Manual is under final review and will soon be published on
          the website
        - meta data for all variables have been revised and updated where appropriate

xTargetCompetencies:
        - due to technical problems with data storage during testing, the mathematical
          competence data for 12 persons
              were set to missing (-95); WLE's and SD's have been updated
        - some competence variables have been renamed with respect to the target group
          element (change from kX to nX)

EditionBackups:
        - this new dataset has been incorporated into the Scientific Use File for the
          first time; it contains raw values
              before data edition (for more details see the updated Data Manual)


================================================
* 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]
```