NEPS
National Educational Panel Study

Research Data

FDZ-LIfBi

Data Manual

NEPS Starting Cohort 5—First-Year Students
From Higher Education to the Labor Market

Scientific Use File Version 10.0.0

LIfBi
LEIBNIZ INSTITUTE FOR
EDUCATIONAL TRAJECTORIES

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

Research Data Documentation

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

Full citation of this document:
FDZ-LIfBi. (2018). *Data Manual NEPS Starting Cohort 5 – First-Year Students, From Higher Education to the Labor Market, Scientific Use File Version 10.0.0*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

This release of Scientific Use Data from Starting Cohort 5—First-Year Students "From Higher Education to the Labor Market" was prepared by the staff of the Research Data Center at Leibniz Institute for Educational Trajectories (Forschungsdatenzentrum, FDZ-LIfBi). It represents a major collaborative effort. *The contribution of the following persons is gratefully acknowledged:*

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

For their support in writing this manual, special thanks go to:
Annika Grieb, Marie Kühn, Uta Liebeskind (DZHW Hannover)

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

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

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

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

Contents

1 Introduction

1.1 About this manual

This manual is intended to facilitate your work with data of NEPS Starting Cohort 5—First-Year Students (NEPS SC5). 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 materials, requirements for data access, rules of data citation, some general rules and recommendations, and selected user services. In the second chapter, Starting Cohort 5 and its sampling strategy are briefly introduced. The main part is subject to the sample development across the waves including field times, realized case numbers, survey mode, and the measurement of competency domains. 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. These portraits include recommendations on how to use the dataset and 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 5.

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 data 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 text adjustments or 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 First-Year Students` > `Documentation`

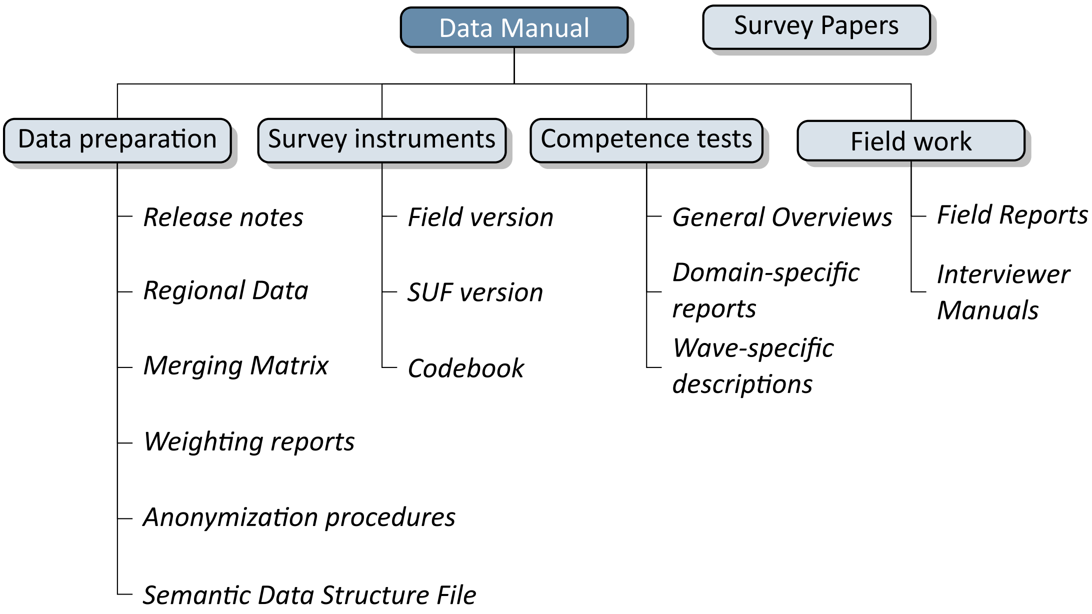


Figure 1: NEPS supplementary data documentation

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

Merging Matrix The Merging Matrix provides an overview of how to link information from different datasets noting the respective relevant identificator variables.

Weighting reports Reports of weighting and sample stratification include information regarding the construct principles and sampling process.

Anonymization procedures This document describes the anonymization procedures of the respective data. Here you are also given 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 scheme options to be used for exploring the data structure or for preparing analyses.

Survey Instruments For each wave, the survey instruments are offered in the form of 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 tests are not publicly available*.

Codebook The codebook lists all variables and their corresponding labels plus the basic frequencies by waves in concordance to 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. All information about survey preparation, interviewer deployment, respondent tracking, initial contacts, incentives, and sample realization are available in German only.

Interviewer Manuals The interview manuals is the basis for the interviewers' training before the computer-assisted interviews were conducted. In particular, it describes the interview process as well as the content of each of the questionnaire modules.

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`

Other Additional documentation 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 panel and episode or spell information (see section 4). The release of NEPS Scientific Use Files follows a cumulative strategy, i. e., the latest release replaces all former releases completely. 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 DOI

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 a 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 names: that of the Scientific Use File, its data files (see Table 3), and the respective DOI.

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 their utilized NEPS data in an easy and precise way (see below), which in turn is the 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:SC5:10.0.0`. Other releases of Scientific Use Files for Starting Cohort 5 can be addressed 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 5

SUF Version	DOI	Date of release
10.0.0 (current)	`doi:10.5157/NEPS:SC5:10.0.0`	**2018-04-19**
9.0.0	`doi:10.5157/NEPS:SC5:9.0.0`	2017-06-23
8.0.0	`doi:10.5157/NEPS:SC5:8.0.0`	2016-12-23
6.0.0	`doi:10.5157/NEPS:SC5:6.0.0`	2016-03-31
4.0.0	`doi:10.5157/NEPS:SC5:4.0.0`	2014-09-30
3.1.0	`doi:10.5157/NEPS:SC5:3.1.0`	2014-05-16
3.0.0	`doi:10.5157/NEPS:SC5:3.0.0`	2013-07-05

1.4 Data access

Access to the NEPS data is free of charge, but limited to the purpose of research and to members of the scientific community only. 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 further persons involved in the agreement.

Application for data access

- Fill out the online form for a NEPS Data Use Agreement either in German or in English. Specify a title, the duration, and a short description of the intended research project. Make sure that all project participants with NEPS data access are included and that these persons have signed the agreement. Submit one copy of the complete agreement by e-mail, fax, or post. Further instructions as well as all 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 login 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 in terms of further project participants or a prolonged project duration.

Modes of data access

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

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

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

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

Sensitive information

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

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

This concept of data dissemination translates into an onion-shaped model of datasets: The most sensitive on-site data represent 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 5.

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

> This paper uses data from the National Educational Panel Study (NEPS): Starting Cohort 5—First-Year Students, doi:10.5157/NEPS:SC5:10.0.0. From 2008 to 2013, NEPS data was collected as part of the Framework Program for the Promotion of Empirical Educational Research funded by the German Federal Ministry of Education and Research (BMBF). As of 2014, NEPS is carried out by the Leibniz Institute for Educational Trajectories (LIfBi) at the University of Bamberg in cooperation with a nationwide network.

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

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

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

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

Citing documentation

To refer to any of the documentation materials published in the *NEPS Research Data Documentation Series* (e. g. this manual), cite the document like this:

> FDZ-LIfBi. (2018). *Data Manual NEPS Starting Cohort 5 – First-Year Students, From Higher Education to the Labor Market, Scientific Use File Version 10.0.0*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

or another example:

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

If no author is given, use an universal *NEPS* instead. For example, to refer to survey instruments, use:

> NEPS (Ed.). (2018). *Starting Cohort 5: First-Year Students (SC5), Wave 10, Questionnaires (SUF Version 10.0.0)*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study

If documentation has not been published in this series, use author and title as in the following citation of a field report (which has been written by our survey institute infas):

> 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 correctly and to indicate any kind of publication resulting from the use of NEPS data (see section 1.5) is just one example. 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 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 of Starting Cohort 2, 3, 4, and 5 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*), direct conclusions to be drawn about a Federal State, or 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 of NEPS data use, 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 lot of efforts are 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 also 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.

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 as well as with other researchers in a transparent dialogue. That way, the forum will become a rich knowledge archive 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 and the entire NEPS user community will benefit from a broad participation. You can find the *NEPSforum* at:

→ `www.neps-data.de` > `NEPSforum`

NEPSplorer

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

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

NEPStools

NEPStools is a free to use collection of Stata commands that is created and supplied by the Research Data Center. The package includes some programs (“ado files”) that make NEPS data handling easier. As an example, the `nepsmiss` command automatically recodes all of the numeric missing values from a Scientific Use File (-97, -98, etc.) into Stata’s “Extended Missings” (.a, .b, etc.) with correctly recoded value labels. Another example ist the `infoquery` commend that displays 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 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 data manual as the guiding document to facilitate your work with NEPS data of Starting Cohort 5.

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 Starting Cohort 5, First-Year Students

2.1 From higher education to the labor market

German higher education system has been facing a number of challenges and developments since the early 2000ies, that raised new issues for research. To name but a few, there is the introduction of a two-stage structure in higher education according to the Bologna Process, a growing demand for outcome orientation, the evolution of higher education towards lifelong learning, an increase of (international) competitiveness, and the emerging shortage of highly qualified professionals. At the same time, key issues remained core challenges for the higher education system, such as student dropouts, social selectivity in university entrance, and the relationship between higher education and working life. In order to answer research questions associated with these issues, a cohort of first-year students was followed through their years of study since winter term 2010/11, including their entrance into working life. Central issues to be studied are educational choices, the outcomes of university education, and the entry into the job market.

The main focus is on

- Educational choices during the course of studies and success in studies: What are the determinants of educational decisions and success in studies while studying at a higher education institution – such as dropping out, changing subjects, studying abroad, and pursuing a Master's degree? What is the importance of competencies and social factors, such as social background, gender or migration experiences in this process? Which consequences do decisions have for subsequent education and working life?
- Entrance into working life and professional success: When thinking about students' transition into the job market and their professional success (e.g., occupational position, income, employment security), how important are acquired competencies, on the one hand based on formal qualifications (diplomas), social background, gender, and on the other hand based on social and cultural capital? What role do general competencies play in comparison to subject-specific ones?
- Students' competencies: Which general competencies do students possess to crucial points of time in their students' and young adults' lifecourse (beginning of studies, end of studies/ labour market entry)? How does the competence level influence transitions during studies and beyond (change of subject, higher education drop out, transition to the labour market)? How do competencies correlate with learning environments provided by higher education institutions?

2.2 Sampling strategy

The target population of Starting Cohort 5 is defined as all first-year students of the academic year 2010/2011, independent of their nationality and their knowledge of the German language, who are:

- enrolled for the first time in a public or state-approved institution of higher education in Germany
- aiming at a Bachelor's degree or a state examination (Staatsexamen) in medicine, law, pharmacy, and teaching, or a diploma or Master's degree in Roman Catholic or Protestant theology or specific art and design degrees
- not attending higher education institutions run by Federal Ministries or Federal States for members of their public services (e. g., University of Applied Labour Studies/Hochschule der Bundesagentur für Arbeit)

The sampling process was designed to incorporate an oversampling of teacher education students and students at private higher education institutions. For that reason, a stratified cluster approach has been applied. Administrative data provided by the Federal Statistical Office of Germany constituted the corresponding sampling frame. Each cluster referred to the total of students enrolled in a certain subject at a particular higher education institution (e. g., social sciences at the University of Bamberg). On the primary level, the stratification differentiated between the following four strata; on the secondary level these strata were combined with groups of related subjects:

- clusters linked to teacher education at public universities
- clusters linked to all other fields of study at public universities
- clusters linked to all fields of studies at public universities of applied sciences (Fachhochschulen)
- clusters linked to all degree programs at private higher education institutions

In a second step, all institutions of selected clusters were contacted by the survey agency in order to gain access to the students. The administration of 261 institutions declared their cooperativeness, thereof 104 public universities, 108 public universities of applied sciences, and 49 private university institutions.

In the subsequent recruitment process, two different modes of contact were employed to approach the students and to receive their consent to participate in the panel study:

- conventional mail via higher education institutions administration
- personal information in lectures for freshmen students in the selected fields of studies via interviewers

The former strategy has been applied at all sampled institutions. Recruiting questionnaires in prepared envelopes were transferred to the university administrations together with detailed instructions on how to select the targeted student population. Part of this instruction was the request to include all non-traditional first-year students, i. e., all students with a higher education admission other than the general higher education certificate (Abitur or Fachabitur). It was the task of the higher education institution to compile the respective postal addresses and to send the letters plus reminder letters. Altogether 16,887 filled questionnaires were sent back to the survey agency. The latter strategy presupposed the explicit agreement by the higher education institution and the lecturer to recruit students in appropriate freshmen courses by professional interviewers. In the course of 299 visits at 99 higher education institutions, another 17,229 filled questionnaires could be collected. While the two strategies were conducted parallel during the winter semester 2010/2011, a simplified procedure was applied in the summer semester 2011. Based on postal distribution and display of reduced questionnaires, so-called NEPS address cards, additional 4,169 contact information were gathered.

The returned information of all 38,285 persons were then checked with regard to the belonging to the target population, the existence of double recruitments, and the quality of provided contact details. Finally, 21,438 cases were administrated in the first CATI survey wave of Starting Cohort 5. This first CATI was the prerequisite for staying in the panel.

The sampling design and its consequences for the derivation of sampling weights are fully described in Zinn, Steinhauer, and Aßmann, 2017. Further remarks on the recruiting process are given in the CATI 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 First-Year Students` > `Documentation`

2.3 Survey overview and sample development

This section informs about the progress of the Starting Cohort 5 sample. For each survey wave included in the current Scientific Use File there is a short characterization in terms of field time, number of realized cases, relevant subsamples and domains of competence testing (if appropriate), survey modus, and the institution(s) responsible for collecting the data. Figure 2 starts with an overview illustrating the field times and survey modes from wave 1 to 10.

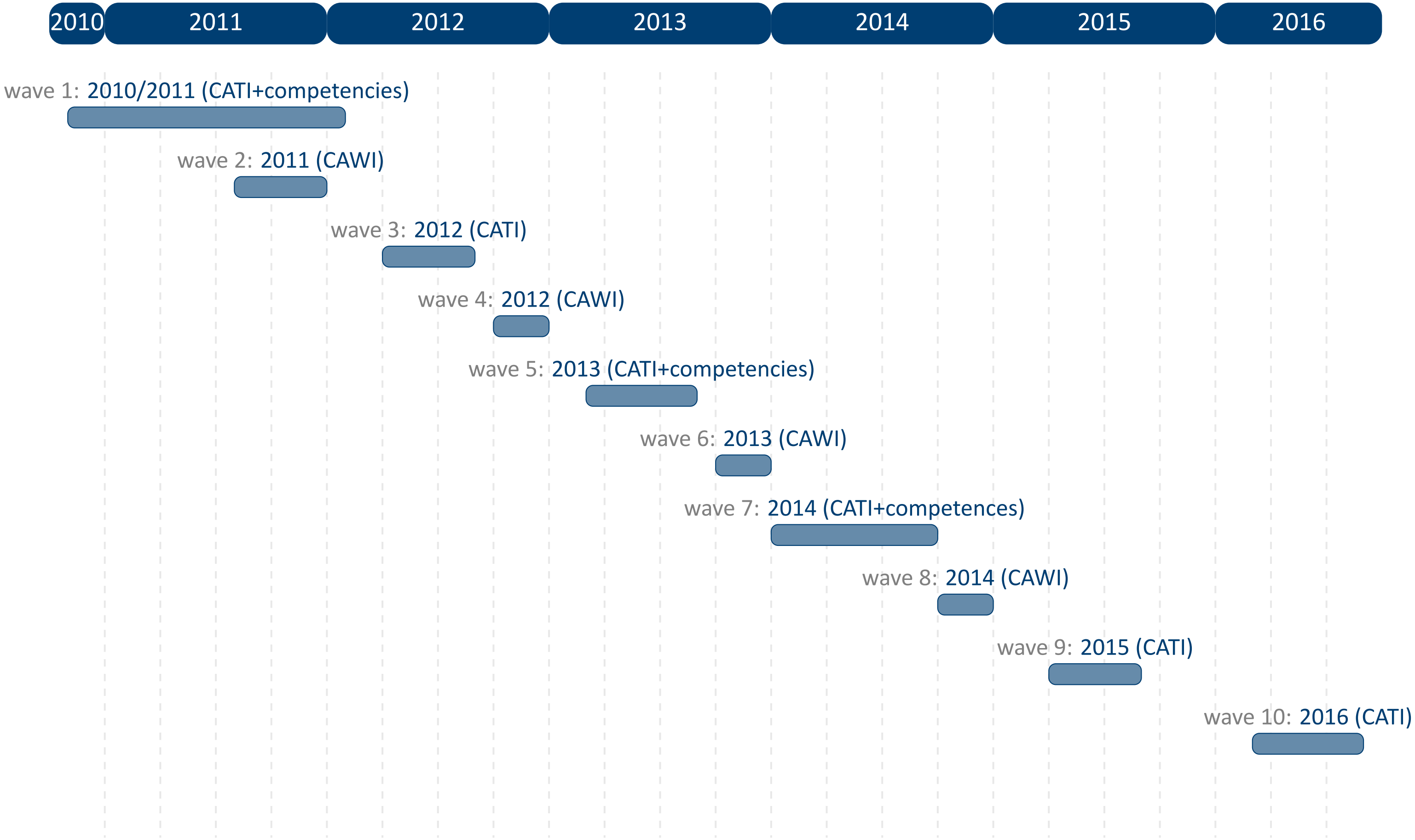


Figure 2: Survey progress of Starting Cohort 5 (waves 1 to 10)

2.3.1 Wave 1: 2010/2011 (CATI+competencies)

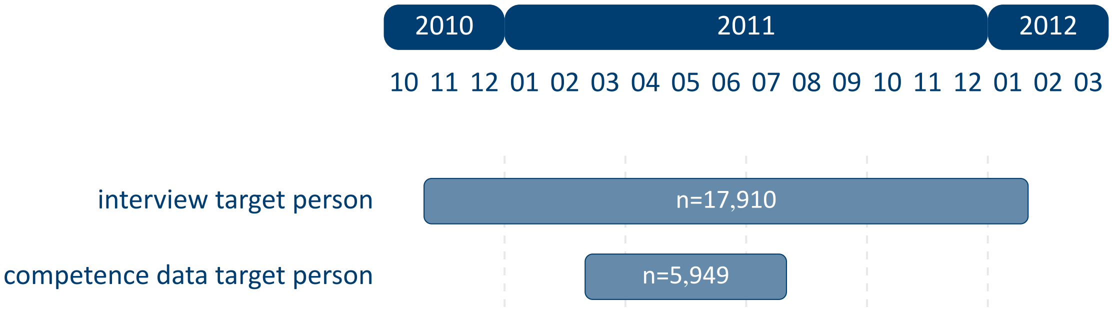


Figure 3: Field times and realized case numbers in wave 1

- Target persons

  Sample First-year students in winter semester 2010/11 (for details about the sampling strategy, see section 2.2)

  Competence tests Reading Competence, Reading Speed, Mathematical Competencies

  Data collection infas – Institute for Applied Social Sciences, Bonn

  Mode of survey Written questionnaires (in each case for recruiting and competence test, PAPI) and computer-assisted telephone interview (CATI)

2.3.2 Wave 2: 2011 (CAWI)

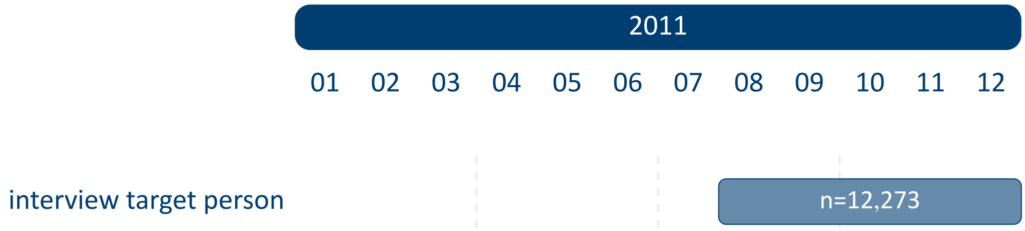


Figure 4: Field times and realized case numbers in wave 2

- Target persons

Sample Survey with the participants of the main survey 2010/2011 additional to CATI-survey

Data collection DZHW - German Centre for Higher Education Research and Science Studies, Hannover

Mode of survey Online survey (CAWI)

2.3.3 Wave 3: 2012 (CATI)

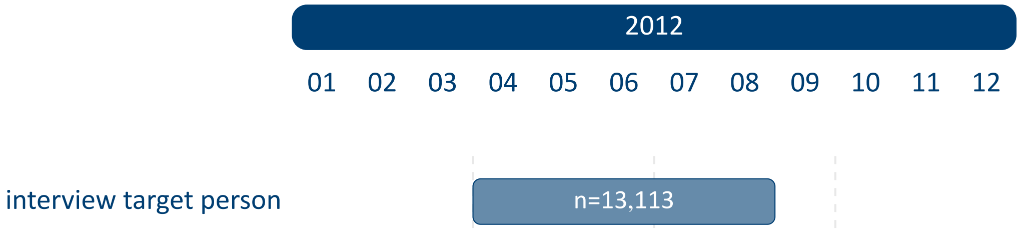


Figure 5: Field times and realized case numbers in wave 3

- Target persons

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Data collection infas – Institute for Applied Social Sciences, Bonn

  Mode of survey Computer-assisted telephone interview (CATI)

2.3.4 Wave 4: 2012 (CAWI)

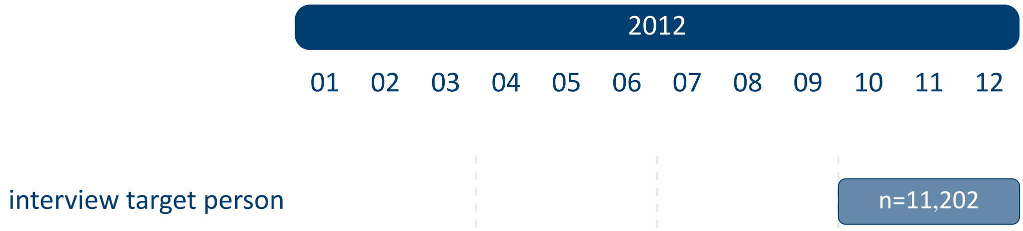


Figure 6: Field times and realized case numbers in wave 4

- Target persons

Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

Data collection DZHW - German Centre for Higher Education Research and Science Studies, Hannover

Mode of survey Online survey (CAWI)

2.3.5 Wave 5: 2013 (CATI+competencies)

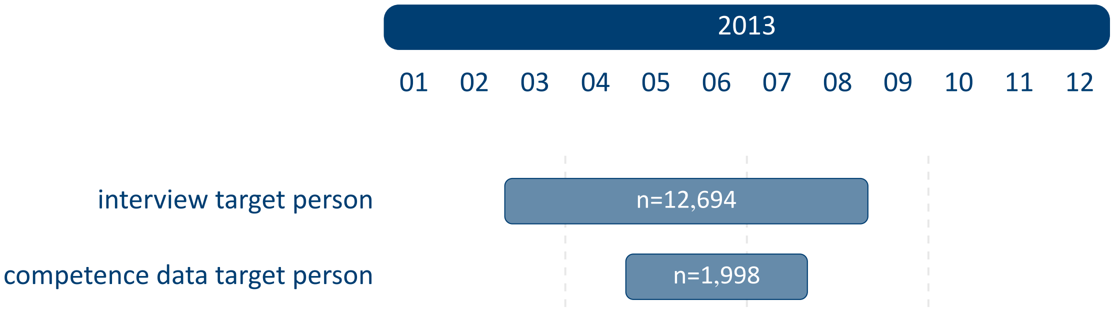


Figure 7: Field times and realized case numbers in wave 5

- Target persons

Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

Competence tests DGCF (Cognitive Basic Skills), Scientific Competence, ICT Literacy

Data collection infas – Institute for Applied Social Sciences, Bonn

Mode of survey Computer-assisted telephone interview (CATI) and group testing (conventional paper-based testing (PAPI), paper-based testing with electronic pens (E-Pen) or computer-based testing with notebooks (CBA)) or individual testing (computer-based online testing, CBWA)

2.3.6 Wave 6: 2013 (CAWI)

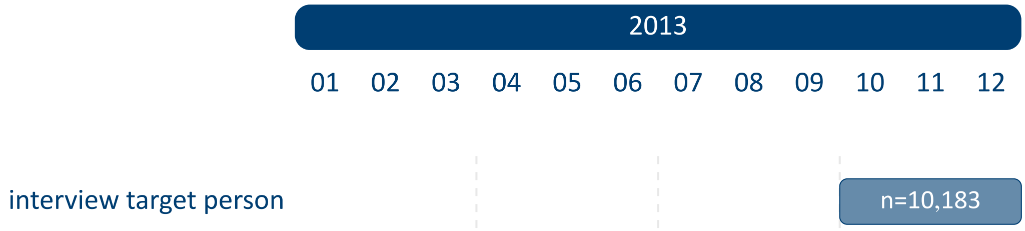


Figure 8: Field times and realized case numbers in wave 6

- Target persons

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Data collection DZHW - German Centre for Higher Education Research and Science Studies, Hannover

  Mode of survey Online survey (CAWI)

2.3.7 Wave 7: 2014 (CATI+competences)

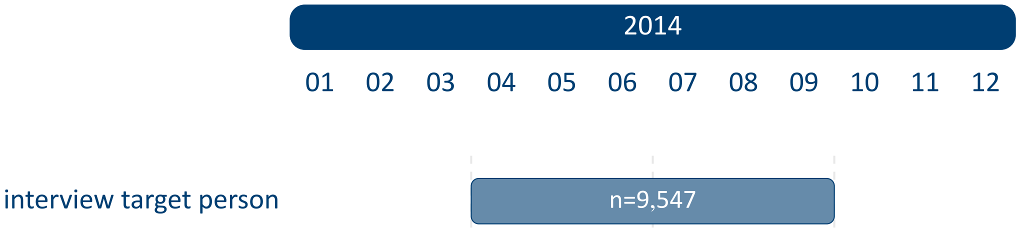


Figure 9: Field times and realized case numbers in wave 7

- Target persons (Subsample A)

  Current wave All students excluding the teaching-oversampling. (see section 2.2 for more information about this subpopulation).

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Data collection DZHW - German Centre for Higher Education Research and Science Studies, Hannover

  Mode of survey Computer-assisted telephone interview (CATI)

- Target persons (Subsample B)

  Current wave Students who study an economic subject or have graduated from such studies. (identifiable via `tx80921` in `CohortProfile`).

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Competence tests Business Administration and Economics

  Data collection DZHW - German Centre for Higher Education Research and Science Studies, Hannover

  Mode of survey Paper-based competence testing within a personal-verbal interview (CAPI)

2.3.8 Wave 8: 2014 (CAWI)

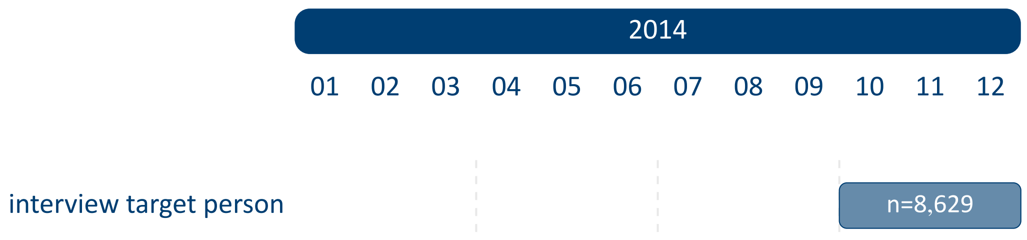


Figure 10: Field times and realized case numbers in wave 8

- Target persons

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Data collection DZHW - German Centre for Higher Education Research and Science Studies, Hannover

  Mode of survey Online survey (CAWI)

2.3.9 Wave 9: 2015 (CATI)

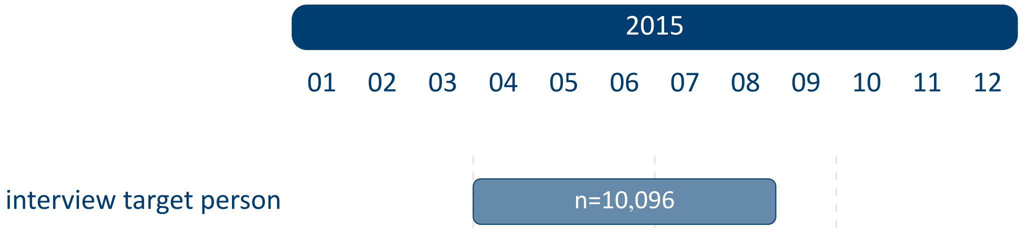


Figure 11: Field times and realized case numbers in wave 9

- Target persons

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Data collection infas – Institute for Applied Social Sciences, Bonn

  Mode of survey Computer-assisted telephone interview (CATI)

2.3.10 Wave 10: 2016 (CATI)

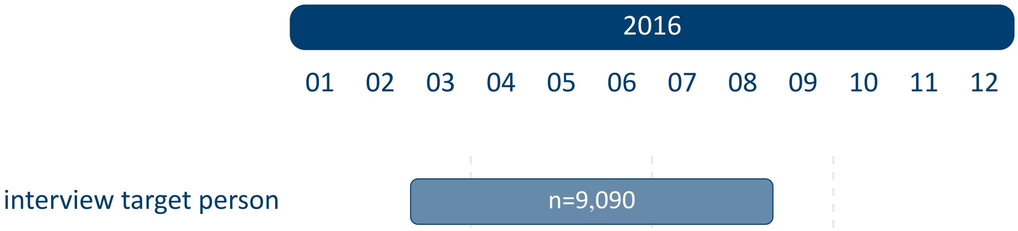


Figure 12: Field times and realized case numbers in wave 10

- Target persons

  Sample Panel sample. Follow-up survey with interviewees willing to participate in the panel.

  Data collection infas – Institute for Applied Social Sciences, Bonn

  Mode of survey Computer-assisted telephone interview (CATI)

2.4 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 implemented in all NEPS starting cohorts covering domain-general cognitive abilities and domain-specific cognitive competencies as well as metacompetencies and stage-specific competencies.

Data from competence tests and direct measures 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 competence scores were estimated can be found in separate reports for the respective competence domains (see Section 1.2).

The scores are compiled in a dataset named `xTargetCompetencies`. This dataset is structured in the so-called WIDE format, that is, all responses of a single respondent are represented in one row of the data matrix. As a consequence, variable names for competence scores follow a specific nomenclature. It not only allows for the identification of the respective domain, the target group, the testing modus, and the kind of scoring, but also informs about the repeated administration of a test item in a different wave or starting cohort (see Section 3.2.2).

The following table shows the schedule of competence measures in Starting Cohort 5 with domains by waves including test modus. The overview contains released data as well as data that is not yet published.

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

		2011	2013	2014	2017
		Wave 1	**Wave 6**	**Wave 7**	**Wave 12**
		(2nd Sem.)	(6th Sem.)	(7th Sem.)	(13th Sem.)[3]
Domain-General Competencies					
DGCF: Cognitive Basic Skills	`dg`	—	P, C, W	—	—
Domain-Specific Competencies					
Reading Competence[1]	`re`	P	—	—	C, W
Reading Speed	`rs`	P	—	—	—
Mathematical Competence[1]	`ma`	P	—	—	C, W
Scientific Competence[1]	`sc`	—	P, C, W	—	—
Metacompetencies					
ICT Literacy[1]	`ic`	—	P, C, W	—	—
Stage-Specific Competencies					
Business Administration and Economics[2]	`ba`	—	—	P	—
English Reading Competence[1]	`ef`	—	—	—	C, W

[1] Subsequent to the respective competence test the target persons had to assess their own test performance (Procedural Metacognition, mp).

[2] Reduced testing: In wave 7, the stage-specific competence test (ba) was realized in a subsample of students and graduates of business sciences only.

[3] Reduced testing: In wave 12, a randomized allocation of competence tests with two out of the three domains (re, ma or re, ef or ma, ef) has been applied.

3 General Conventions

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

The first and foremost concept in creating NEPS SUF data is the one of unaltered data. Wherever possible, the data processing procedures do not change nor destruct the content of the original data. Thus, the full research utility of the data is preserved throughout the whole compilation process. The most prominent (and only systematic) exception to this rule is the recoding of open answers that could have been recorded as a closed answer in the first place (see section 3.4 on page 38 for details on this procedure). In the near future, NEPS Scientific Use Files will ship with a dataset containing backup information for all contents that have been modified by such procedures.

Secondly, all data compilation efforts for NEPS Scientific Use Files implement a strategy for integrating the data as heavily as possible. Our general underlying assumption is that it is far more comfortable for most data users to reduce already integrated data to the focal information for a specific analyses as opposed to (validly) performing integration procedures across scattered source data themselves. This leads to only a few dozen already integrated panel and spell datasets (see section 4.1.1 and section 4.1.2 for details and usage remarks on these structures) for each of NEPS' cohorts, where Scientific Use File compilations starts from several hundred separate dataset files.

In addition to these generic principles, we imposed several conventions to the data structure itself. The general aim of these is to make the NEPS Scientific Use Files from different cohorts as consistent as possible for data users. In other words, we want a data user that is familiar with the data logic in a specific NEPS cohort to instantly recognize these structures when starting to use another. The following sub-sections will give more details about these conventions.

3.1 File names

The naming of data files included in this release follow a number of conventions that are summarized in Table 3 on the next page. Those elements are concatenated with an underscore (_) to generate the full file name.

To give an example, the physical file `SC4_pTarget_D_6-0-0.dta` refers to *Starting Cohort 4*'s data file *pTarget* in its *Download* version of data release *6.0.0*.

Table 3: Naming conventions of file names

Element	Definition
SC[1-6]	**Indicator of starting cohort** 1 = Infants 2 = Kindergarten 3 = 5th grade students 4 = 9th grade students 5 = First-year undergraduate students 6 = Adults
[filename]	**Filename conventions** *Prefix*: x = cross-sectional file; sp = spell file; p = panel file *Keyword/mnemonic*: A keyword or mnemonic 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 school history spells) Filenames of generated datasets do not have a prefix and always start with a capital letter (e. g., CohortProfile, Biography)

[D,R,O]	**Confidentiality Level**
	D = Download version
	R = Remote access version
	O = Onsite version

[#]-[#]-[#](_beta)	**Version** *First digit*: denotes the main release number; the main release number is incremented with every further wave release of a starting cohort; by now, the first digit implies the number of waves included in the release; e. g., in starting cohort 4, the main release number 4 comprises the first four waves of data. *Second digit*: indicates major updates; major updates affect the data structure (e. g., release of imputed datasets); updating your syntax files may be necessary. *Third digit*: indicates minor updates; minor updates affect the content of cells but not the data structure; updating your syntax files is not necessary. _beta: this suffix indicates a preliminary release which allows users to test the data in advance of the main release. The beta version is no longer available after the main release.

3.2 Variable names

The variable naming conventions are aimed at ensuring the consistency of variable names across panel waves. They reflect the panel structure of the NEPS data and allow users to conveniently identify variables across waves.

The principles of the naming conventions are illustrated by Figure 13. More detailed information is given in the following sections:

General conventions for variable names are presented in section 3.2.1. Variables corresponding to test items (competence assessments) follow a separate nomenclature that is optimized for working with competence data. Rules for naming competence variables are introduced in section 3.2.2.

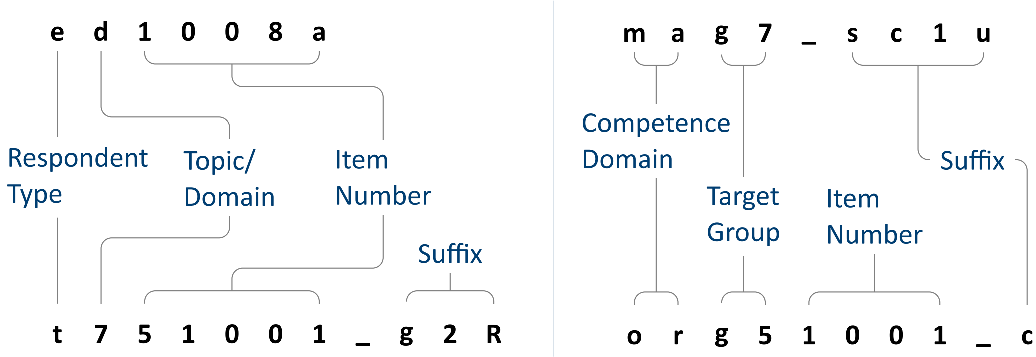


Figure 13: Examples for general variable naming (left) and test data variable naming (right)

3.2.1 General naming conventions

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, e. g., whether the variable is generated and/or only accessible via RemoteNEPS.

Table 4: Naming conventions of variable names

Digit	Description
1	Indicates to which **respondent type** the variable refers; this character can be t (target person), p (parent of target person), e (educator) and h (information about the school/kindergarten given by the head of institution). Sometimes, for the sake of usability names of variables relating to the target begin also with a t even if the target was not the actual respondent. For example this is usually true for generated variables and variables containing para data (e. g., list data from schools/kindergartens).

(...)

Table 4: (continued)

Digit	Description
2	**Topic/domain** (according to the theoretically coordinated dimensions of the NEPS): 1 = competence development (pillar 1) 2 = learning environments (pillar 2) 3 = educational decisions (pillar 3) 4 = migration background (pillar 4) 5 = returns to education (pillar 5) 6 = working group “interest, self-concept and motivation” 7 = socio-demographic information a = from birth to early child care (stage 1) b = Kindergarten to elementary school (stage 2) c = from elementary school to lower secondary school (stage 3) d = from lower to upper secondary school (stage 4) e = from upper secondary school to higher edu./occup. training/labor market (stage 5) f = from vocational training to the labor market (stage 6) g = from higher education to the labor market (stage 7) h = adult education and lifelong learning (stage 8) s = basic program x = generated variables
3–7	**Item number**: The item number typically consists of four numeric characters plus one alphanumeric character.
8–11	**Suffix** (optional): Suffixes are separated from the previous characters by an underscore. There are three types of suffixes: **Suffixes for generated variables** Generated variables are indicated by the suffix `_g#` (`_g1`, `_g2`, etc.). In most cases, the running number after `_g` is a simple enumerator. As scales are generated by a set of other variables, they are also indicated by the above mentioned nomenclature. For the sake of completeness and clarity, it has to be stated that scales are named according to the first variable of the sequence they were generated from. Their running numbers are in so far meaningful as they count up if and only if the first variable of two scales had been identical. **Wide-format suffix**

(...)

Table 4: (continued)

Digit	Description
	Wide-format variables stored are indicated by the suffix `_w#` (e. g., `_w1`, `_w2`, etc.). Note that the wide-suffix not necessarily implies a wave logic. For instance, the presence of a set of variables `a_w1`, `a_w2`, …, `a_w10` means that there are up to 10 values for the variable `a` (e. g., the item corresponding to variable `a` was measured repeatedly in a questionnaire loop) relating to a row entity (e. g., a person or a school episode). Of course, there are cases where suffix `_w#` directly relates to wave-specific values of the underlying variable.
	Confidentiality suffix This suffix pertains to all variables that were anonymized (see section 1.4 on page 5). The suffix indicates a variable's degree of anonymization. This suffix may either stand alone (e. g., country of birth: `t405010_R`) or be combined with other suffixes (e. g., district of place of birth: `t700101_g3R`).
	O Onsite; data on this variable are only available on site **R** RemoteNEPS; data on this variable are available on site or via RemoteNEPS **D** Download; data on this variable has been extracted from the corresponding `O` or `R` variable to make at least some information available in the Download-SUF

3.2.2 Special conventions for variables in test data

Naming of variables corresponding to test items (usually found in competence data files) follow an alternative nomenclature. Variable names consist of three parts and additional suffixes. The first part defines the test instrument (two characters, e. g., `vo` for vocabulary), the second part defines the target group (two characters, e. g., `k1` for children in kindergarten in the first wave, i. e., 2010), and the third part defines the item number.

An overview of the different competence domains is given in the first part of Table 5 on page 34. The first two characters identify competence domains. The target group indicates the cohort or testing wave in which the item was first used. The different target groups are listed in the second part. In some tests (e. g., mathematic competence tests), items are implemented in different testing waves. In these cases, the variable name contains the target group for which the item was first used. The variable name of the item is then fixed and does not change when the item is used again in later waves or other cohorts (e. g., if the item is first used in grade 5, the second part of the variable name will be `G5`, even when the item is reused in grade 7). Thus, the target group identification in the variable name does not necessarily indicate the cohort or testing wave. However, this labeling rule assures items being used in different studies to have the same variable name. Some competence tests are not designed for specific age groups but are implemented unmodified in different cohorts and testing waves. The target group of these tests is indicated by ci (cohort invariant). The item number is defined differently for different competence domains. For most competence domains they only indicate the different items.

The competence data files contain item variables (responses to the test items) as well as overall competence scores. There are two versions of item variables in competence data: scored items named [varname]_c and scored partial credit-items named [varname]s_c. For example, mag9q071_c is a scored variable measuring that the respective math item—targeted at grade 9 students—was *solved* (value 1) or *not solved* (value 0) by the respondent. Note that the item variable does not necessarily indicate that the students' mathematics skills are measured in grade 5. It could also be that the measurement was done in grade 7 and that an item was used that has already been implemented in grade 5. Additionally to the item responses, overall measures of the competence score are given. Suffix _sc[number] is used for several aggregated scores and the meaning of the suffixed number is fixed as follows: 1=WLE (Weighted Maximum Likelihood estimates[1]), 2=standard error of WLEs, 3=sum, 4=mean, 5=difference. For example, variable grk1_sc3 represents the sum score of the grammar test of children being tested in the first wave (2010) in kindergarten. Detailed descriptions on how competence scores are estimated can be found in the respective reports for the different competence domains. If there are several aggregated scores (e. g., different sum scores), letters are appended additionally (e. g., dgg9_sc3a is of the sum score for perceptual speed, while dgg9_sc3b is the sum score for reasoning – both are measures of domain general cognitive functioning).

[1] WLEs are estimated in tests that are scaled based on models of item response theory (cf. Pohl and Carstensen, 2012).

Table 5: Different parts of names of variables in test data

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

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 Cog. Functions (DGCF): Cognitive Basic Skills
ds	Digit Span: Phonological Working Memory
ec	Flanker Task: Executive Control
ef	English Foreign Language: English Reading Competence
gr	Grammar: Listening Comprehension at Sentence Level
hd	Habituation-Dishabituation-Paradigm
ic	Information and Communication Technology (ICT) Literacy
ih	Interaction at Home: Parent-Child Interaction
ip	Identification of Phonemes: Phonological Awareness
li	Listening: Listening Comprehension at Text/Discourse Level
lk	Early Knowledge of Letters
ma	Mathematical Competence
md	Declarative Metacognition
mp	Procedural Metacognition
nr	Native Language Russian: Listening Comprehension
nt	Native Language Turkish: Listening Comprehension
on	Blending of Onset and Rimes: Phonological Awareness
or	Orthography
re	Reading Competence
ri	Rimes: Phonological Awareness
rs	Reading Speed
rx	Early Reading Competence
sc	Scientific Competence
st	Scientific Thinking: Science Propaedeutics
vo	Vocabulary: Listening Comprehension at Word Level

Part II (2-3 chars)**: Target Group** (1 char)**, followed by wave or grade** (digit)

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

(...)

Table 5: (continued)

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

Part IV: Suffix (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 (0=not solved, 1=solved)[1]
_p	Maximum value for an item (only in SC1)
_b	Minimum value for an item (only in SC1)
_m	Mean value for an item (only in SC1)
_s	Sum value for an item (only in SC1)
_n	Number value for an item (only in SC1)
_sc1	Weighted Likelihood Estimate (WLE)[234]
_sc2	Standard error for the WLE[24]
_sc3	Sum score[2]
_sc4	Mean score[2]
_sc5	Difference score (for Procedural Metacognition)[2]
_sc6	Proportion correct score (for Procedural Metacognition)[2]

Identification of repeated test items

Identifying repeatedly measured test items in NEPS data can be easily done by identifying competence variables with an identical word stem. If the same test item is surveyed in different testing waves or starting cohorts, the variable name is marked by an additional suffix while the word stem always indicates the target group for 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. The suffix that points to the repeated use consists of two parts: The first element indicates the starting cohort of current item administration (e. g., sc2 for the Starting Cohort 2—Kindergarten) and the second element indicates the cohort or testing wave (e. g., g1 for students at school in grade 1). To complete the example, the competence variable vok10067_sc2g1_c is a vocabulary item (vo) that was used for the first time in the first kindergarten survey wave (k1)

[1] Partial scored item variables are indicated by s_c (e. g., rea3012s_c: 0=0 out of 2 points, 1=1 out of 2 points, 2=2 out of 2 points).

[2] If there are several aggregated scores for a test available, additional letters are appended to the suffix (e. g., _sc3a, _sc3b).

[3] WLEs and their standard errors are estimated in tests that are scaled based on models of item response theory (cf. Pohl and Carstensen, 2012).

[4] 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).

with the respective item number (0067). It was repeated among the target persons of Starting Cohort 2 at school in grade 1 (`_sc2g1`), and it is available as a scored item response (`_c`).

3.3 Missing values

We provide different missing codes for different situation of missing values. In general, we distinguish between missing codes indicating sorts of item nonresponse, not applicable missings, and edition missings. When working with the NEPS data, make sure that you process those codes in your statistical package correctly. Most packages available provide functions for defining missing values. If you use Stata, you can make use of the `nepsmiss` command provided as a part of the nepstools (see section 1.8 on page 10).

Table 6 provides an overview of missing codes you will encounter in the NEPS data.

Table 6: Overview of missing codes

Code	Meaning	Note
Item nonresponse		
-94	not reached	only applicable for instruments with time restrictions
-95	implausible value	
-97	refused	
-98	don't know	
-29,...,-20	*various*	item-specific missing with informative value labels
Not applicable		
.	filtered	in CATI/CAPI mode
-54	missing by design	not included in sample-specific instrument of this wave
-90	unspecific missing	PAPI mode
-93	does not apply	
-99	filtered	not in CATI/CAPI mode
Edition missings (recoded into missing)		
-52	implausible value removed	
-53	anonymized	
-55	not determinable	
-56	not participated	

We distinguish between three types of missing values:

Item nonresponse occurs if a person did not (validly) respond to a question:

> The most common instances of item nonresponse are *refusals* (–97) and *don't knows* (–98).

Implausible values are coded by a –95 value.

For competence data there is a special missing code (–94) that indicates that a test item has *not been reached* due to timing or other test setting restrictions, so that the respondent had to quit the test somewhere before this item.

Further missing codes (–20, ...,–29) pertain to *item-specific* nonresponse categories (e. g., variable `p407050_D` indicating citizenship of the target child has a missing code –20 for "*stateless*").

Not applicable denotes missing data that occur because the item does not apply to a person. This category comprises two kinds of missings:

The first concerns the survey instruments administered to the different (sub-)samples of a field: If a question is not included in a (sub-)sample-specific questionnaire, it is *missing by design*. The code –54 is assigned to all respondents from this (sub-)sample. This code is used also for the more generic case where values of a variable are not available due to survey design, for instance a survey instrument rotation.

The second concerns individuals: If a question does not apply to a person, it is coded *not applicable* either by the respondent's or the interviewer's remark (–93). If this type of coding is performed via automatic filtering by the survey instrument itself, the system missing value `(.)` is used in CATI/CAPI interviews, and the code -99 (*filtered*) in all other modes.

Missings that can not be classified in one of the above categories are coded by –90 (*unspecific missing*). This value mostly appears in PAPI mode, when a respondent did not fill in a question for unknown reasons.

Edition missings are defined in the process of data preparation:

Implausible values that are removed during data edition are recoded into missing (–52). Data from field instruments are usually incorporated into Scientific Use Files regardless of content plausibility (see section 3 on page 28 for details). However, there are rare exceptions to this rule in cases where item developers explicitly specify the need for data removal.

Sensitive information which is only available via RemoteNEPS and/or Onsite Access is *anonymized* (–53).

In general, coding schemes are used to generate variables (e. g., occupational coding; see section 3.4 on the following page). If the information from the original data is not sufficient to generate a value, we assign the missing code *not determinable* (–55).

In case a person was not present during the interview, or did not fill out a questionnaire at all although it was administered to her, the concerning variables are assigned the missing code *not participated* (-56). This missing code is special in so far as target persons lacking survey data (e. g., due to illness) are usually not entailed in the corresponding datasets.

In the special case of datasets integrating multiple waves widely (such as `xTargetCompetencies`), or including observations for non-participating persons in a wave (such as `CohortProfile`), this missing code is assigned.

3.4 Generated Variables

Coding and recoding processes of open answers and responses

Questions in which the respondent can state the answer in an open format, which is referred to as surveyed string information, can be located at several places within the NEPS survey instruments.

Therefore, NEPS collects many types of information in an open text format so that the respondents can basically state anything they want. A practicable solution for dealing with this kind of entry or answer is the coding and recoding of the information for further processing and later analysis. Generally, coding describes the process of assigning one or many code(s) from selected category scheme(s) or classification(s) to the string information – e. g., in the field of occupations.

The term “recoding” is used here to denote the process of reassigning a code from an already presented closed scheme to open string information from the residual part of the question – e. g., in cases were the respondent hit or choose the answer category “other” and input some self-stated information, not enclosed in the presented scheme, in open format.

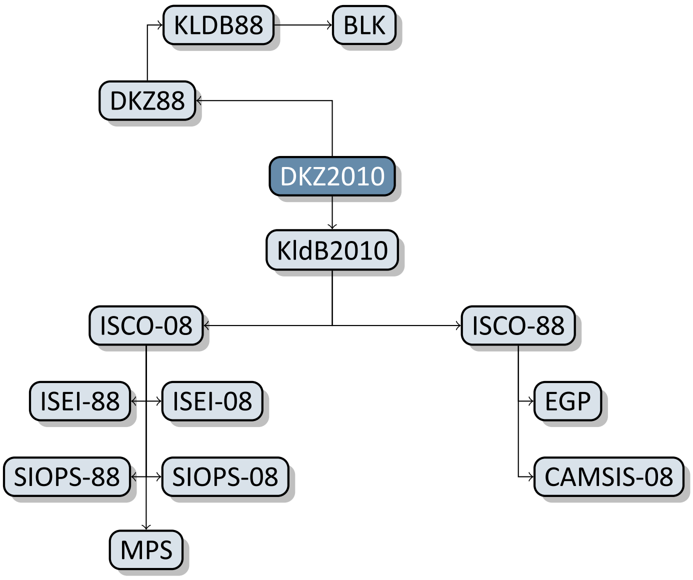
The most common coding scenarios in the field of occupation, education, industry, courses, and regional information are handled by the LIfBi Research Data Center itself. Other coding tasks are spread over the responsible departments at the LIfBi in Bamberg or the partners in the NEPS consortium.

Derived scales and classification

While (re)coding of open or string information into primary classifications (like DKZ2010 or WZ08) is a first and essential step to make the NEPS-SUF data convenient to use, the standardized derivation of other classifications or scales, primarily in the area of educational certificates or occupational titles is a second and different one. We can distinguish at least three types and goals of derivations:

- Derivations from primary classifications (and originated from string/open answers) into other classification which are the primary coding scheme in other studies or for international comparison, e. g., ISCO instead of KldB in the field of occupations;
- Derivations from primary closed form answer schemes into general classifications and schemes making use of auxiliary information – e. g., derivation of ISCED or CASMIN from school certificate and training data plus information on type of school/training;

Figure 14: Derivation paths for several occupational scales and schemes provided in the NEPS



- Combination of the above – e. g., derivation of the EGP class scheme via derived ISCO classifications plus information on self-employment and supervisory status.

Figure 14 provides the derivation paths for several occupational scales and schemes provided in the NEPS. For a detailed description of standard derivations for educational certificates (ISCED, CASMIN and years of education).

4 Data Structure

4.1 Overview

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

Usually, all information collected during a panel wave is appended to the corresponding data file from previous waves. Data files containing longitudinal information from multiple waves are denoted with a *p* in the filename. For instance, the file `pTargetCATI` records data from target's CATI questionnaire, while one row corresponds to one target at one wave. This convention does not fully apply entirely to all panel moments. For example, competence testing has been conducted repeatedly. But because the content of competence tests differs to a large extent, their data structure is best represented in a wide format (see section 4.2.31 on page 99 for a more detailed description). Such data files are denoted with an *x*, which shall indicate the cross-sectional design (one row represents all waves of one respondent).

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

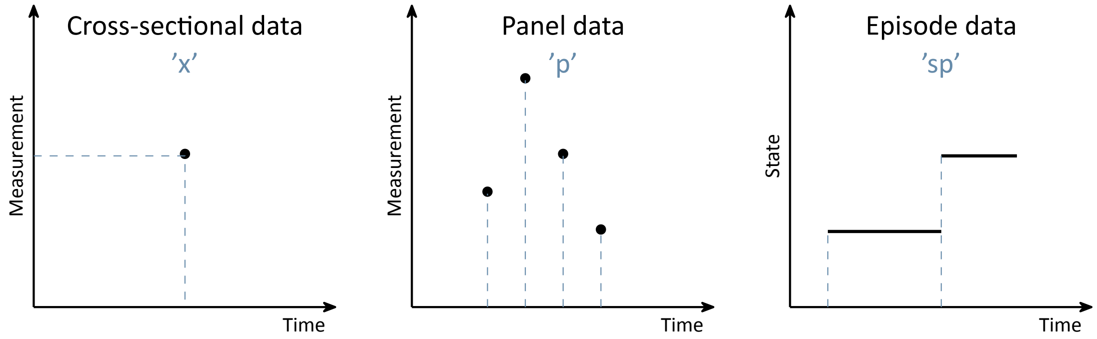


Figure 15: Different types of data structure

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

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

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

wave Indicates the sample wave.

ID_i Identifies a institution. This can be a nursery, school, university, etc. ID_i is unique over waves and Starting Cohorts.

There are additional identifier variables for marking a target's membership to a test group (ID_tg in CohortProfile) and for marking an interviewer in the CATI interviews (ID_int in MethodsCATI and MethodsCompetencies). However, these IDs are not relevant for data merging and negligible for most empirical applications.

4.1.1 Panel data

As stated above, all data from subsequent waves are appended to the already existing data file (as far as possible). We call this method of data handling *integrated panel data*, in contrast to the method of releasing a single file for every wave (where every file only contains data from this unique wave). When first working with integrated panel datafiles, it might be helpful for you to realize the following remarks:

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

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

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

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

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

4.1.2 Episode or spell data

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

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

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

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

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

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

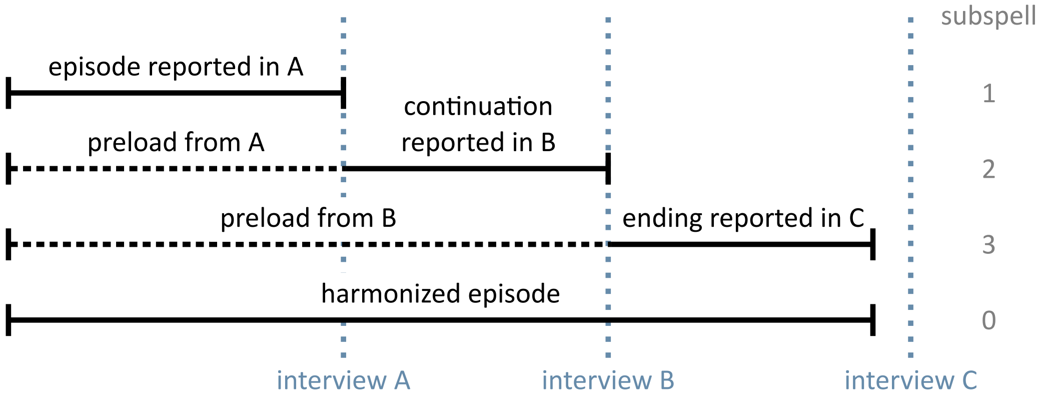


Figure 16: Logic of subspells

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

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

We generally recommend executing

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

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

4.2 Data files

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

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

To ease your understanding of the relationship of those files, figure Figure 17 on the next page 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 datafiles explanatory page.

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

```
** version of this SUF
global version 10-0-0
** path where the data can be found on your local machine
global datapath Z:/Data/SC5/10-0-0
```

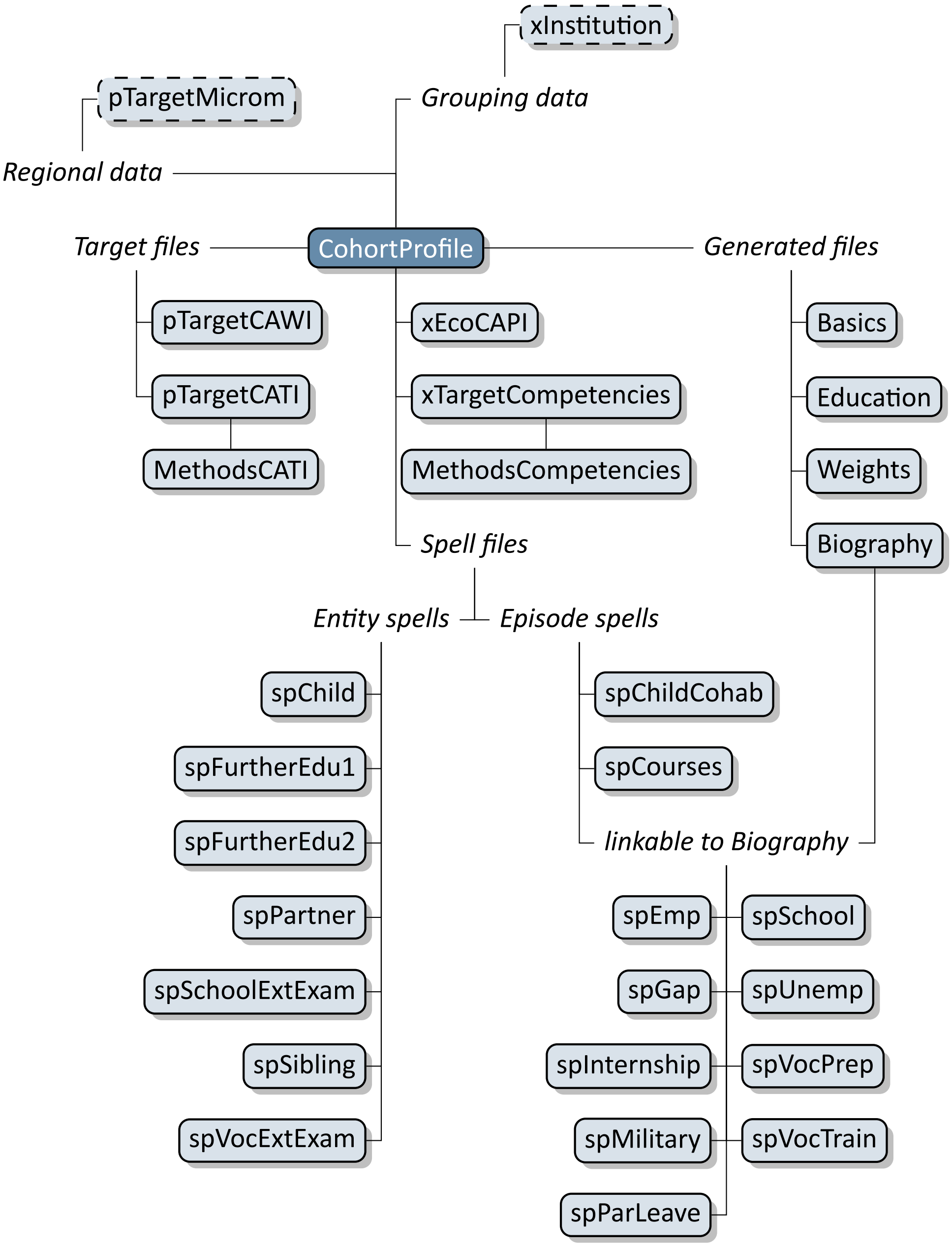


Figure 17: Graphical overview of all data files. Each node represents one data file. Standard merges are indicated by connection lines. Files with a dashed border are not available in the Download SUF. Click on a datafile to get more information.

4.2.1 Basics

« go back to overview

Description

Simplified information about respondents in a plain format

File structure

wide format: 1 row = 1 respondent

ID variables needed to identify a single row

ID_t

Exemplary data snapshot

ID_t	tx29000	t700001	tx29005	t741001	tx29060	tx29904
7001968	25.25	Female	Yes	2	Yes	2
7001969	27.58	Female	Yes	2	Yes	5
7001970	25.67	Female	Yes	2	Yes	2
7001971	22.83	Male	Yes	3	Yes	2
7001972	28.50	Female	Yes	2	No	0

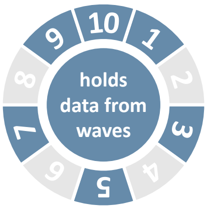
This file contains the latest reported basic information on each respondent, e. g., sociodemographic variables like age in month (tx29000), born in Germany (tx29005), gender (t700001), currently employed (tx29060), but also household characteristics, etc. It also contains meta information about some episodes like the number of main employment spells (tx29904). This data is generated from the pTarget files and a number of spell files. The Basics file is updated prospectively. That is, the file is cross-sectional (i. e., one row per person) and always includes updated information from the latest panel wave a respondent has participated. This simplified data structure can help to gain a first insight in the data. However, it should be handled with care, as it may not feature the *best* information about the respondent. **Please use this file only to get a first overview of the data. Use the original panel or episode files for analyses!**

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

```
** open the data file
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear

** merge the data from Basics, enhancing every entry in CohortProfile
** (i.e. every wave, this is why m:1 merge is needed)
** with information from Basics
merge m:1 ID_t using ${datapath}/SC5_Basics_D_${version}.dta

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

** tabulate gender by wave
tab wave t700001

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

4.2.2 Biography

« go back to overview



Description

Integrated and edited life course data

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t splink

Other ID variables useful for linkage

wave sptype

Exemplary data snapshot

ID_t	splink	wave	sptype	startm	starty	endm	endy
7001968	220001	1	School	August	1996	July	2001
7001968	220002	1	School	August	2001	July	2010
7001968	240001	7	VocTrain	October	2010	August	2013
7001968	260001	7	Emp	September	2006	October	2012
7001968	260002	7	Emp	January	2014	May	2014
7001968	260003	7	Emp	January	2012	May	2014
7001968	360001	1	Internship	September	2010	September	2010
7001968	360002	1	Internship	October	2010	October	2010
7001968	360003	1	Internship	February	2011	March	2011
7001968	360004	7	Internship	April	2012	September	2012

The Biography file is designed to facilitate the analysis of complex life course data that were collected both retro- and prospectively. This dataset pulls together episodes from educational and employment relevance from the following duration spell files: spSchool, spVocPrep, spMilitary, spVocTrain, spEmp, spUnemp, spGap, spInternship, and spParLeave. Use variable sptype to identify this source of the episode.

In contrast to the *raw* life course data from each of these modules, the Biography file offers more consistent life course data that are thoroughly checked and edited. During the interview, inconsistencies in individual life course data were identified and corrected by the data revision module (also “check module (Prüfmodul)”). Those corrected times can be found in the duration spell files as _g1 variables, e. g., variable ts2311y_g1 in spEmp contains the starting date of an employment spell as corrected by the check module. Those corrected times are the starting point for further corrections that have been implemented in the data editing process for Biography.

Overall, the following measures were taken to ensure the integrity of the life course data in the Biography file:

- All subspells were removed; Biography includes only completed, harmonized, or right-censored episodes (i. e., subspell = 0).

- Episodes revoked by the respondents during the interview (i. e., during the check module that cross-checks the biography for gaps and overlaps) or in the next wave (i. e., disagreement in the introductory question for episode updating in the panel questionnaire) were deleted. Note that the revoked episodes are included in the original spell files and can be identified using the corresponding marker variables (`spms` and `disagint`, respectively).
- Starting and end dates of episodes were smoothed and corrected: One-month overlaps between adjacent episodes were resolved.
- Gaps between adjacent episodes that did not exceed two months were closed; gaps of more than two months were defined as specific gap episodes (edition gaps) within the Biography file.

Therefore, we recommend using Biography as a starting point for life course analyses.

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

```
** open the data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

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

4.2.3 CohortProfile

« go back to overview

holds data from waves 1 2 3 4 5 6 7 8 9 10

Description

Para data 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

ID_i ID_tg

Exemplary data snapshot

ID_t	wave	tx80220	tx80521	tx80522	tx80524	inty	testy
7011366	1	Participation	Yes	1	1	2011	2011
7011366	2	Temporary drop-out	No	-54	-55	-56	-54
7011366	3	Temporary drop-out	No	-54	-55	-56	-54
7011379	1	Participation	Yes	1	1	2010	2011
7011379	2	Participation	Yes	-54	1	2011	-54
7011379	3	Participation	Yes	-54	1	2012	-54

The file CohortProfile contains all target persons of the panel sample. These are all targets with an initial agreement to participation. For each respondent in each wave, the CohortProfile contains meta information like the ID of the institution (ID_i), various variables indicating participation (tx80220), availability of survey (tx80521), or availability of test data (tx80522). Furthermore, there are variables on the date of competencies tests (testy, testm) and the date of interview (inty intm intd) beeing conducted.

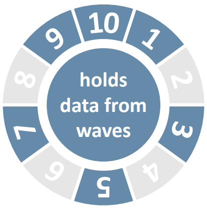
In general, we strongly recommend using this file as a starting point of any analysis!

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

```
** open the data file
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear

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

** how many different respondents are there?
distinct ID_t

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

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

4.2.4 Education

« go back to overview



Description

Generated: upward transitions in educational careers

File structure

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

ID variables needed to identify a single row

ID_t splink

Other ID variables useful for linkage

tx28100

Exemplary data snapshot

ID_t	wave	number	datem	datey	tx28101	tx28102	tx28103
7001974	1	1	7	2003	0	-20	0
7001974	1	2	7	2007	3	10	2
7001974	1	3	5	2010	5	13	3
7001975	1	1	8	1999	0	-20	0
7001975	1	2	8	2005	3	10	2
7001975	1	3	7	2006	5	13	3
7001975	1	4	12	2008	6	15	6

This generated file provides longitudinal information on transitions in respondents' educational careers. It contains only persons who have an educational degree at a lower secondary level or higher. We used all information on educational attainment from `spSchool` (lower, intermediate, and upper secondary school degrees – Hauptschule, Realschule, (Fach-)Abitur), `spVocPrep` (participation in vocational preparation schemes), and `spVocTrain` (all successfully completed trainings). Also, data from `spVocExtExam` and `spSchoolExtExam` have been integrated. Three measures of educational attainment are available: CASMIN (variable `tx28101`), ISCED-97 (`tx28103`), and years of education (`tx28102`; derived from CASMIN). You can easily merge data from the original spells to Education using the variable splink. The file stores transitions in a long event time format. That is, each row represents a transition in at least one classification (CASMIN and/or ISCED-97). Variables on month and year of the transition (datem and datey) specify the event time. We considered only upward educational transitions in CASMIN levels and upward as well as lateral transitions in ISCED-97 levels (CASMIN is ordinal, whereas ISCED-97 has some nominal elements). Because ISCED-97 and CASMIN follow different concepts, some educational transitions are captured by only one of these classifications.

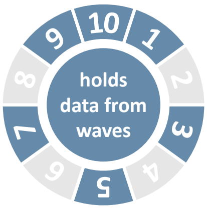
Example 4 (Stata): Working with Education (find R example here)

```
** we want to merge the school type from spSchool to this datafile.
** For this to work, we first have to prepare spSchool and keep only
** harmonized episodes (subspell==0)
use ${datapath}/SC5_spSchool_D_${version}.dta, clear
label language en
keep if subspell==0
```

```
tempfile temp
save `temp'

** now, open the Education data file
use ${datapath}/SC5_Education_D_${version}.dta, clear

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

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

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

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

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

4.2.5 MethodsCATI

« go back to overview



Description

Para data from the targets CATI interview

File structure

long format: 1 row = 1 target in 1 wave

ID variables needed to identify a single row

ID_t wave

Other ID variables useful for linkage

ID_int

Exemplary data snapshot

ID_t	wave	ID_int	tx80302	tx80301	intm	inty	tx80209
7001968	1	1028	50-65 years	2	4	2011	30.55
7001968	3	1405	Up to 29 years	2	-54	-54	-54.00
7001969	1	1111	50-65 years	1	2	2011	39.05
7001969	3	-54	Missing by design	-54	-54	-54	-54.00

This dataset offers a variety of information on the data collection, e. g., gender (tx80301) and age (tx80302) of the interviewer; interview date (intm, inty); interview duration (tx80209); incentives (tx80210); and individual survey participation (tx80220).

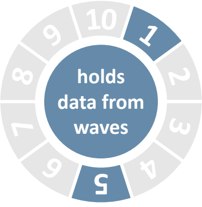
Importantly, MethodsCATI contains all contacted respondents whether an interview was realized or not. Thus, MethodsCATI includes more cases than pTargetCATI.

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

```
** open the data file
use ${datapath}/SC5_MethodsCATI_D_${version}.dta, clear

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

** check out participation status by wave
tab wave tx80220

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

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

4.2.6 MethodsCompetencies

« go back to overview



Description

Para data from the targets competency tests

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_i ID_tg ID_int

Exemplary data snapshot

ID_t	wave	ID_int	tx80301	tx80302	tx80303	tx80661	tx80619
7002163	1	1385	Male	2	7	25	1
7002189	1	1385	Male	2	7	9	2
7002204	1	1318	Female	2	2	24	3
7002204	5	1466	Female	3	18	-54	-54

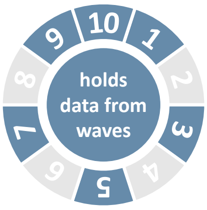
Parallel to other Methods files, this dataset contains information about the testing situation, like durations, dates, interviewer IDs (`ID_int`), information about the interviewer (e. g., sex (`tx80301`), age (`tx80302`), and education (`tx80303`)), individual survey participation (`tx80220`), number of participants (`tx80661`), and disruptions and influences during testing (`tx80619`).

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

```
** open the data file
use ${datapath}/SC5_MethodsCompetencies_D_${version}.dta, clear

** how many respondents have been tested together in a group
bysort ID_tg: generate groupsize=_N if ID_tg>0 & !missing(ID_tg)
summarize groupsize

** create duration of math test; to achieve this, you first have to edit
** both start and end variables (which are stored in time format h:mm)

foreach var in tx80603 tx80604 {  // do the following for both variables
** convert to string, add leading zero
 tostring `var', gen(`var'_str) format(%04.0f)
** generate the etc datetime (ms. since 01jan1960 00:00:00.000)
** take care of missing values!
 gen `var'_ms=clock(`var'_str,"hm") if `var'>0 & !missing(`var')
}
** now the duration is the subtraction of start from end.
** this is recoded then from miliseconds to minutes
generate duration = (tx80604_ms - tx80603_ms)/(60*1000)

summarize duration
```

4.2.7 pTargetCATI

« go back to overview



Description

Data from respondents CATI questionnaires

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_i

Exemplary data snapshot

ID_t	wave	t700001	t70000y	t414000_g2	t531260	t514008
7014289	1	Male	1991	4	-54	-54
7014289	3	Male	1991	-54	-54	-54
7014289	5	Male	1991	-54	-54	-54
7014289	7	Male	1991	-54	-54	9
7014289	9	Male	1991	-54	450	9
7014289	10	Male	1991	-54	.	.
7014290	1	Female	1990	4	-54	-54
7014290	3	Female	1990	-54	-54	-54
7014290	5	Female	1990	-54	-54	-54
7014290	7	Female	1990	-54	-54	7
7014290	9	Female	1990	-54	650	6

The data in file pTargetCATI are from computer assisted telefone interviews (CATI). As many questions are asked repeatedly over different waves, data integration follows a long data format. This means, for each wave participated, there is an additional line for each participating target in this wave. Therefore, targets are uniquely identified by ID_t but lines are unique identified by ID_t and wave together. As there are only lines within pTargetCATI for persons who responded, there are less lines in pTargetCATI than in CohortProfile.[3]

This file contains hundreds of variables, which is the gross of all items surveyed. Some of them are sociodemographic like gender (t700001), year of birth (t70000y), country of birth (t405010_g2), or spoken languages (t414000_g2). Others are repeatedly administered in different waves (e. g., financial means for studying (t531260), satisfaction with studies (t514008)).

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

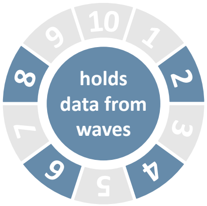
```
** open the CohortProfile
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear

** merge some variable from pTargetCATI
merge 1:1 ID_t wave using ${datapath}/SC5_pTargetCATI_D_${version}.dta, ///
```

[3] includes all students of the panel sample regardless of their questionnaire participation.

```
  keepusing(t400500_g1 t525204) nogen assert(master match)

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

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

tab wave t400500_g1
```

4.2.8 pTargetCAWI

« go back to overview



Description

Data from respondents CAWI questionnaires

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_i

Exemplary data snapshot

ID_t	wave	tg51001	tg51004	t289902	t514001	t30300b
7013263	2	-54	-54	-99	8	-20
7013263	4	-54	-54	-97	9	-20
7013263	6	1	-99	-99	6	-20
7013263	8	1	-99	-99	8	-93
7013322	2	-54	-54	0	8	800
7013322	6	1	-99	1	9	388

Apart from computer assisted telefone interviews (CATIs), data collection via computer assisted web interviews (CAWIs) has been conducted. `pTargetCAWI` also covers similar constructs collected in the CATI. There are items related to the amount of rent (`t30300b`), satisfaction with life (`t514001`), having a roomate (`t289902`), and there are also variables to help you to identify if a target is currently studying (`tg51000, tg51001, tg51004`). In contrast to CATIs, CAWIs are self-administered. Furthermore, biographical data such as episodes of employment or episode of vocational training were not collected.

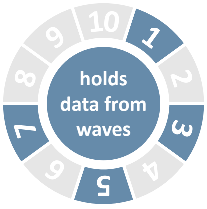
Example 8 (Stata): Working with pTargetCAWI (find R example here)

```
** open pTargetCAWI
use ${datapath}/SC5_pTargetCAWI_D_${version}.dta, clear

** only keep a single variable, and IDs
keep ID_t wave t289902

** suppose you want to know if somebody ever lived with roommates.
** Then you could make use of the expression "t289902==1", which is true (1)
** if there has been a roommate, or false (0) otherwise. The maximum of
** this expression over waves results in 1 if any wave ever evaluated to true,
** and 0 otherwise.
egen roommate = max(t289902==1), by(ID_t)

** only keep this variable; as all waves contain the same information, we
** can fall back to cross-sectional structure
keep ID_t roommate
duplicates drop
tempfile room
```

```
save `room', replace

** finally, open CohortProfile and merge this variable
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear
merge m:1 ID_t using "`room'", nogen

tab wave roommate
```

4.2.9 pTargetMicrom

« go back to overview



Description

regional data about respondents' 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

Exemplary data snapshot

ID_t	wave	regio	ID_regio	mso_k_ausland	mso_k_familie	mpi_k_dichte
7009879	5	1	152322	8	1	1
7009879	5	2	245129	7	1	2
7009879	5	3	305275	8	1	2
7009879	5	4	428884	6	1	.
7009879	5	5	503680	8	2	2
7009879	7	1	145167	8	1	1
7009879	7	2	239686	7	1	2
7009879	7	3	305174	8	1	2
7009879	7	4	426799	7	1	.
7009879	7	5	503553	9	2	2

The data file `pTargetMicrom` is only available **Onsite**. You can not work with this file having only access to the Download or RemoteNEPS SUF.

It contains some regional details of the residence of the respondent on five different regional levels: house area, road section, zip code, zip code 8, municipality.

All those levels are available for every respondent and every wave. There is a lot of regional information in this file, including percentage of foreigners, unemployment rate, family structure, milieu types, car type/density, insurances, only to name a few. To clarify this, those details are **not** about the respondents but about the regional level (e. g., the unemployment rate is not the rate of the respondent but the rate in this municipality). Please be aware that there is a complete documentation about this data file that not only lists all variables but also has a description of the background. See section 1.2 on page 1 on how to find this document.

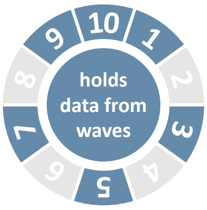
Example 9 (Stata): Working with pTargetMicrom (find R example here)

```
** open Microm datafile. Note that this data file is only available OnSite!
use ${datapath}/SC5_pTargetMicrom_O_${version}.dta, clear

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

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

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

** now you can enhance CohortProfile with regional data
merge 1:1 ID_t wave using ${datapath}/SC5_CohortProfile_O_${version}.dta
```

4.2.10 spChild

« go back to overview



Description

information about all children of respondent

File structure

entity format: 1 row = 1 child of 1 respondent

ID variables needed to identify a single row

ID_t child subspell

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	child	subspell	wave	ts3320y	ts33203
7002313	1	0	5	2006	Male
7002313	1	1	3	2006	Male
7002313	1	2	5	-29	.
7002426	1	0	9	2007	Female
7002426	1	1	5	2007	Female
7002426	1	2	9	-29	.
7002426	2	0	9	2005	Male
7002426	2	1	5	2005	Male
7002426	2	2	9	-29	.

This module contains information on all biological, foster, and adopted children of the respondent, and any other child that currently lives or has ever lived together with the respondent (e. g., children of former and current partners). In cases of twins and higher orders of multiple births, separate episodes are generated for each child. Episodes generally refer to the periods in which the respondent and the child shared a household. The enumerator variable `child` identifies children within respondents. Note that a child episode was skipped in the interview if the respondent reported that the child was deceased. Spell data on cohabitation with children is stored in file `spChildCohab` and spell data on parental leaves relating to children is stored in `spParLeave`.

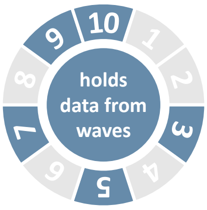
Example 10 (Stata): Working with spChild (find R example here)

```
** open the data file
use ${datapath}/SC5_spChild_D_${version}.dta, clear

** switch to english language
label language en

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

** generate the total count of children for each respondent
** you can do this either by taking the maximum child number:
bysort ID_t: egen children=max(child)
** or counting the number of rows:
```

```
bysort ID_t: gen children2=_N
** which both computes the same result
assert children==children2

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

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

summarize age
```

4.2.11 spChildCohab

« go back to overview



Description

file listing cohabitation spells with children

File structure

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

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

child wave

Exemplary data snapshot

ID_t	child	spell	subspell	wave	ts3331m	ts3331y	ts3332m	ts3332y
7002313	1	101	0	5	3	2011	5	2013
7002313	1	101	1	3	3	2011	7	2012
7002313	1	101	2	5	3	2011	5	2013
7002426	1	101	0	5	1	2013	4	2013
7002426	2	202	0	5	1	2013	4	2013

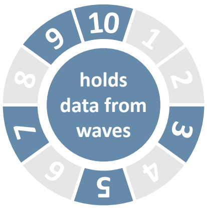
If a respondent lives together with children, durations are registered in spChildCohab. Cohabitation spells are related to children by the child number. Please note that those durations do not necessarily match birth and death events; rather see spChild for direct information on children.

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

```
** open the data file
use ${datapath}/SC5_spChildCohab_D_${version}.dta, clear

** switch to english language
label language en

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

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

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

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

4.2.12 spCourses

« go back to overview



Description

dynamic course module

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t wave splink

Other ID variables useful for linkage

sptype course_w1 course_w2 course_w3

Exemplary data snapshot

ID_t	wave	splink	sptype	course_w1	course_w2	course_w3
7002316	10	260002	26	1001	1002	1003
7002421	3	260002	26	301	302	.
7002421	7	260002	26	701	702	703
7002421	9	260002	26	901	.	.
7002421	10	260002	26	1001	1002	1003
7002942	7	260006	26	701	.	.
7002942	7	260007	26	702	.	.
7003543	10	260004	26	1001	1002	1003

This module comprises courses and trainings attended within the past 12 months during episodes of employment (spEmp), unemployment (spUnemp), parental leave (spParLeave), military, or civilian service(spMilitary), as well as episodes from the spGap module. The starting and end dates of the spells in this module represent the original episodes (in which a course was taken) from those modules. For each of these episodes, information on up to three courses is included in wide format. spCourses comprises all spells from the past 12 months that were recorded in the modules mentioned above. Spells may also be included if no course was taken during this episode. The only criterion for inclusion in the module is that a person provided information on at least one course. Note that in spCourses, the course enumerator is stored in wide format (course_w1, course_w2, and course_w3), whereas in the other course modules (spFurtherEdu1 and spFurtherEdu2) there is only a single enumerator (course). Please note that this information has been integrated into datafile Education. If your interest in this data is not too profound, you are best advised to use Education instead.

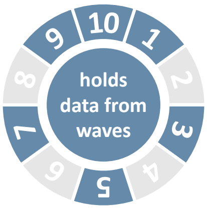
Example 12 (Stata): Working with spCourses (find R example here)

```
** open the data file
use ${datapath}/SC5_spCourses_D_${version}.dta, clear

** check which modules provided course information
tab sptype

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

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

** open the employment module
use ${datapath}/SC5_spEmp_D_${version}.dta, clear

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

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

4.2.13 spEmp

« go back to overview



Description

spell data on employment episodes

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	splink	spell	subspell	ts23203	ts23410	tg26190	ts23320
7006928	260001	1	0	-54	.	5	.
7006928	260001	1	1	-54	420	5	.
7006928	260001	1	2	-29	.	5	.
7006928	260002	2	0	2	1500	5	No
7007631	260001	1	0	-54	.	1	.
7007631	260001	1	1	-54	320	1	.
7007631	260001	1	2	-54	400	1	.
7007631	260002	2	0	2	1700	5	No
7008665	260001	1	0	-54	.	5	.
7008665	260002	2	0	2	4700	3	No
7010182	260001	1	0	-54	.	4	.
7010182	260002	2	0	2	851	5	No

This extensive module covers all spells of regular employment, including traineeships. Information on second jobs is only collected for activities that continue to the interview date. 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., unemployment or military service)

The file comprises information like professional position (`ts23203`), net income (`ts23410`), relevance to degree course (`tg26190`), or permanent contract (`ts23320`).

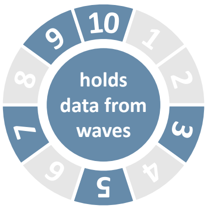
Example 13 (Stata): Working with spEmp (find R example here)

```
** open the data file
use ${datapath}/SC5_spEmp_D_${version}.dta, clear

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

** save this file temporarily
```

```
tempfile tmp
save `tmp'

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.14 spFurtherEdu1

« go back to overview



Description

information about additional courses

File structure

entity format: 1 row = 1 course of 1 respondent

ID variables needed to identify a single row

ID_t course

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	course	t271043	t271048	t271049	t271050	t271051
7003258	901	2	No	2	Yes	.
7003258	902	2	No	2	Yes	.
7003258	903	2	No	2	Yes	.
7003258	904	2	No	2	No	No
7003434	901	10	No	2	.	Yes
7003434	902	10	No	2	.	Yes
7003434	903	10	No	2	.	No

This module contains information on further courses (also private courses) attended within the past 12 months that have not been reported in spCourses or in spVocTrain. These include both professional trainings (similar to those from spCourses) and courses attended for private purposes (e. g., cookery course, yoga course, fortune telling, NLP coaching). Please note that this information has been integrated into datafile Education. If your interest in this data is not too profound, you are best advised to use Education instead.

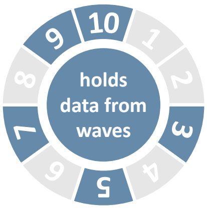
Example 14 (Stata): Working with spFurtherEdu1 (find R example here)

```
** open the datafile
use ${datapath}/SC5_spFurtherEdu1_D_${version}.dta, clear

** one row contains information for one course.
** The only possibility to use this file is to merge it to the data for this
** respondents wave (we use CohortProfile). So first, we have to remodel
** the file so one row contains one wave. We do this by Stata's reshape command
bysort ID_t wave (course): gen course_nr=_n
reshape wide course t*, i(ID_t wave) j(course_nr)

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

** open CohortProfile
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear

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

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

4.2.15 spFurtherEdu2

« go back to overview



Description

information about courses

File structure

entity format: 1 row = 1 course of 1 respondent

ID variables needed to identify a single row

ID_t course

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	wave	course	t279046	t279041	t272043
7002271	3	301	Fully	Little effort	1
7002271	7	702	Fully	Little effort	1
7002271	7	703	Fully	No effort at all	1
7002271	9	901	Fully	No effort at all	3
7002271	9	902	Fully	No effort at all	1
7016013	7	701	.	Some effort	3
7016013	9	901	Fully	No effort at all	3
7016013	9	902	Fully	No effort at all	3
7016013	10	1002	Fully	A lot of effort	3
7016013	10	1004	Fully	No effort at all	3

The survey instrument randomly selected two courses from the spCourses and spFurtherEdu1 modules, collecting additional information on these courses (e. g., costs incurred by employer t279046, motivation t279041, and certificates t272043). These data are included in spFurtherEdu2. Please note that this information has been integrated into datafile Education. If your interest in this data is not too profound, you are best advised to use Education instead.

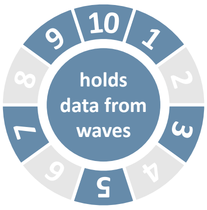
Example 15 (Stata): Working with spFurtherEdu2 (find R example here)

```
** Two possibilities to use spFurtherEdu2

** A) Merge data to spCourses

** open spCourses datafile
use ${datapath}/SC5_spCourses_D_${version}.dta, clear

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

** merge spFurtherEdu2 using ID_t and course
merge m:1 ID_t course using ${datapath}/SC5_spFurtherEdu2_D_${version}.dta, keep(
  master match)
```

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

** open spFurtherEdu1 datafile
use "${datapath}/SC5_spFurtherEdu1_D_${version}.dta", clear

** merge spFurtherEdu2 using ID_t and course
merge 1:1 ID_t course using ${datapath}/SC5_spFurtherEdu2_D_${version}.dta, keep(
  master match)
```

4.2.16 spGap

« go back to overview



Description

reported gap episodes

File structure

spell format: 1 row = 1 gap of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Exemplary data snapshot

ID_t	wave	spell	subspell	splink	ts29101	ts2911y_g1	ts2912y_g1
7002040	1	1	0	300001	12	1991	1991
7002040	1	2	0	300002	7	1998	1998
7002040	1	3	0	300003	7	2009	2010
7002223	1	1	0	300001	11	2003	2003
7002223	1	2	0	300002	11	2006	2007
7002223	1	3	0	300003	9	2010	2010
7002223	7	4	0	300004	11	2011	2011

Gaps in individual life courses are identified by a check module. Such gap episodes are included in the `spGap` module. The spells in this file refer to different types of gaps that can be distinguished by the variable `ts29101` (Type of gap episode). The most common gap episode is (extended) holidays.

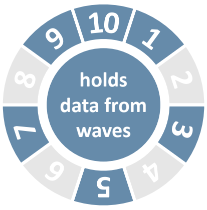
Example 16 (Stata): Working with spGap (find R example here)

```
** open the data file
use ${datapath}/SC5_spGap_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

```
tab sptype _merge
```

4.2.17 spInternship

« go back to overview



Description

reported internship episodes

File structure

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

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Exemplary data snapshot

ID_t	wave	subspell	spell	splink	tg36111_ha	tg3607y_g1	tg3608y_g1
7010227	3	0	1	360001	20	2012	2012
7010227	9	0	2	360002	20	2014	2014
7010227	9	0	3	360003	15	2015	2015
7017880	1	0	1	360001	-54	2009	2009
7017880	1	1	2	360002	-54	2011	2011
7017880	3	0	2	360002	30	2011	2011
7017880	3	2	2	360002	30	2011	2011
7017880	5	0	3	360003	38	2012	2013

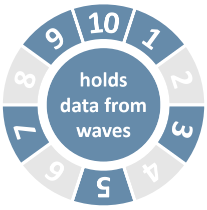
As internships during studies are regarded as central to professional success, both compulsory and voluntary internships have been surveyed and made available in this datafile. Information about duration, renumeration, learning content, and other key aspects have been surveyed.

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

```
** open the data file
use ${datapath}/SC5_spInternship_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.18 spMilitary

« go back to overview



Description

military / civilian service and voluntary gap years

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Exemplary data snapshot

ID_t	wave	spell	subspell	splink	ts2111y_g1	ts2112y_g1
7002424	1	1	1	250001	2009	2011
7002424	3	1	0	250001	2009	2011
7002424	3	1	2	250001	2009	2011
7002424	5	2	1	250002	2012	2013
7002424	7	2	2	250002	2012	2014
7002424	9	2	3	250002	2012	2015
7002424	10	2	0	250002	2012	2016
7002424	10	2	4	250002	2012	2016
7002723	1	1	0	250001	2009	2010

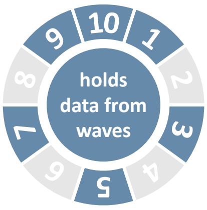
This module includes episodes of military or civilian service as well as gap years taken to do voluntary work in the social or environmental sector. Regular or professional soldiers are considered employed and are therefore included in the employment module.

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

```
** open the data file
use ${datapath}/SC5_spMilitary_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.19 spParLeave

« go back to overview



Description

episodes of parental leave

File structure

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

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave child splink

Exemplary data snapshot

ID_t	wave	child	splink	spell	subspell	ts2711y_g1	ts2712y_g1
7015492	1	1	291001	101	0	2009	2009
7015492	1	1	291002	102	0	2009	2009
7017468	1	1	291001	101	0	1990	1991
7017468	1	1	291002	102	0	1992	1993
7017468	1	2	292003	203	0	1994	1996
7017468	1	3	293004	304	0	1996	1999
7017468	1	3	293005	305	0	1999	2010

For each child in spChild (except for deceased children), information is collected on whether the respondent took a parental leave. Each parental leave episode contributes one record to spParLeave. Parental leaves do not include maternity protection. These periods are added to the corresponding employment episode. As a result, an employment spell is not interrupted if the mother only takes the maternity leave without an additional parental leave.

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

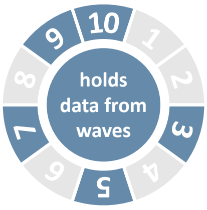
```
** open the data file
use ${datapath}/SC5_spParLeave_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.20 spPartner

« go back to overview

Description

history of partners in the household



File structure

entity format: 1 row = 1 partner of 1 respondent

ID variables needed to identify a single row

ID_t partner subspell

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	partner	subspell	tg2811m	tg2811y	tg2804m	tg2804y
7019619	1	0	9	2014	3	2015
7020618	1	0	1	2010	5	2012
7020618	2	0	12	2012	3	2013
7020618	3	0	5	2013	.	.
7020618	3	1	5	2013	.	.
7020618	3	2	.	.	.	.
7025079	1	0	1	2010	11	2013
7025079	1	1	1	2010	.	.
7025079	1	2	.	.	.	.
7025079	2	0	1	2014	3	2014

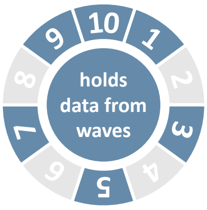
This module covers the partnership history of the respondent. Respondents' subjective reports define whether they live in a relationship and whether they cohabit or not. A comprehensive set of additional questions refers to the present partner. For earlier partners, only information on the year of birth and education is available. Information on the current partner is collected regardless of the cohabitation status, whereas previous partners are only included if they cohabitated with the respondent. The enumerator variable partner identifies partners *within* respondents. This variable is coded 1 for the first partner and counts upwards until the last (current) partner.

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

```
** open the data file
use ${datapath}/SC5_spPartner_D_${version}.dta, clear

** switch to english language
label language en

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

** to find out if a respondent has ever been lived together with a partner,
** you could
t733030
```

4.2.21 spSchool

« go back to overview



Description

general schooling history

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

Exemplary data snapshot

ID_t	wave	splink	spell	subspell	ts1111y_g1	ts1112y_g1
7015643	1	220001	1	0	1996	2000
7015643	1	220002	2	0	2000	2009
7015643	1	220003	3	0	2006	2006
7015770	1	220001	1	0	1996	2002
7015770	1	220002	2	0	2002	2009
7016710	1	220001	1	0	1996	2000
7016710	1	220002	2	0	2000	2006
7016710	1	220003	3	0	2007	2010

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.

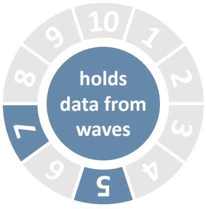
Example 21 (Stata): Working with spSchool (find R example here)

```
** open the data file
use ${datapath}/SC5_spSchool_D_${version}.dta, clear

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

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

```
** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.22 spSchoolExtExam

« go back to overview



Description

school exam certificates acquired outside of the regular German educational system

File structure

entity format: 1 row = 1 exam of 1 respondent

ID variables needed to identify a single row

ID_t exam

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	wave	exam	ts11300_g1	ts1130y	ts11302
7013207	5	1	1	2012	4
7014263	5	1	1	2013	4
7014263	7	2	1	2013	4
7017591	5	1	1	2012	5

The file spSchoolExtExam comprises information about school exam certifications that have not been acquired through “regular” schooling in the German educational system. These can consist of:

- certificates that have been acquired abroad and were accredited by German authorities
- certificates that have been acquired in a German school as external examinee (i. e., without attending class lessons)
- certificates that are automatically awarded by advancing through grades in upper secondary education

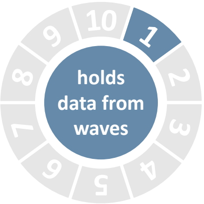
Example 22 (Stata): Working with spSchoolExtExam (find R example here)

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

** first, we have to get the birth date of the respondent
use ${datapath}/SC5_pTargetCATI_D_${version}.dta, clear
keep if wave==1  // only first wave as this data is time-invariant
keep ID_t t70000m t70000y
label language en
tempfile temp
save `temp'

** now, open the data file
use ${datapath}/SC5_spSchoolExtExam_D_${version}.dta, clear
label language en

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

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

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

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

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

4.2.23 spSibling

« go back to overview



Description

siblings of respondent

File structure

entity format: 1 row = 1 sibling of 1 respondent

ID variables needed to identify a single row

ID_t sibling

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	sibling	wave	tg3270m	tg3270y	tg32708	tg32711
7001985	1	1	3	1992	Unemployed	.
7001986	1	1	7	1985	Part-time employed	5
7002004	1	1	7	1987	Full-time employed	5
7002004	2	1	1	1993	Full-time employed	3
7002004	3	1	7	1994	Full-time employed	3
7002020	1	1	7	1987	Full-time employed	3
7002020	2	1	9	1977	Full-time employed	3

The file spSibling contains all reported siblings of the respondent. Each sibling is stored in one row, containing information about birth date (tg3270m/y), employment status (tg32708), and highest degree (tg32711).

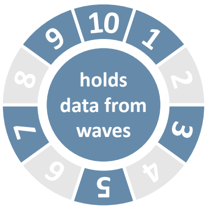
Example 23 (Stata): Working with spSibling (find R example here)

```
** 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 ${datapath}/SC5_pTargetCATI_D_${version}.dta, clear
keep if wave==1  // only first wave as this data is time-invariant
keep ID_t t70000m t70000y
label language en
tempfile temp
save `temp'

** now, open the spSibling data file
use ${datapath}/SC5_spSibling_D_${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:
gen sibling_bdate=ym(tg3270y,tg3270m)
gen  target_bdate=ym(t70000y,t70000m)
format *_bdate %tm
```

```
** check the difference between the two
gen 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
bysort ID_t: egen total_older=total(older)
** generate the total amount of younger siblings
bysort ID_t: egen total_younger=total(1-older)

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

4.2.24 spUnemp

« go back to overview



Description

spell data on unemployment episodes

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Exemplary data snapshot

ID_t	wave	spell	subspell	ts2511m	ts2511y	ts2512m	ts2512y
7001994	1	1	0	8	2010	9	2010
7001994	5	2	0	8	2012	10	2012
7015283	1	1	0	7	2009	9	2009
7015283	1	2	0	9	2010	9	2010
7015283	7	3	0	6	2012	10	2013
7015283	7	4	0	2	2014	7	2014
7015283	10	5	0	5	2016	5	2016

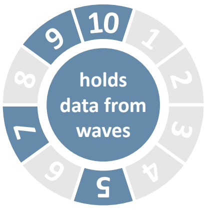
This module includes all episodes of unemployment irrespective of whether a person was registered as unemployed or not. Questions on registration of unemployment and receipt of benefits refer to both the beginning and the end of an unemployment spell.

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

```
** open the data file
use ${datapath}/SC5_spUnemp_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.25 spVocExtExam

« go back to overview



Description

vocational education certificates acquired outside of the regular German educational system

File structure

entity format: 1 row = 1 exam of 1 respondent

ID variables needed to identify a single row

ID_t exam

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	wave	exam	ts1530m	ts1530y	tg24310
7005032	9	1	5	2015	1.6
7005032	10	2	7	2015	1.6
7013153	9	1	9	2014	1.3
7018415	9	1	11	2014	1.8
7018469	10	1	4	2016	1.3
7018514	10	1	4	2016	1.5

The file `spVocExtExam` comprises information about vocational training certifications that have not been received by “regularly” passing through the German vocational training system. These can consist of:

- certificates that have been acquired abroad and were accredited by German authorities
- certificates that have been acquired in a German vocational trainig exam as external examinee (i. e., without attending lessons or courses registered with German authorities)

This especially includes second and third state examinations for alumni of medicine and law studies.

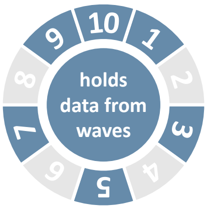
Example 25 (Stata): Working with spVocExtExam (find R example here)

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

** first, we have to get the birth date of the respondent
use ${datapath}/SC5_pTargetCATI_D_${version}.dta, clear
keep if wave==1  // only first wave as this data is time-invariant
keep ID_t t70000m t70000y
label language en
tempfile temp
save `temp'

** now, open the data file
use ${datapath}/SC5_spVocExtExam_D_${version}.dta, clear
label language en
```

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

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

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

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

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

4.2.26 spVocPrep

« go back to overview



Description

vocational preparation schemes

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave

Exemplary data snapshot

ID_t	wave	spell	subspell	ts1311m	ts1311y	ts1312m	ts1312y
7003190	1	1	0	8	1998	8	1999
7003190	1	2	0	10	1999	7	2000
7003572	1	1	0	10	2008	6	2009
7005560	1	1	0	6	2008	6	2008
7005560	1	2	0	8	2008	6	2009

This module comprises episodes of vocational preparation after general education, including

- pre-training courses,
- basic vocational training years, and
- work preparation courses of the employment agency.

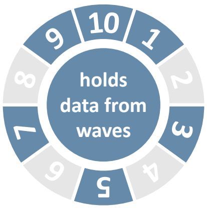
Data were collected on the duration from taking up until completing a vocational preparation scheme, including possible intermissions.

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

```
** open the data file
use ${datapath}/SC5_spVocPrep_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.27 spVocTrain

« go back to overview



Description

vocational education history

File structure

spell format: 1 row = 1 episode of 1 respondent

ID variables needed to identify a single row

ID_t spell subspell

Other ID variables useful for linkage

wave splink

Exemplary data snapshot

ID_t	subspell	tx20100	wave	ts1511m	ts1511y	ts1512m	ts1512y
7002000	0	1	5	10	2010	5	2013
7002000	1	1	1	10	2010	2	2011
7002000	2	1	3	10	2010	5	2012
7002000	3	1	5	10	2010	5	2013
7017520	0	1	10	10	2010	7	2016
7017520	1	1	1	10	2010	6	2011
7017520	2	1	3	10	2010	6	2012
7017520	3	1	7	10	2010	7	2014
7017520	4	1	10	10	2010	7	2016

This module covers all further trainings, vocational and/or academic, that a respondent ever attended:

- vocational training and retraining
- training at technical schools such as schools of public health, full-time vocational schools (excluding basic vocational training years), other vocational schools, and master craftsmen's colleges
- training in specialized fields of medicine
- accredited training courses to receive licenses
- conferral of a doctorate or postdoctoral thesis
- tertiary education at universities, specialized colleges for higher education, colleges of advanced vocational studies, and colleges of advanced administrative and commercial studies. Note: Only the main subjects are surveyed. New episodes are generated if
  - a main subject changes over the course of studies, or
  - the attainable degree changes over the course of studies (e. g., from MA to teaching certification).

  Episodes are continued in case of location changes unless the main subjects change as well.

Training courses for licenses are comparable to courses in the spCourses, spFurtherEdu1, and spFurtherEdu2 modules and can therefore be identified by the spell indicator course. This enumerator allows linking information about the few courses included in this module to the courses in those modules. Interruptions of vocational training spells, so-called vocational interruption episodes, are stored in wide format (be aware of this when working with harmonized spell data!).

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

```
** open the data file
use ${datapath}/SC5_spVocTrain_D_${version}.dta, clear

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

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

** open the Biography data file
use ${datapath}/SC5_Biography_D_${version}.dta, clear

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

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

4.2.28 Weights

« go back to overview

Description

Sample weights for various occasions

File structure

wide format: 1 row = 1 target

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

ID_i

Exemplary data snapshot

ID_t	ID_i	stratum	ID_cl	w_h	w_t1	w_t12468
7001968	1002268	2	1	6.28600	0.84940	-1.0e+03
7001969	1002237	3	254	6.36600	0.53758	0.80384
7001970	1003006	1	202	1.66700	0.24369	0.12049
7001971	1002988	4	396	1.88700	1.08269	-1.0e+03
7001972	1002103	2	112	6.28600	1.30859	-1.0e+03

Weighting variables (starting with w_) are included in the Weights dataset. Also, you find cluster (ID_cl) and stratification (stratum) identifiers here. Given the quite complex structure of the sample, no final recommendations are at hand concerning the use of design and adjusted weights. More information about weight estimation can be found in Zinn et al., 2017. There are no general rules available on how the use of design or adjusted weights render any possible analysis more stable. Weights may possibly help to highlight important features of the analysis, or at least serve as a robustness check for the performed analysis.

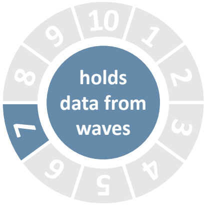
Example 28 (Stata): Working with Weights (find R example here)

```
** open Weights datafile
use ${datapath}/SC5_Weights_D_${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_t123456789

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

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

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

```
** open CohortProfile
use ${datapath}/SC5_CohortProfile_D_${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_t123456789!=0
```

4.2.29 xEcoCAPI

« go back to overview



Description

additional competencies for students of economics and business administration

File structure

wide format: 1 row = 1 student

ID variables needed to identify a single row

ID_t

Other ID variables useful for linkage

wave ID_int

Exemplary data snapshot

ID_t	tx80921	bas7_sc1	bas7_sc2	tg90308	tg24160_g2	tx80200
7002039	6	0.05277	0.38648	7	3	2
7003701	6	-0.99812	0.37384	2	3	6
7003705	6	0.63047	0.42876	7	3	5
7003707	6	-1.08291	0.45696	3	1	10

Apart from the basic CATI-data collection in wave 7, additional data was collected for students of economics and business administration. A paper-based competency test containing questions specificially for the target's field of study was embedded within a short computer assisted personal interview (CAPI).

This data was part of pTargetCATI and xTargetCompetencies in releases prior to data version 10-0-0. To emphasize the focus on this small subgroup of targets, all this information is now gathered in xEcoCAPI. As this file contains data from wave 7 only, ID_t is a unique identifier in this wide-format dataset. To make things simpler, participation in CAPI, CATI, and competency testing is indicated by tx80921. Additional methods data – like number of contact tries (tx80200) and reasons for item-nonresponse in testing (e. g., tx80411) – are availble as well. CAPI data are basically focussing on the student's area of studies (e. g., tg24160_g2).

Example 29 (Stata): Working with xEcoCAPI

```
** open the CohortProfile
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear

** merge some variables from xEcoCAPI
merge 1:1 ID_t wave using ${datapath}/SC5_xEcoCAPI_D_${version}.dta, ///
  keepusing(bas7_sc1 bas7_sc2) nogen assert(master match)

** note that this information in now available only in waves which have
** surveyed the topic
tab wave bas7_sc1
```

4.2.30 xInstitution

« go back to overview

Description

context information about the institution

File structure

wide format: 1 row = 1 area of studies in 1 institution

ID variables needed to identify a single row

ID_i tg04001_g7

Exemplary data snapshot

ID_i	tg04001_g7	tg92104_O	tg92301_O	tg92601_R
1002137	1	0	1	2
1002137	2	0	1	2
1002137	3	0	1	2
1002137	10	0	1	2
1002141	1	0	1	1
1002141	2	0	1	1
1002141	3	0	1	1
1002141	10	0	1	1

Data file `xInstitution` has been constructed during data edition of the first wave. At this time, information about the participating institutions (e. g., universities) has been collected. The file contains data on 10 area of studies for 322 institutions, e. g., about the university region, if the university has been winner or nominee of different prices, the funding body, and number of students, lecturers, and professors. Note that due to data protection issues, this file is not available in the Download version of SUF. You find it in **RemoteNEPS** and **Onsite**. Please not that higher education context data are only available for winterterm 2010/11. The provision of panel data on higher education contexts is currently not planned.

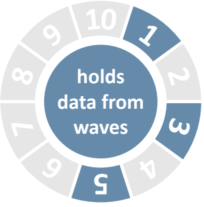
Example 30 (Stata): Working with xInstitution (find R example here)

```
** open datafile
use ${datapath}/SC5_pTargetCATI_O_${version}.dta, clear

foreach var in ID_i tg04001_g7 { // do the following for both variables
** copy the information from the first wave downwards for each target,
** unless a new value has been reported
bysort ID_t: replace `var' = `var'[_n-1] ///
        if `var' == -54|missing(`var')
}
** drop all observations where no satisfaction with studies was reported
drop if t514008 == -98|t514008 == -97|t514008 == -93|t514008 == -54|missing(t514008)

** some respondents reported satisfaction with studies in 7th and in 9th waves
** to keep the latest information, create a seq and a max variables
bysort ID_t: gen seq = _n
bysort ID_t: gen max = _N
```

```
** only keep the latest reported iformation
keep if seq == max
** only keep the variables relevant for the merge and the analysis
keep ID_t ID_i tg04001_g7 t514008

** merge two variables from xInstitution
merge m:1 ID_i tg04001_g7 using ${datapath}/SC5_xInstitution_O_${version}.dta, ///
        keepusing(tg92601_R tg92104_O) nogen assert(master match)

** assuming that the less students at university the more intensive the support by
  the
** university staff per student and the more satisfied are students with their
  studies
** tabulate Satisfaction with studies by Students 2010 total
** note that the following analysis is feasible in both, RemoteNEPS and Onsite
tab t514008 tg92601_R, col

** assuming that students at excellence universities are more satisfied with
** their studies, tabulate the distribution of satisfaction by tg92104_O
** note that the following analysis is only feasible in the Onsite version of SUF,
** since the variable tg92104_O is anonymized in RemoteNEPS
tab t514008 tg92104_O, col
```

4.2.31 xTargetCompetencies

« go back to overview



Description

Test 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*

Exemplary data snapshot

ID_t	wave_w1	wave_w5	mas1_sc1	mas1_sc2
7002311	1	0	-0.91571	0.59815
7002365	1	1	0.43645	0.54009
7012616	1	0	-0.54348	0.53773
7013346	1	0	0.20376	0.55999
7017582	1	0	1.06504	0.60283
7017630	1	1	0.25103	0.67600

File xTargetCompetencies contains data from competence assessments conducted. Scored item variables as well as scale variables are available in a cross-sectional format. Note that not all respondents took part in the assessment. Since assessments were conducted in CAPI mode, those persons who were interviewed in CATI-mode have been excluded from testing. Additionally, those who had severe visual impairments or were even blind were excluded from the assessment.

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

```
** open datafile
use ${datapath}/SC5_xTargetCompetencies_D_${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 1.
generate wave=1
```

```
** now, remove cases which did not took part in the testing
drop if wave_w1==0

** and reduce the dataset to the relevant variables
keep ID_t wave mas1_sc1 mas1_sc2

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

** and merge this to CohortProfile
use ${datapath}/SC5_CohortProfile_D_${version}.dta, clear
merge 1:1 ID_t wave using `tmp', nogen
```

5 Special Issues

5.1 Service Variables (Area of studies, ISCED-97 subject)

subject of study The variables tg2416* were edited due to discrepancies between subspells. Subjects are filled for the first explicit mention only, missing information was labeled accordingly.

Currently the code *-29 "Value from last-mentioned sub-episode"* describes two cases: missing information can be found in the previous sub-spell or in the previous spell (the latter means a person started a new study-episode but claims that the subject is still the same as in the previously recorded episode).

The missing code *-28 "Value from recruitment pTargetCATI"* denotes that the missing information can be found in the recruitment data in file pTargetCATI.

The service variables tg2417* contain the respective subject of study, thus the variables tg24170_g1-5, tg24173_g1-5, tg24176_g1-5 provide complete subject information for all study episodes. Working with the service variables is recommended.

type of university The variable tg01003_g1 (*type of university*, four levels) is originally a part of the first wave recruitment information contained in dataset pTargetCATI. The variable ts15201 (*type of vocational training program*, twenty-four levels) is part of the core education questionnaire and is recorded for each educational spell; it is part of spVocTrain. The service variable tg01003_ha (*type of university*) provides an aggregated version of ts15201 in spVocTrain partly using information from tg01003_g1 for first wave spells, as seen in table 7.

Table 7: Harmonization of type of university

	tg01003_ha		tg01003_g1		ts15201
1	University of applied sciences (incl. cooperative state university)	1	University of applied sciences (incl. cooperative state university)	6	Degree course at an administration and business academy (VWA)
				7	Degree course at a Berufsakademie/cooperative state university
				8	Degree course at a college of public administration
				9	Degree course at a university of applied sciences (not a college of public administration)
2	University	2	University	10	Degree course at a university, including college of education, art college, music college

vocational education history In waves 3, 5, and 7, an attempt has been made to retrospectively gather additional information about vocational education episodes that were concurrent with the first study episode of the winter term 2010/11. This has led to duplicate and/or right-censored episodes in the dataset `spVocTrain`. In order to deal with those episodes, the variable `tx20100` was introduced to give a recommendation which episodes should be used for analyses. The rule applies that episodes from wave 1 are always recommended when the start date lies at or before the beginning of the first study episode of the winter term 2010/11. Episodes from wave 1 are never recommended when the start date lies after the beginning of the first study episode of the winter term 2010/11.

5.2 Coding subject of study

5.2.1 Recruitment

data collection Information on subject of study of initial studies was collected in PAPI and CATI mode (for information on sampling in SC5, see Aßmann et al., 2011, and Zinn et al., 2017). PAPI questionnaires were typewritten and delivered to NEPS by the data collecting institute (infas). Information on subject of study collected in first CATI was delivered to NEPS as original string variable.

coding Coding of subject of study was provided by the NEPS department *From Higher Education to the Labor Market* at DZHW Hannover (formerly HIS), based on data delivered by the data collecting institute (infas) from both modes (CATI and PAPI). The coding process faced a few challenges due to a change of the destatis classificiation between recruitment and first wave data collection: sampling was based on the destatis-classification of 2009/10 while the coding of recruitment information was based on the destatis-classification of 2010/11.

Coding was done manually by occasionally using additional information when a decision could not be taken only based on the string variable.

classification used The classification used for coding the recruitment information on subject of study is based on the destatis classification of 2010/11. Coding decisions can differ from destatis recommendations for coding degree programs into subjects of study due to individual decisions based on extensive research.

5.2.2 Panel Waves

data collection For higher education episodes reported after recruitment, the subject of study has been recorded using lists – in CATI as well as in online surveys. In cases where interviewers were unable to fit a respondents answer into the respective list, the subject of study been recorded as an open string. Both in CATI and online panel waves, the lists are based on the destatis classification 2010/11 and the recruitment information.

To facilitate the allocation of respondent answers, the CATI-list has been continuously extended with supplementary information (based on open responses and changes in the academic landscape in Germany); the online list has remained the same.

Up until wave 13 subjective decisions in the maintenance of the CATI-lists and technical restrictions have led to deviations from the original classification. In some cases, subjects of study were assigned to different codes within the list. The idea behind this was for the other subjects within the same code to serve as covariates, so interviewers (and respondents) could choose the *right* list entry. Starting with wave 13, the CATI-lists will only be extended in the sense that new subject names will be added to the existing subject groups corresponding to a code if those subject names are not already listed under another code. The allocation will follow the coding rules described below to ensure consistency and transparency. This way, the list documented below will not be changed but will be enhanced over time. Starting with wave 14, online waves will use the CATI-list of the previous CATI-wave to harmonize the recording of subject of study in CATI and online mode.

coding Coding of open responses on subject of study has been provided by the NEPS department *From Higher Education to the Labor Market* for all panel waves so far. Since SUF 6.0.0 all strings that have been coded once have been collected in a reference list with their corresponding code by the LIfBi Research Data Center to avoid inconsistencies. In the following waves, open strings have been matched with that list first and strings in the list automatically get assigned the same code. Open strings that have been reported for the first time were coded manually until SUF 9.0.0. Starting with SUF 10.0.0, coding has followed a set of standardized rules and the software CODI has been used.

classification used Data collection and coding of subject of study largely follows the destatis 2010/11 classification of subjects of study.

derivation of SUF-variables In the Scientific Use File, several alternative variables containing subject of study are offered. Variables with the suffix _g1R and _g2 contain the first four digits of the seven digit destatis 2010/11 classification ("Studienbereich" and "Fächergruppe"), _g3R, _g4R and _g5 contain derivations of the destatis classification into different levels of the ISCED 97 classification.All derivations are based on the seven digit version of the destatis classification, using a transcoding table supplied by the Federal Statistical Office.

6 References

Aßmann, C., Steinhauer, H. W., Kiesl, H., Koch, S., Schönberger, B., Müller-Kuller, A., … Blossfeld, H.-P. (2011). Sampling designs of the National Educational Panel Study: Challenges and Solutions, 51–65. doi:10.1007/s11618-011-0181-8

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*.

Dahm, G. (2014). *Starting Cohort 5 - Dokumentation der Variable tg24150_g2 ”NTS” (Nicht-traditionelle Studierende)*. DZHW - Deutsches Zentrum für Hochschul- und Wissenschaftsforschung GmbH.

FDZ-LIfBi. (2018). *Data Manual NEPS Starting Cohort 5 – First-Year Students, From Higher Education to the Labor Market, Scientific Use File Version 10.0.0*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Lauterbach, O. (2015). *Erfassung wirtschaftswissenschaftlicher Fachkompetenzen von Studierenden in Startkohorte 5 des Nationalen Bildungspanels - Technischer Bericht* (NEPS Working Paper No. 51). Bamberg: Leibniz-Institut für Bildungsverläufe, Nationales Bildungspanel.

NEPS (Ed.). (2018). *Starting Cohort 5: First-Year Students (SC5), Wave 10, Questionnaires (SUF Version 10.0.0)*. NEPS Research Data Documentation Series. Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

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

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

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.

Zinn, S., Steinhauer, H. W., & Aßmann, C. (2017). *Samples, Weights, and Nonresponse: the Student Sample of the National Educational Panel Study (Wave 1 to 8)* (NEPS Survey Paper No. 18). Bamberg, Germany: Leibniz Institute for Educational Trajectories, National Educational Panel Study.

A Appendix

A.1 R examples

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

```
setwd("C:/User/..../Desktop/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:

```
'** 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 does not work anymore after modifying the data by, e. g., deleting or merging variables since the single variable labels are not attached to the single variable names. To prevent that, the following steps are necessary:

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

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

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

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

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

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

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

names(spEmp_meta_labels) = spEmp_meta_names
#assigns the names to the labels, so that the vector "spEmp_meta_labels" is now a
  named vector
#this procedure produces the same result as the following command:
#spEmp_meta_labels = c(ID_t = "Target-ID", splink = "Link 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 32 (R): Working with Basics

```
'** import the data files'
CohortProfile =
        read.dta13("SC5_CohortProfile_D_version.dta",
         convert.factors = T)

Basics =
        read.dta13("SC5_Basics_D_version.dta",
        convert.factors = T)

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

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

Example 33 (R): Working with Biography

```
'** import the data file'
Biography =
        read.dta13("SC5_Biography_D_version.dta",
        convert.factors = T)

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

'** check that you will need splink to merge information
 ** from other modules to this file'
```

```
anyDuplicated(Biography[,c("ID_t","splink")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate
```

Example 34 (R): Working with CohortProfile

```
'** import the data file'
CohortProfile =
        read.dta13("SC5_CohortProfile_D_version.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 35 (R): Working with Education

```
'** we want to merge the school type from spSchool to this datafile.
 ** For this to work, we first have to prepare spSchool and keep only
 ** harmonized episodes (subspell == 0)'
spSchool =
        read.dta13("SC5_spSchool_D_version.dta",
        convert.factors = T)

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

'** open the Education data file'
Education =
        read.dta13("SC5_Education_D_version.dta",
        convert.factors = T)

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

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


'** merge spSchool to Education'
#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
```

```
#Since we need a merge indicator here, the merge command has to be extended:
Education = transform(merge(
  x = cbind(Education,source = "master"),
  #x contains the Education data set plus one extra column "source",
  #where source = "master"
  y = cbind(spSchool[,c("ID_t", "splink", "ts11204")],source = "using"),
  # y contains only the columns ID_t, splink and ts11204 from spSchool
  # plus one extra column "source" where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  # merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  # in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  # the columns "source" in x and y are deleted
)


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

Example 36 (R): Working with MethodsCATI

```
'** import the data file'
MethodsCATI =
        read.dta13("SC5_MethodsCATI_D_version.dta",
        convert.factors = T)

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

'** how many different interviewers did CATI surveys?'
length(unique(MethodsCATI$ID_int))

'** create one single variable containing the interview date'
Sys.setlocale("LC_TIME", "C")
#turns off the location-specific language, so that the english months are recognized.

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

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

Example 37 (R): Working with MethodsCompetencies

```
'** open the data file'
MethodsCompetencies =
        read.dta13("SC5_MethodsCompetencies_D_version.dta",
        convert.factors = T)


'** how many respondents have been tested together in a group'
MethodsCompetencies = within(MethodsCompetencies,{
  groupsize = ave(ID_tg, ID_tg, FUN = length)})
#creates a new variable "groupsize" and counts the observations in each ID_tg group

#Problem: NEPS-Missings are also counted as regular values and summirized in groups
for (i in 1:length(MethodsCompetencies$ID_tg)) {
  if(!is.na(MethodsCompetencies$ID_tg[i]) & MethodsCompetencies$ID_tg[i] < 0){
    MethodsCompetencies$groupsize[i] = NA
    #sets all observations to NA for which ID_tg < 0 (here -55 and -54)
  }
}

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


'** create duration of math test'
for (t in names(MethodsCompetencies[,c(38, 39)])) {
# run over columns 38 and 39 (variables tx80603 and tx80804)
  for (i in 1:length(MethodsCompetencies[[t]])) {
      #runs over every single observation
    if(nchar(MethodsCompetencies[[t]][i]) ==   3 & MethodsCompetencies[[t]][i] > 0) {
      #if the observation length is 3 and positive (e.g., "923", but not "-54")
      MethodsCompetencies[[t]][i] = paste("0", MethodsCompetencies[[t]][i], sep = "")
      #adds a leading 0 character, such that 923 becomes 0923
    }
  }
}

install.packages("chron")
library(chron)
#package for creating chronological objects

for (i in names(MethodsCompetencies[,c(38, 39)])){
  MethodsCompetencies[[paste(i, 't', sep = "_")]] =
    #creates new variables tx80603_t and tx80604_t
  times((strftime(strptime(MethodsCompetencies[[i]], format = "%H%M"),"%H:%M:%S")))
    #assigns the values from tx80603 and tx80604 in time format to them
}

MethodsCompetencies$duration =
        MethodsCompetencies$tx80604_t - MethodsCompetencies$tx80603_t
#creates a new variable "duration", subtracting start time from end time
```

```
summary(MethodsCompetencies$duration)
#displays Min, Max and Mean for "duration" in time format
mean(MethodsCompetencies$duration) * 60 * 24
#displays the mean in minutes format
#one unit equals one day, thus it has to be multiplied by 60 minutes and 24 hours

sd(MethodsCompetencies$duration, na.rm = TRUE) * 60 * 24
#displays Std.Dev. for "duration" in minutes format
times(sd(MethodsCompetencies$duration, na.rm = TRUE))
#displays Std.Dev. in time format

length(MethodsCompetencies$duration[!is.na(MethodsCompetencies$duration)])
#displays the number of observations in "duration" without NA
```

Example 38 (R): Working with pTargetCATI

```
'** open the CohortProfile dataset'
CohortProfile =
        read.dta13("SC5_CohortProfile_D_version.dta",
        convert.factors = T)

'** merge some variable from pTargetCATI'

pTargetCATI =
        read.dta13("SC5_pTargetCATI_D_version.dta",
        convert.factors = T)
#imports the pTargetCATI dataset

CohortProfile =
        merge(x = CohortProfile,
        y = pTargetCATI[,c("ID_t",  "wave", "t400500_g1", "t525204")],
        by = c("ID_t", "wave"), all.x = TRUE)
#merges only variables "t400500_g1" and "t525204" from pTargetCATI to CohortProfile

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

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

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

Example 39 (R): Working with pTargetCAWI

```
'** open the pTargetCAWI dataset'
pTargetCAWI = read.dta13("SC5_pTargetCAWI_D_version.dta", convert.factors = T)

'** only keep single variables and IDs'
pTargetCAWI = subset(pTargetCAWI, select = c(ID_t, wave, t289902))

'** suppose you want to know if somebody ever lived with roommates.
 ** t289902 == "Specified" if there has been a roommate,
 ** and t289902 == "Not specified" otherwise. The maximum of
 ** this expression over waves results in 1 if any wave ever evaluated to true,
 ** and 0 otherwise.'
for (i in 1:length(pTargetCAWI$ID_t)){
  if(pTargetCAWI$t289902[i] == "Specified")pTargetCAWI$roommate[i] = 1
        else pTargetCAWI$roommate[i] = 0
}

pTargetCAWI = within(pTargetCAWI, {roommate = ave(roommate, ID_t, FUN = max)})
#for every ID_t with at least one roommate == 1, all other roommate observations
#are also replaced by 1 within this ID_t.

'** only keep this variable; as all waves contain the same information, we
 ** can fall back to cross-sectional structure'
pTargetCAWI = subset(pTargetCAWI, select = c(ID_t, roommate))
pTargetCAWI = pTargetCAWI[!duplicated(pTargetCAWI),]

'** finally, open CohortProfile and merge this variable'
CohortProfile = read.dta13("SC5_CohortProfile_D_version.dta", convert.factors = T)
CohortProfile = merge(CohortProfile, pTargetCAWI, by = c("ID_t"), all = TRUE)
addmargins(table(CohortProfile$wave, CohortProfile$roommate))
```

Example 40 (R): Working with pTargetMicrom

```
'** open pTargetMicrom datafile. Note that this data file is only available OnSite!'
pTargetMicrom = read.dta13("SC5_pTargetMicrom_O_version.dta", convert.factors = T)


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


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

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

```
'** now you can enhance CohortProfile with regional data'
CohortProfile = read.dta13("SC5_CohortProfile_O_version.dta", convert.factors = T)
pTargetMicrom = merge(CohortProfile, pTargetMicrom, by = c("ID_t", "wave"), all =
  TRUE)
```

Example 41 (R): Working with spChild

```
'** open the data file'
spChild = read.dta13("SC5_spChild_D_version.dta", convert.factors = T)

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

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

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

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

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

'** compute the age of ’ones children today
 ** first, create a date of the birth variables'
spChild$ts3320m = match(spChild$ts3320m, month.name)

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

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

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

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

summary(spChild$age)
# displays Min, Max and Mean of "age"
sd(spChild$age, na.rm = TRUE)
# displays Std.Dev. of "age"
length(spChild$age[!is.na(spChild$age)])
```

```
# displays the number of observations in "age" without NA
```

Example 42 (R): Working with spChildCohab

```
'** open the data file'
spChildCohab = read.dta13("SC5_spChildCohab_D_version.dta", convert.factors = T)


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

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

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

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

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

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

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


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

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

Example 43 (R): Working with spCourses

```
'** open the data file'
spCourses = read.dta13("SC5_spCourses_D_version.dta", convert.factors = T)

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

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

'** open the employment module'
spEmp = read.dta13("SC5_spEmp_D_version.dta", convert.factors = T)

'** merge spCourses to spEmp
 ** note that this is an m:1 merge, as there are still subspells in spEmp'
#Since the variable tx80211 is in both data sets spCourses AND spEmp
intersect(names(spCourses), names(spEmp))
#and since the variable is not one of the merging variables, both versions
#are contained in the new merged data set as tx80211.x and tx80211.y.

#To avoid that there are two possibilities:

#1. You can include the variable in the merging process by:
spEmp =
  merge(spEmp, spCourses, by = c("ID_t", "wave", "splink", "tx80211"), all.x = TRUE)
# In that case the version from the master data set, here spEmp, is kept

#OR

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

#compare the two versions of the variable tx80211 by:
addmargins(table(spEmp$tx80211.x, spEmp$tx80211.y))

#and then drop one of the variables by:
spEmp$tx80211.y = NULL

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

Example 44 (R): Working with spEmp

```
'** open the data file'
spEmp = read.dta13("SC5_spEmp_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

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

#Since the variables wave and spms are in both data sets, Biography AND spEmp
#check before merging by: intersect(names(Biography), names(spEmp))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 45 (R): Working with spFurtherEdu1

```
'** open the datafile'
spFurtherEdu1 = read.dta13("SC5_spFurtherEdu1_D_version.dta", convert.factors = T)
```

```
'** one row contains information for one course.
 ** The only possibility to use this file is to merge it to the data for this
 ** respondents wave (we use CohortProfile). So first, we have to remodel
 ** the file so one row contains one wave.'
spFurtherEdu1$course_nr = ave(spFurtherEdu1$ID_t, spFurtherEdu1$ID_t,
                              spFurtherEdu1$wave, FUN = seq_along)

spFurtherEdu1 = reshape(data = spFurtherEdu1,
                        #data in long format
                 idvar = c("ID_t","wave"),
                #idvar is/are the variable/s that need/s to be left unaltered
                 v.names = names(spFurtherEdu1[,3:11]),
                #v.names contains names of variables in the long format that
                #correspond to multiple variable in the wide format
                 timevar = "course_nr",
                #timevar is/are the variable/s that need/s to be converted to
                #wide format
                 direction = "wide")
                #direction is to which format the data needs to be transformed


'** open CohortProfile'
CohortProfile = read.dta13("SC5_CohortProfile_D_version.dta", convert.factors = T)


'** merge the data'
CohortProfile =
        merge(CohortProfile, spFurtherEdu1, by = c("ID_t", "wave"), all.x = TRUE)
'** Please note that you now have multiple variables added to CohortProfile,'
'** one set of variables for each course reported in spFurtherEdu1'
```

Example 46 (R): Working with spFurtherEdu2

```
'** Two possibilities to use spFurtherEdu2'

'--------------------------------------------------------'
'** A) Merge data to spCourses'

'** open spCourses datafile'
spCourses = read.dta13("SC5_spCourses_D_version.dta", convert.factors = T)

'** one row contains information for up to three courses.
** To make merging possible, you first have to reshape the datafile
** so one row contains only one course'
spCourses = reshape(data = spCourses,
                    # data in wide format
                    idvar = c("ID_t","wave","splink"),
                    #idvar is/are the variable/s that need/s to be left unaltered
                    varying = c("course_w1","course_w2","course_w3"),
                    #varying are the variables that need to be converted from
                    #wide to long
                    v.names = c("course"),
                    #v.names defines the name of the variable in that the in
                    #varying defined variables are summarized
                    times = c(1,2,3),
                    #new variable "time" is created with levels 1, 2 and 3
                    #for the three courses
                    new.row.names = 1:100000,
                    #sets row names as numeric
                    direction = "long"
                    ##direction is to which format the data needs to be transformed
                    )

names(spCourses)[names(spCourses) == "time"] <- "course_nr"
#renames the variable "time" to "course_nr"


'** merge spFurtherEdu2 using ID_t and course'
#open spFurtherEdu2 datafile
spFurtherEdu2 = read.dta13("SC5_spFurtherEdu2_D_version.dta", convert.factors = T)

intersect(names(spCourses), names(spFurtherEdu2))
#common variables in the both data sets are "ID_t", "wave", "tx80211" and "course"
#Since the variables "wave" and "tx80211" are not part of the merging process,
#both versions are contained in the new merged data set
#as wave.x/wave.y and tx80211.x/tx80211.y.

'**To avoid that, there are two merging options:'
#1. You can include the variables in the merging process by:
spCourses =
        merge(spCourses, spFurtherEdu2,
        by = c("ID_t", "course", "wave", "tx80211"), all.x = TRUE)
#In that case the versions from the master data set are kept (wave.x and tx80211.x)

#OR

#2. If you'd like to compare the both versions first,
#you can merge the data sets as usual by:
spCourses = merge(spCourses, spFurtherEdu2, by = c("ID_t", "course"), all.x = TRUE)

#compare the two versions of the variables by:
addmargins(table(spCourses$wave.x, spCourses$wave.y))
addmargins(table(spCourses$tx80211.x, spCourses$tx80211.y))

#and then drop one of the versions by:
spCourses$wave.y = NULL
spCourses$tx80211.y = NULL
'---------------------------------------------------------'



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

'** open spFurtherEdu1 and FurtherEdu2 datafiles'
spFurtherEdu1 = read.dta13("SC5_spFurtherEdu1_D_version.dta", convert.factors = T)
spFurtherEdu2 = read.dta13("SC5_spFurtherEdu2_D_version.dta", convert.factors = T)
```

```
'** merge spFurtherEdu2 using ID_t and courses'

intersect(names(spFurtherEdu1), names(spFurtherEdu2))
#common variables in the both data sets are "ID_t", "wave", "course" and "tx80211"
#Since the variables "wave" and "tx80211" are not part of the merging process,
#both versions are contained in the new merged data set
#as wave.x/wave.y and tx80211.x/tx80211.y.

'**To avoid that, there are two merging options:'
#1. You can include the variables in the merging process by:
spFurtherEdu1 =
        merge(spFurtherEdu1, spFurtherEdu2,
        by = c("ID_t", "course", "wave", "tx80211"), all.x = TRUE)
#In that case the versions from the master data set are kept (wave.x and tx80211.x)

#OR

#2. If you'd like to compare the both versions first,
#you can merge the data sets as usual by:
spFurtherEdu1 =
        merge(spFurtherEdu1, spFurtherEdu2,
        by = c("ID_t", "course"), all.x = TRUE)

#compare the two versions of the variables by:
addmargins(table(spFurtherEdu1$wave.x, spFurtherEdu1$wave.y))
addmargins(table(spFurtherEdu1$tx80211.x, spFurtherEdu1$tx80211.y))

#and then drop one of the versions by:
spFurtherEdu1$wave.y = NULL
spFurtherEdu1$tx80211.y = NULL
'---------------------------------------------------------'
```

Example 47 (R): Working with spGap

```
'** open the data file'
spGap = read.dta13("SC5_spGap_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge the spGap to Biography'

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

#Since the variables wave and spms are in both data sets, Biography AND spGap
#check before merging by: intersect(names(Biography), names(spGap))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 48 (R): Working with spInternship

```
'** open the data file'
spInternship = read.dta13("SC5_spInternship_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

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

#Since the variables wave and spms are in both data sets, Biography AND spInternship
#check before merging by: intersect(names(Biography), names(spInternship))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 49 (R): Working with spMilitary

```
'** open the data file'
spMilitary = read.dta13("SC5_spMilitary_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge spMilitary to Biography'

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

#Since the variables wave and spms are in both data sets, Biography AND spMilitary
#check before merging by: intersect(names(Biography), names(spMilitary))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 50 (R): Working with spParLeave

```
'** open the data file'
spParLeave = read.dta13("SC5_spParLeave_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge spParLeave to Biography'

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

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

#Since the variables wave and spms are in both data sets, Biography AND spParLeave
#check before merging by: intersect(names(Biography), names(spParLeave))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 51 (R): Working with spPartner

```
'** open the data file'
spPartner = read.dta13("SC5_spPartner_D_version.dta", convert.factors = T)

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

'** to find out if a respondent has ever been lived together with a partner,
 ** you could'
cbind(addmargins(table(spPartner$t733030)),
        addmargins(prop.table(table(spPartner$t733030))))
```

Example 52 (R): Working with spSchool

```
'** open the data file'
spSchool = read.dta13("SC5_spSchool_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge spSchool to Biography'

#After merging, Stata merge has one variable more than R, because in Stata
#a merge indicator is produced during the merging process and in R isn't.
```

```
#Since we need a merge indicator here, the merge command has to be extended:
Biography = transform(merge(
  x = cbind(Biography,source = "master"),
  #x contains the Biography data set plus one extra column "source",
  #where source = "master"
  y = cbind(spSchool,source = "using"),
  #y contains the spSchool data set plus one extra column "source",
  #where source = "using"
  all.x = TRUE, by = c("ID_t", "splink")),
  #merges x and y by ID_t and splink
  source = ifelse(!is.na(source.x) & !is.na(source.y), "both",
  #in the merged dataset, source = "both" if the observations is in x AND in y
           ifelse(!is.na(source.x), "master", "using")),
           #otherwise, source = "master" if the obs. is only in x
           #and source = "using" if the obs. is only in y
  source.x = NULL,
  source.y = NULL
  #the columns "source" in x and y are deleted
)

#Since the variables wave and spms are in both data sets, Biography AND spSchool
#check before merging by: intersect(names(Biography), names(spSchool))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 53 (R): Working with spSchoolExtExam

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

'** first, we have to get the birth date of the respondent'
#open pTargetCATI
pTargetCATI = read.dta13("SC5_pTargetCATI_D_version.dta", convert.factors = T)

#display value labels
levels(pTargetCATI$wave)

#keep only the first wave as this data is time-invariant
pTargetCATI =
        subset(pTargetCATI, pTargetCATI$wave ==  "2010/2011 (CATI+competencies)")

#keep only ID_t, t70000m and t70000y from pTarget
```

```
pTargetCATI =
        subset(pTargetCATI, select = c("ID_t", "t70000m", "t70000y"))


'** now, open the data file spSchoolExtExam'
spSchoolExtExam =
  read.dta13("SC5_spSchoolExtExam_D_version.dta", convert.factors = T)


'** merge the previously extracted birth dates in pTargetCATI to spSchoolExtExam'
spSchoolExtExam = merge(spSchoolExtExam, pTargetCATI, 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.

spSchoolExtExam$ts1130m = match(spSchoolExtExam$ts1130m, month.name)
spSchoolExtExam$t70000m = match(spSchoolExtExam$t70000m, month.name)
#transforms month names into month numbers


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

spSchoolExtExam$exam_date =
        as.yearmon(paste(spSchoolExtExam$ts1130y, spSchoolExtExam$ts1130m), "%Y %m")
spSchoolExtExam$birth_date =
        as.yearmon(paste(spSchoolExtExam$t70000y, spSchoolExtExam$t70000m), "%Y %m")
#recode the two date variables (year, month) into one

'** calculate the age (in years)'
spSchoolExtExam$age = (spSchoolExtExam$exam_date - spSchoolExtExam$birth_date)


'** show some deviation'
aggregate(spSchoolExtExam$age, by = list(spSchoolExtExam$ts11302),
        FUN = function(x)
                c(mean = mean(x, na.rm = TRUE),
                sd = sd(x, na.rm = TRUE),Freq = length(x)))
#displays mean and sd of age by school-leaving qualification

summary(spSchoolExtExam$age)
#display mean of age in general

sd(spSchoolExtExam$age, na.rm = TRUE)
#display sd of age in general
```

Example 54 (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 pTargetCATI
pTargetCATI = read.dta13("SC5_pTargetCATI_D_version.dta", convert.factors = T)

#display value labels
levels(pTargetCATI$wave)

#keep only the first wave as this data is time-invariant
pTargetCATI =
        subset(pTargetCATI, pTargetCATI$wave ==  "2010/2011 (CATI+competencies)")

#keep only ID_t, t70000m and t70000y from pTarget
pTargetCATI = subset(pTargetCATI, select = c("ID_t", "t70000m", "t70000y"))

'** now, open the data file spSibling'
spSibling = read.dta13("SC5_spSibling_D_version.dta", convert.factors = T)

'** merge the previously extracted birth dates in pTargetCATI to spSibling'
spSibling = merge(spSibling, pTargetCATI, 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.

spSibling$tg3270m = match(spSibling$tg3270m, month.name)
spSibling$t70000m = match(spSibling$t70000m, month.name)
#transforms month names into month numbers

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

spSibling$sibling_bdate =
        as.yearmon(paste(spSibling$tg3270y, spSibling$tg3270m), "%Y %m")
spSibling$target_bdate =
        as.yearmon(paste(spSibling$t70000y, spSibling$t70000m), "%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 55 (R): Working with spUnemp

```
'** open the data file'
spUnemp = read.dta13("SC5_spUnemp_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge spUnemp to Biography'

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

#Since the variables wave and spms are in both data sets, Biography AND spUnemp
#check before merging by: intersect(names(Biography), names(spUnemp))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 56 (R): Working with spVocExtExam

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

'** first, we have to get the birth date of the respondent'
#open pTargetCATI
pTargetCATI = read.dta13("SC5_pTargetCATI_D_version.dta", convert.factors = T)

#display value labels
levels(pTargetCATI$wave)

#keep only the first wave as this data is time-invariant
pTargetCATI =
        subset(pTargetCATI, pTargetCATI$wave ==  "2010/2011 (CATI+competencies)")

#keep only ID_t, t70000m and t70000y from pTarget
pTargetCATI = subset(pTargetCATI, select = c("ID_t", "t70000m", "t70000y"))

'** open the data file spVocExtExam'
spVocExtExam = read.dta13("SC5_spVocExtExam_D_version.dta", convert.factors = T)

'** merge the previously extracted birth dates in pTargetCATI to spVocExtExam'
spVocExtExam = merge(spVocExtExam, pTargetCATI, 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.

spVocExtExam$ts1530m = match(spVocExtExam$ts1530m, month.name)
spVocExtExam$t70000m = match(spVocExtExam$t70000m, month.name)
#transforms month names into month numbers

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

spVocExtExam$exam_date =
        as.yearmon(paste(spVocExtExam$ts1530y, spVocExtExam$ts1530m), "%Y %m")
spVocExtExam$birth_date =
        as.yearmon(paste(spVocExtExam$t70000y, spVocExtExam$t70000m), "%Y %m")
#recode the two date variables (year, month) into one

'** calculate the age (in years)'
spVocExtExam$age = (spVocExtExam$exam_date - spVocExtExam$birth_date)

'** show some deviation'
aggregate(spVocExtExam$age, by = list(spVocExtExam$ts15304),
        FUN = function(x)
               c(mean = mean(x, na.rm = TRUE),
               sd = sd(x, na.rm = TRUE),Freq = length(x)))
#displays mean and sd of age by school-leaving qualification

summary(spVocExtExam$age)
#displays mean of age in general

sd(spVocExtExam$age, na.rm = TRUE)
#displays sd of age in general
```

Example 57 (R): Working with spVocPrep

```
'** open the data file'
spVocPrep = read.dta13("SC5_spVocPrep_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge spVocPrep to Biography'

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

#Since the variables wave and spms are in both data sets, Biography AND spVocPrep
#check before merging by: intersect(names(Biography), names(spVocPrep))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 58 (R): Working with spVocTrain

```
'** open the data file'
spVocTrain = read.dta13("SC5_spVocTrain_D_version.dta", convert.factors = T)

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

'** open the Biography data file'
Biography = read.dta13("SC5_Biography_D_version.dta", convert.factors = T)

'** merge spVocTrain to Biography'

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

#Since the variables wave and spms are in both data sets, Biography AND spVocTrain
#check before merging by: intersect(names(Biography), names(spVocTrain))
#and since the variables are not part of the merging process,
#both versions are contained in the new merged data set as
#wave.x/wave.y and spms.x/spms.y
#For each variable, one of the versions can be dropped by:
Biography$wave.y = NULL
Biography$spms.y = NULL

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

Example 59 (R): Working with Weights

```
'** open the data file'
Weights = read.dta13("SC5_Weights_D_version.dta", convert.factors = T)

'** note that this file is cross-sectional,
 **although the weights seem to contain panel logic'
attr(Weights, "var.labels")

'** only keep weights corresponding to all waves'
Weights = subset(Weights, select = c(ID_t, w_t123456789))

'** create a "panel" logic, i.e., clone each row'
Weights = Weights[rep(seq_len(nrow(Weights)), each = 9),]

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

'** open CohortProfile'
CohortProfile = read.dta13("SC5_CohortProfile_D_version.dta", convert.factors = T)

#value labels of "wave" in "CohortProfile" and "Weights"
#have to be synchronized before merging
```

```
levels((CohortProfile$wave))
levels(Weights$wave)

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

for (i in 1:9) {
  levels(Weights$wave)[i] = levels(CohortProfile$wave)[i]
  #assigns the same value labels to "wave" in "Weights" as in "CohortProfile"
}

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

'** note that this weight is only nonzero if respondents participated in all waves'
with(subset(CohortProfile, w_t123456789 ! =  0), addmargins(table(wave, tx80220)))
```

Example 60 (R): Working with xInstitution

```
'** open datafile pTargetCATI'
pTargetCATI = read.dta13("SC5_pTargetCATI_D_version.dta", convert.factors = T)

'** copy the information from the first wave downwards for each target,
 ** unless a new value has been reported'
for (t in names(pTargetCATI[c("ID_i", "tg04001_g7")])) {
#run over variables ID_i and tg04001_g7
 for (i in 2:length(pTargetCATI$ID_t)) {
 #run over all observations
   if(pTargetCATI$ID_t[i] ==  pTargetCATI$ID_t[i-1]){
          #for the same ID_t, check...
     if(is.na(pTargetCATI[[t]][i]) | pTargetCATI[[t]][i] ==  "Missing by design"){
       #...whether missing value or -54(Missing by design)
       pTargetCATI[[t]][i] = pTargetCATI[[t]][i-1]
       #copy information downwards, unless a new value has been reported
      }
    }
  }
}

'** drop all observations where no satisfaction with studies was reported'
levels(pTargetCATI$t514008)

#remove observations with NA in t514008
pTargetCATI = pTargetCATI[!(is.na(pTargetCATI$t514008)),]

#remove observations with other missings in t514008
pTargetCATI = subset(pTargetCATI, !(t514008 ==  "Don't know"
                                  | t514008 ==  "Refused"
                                  | t514008 ==  "Does not apply"
                                  | t514008 ==  "Missing by design"))


'** some respondents reported satisfaction with studies in 7th and in 9th waves
```

```
 ** to keep the latest information, create a seq and a max variables'
pTargetCATI = within(pTargetCATI,{seq = ave(ID_t, ID_t, FUN = seq_along)})
pTargetCATI = within(pTargetCATI,{max = ave(ID_t, ID_t, FUN = length)})

'** only keep the latest reported iformation'
pTargetCATI =
        subset(pTargetCATI, pTargetCATI$seq ==  pTargetCATI$max)

'** only keep the variables relevant for the merge and the analysis'
pTargetCATI =
        subset(pTargetCATI, select = c("ID_t", "ID_i", "tg04001_g7", "t514008"))

'** merge two variables from xInstitution'

#open datafile xInstitution
xInstitution = read.dta13("SC5_xInstitution_O_version.dta", convert.factors = T)

#merge xInstitution to pTargetCATI
pTargetCATI =
  merge(x = pTargetCATI,
              y = xInstitution[,c("ID_i", "g04001_g7", "tg92601_R", "tg92104_O")],
              by = c("ID_i", "g04001_g7"), all.x = TRUE)

'** assuming that the less students at university the more intensive the support by
 ** the university staff per student and the more satisfied are students with their
 ** studies tabulate Satisfaction with studies by Students 2010 total
 ** note that the following analysis is feasible in both, RemoteNEPS and Onsite'
cbind(addmargins(table(pTargetCATI$t514008, pTargetCATI$tg92601_R)))
cbind(addmargins(prop.table(table(pTargetCATI$t514008, pTargetCATI$tg92601_R))))


'** assuming that students at excellence universities are more satisfied with
 ** their studies, tabulate the distribution of satisfaction by tg92104_O
 ** note that the following analysis is only feasible in the Onsite version of SUF,
 ** since the variable tg92104_O is anonymized in RemoteNEPS'
cbind(addmargins(table(pTargetCATI$t514008, pTargetCATI$tg92104_O)))
cbind(addmargins(prop.table(table(pTargetCATI$t514008, pTargetCATI$tg92104_O))))
```

Example 61 (R): Working with xTargetCompetencies

```
'** open the data file xTargetCompetencies'
xTargetCompetencies =
        read.dta13("SC5_xTargetCompetencies_D_version.dta", convert.factors = T)

'** as the x in the filename indicates, this is a cross sectional file
 ** (no wave structure). You can verify this by asking if one row is
 ** solely identified by the respondents ID'
anyDuplicated(xTargetCompetencies[,c("ID_t")])
#returns "0" if there are no duplicates.
#If there are duplicates this command returns the index of the first duplicate

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

```
table(xTargetCompetencies$wave_w1)
table(xTargetCompetencies$wave_w5)
table(xTargetCompetencies$wave_w7)

'** to work with competence data, you might want to merge it to CohortProfile.
 ** if you want to keep the panel logic (and not only add all competencies
 ** to every wave), you need a mergeable wave variable in xTargetCompetencies.
 ** here, we focus on math competencies, that have been tested in wave 1.'
xTargetCompetencies$wave =
        rep(levels(CohortProfile$wave)[1],length(xTargetCompetencies$ID_t))
xTargetCompetencies$wave = as.factor(xTargetCompetencies$wave)

'** now, keep cases which did took part in the testing'
xTargetCompetencies = subset(xTargeCompetencies, wave_w1 ==  "ja")

'** and reduce the dataset to the relevant variables'
xTargetCompetencies =
        subset(xTargetCompetencies, select = c(ID_t, wave, mas1_sc1, mas1_sc2))

'** and merge this to CohortProfile'

#open the data file Cohort Profile
CohortProfile = read.dta13("SC5_CohortProfile_D_version.dta", convert.factors = T)

#look for common variables in both data sets
intersect(names(CohortProfile), names(xTargetCompetencies))

#merge CohortProfile with xTargetCompetencies
CohortProfile =
  merge(CohortProfile, xTargetCompetencies, by =  c("ID_t", "wave"), all = TRUE)
```

A.2 Release notes

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

```
==================================================
**
** NEPS STARTING COHORT 5 – RELEASE NOTES a.k.a CHANGE LOG
** changes and updates for release NEPS SC5 10.0.0
** (doi:10.5157/NEPS:SC5:10.0.0)
**
==================================================

* Known Issues *



===================================================
* Changes introduced to NEPS:SC5 by version 10.0.0 *
===================================================

General remarks:
        – several variables surveyed prior to wave 10 have been renamed to *_v1 and *
            _v2,
                as wording of question texts has changed in recent survey instruments

CohortProfile:
        – testy testm testd erroneously had been coded to –56 even though tx80522==1;
            this has been fixed
        – new indicator variable tx80121 has been introduced: subsample ”students of
            economics”
        – tx80921 has been revised

xEcoCAPI:
        – new dataset featuring items from CAPI–shortquestionaire, economics–competency
            –test and the corresponding
                methods data that has been administered to students of economics in
                    wave 7; all of these data
                has been removed from pTargetCATI, xTargetcompetencies, and
                    MethodsCompetencies, respectively, for this subsample


==================================================
* Changes introduced to NEPS:SC5 by version 9.0.0 *
==================================================

pTargetCATI:
        – ts15911 (highest degree obtained) was falsely programmed in wave 9. Therefore
             ts15911_g1 was generated for all participants.

spVocTrain:
        – original variables tg2416* (subjects) were edited due to discrepancies
            between subspells. Subsequently, subjects are filled for the first
            explicit mention only. Missing information was labeled accordingly.
            Working with service variables is recommended.
        – service variables tg2417* (subjects) have been revised so that each subspell
            of a corresponding spell is now filled with the first information
            available, still variables tg24170_g1–_g5, tg24173_g1–_g5 and tg24176_g1–
            _g5 provide complete information for all study episodes.
        – ts15221 (qualification sought) was falsely derived in some cases. Therefore,
            ts15221_g1 was generated for the affected episodes



==================================================
* Changes introduced to NEPS:SC5 by version 8.0.0 *
==================================================

General remarks on harmonization of variables concering subjects, type of university
    and type of vocational training program:
        – harmonization of type of university−variable:  tg01003_g1(pTargetCATI) >>
            tg01003_ha (spVocTrain, considering values of ts15201)
        – harmonized service variables on subjects: tg24160_g*, tg24163_g*, tg24166_g*
            (spVocTrain) >>  tg24170_g*, tg24173_g*, tg24176_g* in spVocTrain (
            considering values of tg04001_g1−5, tg04004_g1−5, tg04007_g1−5 in
            pTargetCATI)
        – harmonization provides valid values for type of university and subjects where
             information on study episode from winter term 2010/11 was missing
        – missing codes −28, −29 were introduced in the original variables tg24160_g*,
            tg24163_g*, tg24166_g*, tg01003_g1, ts15201


CohortProfile:
        – tx80951 indicates the participation status for students of economics in wave
            7. Besides CATI survey and competency testing, these students had also the
             possibility of taking parting in a short CAPI questionaire as well.


pTargetCATI:
        – the concept of reflecting migrational background in NEPS SUFs has been
            improved in order to also represent migrants in 3.75th generation; thus,
            the older variables on migrational background [t400500_g1,t400500_g2,
            t400500_g3] in the pTargetCATI dataset have been renamed using the "v1"
            suffix [t400500_g1v1,t400500_g2v1,t400500_g3v1], and the new ones have
            been introduced
        – variables of students of economics who took part in a short CAPI questionaire
             were added to pTargetCATI


spVocTrain:
        –  service variables tg2417* (subjects) and tg01003_ha  (type of university)*
            were introduced to simplify working with the dataset. Small discrepancies
            from the original variables (tg2416*) cannot be ruled out and have to be
            considered by the user.
        – each subspell of a corresponding spell was filled with the most recent
            information available, so that the variables tg24170_g1−5, tg24173_g1−5,
            tg24176_g1−5 provide complete information for all study episodes.



==================================================
* Changes introduced to NEPS:SC5 by version 6.0.0 *
==================================================

General:

        – starting with this release, all NEPS Scientific Use Files will ship with an
            additional, unicode−enabled Stata data set version;
                this version is only readable in Stata version 14 or younger, and is
                    placed in the subdirectory "Stata14"
```

- translation for all meta data (variable and value labels, question texts, etc) have been revised and completed
- meta data for all variables have been revised and updated where appropriate
- additional waves 5 (CAWI) and 6 (CATI/CAPI) have been incorporated into the data
- the subspell harmonization routine in all spell datasets ("sp*") has been updated, leading to more accurate harmonized subspell information (subspell==0) for panel continuation spells
- staff from NEPS stage 7 at the DZHW excessively reviewed and overworked all syntax for generated tg*-variables, which may lead to slightly different contents
- staff from NEPS stage 7 at the DZHW reviewed the cohorts' sample frame in consultation with NEPS methods department, leading to 3 observations removed from the SUF
- all datasets from version 4.0.0 did not reflect the correct doi in their dataset labels; the correct doi would have been "10.5157/NEPS:SC5:4.0.0", not "none"; this issue has been fixed and all datasets of version 6.0.0 correctly are labeled with doi:10.5157/NEPS:SC5:6.0.0

xTargetCompetencies:

- all variables of domains "maths" and "reading" erroneously contained the missing value -54 ("missing by design") in versions 4.0.0 and 3.1.0; as there were no additional competency assessments in wave 4, it was safe to use the xTargetCompetencies dataset file from version 3.0.0 instead without missing any information; this has been fixed

pTargetCATI:

- variables "Specialized fair/congress: professional/personal reasons" [t272802_w1] and "Specialized fair/congress: Learned something new" [t272802_w1] as well as the corresponding variables for "Lectures" [t272802_w2, t272802_w2] and "Self-instruction programs" [t272802_w3,t272802_w3] in version 4.0.0 and earlier erroneously are not filled for all interviewees reporting the specific further education activity; this has been fixed
- variable names of variables "Father's mother: Country of birth" [t405240*] and "Mother's father: Country of birth" [t405230*] in dataset pTargetCATI erronneously had been flipped in version 4.0.0, also leading to slight inconsistencies in generated variables for migrational background; this has been fixed

spChild:

- all wide variables documenting cohabitation (*_w*) in version 4.0.0 and earlier with the focal child have been extracted and are now saved in the separate dataset "spChildCohab"

spChildCohab:

- new dataset containing chidl cohabitation spells that formerly had been saved in wide format inside of spChild

spEmp:

- version 4.0.0 and earlier did not contain coded occupational information for studentical employment episodes reported in wave 1; this has been fixed

Biography:

```
        − additional spells of type ”data edition gap” have been inserted to fill gaps
          between
              (a) the eighth birth day and the first reported episode and
              (b) the most recently reported episode and the most recent interview
                  date

====================================================
* Changes introduced to NEPS:SC5 by version 4.0.0 *
====================================================

General:

        − full translations have been added
        − wave 4 (online survey in semester 5) has been added
        − several minor bug fixes to data edition scripts have been introduced

pTargetCATI:

        − when generating variable ”Global self−esteem” [t66003a_g1] in the pTargetCATI
           dataset, variable ”Global self−esteem: competence” [t66003d] erroneously
          had been ignored;
              this has been fixed;
              t66003a_g1 can be re−generated in 3.1.0 using the following Stata
                  syntax:
              * −−−−−−−−−−−−−−−−−−−−−−−−−BEGIN Stata−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
              local target_variable t66003a_g1
              nepsmiss t66003a t66003b t66003c t66003d t66003e t66003f t66003g
                  t66003h t66003i t66003j
              tempvar t66003b_r t66003e_r t66003f_r t66003h_r t66003i_r rowmissings
              recode t66003b (1=5) (2=4) (3=3) (4=2) (5=1), generate(‘t66003b_r’)
              recode t66003e (1=5) (2=4) (3=3) (4=2) (5=1), generate(‘t66003e_r’)
              recode t66003f (1=5) (2=4) (3=3) (4=2) (5=1), generate(‘t66003f_r’)
              recode t66003h (1=5) (2=4) (3=3) (4=2) (5=1), generate(‘t66003h_r’)
              recode t66003i (1=5) (2=4) (3=3) (4=2) (5=1), generate(‘t66003i_r’)
              egen ‘rowmissings’=rowmiss(t66003a ‘t66003b_r’ t66003c t66003d ///
              ‘t66003e_r’ ‘t66003f_r’ t66003g ‘t66003h_r’ ‘t66003i_r’ t66003j)
              egen ‘target_variable’=rowtotal(t66003a ‘t66003b_r’ t66003c t66003d ///
              ‘t66003e_r’ ‘t66003f_r’ t66003g ‘t66003h_r’ ‘t66003i_r’ t66003j) if ‘
                  rowmissings’==0 & wave==3
              replace ‘target_variable’=−54 if wave!=3
              label variable ‘target_variable’ ”Global self−esteem”
              replace ‘target_variable’=−55 if missing(‘target_variable’)
              * −−−−−−−−−−−−−−−−−−−−−−−−−−−END Stata−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

xTargetCAWI:

        − as wave 3 data makes this a panel dataset, the filename has changed from ”
          xTargetCAWI” to ”pTargetCAWI”


====================================================
* Changes introduced to NEPS:SC5 by version 3.1.0 *
====================================================

General:
        − meta data in all datasets have been revised and updated where appropriate
        − English translation for all datasets except xTargetCAWI have been introduced
          to the data
        − end dates in episodes neglected in the panel interview erroneously contained
          the interview
```

```
                date of the panel wave instead of the first interview's date; this has
                    been fixed
        - 185 duplicate respondents have been identified by the survey institute;
                the redundant observations have been dropped from the data, resulting
                    in slightly smaller number of cases

pTargetCATI:
        - variables indicating migrational background (t400500_g1 through _g3) have
            been added

spVocTrain:
        - spell integration and recommendation (via variable tx20100) was erroneous;
            this has been fixed
        - spell linkage between waves 1 and 3 was erroneous; this has been fixed

spEmp:
        - spell linkage between waves 1 and 3 was erroneous; this has been fixed

Weights:
        - dataset containing weighting variables has been added

Basics:
        - dataset containing oversimplified, "flat" cross-sectional data on the cohort
            has been added;
                use for orientation, not for analyses!

xInstitution:
        - dataset containing detailed information on the targets' institutions has been
             added for onsite access in Bamberg
```