The German
Socio-Economic
Panel study

492

SOEP Survey Papers

Series G – General Issues and Teaching Materials

SOEP — The German Socio-Economic Panel study at DIW Berlin

2018

Do-files for working with SOEP spell data

Maik Hamjediers, Paul Schmelzer, Tobias Wolfram

Please cite this paper as follows:

Maik Hamjediers, Paul Schmelzer, Tobias Wolfram. 2018. Do-files for working with SOEP spell data: SPELL_TO_PANEL, PANEL_TO_SPELL, SPELL_TO_SPELL, EVENT_TO_SPELL, and SPITTING_OVERLAPS. SOEP Survey Papers 492: Series G. Berlin: DIW/SOEP

ISSN: 2193-5580 (online)
DIW Berlin
German Socio-Economic Panel (SOEP)
Mohrenstr. 58
10117 Berlin
Germany
soeppapers@diw.de

Do-files for working with SOEP spell data: SPELL_TO_PANEL, PANEL_TO_SPELL, SPELL_TO_SPELL, EVENT_TO_SPELL, and SPITTING_OVERLAPS

Maik Hamjediers, Paul Schmelzer*, Tobias Wolfram

***Corresponding author. DIW Berlin /SOEP, pschmelzer@diw.de**

Introduction

Spell, duration or event history data are used frequently in the social sciences. A typical spell dataset contains one or multiple observations for each subject, each covering a span of time (a spell) during which the subject is in a given state, such as employed or unemployed. Handling this kind of data can be cumbersome, especially when you want to supplement it with panel data or vice versa. Alternatively, you might want to combine two spell files or enrich spell data with information on events while keeping the spell structure of your data. To accomplish these tasks, we provide four do-files to facilitate the data management.

- **PANEL_TO_SPELL:** This do-file helps to supplement information contained in a spell dataset with panel information. For instance, one can enrich employment spells with annually surveyed information on the branch of industry the respondents are working in while maintaining the spell structure (for an example see Table 1, Panel D).
- **SPELL_TO_PANEL:** Alternatively, you might want to do the opposite and enrich a panel dataset with information from spell data. An example for such a procedure is extracting incidences or the duration of potential unemployment spells between two annually conducted interviews. The resulting file will be in a panel structure (for an example see Table 1, Panel E).
- **SPELL_TO_SPELL:** We provide another do-file for combining two spell files. For example you might want to supplement employment histories stored in one spell dataset with marital histories contained in another one. The spell structure will be maintained in the resulting dataset, enabling for instance to model two different processes over time such as the selection into marrying and how this affects employment history (for an example see Table 2, Panel D).
- **EVENT_TO_SPELL:** Finally, you may want to introduce events to spell data while keeping a spell structure. This occurs for instance when you are interested in adding the event of a giving birth for the first time to a spell dataset of marital histories. (for an example see Table 3, Panel D).

Additionally to these four do-files for combining two datasets, some spell data might contain overlapping spells meaning that one spell starts before a previous spell ends. For many analyses or for data preparation procedures it is useful to adjust overlapping spells in such a way that a spell is split at the beginning and the end of its intersection with another spell (or several spells). In the following the spells that result from the original spell by splitting procedures are called subspells. Such subspells do not intersect with any other spells (of subspells); they are at maximum parallel to another subspell of the same duration but with a different state. Imagine, there is an overlap of two spells; vocational training and part-time employment, respectively (this example is also depicted in Table 4). Part-time employment starts in February 1995 and lasts until the end of May 1986; vocational training starts within this spell and overlaps with it from March 1995 onwards. At the same time the end of the part-time spell falls within the vocational training at May 1995 (Table 4, Panel A). After splitting each original spell, we end up with four subspells (Table 4, Panel C): spells are split according to the time of intersection such that the time of the overlap is depicted by two subspells which have the same beginning and ending (March 1995 and May 1995). To adjust overlapping spells in such a way you should use the **SPITTING_OVERLAPS** do-file. We strongly recommend to run this do-file before combining two spell data (by using the SPELL_TO_SPELL do-file), introducing information from panel to spell (PANEL_TO_SPELL), or introducing information on events to spell data file (EVENT_TO_SPELL).

We automated all the procedures for SPITTING_OVERLAPS, for SPELL_TO_SPELL, for PANEL_TO_SPELL and for EVENT_TO_SPELL do-files such that you do not need to make any adjustments in the do-files except for defining some initial inputs such as location of your data files. Though we also automatized most generating procedures for the SPELL_TO_PANEL do-file, it is not possible to completely automate the whole process because the information to be extracted from the spell data depends on your research question and on assumptions you would like to make. We indicate the places in the do-file where you should intervene and give examples of some important procedures.

All do-files are written to manage monthly spell data since handling such data is more complicated than using spell data on an annual basis. While the do-files SPITTING_OVERLAPS and SPELL_TO_SPELL also work with yearly based spell data, SPELL_TO_PANEL and PANEL_TO_SPELL are superfluous since information form spell data can be easily introduced to panel data and vice versa. Additionally, all do-files are written to handle individual data since all SOEP spell datasets are based upon the individual level. If you want to merge information on the household level to the spell data you can do so by merging it first to the individual level and then apply the provided PANEL_TO_SPELL or SPELL_TO_PANEL do-file.

Censored data

When working with spell data one should pay attention to censoring of spells. Censoring refers to information about the accuracy of the beginning and ending of a spell. If the exact beginning of a spell is unknown this spell is marked as left-censored; if the ending is unknown then it is right-censored; if both, the beginning and ending are unknown this spell is right- and left-censored. The censoring information is written into a censor variable which distinguishes for each spell four states: uncensored, left-censored, right-censored, left- and right-censored.[1]

When splitting spells by adding events, information of panel data or of other spell files one has to be careful in assigning the correct censoring information for each newly generated subspell. For instance, imagine a right and left censored spell of employment which is supposed to be supplemented with an uncensored spell of being married contained in a second spell file (see Table 2). If we merge these spell files together, we end up with one subspell of employment which lasts until the beginning of the marriage and another subspell which continues from the marriage on. While the original employment spell was left and right censored, the first newly generated subspell is only left censored and the second subspell is only right censored since only the overall beginning and ending of the original spells were unknown whereas the beginning and the timing of the subspells is induced by the beginning of the married spell. Without properly adjusting the censor variable you are very likely to end with wrong results (this could affect descriptive statistics as well as regression models). Therefore, depending on your research question you should be aware of such necessary adjustment. To deal with the problem of censoring we wrote two programs (`censor_unfold` and `censor_contract`) stored in CENSORING_PROGRAMS do-file. However, these programs are only applicable to SOEP spell data. You are free to change `censor_unfold` and `censor_contract` programs if you want to apply these do-file to other spell datasets than the SOEP (You can find the description of both programs in Appendix A).

Intuition of the general procedure

[1] There might be also subdivisions of left- and right- censoring. The beginning of a spell might for example be unknown either because the spell follows a period of missing information, because the start of the spell is also the start of the period of observation or because the exact timing of the transition into a status is unknown and could only be reconstructed as an estimate from panel data. Such information on the nature of left-censoring can be also stored in the censor variable and the distinction of kinds of censoring applies in a similar way for the endings of spells and right-censoring.

The underlying idea of combining data is similar for all provided do-files. Suppose we are dealing spell data on a monthly base for vocational training and part-time employment and we want to combine it with panel data containing information on the branch of industry and marital status (this example is also depicted in Table 1). Based upon the beginning and ending of each spell, we can then expand the spell data by partitioning each spell into subspells lasting each only month (Table 1, Panel C). This enables us to merge information from panel data by the month of each interview or information on events by using the month of the event.

Depending on the desired data structure of the resulting file (spell or panel data), we collapse the dispensable months and end up with data in the spell structure (PANEL_TO_SPELL; see Table 1, Panel D or EVENT_TO_SPELL; see Table 3) or we just extract the information from spell data and keep the time of the interview to end up with panel structured data (SPELL_TO_PANEL; see Table 1, Panel E). If you combine two spell datasets or deal with overlapping spells we partition all spells of interest in the same way as described above and merge the spell data or the overlapping spells by all months of the subspells (compare respectively SPELL_TO_SPELL and Table 2 or SPLTTING_OVERLAPS and Table 4). Though this general logic is quite the same in all provided do-files, the commandos in the do files vary considerably. Therefore, we wrote separate descriptions of how to use each do-file. We also make extensive comments in each file to facilitate working with them. If you want to replicate the fictional examples presented in Table 1 to Table 4, we provide corresponding datasets in the folder `examples/example_input_data`. For your research question you are invited to adjust provided do-files to your needs.

The do files you will find on

http://www.diw.de/documents/dokumentenarchiv/17/diw_01.c.581431.de/do-files_spell-data.zip

Details of merging spell dataset to panel dataset (SPELL_TO_PANEL do-file)

The aim of the SPELL_TO_PANEL do-file is to extract information from spell data and integrate it to panel data. Thereby, it enables to enrich panel data with information that occurs between interviews (for example, if you want to construct a variable reflecting the duration of unemployment between interviews). In the following, we will explain the code of the do-file. Please note that it is not possible to provide automated procedures for all potential information one could extract from spell data. Nevertheless, we automated the procedures for incidences of different states (for example vocational training and part-time employment) and for duration of these states between interviews. Additionally, we provide examples of how to utilize further information from the spell data manually. If you want to use only information on incidence and duration of various states it is only necessary to read the section of how to set the globals (section 1); if you want to utilize information other than incidences and durations you should read all sections, especially, the section “How to construct your variables” (section 4.2).

In the following we describe each section of the do-file SPELL_TO_PANEL with each heading in the description referring to the sections marked in the do-file. In the description of the do-file we employ an example depicted in Table 1. In our example we want to combine two spells of the type vocational training and part-time employment of the ARTKALEN dataset with SOEPlong data on the individual level that entail information on marital status and branch of industry. You can replicate the example in Table 1 using the do-file and corresponding data stored in the folder `examples/example_input_data` .

1. Definition of globals

1.1 In the **first** part ”Globals defining your paths” of the do-file you need to define paths indicating the folder where your do-files, data files, temporary data files, and output data files are located.

1.2 In the **second** part ”Globals defining your data” globals for the variables of the spell and the panel data are defined.

1.2.a) Define globals for the spellfile

`spellfile:` Name of the spellfile (in our example ARTKALEN)

`s_pid:` Identifier for the individuals (in ARTKALEN it is `persnr`) `begin` and `end:` Define the begin and end of each spell (in ARTKALEN `begin` and `end`)

`spelltype`: Variable that contains information of the status of each spell (in ARTKALEN the variable is `spelltype` and contains 15 categories: part-time employment, full-time employment, vocational training and so forth)

`spells`: Value of spells or values of several spells of which information should be extracted (in our example we use `"3 4"` which refer in ARTKALEN to the value 3 and 4 of the variable `spelltyp` and stand for part-time employment and vocational training spells)

`spellnr:` Identifier of spell number of each person

`censor:` Variable that contains information of the censoring in the spell data

1.2.b) Define globals for the panel file

`panelfile:` Name of the panelfile (in our example pl)

`p_pid:` Identifier for the individuals (in pl it is `pid`)

`year`: Year of interview (in SOEP it is denoted by `syear`)[2]

`month`: Month of interview[3]

`variables_panel:` List of variables you would like to use of your panel file

2. Preparation of datasets

2.1 Preparation of spell data

First, after opening the spell data we synchronize the observation identifier (personal id or household id) between spell and panel data by renaming the personal identifier of the spell data to the identifier used in the panel data. In our example we rename variable `persnr` into `pid`.

In the next step, we keep only those spells that were set by the global `spells.` The global `spells` defines the values of the variable `spelltyp` in your data (in our example we want to keep “vocational training” and “part-time employment and defined the global `spells` to

[2] From SOEP Version 34 on, you should use the information provided in the variable `iyear`.

[3] If the month of interview is not provided within the `pl` dataset, you need to merge it from the `pl2` dataset using

```
merge 1:1 ${p_pid} ${year} using ${ MY_IN_PATH}\\pl2, ///
nogen keepusing(${month}) keep(1 3)
```

before running the provided do-file.

the respective values). Additionally, we adjust the endings of all spells to be exclusive by adding one month to the ending of each spell.[4] Note, that the values of the ending and beginning variables for spell data in SOEP are usually provided as the number of months elapsed since January 1983 (the beginning of the first spell with the value of 135 thus represents 135 months starting from January 1983 and therefore corresponds to March 1994).

Finally, we partition each spell into subspells each lasting only one month. Therefore, we first generate a variable for the duration of each spell (`_spell_duration`)[5] and second use the "expand"-command to multiply each spell according to its duration. E.g. the first spell in our example lasts 4 months (139-135) and after expanding we end up with four subspells (see Table 1, Panel C). Afterwards, we adjust the beginning and ending of all spells in such a way that the duration of each subspell of an original spell amounts to one month and the ending of each subspell equals to the beginning of the next subspell.

2.2 Preparation of panel data

Now, we describe how to prepare the panel dataset. The "`use varlist using`"-command extracts indispensable variables (in our example: `pid`, `year` and `month of interview`) and variables relevant to your research (which were defined in the global `variables_panel`) from the panel file. In the SOEP data, the months in which interviews took place are not always known because some interviews are conducted by mailback surveys. Since most of the SOEP-Interviews take place in February we impute this month for all missing values and generate a marker variable "`not_valid`" for those observations. For merging panel data to spell data, the panel data should be first transformed in long format, which is already the case for SOEPlong. Yet, SOEPlong has the year of observation and the associated month saved in separate variables. Using both variables we utilize the date-function "`ym`" of Stata to construct a variable called `begin` which denotes elapsed months similarly to preparation of the spell data described above. In contrast to SOEP, the reference

[4] The ending month of a spell in any spell data can be of two kinds: Either the month denoted in the variable can represent the last month in a respective status or the month can represent the month when the status is past and therefore does not apply anymore in the month denoted in the variable. The first case, we call inclusive, whereas the second case is denoted as exclusive. For instance, if the ending variable of an employment spell takes on the value 12 an inclusive ending would imply that the respective individual is still in employment, whereas an exclusive ending implies that the employment ended already in the month 11 and the month 12 is the first month in which it does not apply anymore. Exclusive endings further imply for complete data of disjoint spells that the ending of one spell is equal to the beginning of the following spell.
While we adjust all ending months throughout the procedures into the exclusive form, we adjust them all back into the inclusive form since all spell data in SOEP is delivered in this manner.

[5] Furthermore please note that SPLITTING will set `duration` = 0 for missing values of the beginning or end of a spell and therefore disregards them when expanding.

month in Stata is January 1960; hence, we subtract 275 (months) from the newly generated variable to match its values with the `begin` variable in the SOEP spell data which starts in January 1983 as described above.

3. Merging

Both datasets are now sorted by the unique personal identification numbers of the SOEP-participants (denoted by global `p_pid)` and the starting month of the spell and month of interview of the panel data (denoted by global `begin`). Using both identifiers we merge the datasets by the many-to-one-scheme (`merge m:1`): Potential multiple months of the spell data (many) will be merged with each unique person-month-combination of each interview (one) in a respective month (see Table 1, Panel C).[6] In such a way we enrich the panel data by information of the spell data. In our example the person with time of interview in March 1994 was in vocational training. Now we have a single dataset in which all spell information we are interested in are merged to the same month and year at which panel observations took place.

4. Creating the final datasets

4.1 Automatic generation of incidence and duration of a respective state

Now it is time to choose what to do with the newly gained information. In this section of the do-file we automatized the generation procedure of variables for incidences and durations of respective spells. In our example with two states (vocational training and part-time employment) you will end with four automatically generated variables: incidence_of_the_state_3, incidence_of_the_state_4, duration_of_the_state_3, and duration_of_the_state_4. Variables incidence_of_the_state_3, incidence_of_the_state_4 variables provide information weather a person was in vocational training or/and in part-time employment between the interviews (incidence of a state). Variables duration_of_the_state_3 and duration_of_the_state_4 inform you about the duration in these sates. **Note, depending on your research question one should pay attention to the censoring variable of the spell data.** For instance in our example part-time employment is left censored. It implies that the beginning of the part-time employment is not exactly known and may have started before February 1995, hence, part-time employment may last longer than two months. To handle this problem you might drop observation of those persons who are either right, left, or right-and left censored. Depending on

[6] Note, we apply many-to-one-scheme because some spell data might contain parallel spells, i.e. in a given month an individual is in part-time employment and simultaneously in vocational training.

your research question, feel free to adjust the censoring variable in the section 4.1 according to your needs.

However, the four variables are insufficient for other purposes; one might also be interested in for instance specific transitions between part-time employment and vocational trainings or the number of interruptions of vocational training between interviews. If you need such specific information you are asked to construct your own variables (see the next section).

4.2 How to construct your own variables

Here, we present couple of examples how to construct different information from the spell data. First, we demonstrate how to construct the code for variables of incidences and the duration of a state manually (we constructed manually variables incidence and duration of vocational training and part-time employment: incidence_voc_tr, incidence_part_time, duration_voc_tr, and duration_part_time). In our example do file we take into account the problem of censoring (the exactly steps are explained in the do file). Additionally, we demonstrate how to construct a variable that reflects the number of interruptions of vocational training between interviews (interruption_voc_tr_nr). In our example between both interviews (in March 1994 and April 1995) person was in vocational training until beginning of July 1994 (or in other words until the end of June 1994) and started with vocational training again in February 1995. Therefore, the person interrupted his/her state one time between interviews. One might be also interested in information on the start and the end date of vocational training in the given year. The way to collect such information also see the do-file. There are many other information you might want utilize from the spell data. **Note**, **after you prepared your variables you should introduce them into the global `variables_between_interviews`. This global guarantees that the information will be assigned to the observation of the next interview and hence it assures that the information is maintained in the final panel structure.**

5. Reducing to panel data structure

Finally, in the part 6 of the do-file we compress the data into a panel structure. However, since some of spell data might have two or more parallel spells (for example vocational training and part-time employment at the time of interview), you might end with multiple observations for the same interview in a respective year. In our example we have to deal with two observations for the survey year 1995. Since panel models in Stata use only one time point per person (or household) (for example by using xtset command) we keep only one observation for each interview.

Details of merging panel dataset to spell dataset (PANEL_TO_SPELL do-file)

The aim of the PANEL_TO_SPELL do-file is to enrich the spell data with information from panel data for (for example, if you want to introduce information on the branch of industry available in panel data to employment spells in spell data). Adding information to spell data is usually used to enhance the analysis of spell data with further variables in methods based on the spell structure, such as survival models. In the following, we will explain the code of the do-file. Depending on assumptions you are willing to make, you can add information of an interview of the panel in three different ways: The added information may apply for the duration of a whole spell, it may apply for the consecutive months of a spell from the time of the interview on or it may apply only for the rest of the year of the interview. The provided do-file can handle all three options and we describe below how you can apply each. Though we automatized the procedures we strongly recommend reading all sections. It helps you to understand the assumptions you opt for.

In the following we describe each section of the do-file PANEL_TO_SPELL with each heading in the description referring to the sections marked in the do-file. In the description of the do-file we employ an example depicted in Table 1, Panel A-C and E. In our example we want to combine two spells of the type vocational training and part-time employment of the ARTKALEN dataset with SOEPlong data on the individual level that entail information on marital status and branch of industry. You can replicate the example in table 1 using the do-file and corresponding data stored in the folder `examples/example_input_data` .

1. **Definition of globals**

   1.1 In the **first** part ”Globals defining your paths” of the do-file you need to define paths indicating the folder where your do-files and data files are located where you want to save the merged dataset. In this section you should also define the path where you copied the Censor_Programs do-file. In the do-file several programs are defined to guarantee correct censoring of your censor variable.

1.2 In the **second** part ”Globals defining your data” globals for the variables of the spell and the panel data are defined.

1.2.a) Define globals for the spellfile

`spellfile:` Name of the spellfile (in our example ARTKALEN)

`s_pid:` Identifier for the individuals (in ARTKALEN it is `persnr`)

`begin` and `end:` Define the begin and end of each spell (in ARTKALEN `begin` and `end`)

`spelltype`: Variable that contains information of the status of each spell (in ARTKALEN the variable is `spelltype` and contains 15 categories: part-time employment, full-time employment, vocational training and so forth)

`spells`: Value of spells or values of several spells of which information should be extracted (in our example we use `"3 4"` which refer in ARTKALEN to the value 3 and 4 of the variable `spelltyp` and stand for part-time employment and vocational training spells)

`spellnr:` Identifier of spell number of each person.

`censor:` Variable that contains information of the censoring in the spell data

1.2.b) Define globals for the panel file

`panelfile:` Name of the panelfile (in our example pl)

`p_pid:` Identifier for the individuals (in pl it is `pid`)

`year`: Year of interview (in SOEP it is denoted by `syear`)[7]

`month`: Month of interview[8]

`variables_panel:` List of variables you would like to use of your panel file.

`complete_spell:` List of variables of which the panel information is assigned to a whole spell (appropriate for time-constant variables)

`within_following_spell:` List of variables of which the panel information is carried forward within a given spell starting at the time of interview

[7] From SOEP Version 34 on, the information is provided in the variable `iyear`.

[8] If the month of interview is not provided within the `pl` dataset, you need to merge it from the `pl2` dataset using

```
merge 1:1 ${p_pid} ${year} using ${ MY_IN_PATH}\\pl2, ///
nogen keepusing(${month}) keep(1 3)
```

before running the provided do-file.

`within_following_year:` List of variables of which the panel information from is carried forward until the next time of interview

2. Preparation of datasets

2.1 Preparation of spell data

First, after opening the spell data we synchronize the observation identifier (personal id or household id) between spell and panel data by renaming the personal identifier of the spell data to the identifier used in the panel data. In our example we rename variable `persnr` into `pid`.

In the next step, we keep only those spells that were set by the global spells. The global spells defines the values of the variable `spelltyp` in your data (in our example we want to keep "vocational training" and "part-time employment and defined the global `spells` to the respective values). Additionally, we adjust the endings of all spells to be exclusive by adding one month to the ending of each spell.[4] Note, that the values of the ending and beginning variables for spell data in SOEP are usually provided as the number of months elapsed since January 1983 (a value of 135 thus represents 135 months starting from January 1983 and therefore corresponds to March 1994).[9]

Finally, we partition each spell into subspells each lasting only one month. Therefore, we firstly generate a variable for the duration of each spell (`_spell_duration`)[10] and secondly use the "expand"-command to multiply each spell according to its duration. E.g. the first spell in our example lasts 4 months (139-135) and after expanding we end up with four subspells (see Table 1, Panel C). Afterwards, we adjust the beginning and ending of all spells in such a way that the duration of each subspell of an original spell amounts to one month and the ending of each subspell equals to the beginning of the next subspell.

2.2 Preparation of panel data

Now, we describe how to prepare the panel dataset. The "`use varlist using`"-command extracts indispensable variables (in our example: `pid`, `year` and `month of interview`) and variables relevant to your research (which were defined in the global

[9] Though we demonstrate the preparation of the data using two different spells ("vocational training" and "part-time employment") we recommend you to keep only one spell. If your research question requires preparation of several spells in one data (for example you want to run simultaneous survival models with two destination states) then we strongly recommend to get rid of overlapping spells using SPITTING_OVERLAPS do-file (see the section on SPITTING_OVERLAPS).

[10] Furthermore please note that SPLITTING will set `duration` = 0 for missing values of the beginning or end of a spell and therefore disregards them when expanding.

variables_panel) from the panel file. In the SOEP data, the months in which interviews took place are not always known because some interviews are conducted by mailback surveys. Since most of the SOEP-Interviews take place in February we impute this month for all missing values and generate a marker variable "not_valid" for those observations. For merging panel data to spell data, the panel data should be first transformed in long format, which is already the case for SOEPlong. Yet, SOEPlong has the year of observation and the associated month saved in separate variables. Using both variables we utilize the date-function "ym" of Stata to construct a variable called begin which denotes elapsed months similarly to preparation of the spell data described above. In contrast to SOEP, the reference month in Stata is January 1960; hence, we subtract 275 (months) from the newly generated variable to match its values with the begin variable in the SOEP spell data which starts in January 1983 as described above.

3. Merging

Both datasets are now sorted by the unique personal identification numbers of the SOEP-participants (denoted by global p_pid) ,the starting month of the spell and month of interview of the panel data (denoted by global begin). Using both identifiers we merge the datasets by the many-to-one-scheme (merge m:1): Potential multiple months of the spell data (many) will be merged with each unique person-month-combination of each interview (one) in a respective month (see Table 1, Panel C).[11] In such a way we enriched spell data by information from the panel data. In our example person was at the second interview (in April 1995) single and in education. Though the information is available only to one specific time point (April 1995) making assumption we can transform this information for other time points of the spell.

4. Expanding information from panel to spells

4.1 Expanding information from panel to the whole spell

(global complete_spell)

For some variables it is reasonable to assume that the information gained at the time of interview applies also for the time before and after the interview and hence for the whole duration of a given spell: in our example a person worked in branch of industry "education " in April

[11] Note, we apply many-to-one-scheme because some spell data might contain parallel spells, i.e. in a given month an individual is in part-time employment and simultaneously in vocational training.

1995 (the month the second interview took place) and we could assume that this information on the branch of industry applies for the whole employment spell.[12] Therefore, it seems reasonable to expand the information for the whole time period of all spells that cover the time of interview at which the reports were gathered (since the spells of number 9 and 10 are parallel and occur both at the time of the interview the branch of industry is assigned to both and expanded for both whole spells). In our case we ascribe the information for branch of industry for the spells with spell number 9 and 10 (see Table 1, Panel C). If for certain variable you would like to make such an assumption you should list them in the global `complete_spell`.

4.2 Expanding information for following months only (global `within_following_spell` and `within_following_year)`

In the case that information of the time of the interview applies only to a spell from the time of the interview onwards you should list the respective variables in the global `within_following_spell`. In our example, we make the assumption that the marital status does not change in the following months after an interview; yet, in contrast to the branch of industry, we do not assigne the marital status of the interview to the time before the interview within a spell.[13] Therefore, we carry forward the information to the next months **within a given spell**. For example, a person reports to be married in March 1994 and we carry forward this information to the next months within the spell number 8 (Table 1, Panel C). In such a case you might end with the missing information in your resulting data (see marital status in Panel D in Table 1). If you can assume, that the information of the panel data does not necessarily change together with the information captured by the spell data, one could also carry forward the information not only until the end of a given spell but **until the next interview**. List all variables for which this assumption applies in the global `within_follow_year`. In such a case the information will be assigned to all following months until the next interview in April 1995 (the example is not demonstrated in Table 1).

[12] For didactical reasons we do not take into account potential job changes. The status in ARTKALEN only refers to the labor market status and not to job history. While for example a person might be continuously in full-time employment, he or she might change the employers without changing the actual status of full time employment. Job changes are not denoted in ARTKALEN and you would have to introduce the information from panel data.

[13] Of course you can also make an assumption with regard to marital status and expand the information to the whole spell. It depends on your research question.

5. Reducing to spell data

Finally, in the part 6 of the do-file we collapse those spells that have identical information with respect to combination of variables introduced from the panel data. In the example depicted in Table 1, Panel C the original spell with the number 8 is depicted amongst others by four month-long subspells which contain the same information with respect to marital status and branch of industry starting from March 1994 until July 1994. These four subspells are compressed into one spell starting from February 1995 to April 1995. Therefore we identify the beginning of the first and the ending of last subspell for each change in the variables of the panel data with the indicator-variables `ch_first` and `ch_last`.[14] Additionally, we identify first and last observation of each original spell by generating variables `first` and `last`. Using these variables we compress the monthly subspells back into a manageable spell structure. To restore the censor-variable we apply the program `censor_contract.` Finally, we adjust endings to their original form making them “exclusive” (substract one month for variable end).[4]

As you can see in the Table 1, we started with three spells (Panel A) and ended up with five spells (Panel D). While the first spell (with spell number 8) were left unchanged, other spells were split in two spells to account for the changes in the variables added form the panel data.[15]

Note, when working with spell data one should pay attention to the censoring variable. To address the problem we again refer to our example in the Table 1, Panel D. In our example the part-time employment is left censored. It implies that “part-time employment” started before February 1995. Therefore, depending on your research question one should decide how to deal with this problem.

Details of combining two spell datasets (SPELL_TO_SPELL do-file)

The aim of the SPELL_TO_SPELL do-file is to combine two datasets which both contain spell data. Such a combination of two (or several) spell datasets might be used in several ways. For example you might want to enrich information of one spell dataset by introducing

[14] You may also include an indicator of the each survey wave (i.e. the year of the survey) in the list of variables which are used for checking. By this, your final spell-dataset will be also split by survey-year.

[15] For working with survival models in Stata (using `stset`) it is alike whether you have got one spell or several subspells. It just necessary to account for the multiple-record-per-subject structure by invoking the `id`-option.

information from the other dataset to model processes over time of two kinds (such as employment trajectories and their effect on the transition into marriage).

In the following we describe each section of the do-file SPELL_TO_SPELL with each heading in the description referring to the sections marked in the do-file. In the description of the do-file we employ an example from Table 2 which depicts the combining of spell data with information on employment status (from ARTKALEN) with spell data containing information on the marital status (from BIOMARSM). If you are less interested in the logic of preparation of the data you should carefully read section on how to set the globals (section 1). If these are set correctly the rest of the do-file can be executed unsupervised. Before combining spell datasets, we strongly recommend to adjust intersecting spells by using the SPITTING_OVERLAPS do-file which is described below. Of course, if you are using only one kind of spells in each spell data file (for example only "full-time employment" of ARTAKLEN data file and "married" in BIOMARSM data file) it is not necessary to use the SPITTING_OVERLAPS do-file.

1. Definition of globals

1.1 In the **first** part "Globals defining your paths" of the do-file you need to define paths indicating the folder where your do-files and data files are located where you want to save the merged dataset. In this section you should also define the path where you copied the Censor_Programs do-file. In the do-file several programs are defined to guarantee correct censoring of your censor variables.

1.2 In the **second** part "Globals defining your data" globals for the variables of both spell datasets. Note, that we call the first dataset the "master" dataset and the second dataset the "using" one.

1.2.a) Define globals for the master dataset

`spellfile:` Name of the master spell dataset (in our example ARTKALEN)

`s_pid:` Identifier for the individuals (in ARTKALEN it is `persnr`)

`begin` and `end:` Define the begin and end of each spell (in ARTKALEN `begin` and `end`)

`spelltype`: Variable that contains information of the status of each spell (in ARTKALEN the variable is `spelltype` and contains 15 categories: part-time employment, full-time employment, vocational training and so forth) spells: Value of spells or values of several spells of which information should be extracted (in our example we use "1 3 4" which refer in ARTKALEN to the value 1, 3 and 4 of the variable spelltyp and stand for full-time employment, part-time employment and vocational training spells)

spellnr: Identifier of spell number of each person

name: How to rename the spelltype variable of the master dataset (in our example spelltype is renamed into employment_status)

censor: Variable that contains information of the censoring in the master data

1.2.b) Define globals for the using dataset

spellfile2: Name of the using spell dataset (in our example BIOMARSM)

s_pid2: Identifier for the individuals (in BIOMARSM it is persnr)

begin2 and end2: Define the begin and end of each spell (in BIOMARSM begin and end)

spelltype2: Variable that contains information of the status of each spell

spells2: Value of spells or values of several spells of which information should be extracted (in our example we use "1 2" which refer in BIOMARSM to the value 1 and 2 of the variable spelltyp and stand respectively for being single and married)

spellnr2: Identifier of spell number of each person

name2: How to rename the spelltype variable of the using dataset (in our example spelltype is renamed into marital_status)

censor2: Variable that contains information of the censoring in the using data

2. Preparation of datasets

The preparation of both spell datasets, the master and using one, follow the exact same structure, except for the renaming of variables which ensures the merging procedure.

First, after opening the master spell data we synchronize the observation identifier (personal id or household id) between the master and using datasets by renaming the personal identifier of the master spell data to the identifier used in the using spell data. In our example, however, there are no changes since personal id is identical in both datasets (persnr). In the next step, we keep only those spells that were set by the global spells. The global spells defines the values of the variable spelltyp in your spell data. Additionally, we adjust the endings of all spells to be exclusive by adding one month to the end of each.[4] Note, that the values of the ending and beginning variables for spell data in SOEP are usually provided as the number of elapsed month since January 1983 (the beginning of the first spell with the value of 36 thus represents 36 months starting from January 1983 and therefore corresponds to December 1985).

In the next step we unfold each spell into subspells of duration of a single month. Therefore, we firstly generate a variable for the duration of each spell (_spell_duration)[16] and secondly use the "expand"-command to multiply each spell according to its duration. E.g. the first spell in our example lasts 4 months (139-135) and after expanding we end up with four subspells (see Table 2, Panel C). Afterwards, we adjust the beginning and ending of all spells in such a way that the duration of each subspell of an original spell amounts to one month and the ending of each subspell equals to the begin of the next subspell.

For the using spell file we additionally rename three variables denoted by following globals: ${begin2}, ${end2}, ${pid2} (in our example, however, these variables are identical on both datasets and no changes are made).

3. Merging

Both datasets are saved and sorted by the unique personal identification numbers of the SOEP-participants (denoted by global pid) and the begin date of each spell (denoted by global begin). Using both identifiers we combine both datasets. If you using several spelltypes for each spell file one might end up with potential parallel spells in both spell files. In such cases Stata's merging command should not be employed since no unique combinations of the identifier variables is given. Therefore, we employ the joinby command which results in separate observations for each combination of the identifier variables. For instance, if you have two parallel spells in the master file (such as part-time employment and vocational training) and two parallel spells in the using file (being in a relationship and married, but separated) you end up with four spells each denoting one combination of the employment status and marital status.

[16] Furthermore please note that SPLITTING will set duration = 0 for missing values of the beginning or end of a spell and therefore disregards them when expanding.

4. Creating the final datasets

We end up with a combined dataset that is splitted on monthly basis (duration for each subspell is 1). Since this structure is inflated, we compress the subspells again into a manageable spell structure according to the changes of the two spelltypes. In our example we got five combinations (part-time employment with single, vocational training with single, full-time employment with single, and full-time employment with married, and missing information in the employment status with married). The combination for instance of part-time employment and begin single consists of four subspells and the aim is to reduce these four subspells to one spell starting at 36 and ending at 40 elapsed months. To define the starting and the ending month of each combination we construct variables that mark the first and last subspell of a given combination by referring changes in both, the marital and employment status. By collapsing subspells to one spell we are almost done with the preparation of the final dataset. We just have to contract the previously “unfolded” censor variables of each original spell data using the program `censor_contract`. Finally, we adjust the ending months to their original form making them “exclusive” (substract one month for variable end).[4] In the last step we get rid of variables that we used in our procedures and save the final dataset.

After you combined two spell files you can of course generate a new variable for spelltype that additionally contains information on the combination of both states (for example part-time and single, full-time and single and so forth). After you have combined two spell files it is also possible to supplement the resulting spell data with further spell

Note, depending on your research question one should pay attention to the censoring variable. For instance, take the right and left censored spell of full-time employment which is supplemented with the left censored spell of being married in our example in Table 2. While the original full-time employment spell was left and right censored, the first newly generated subspell is only left censored and the second subspell is only right censored (since only the overall beginning and ending of the original spell were unknown whereas the split of the subspells is induced by the beginning of the married spell). However, since the married spell is left-censored, the timing of the splitting of the employment spell is inaccurate as well. The resulting file encompasses two censor variables: the censor variable of the master file and the censor variable of the using file and if contradictions between these occur you would have to decide how to deal with this problem.

Details of merging events to a spell dataset (EVENT_TO_SPELL do-file).

The aim of the EVENT_TO_SPELL do-file is to introduce information on events (for example birth of the child) to spells but keeping the spell structure of spell data. Using the example in the Table 3 we will demonstrate of how to combine spell data (from BIOMARSM) containing information on marital status with a dataset containing monthly information on giving a birth to the first child (BIOBIRTH).

In the following we describe each section of the do-file EVENT_TO_SPELL with each heading in the description referring to the sections marked in the do-file. In the description of the do-file we employ an example depicted in Table 3. In this example, a person is single from December 1985 to April 1986 and gets married at May 1986. The observation of the marriage ends in August 1986 because the person is no longer observed (right censoring). While being a single, the individual gives birth to the first child in February 1986. Joining the event data with the spell data on the marital histories, the original single spell gets split into two subspells (Table 3, Panel D): The first subspell (single no child) starts at December 1985 and ends at the date of childbirth (January 1986) and the second subspells (single and a child) starts at February 1986 and ends in April 1986. Because the event does not occur within the marriage spell (May 1986 to August 1986) this spell is left unchanged.You can replicate the example in the Table 3 using the do-file and corresponding data stored in the folder `examples/example_input_data` . In the following, we will explain the code of the do-file.

1. **Definition of globals**

   1.1 In the **first** part ”Globals defining your paths” of the do-file you need to define paths indicating the folder where your do-files and data files are located where you want to save the merged dataset. In this section you should also define the path where you copied the Censor_Programs do-file. In the do-file several programs are defined to guarantee correct censoring of your censor variables.

   1.2 In the **second** part ”Globals defining your data” globals for the variables of both datasets.

   1.2.a) Define globals for the spell dataset

   `spellfile:` Name of the master spell dataset (in our example BIOSMARSM)

   `s_pid:` Identifier for the individuals (in BIOMARSM it is `persnr`) begin and end: Define the begin and end of each spell (in BIOMARSM begin and end)

spelltype: Variable that contains information of the status of each spell

spells: Value of spells or values of several spells of which information should be extracted (in our example we use "1 2" which refer in BIOMARSM to the value 1 and 2 of the variable spelltyp and stand for being single and being married)

spellnr: Identifier of spell number of each person

censor: Variable that contains information of the censoring in the master data

1.2.b) Define globals for the event dataset

eventfile: Name of the event dataset (in our example BIOBIRTH)

event_pid: Identifier for the individuals (in BIOBIRTH it is persnr)

event_year: Variable containing the information on the year of the event

event_month: Variable containing the information on the month of the event

to_keep: List of variables you would like to keep from the dataset

2. Preparation of datasets

2.1 Preparation of spell data

First, after opening the spell data we synchronize the observation identifier (personal id or household id) between spell and event data by renaming the personal identifier of the spell data to the identifier used in the event data. In our example, this step is redundant since the identifiers are the same.

In the next step, we keep only those spells that were set by the global spells. The global spells defines the values of the variable spelltype in your data (In our example we want to keep "single" and "married" and defined the respective values in the global spells). Additionally, we adjust the endings of all spells to be exclusive by adding one month to the end of each.[4] Note, that the values of the ending and beginning variables for spell data in SOEP are usually provided as the number of elapsed month since January 1983 (the beginning of the first spell with the value of 36 thus represents 36 months starting from January 1983 and therefore corresponds to December 1985).

Finally, we unfold each spell into subspells of duration of a single month. Therefore, we firstly generate a variable for the duration of each spell (`_spell_duration`)[17] and secondly use the "expand"-command to multiply each spell according to its duration. E.g. the first spell in our example lasts 4 months (36 to 40) and after expanding we end up with four subspells (see Table 3, Panel C). Afterwards, we adjust the beginning and ending of all spells in such a way that the duration of each subspell of an original spell amounts to one month and the ending of each subspell equals to the beginning of the next sub-spell.

2.2 Preparation of event data

Merging the event data is actually quite similar to merging panel data; only in the latter case the interview is the event. Therefore, it is necessary to prepare the event data in a way that we end up with a variable containing the timing of the event. Depending on how your event data is set up, different steps might be necessary to get there. We provide code for data which is in wide format (each occurrence of the same kind of event is stored in one row in different variables) and which contains the information on the timing of the event in two separate variables (the month of the event is stored separately from the year of the event).

In a first step, we reshape this data into a "long format" (the same kind of event is stored in several rows but in one column) To make the reshape work, we included in the global definitions in section 1.2.b only the prefix of each variable without the part of the variable name which demarks the number of the events (e.g. while the month of birth of the first child is saved in `kidmon01` and the month of the second child is stored in `kidmon02`, we only use the prefix of `kidmon` in the global `event_month`). If your data is already in the long format or if you deal with events which only occur once per individual, you do not need to execute this step.

Secondly, for merging both datasets, it is necessary to define the timing of the event(s) in such a way that it is denoted by a variable which is similar to the time-variables used in the spell file described above. Hence, the timing of the events shall be defined as elapsed month since January, 1983. In our example the first child was born in February 1986, hence, 38 months elapsed since January, 1983. We utilize the function available in Stata "`ym`" to construct the elapsed date from the variable definied in `event_month` and `event_year`. In contrast to SOEP, the elapsed date in Stata begins with January 1960. For this reason we sub-

[17] Furthermore please note that SPLITTING will set `duration` = 0 for missing values of the beginning or end of a spell and therefore disregards them when expanding.

tract 275 (months) from the result generated by "ym"-command so that the value of 1 represents January, 1983 as it does in the SOEP spell data. Events in which neither its month of its year are known are disregarded completely. If the data you start with does have a different way of storing the information on the time of the event you would need to adjust the variable by yourself.

3. Merging

Both datasets are sorted by the unique personal identification numbers of the SOEP-participants (denoted by global s_pid) and a time variable (denoted by global begin). Using this identifier we combine both employing the joinby command which takes into account potential parallel spells as well as multiple events (e.g. the birth of twins). Based upon matches of the event-data and the spell-timing we generate a dummy event which indicates when an event happens as well as a variable counting the number of events over the whole observation period for each case.

If you want to extract any other kind of information form the event data, we additionally exemplify some helpful procedures:

For instance, we provide code for carrying forward the gender of the child until another child is born or until the end for the period of observation. Or you might also specify a time period of for instance three month after an event has happened. For the exact procedures see section 3.1 (constructing your own variables) in the do-file. s

4. Creating the final dataset

After combining both data sets we end up with a combined dataset that is splitted on monthly basis (duration for each subspell is 1). Now we want to compress dataset into spells which also lasts longer than one month based upon the combinations of spelltype and events. In our example we have to deal with three combinations (single without child, single with a child, married with a child). Each of these combinations might have redundant observations in the sense of multiple subspells for the same status. For example the combination single with a child consist of three subspells and the aim is to reduce these subspells to one (sub)spell starting at 38 and ending at 41 elapsed month. This is achieved by identifying the begining of the first and the ending of the last spell of each event by generating ch_first and ch_last variables. Additionally, we identify first and last observation of each original spell by generating variables first and last. Using these variables and after couple of adjustments we compress the data in manageable size. To restore our censoring variable we apply the program `censor_contract.` Finally, we adjust endings to their original form making them "exclusive" (substract one month for variable end).[4] After these adjustments we end up with three spells (Table 3, Panel D)

Details of splitting intersecting spells in a spell dataset (SPLITTING_OVERLAPS do-file)

Some spell data may contain spells which start while another spell has not ended yet. For some analysis or for data preparation it might be useful to adjust overlapping spells in such a way that a spell is split at the beginning and the ending of its intersection with another spell (or several spells). For example in Table 4 an intersection of two spells occurs (vocational training and part-time employment). While part-time employment starts at February 1995 and lasts until the end of the May 1986 vocational training starts within this spell and intersects at March 1995. At the same time the end of the part-time spell falls within the vocational training at May 1995 (Table 4, Panel A). After splitting each original spell we end up with four subspells (Table 4, Panel C): spells are split according to the time of intersections and the period of time when both statuses occur at the same time are denoted by two subspells which have the same beginning and ending date (March 1995 and May 1995).

In the following we describe each section of the do-file SPLITTING_OVERLAPS with each heading in the description referring to the sections marked in the do-file. In the description of the do-file we employ the aforementioned example of Table 4. If you are less interested in the logic of preparation of the data you should carefully read the section on how to set the globals (section 1). If these are set correctly, you should be able to execute the commands without supervision. **Note, depending on your research question you should pay attention to potential adjustment of the censoring variables.** For instance, take the right and left censored spell of vocational training which intersects with the right censored spell of being part-time employed in our example in Table 4. While the original vocational training spell was left and right censored, the first newly generated subspell is only left censored and the second subspell is only right censored (since only the overall beginning and ending of the original spell were unknown whereas the split of the subspells is induced by the ending of the part-time employment spell). However, since the part-time employment spell is right censored, the timing of the splitting of the vocational training spell is inaccurate as well. The resulting file encompasses two parallel subspells ranging from March 1995 to May 1995 but these are differently censored (the subspell of part-time employment is right censored while the subspell of vocational training is left censored). How to deal with this contradiction may depend on your research question and the current code does not carry forward the censoring from one spelltype to another.

If you want to use data generated by SPLITTING_OVERLAPS do-file for generation of data using other do-files (SPELL_TO_PANEL, PANEL_TO_SPELL, SPELL_TO_SPELL, or EVENT_TO_SPELL) you should save it under the original spell file name.

1. Definition of globals

1.1 In the **first** part ”Globals defining your paths” of the do-file you need to define paths indicating the folder where your do-files and data files are located where you want to save the transformed dataset. In this section you should also define the path where you copied the Censor_Programs do-file. In the do-file several programs are defined to guarantee correct censoring of your censor variables.

1.2 In the **second** part ”Globals defining your data” globals for the variables of the spell dataset

`spellfile:` Name of the master spell dataset (in our example ARTKALEN)

`pid:` Identifier for the individuals (in ARTKALEN it is `persnr`)

`begin` and `end:` Define the begin and end of each spell (in ARTKALEN `begin` and `end`)

`spelltype`: Variable that contains information of the status of each spell

`spells`: Value of spells or values of several spells of which information should be extracted (in our example we use `"3 4"` which refer in ARTKALEN to the value 3 and 4 of the variable `spelltyp` and for being part-time employed and in vocational training)

`spellnr:` Identifier of spell number of each person

`censor:` Variable that contains information of the censoring in the master data

2. Preparation of datasets

A spellfile consist of several spelltypes stored in one variable (for example in ARTKALEN in spelltyp). In the first step using a loop we store all spells of each spelltype in separate files.

For ARTKALEN you will end up with 15 spell files since the `spelltype` variable has 15 categories (full-time employment, part-time employment, unemployment and so forth). However, if you are not interested in combining all spelltypes you should use the global `spells` to define the values of the spelltyp variable (in our example we want to keep “vocational training” and “part-time employment” only and therefore set the global `spells` to the values 4 and 3).

Within a loop we perform following procedures:

To facilitate the preparation of dataset, we adjust the endings of all spells to be exclusive by adding one month to the ending of each spell.[4] Note, that the values of the ending and beginning variables for spell data in SOEP are usually provided as the number of months elapsed since January 1983 (a value of 135 thus represents 135 months starting from January 1983and therefore corresponds to March 1994).

We then partition each spell into subspells each lasting only one month. Therefore, we first generate a variable for the duration of each spell (`_spell_duration`)[18] and second use the “expand”-command to multiply each spell according to its duration. E.g. the first spell of part-time employment in our example lasts 4 months (139-135) and after expanding we end up with four subspells of part-time employment (see Table 3, Panel B). Afterwards, we adjust the beginning and ending of all spells in such a way that the duration of each subspell of an original spell amounts to one month and the ending of each subspell equals to the beginning of the next subspell.

The same procedures are performed for all spelltypes that were defined in global `spells` (in our example for vocational training).

3. Merging

In the first step we construct an auxiliary data file with zero observations and with empty variables named after the globals `pid` and `begin`. Using a loop of all spelltypes of the global spells we merge all `spell` files to this empty data file (in our example part-time employment and vocational training spell files) .

[18] Furthermore please note that SPLITTING will set `duration` = 0 for missing values of the beginning or end of a spell and therefore disregards them when expanding.

4. Creating the final dataset

After combining the stored files we end up with a dataset that is splitted on monthly basis (duration for each subspell is 1) and in which all information of the different spelltypes is stored separate variables. Overlapping subspells display the same beginning and ending and thus occur in one observation. In our example we have three subspells that belong to part-time employment and to vocational training (Table 4, Panel B). According to the combinations of spelltypes the monthly subspells are redundant and we can compress them again leaving us with the usual spell structure of spells with different durations. In our example the final dataset consists of two spelltypes and for the person in our example we end up with three combinations (part-time employment without vocational training, part-time employment and vocational training, and vocational training without part-time employment).[19] Each of these combinations might have redundant subspells. For example the combination part-time employment and vocational training consist of three subspells and the aim is to reduce these three subspells to one spell starting at 147 and ending at 150 elapsed months. To define the starting and the ending subspell of such combinations we construct `first` and `last` variables and the potential intermediate observations are dropped. Adjusting additionally the ending by passing on the information of the last subspell to the first subspell, we can also drop the observation of the last subspell and end up with one spell lasting the three months from 147 to 150. In a final step, we just have to adjust the censor variables of each original spell data using the program censor_contract the endings to their original form making them "exclusive" (substract one month for variable end).[4]

5. Reshaping into parallels

In the last step we reshape the parallel spelltypes. While until now each spelltype is stored in a separate variable, the reshaping saves all statuses in one variable with values accordingly to the spelltype of the starting dataset. This is also known as reshaping the data file from a "wide" format to a "long" format (parallel spells are stored in the same row but in separate observations; not shown in the Table).

[19] If you selected three spelltypes (by defining global spells) you will end up with a combination of three spelltypes and so forth.

Tables

Table 1: Example of merging panel and spell data

A: Original spell data

begin	end	status	spell nr	censor
[135] 1994 Mar	[138] 1994 Jun	[4] Vocational Training	8	[1] Uncensored
[146] 1995 Feb	[149] 1995 May	[4] Vocational Training	9	[1] Uncensored
[146] 1995 Feb	[149] 1995 May	[3] Part-Time Employment	10	[4] Left Censored

B: Original panel data

month of interview	year of interview	marital status	branch of industry
[3] March	1994	[1] married	[-2] Does not apply
[4] April	1995	[2] single	[80] Education

C: Spell data after splitting with panel information

		Information from spell data			Information from panel data	
begin	end	status	spell nr	censor	marital status	branch of industry
[135] 1994 Mar	[136] 1994 Apr	[4] Vocational Training	8	[1] Uncensored	[1] Married	[-2] Does not apply
[136] 1994 Apr	[137] 1994 May	[4] Vocational Training	8	[1] Uncensored	[1] Married	[-2] Does not apply
[137] 1994 May	[138] 1994 Jun	[4] Vocational Training	8	[1] Uncensored	[1] Married	[-2] Does not apply
[138] 1994 Jun	[139] 1994 Jul	[4] Vocational Training	8	[1] Uncensored	[1] Married	[-2] Does not apply
[146] 1995 Feb	[147] 1995 Mar	[4] Vocational Training	9	[1] Uncensored		[80] Education
[146] 1995 Feb	[147] 1995 Mar	[3] Part-Time Employment	10	[4] Left Censored		[80] Education
[147] 1995 Mar	[148] 1995 Apr	[4] Vocational Training	9	[1] Uncensored		[80] Education
[147] 1995 Mar	[148] 1995 Apr	[3] Part-Time Employment	10	[1] Uncensored		[80] Education
[148] 1995 Apr	[149] 1995 May	[4] Vocational Training	9	[1] Uncensored	[3] Single	[80] Education
[148] 1995 Apr	[149] 1995 May	[3] Part-Time Employment	10	[1] Uncensored	[3] Single	[80] Education
[149] 1995 May	[150] 1995 Jun	[4] Vocational Training	9	[1] Uncensored	[3] Single	[80] Education
[149] 1995 May	[150] 1995 Jun	[3] Part-Time Employment	10	[1] Uncensored	[3] Single	[80] Education

D: Collapsed spell data by enriching information from panel data

begin	end	status	spell nr	marital status	branch of industry	censor
[135] 1994 Mar	[138] 1994 Jun	[4] Vocational Training	8	[1] Married	[-2] Does not apply	[1] Uncensored
[146] 1995 Feb	[147] 1995 Mar	[4] Vocational Training	9		[80] Education	[1] Uncensored
[146] 1995 Feb	[147] 1995 Mar	[3] Part-Time Employment	10		[80] Education	[4] Left Censored
[148] 1995 Apr	[149] 1995 May	[4] Vocational Training	9	[3] Single	[80] Education	[1] Uncensored
[148] 1995 Apr	[149] 1995 May	[3] Part-Time Employment	10	[3] Single	[80] Education	[1] Uncensored

E: Generated panel data by enriching information from panel data

month of interview	year of inter-view	marital status	branch of industry	incidence_of_the_state_3	incidence_of_the_state_4	duration_of_the_state_3	duration_of_the_state_4
[3] March	1994	[1] Married	*[-2] Does not apply*				
[4] April	1995	[3] Single	[80] Education	1	1	2	6

Notes: The blue observations in Panel C mark the time points of the interviews whereas the orange observations mark the period of times for which information of the panel data was carried forward or backwards within a spell (cf. description of PANEL_TO_SPELL). For space reasons we abbreviated the names of some variables

Table 2: Example of combining two spell datasets

A: Original spell data (master file)

begin	end	status	spell number	censor
[36] 1985 Dec	[39] 1986 Mar	[3] Part-Time Employment	4	[4] left censored
[36] 1985 Dec	[39] 1986 Mar	[4] Vocational Training	2	[4] left censored
[40] 1986 Apr	[43] 1986 Jul	[1] Full-Time Employment	5	[5] right and left

B: Original spell data (using file)

begin	end	status	spell number	censor
[36] 1985 Dez	[40] 1986 Apr	[1] Single	1	[2] left censored
[41] 1986 Mai	[44] 1986 Aug	[2] Married	2	[1] left censored

C: Combined spell data in extended version

		master spell file			using spell file		
begin	end	status	spell nr	censor	status	spell nr	censor
[36] 1985 Dec	[37] 1986 Jan	[3] Part-Time	4	[4] left censored	[1] Single	1	[2] left censored
[37] 1986 Jan	[38] 1986 Feb	[3] Part-Time	4	[0] uncensored	[1] Single	1	[0] uncensored
[38] 1986 Feb	[39] 1986 Mar	[3] Part-Time	4	[0] uncensored	[1] Single	1	[0] uncensored
[39] 1986 Mar	[40] 1986 Apr	[3] Part-Time	4	[0] uncensored	[1] Single	1	[0] uncensored
[36] 1985 Dec	[37] 1986 Jan	[4] Vocational Training	2	[4] left censored	[1] Single	1	[0] uncensored
[37] 1986 Jan	[38] 1986 Feb	[4] Vocational Training	2	[0] uncensored	[1] Single	1	[0] uncensored
[38] 1986 Feb	[39] 1986 Mar	[4] Vocational Training	2	[0] uncensored	[1] Single	1	[0] uncensored
[39] 1986 Mar	[40] 1986 Apr	[4] Vocational Training	2	[0] uncensored	[1] Single	1	[0] uncensored
[40] 1986 Apr	[41] 1986 May	[1] Full-Time	5	[2] left censored	[1] Single	1	[0] uncensored
[41] 1986 May	[42] 1986 Jun	[1] Full-Time	5	[0] uncensored	[2] Married	2	[1] left censored
[42] 1986 Jun	[43] 1986 Jul	[1] Full-Time	5	[0] uncensored	[2] Married	2	[0] uncensored
[43] 1986 Jul	[44] 1986 Aug	[1] Full-Time	5	[2] right censored	[2] Married	2	[0] uncensored
[44] 1986 Aug	[45] 1986 Sep	.	.	.	[2] Married	2	[0] uncensored

D: Collapsed Spell data

begin	end	status of master file	spell nr of master	censor of master	status of using file	spell nr of using	censor of using
[36] 1985 Dec	[40] 1986 Apr	[3] Part-Time	4	[4] left censored	[1] Single	1	[2] left censored
[36] 1985 Dec	[40] 1986 Apr	[4] Vocational Training	2	[4] left censored	[1] Single	1	[0] uncensored
[40] 1986 Apr	[41] 1986 May	[1] Full-Time	5	[4] left censored	[1] Single	1	[0] uncensored
[41] 1986 May	[44] 1986 Aug	[1] Full-Time	5	[2] right censored	[2] Married	2	[1] left censored
[44] 1986 Aug	[45] 1986 Sep	.			[2] Married	2	[0] uncensored

Notes: The colors in Panel C and Panel D denote the separate spells of the resulting dataset. For space reasons we abbreviated the names of some variables

Table 3: Example of combining spell dataset with information on events.

A: Original spell data

begin	end	status	spell number	censor
[36] 1985 Dec	[40] 1986 Apr	[1] Single	1	[0] Uncensored
[41] 1986 May	[44] 1986 Aug	[2] Married	2	[5] Right censored

B: Event Data

month of the birth	year of birth	gender of first child
[2] February	1986	[1] Male

C: Spell data in extended version enriched with information on event (child birth of the first child)

		spell file			event file	
begin	end	status	spell number	censor	_merge	gender of first child
[36] 1985 Dec	[37] 1986 Jan	[1] Single	1		[1] master only	.
[37] 1986 Jan	[38] 1986 Feb	[1] Single	1		[1] master only	.
[38] 1986 Feb	[39] 1986 Mar	[1] Single	1		[3] matched	[1] Male
[39] 1986 Mar	[40] 1986 Apr	[1] Single	1		[1] master only	.
[36] 1985 Dec	[37] 1986 Jan	[1] Single	1		[1] master only	.
[37] 1986 Jan	[38] 1986 Feb	[1] Single	1		[1] master only	.
[38] 1986 Feb	[39] 1986 Mar	[1] Single	1		[1] master only	.
[39] 1986 Mar	[40] 1986 Apr	[1] Single	1		[1] master only	.
[40] 1986 Apr	[41] 1986 May	[1] Single	1		[1] master only	.
[41] 1986 May	[42] 1986 Jun	[2] Married	2		[1] master only	.
[42] 1986 Jun	[43] 1986 Jul	[2] Married	2		[1] master only	.
[43] 1986 Jul	[44] 1986 Aug	[2] Married	2		[1] master only	.
[44] 1986 Aug	[45] 1986 Sep	[2] Married	2		[1] master only	.

D: Spell data in compressed version

begin	end	status	spell number	censor	event	nr. of event	gender of first child
[36] 1985 Dec	[37] 1986 Jan	[1] Single	1	[0] Uncensored	0	0	.
[38] 1986 Feb	[40] 1986 Apr	[1] Single	1	[0] Uncensored	1	1	[1] Male
[41] 1986 May	[44] 1986 Aug	[2] Married	2	[5] RZ: Last Spell	0	1	.

Notes: The colors in Panel C and Panel D denote the separate spells of the resulting dataset. For space reasons we abbreviated the names of some variables

Table 4: Example of overlapping spells

A: Original spell data

begin	end	status	spell number	censor
[146] 1995 Feb	[149] 1995 May	[3] Part-Time Employment	10	[2] Right Censored
[147] 1995 Mar	[152] 1995 Aug	[4] Vocational Training	9	[5] Left And Right Censored

B: Combined spell data in extended version

		spelltype one			spelltype two		
begin	end	status	spell nr	censor	status	spell nr	censor
[146] 1995 Feb	[147] 1995 Mar	[3] Part-Time	10	[0] Uncensored			
[147] 1995 Mar	[148] 1995 Apr	[3] Part-Time	10	[0] Uncensored	[4] Voc. Training	9	[1] Left
[148] 1995 Apr	[149] 1995 May	[3] Part-Time	10	[0] Uncensored	[4] Voc. Training	9	[0] Uncensored
[149] 1995 May	[150] 1995 Jun	[3] Part-Time	10	[2] Right	[4] Voc. Training	9	[0] Uncensored
[150] 1995 Jun	[151] 1995 Jul				[4] Voc. Training	9	[0] Uncensored
[151] 1995 Jul	[152] 1995 Aug				[4] Voc. Training	9	[0] Uncensored
[152] 1995 Aug	[153] 1995 Sep				[4] Voc. Training	9	[2] Right

C: Split spell data with parallel subspells

begin	end	status	spell number	censor
[146] 1995 Feb	[146] 1995 Feb	[3] Part-Time Employment	10	[1] Uncensored
[147] 1995 Mar	[149] 1995 May	[4] Vocational Training	9	[4] Left Censored
[147] 1995 Mar	[149] 1995 May	[3] Part-Time Employment	10	[2] Right Censored
[150] 1995 Jun	[152] 1995 Aug	[4] Vocational Training	9	[2] Right Censored

Notes: The colors in Panel B and Panel C denote the separate spells of the resulting dataset. For space reasons we abbreviated the names of some variables

Appendix A

Description of the do-file CENSORING_PROGRAMS

As described above, censoring refers to the accuracy of the beginning (left-censoring) and ending (right-censoring) of a respective spell. Each SOEP spell dataset encompasses a censoring variable which encodes occurrences of left- and right-censoring into values (see for instance Table 5).

Table 5: Coding of the variable ZENSOR of the ARTKALEN dataset

Right: Left:	not censored	censored missing	censored before gap
not censored	1	2	3
censored missing	4	5	6
censored after gap	7	8	9

The basic intuition of the procedure of the CENSORING_PROGRAMS do-file is to decode these values and separately store the information on left censoring and right censoring into two new variables. This is achieved by the `censor_unfold` program which takes as an input the original censor-variable of the used SOEP dataset. It leaves you with a copy of the original censor-variable (`org_censor`) as well as with one variable `left` containing the information on left-censoring and one variable `right` containing the right-censoring.

Therefore, we can split spells and merge new information while the information on left censoring is limited only to the first subspell of an original spell and the information on right censoring applies only to the last subspell of an original spell. Take for instance the example of the employment spells depicted in Table 2. In the original data of Panel A the part-time employment and vocational training spells are both left-censored, the full-time employment spell is left- and right-censored. When these are split into subspells in Panel C, the original censoring does only apply to the first and respectively last subspell of each spelltyp. The intermediate subspells are all uncensored. Please note, that the example in Table 2 does however not represent that the censoring information is actually stored in two separate variables: one indicating left-censoring and one indicating right-censoring. Note furthermore, that the encoding of the censoring information differs across SOEP spelldata. The programs of the CENSORING_PROGRAMS do-file identify the respective coding employing the range of the value labels of the censor-variable. For instance, while the censor-variable of the ART-

KALEN dataset ranges from 1 to 9 (see Table 5), the range in the BIOMARSM dataset is from 0 to 14. If you use this program with a different encoding of the censoring information the program can not unfold it and will display an error message.

Finally, the `censor_contract` program merges the `left` and `right` variables together in one censor variable. Therefore, it takes as an input the copy of the original censor variable (`org_censor`) which was generated by the `censor_unfold` program and a name of the resulting censor-variable. It is also necessary that the `left` and `right` variables exist as they were generated before. If these do not exist or if the encoding of the censoring information is not identified the program will display an error message. Otherwise, it will merge the information on left- and right-censoring into one variable which is coded coherently with the variable you started with in the `censor_unfold` program.