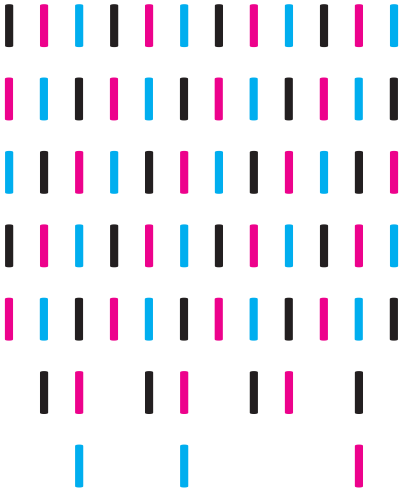

NEPS *SURVEY PAPERS*

Insa Grüttgen, Alexander Helbig and
Martin Ehlert

NEPS-DATA ON NON-FORMAL ADULT LEARNING ACTIVITIES - STRUCTURE; SPECIFICATIONS AND LINKING OF THE DATA OF THE STARTING COHORT 6

NEPS *Survey Paper* No. 101
Bamberg, February 2023

LIfBi
LEIBNIZ INSTITUTE FOR
EDUCATIONAL TRAJECTORIES

NEPS
National Educational Panel Study

Survey Papers of the German National Educational Panel Study (NEPS)
at the Leibniz Institute for Educational Trajectories (LIfBi) at the University of Bamberg

The NEPS *Survey Paper* series provides articles with a focus on methodological aspects and data handling issues related to the German National Educational Panel Study (NEPS).

They are of particular relevance for the analysis of NEPS data as they describe data editing and data collection procedures as well as instruments or tests used in the NEPS survey. Papers that appear in this series fall into the category of 'grey literature' and may also appear elsewhere.

The NEPS *Survey Papers* are edited by a review board consisting of the scientific management of LIfBi and NEPS.

The NEPS *Survey Papers* are available at www.neps-data.de (see section “Publications“) and at www.lifbi.de/publications.

Contact: German National Educational Panel Study (NEPS) – Leibniz Institute for Educational Trajectories – Wilhelmsplatz 3 – 96047 Bamberg − Germany − contact@lifbi.de

NEPS-Data on Non-Formal Adult Learning Activities – Structure, Specifications and Linking of the Data of the Starting Cohort 6

Insa Grüttgen, Alexander Helbig, Martin Ehlert

Berlin Social Science Center

E-Mail address of authors:

insa.gruettgen@wzb.eu, alexander.helbig@wzb.eu, martin.ehlert@wzb.eu

Bibliographic data:

Grüttgen, I, Helbig, A & Ehlert, M (2023). *NEPS-Data on Non-Formal Adult Learning Activities – Structure, Specifications and Linking of the Data of the Starting Cohort 6* (NEPS Survey Paper No. 101). Leibniz Institute for Educational Trajectories, National Educational Panel Study. https://doi.org/10.5157/NEPS:SP101:1.0

NEPS-Data on Non-Formal Adult Learning Activities – Structure, Specifications and Linking of the Data of the Starting Cohort 6

Abstract

Starting cohort 6 of the National Education Panel Study (NEPS) collects, among other things, longitudinal data on non-formal learning activities for adults. This paper provides guidance for researchers interested in analyzing this data. We describe the available data on non-formal learning activities among adults, i.e. further education courses that do not lead to formal educational certificates, and their structure within the NEPS-data. Furthermore, we introduce a Stata-do-File and an R script to connect the different files on further education and prepare the data for further analysis. The syntax is written and published in order to serve as an overview and starting point for users who want to work with the NEPS-data on further education. The generated data set contains detailed data on further education courses and shows a possible linking of the different data on further education in the NEPS-data. It can be used to link further information such as longitudinal data of life courses or socio-demographic information.

Keywords

further education, non-formal learning, courses, generated file, starting cohort 6

This paper is supplemented by a STATA-do file and a R-syntax which can also be found in the appendix. The files can be downloaded from https://www.lifbi.de/Supp/SP-101.

1. Introduction

The National Education Panel Study collects extensive data on further education and lifelong learning, including formal, non-formal and informal learning activities. In this paper we present the structure of the data on non-formal learning activities. More precisely, the NEPS aims to survey participation in non-formal learning activities, which are self-imposed but other-directed (Bäumer et al. 2011, Janik et al. 2016). To survey non-formal learning activities, the NEPS collects data about participation in courses and trainings.

In the adult starting cohort 6, respondents are asked about further education courses and trainings in different contexts, and there is manifold information on different aspects of (some of) these courses. A particular strength of the NEPS-data is the collection of longitudinal data on the interviewees' life courses, such as episodes of (un-)employment, education, parental leave, civilian/military service. For each of those episodes, the respondents are asked whether they participated in any courses or trainings. This allows analysts to precisely contextualize the participation and specific aspects of non-formal adult education. Furthermore, the reference to life episodes serves as a stimulus to facilitate the recollection of visited courses and trainings (Janik et al. 2016, 385). In addition, the respondents are asked about additional courses and trainings that had not yet been mentioned. Finally, out of all the reported courses and trainings one is chosen randomly and additional detailed questions on this course follow. On top of that, there are also some courses that were originally reported as vocational training but actually are considered as non-formal courses in the NEPS. In sum, this creates a data structure in which information on adult education courses is spread over multiple data sets.

To show how this data structure emerges and in order to facilitate the use of the NEPS data on adult education, we will first explain the general structure of the data and the available information as well as some noteworthy specifications.

In addition, we created a Stata-do-file and an R script to compile a data set with information from the different data sets on non-formal adult education courses. This includes details on reasons for visiting a course, whether the costs were covered by the employer, and whether the course was visited during working hours or free time in order to classify the courses as private or job-related. The created data set can therefore be used as a starting point for possible analysis of the adult education data by showing how the different information can be linked. We then demonstrate a possible way to deal with changes within the data collection over different waves of the National Education Panel Study, including the creation of weights. In addition, we transform the data set in a person-wave format with aggregated data on, e.g. the number of visited courses per person and wave, and demonstrate some exemplary descriptive analyses.

2. General Structure of the NEPS-Data on Non-Formal Learning

The extensive questionnaire of the NEPS includes the collection of information on further education courses at various points during the interview. All of the reported courses are included in the generated data set *FurtherEducation*. More detailed information can be found in the different data sets *spvocTrain*, *spcourses*, *spfurtherEdu1* and *spfurtherEdu2*, which are described in detail below.

2.1 Data Structure: SUF Wave 13 (Survey From 2020/2021)[1]

At the beginning of the survey the interviewees are asked about different life course episodes since the last interview, e.g. whether they have (continuously) been unemployed, in education, employed or on parental leave. After several questions about those contexts, they are asked whether they participated in any courses during that time[2]. If so, there is a follow-up with some basic questions about these courses, such as the course's content and whether it was attended for private or professional reasons. The information on courses that have been reported in the context of a life course episode is stored in the *spCourses*-data set. The variable *sptype* provides information on the life course context in which the courses were reported (military service, employment, unemployment, parental leave, gap or license courses in vocational training). An overview of the connection of the different data sets can be found in Figure 1.

Some non-formal courses are also collected in the questionnaire module on vocational training and university episodes. Most of the recorded episodes and activities in this module are formal education, as they lead to generally recognized formal educational certificates. However, certain courses mentioned in this module belong to non-formal education and are therefore also listed in the *FurtherEducation* data set. This mainly includes license courses, e.g. to become a certified taxi driver or welder, but also some trainings recognized by the Chamber of Commerce and Industry (Industrie- und Handelskammer, IHK) or other courses. Up until wave 10, detailed information on these courses was stored in the *spVocTrain*-data set. In wave 11, the license courses as well as the IHK courses were not recorded in the context of vocational training but as part of further education, which is why more detailed information on these courses is stored in *spFurtherEdu1* for this wave (FDZ-LIfBi 2022, 147). From wave 12 onwards, all reported license courses are directly treated as non-formal education courses (just like courses reported during other life course episodes) and are therefore included in *spCourses* (FDZ-LIfBi 2022, 147, 148) with the same questions on some basic information of these courses[3].

After the life course modules, respondents are asked whether they have taken any other courses apart from the ones already mentioned[4]. For all these courses there are also some basic questions on the content and duration. The information on these courses is stored in the data set *spFurtherEdu1*.

Finally, further information about randomly selected courses from all of the above-mentioned courses is stored in the *FurtherEdu2*-data set. Two courses were randomly selected for further

[1] Note: This paper uses data from the National Educational Panel Study (NEPS; see Blossfeld & Roßbach, 2019). The NEPS is carried out by the Leibniz Institute for Educational Trajectories (LIfBi, Germany) in cooperation with a nationwide network. The data used in this paper relates to the adult starting cohort 6 (NEPS Network, 2021; 2022).
Because of the first survey ALWA (Arbeiten und Lernen im Wandel) in 2007/2008, the first NEPS-wave equals the second wave in the data. This paper follows the variable *wave*, therefore the here mentioned 13th wave corresponds with the 12th NEPS-wave.

[2] E.g. in the context of employment: „While working as <26109> from <26122> until today, did you attend training programs or courses, which you have not yet mentioned?" „Haben Sie während Ihrer Tätigkeit als <26109> von <26122> bis heute Lehrgänge oder Kurse besucht, von denen Sie bisher noch nicht berichtet haben?"

[3] Only very few other vocational trainings that are considered non-formal education, mainly IHK-courses, are still stored in spVocTrain from wave 12 onwards.

[4] „Let's return to the subject of further training. Up until now you have stated that, since the last interview, you attended the following courses or training programs: <[Kursliste]> Since the last Interview, have you, in addition to this, i.e. from <intmPRE/intjPRE> to the present, attended courses or training programs that you have not yet mentioned?" / „Kommen wir noch einmal zurück auf das Thema Fortbildung. Bisher haben Sie berichtet, dass Sie seit dem letzten Interview folgende Kurse oder Lehrgänge besucht haben: <[Kursliste]> Haben Sie darüber hinaus seit dem letzten Interview, also von <intmPRE/intjPRE> bis heute, Kurse oder Lehrgänge besucht, von denen Sie bisher noch nicht berichtet haben?"

questions up to wave 10. From wave 11 onwards, only one course is chosen for further questions. The additional questions asked for the random courses have changed over time (see section "Available information on non-formal learning"). Courses included in this module are randomly drawn from all completed courses reported in the context of a life course episode (*spCourses*), from additional courses reported afterwards (*spFurtherEdu1*), or from reported license courses (*spVocTrain*/*spCourses* since wave 12).

Since respondents were not asked detailed questions about further education in the very first wave in 2007/08, the so called ALWA-wave ("Working and learning in a Changing World"), we focus here on information on further education courses available since the second wave (2009/10).

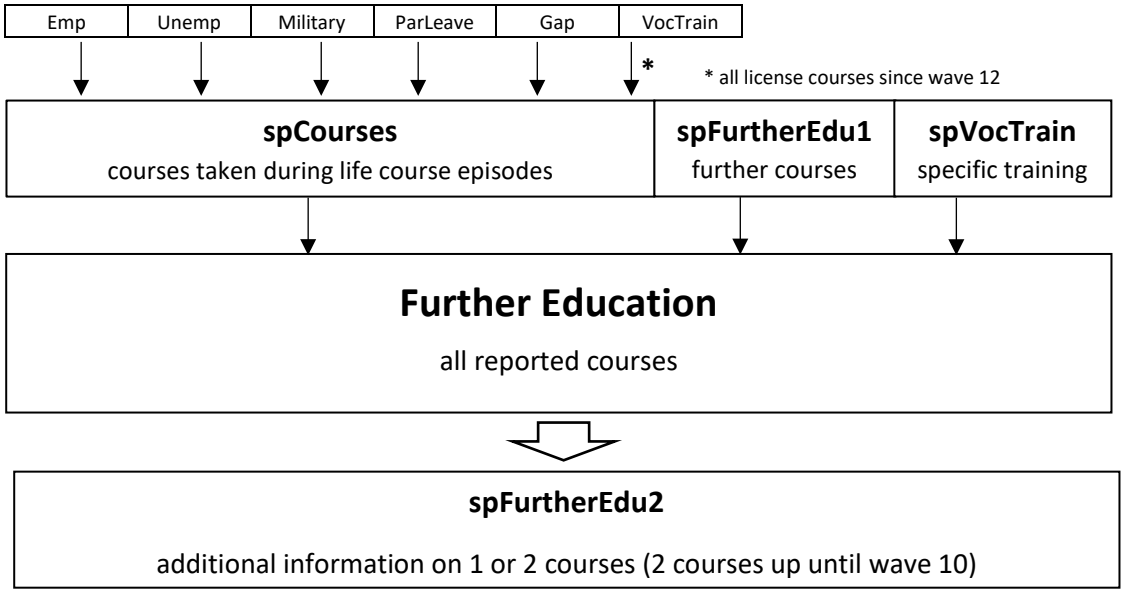

Figure 1. Structure of the data on further education in the Scientific Use Files. Own presentation.

All of these courses and trainings from the different data sets, reported at different points in the interview, are stored in the *FurtherEducation* data set, which therefore provides a good basis for any analysis of non-formal adult education. Additional information can be linked from other data sets. An overview of the different data sets and their connection can be found in Figure 1. Figure 2 shows an exemplary data extract from the *FutherEducation* data set. It illustrates that the person with the *ID_t* 8000413 reported five courses in their interview in wave 9 (2016/2017). The variable *tx28200* contains information on whether the given course was reported as vocational training (*spVocTrain*), in the context of a life course episode (*spCourses*) or subsequently as an additional course (*spFurtherEdu1*). Four of the courses were reported as part of a specific life course episode (*tx28200* = *spCourses*), in this case during two different employment episodes. This can be seen in the two different *splink* values. The two first numbers of the *splink* variable also indicate the life course context in which the courses were reported[5], in this case these were episodes of employment because they start with the number 26 (*splink* = 260005 & 260007). One additional course was reported after the questions on life course episode (*tx28200* = *spFurtherEdu1*). The *course* variable identifies the

[5] The first two numbers of the *splink* variable represent the different life courses: 24 = vocational training, 25 = military, 26 = employment, 27 = unemployment, 29 = parental leave, 30 = gap.

different courses within the wave and is needed to merge further information from the data sets *spCourses*, *spFurtherEdu1,* and possibly *spFurtherEdu2* for the randomly chosen courses for which additional questions are asked.

ID_t	wave	splink	course	tx28200
8000413	2016/2017 (8th NEPS main survey)	260005	902	spCourses
8000413	2016/2017 (8th NEPS main survey)	260005	901	spCourses
8000413	2016/2017 (8th NEPS main survey)	260007	903	spCourses
8000413	2016/2017 (8th NEPS main survey)	260007	904	spCourses
8000413	2016/2017 (8th NEPS main survey)	.	905	spFurtherEdu1

Figure 2. Exemplary data extract from FurtherEducation.
Source: NEPS SC6 SUF 12.

2.2 Available Information on Non-Formal Learning

There is some general information that is available for all courses and can therefore be found in the *FurtherEducation* data set. This includes, for example, information on the content and duration of the course or the start and end dates. For the courses from *spCourses* and *spFurtherEdu1* the course dates are based on either the life course episodes or the time period between the last and the current interview, so no specific start and end dates are given, but rather the time frames in which they took place. Only in wave 2, information on timing was directly surveyed for courses collected as part of *spFurtherEdu1.*

The data set *spFurtherEdu2* stores the additional information on randomly chosen courses, which includes, for example, questions on the financing of the course, the provider of the course, and an item set for course evaluation. Regarding the more detailed information for randomly chosen courses, there was a shift in wave 11. Previously, two courses were selected, but since wave 11 in 2018/19 it is only one with detailed follow-up questions. In exchange, there are six additional variables that are no longer asked only for the randomly selected courses, but for every course stated, and are therefore stored in the data sets *spCourses* and *spFurtherEdu1*. This includes information on certification[6], obligation and motivation as well as whether the course was taken for private or professional reasons. Table 1 gives an overview of all the available information in the different data sets and also shows the adjustment in wave 11, which means that information on the six shifted questions is stored in different data sets depending on the wave.

[6] The variable on certification is available in spFurtherEdu1 from wave 11 on, but only from wave 12 on for *spCourses*

Table 1

Available Information Before and After Changes in Wave 11. Own presentation.

Wave 2-10 (2009/10-2017/18)

for **all** courses:

spCourses / **spFurtherEdu1** / **FurtherEducation**

Variables:

- Content
- Duration in hours
- Participation discontinued or until the end
- Course ongoing
- Course attendance start and end (date/interval)

for **two** randomly selected courses:

spFurtherEdu2

Variables:

- Professional or private reasons
- Motivation
- Obligation
- Who obligated?
- Certification
- Type of certificate
- Parallel to (un-)employment (from wave 4 on)
- During working or free time
- Own financial contribution
- Course costs employer
- Course costs employment agency
- Provider
- External or internal staff
- Item battery on course assessment
- Information from personal environment (from wave 7 on)
- Financial support & care support (wave 3)

Since wave 11 (2018/19)

for **all** courses:

spCourses / **spFurtherEdu1** / **FurtherEducation**

Variables:

- Content
- Duration in hours
- Participation discontinued or until the end
- Course ongoing
- Course attendance start and end (date/interval)

spCourses / **spFurtherEdu1**

Variables:

- Professional or private reasons
- Motivation
- Obligation
- Who obligated?
- Certification
- Type of certification

for **one** randomly selected course:

spFurtherEdu2

Variables:

- Parallel to (un-)employment (from wave 4 on)
- During working or free time
- Own financial contribution
- Course costs employer
- Course costs employment agency
- Provider
- External or internal staff
- Item battery on course assessment
- Information from personal environment (from wave 7 on)

3. Stata-Do-File

In order to conduct an extensive analysis with the NEPS data on further education courses, it is likely necessary to link the different data sets of the Scientific Use Files for the adult starting cohort 6. To facilitate the users' entry, we created a Stata-do-file (see Appendix A) that generates a data set that contains information from the *spCourses*, *spFurtherEdu1*, and *spFurtherEdu2* data sets, merged with *FurtherEducation*. This creates an exemplary data set that can serve as a basis for users' data preparation and illustrates the described structure of the available information by putting it into practice. Additionally, we create a variable indicating whether the courses were private or job-related, and if job-related, whether they were firm-sponsored or not, by linking the corresponding information and showing a possible way to handle the difference in availability of information before and after wave 11 (see Table 1).

3.1 Linking Data of Further Education

The generated data set *FurtherEducation* serves as the starting point, as it contains all courses that respondents have reported at different points during the interview. The data set is in a long format with one observation per course (see Figure 2) and includes basic information as well as some variables showing the context in which the course was reported (*splink*, *tx28200*). It allows to link the data with information from other data sets (using *course* as the linkage variable).

Course variable missing in FurtherEducation

One aspect that users will come across when using the *FurtherEducation* data set is that for some courses the variable course is missing. However, this variable is necessary to link data from *spFurtherEdu2* to *FurtherEducation*. The missing values on this variable stem from courses from *spVocTrain* (*tx28200* = *spVocTrain*). These are courses that were reported as vocational training but classified as non-formal courses rather than formal education. As mentioned before, the courses from *spVocTrain* are mostly license courses that are taken in order to receive a license, e.g. a taxi driving or welding license. These courses are included in the random selection process of courses with additional detailed questions. On top of that, the research data center retrospectively classifies some courses as non-formal courses rather than formal training. These courses from *spVocTrain* as well as courses that ended more than 12 months prior to the interview do not receive a course number. They are not included in the random selection process and hence, these courses cannot be linked with the *spFurtherEdu2* data set. No courses from *spCourses* or *spFurtherEdu1* have a missing on course though, so that the variable can be used for linking data from those data sets.

Generating course and interview dates

In the do-file we first create start and end variables in a year-month-format and drop courses with missing dates as well as all courses from wave 1 because respondents were not asked detailed questions about courses in this first ALWA-wave in 2007/08.

```
gen start=ym(tx2821y, tx2821m)
      format start %tm
gen ende=ym(tx2822y, tx2822m)
      format end %tm

drop if wave==1
```

We then add the interview dates of each respondent from the *CohortProfile* data set.

```
// Interview date from CohortProfile data set

      preserve

use ID_t wave tx8600y tx8600m using "$DATA\SC6_CohortProfile_D_$suf.dta",
clear
drop if wave == 1
gen interv=ym(tx8600y, tx8600m) // interview date in each wave
format interv %tm
tempfile intd
save `intd'

      restore

merge m:1 ID_t wave using `intd', keepusing(interv) keep(master match) nogen
```

When comparing course and interview dates, one can see that some end dates of the courses are dated a very long time before the interview dates. This is partly due to transfer errors[7]. Another reason is that in the second and fourth wave, part of the sample was interviewed for the first time and, for these individuals, participation in vocational training (courses from *spVocTrain*) was included in the survey retrospectively for the entire life course. Additionally, respondents in wave 2 could technically also report non-formal courses (courses from *spFurtherEdu1*) for their whole life course, which happened four times. In general, larger time gaps may also exist due to respondents who missed one wave and therefore had a longer time period since the last interview, or respondents who participated in one wave at the beginning of the field period and at the end of the field period for the next wave, which also leads to a larger but plausible time frame. To maintain a comparability between the different waves, one could delete courses that ended more than 18 months before the respective interview date[8]. This decision, of course, depends on the individual research interest. In our do-file, we keep all reported courses in the data set, but provide code for the mentioned restriction. Another possible correction is to control for the duration of the reference period when using training information in multivariate models, as suggested by Rzepka (2016).

```
** Problem: some courses have enddates a long time before the interview

tab tx2822y wave, mis
gen timegap = interv-ende if interv!=. & ende!=.



**# Irregular end dates

tab tx2822y wave, mis
gen timegap = interv-ende if interv!=. & ende!=.
fre timegap

fre timegap if tx28200 == 31 & wave == 2
*drop if timegap > 12 & timegap != . & tx28200 == 31 & wave == 2

fre timegap if tx28200 == 35
*drop if timegap > 18 & timegap != . & tx28200 == 35 // Assuming a maximum of
18 months between interviews of two consecutive waves
```

[7] End dates that end after the respective interview date are also due to transfer errors. These data errors from courses with end dates after the interview dates should mostly be corrected in the upcoming versions of Scientific Use Files.

[8] Taking 18 months as a threshold ensures that courses of respondents who just had a longer period of time between two interviews aren't excluded. An analysis of the data shows, that in the other waves - apart from wave 2 and 4 - 99.8% of all courses ended up to 18 months prior to the interview.

```
tabstat timegap if tx28200 ==24, by(wave)
*drop if timegap > 18 & timegap != . & inlist(wave,2,4) & tx28200==24
*drop if timegap > 18 & timegap != . & tx28200==24
```

Merging information from spFurtherEdu2

We then add detailed data from the *spFurtherEdu2*-data set, which contains information from the loop of further questions on one (or two) randomly chosen course(s). This data set is also in long format and can be merged using the course-variable. Note that we use a m:1 merge to account for the missing values on the course variable, as described earlier.

```
merge m:1 ID_t course using "$DATA/SC6_spFurtherEdu2_D_$suf.dta"
```

Merging information from spFurtherEdu1 and spCourses

Since not all courses were selected for additional questions, there are many courses that cannot be merged between *spFurtherEdu2* and *FurtherEducation*. Even more importantly, starting in wave 11, some information, e.g. on private or professional reasons, is included for all reported courses (see Table 1) and is therefore not available in the *spFurtherEdu2*-data set, but in *spCourses* or *spFurtherEdu1*. Consequently, for all courses this information has to be linked from those data sets from wave 11 onwards.

```
      preserve
use "$DATA/SC6_spFurtherEdu1_D_$suf.dta", clear

drop if wave<11 // information only to be added for courses from wave 11 on

replace t272043 = t272043_v1 if wave==11
recode t272043 (2 = 4)
label def de5626ext1 3 "beides, Teilnahmebescheinigung und anerkannte
Lizenz", modify
label val t272043 de5626ext1

tempfile fe1
save `fe1'
      restore

merge m:1 ID_t wave course using `fe1', keepusing(t279040 t279041 t279042
t279043 t272043) gen(_mergewb) update
```

Note that the data set *spCourses* is in long format with one observation per life course episode. The courses, however, have a wide format with up to five reported courses for every life course episode. For Example, in the data extract in Figure 3 the respondent with the *ID_t* 8000413 reported two employment episodes in the 9th wave (*splink* 260005 & 260007) in which courses were taken. Information on the different courses in one observation is marked by the suffixes _w1, _w2, and so on. The data extract also shows that the information whether the course was visited for personal or professional reasons, or both, has only been collected for all courses since wave 11, as variable *t279030* is “missing by design” before that. Additionally, you can see that *ID_t* 8000516 only visited one course during an unemployment episode in wave 11, as there is missing information in the data for the second course.

ID_t	wave	splink	course~1	course~2	t279030_w1	t279030_w2
8000413	2014/2015 (6th NEPS main survey)	260005	701	702	missing by design	missing by design
8000413	2016/2017 (8th NEPS main survey)	260005	901	902	missing by design	missing by design
8000413	2016/2017 (8th NEPS main survey)	260007	903	904	missing by design	missing by design
8000413	2018/2019 (10th NEPS main survey)	260008	1101	1102	both	both
8000516	2018/2019 (10th NEPS main survey)	270003	1101	.	for professional reasons	.

Figure 3. Exemplary data extract from spCourses.
Source: NEPS SC6 SUF 12.

To link the data from *spCourses* with the *FurtherEducation* data set, it first has to be transformed into long format, with one observation per course.

```
      preserve
use "$DATA/SC6_spCourses_D_$suf.dta", clear

bys ID_t wave: gen n =_n
keep ID_t wave t279030_w* t279031_w* t279032_w* t279033_w* t272043_w*
course_w* n
reshape long t279030_w t279031_w t279032_w t279033_w t272043_w course_w,
i(ID_t wave n) j(course_nr)
drop if course_w==. // course_w missing when no (further) course reported

rename course_w course // rename for merging
rename t279030_w t279040
rename t279031_w t279041
rename t279032_w t279042
rename t279033_w t279043
rename t272043_w t272043

drop if wave<11

tempfile co
save `co'
      restore

merge m:1 ID_t wave course using `co', keepusing(t279040 t279041 t279042
t279043 t272043)gen(_mergeco) keep(master match match_update) update
```

3.2 Generating a Variable of the Courses' Classification as Job-Related or Private

After merging the required information from all four data sets, we create variables to classify courses as private or job-related, as well as firm-sponsored or non-firm-sponsored. These variables are important indicators for many further education-related analyses as well as good examples of how to deal with the varying availability of survey data over several waves. Following the report of the Adult Education Survey (AES) on further Education (cp. Bilger et al. 2017), we first distinguish course attendance as either private or job-related, depending on the reason respondents report as private or professional (*t279040*). If respondents choose the "both" option, we classify the course as job-related.

```
gen jobrelated = .
replace jobrelated = 0 if t279040==2
replace jobrelated = 1 if t279040==1 | t279040==3
```

For the courses classified as job-related, there is a further specification as firm-sponsored if either the costs were (at least partly) covered by the employer (*t279046*) or the course was

(at least partly) attended during working hours (*t279044*). If neither is the case, the course is classified as non-firm-sponsored.

```
gen firmsponsored = .
replace firmsponsored = 1 if jobrelated ==1 & (inlist(t279044,1,2) |
inlist(t279046,1,2))
replace firmsponsored = 0 if jobrelated ==1 & t279044==3 & t279046==3
```

The variables we use to specify courses as (non-)firm-sponsored are only available for those courses that were randomly chosen for further questions. The same applies for the variable on professional or personal reasons for participation (*t279040*) up until wave 10.

Addressing the Issue of Missing Information for Some Courses

There are several ways to address the problem of missing additional information on courses:

1. Users can use only courses with non-missing information. This drastically reduces the number of courses. However, if the interest is mainly on the individual level, the loss is mostly not severe, especially for individuals with only a few courses. There is a problem, though, if individuals have taken many courses and the one(s) selected is(are) not a good representation. In addition, this may lead to bias if the information becomes available for all courses in later waves.

2. Users can also use multiple imputation to infer the missing information. Since the missing information is random by construction, imputation is straightforward. However, it is also time-consuming and complicates the analyses. For illustration, see the application in Ehlert (2017) and Ebner and Ehlert (2018).

3. Users could logically impute some of the missing information based on available data about the context in which the courses were reported. An example of how this could be done is presented below.

Missing information about whether the course was attended for job-related reasons can be deduced from the life course context in which the course was reported. To achieve this, we first get that information from *spCourses* by merging the variable *sptype*[9]. Then, we generate a new variable that copies the already existing variable used to differentiate between private and job-related courses. For all the courses with missing information, we assign them as job-related if they were reported as a vocational training or in the context of an employment or unemployment episode (coming from *spVocTrain* or coming from *spCourses* with the *sptype* Employment, Unemployment or Vocational Training).

```
      preserve
use "$DATA/SC6_spCourses_D_$suf.dta", clear

bys ID_t wave: gen n =_n
keep ID_t wave course_w* n sptype
reshape long course_w, i(ID_t wave n) j(course_nr)
drop if course_w==. // course_w missing when no (further) course reported
rename course_w course // rename for merging
```

[9] It is also possible to use the variable *splink* which is already available in the *FurtherEducation* data set by using the first two numbers of *splink* as an indicator.

```
tempfile co2
save `co2'
      restore

merge m:1 ID_t wave course using `co2', keepusing(sptype) keep(master match)
gen(_mergeco2)

generate jobrelated_g = job-related
replace jobrelated_g=1 if tx28200==24 & job-related ==.
replace jobrelated_g=1 if tx28200==35 & inlist(sptype,24,26,27)
& jobrelated ==.
```

This assignment is based on frequency distributions on the share of courses in these contexts which respondents classified as job-related instead of private – where the information was available – over waves 2-12 (see Table 2). For example, for all courses mentioned in the context of an unemployment episode, 87% were declared as job-related when this information was in fact recorded. It is also theoretically plausible to consider courses from these contexts as job-related, but depending on the research question and perspective users can adapt or alter the classification.

Table 2

Share of Courses in Different Contexts That Were Reported as Job-Related (Where Information Available)

context	Employ-ment	Unemploy-ment	Parental Leave	Retire-ment	Gap	Military / Voluntary Service	Vocational Training	Further courses
Wave 2-12	96,3%	87,1%	67,6%	11,1%	42,2%	64,9%	88,9%	38,8%
N	39306	1120	102	650	436	37	1078	12447

Source: NEPS SC6 SUF 12. Own analyses.

Note that this logically imputed variable should be interpreted with caution. Depending on the research question, the advantages of larger numbers of cases may be offset by the imprecise measurement.

3.3 Weights Accounting for Random Selection of Courses

For analyses on the level of the courses, we can also derive weights that account for the random selection of courses within individuals. The idea is that information about courses from individuals with many courses should be weighted higher because it represents many courses. In addition, this method partly corrects the break in the time-series caused by the reduction of randomly drawn courses and the shift of variables from the random courses to all courses in wave 11 (see Table 1). The construction of the weights is adapted from the course weights in the German AES SUF (Kantar 2020). The weights can only be used for courses selected in the loop for further questions during the interview, that is, courses that don't have a missing value on the course variable.

We first generate a weight for information that is only available in *spFurtherEdu2* for all waves (see bottom of Table 1). Here, we only have to consider the shift from two courses to one course for the additional questions. This means that up until wave 10, the chance of being included in random selection was 100% if only one or two courses were reported, and correspondingly lower for each additional course. From wave 11 on, the chance of being included for further questions is 100% if only one course is reported, 50% if two courses are reported and so on.

```
gen counter = 1 if course!=.
bysort ID_t wave: egen number_loop = total(counter)

gen cweight_FE2 = . if course==.
replace cweight_FE2 = 1 if wave<11 & inlist(number_loop,1,2) & course!=.
replace cweight_FE2 = number_loop/2 if wave<11 & number_loop>2 & course!=.

replace cweight_FE2 = number_loop if wave>=11 & course!=.
```

The next weight is created for those variables that are available in *spFurtherEdu2* up until wave 10 and from wave 11 onwards in *spFurtherEdu1* and *spCourses*, e.g. *t279040* (see Table 1). Up to wave 10, these variable weights are equal to the cweight_FE2 variable. From wave 11 onwards the weights for these variables get the value 1, since the probability that the information is available is 100% for all courses.

```
gen cweight_change = . if course==.
replace cweight_change = 1 if wave>=11 & course!=.
replace cweight_change = cweight_FE2 if wave<11 & course!=.
```

Two example analyses show small but no major differences when using the created weights. Depending on the research question, the use of such weights should be considered, especially when the availability of information changes between waves and if information is required that goes beyond the classification of job-related vs. private courses (because in this case logical imputation, as shown above, is an alternative to using weights).

3.4 Transformation into person-wave-format and merging with *CohortProfiles*

To generate a data set that can be used as a starting point for analyses on the individual level, we reshape the course data into a wide format with one observation per person and wave. Before that, we also generate some exemplary summarizing variables: The number of courses attended, total training hours and job-related courses per person and wave.

```
bysort ID_t wave: gen nr = _n // course number
drop _merge* number_loop counter nepswave sptype

** some aggregated statistics on the person-wave level

bysort ID_t wave: gen n_courses = _N // courses per person and wave
nepsmiss tx28203
bys ID_t wave: egen hours_courses = total(tx28203) // total duration of
courses per person and wave
bys ID_t wave: egen n_jobtotal = total(jobrelated) // job-related courses per
person and wave
bys ID_t wave: egen n_jobtotal_w = total(jobrelated*cweight_change) // job-
```

```
related courses per person and wave extrapolated using the weight adjusting
for random drawing
bys ID_t wave: egen n_jobtotal_g = total(jobrelated_g) // job-related courses
(including the imputed ones) per person and wave
```

We reduce the file to one observation for each person-wave combination. The aggregated information of courses is preserved.

```
** create a data file with one observation per person and wave, keeping wave-
aggregated variables

bysort ID_t wave: gen n = _n
keep if n==1
keep ID_t number wave interv n_courses hours_courses n_jobtotal n_jobtotal_g
```

Since the data set so far only includes respondents who reported at least one course, we then merge the data with the data set *CohortProfile,* in order to get information on e.g. participation quotas. *CohortProfile* holds observations for all respondents for all waves, including information on whether they participated in the corresponding wave (*tx80220*). We use the information of this variable to reduce the data set to respondents who participated in the corresponding wave. We further remove the ALWA wave, as there is no information on courses. Now, all person-wave combinations with no course participation have system missings because they have no entry in the data files that record course participation. We therefore recode the variables counting the courses to 0 for cases with system missings indicating no participation. The resulting data set in a person-wave-format is therefore a good starting point for any analysis on the individual level (see FDZ-LIfBi 2022, p. 64).

```
// merge with CohortProfile data set which includes all respondents over all
waves

merge 1:1 ID_t wave using "$DATA\SC6_CohortProfile_D_$suf.dta", nogen

* keep only those who participated in the wave
keep if tx80220 == 1

drop if wave == 1

foreach var of varlist n_courses n_jobtotal n_jobtotal_g {
      replace `var' = 0 if `var' == .
}
```

Based on this information, we can also calculate dummies for course participation.

```
recode n_courses (1/max = 1), gen(nf_dummy) // DUMMY: courses in this wave
yes/no
recode n_jobtotal (1/max = 1), gen(nfjr_dummy) // DUMMY: job-related courses
in this wave yes/no
recode n_jobtotal_g (1/max = 1), gen(nfjr_g_dummy) // DUMMY: job-related
courses (including the imputed ones) in this wave yes/no
```

3.5 Example analyses

Finally, we use the generated dataset to calculate some descriptive statistics as examples of what can be done with it. Please note that the data presented here should not be interpreted because of the technical issues described above, which make comparisons between waves difficult. It is recommended to modify the data depending on the research question in order to obtain meaningful results.

First, we calculate participation rates for each wave.

```
* percentage of course participation
graph bar (mean) nf_dummy, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3
"2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8
"2016/2017" 9 "2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021")
label(angle(90))) ytitle(share of participants) title("All courses")
name(all, replace)

* percentage of job-related course participation (without logical imputation)
graph bar (mean) nfjr_dummy, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3
"2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8
"2016/2017" 9 "2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021")
label(angle(90))) ytitle(share of participants ) title("Job-related courses")
name(perc_jr1, replace)

* percentage of job-related course participation (with logical imputation)
graph bar (mean) nfjr_g_dummy, over(wave, relabel(1 "2009/2010" 2 "2010/2011"
3 "2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8
"2016/2017" 9 "2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021")
label(angle(90))) ytitle(share of participants) title("Job-related courses
(log. imputation)") name(perc_jr2, replace)
```

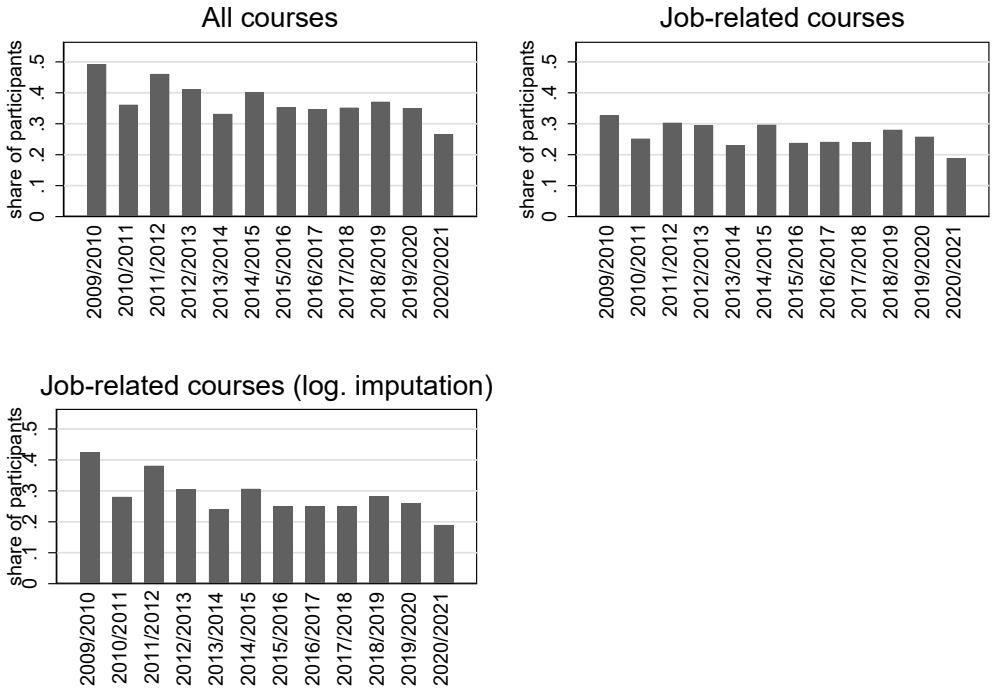

Figure 4. Participation rates in courses by wave.
Source: NEPS SC6 SUF13.0.0, own calculations

Figure 4 shows that participation varies and is especially high in the waves 2009/2010 and 2011/12. This is partly due to new participants in the survey reporting courses that date back longer. In addition, we can observe that the logical imputation of job-related courses does not change the participation rates much.

Second, we calculate the average number of courses per person and wave among the participants.

```
* average number of courses per person over time
graph bar (mean) n_courses if n_courses >0, over(wave, relabel(1 "2009/2010"
2 "2010/2011" 3 "2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7
"2015/2016" 8 "2016/2017" 9 "2017/2018" 10 "2018/2019" 11 "2019/2020" 12
"2020/2021") label(angle(90))) ytitle(average no. of courses) name(raw,
replace) title("All")

* average number of job-related courses per person over time (without logical
imputation)
graph bar (mean) n_jobtotal if n_courses >0, over(wave, relabel(1 "2009/2010"
2 "2010/2011" 3 "2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7
"2015/2016" 8 "2016/2017" 9 "2017/2018" 10 "2018/2019" 11 "2019/2020" 12
"2020/2021") label(angle(90))) ytitle(average no. of courses ) name(jr1,
replace) title("Job-related")

* average number of job-related courses per person over time (with weights
adjusting for random selection of courses)
graph bar (mean) n_jobtotal_w if n_courses >0, over(wave, relabel(1
"2009/2010" 2 "2010/2011" 3 "2011/2012" 4 "2012/2013" 5 "2013/2014" 6
"2014/2015" 7 "2015/2016" 8 "2016/2017" 9 "2017/2018" 10 "2018/2019" 11
"2019/2020" 12 "2020/2021") label(angle(90))) ytitle(average no. of courses
) name(jr2, replace) title("Job-related with weights")

* average number of job-related courses per person over time (with logical
imputation)
graph bar (mean) n_jobtotal_g if n_courses >0, over(wave, relabel(1
"2009/2010" 2 "2010/2011" 3 "2011/2012" 4 "2012/2013" 5 "2013/2014" 6
"2014/2015" 7 "2015/2016" 8 "2016/2017" 9 "2017/2018" 10 "2018/2019" 11
"2019/2020" 12 "2020/2021") label(angle(90))) ytitle(average no. of courses )
title("Job-related with logical imputation") name(jr3, replace)
```

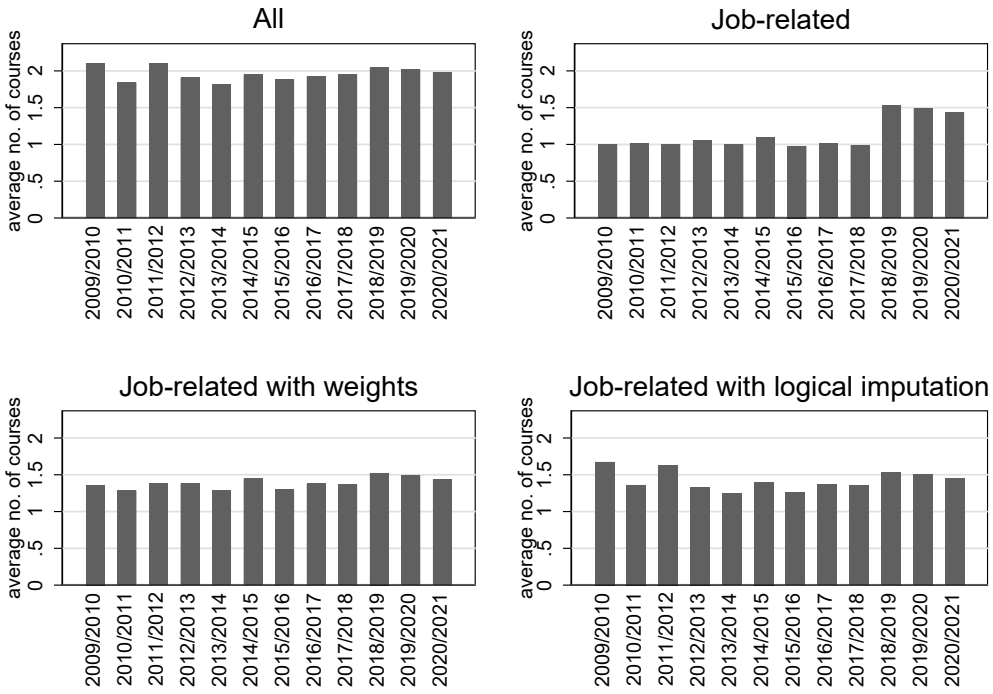

Figure 5. Average number of courses among participants for each wave.
Source: NEPS SC6 SUF 13.0.0, own calculations.

The shift from surveying the motivation for participation only for randomly selected courses to all courses is clearly visible between waves 2017/18 and 2018/19. The weights and the logical imputation both smooth the time-series.

4. R Script

We also provide an R script that generates the same data (see Appendix B).

5. Conclusion

The NEPS SC6 offers a wide range of information on non-formal training courses. However, the information is partly scattered across different data sets. This serves the purpose of allowing researchers to compile their data sets in many different ways, depending on their research question. Nevertheless, it also creates complexity. With this paper, we have shown an example for a data retrieval for a data set on the course level and on the person-wave level. In addition, we provided some solutions to issues arising from changes in the questionnaire over time and common data manipulations. We hope that this guide is helpful for the scientific community.

References

Bäumer, T., Preis, N., Roßbach, H.-G., Stecher, L., & Klieme, E. (2011). 6 Education processes in life-course-specific learning environments. Zeitschrift Für Erziehungswissenschaft, 14(2), 87–101. DOI: 10.1007/s11618-011-0183-6.

Bilger, F., Behringer, F., Kuper, H., & Schrader, J. (Ed.) (2017). Weiterbildungsverhalten in Deutschland 2016: Ergebnisse des Adult Education Survey (AES) (DIE Survey: Daten und Berichte zur Weiterbildung). Bielefeld: W. Bertelsmann Verlag. DOI: 10.3278/85/0016w.

Blossfeld, H.-P. & Roßbach, H.-G. (Eds.). (2019). Education as a lifelong process: The German National Educational Panel Study (NEPS). Edition ZfE (2nd ed.). Springer VS.

Ebner, C., Ehlert, M. (2018). Weiterbilden und Weiterkommen? Non-formale berufliche Weiterbildung und Arbeitsmarktmobilität in Deutschland. In: Kölner Zeitschrift für Soziologie und Sozialpsychologie, Jg. 70, H. 2, S. 213-235. https://doi.org/10.1007/s11577-018-0518-x

Ehlert, M. (2017). Who Benefits from Training Courses in Germany? Monetary Returns to Non-formal Further Education on a Segmented Labour Market. In: European Sociological Review, Vol. 33, No. 3, S. 436-448.

FDZ-LIfBi (2022). Data Manual NEPS Starting Cohort 6– Adults, Adult Education and Lifelong Learning, Scientific Use File Version 13.0.0. Bamberg, Germany, Leibniz Institute for Educational Trajectories, National Educational Panel Study.

Janik, F., Wölfel, O., Trepesch, M. (2016). Measurement of further training activities in life-course studies. In: Blossfeld, H. (Ed.); Maurice, J. (Ed.); Bayer, M. (Ed.), Skopek, M. (Ed.): Methodological issues of longitudinal surveys. The example of the National Educational Panel Study. Wiesbaden: Springer VS (2016) S. 385-397.

Kantar (2020) Erhebung zum Weiterbildungsverhalten in Deutschland 2018 (AES 2018). Handbuch zur Datennutzung. München: Kantar-Public Division.

NEPS Network (2021). *National Educational Panel Study, Scientific Use File of Starting Cohort Adults.* Leibniz Institute for Educational Trajectories (LIfBi), Bamberg. https://doi.org/10.5157/NEPS:SC6:12.0.0

NEPS Network (2022). *National Educational Panel Study, Scientific Use File of Starting Cohort Adults.* Leibniz Institute for Educational Trajectories (LIfBi), Bamberg. https://doi.org/10.5157/NEPS:SC6:13.0.0

Rzepka, S. (2016). Analyzing further training participation rates across waves in the NEPS data. Ruhr Economic Papers 655, RWI - Leibniz-Institut für Wirtschaftsforschung, Ruhr-University Bochum, TU Dortmund University, University of Duisburg-Essen.

Appendix A

```
*******************************************************************************
* Stata do-file to link and prepare data on further training in the NEPS starting cohort 6
* Martin Ehlert, Insa Grüttgen, Alexander Helbig
*******************************************************************************

clear
version 16
set more off
set matsize 800

global DATE = string( d(`c(current_date)'), "%dYND" )
global DATA "DATAPATH" // datapath for easier access - to be adjusted
global suf "13-0-0"
global suf_kurz 13

*******************************************************************************
* FurtherEducation-Data set
*******************************************************************************

// using the FurtherEducation-Data set as basis
use "$DATA\SC6_FurtherEducation_D_$suf.dta", clear


*******************************************************************************
**# start and end dates
*******************************************************************************

// Start and end-variables of the courses in year-month-format, deleting original variables
```

```
// Note that for all courses from spCourses and from FurtherEdu1 after wave 2 no exact dates are recorded.

// Start and end were derived from interview and spell dates, see variable tx28201


foreach var of varlist tx2821y tx2821m tx2822y tx2822m {
        replace `var' = missing() if `var' == -55
}

gen start=ym(tx2821y, tx2821m)
        format start %tm
gen ende=ym(tx2822y, tx2822m)
        format ende %tm
order start ende, after (tx28200)
keep if start < . & ende < .


// no course- and further training-module in wave 1 (ALWA study), only courses from vocTrain module in this data set
// not comparable to the rest, therefore drop wave 1
drop if wave == 1


// Interview date from CohortProfile data set
        preserve
use ID_t wave tx8600y tx8600m using "$DATA\SC6_CohortProfile_D_$suf.dta", clear
drop if wave == 1


gen interv=ym(tx8600y, tx8600m) // interview date in each wave
format interv %tm


tempfile intd
save `intd'
        restore
```

```
merge m:1 ID_t wave using `intd', ///
        keepusing(interv) keep(master match) nogen

*******************************************************************************
**# Irregular end dates
*******************************************************************************

tab tx2822y wave, mis
gen timegap = interv-ende if interv!=. & ende!=.
fre timegap // some course dates have end dates after the interview (negative values in timegap) due to transfer errors. We disregard these transfer errors here, these should mostly be corrected in upcoming Scientific Use Files.

**Problem: some courses have end dates a long time before the interview
// Reasons for large time gaps (course end dates long before the interview date):

// 1. Four courses in FurtherEdu1 were dated by the respondents in wave 2 as ending a long time ago, which is contrary to the reference period of 12 months (from wave 3 on this was no longer possible)
fre timegap if tx28200 == 31 & wave == 2
// -> Depending on the research interest, they could be removed because they are not in the reference period of the wave (12 months prior or since last interview)
*drop if timegap > 12 & timegap != . & tx28200 == 31 & wave == 2

// 2. A few courses in spCourses are linked to spells that end a long time before the interview
fre timegap if tx28200 == 35
// This may be because of respondents who missed a wave -> reported episodes since last interview
// Another reason may be respondents who took part at the beginning of field time T and at the end of field time T+1 for the next interview
// courses with a very long time gap in wave 3 are due to data errors
// -> Depending on the research interest, they could be removed because they are not in the reference period of the wave (12 months prior or since last interview)
```

*drop if timegap > 18 & timegap != . & tx28200 == 35 // Assuming a maximum of 18 months between interviews of two consecutive waves

// 3. Many courses from vocTrain in waves 2 and 4 happened a long time before the interview

tabstat timegap if tx28200 ==24, by(wave)

// in wave 2 and wave 4 life course episodes (such as vocational training) were reported retrospectively for the entire life course for respondents who took part for the first time

//-> for better comparison one could drop all spVocTrain courses that ended more than 18 months before interview in wave 2 or 4

*drop if timegap > 18 & timegap != . & inlist(wave,2,4) & tx28200==24

// Depending on research interest, reference period can also be harmonized for all waves as for the rest of the courses

*drop if timegap > 18 & timegap != . & tx28200==24

// Courses that ended more than 12 months prior to the interview from spVocTrain usually have a missing on the course variable. Hence, users can drop all such courses with a missing course-variable if needed, but this would also exclude all non-licence courses from spVocTrain that have been recoded as non-formal courses afterwards regardless of their end date. Note that these can not be merged with spFurtherEdu2 anyway.

* drop if course==.

// drop unnecessary variables

drop tx2822c tx28202_R tx28204 tx2821m tx2821y tx2822m tx2822y timegap

****

**# adding additional information from spFurtherEdu2

****

* professional/private reasons, attendance during working hours, course costs Employer

merge m:1 ID_t course using "$DATA/SC6_spFurtherEdu2_D_$suf.dta" // merge 1:1 is not possible because of missing values in course-variable for most courses from vocTrain

```
drop if _merge==2 // 14 infos from using can't be linked (these are courses with missing start or endddates, which we dropped at the beginning)

drop _merge


*******************************************************************************

**# linking additional information (professional/private reasons)

**  from spFurtherEdu1 for courses from wave 11 on (see table 1 in survey paper)

*******************************************************************************


foreach var in 9040 9041 9042 9043 2043 {

        replace t27`var'=.m if t27`var'==-54

}


// information on these variables not stored in spFurtherEdu2 from wave 11 on, therefore missing; here changed to .m for later merging


        preserve

use "$DATA/SC6_spFurtherEdu1_D_$suf.dta", clear


drop if wave<11 // information only to be added for courses from wave 11 on


replace t272043 = t272043_v1 if wave==11

recode t272043 (2 = 4)

label def de5626ext1 3 "beides, Teilnahmebescheinigung und anerkannte Lizenz", modify

label val t272043 de5626ext1


tempfile fe1

save `fe1'

        restore


merge m:1 ID_t wave course using `fe1', ///
```

```
        keepusing(t279040 t279041 t279042 t279043 t272043) gen(_mergewb) update // update variables in FurtherEducation if wave>=11


assert _mergewb!=2 // assert m:1-merge worked well

drop _mergewb



*******************************************************************************

**# linking additional information (professional/private reasons)

**  from spCourses for courses from wave 11 on (see table 1 in survey paper)

*******************************************************************************


// courses in spCourses are stored in wide format (for each person-wave combination), data set first transformed into  long format (person-wave-course as observations)


        preserve

use "$DATA/SC6_spCourses_D_$suf.dta", clear


bys ID_t wave: gen n =_n

keep ID_t wave t279030_w* t279031_w* t279032_w* t279033_w* t272043_w* course_w* n

reshape long t279030_w t279031_w t279032_w t279033_w t272043_w course_w, i(ID_t wave n) j(course_nr)

drop if course_w==. // course_w missing when no (further) course reported


rename course_w course // rename for merging

rename t279030_w t279040

rename t279031_w t279041

rename t279032_w t279042

rename t279033_w t279043

rename t272043_w t272043


drop if wave<11
```

```
tempfile co
save `co'
        restore

merge m:1 ID_t wave course using `co', ///
        keepusing(t279040 t279041 t279042 t279043 t272043) gen(_mergeco) keep(master match match_update) update

assert _mergeco!=2
drop _mergeco

label language en

*******************************************************************************
**# Categorization of Further Education activities in private or job-related (firm-sponsored/non-firm-sponsored)
*******************************************************************************

** private or job-related **
// where data is available
// -> courses from spFurtherEdu2 (randomly chosen courses for extra questions) and courses from spCourses & spFurtherEdu1 after wave 10

gen jobrelated = .
replace jobrelated = 0 if t279040==2
replace jobrelated = 1 if t279040==1 | t279040==3 // "both" = job-related

label var jobrelated "classfification of further education courses"
label def format_fe 0 "private course" 1 "job-related course"
label val jobrelated format_fe
```

```
** firm-sponsored / non-firm-sponsored **
// more detailed classification for job-related courses **
// -> data available for courses from spFurtherEdu2

gen firmsponsored = .
replace firmsponsored = 1 if jobrelated==1 & (inlist(t279044,1,2) | inlist(t279046,1,2))
// partly during working hours / costs partly covered by the employer = firm-sponsored
replace firmsponsored = 0 if jobrelated==1 & t279044==3 & t279046==3

label var firmsponsored "classification of job-related courses"
label def format_fs 1 "firm-sponsored" 0 "non-firm-sponsored"
label val firmsponsored format_fs


*******************************************************************************
**# missing information on private or job-related course motivation before wave 11
** logical imputation based on the corresponding life episode
*******************************************************************************

// before wave 11 data on whether course was for private or job-related reasons only collected for (up to) 2 courses per person (spFurtherEdu2) (see survey paper for details)
// -> option to distinguish between private and professional Further Training for missing values based on corresponding life episode of reported course

// get information on sptype (life episode context of reported courses)
        preserve
use "$DATA/SC6_spCourses_D_$suf.dta", clear

bys ID_t wave: gen n =_n
keep ID_t wave course_w* n sptype
```

```
reshape long course_w, i(ID_t wave n) j(course_nr)

drop if course_w==. // course_w missing when no (further) course reported


rename course_w course // rename for merging


tempfile co2

save `co2'

        restore


merge m:1 ID_t wave course using `co2',          keepusing(sptype) keep(master match)
gen(_mergeco2)


assert _mergeco2!=2


// generate variable with addition of job-related courses depending on context

generate jobrelated_g = jobrelated

replace jobrelated_g=1 if tx28200==24 & jobrelated==. // courses stated as vocational training

replace jobrelated_g=1 if tx28200==35 & inlist(sptype,24,26,27) & jobrelated==. // courses registered in the course-module and mentioned in context of employment, unemployment or vocational training (from wave 12 on licence courses mentioned in the module for vocational training (24) go into the course-module-loop, so they are included in spCourses (35))

label val jobrelated_g format_fe



*******************************************************************************

**# weighting

*******************************************************************************

// adapted from course weights from German AES

// adjustment for the fact that some information available for all courses, some only for a few, different for different waves

// Note: This is only available for courses that could be selected into the loop for further questions. Thus it excludes most courses from vocTrain (e.g. non-licence courses) as they have a missing on the course-variable
```

** Generate a weight for variables that were only available in spFurtherEdu2 over all waves

// courses before wave 11: some information only available in FurtherEdu2 for TWO randomly chosen courses (see table 1 in survey paper)

// First count all courses that were considered in the random assignment loop -> all courses with a course number

gen counter = 1 if course!=.

bysort ID_t wave: egen number_loop = total(counter)

gen cweight_FE2 = . if course==. // non-license courses not included in loop

replace cweight_FE2 = 1 if wave<11 & inlist(number_loop,1,2) & course!=.

replace cweight_FE2 = number_loop/2 if wave<11 & number_loop>2 & course!=.

// courses from wave 11 on: some information only available in FurtherEdu2 for ONE randomly chosen course (see table 1 in survey paper)

replace cweight_FE2 = number_loop if wave>=11 & course!=.

label var cweight_FE2 "Weight for information only available in spFurtherEdu2"

** Generate a weight for variables that were included in spFurtherEdu2 until wave 11 and then available for all courses in spCourses and spFurtherEdu1 (e.g. t279040) (see table 1 in survey paper)

gen cweight_change = . if course==.

replace cweight_change = 1 if wave>=11 & course!=.

replace cweight_change = cweight_FE2 if wave<11 & course!=.

label var cweight_change "Weight for information that was first available for selected courses then for all"

** Example Analyses:

// How did the overall assessment of courses change over time?

// (This is an item that is always asked in the loop, available for 1-2 courses)

```
mvdecode t273023, mv(-54 -97 -98)

tab t273023 wave, col nofreq // without weights

tab t273023 wave [aw=cweight_FE2], col nofreq // with weights


// How did the share of compulsory courses change over time?

// (This is an item that is asked in the loop until wave 11, then for all courses)


mvdecode t279042, mv(-54 -97 -98)

tab t279042 wave, col nofreq // without weights

tab t279042 wave [aw=cweight_change], col nofreq // with weights


// Note that these weights may be multiplied with the survey weights for descriptive analyses on the course level


*******************************************************************************

**# Transformation in person-wave-format to merge with CohortProfile data set

*******************************************************************************


bysort ID_t wave: gen nr = _n // course number


drop _merge* number_loop counter nepswave sptype


// some aggregated statistics on the person-wave level

bysort ID_t wave: gen n_courses = _N // courses per person and wave

nepsmiss tx28203

bys ID_t wave: egen hours_courses = total(tx28203) // total duration of courses per person and wave; no information on duration for courses from spVocTrain

bys ID_t wave: egen n_jobtotal = total(jobrelated) // job-related courses per person and wave

bys ID_t wave: egen n_jobtotal_w = total(jobrelated*cweight_change) // job-related courses per person and wave extrapolated using the weight adjusting for random drawing

bys ID_t wave: egen n_jobtotal_g = total(jobrelated_g) // job-related courses (including the imputed ones) per person and wave
```

```
// create a data file with one observation per person and wave, keeping wave-aggregated variables
bysort ID_t wave: gen n = _n
keep if n==1
keep ID_t number wave interv n_courses hours_courses n_jobtotal n_jobtotal_w n_jobtotal_g

// merge with CohortProfile data set which includes all respondents over all waves
merge 1:1 ID_t wave using "$DATA\SC6_CohortProfile_D_$suf.dta", nogen

// keep only those who participated in the respective wave
keep if tx80220 == 1

// drop ALWA again
drop if wave == 1

// Person-years without courses now have system missings on the course variables
// To generate variables that also indicate "no courses", replace missing with 0 for summary variables
foreach var of varlist n_courses n_jobtotal n_jobtotal_g {
        replace `var' = 0 if `var' == .
}

** generate dummies on some basic information

recode n_courses (1/max = 1), gen(nf_dummy) // DUMMY: courses in this wave yes/no
recode n_jobtotal (1/max = 1), gen(nfjr_dummy) // DUMMY: job-related courses in this wave yes/no
recode n_jobtotal_g (1/max = 1), gen(nfjr_g_dummy) // DUMMY: job-related courses (including the imputed ones) in this wave yes/no

** basic distributions
// please note that these distributions have to be interpreted with caution because they also reflect technical differences between waves, see survey paper for details
```

```
* percentage of course participation

graph bar (mean) nf_dummy, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3 "2011/2012" 4
"2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9 "2017/2018" 10
"2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(share of participants) title("All
courses") name(all, replace)


* percentage of job-related course participation (without logical imputation)

graph bar (mean) nfjr_dummy, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3 "2011/2012" 4
"2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9 "2017/2018" 10
"2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(share of participants )
title("Job-related courses") name(perc_jr1, replace)


* percentage of job-related course participation (with logical imputation)

graph bar (mean) nfjr_g_dummy, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3 "2011/2012" 4
"2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9 "2017/2018" 10
"2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(share of participants)
title("Job-related courses (log. imputation)") name(perc_jr2, replace)


graph combine all perc_jr1 perc_jr2, ycommon


* average number of courses per person over time

graph bar (mean) n_courses if n_courses >0, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3
"2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9
"2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(average no. of
courses) name(raw, replace) title("All")


* average number of job-related courses per person over time (without logical imputation)

graph bar (mean) n_jobtotal if n_courses >0, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3
"2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9
"2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(average no. of
courses ) name(jr1, replace) title("Job-related")


* average number of job-related courses per person over time (with weights adjusting for random
selection of courses)

graph bar (mean) n_jobtotal_w if n_courses >0, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3
"2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9
```

```
"2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(average no. of courses  ) name(jr2, replace) title("Job-related with weights")


* average number of job-related courses per person over time (with logical imputation)

graph bar (mean) n_jobtotal_g if n_courses >0, over(wave, relabel(1 "2009/2010" 2 "2010/2011" 3 "2011/2012" 4 "2012/2013" 5 "2013/2014" 6 "2014/2015" 7 "2015/2016" 8 "2016/2017" 9 "2017/2018" 10 "2018/2019" 11 "2019/2020" 12 "2020/2021") label(angle(90))) ytitle(average no. of courses ) title("Job-related with logical imputation") name(jr3, replace)


graph combine raw jr1 jr2 jr3, ycommon
```

Appendix B

```
# R Script to link and prepare data on further training in the NEPS starting cohort 6
# mostly equivalent to the corresponding STATA-dofile from Ehlert/Gruettgen/Helbig
# Alexander Helbig
# 25.10.22

################################################################################
# if necessary install required packages before!!
################################################################################
# install.packages("dplyr")
# install.packages("ggplot2")
# install.packages("tidyr")
# install.packages("haven")
# install.packages("naniar")
# install.packages("gmodels")
# install.packages("janitor")
# install.packages("rqdatatable")
# install.packages("gridExtra")
# install.packages("grid")
library(tidyr)
library(dplyr)
library(haven)
library(naniar)
library(gmodels)
library(janitor)
library(rqdatatable)
library(ggplot2)
library(gridExtra)
library(grid)
```

```
#clear workspace
rm(list = ls())

################################################################################
# specify datapath and suf version!!
################################################################################
datapath = "DATAPATH"
suf_version = "13-0-0"

############################ spFurtherEducation data set ##########################

# read data FurtherEducation
ft_data <-
  read_dta(paste0(datapath, "SC6_FurtherEducation_D_", suf_version,".dta")
  )

# Start and end-variables in year-month-format, deleting original variables
missings_list <-
  c(seq(-99, -90), seq(-56, -51), seq(-29, -20)) # possible missing values
ft_data <- ft_data %>%
  replace_with_na(
    replace = list(
      tx2821y = missings_list,
      tx2821m = missings_list,
      tx2822y = missings_list,
      tx2822m = missings_list
    )
  ) # replace all missing values with NA
ft_data <- ft_data %>%
  mutate(start = ((ft_data$tx2821y - 1960) * 12) + ft_data$tx2821m - 1,
```

```
      ende = ((ft_data$tx2822y - 1960) * 12) + ft_data$tx2822m - 1) %>% #startdate in months since
jan. 1960
  filter(!is.na(start) & !is.na(ende))


# no course- and further training-module in wave 1 ("ALWA")
ft_data <- ft_data %>%
  filter(wave > 1)


################### join Interview date from cohort profile ####################


# read data cohortprofile in order to get access to interview dates
interview_data <-
  read_dta(paste0(datapath, "SC6_CohortProfile_D_", suf_version,".dta")
  )%>%
  select(ID_t, wave, tx8600y, tx8600m)


# take "FurtherEducation" dataset as a base for the merging with cohort profile
# we use natural_join because its functionality is very similar to statas merge command
ft_data <-
  natural_join(ft_data,
               interview_data,
               by = c("ID_t", "wave"),
               jointype = "LEFT") #"natural join" from rqdatatable package copys statas merge behaviour


# generate interview date
ft_data <- ft_data %>%
  mutate(interv = ((ft_data$tx8600y - 1960) * 12) + ft_data$tx8600m - 1) %>%
  select(-tx8600y,-tx8600m)



############################## problem: course dates ################################
```

```
# some courses ended a long time before the interview

table(ft_data$tx2822y, ft_data$wave)

# 1. respondents who missed one wave -> reported episodes since last interview

# 2. respondents who took part at the beginning of field time and at the end of field time for the next interview


# 3. in wave 2 and 4 all episodes of vocational training were registered retrospectively -> reported vocational training episodes that were classified as courses (from spVocTrain) possibly ended years before the interview

ft_data %>%

  filter(ft_data$tx28200 != 24) %>%

  tabyl(tx2822y, wave) # tabyl from janitor package is working well with dplyr and piping


ft_data <-

  ft_data %>% # some courses with transfer errors (enddate later than respective interview date)

  mutate(timegap = interv - ende)


ft_data %>%

  filter(wave != 2 & wave != 4) %>%

  tabyl(timegap) # apart from wave 2 and 4, 99.9% of all courses ended up to 24 months prior to the interview


# -> for better comparison option to drop all spVocTrain courses that ended more than 24 months before interview in wave 2 or 4, here only marked with variable "early"

ft_data <- ft_data %>%

  mutate(early = ifelse((ende < interv - 24) &

                        (wave == 2 | wave == 4) & tx28200 == 24, 1, 0))


# show means and destribution

mean(ft_data$early, na.rm = TRUE)

tabyl(ft_data$early)
```

```
mean(ft_data$start, na.rm = TRUE)
mean(ft_data$ende, na.rm = TRUE)

# keep it simple
ft_data <- ft_data %>%
  select(-c(
    tx2822c,
    tx28202_R,
    tx28204,
    tx2821m,
    tx2821y,
    tx2822m,
    tx2822y,
    timegap
  ))

# adding additional information from spFurtherEdu2
# professional/private reasons, attendance during working hours, Course costs Employer



############################## join furtherEdu2 ################################

# load furtheredu2 (looped detailed training data)
further_edu2 <-
  read_dta(paste0(datapath, "SC6_spFurtherEdu2_D_", suf_version,".dta")
  )

# merge it with ft_data
ft_data <-
  natural_join(ft_data,
               further_edu2,
```

```
            by = c("ID_t", "course"),
            jointype = "LEFT") %>%
  arrange(ID_t, wave, number) %>%
  mutate(across(everything(), ~na_if(., -54))) # set -54 to NA in all columns

ft_data %>% tabyl(t279042,wave, show_na = TRUE)

# linking additional information (eg professional/private reasons) from spFurtherEdu1 for courses from wave 11 on(see table 2 in text)
############################ join furtherEdu1 ##############################

#load furtheredu1 data
further_edu1 <-
  read_dta(paste0(datapath, "SC6_spFurtherEdu1_D_", suf_version,".dta")
  )

further_edu1 <- further_edu1 %>%
  mutate(t272043 = ifelse(wave == 11, t272043_v1, t272043),
       t272043 = ifelse(t272043 == 2, 4, t272043)) %>% # overwrite t272043 (certificate) with t272043_v1 var in wave 11 and recode code 2 to code 4 for harmonization of the two variables ("anerkannte lizenz")
  filter(wave > 10) %>%
  select(ID_t, wave, course, t279040, t279041, t279042, t279043, t272043) #keep only waves above 10

ft_data <-
  natural_join(ft_data,
            further_edu1,
            by = c("ID_t", "wave", "course"),
            jointype = "LEFT")
tabyl(ft_data$t279042)
############################ join spcourses ##############################
```

```
# load course data
spcourses <-
  read_dta(paste0(datapath, "SC6_spCourses_D_", suf_version,".dta")
  )


# generate counter per ID and wave
spcourses <- spcourses %>%
  group_by(ID_t, wave) %>%
  mutate(n = row_number())

# generate overall N per ID and wave
spcourses <- spcourses %>%
  group_by(ID_t, wave) %>%
  mutate(N = max(row_number()))


# select columns and reshape to long format
spcourses <- spcourses %>%
  select(
    starts_with("t279030_w"),
    starts_with("t279031_w"),
    starts_with("t279032_w"),
    starts_with("t279033_w"),
    starts_with("t272043_w"),
    starts_with("course_w"),
    "wave",
    "ID_t",
    "n"
  ) %>%
```

```
  gather("key", "value", c(starts_with("t27"), starts_with("course_w"))) %>%
  extract(col = "key",
          into = c("colname", "number"),
          regex = "([a-z\\d]+)(_\\w\\d+)") %>%
  spread("colname", "value")

# course is missing when no (further) course reported - remove!
spcourses <- spcourses %>%
  filter(!is.na(course) & wave > 10)

# renaming to the names of the variables in furtherEdu1
spcourses <- rename(
  spcourses,
  t279040 = t279030,
  t279041 = t279031,
  t279042 = t279032,
  t279043 = t279033,
  t272043 = t272043
)

ft_data <-
  natural_join(ft_data,
               spcourses,
               by = c("ID_t", "wave", "course"),
               jointype = "LEFT")

################################################################################
# Categorization of Further Education activities in private/job-related (on the job/individual-professional)
################################################################################
```

```
# generate private/job-related dummy variable

ft_data <- ft_data %>%

  mutate(jobrelated = NA,

         jobrelated = ifelse(t279040 == 2, 0, jobrelated),

         jobrelated = ifelse(t279040 %in% c(1,3), 1, jobrelated))


# generate firmsponsored/non-firmsponsored dummy variable

ft_data <- ft_data %>%

  mutate(firmsponsored = NA,

         firmsponsored = ifelse(jobrelated == 1 & (t279044 %in% c(1,2) | t279046 %in% c(1,2)), 1,
firmsponsored),

         firmsponsored = ifelse(jobrelated == 1 & t279044 == 3 & t279046 == 3, 0, firmsponsored))


################################################################################

# missing information on private or job-related course motivation before wave 11

################################################################################


# before wave 11 data on whether course was for private or job-related reasons only collected for
(up to) 2 courses per person (spFurtherEdu2)


# -> allocation of distinction between private and professional Further Training for missing values -
based on source of mentioned course. See paper for details


rm(spcourses)


# load course data again

spcourses <-

  read_dta(paste0(datapath, "SC6_spCourses_D_", suf_version,".dta")

  )


# generate counter per ID and wave

spcourses <- spcourses %>%
```

```
  group_by(ID_t, wave) %>%
  mutate(n = ave(ID_t, wave, FUN = seq_along)) %>%
  select(c(ID_t,wave,starts_with("course_w"), n,sptype))

# select columns and reshape to long format
spcourses <- spcourses %>%
  gather("key", "value", starts_with("course_w")) %>%
  extract(col = "key",
          into = c("colname", "number"),
          regex = "([a-z\\d]+)(_\\w\\d+)") %>%
  spread("colname", "value") %>%
  filter(!is.na(course)) %>%
  select(-n, -number)

#join ft_data with this to get access to sptype info of courses in spcourses
ft_data <-
  natural_join(ft_data,
               spcourses,
               by = c("ID_t", "wave", "course"),
               jointype = "LEFT")

# generate variable with addition of job-related courses depending on context
# generate firmsponsored/non-firmsponsored dummy variable
ft_data <- ft_data %>%
  mutate(
    jobrelated_g = jobrelated,
    jobrelated_g = ifelse((tx28200 == 24 & is.na(jobrelated)) |
                            tx28200 == 35 & sptype %in% c(24, 26, 27) & is.na(jobrelated),
                          1,
                          jobrelated_g
    )
```

```
 )

rm (further_edu1, further_edu2, interview_data, spcourses)

################################################################################

# weighting

################################################################################

# adapted from course weights from German AES

# adjustment for the fact that some information available for all courses, some only for a few, different for different waves

# Note: This is only available for courses that could be selected into the loop for further questions. Thus it excludes most courses from vocTrain

# First generate a weight for information that is only available in spFurtherEdu2 over all waves (see Table 1)

# courses before wave 11: some information only available in FurtherEdu2 for TWO randomly chosen courses (see table 1 in survey paper)

# First count all courses that were considered in the random assignment loop -> all courses with a course number

ft_data <- ft_data %>%

   group_by(ID_t, wave) %>%

    mutate(a = cumsum(!is.na(course))) %>%

    mutate(number_loop = max(a))

ft_data <- ft_data %>%

 mutate(cweight_FE2 = NA,

       cweight_FE2 = ifelse(wave <11 & number_loop %in% c(1,2) & !is.na(course),1,cweight_FE2),

       cweight_FE2 = ifelse(wave <11 & number_loop > 2 & !is.na(course),number_loop/2,cweight_FE2),

       cweight_FE2 = ifelse(wave >=11 & !is.na(course),number_loop,cweight_FE2)) # courses from wave 11: some information only available in FurtherEdu2 for ONE randomly chosen course (see table 1 in text)
```

```
# Generate a weight for variables that were included in spFurtherEdu2 until wave 11 and then available for all courses in spCourses and spFurtherEdu1 (e.g. t279040) (see table 1 in survey paper)

ft_data <- ft_data %>%
  mutate(cweight_change = NA,
         cweight_change = ifelse(wave>=11 & !is.na(course),1,cweight_change),
         cweight_change = ifelse(wave < 11 & !is.na(course),cweight_FE2,cweight_change))

################################################################################
# Example Analyses to showcase unweighted vs. weighted data:
################################################################################

################################################################################
# How did the overall assessment of courses change over time?
# (This is an item that is always in the loop)

# (compare unweighted two way table of wave and t273023 to weighted freqency table - since there is no aw-weighting for two-way tables in R, we replicate this step from the stata do file according to the following statalist thread: https://www.statalist.org/forums/forum/general-stata-discussion/general/1459139-what-is-the-formula-for-aweight)
################################################################################

# Pre step: delete missings from target-variable
ft_data2 <- ft_data %>%
  replace_with_na(replace = list(
    t273023 = missings_list)) %>%
  filter(!is.na(t273023))

# generate mean of weight for all obs
ft_data2 <- cbind(ft_data2, mean_cweight_FE2 = rep(mean(ft_data2$cweight_FE2, na.rm =TRUE), 46432))
```

```
# generate weight / mean-weight variable
ft_data2 <- ft_data2 %>%
  mutate(w_de_mean = cweight_FE2/mean_cweight_FE2)

# generate total sum grouped by wave and target-variable
ft_data2 <- ft_data2 %>%
  arrange(t273023) %>%
  group_by(wave, t273023) %>%
  mutate(sum_w_t273023_mean = max(cumsum(w_de_mean))) %>%
  ungroup()

# select vars
ft_data2 <- ft_data2 %>%
  select(t273023,wave, sum_w_t273023_mean)

# reduce df to distinct values and create percentage variable
ft_data3 <- ft_data2 %>%
  distinct() %>%
  arrange(wave,t273023) %>%
  group_by(wave) %>%
  mutate(percentage = 100* sum_w_t273023_mean/sum(sum_w_t273023_mean, na.rm= TRUE))
%>%
  ungroup()

##############################################################################

# results: course assessment: summarize - unweighted frequncies and percentages across waves vs
weighted percentages across waves

# unweighted two-way table
ft_data2 %>% tabyl(wave,t273023, show_na = FALSE) # frequencies
ft_data2 %>% tabyl(wave,t273023, show_na = FALSE) %>% adorn_percentages("row", na.rm =TRUE)
# percentages
```

```
# weighted frequencies and values

print(ft_data3, n =55)


rm(ft_data2, ft_data3)


################################################################################

# How did the share of compulsory courses change over time?

# (This is an item that is first in the loop, then for all courses)


# (compare unweighted two way table of wave and t273023 to weighted freqency table - since there is no aw-weighting for two-way tables in R, we replicate this step from the stata do file according to the following statalist thread: https://www.statalist.org/forums/forum/general-stata-discussion/general/1459139-what-is-the-formula-for-aweight)

################################################################################


# Pre step: delete missings from target-variable

ft_data2 <- ft_data %>%

  replace_with_na(replace = list(

    t279042 = missings_list)) %>%

  filter(!is.na(t279042))


# generate mean of weight for all obs

ft_data2 <- cbind(ft_data2, mean_cweight_change = rep(mean(ft_data2$cweight_change, na.rm =TRUE), 61631))


# generate weight / mean-weight variable

ft_data2 <- ft_data2 %>%

  mutate(w_de_mean2 = cweight_change/mean_cweight_change)


# generate total sum grouped by wave and target-variable

ft_data2 <- ft_data2 %>%
```

```
  arrange(t279042) %>%
  group_by(wave, t279042) %>%
  mutate(sum_w_t279042_mean = max(cumsum(w_de_mean2))) %>%
  ungroup()

# select vars
ft_data3 <- ft_data2 %>%
  select(t279042,wave, sum_w_t279042_mean)

# reduce df to distinct values and create percentage variable
ft_data3 <- ft_data3 %>%
  distinct() %>%
  arrange(wave,t279042) %>%
  group_by(wave) %>%
  mutate(percentage = 100* sum_w_t279042_mean/sum(sum_w_t279042_mean, na.rm= TRUE))
%>%
  ungroup()

################################################################################
# results: compulsory courses - unweighted percentages across waves vs weighted percentages
across waves

# unweighted two-way table of percentages of compulsory courses dummy
ft_data2 <- replace_with_na(ft_data2, replace = list(t279042 = c(-54,-97,-98)))
ft_data2 %>% tabyl(wave,t279042, show_na = FALSE) %>% adorn_percentages("row", na.rm =TRUE)
# percentages

# weighted frequencies and values
print(ft_data3, n =24)

# Note that these weighted values may be multiplied with the survey weights for descriptive analyses
on the course level
```

```
rm(ft_data2, ft_data3)

################################################################################
# transformation in person-wave-format to merge with CohortProfile data set
################################################################################

# generate counter per ID and wave
ft_data <- ft_data %>%
  group_by(ID_t, wave) %>%
  mutate(n = row_number()) %>%
  select(-sptype)

##### generate some aggregated statistics on the person-wave level #####

# generate overall N per ID and wave
ft_data <- ft_data %>%
  group_by(ID_t, wave) %>%
  mutate(n_courses = max(row_number()))

# set tx28203 = -55 to NA
ft_data <- replace_with_na(data = ft_data, replace = list(tx28203 = -55))

#first set missings to zero in a new var "_e" and then create a new variable which shows the max value of cumsum of the _e variable version
ft_data <- ft_data %>%
  group_by(ID_t, wave) %>%
  mutate(tx28203_e = ifelse(is.na(tx28203),0, tx28203),
         jobrelated_e = ifelse(is.na(jobrelated),0, jobrelated),
         jobrelated_g_e = ifelse(is.na(jobrelated_g),0, jobrelated_g),
         cweight_change_e = ifelse(is.na(cweight_change),0, cweight_change),
```

```
    hours_courses = max(cumsum(tx28203_e)),
    n_jobtotal = max(cumsum(jobrelated_e)),
    n_jobtotal_w = max(cumsum(jobrelated_e*cweight_change_e)),
    n_jobtotal_g = max(cumsum(jobrelated_g_e)))

# create a data file with one observation per person and wave, keeping wave-aggregated variables
ft_data <- ft_data %>%
  filter(n == 1) %>%
  select(ID_t, number, wave, interv, n_courses, hours_courses, n_jobtotal, n_jobtotal_w,
n_jobtotal_g)

# load cohort_profile data
cohort_profile <-
  read_dta(paste0(datapath, "SC6_CohortProfile_D_", suf_version,".dta")
  )

# join ft_data with cohort_profile, keep only those who participated in the wave and drop alwa cases
again
ft_data<- natural_join(cohort_profile,ft_data, by = c("ID_t", "wave"), jointype = "LEFT")

# keep only participating observations and get rid of ALWA wave again
ft_data <- ft_data %>%
  filter(ft_data$tx80220 == 1 & wave > 1)

# Person-years without courses now have system missings on the course variables
# To generate variables that also indicate "no courses", replace missing with 0 for summary variables
ft_data <- ft_data %>%
  mutate(across(c(n_courses, n_jobtotal, n_jobtotal_g), ~ifelse(is.na(.),0,.)))

# generate 3 dummy variables of the aforementioned and recoded variables
ft_data <- ft_data %>%
  mutate(nf_dummy = if_else(n_courses > 0,1,0),
```

```
      nfjr_dummy = if_else(n_jobtotal > 0,1,0),
      nfjr_g_dummy = if_else(n_jobtotal_g > 0,1,0))

tabyl(ft_data$nf_dummy)
tabyl(ft_data$n_jobtotal)
tabyl(ft_data$n_jobtotal_g)
##############################################################################
# basic distributions
# please note that these distributions have to be interpreted with caution because
# they also reflect technical differences between waves, see survey paper for details
##############################################################################

#change wave to numeric from class haven for plotting
ft_data$wave <- as.vector(ft_data$wave)

##############################################################################
# prepare plot, create theme
##############################################################################

theme_1 <-
 theme(
   plot.title = element_text(size = 15, color = "black", face = "bold"),
   axis.title.y = element_text(face="bold"),
   axis.title.x = element_text(face="bold"),
   axis.text.x = element_text(
     face = "bold",
     color = "black",
     size =
       10,
     angle = 1
   ),
```

```
   axis.text.y = element_text(
    face = "bold",
    color = "black",
    size =
     10,
    angle = 1
   )
 )
# prepare plot, create x-axis specs
scale_x <- scale_x_continuous(
   name = "Year",
   breaks = c(2:13),
   labels = c("2" ="09/10", "3"="10/11", "11/12", "12/13", "13/14",
              "14/15", "15/16", "16/17", "17/18", "18/19", "19/20", "20/21"),
 )

# prepare plot, create y-axis specs
scale_y <-
 scale_y_continuous(name = "Training Rate",
 limits = c(0,0.5)
)
# prepare plot, create y-axis specs for "number of courses" plots
scale_y2 <-
 scale_y_continuous(name = "Average number of courses")

################################################################################
# create plots with training rates - differentiated
################################################################################

plot1 <- ggplot(ft_data %>% group_by(wave) %>% summarise(perc = mean(nf_dummy)), aes(wave,
perc)) +
```

```
  geom_col(color = "black", fill = "purple") +
  ggtitle("All Training") +
  theme_1 +
  scale_x +
  scale_y +
  geom_text(aes(label = round(perc, 2)), vjust = 1.4, color = "white")



# plot job related training over waves
plot2 <- ggplot(ft_data %>% group_by(wave) %>% summarise(perc = mean(nfjr_dummy)), aes(wave,
perc)) +
  geom_col(color = "black", fill = "purple") +
  ggtitle("Job-related Training") +
  theme_1 +
  scale_x +
  scale_y +
  geom_text(aes(label = round(perc, 2)), vjust = 1.4, color = "white")


# plot job related training over waves
plot3 <- ggplot(ft_data %>% group_by(wave) %>% summarise(perc = mean(nfjr_g_dummy)),
aes(wave, perc)) +
  geom_col(color = "black", fill = "purple") +
  ggtitle("Job-related Training with logical imputation") +
  theme_1 +
  scale_x +
  scale_y +
  geom_text(aes(label = round(perc, 2)), vjust = 1.4, color = "white")


#combine the plots
  grid.arrange(plot1,
               plot2,
               plot3,
```

```
  top = textGrob(
   "Training participation rate: Wave 2 to Wave 13",
   gp = gpar(
    fontsize = 20,
    font = 3,
    face = "bold"
   )
  ))

##############################################################################
# create plots with average number of courses after wave - differentiated
##############################################################################


# average number of courses per person over time
plot4 <- ggplot(ft_data %>% group_by(wave) %>% filter(n_courses > 0) %>% summarise(number =
mean(n_courses)), aes(wave, number)) +
 geom_col(color = "black", fill = "purple") +
 ggtitle("All Training") +
 theme_1 +
 scale_x +
 scale_y2 +
 geom_text(aes(label = round(number, 2)), vjust = 1.4, color = "white")


# average number of job-related courses per person over time (without logical imputation)
plot5 <- ggplot(ft_data %>% group_by(wave) %>% filter(n_courses > 0) %>% summarise(number =
mean(n_jobtotal)), aes(wave, number)) +
 geom_col(color = "black", fill = "purple") +
 ggtitle("Job-Related Training") +
 theme_1 +
 scale_x +
```

```
  scale_y2 +
  geom_text(aes(label = round(number, 2)), vjust = 1.4, color = "white")

# average number of job-related courses per person over time (with weights adjusting for random selection of courses)
plot6 <- ggplot(ft_data %>% group_by(wave) %>% filter(n_courses > 0) %>% summarise(number = mean(n_jobtotal_w)), aes(wave, number)) +
  geom_col(color = "black", fill = "purple") +
  ggtitle("Job-Related Training (weighted)") +
  theme_1 +
  scale_x +
  scale_y2 +
  geom_text(aes(label = round(number, 2)), vjust = 1.4, color = "white")

# average number of job-related courses per person over time (with weights adjusting for random selection of courses)
plot7 <- ggplot(ft_data %>% group_by(wave) %>% filter(n_courses > 0) %>% summarise(number = mean(n_jobtotal_g)), aes(wave, number)) +
  geom_col(color = "black", fill = "purple") +
  ggtitle("Job-Related Training (logical imputation)") +
  theme_1 +
  scale_x +
  scale_y2 +
  geom_text(aes(label = round(number, 2)), vjust = 1.4, color = "white")

#combine the plots
grid.arrange(plot4,
             plot5,
             plot6,
             plot7,
             top = textGrob(
               "Average number of courses: Wave 2 to Wave 13",
```

```
  gp = gpar(
    fontsize = 20,
    font = 3,
    face = "bold"
  )
))
```