

Erhardt, Klaudia

Research Report

How to generate spell data from data in "wide" format: Based on the migration biographies of the IAB-SOEP migration sample

DIW Data Documentation, No. 79

Provided in Cooperation with:

German Institute for Economic Research (DIW Berlin)

Suggested Citation: Erhardt, Klaudia (2015) : How to generate spell data from data in "wide" format: Based on the migration biographies of the IAB-SOEP migration sample, DIW Data Documentation, No. 79, Deutsches Institut für Wirtschaftsforschung (DIW), Berlin

This Version is available at:

<https://hdl.handle.net/10419/122163>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Data Documentation

How to Generate Spell Data from Data in "Wide" Format Based on the Migration Biographies of the IAB-SOEP Migration Sample

IMPRESSUM

© DIW Berlin, 2015

DIW Berlin

Deutsches Institut für Wirtschaftsforschung

Mohrenstr. 58

10117 Berlin

Tel. +49 (30) 897 89-0

Fax +49 (30) 897 89-200

www.diw.de

ISSN 1861-1532

All rights reserved.

Reproduction and distribution

in any form, also in parts,

requires the express written

permission of DIW Berlin.

Data Documentation 79

Klaudia Erhardt*

How to Generate Spell Data from Data in “Wide” Format

Based on the Migration Biographies of the IAB-SOEP Migration Sample (SOEP 2013, data set bdp_mig)

Including the Employed Stata Syntax

Last Update: 2014-11-24

* Klaudia Erhardt, SOEP, DIW Berlin, kerhardt@diw.de

Contents:

1	Introduction	3
2	Main working principles of a transformation from “wide” data into spell data	4
3	Determining the relation between source and target files	6
3.1	Structure of the source file	6
3.2	The structure of the target file	11
3.3	Detailed correspondence between source variables and target variables	12
4	How to do the transformation in Stata	17
4.1	Layout of the do-file migspell_transform.do	18
4.1.1	Sections 0 to 1: preliminary operations	18
4.1.2	Section 2: Generating spell data for “stays abroad”	19
4.1.3	Section 3: Generating spell data for “coming to Germany”	20
4.1.4	Section 4: Integration of the two spell data sets	21
4.2	Some specific procedures within the syntax	21
4.2.1	Invoking the variable names of the migration biographies within a loop structure	21
4.2.2	How to generate the target spell data file	23
	Appendix A: migspell_transform.do	24
	Appendix B: Questions of the migration biography	34
	German version	35
	Englisch translation	39

1 Introduction

This paper describes the generation of spell data from data in “wide” format on the basis of the migration biographies obtained from the integrated person-biography questionnaire administered to the IAB-SOEP Migration Sample (sample M), wave 30 (wave bd). The questions on respondents’ migration biography are numbers 16 to 33, with questions 16 to 27 being addressed to respondents born outside Germany and questions 28 to 33 to German-born respondents (see appendix). The original questionnaires are reproduced as SOEP Survey Paper No. 218 (see SOEP Survey Paper No. 219 for an English translation).

The transformation presented in this paper is done with Stata. The syntax is explained in detail and printed in Appendix A.

This paper not only describes the transformation of the migration biographies into spell format but also explains the working principles in a generalized way. Although the structure of the migration biography data is somewhat idiosyncratic and complex (discussed in detail in chapter 3), it can be deconstructed into a series of basic forms that increase in complexity. This paper can therefore serve as a manual for similar kinds of jobs.

In the strict sense of the word, spell data are about time periods with a defined start and end. The technique described here is not dependent on time or date variables. It can be used for any data transformation job that reshapes data from wide to long, and that is too complex for Stata’s “reshape” command, whether or not the data at hand are spell data. However, for data other than spell data, this kind of transformation is only viable and sensible if the sequence of variables in the pre-transformation form consists of repeating characteristics for multiple instances of the same kind of object. Examples of such a data structure with non-spell (non-time-related) data are: sets of characteristics for all major consumer goods in a household, for each training program or degree completed by an individual, or for each child in a family.¹

¹ A good alternative to reshaping the data at a post-production stage would be to organize them from the outset into “long” format. Thus, for individuals within a household—basically the same data structure as described here—it would be quite unusual to store the data in “wide” format. An advantage of the “long” data format is that the number of possible repetitions does not have to be predefined. Yet often the original data format is not determined by the data user but by the institution conducting the survey.

2 Main working principles of a transformation from “wide” data into spell data

An individual’s complete migration biography comprises a single record in the source file. Thus, moves between countries constitute a very long chain of variables. The transformation into a spell data structure results in a record for each move that a person made, so that every person who has moved in his or her life has more than one record in the target spell data set. The resulting target spell file will therefore have far more observations but far fewer variables than the original source file.

The basic method of generating the target spell data file is to split the original chain of variables from the source file into data segments consisting of several variables each, with each segment relating to one move. Each data segment, supplemented with the case ID variable from the source file, can thus be transformed into a record of the target spell data set. Figuratively, this is like chopping a very long log into similarly sized pieces of firewood and stacking them one on top of the other.

In order to do this, the following operations dealing with the relation between source and target file have to be carried out:

- Decide on the structure and the variables for the target file.
- Identify the variables in the source file that fit the variables of the target file for each spell.

More on these two steps will follow later. At this point, we will proceed as if we had already determined the data structure of the source and target file as well as the relations between the two and will continue describing the general method of doing the transformation. The steps are as follows:

1. Even if we had the simplest possible data structure in the source file, we would still have to deal with the problem of the variable names: The variables in the source file that capture the same characteristics are inevitably named differently for each repetition, whereas the target file uses the same variable names for a characteristic in every repetition. The variables therefore have to be renamed before adding a data segment from the source file to the target file as a new spell.
2. The different variables of the source file that flow into one variable of the target file might have different codings for the same semantic values, so this has to be checked and, if necessary, captured by appropriate recoding before adding the spell to the target file. With the migration biographies of the IAB-SOEP migration sample this is in fact necessary.

3. Preferably, “empty” spells should not be transferred to the target file. Empty spells, i.e., spells with code “not applicable” in every variable, result from respondents dropping out or from filtering. For example, with the migration biographies of data set `bdp_mig`, the different loop segments² are moves to Germany and moves to other countries, so respondents skip the one or the other segment. It is more convenient to check for empty spells before appending the new spell than to identify and delete them after the compilation of the target file. The condition that identifies empty spells depends on the data at hand. Within data file `bdp_mig`, “empty” values are coded -2 (does not apply). To identify empty spells, it suffices to check if the variable `starty` (year when the move took place) has the value -2 because if this is true, all the other variables in the data segment relating to the same move will also have the value -2.
4. The target file needs a new variable that does not stem from the source file to identify the spells within a case. It is better to create such a spell-ID as a consecutive number the moment the spells are generated instead of doing so after the target file has been compiled. In this way, the spell ID mirrors the succession of moves in the original data, including possible date errors which ought not to be corrected automatically because they then would probably stay unnoticed. In fact, this practice may cause gaps in the sequence of the generated spell IDs (due to the mentioned “empty spells”), but this does not create problems at this stage and can, if desired, be corrected easily after the target file has been compiled.

In principle, we have now outlined all the necessary steps for transferring “wide” data to long (spell) data. The Stata syntax file `migspell_transform.do` (see Appendix, p. 24ff.), which will be explained in detail in chapter 4, executes these steps.

Additionally, the syntax contains a data transformation that is usually unnecessary. It captures an idiosyncratic feature of the migration biographies of `bdp_mig`: the country variables do not contain Germany because of the specific design of the questions. Respondents were first asked if they had moved to Germany or to another country. This question was followed by “to which country?” if the move was to another country. Germany therefore has to be integrated into the country variables before generating the spells.

Before we deal with the Stata syntax, we have to determine the relationship between source file and target file. This is the subject of the next chapter.

² In the following, we use the term “loop” instead of “repetition”. In the questionnaire, if the respondent answers that she moved again after a first move, the sequence of questions is interrupted by jumping back to the first question about a move, thus forming a loop. We call one passage through the loop a “loop transit”.

3 Determining the relation between source and target files

This section returns to the two issues postponed above: deciding on the structure and the variables of the target file, and determining the correspondence between the variables of the source and the target file. To address these issues, we have to consider the information collected with the questionnaire and the variables in the source file. The questions we need to answer are: What information is to be transferred and which exact spell does it belong to? When does the migration history of a person begin and when does it end?

The following sections refer to the migration biographies of `bdp_mig.dta`, but the underlying ideas described here fit similar jobs as well.

3.1 *Structure of the source file*

The “migration biographies” are formed by the variables originating from questions 16 to 33 of the integrated individual biographical questionnaire for the IAB-SOEP Migration sample (sample M), wave 30 (wave `bd`).

The questions are repeated in a loop for each move from one country to another or from and to Germany. The data set contains a series of variables for each possible loop transit, the number of transits being limited to 15. To make things more complicated, moves to Germany and moves to another country result in different sets of variables per loop transit. A given respondent may have moved from Germany to another country, but might also have moved between other countries. Respondents might therefore have values in both sets of variables, and might have two moves within the same loop transit, if those were between Germany and another country, and between two other countries.

And finally, there are different sets of variables for German-born and non-German-born migrants. As these two groups are genuinely separate (no person can switch from one group to the other), the treatment of the data by the syntax file `migspell_transform.do` is divided into two parts: in a first step, the “stays abroad” data on the German-born migrants are transformed into spell data. Then in a second step, spell data is generated from the “coming to Germany” data on the migrants who were not born in Germany.

The principle of the transformation is roughly the same for both groups, but the source variables are different, and the structure of the “coming to Germany” migration biographies is a bit more complicated than the “stays abroad” migration biographies.

In the following, the flow charts for the two parts of the migration biography show the questions involved and the succession of the questions, which depends partly on the answers. As mentioned, the “stays abroad” section addresses the migrants born in Germany and is entirely independent of the “coming to Germany” section, which addresses migrants born outside Germany. No respondent can have valid information in both segments.

Figure 1 (see following page) shows the variables for the migration biographies of the German-born migrants, which by itself consists of two distinguishable parts, that is, two interlaced variable sequences, one for the stays abroad and the other for the stays in Germany. A respondent may have several consecutive loop transits in part 1 of the loop if he or she moved between different foreign countries, but can transit loop part 2 only once before switching back to loop part 1. Note also that moves have to take place between different countries; moves between different parts of the same country are not considered as a move³. With question 30 “When did you move to the other country?” the respondent enters the interlaced loop. Question 32 produces a bifurcation: either the respondent moved to another “other country” or returned home to Germany. If she stayed in Germany without any further moves abroad (question 33), she exits the loop permanently.

As a first step to determine the relationship between source and target file, we compile the kinds of information that come from the loops:

Table 1: List of information types from the “stays abroad” section of the migration biography

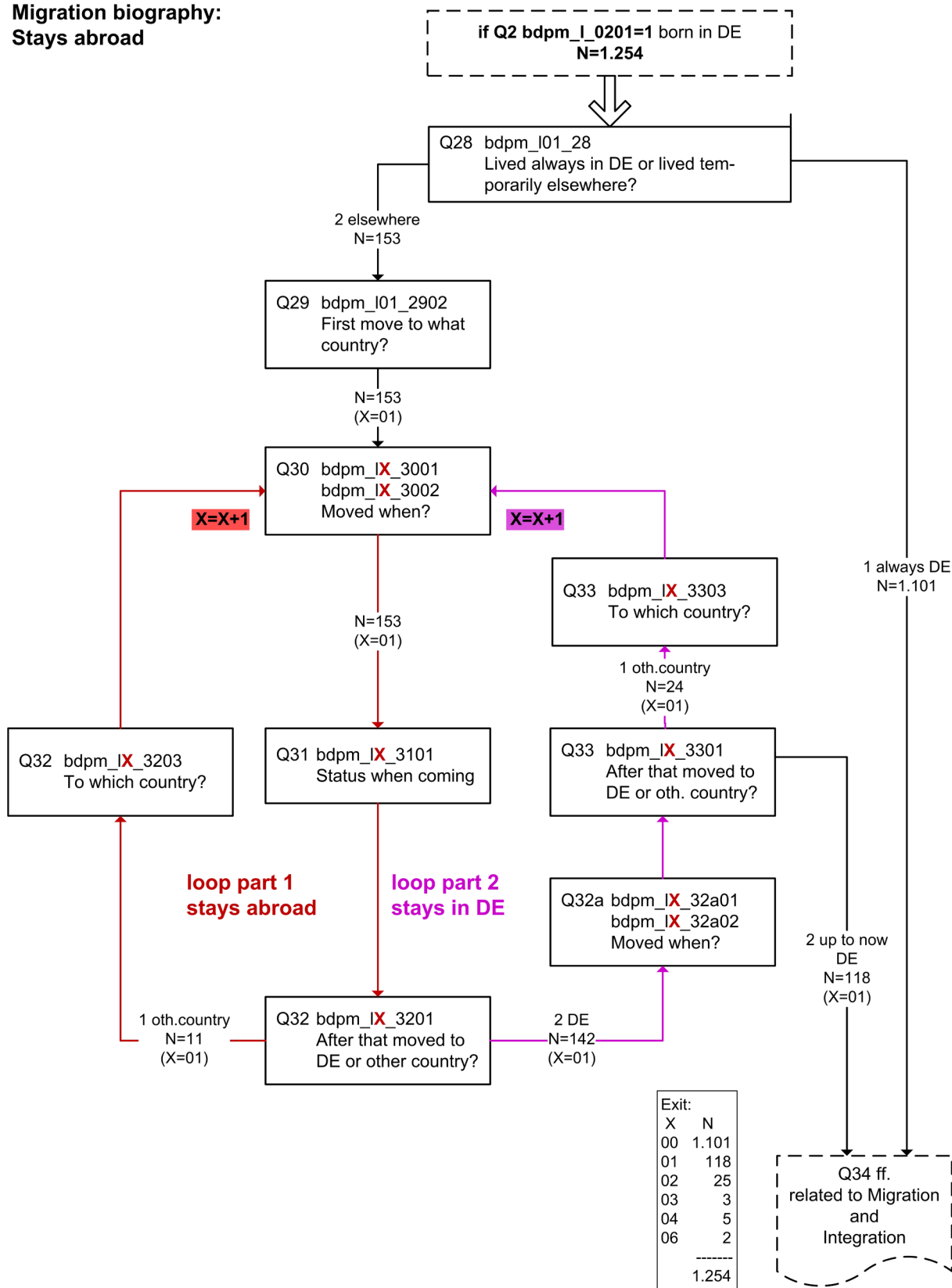
- | |
|--|
| <ul style="list-style-type: none"> • Kind of next move (to Germany or to another country: bifurcation between the two loop parts) • What country was the (next) move to? • When did the move take place? • Migration status at move to other country |
|--|

This list is fairly short. It shows that the two loop parts can easily be mapped onto each other, because both parts contain basically the same kind of information with exception of question 31 (Migration status at move to other country), which is only asked when the respondent moves to a foreign country.

³ In a few cases the interviewee reported moves to faraway regions or counties of the same country, such as from France to La Reunion, or from Russia to Siberia. As the SOEP country list has no codes for subdivisions of countries, this may look like a data error because it was coded as a move from a country to the same country—while as a rule, moves within the same country were not to be reported.

Figure 1: Flow chart of the „stays abroad“ section of the migration biographies

**Migration biography:
Stays abroad**

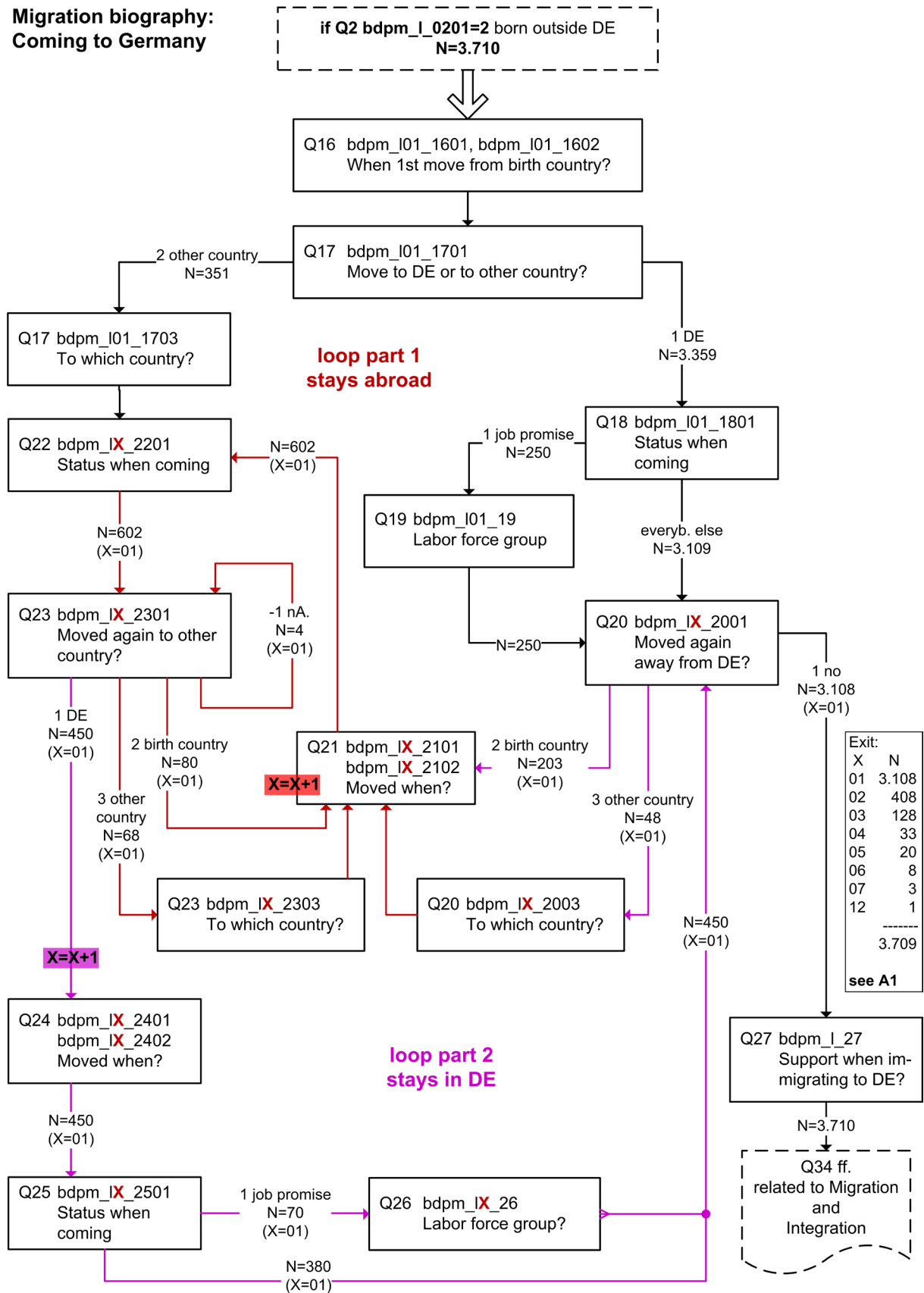


Exit:	
X	N
00	1.101
01	118
02	25
03	3
04	5
06	2

	1.254

Figure 2: Flow chart of the „coming to Germany“ section of the migration biographies

**Migration biography:
Coming to Germany**



Now we perform the same examination for the migration biographies of the migrants born outside Germany. Figure 2 shows the respective flow chart. At first sight it looks intimidatingly complex. But a closer look reveals the same two loop segments for “stays in Germany” and “stays abroad”. In this case, however, there are more variables, and, due to lack of space, it was not possible to place the two loop parts as neatly side by side as in Figure 1. We will see below that the complex structure can be broken down into a very similar list to that shown in Table 1.

We find two bifurcation points instead of one: question 17 and question 23, each leading either to loop part 1 or to loop part 2. But because question 17 is outside the loop, this does not really increase the level of difficulty.

In part 1 of the loop (“stays abroad”) we do not find any other type of information than that listed in Table 1. The flow chart here is slightly more complex than the previous flow chart because question 23 branches off in three directions instead of two. The possible answers to the question “Where did you move to” were: to Germany, to my birth country, to another country. This also only seems to be more complicated. In fact it is still the decision between Germany and another country. For the branches “birth country” and “other country” the same questions are posed except for “which country?”.

In loop part 2 “stays in Germany” we find a new kind of information: “labor force group” from question 19 and question 26 which were posed when the answer to the preceding questions 18 or 25 was “with job offer”. For the rest there are no surprises: all the information types are already known from the examination of the migration biographies of the non-German-born migrants (see Table 1).

The list of information types from the “coming to Germany” section of the migration biography can be determined as is listed in Table 2:

Table 2: List of information types from the “coming to Germany” section of the migration biography

- | |
|---|
| <ul style="list-style-type: none"> • Kind of next move (to Germany or to another country: bifurcation between the two loop parts) • What country was the (next) move to? • When did the move take place? • Migration status at move to other country • Labor force group |
|---|

In spite of the overwhelmingly crowded flow chart for the “coming to Germany” section of the migration biographies, the list of information types has only one more entry than the previous list and is still very short.

The real challenge for programming the transformation into spell data comes from the fact that several variables precede the loops containing the same information types as the variables within the loop. At this point it is not yet apparent why this is a problem, and we leave the issue aside for the moment.

The list in Table 1 is contained entirely in the list in Table 2. The list in Table 2 will therefore be used as the source to determine the variables in the target file.

3.2 *The structure of the target file*

From Table 2 the following variable list can be generated:

- **country:** country where the spell was spent
- **starty:** start date of the spell, year
- **startm:** start date of the spell, month

The migration biography data contain no time variable, but two separate variables for year and month of each move. For the moment we leave this as it is. Also we do not concern ourselves with the end dates at this point. Apart from date errors or missing data, each spell is connected to the next (because there is no alternative to the respondents being either in Germany or in some other country), and the end of a spell can be derived from the start date of the following spell or—for the last spell—from the interview date. This can be carried out after the compilation of the target file.

- **status:** status when coming to the respective country
- **lfgroup:** labor force group the respondent belonged to when coming to Germany as a working person
- **nmtyp:** type of the next move
- **tcountry:** target country of the next move

The latter two variables, *nmtyp* and *tcountry*, point to the **next** spell, but they belong to the **current** spell. Variable *nmtyp* contains the information type “kind of next move” and it produces the ramification between “stay in Germany” and “stay in another country”. It therefore has to precede the spell relating to one of the branches, that is, it belongs to the current spell and points to the next.⁴ As to *tcountry*, it is debatable whether this variable is really required, because its information (where was

⁴ Treating the information type “kind of next move” as a characteristic of the following spell would obscure its function as a bifurcation point and therefore make it more difficult to depict the underlying structure.

the move to) could just as well be transferred directly to the variable *country* in the next spell. It is essentially a question of taste. I found it more convincing to transfer the related information to a variable “target country of the next move” when generating the spell data and deduce the value of *country* from the target country of the preceding spell after the compilation of the spell file.

In addition to the aforementioned variables that are generated from the loops, the spell data file needs two identifiers:

- **persnr**: Case ID
- **spid**: Spell ID⁵

persnr and *spid* are added to each spell in the process of generating it. While *persnr* is contained in the source file and just has to be copied to each spell, *spid* is a newly generated variable. The two identifiers are not covered in the following two synopses however, in order to save space.

What remains to be addressed is the question “When does the migration history of a person begin and when does it end?” (see p. 6). It seems appropriate to let the migration biography begin at birth and end at the time of the interview. In consequence, no migration biography is left-censored except if the respondent did not report her birth year. So we insert an additional spell for the time between birth and first move (or the interview date if no move has taken place). The variable *country* then contains the birth country in the first spell of each respondent; *starty* and *startm* contain birth year and month. Allowing the migration biography to begin at birth entails the advantage that every respondent appears in the target spell data file with at least one record.

Up to now, the mapping of the source variables to the target variables was done somewhat sketchily. However, in order to write a concise and well structured program for the transformation of wide data into spells we have to set up a clear and unambiguous table of the correspondences. This will be provided in the next chapter.

3.3 Detailed correspondence between source variables and target variables

We need a detailed table of the source variables in each particular loop transit. In this context, we will come back to the aforementioned problem caused by several variables

⁵ Usually in the SOEP-data the spell ID variable is named *spellnr*. The variable *spid* generated here in the transformation process will have gaps and is replaced later by a spell ID variable named *spellnr* that shows the same consecution of spells as *spid* but has no gaps.

preceding the actual loop and containing the same information types as the variables within the loop (see p. 11).

The correspondence between the source variables by loop transit and the target variables is shown in the following two synopses “stays abroad” and “coming to Germany”.

The columns of the tables are formed by the above listed variables in the target file (see p. 11). The first column, named somewhat awkwardly “VARinfix” relates to the part of the variable names of *bdp_mig* that refers roughly to an index of the loop transits, and that is neither a prefix nor a suffix, since it is to be found in the middle of the variable name. In other words, it designates the varying part of the variable names. The last column, “coverage”, shows the conditions under which the variables listed in the same row of the table apply.

The lines of the synopses indicate the source variables in file *bdpm_mig* for each spell and each variable of the target file. Synopsis 1 shows, for example, that the value of variable *status* in the second spell in the target file stems from variable *bdpm_l01_3101* in the source file.

The first line of each synopsis, that is, the first spell, contains the aforementioned variables birth country, birth year and birth month, which are not part of the migration biographies. Thus we create an additional spell that does not correspond with a move but that represents the first episode of each migration history, the episode from birth to first move (or the interview date if no move occurred).

From here on, we no longer refer to “lines” of the synopses but to “spells”.

We see that the column “lfgroup” (i.e., the variable *lfgroup* in the target file) does not apply in the “stays abroad” section of the migration biographies. We could have omitted it from synopsis 1 but chose to maintain the column for reasons of symmetry and because we want to merge all spells from all biographies into one target file.

Following the columns downwards, we see that from a certain point on there is welcome regularity and repetition, which is what we need if we want to program loops for the transformation into spells instead of writing specific syntax for each particular move. From the point of repetition onwards (after spell 1 in synopsis 1 and spell 4 in synopsis 2) there are always two spells with the same VARinfix. These represent the two interlaced loop parts constituted by stays in Germany and stays abroad (remember the flow charts).

We can now understand the reason for including the variables *nmtyp*e and *tcountry* (which point to the next spell) in the current spell: their source variables bear the infix of the actual spell. Note that the column *country* is empty in both synopses except for

the first (artificial) spell. Because we decided on assigning the values of the “country” variables of *bdpm_mig* to *tcountry* to keep the regularity of the VARinfixes, *country* will be filled after the compilation of the target file from *tcountry* of the previous spell.

However, the regularity of the VARinfixes within a spell is broken in synopsis 2 with the variables highlighted in yellow. For the moment, we postpone the meaning of the yellow highlighting and the reasons for the irregularity and concentrate first on synopsis 1 which shows the migration biographies of the German-born migrants (“stays abroad”). Here we find an almost perfect regularity. If it were not for the interlacing of the two loop parts representing stays in Germany and stays abroad (i.e., if we had only every second spell), we would have a basic structure that is easy to transform into spell data. Even so, transformation is not very difficult because the interlaced loops only require a second transformation procedure of the same type. Although this may seem a bit cryptic at the moment, it will become clearer when we explain the transformation syntax of the do-file.

Keep in mind that the variables of the columns *nmtpe* and *tcountry* in the first spell of synopsis 1 have the same VARinfix “01” as the two following spells. These are the variables *bdpm_101_28* and *bdpm_101_2902* preceding the loop (see Figure 1 on page 8). Those variables bearing the infix “01” has no reason, because there are no variables *bdpm_102_28*, *bdpm_103_28*, and so on. In any case, as these two variables are not only outside the loop but also outside the naming conventions of the loop variables, they do not disturb the regularity.

The same is unfortunately not true for the section of the migration biography that applies to foreign-born migrants (“coming to Germany”), presented in synopsis 2. As in synopsis 1, the first spell represents the episode from birth to first move. But it lasts three more spells before regularity takes effect.

Assessing the variable correspondence of those three spells was a fairly difficult operation due to the apparently arbitrary assignment of variable names and VARinfixes. Having established the correspondence, the inner logic emerged. In hindsight, it was worth the trouble because the synopses in general, but especially synopsis 2, were a great help in developing the transformation syntax to produce spell data.

Synopsis 1: Stays abroad (for German-born migrants)

VARinfix	country	starty	startm	status	lfgroup	nmtyp	tcountry	coverage
01	bdpm_l_0203	bdpm_l_0103	bdpm_l_0102	---	---	bdpm_l01_28	bdpm_l01_2902	all cases

from here: repetition for each value of VARinfix

01		bdpm_l01_3001	bdpm_l01_3002	bdpm_l01_3101	---	bdpm_l01_3201	bdpm_l01_3203	stay abroad
01		bdpm_l01_32a01	bdpm_l01_32a02	---	---	bdpm_l01_3301	bdpm_l01_3303	stay in DE
02		bdpm_l02_3001	bdpm_l02_3002	bdpm_l02_3101	---	bdpm_l02_3201	bdpm_l02_3203	stay abroad
02		bdpm_l02_32a01	bdpm_l02_32a02	---	---	bdpm_l02_3301	bdpm_l02_3303	stay in DE

etc.

15		bdpm_l15_3001	bdpm_l15_3002	bdpm_l15_3101	---	bdpm_l15_3201	bdpm_l15_3203	stay abroad
15		bdpm_l15_32a01	bdpm_l15_32a02	---	---	bdpm_l15_3301	bdpm_l15_3303	stay in DE (up to int.date)

Synopsis 2: Coming to Germany (for migrants not born in Germany)

XX index minus 1 due to variables highlighted in yellow from loop transit 1

VARinfix	country	starty	startm	status	lfgroup	nmtype	tcountry	coverage
01	bdpm_l_0203	bdpm_l_0103	bdpm_l_0102	---	---	bdpm_l01_1701	bdpm_l01_1703	all cases
01		bdpm_l01_1601	bdpm_l01_1602	bdpm_l01_2201	---	bdpm_l01_2301	bdpm_l01_2303	stay abroad (birth country → not DE)
01		bdpm_l01_1601	bdpm_l01_1602	bdpm_l01_1801	bdpm_l01_19	bdpm_l01_2001	bdpm_l01_2003	stay in DE
01		bdpm_l01_2101	bdpm_l01_2102	bdpm_l01_2201	---	bdpm_l01_2301	bdpm_l01_2303	stay abroad (birth country → DE → not DE)

from here: repetition for each value of VARinfix

02		bdpm_l01_2401	bdpm_l01_2402	bdpm_l01_2501	bdpm_l01_26	bdpm_l02_2001	bdpm_l02_2003	stay in DE
02		bdpm_l02_2101	bdpm_l02_2102	bdpm_l02_2201	---	bdpm_l02_2301	bdpm_l02_2303	stay abroad
03		bdpm_l02_2401	bdpm_l02_2402	bdpm_l02_2501	bdpm_l02_26	bdpm_l03_2001	bdpm_l03_2003	stay in DE
03		bdpm_l03_2101	bdpm_l03_2102	bdpm_l03_2201	---	bdpm_l03_2301	bdpm_l03_2303	stay abroad

etc.

15		bdpm_l14_2401	bdpm_l14_2402	bdpm_l14_2501	bdpm_l14_26	bdpm_l15_2001	bdpm_l15_2003	stay in DE
15		bdpm_l15_2101	bdpm_l15_2102	bdpm_l15_2201	---	bdpm_l15_2301	bdpm_l15_2303	stay abroad
16		bdpm_l15_2401	bdpm_l15_2402	bdpm_l15_2501	bdpm_l15_26	---	---	stay in DE (up to int.date)

We do not go into further detail on the idiosyncratic variable names of the second to fourth spells in synopsis 2. Instead, we focus on the spells after the point where regularity emerges. In contrast to synopsis 1, the VARinfixes of synopsis 2 are not the same for all variables within a spell, but this lack of consistent correspondence applies only to the spells for stays in Germany. Here we find that *starty*, *startm*, *status*, and *lfgroup* have a lower VARinfix than the other variables in the same spell.⁶ Yet once this anomaly—highlighted in yellow in the synopsis—and the reason for it have been identified it is easy enough to address. The reason is apparent from the third row of the synopsis, where the four variables highlighted in yellow take the place of the variables that appear in later spells. When the latter appear for the first time, their infix part begins with “01”, although their “colleagues” in the same row have already reached “02”.

Since usually most of the variables are derived from a questionnaire that is designed to cover multiple aspects, with data handling requirements not taking top priority, similar problems might occur often enough when it comes to transforming data from wide to long format. In those cases a correspondence table like the one presented in the synopses is extremely useful.

4 How to do the transformation in Stata

As described above, we transform the data into spell format by splitting a long variable chain into similarly sized chunks that are stored one atop the next. This chapter describes precisely how this is done with Stata, using the example of the migration biographies in file `bdp_mig.dta`.

The two synopses presented in the previous chapter show exactly what pieces the long chain has to be split into. These can be used as a template for a line-by-line definition of each spell to be created. This is only done, however, for the spells before regular repetitions occur (after spell 1 in synopsis 1 and spell 4 in synopsis 2). As soon as we find regular patterns, we can shorten the process by using looping techniques.

Let us recall the four necessary steps for the treatment of every segment of the variable chain listed on page 4, now in the same order as they are performed in the program:

⁶ Seeing it this way is a matter of decision. One could just as well decide that variables *nmtree* and *tcountry* should have a higher VARmidfix than the other variables in the same spell. In that case the point of regularity was reached after spell 3 with spell 5 still having the value “01” in the VARmidfix column. The way we put it in this paper is more in accordance with the explanation of the reason for this anomaly.

1. If necessary, recode the variables in the source file that flow into the same target variable, so that same meanings have same the numeric codes.
2. Rename the variables in the source file to fit the variables in the target file.
3. Capture empty spells and prevent them from being added to the target file.
4. Generate the spell identifier.

So far, the steps are universal for the transformation of “wide” format data into spell format data. In addition, we had to address the idiosyncrasy that stays in Germany were not represented in the country variables but in extra variables.

The listed steps give the structure for the treatment of each particular spell. In the following, the syntax file `migspell_transform.do` is explained in detail. The entire Stata syntax file is to be found in appendix A of this paper. It might be useful to print it out in order to compare the following explanations to the related syntax.

The reader should have a general understanding of macros in Stata and of how loops are used in programming.

4.1 Layout of the do-file `migspell_transform.do`

The file `migspell_transform.do` has 5 sections, numbered from 0 to 4:

- Section 0: Definition of globals
- Section 1: Preparation of the source files for “coming to Germany” and for “stays abroad”
- Section 2: Generating spell data for “stays abroad” (migrants born in DE)
- Section 3: Generating spell data for “coming to Germany” (migrants not born in DE)
- Section 4: Integration of the two spell data sets, generation of some useful variables, generation of variable and value labels

4.1.1 Sections 0 to 1: preliminary operations

Global macros are defined only for the local paths and the log file. All other macros within the syntax are local macros⁷. In general, local macros should be given prefer-

⁷ Simply put, macros in Stata are a kind of text module.

ence whenever possible: “You should never use a global macro where a local macro would suffice” (Stata User’s Guide, Chapter 18.3.10, Advanced global macro manipulation). However, defining the local paths globally is more convenient when developing syntax because this makes it easier to run a job partially.

In section 1 the source files are prepared. In order to construct the first (additional) spell from birth to the first move, the variables birth country, birth year, birth month, and German-born/not German-born are combined with the variables that relate to the “stays abroad” or “coming to Germany” sections of the migration biographies. It was a matter of preference to create separate source files for each section, containing only the cases and the variables that relate to the respective part. Alternatively, we could have kept all variables in one source file and selected the appropriate data when using the file.

4.1.2 Section 2: Generating spell data for “stays abroad”

The general procedure is as follows: For each spell those variables from the source file are used that are listed in the respective line of the synopses. Then—as preliminary steps—the variables with differing semantic codes are recoded uniformly and Germany is integrated into the country variable. After that, the main tasks are: first, to rename the set of source variables to make them a set of target variables, then to generate the spell ID, drop the “empty” spells, and save the subfile separately. The subfile contains the xth spell of all persons having performed an xth move.

In section 2, “stays abroad”, only the first spell has to be generated “manually”. From then on regular repetitions begin, which are processed in a “forvalues loop”, the loop index ranging from one to 15, as we can see from synopsis 1.

Each loop transit consists of two interlaced parts (see synopsis 1): after the first spell there are always two spells having the same VARinfix. The syntax within the loop is therefore divided into the production of the first and the second spell of each loop. Each of these results in a separate file. The problems that had to be solved for the syntax within the loops were:

- a) invoking the variables without spelling out their specific names
- b) generating a spell ID
- c) generating an index number for the target subfiles.

For point a see section 4.2.1 (p. 21) of this chapter. We have to reconstruct the “infix” part of the variable name, and the procedure for doing so is explained there.

For points b and c the same consideration applies: We cannot use the loop index to enumerate the spells and spell files because every loop produces two of them. The algorithm is therefore: twice the loop index for the first and twice the loop index plus one for the second spell. The result is the value of the spell identifier and also the index of the name of the resulting “stays abroad” subfile.

Having thus produced all 31 spell files (the first one plus 15 times 2 from the 15 loop transits), they have to be combined into the target file of the “stays abroad” section of the migration biography. This is also done by means of a forvalues loop. After having opened the file of the first spell the loop index ranges from 2 to 31. In the same loop the now dispensable single spell files are erased.

4.1.3 Section 3: Generating spell data for “coming to Germany”

The general proceeding is as described in section 2. The main differences are:

- a) There are more spells until regularity is reached and loop programming begins.
- b) We have to address the issue that the source variable names from the same spell may have different infixes (see synopsis 2, infixes highlighted in yellow).
- c) The last spell is treated outside the loop.

Synopsis 2 shows that there are four spells before regular repetitions occur. Strictly speaking, there are only three spells, because two of them can only occur alternately: line 2 of the synopsis applies to persons whose first move was to another country than Germany, while line 3 applies to persons whose first move was to Germany. No respondent can have values in both spells that correspond to these lines of the synopsis. Therefore lines 2 and 3 form the template for the second spell and accordingly line 4 forms the template for the third spell.

There would have been nothing wrong with mapping those four lines of the synopsis onto four spells. To the contrary, the syntax would have been simpler. However, mapping them onto three spells results from the process of clarifying the complex data structure and is more in accordance with the semantics of the data.

After that, the processing continues with a forvalues loop, again done in basically the same way as in section 2: There are two interlacing loop parts, that is, two spells are generated from each loop transit. However, the different infixes in the spells relating to stays in Germany require a special treatment that was not necessary in section 2. As mentioned above, section 4.2.1 (p. 21) explains in detail how to invoke the variable names.

After the `forvalues` loop is finished, only the last spell has to be transformed—outside the loop. Due to the fact that some variables have a lower infix than their “colleagues” in the spells that relate to stays in Germany, the variable list for the last spell (which had to be a stay in Germany) is different. It contains only the yellow highlighted variables with a infix of the regular value minus 1 (*starty startm status lfgroup*, see last line of synopsis 2). Their infix is 15 in the last spell, while the regular (logical) infix value would have already reached 16 if treated within the loop—but there are no source variables with a infix of 16. For this reason, the last spell has to be treated outside the loop, in contrast to the approach in section 2.

The last step of section 3 is to compile the target file for the “way to Germany” section of the migration biography and to erase the now useless separate spell files.

4.1.4 Section 4: Integration of the two spell data sets

Section 4 deals with the integration of the two spell data sets. The variable *country* is derived from *tcountry* of the previous spell; a new consecutive spell ID is generated, that keeps the consecution but closes the gaps in the original spell ID; and some other useful procedures are carried out, such as defining variable and value labels. This section is quite transparent and requires no explanation.

4.2 Some specific procedures within the syntax

4.2.1 Invoking the variable names of the migration biographies within a loop structure

As emphasized, we do not want to write syntax for each spell to be added to the target file, but rather to abbreviate this procedure. Usually this is done by loops. It is crucial for programming loops that treat variables to find a way to invoke the variables without having to spell out their precise names. Variable names that are composed of both constant name parts and *regularly varying* name parts provide a good basis for this. One then has to find an algorithm to reproduce the varying parts and to put the variable name together from the constant part and the varying part.

The obvious easy solution to invoking the variables one after the other within a loop is for the varying part of their name to consist of a consecutive number that has a mathematical relation to the loop index (the *x*th loop transit). In every programming language, loops are defined by the number of loop transits (repetitions) until the program exits the loop. So if we had to treat *var1*, *var2*, *var3* and so on with the same sequence

of commands, we could do so easily using the index of the loop: in transit 1 treat var1, then go back (transit 2), treat var2, then go back (transit 3), treat var3, and so on.

In principle, the variables of the source files meet the requirement of having consecutive numbers as varying name parts (their “infixes”)—with the exception that numbers smaller than 10 result in infixes which consist of a consecutive number preceded by zero. This is not an integer and therefore cannot be replaced directly by the index that counts the loops. For this reason, the procedure takes the intermediate of a local macro to indicate the infixes. The content of the macro depends not only on the index number of a specific loop transit, but also on whether the index number is smaller than 10 / greater or equal to 10. If the index is smaller than 10, a leading zero has to be added. The macro used to reproduce the infix was named “in”.

The next small complication—which occurs only for the “way to Germany” data on foreign-born migrants—are the differing infixes in the variables of the same spell when related to a stay in Germany (see the yellow highlighted positions in synopsis 2), the infixes being one unit lower than those of their colleagues. To invoke those variables, a second macro is needed with an appropriate definition of the infix, named “iu”.

See below the related part of the syntax for the “coming to Germany” data:

```
local n = 15 /* because the infixes range to 15 maximally */
forvalues i = 2(1)`n' { /* from 2 to 15 in steps of 1 */

    /* Generation of the variable parts of the variable names: */
    local j = `i'-1 /* j is the lower infix, i is the regular infix
                    and at the same time the loop index */

    if `i' < 10 {
        local in 0`i'/* a leading 0 is added if i < 10 */
    }
    if `i' >= 10 {
        local in `i'
    }
    if `j' < 10 {
        local iu 0`j'/* a leading 0 is added if j < 10 */
    }
    if `j' >= 10 {
        local iu `j'
    }

    /* follows syntax: what to do with the variables having the infixes
    `in' and `iu' in their names */

}
```

Because the macros “in” and “iu” are linked with the loop index “i” in every loop transit, they take different values from 02 to 15 and from 01 to 14, respectively. Note that

we avoid the problem that “iu” takes on the value “00” by defining the loop index range from 2 to 15, not from 1 to 15.⁸

4.2.2 How to generate the target spell data file

As explained above, each move is represented by a segment of the long variable chain that makes up the whole migration biography. The program captures sequentially each segment and transforms it into spells. The spells belonging to one segment of the variable chain are saved in a separate data file that contains one observation (spell) for each person having done the xth move. At the end all separate data files are appended to form the target file.

The only problem to be solved with this approach is how to name the intermediate data files. Having already arranged for a consecutive spell number, it makes obvious sense to use this number as a suffix to the file name. Note that the spell number may have gaps for the spells of respondents (because of dropping empty spells), but it is consecutive without gaps for the intermediate spell files, although the files may have no observations.

This is important for the procedure of compiling the separate data files, again with a loop. If there were gaps in the suffixes of the file names, or if a file were not saved because it contains no observations, the loop through the files would generate an error as soon as a file is invoked that does not exist. But since this is not the case, no provisions have to be made to capture a possible error of invoking non-existing files.

The same loop that compiles the target file also erases the intermediate data files.

Note: Instead of creating and deleting the intermediate data files in the course of the syntax, they could have been created using the `tempfile` command. If you are programming Stata syntax that is to be used by others, this would be a better choice because it does not conflict with already existing files that have the same names as the files created by your syntax.

However, using “tempfile” implies that you cannot access the intermediate after the end of the program without having to carry out major syntax alterations. For example, you might want to check whether your program does what you want it to do by examining one of the intermediate files. Creating them as ordinary data files and erasing them when they are no longer needed permits access to the intermediate files by simply converting the “erase” command in the syntax into a comment.

⁸ Doing so, the variables with the midfixes “01” have to be treated before the loop starts.

Appendix A: migspell_transform.do

```
version 13
/* #####
##### Generation of spell data from migration biographies #####
##### IAB-SOEP migration sample 2013 (bdp_mig.dta) #####
##### Author: Klaudia Erhardt, SOEP, DIW #####
##### Last updated: 2014-11-12 #####
##### Source file: bdp_mig.dta #####
##### Resulting file: migspell_transformed.dta #####
##### */

/* contents:

SECTION 0: Definition of globals

SECTION 1: Preparation of the source files for "Coming to Germany" and for "Stays abroad"

SECTION 2: Generating spell data for "Stays abroad" (migrants born in DE)

SECTION 3: Generating spell data for "Coming to Germany" (migrants born outside DE)

SECTION 4: Integration of the two spell data sets, generation of some useful variables,
generation of variable and value labels
*/

/* #####
##### SECTION 0: Definition of Globals #####
##### Replace content of globals by values of your own environment #####
##### If you don't differentiate main directory and /data /output directories, #####
##### assign same directory to all globals or change related syntax #####
##### */

global data s:/DATA/soep30_de/stata /* Path to source-files directory */
global pfad1 H:/Zuwanderer/Datenverknüpfung_STATA/migspells /* Main local path */
global pfadld ${pfad1}/Datenfiles /* local data directory (--> target file) */
global pfadlo ${pfad1}/Output /* local output directory */
global lg ${pfadlo}/migspells_${S_DATE}.log /* local log-file name */
```

Data Documentation 79
Appendix A: migspell_transform.do

```
/* ##### Start of program ##### */

clear
set more off
set varabbrev off
capture log close
log using "$lg", append

/* ##### SECTION 1: Preparation of the source file ##### */

/* The source file contains:

- Variables that are not part of the migration biographies: country of birth place (current state),
  birth year, birth month, German-born/not German-born (only used as a selection criterion)

plus alternatively:

- the variables describing the "stays abroad" section of the migration biography (relating to migrants born
  in Germany) --> bdp_mig_sabr.dta

- the variables describing the "coming to Germany" section of the migration biography (relating to migrants
  born outside Germany) --> bdp_mig_ctde.dta

*/

use "${data}/bdp_mig", clear
keep persnr bdpm_1_0203 bdpm_1_0103 bdpm_1_0102 bdpm_1_0201 bdpm_101_28-bdpm_115_3303
drop if bdpm_1_0201==2 /* drop the abroad-born migrants */
save "${pfadld}/bdp_mig_sabr_1.dta", replace
use "${data}/bdp_mig", clear
keep persnr bdpm_1_0203 bdpm_1_0103 bdpm_1_0102 bdpm_1_0201 bdpm_101_1601-bdpm_115_26
drop if bdpm_1_0201==1 /*drop the german-born migrants */
save "${pfadld}/bdp_mig_ctde_1.dta", replace

/* ##### SECTION 2: Generating spell data for "Stays abroad" (migrants born in DE) #####
##### See synopsis 1 of the documentation for an overview of the structure #####
##### of source and target file ##### */
```

Data Documentation 79
Appendix A: migspell_transform.do

```
/* ##### 1st Spell (everybody): this is an additional spell from birth to first move ##### */
use "${pfadld}/bdp_mig_sabr.dta", clear
keep persnr bdpm_1_0203 bdpm_1_0103 bdpm_1_0102 bdpm_101_28 bdpm_101_2902

/* Harmonization of codes of nmtype
   different codes only in 1st spell: 1: always lived in Germany, 2: also lived in other countries.
   With all other spells: 1: move to other country, 2: move to DE --> recoded 1=3 2=1 at the end of SECTION 2 */
recode bdpm_101_28 (2=3)

/* generate spell data */
rename (bdpm_1_0203 bdpm_1_0103 bdpm_1_0102 bdpm_101_28 bdpm_101_2902) (country starty startm nmtype tcountry)
replace country=1 /* birth country resp. 1st country of habitation is DE for all cases */
gen byte spid = 1 /* generation of the spell ID */
drop if starty == -2 /* drop empty spells */
save "${pfadld}/sabr-1.dta", replace

/* ##### Loop for all the following moves ##### */

local n = 15
forvalues i = 1(1)`n' {
    /* Generation of the varying parts of the variable names */
    if `i' < 10 {
        local in 0`i'/* the leading zero is added */
    }
    if `i' >= 10 {
        local in `i'
    }
    /* ##### the 1st spell of each loop (a stay abroad) ##### */

    use "${pfadld}/bdp_mig_sabr.dta", clear
    keep persnr bdpm_1`in'_3001 bdpm_1`in'_3002 bdpm_1`in'_3101 bdpm_1`in'_3201 bdpm_1`in'_3203

    /* generate spell data */
    /* if target of next move is DE, replace value -2 of variable bdpm_1`in'_3203 (target country) with 1 */
    replace bdpm_1`in'_3203=1 if bdpm_1`in'_3201 == 2
    rename (bdpm_1`in'_3001 bdpm_1`in'_3002 bdpm_1`in'_3101 bdpm_1`in'_3201 bdpm_1`in'_3203) ///
           (starty startm status nmtype tcountry)
    gen byte spid = 2 * `i' /* generate spell ID */
    drop if starty == -2 /* drop empty spells */
    local xt = 2 * `i'
    save "${pfadld}/sabr-`xt'.dta", replace
}
```

```

/* ##### the 2nd spell of each loop (a stay in DE) ##### */

use "${pfadld}/bdp_mig_sabr.dta", clear
keep persnr bdpm_l`in' _32a01 bdpm_l`in' _32a02 bdpm_l`in' _3301 bdpm_l`in' _3303

* generate spell data
rename (bdpm_l`in' _32a01 bdpm_l`in' _32a02 bdpm_l`in' _3301 bdpm_l`in' _3303) (starty startm nmtype tcountry)
gen byte spid = 2 * `i' + 1 /* generate spell ID */
local xt = 2 * `i' + 1
drop if starty == -2 /* drop empty spells */
save "${pfadld}/sabr-`xt'.dta", replace
}
/* ##### compile the single files into target file ##### */

use "${pfadld}/sabr-1.dta", clear
forvalues i = 2(1)31 {
    quietly append using "${pfadld}/sabr-`i'", nolabel
    quietly erase "${pfadld}/sabr-`i'.dta"
}
quietly erase "${pfadld}/sabr-1.dta"

/* Harmonization of the codes of 'nmtype', which are different for "Stays abroad" and "Coming to Germany" */
recode nmtype (1=3) (2=1)

/* Harmonization of the codes of 'status', which are different for "Stays abroad" and "Coming to Germany" */
recode status (2=3) (3=5) (4=6) (5=7)

sort persnr spid
order persnr spid country starty startm status nmtype tcountry, first
save "${pfadld}/migspell_sabr_all.dta", replace

/* #####
##### SECTION 3: Generating spell data for "Coming to Germany" (migrants not born in DE) #####
##### See synopsis 2 of the documentation for an overview of the structure #####
##### of source and target file #####
##### */

/* ##### Reconstruction of the first loop ##### */

/* ##### 1st Spell (everybody): this is an additional spell from birth to first move ##### */

```

Data Documentation 79
Appendix A: migspell_transform.do

```
use "${pfadld}/bdp_mig_ctde.dta", clear
keep persnr bdpm_1_0203 bdpm_1_0103 bdpm_1_0102 bdpm_101_1701 bdpm_101_1703

/* target country: integrate DE */
replace bdpm_101_1703 = 1 if bdpm_101_1701==1

/* Harmonization of codes (different codes only in 1st spell) */
replace bdpm_101_1701 = 3 if bdpm_101_1701==2

/* generate spell data */
rename (bdpm_1_0203 bdpm_1_0103 bdpm_1_0102 bdpm_101_1701 bdpm_101_1703) (country starty startm nmtype tcountry)
gen byte spid = 1
drop if starty == -2 /* drop empty spells */
save "${pfadld}/ctde-1.dta", replace

/* ##### 2nd Spell - only persons moving from birth country to another country ##### */

use "${pfadld}/bdp_mig_ctde.dta", clear
keep bdpm_101_1701 persnr bdpm_101_1601 bdpm_101_1602 bdpm_101_2201 bdpm_101_2301 bdpm_101_2303
drop if bdpm_101_1701==1 /* drop persons going to Germany */
drop bdpm_101_1701

/* target country: integrate DE */
replace bdpm_101_2303 = 1 if bdpm_101_2301==1

/* harmonization of codes */
recode bdpm_101_2201 (6=7) (5=6) (4=5) (3=4) (2=3)

/* generate spell data */
rename (bdpm_101_1601 bdpm_101_1602 bdpm_101_2201 bdpm_101_2301 bdpm_101_2303) ///
(starty startm status nmtype tcountry)
gen byte spid = 2 /* generate spell ID */
drop if starty == -2 /* drop empty spells */
save "${pfadld}/ctde-2a.dta", replace

/* ##### 2nd Spell - only persons moving from birth country directly to Germany ##### */

use "${pfadld}/bdp_mig_ctde.dta", clear
keep bdpm_101_1701 persnr bdpm_101_1601 bdpm_101_1602 bdpm_101_1801 bdpm_101_19 bdpm_101_2001 bdpm_101_2003
drop if bdpm_101_1701==2 /* drop persons moving to another country than Germany */
drop bdpm_101_1701

/* target country: integrate DE */
replace bdpm_101_2003 = 1 if bdpm_101_2001==1
```

Data Documentation 79
Appendix A: migspell_transform.do

```
/* generate spell data */
rename (bdpm_101_1601 bdpm_101_1602 bdpm_101_1801 bdpm_101_19 bdpm_101_2001 bdpm_101_2003) ///
      (starty startm status lfgroup nmtype tcountry)
gen byte spid = 2
drop if starty == -2 /* drop empty spells */
append using "${pfadld}/ctde-2a.dta"
save "${pfadld}/ctde-2.dta", replace
capture erase "${pfadld}/ctde-2a.dta"

/* ##### 3rd spell - only persons moving to DE first, then to another country ##### */

use "${pfadld}/bdp_mig_ctde.dta", clear
keep bdpm_101_1701 bdpm_101_2001 persnr bdpm_101_2101 bdpm_101_2102 bdpm_101_2201 bdpm_101_2301 bdpm_101_2303
keep if bdpm_101_1701==1 & bdpm_101_2001 > 1 /* keep persons moving to DE first, then to another country */
drop bdpm_101_1701 bdpm_101_2001

/* target country: integrate DE */
replace bdpm_101_2303 = 1 if bdpm_101_2301==1

/* harmonization of codes which are different for moves to DE and moves to other countries */
recode bdpm_101_2201 (6=7) (5=6) (4=5) (3=4) (2=3)

/* generate spell data */
rename (bdpm_101_2101 bdpm_101_2102 bdpm_101_2201 bdpm_101_2301 bdpm_101_2303) ///
      (starty startm status nmtype tcountry)
gen byte spid = 3 /* generate spell ID */
drop if starty == -2 /* drop empty spells */
save "${pfadld}/ctde-3.dta", replace

/* ##### Loop for the following moves (except for the last move to DE) ##### */

local n = 15
forvalues i = 2(1)`n' {
    /* Generation of the varying parts of the variable names */
    local j = `i'-1
    if `i' < 10 {
        local in 0`i'/* the leading zero is added */
    }
    if `i' >= 10 {
        local in `i'
    }
    if `j' < 10 {
        local iu 0`j'/* the leading zero is added */
    }
}
```



```
    }
    if `j' >= 10 {
        local iu `j'
    }

    /* ##### the 1st spell of a loop (a stay in DE) ##### */

    use "${pfadld}/bdp_mig_ctde.dta", clear
    keep persnr bdpm_l`iu'_2401 bdpm_l`iu'_2402 bdpm_l`iu'_2501 bdpm_l`iu'_26 bdpm_l`in'_2001 bdpm_l`in'_2003

    /* target country: integrate DE */
    replace bdpm_l`in'_2003 = 1 if bdpm_l`in'_2001==1

    /* generate spell data */
    rename (bdpm_l`iu'_2401 bdpm_l`iu'_2402 bdpm_l`iu'_2501 bdpm_l`iu'_26 bdpm_l`in'_2001 bdpm_l`in'_2003) ///
        (starty startm status lfgroup nmtype tcountry)
    gen byte spid = 2 * `i' /* generate spell ID */
    local xt = 2 * `i' /* sequential number as a file identifier */
    drop if starty == -2 /* drop empty spells */
    save "${pfadld}/ctde-`xt'.dta", replace

    /* ##### the 2nd spell of a loop (a stay abroad) ##### */

    use "${pfadld}/bdp_mig_ctde.dta", clear
    keep persnr bdpm_l`in'_2101 bdpm_l`in'_2102 bdpm_l`in'_2301 bdpm_l`in'_2303 bdpm_l`in'_2201

    /* target country: integrate DE */
    replace bdpm_l`in'_2303 = 1 if bdpm_l`in'_2301==1

    /* harmonization of codes */
    recode bdpm_l`in'_2201 (6=7) (5=6) (4=5) (3=4) (2=3)

    /* generate spell data */
    rename (bdpm_l`in'_2101 bdpm_l`in'_2102 bdpm_l`in'_2201 bdpm_l`in'_2301 bdpm_l`in'_2303) ///
        (starty startm status nmtype tcountry)
    gen byte spid = 2 * `i' + 1 /* generate spell ID */
    drop if starty == -2 /* drop empty spells */
    local xt = 2 * `i' + 1
    save "${pfadld}/ctde-`xt'.dta", replace
}
```

Data Documentation 79
Appendix A: migspell_transform.do

```
/* ##### Last move to DE is treated outside the loop, because the variable list is different - now there
are only the variables left which had the infix "iu". iu could maximally be 14 inside the loop. In the next
step these variables having the infix 15 are invoked ##### */

local iu 15 /* set macros to 15 and 16, respectively, in order to allow for using the same syntax as within
the loop */

local i 16
use "${pfadld}/bdp_mig_ctde.dta", clear
keep persnr bdpm_l`iu'_2401 bdpm_l`iu'_2402 bdpm_l`iu'_2501 bdpm_l`iu'_26

/* generate spell data - tcountry und nmttype do not occur any more because there are no next moves */
rename (bdpm_l`in'_2401 bdpm_l`in'_2402 bdpm_l`in'_2501 bdpm_l`in'_26) (starty startm status lfgroup)
gen byte spid = 2 * `i' /* generate spell ID */
local xt = 2 * `i'
drop if starty == -2 /* drop empty spells */
save "${pfadld}/ctde-`xt'.dta", replace

/* ##### compile the single files into target file ##### */

use "${pfadld}/ctde-1.dta", clear
forvalues i = 2(1)32 {
    quietly append using "${pfadld}/ctde-`i'", nolabel
    quietly erase "${pfadld}/ctde-`i'.dta"
}
quietly erase "${pfadld}/ctde-1.dta"

sort persnr spid
order persnr spid country starty startm status lfgroup nmttype tcountry, first
save "${pfadld}/migspell_ctde_all.dta", replace

/* #####
##### SECTION 4: Integration of the two spell data sets, generation of some useful variables #####
##### and generation of variable and value labels #####
##### */

use "${pfadld}/migspell_ctde_all.dta", clear
append using "${pfadld}/migspell_sabr_all.dta", nolabel
sort persnr spid

/* target country --> -1 if nmttype== -1 (No answer) */
replace tcountry= -1 if nmttype== -1

/* replace tcountry with birth country if nmttype==2 (meaning: going back to birth country) */
by persnr: replace tcountry = country[1] if nmttype==2
```

```
/* fill in variable country from tcountry of previous spell, replace values of tcountry & nmtype in last spell */
by persnr: replace country = tcountry[_n-1] if _n > 1
by persnr: replace tcountry = -2 if _n==_N /* in the last spell target country of next move is not applicable */
by persnr: replace nmtype = -2 if _n==_N /* in the last spell next move type is not applicable */

/* create Variable birth country */
by persnr: gen int bcountry = country[1]

/* replace missings with -2 (not applicable) - missings stem from merge-command */
replace status = -2 if status == .
replace lfgroup = -2 if lfgroup == .

/* new consecutive spell ID (closing the gaps, but keeping the original succession) */
sort persnr spid
by persnr: gen byte spellnr = _n
drop spid

/* Number of spells */
by persnr: egen nspell = max(spellnr)

/* Define variable labels */
lab var persnr ""
lab var spellnr "spell number"
lab var nspell "number of spells per case"
lab var starty "startyear"
lab var startm "startmonth"
lab var country "country of habitation"
lab var status "status when coming"
lab var lfgroup "labor force type"
lab var tcountry "target country next move"
lab var nmtype "next move type"
lab var bcountry "birth country"

/* Define Value labels */
lab def nmtype ///
  -3 "[-3] Answer implausible" ///
  -2 "[-2] Does not apply" ///
  -1 "[-1] No answer" ///
  1 "[1] To Germany" ///
  2 "[2] Back to birth country" ///
  3 "[3] To another country" ///
  , replace
lab val nmtype nmtype
```

Data Documentation 79
Appendix A: migspell_transform.do

```
lab def status          ///
-3 "[-3] Answer implausible" ///
-2 "[-2] Does not apply"    ///
-1 "[-1] No answer"        ///
 1 "[1] Employee with job offer" ///
 2 "[2] Ethnic German immigrant fr. Eastern Europe"    ///
 3 "[3] Spouse/child/other family member"              ///
 4 "[4] Asylum seeker or refugee"                    ///
 5 "[5] Student/vocational trainee"                    ///
 6 "[6] Job seeker"                                    ///
 7 "[7] Other status"                                  ///
, replace
lab val status status

lab def lfgroup        ///
-3 "[-3] Answer implausible" ///
-2 "[-2] Does not apply"    ///
-1 "[-1] No answer"        ///
 1 "[1] Self-employed/entrepreneur"    ///
 2 "[2] Seasonal worker/contractor"    ///
 3 "[3] Relocated to Germany by employer"    ///
 4 "[4] Sent to Germany from company"    ///
 5 "[5] High qualified with special entry conditions"    ///
 6 "[6] Other kind of employee"        ///
, replace
lab val lfgroup lfgroup

label copy bdpm_1_0203 country
label copy bdpm_1_0203 tcountry
label copy bdpm_1_0203 bcountry
lab val country country
lab val tcountry tcountry
lab val bcountry bcountry

order persnr bcountry spellnr nspell country starty startm status lfgroup nmtpe tcountry, first

save "${pfadld}/migspell_transformed.dta", replace
quietly erase "${pfadld}/migspell_ctde_all.dta"
quietly erase "${pfadld}/migspell_sabr_all.dta"
capture log close
```

Appendix B: Questions of the migration biography

extract from the questionnaires for the IAB-SOEP Migration sample (sample M) of wave 30 (wave bd) of the SOEP (SOEP Survey Paper 218)

German version

Migrationsbiographie: Ihr Weg nach Deutschland

Bitte folgen Sie für Ihre Angaben der Filterführung im CAPI!

16. Viele Menschen lassen sich im Laufe Ihres Lebens in mehreren Ländern nieder. Wie war das bei Ihnen? Uns interessiert dabei, in welchen Ländern Sie für mehr als drei Monate gelebt haben. Zunächst möchten wir gerne wissen, wann Sie das erste Mal aus Ihrem Geburtsland weggezogen sind? Wenn Sie es nicht mehr exakt sagen können, geben Sie bitte einen Schätzwert an.

Gemeint ist hier eine durchgehende Aufenthaltsdauer in einem anderen Land von mehr als 3 Monaten. Kürzere Aufenthalte, z.B. Urlaubsreisen oder Besuche bei Verwandten, sind hier nicht gemeint.

Jahr				Monat		

17. In welches Land sind Sie da gezogen?

In ein anderes Land..... → und zwar: → Frage 22!
 Nach Deutschland..... ↓

18. Es gibt verschiedene Wege in ein anderes Land zu ziehen. Auf welchem Weg sind Sie nach Deutschland zugezogen?

Als Erwerbstätiger, der bereits eine Jobzusage hatte → Frage 19!
 Als Aussiedler, d.h. deutschstämmige Person aus osteuropäischen Staaten.....
 Als Ehegatte, Kind oder anderer Familienangehöriger
 Als Asylbewerber oder Flüchtling
 Als Student, Schüler oder Auszubildender → Frage 20!
 Als Arbeitssuchender.....
 Auf einem anderen Weg.....
 und
 zwar:

19. Als Sie nach Deutschland als Erwerbstätiger zugezogen sind: Zu welcher Gruppe haben Sie zu diesem Zeitpunkt gehört?

Selbständige und Unternehmer.....
 Saisonarbeiter und Werkvertragsarbeitsnehmer.....
 Ich wurde innerhalb meines Unternehmens nach Deutschland versetzt
 Ich wurde von meinem Unternehmen im Heimatland für eine Tätigkeit nach Deutschland entsendet
 Hochqualifizierte Arbeitnehmer, Wissenschaftler und andere Spezialisten mit erleichterten Einreisebedingungen.....
 Andere Arbeitnehmer

20. Sind Sie danach nochmals aus Deutschland weggezogen?

Gemeint ist hier eine durchgehende Aufenthaltsdauer in einem anderen Land von mehr als 3 Monaten. Kürzere Aufenthalte, z.B. Urlaubsreisen oder Besuche bei Verwandten, sind hier nicht gemeint.

Nein, bin bis heute in Deutschland geblieben → Frage 27!

Ja, zurück ins Geburtsland

Ja, in ein anderes Land und zwar:



21. Wann sind Sie in Ihr Geburtsland zurückgekehrt bzw. in das andere Land gezogen?

Wenn Sie es nicht mehr exakt sagen können, geben Sie bitte einen Schätzwert an.

Jahr				Monat		

22. Und auf welchem Weg sind Sie in das andere Land gezogen?

- Als Erwerbstätiger, der bereits eine Jobzusage hatte
 - Als Ehegatte, Kind oder anderer Familienangehöriger
 - Als Asylbewerber oder Flüchtling
 - Als Student, Schüler oder Auszubildender
 - Als Arbeitssuchender
 - Auf einem anderen Weg
- und zwar:

23. Sind Sie danach erneut in ein anderes Land gezogen oder direkt nach Deutschland?

Gemeint ist hier eine durchgehende Aufenthaltsdauer von mehr als 3 Monaten. Kürzere Aufenthalte, z.B. Urlaubsreisen oder Besuche bei Verwandten, sind hier nicht gemeint.

Ja, in ein anderes Land und zwar: → Frage 21!

Ja, zurück ins Geburtsland

Nein, direkt nach Deutschland



24. Wann sind Sie da nach Deutschland gezogen?

Wenn Sie es nicht mehr exakt sagen können, geben Sie bitte einen Schätzwert an.

Jahr				Monat		

25. Auf welchem Weg sind Sie nach Deutschland zugezogen?

Als Erwerbstätiger, der bereits eine Jobzusage in Deutschland hatte → Frage 26!

Als Aussiedler, d.h. deutschstämmige Person aus osteuropäischen Staaten.....

Als Ehegatte, Kind oder anderer Familienangehöriger

Als Asylbewerber, Vertriebener oder Flüchtling

Als Student, Schüler oder Auszubildender → Frage 20!

Als Arbeitssuchender.....

Auf einem anderen Weg.....

und
zwar:

**26. Als Sie nach Deutschland als Erwerbstätiger zugezogen sind:
Zu welcher Gruppe haben Sie zu diesem Zeitpunkt gehört?**

Selbständige und Unternehmer.....

Saisonarbeiter und Werkvertragsarbeitsnehmer.....

Ich wurde innerhalb meines Unternehmens nach Deutschland versetzt

Ich wurde von meinem Unternehmen im Heimatland für
eine Tätigkeit nach Deutschland entsendet → Frage 20!

Hochqualifizierte Arbeitnehmer, Wissenschaftler und andere Spezialisten
mit erleichterten Einreisebedingungen.....

Andere Arbeitnehmer

**27. Hatten Sie bei Ihrem Zuzug nach Deutschland Unterstützung von Verwandten oder
Bekanntem, die bereits in Deutschland lebten?**

☞ Falls Sie im Laufe Ihres Lebens mehrmals nach Deutschland zugezogen sind, beziehen Sie
sich bitte auf den letzten Zuzug.

Ja, Verwandte.....

Ja, Bekannte

Ja, beides

Nein.....

Sie springen auf Frage 34!

Migrationsbiographie: Ihre Auslandsaufenthalte																																
Bitte folgen Sie für Ihre Angaben der Filterführung im CAPI!																																
<p>28. Viele Menschen lassen sich im Laufe Ihres Lebens in mehreren Ländern nieder. Wie ist das bei Ihnen? Haben Sie immer in Deutschland gelebt oder haben Sie zeitweise auch woanders gelebt?</p> <p><i>☞ Gemeint ist hier eine durchgehende Aufenthaltsdauer in einem anderen Land von mehr als 3 Monaten. Kürzere Aufenthalte, z.B. Urlaubsreisen oder Besuche bei Verwandten, sind hier nicht gemeint.</i></p> <p>Ja, habe immer in Deutschland gelebt..... <input type="checkbox"/> ➔ Frage 34!</p> <p>Nein, habe auch woanders gelebt..... <input type="checkbox"/></p> <p>29. In welches Land sind Sie zunächst gezogen?</p> <p><i>☞ Falls Sie öfter als einmal durchgehend mindestens 3 Monate in einem anderen Land gelebt haben, beginnen Sie bitte mit dem Land, in das Sie zuerst gezogen sind.</i></p> <div style="border: 1px solid black; height: 20px; width: 100%;"></div> <p>30. Wann sind Sie da in das andere Land gezogen?</p> <p><i>☞ Wenn Sie es nicht mehr exakt sagen können, geben Sie bitte einen Schätzwert an.</i></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> </tr> <tr> <td colspan="4" style="text-align: center;">Jahr</td> <td colspan="4" style="text-align: center;">Monat</td> </tr> </table> <p>31. Es gibt verschiedene Wege in ein anderes Land zu ziehen. Auf welchem Weg sind Sie in das andere Land zugezogen?</p> <p>Als Erwerbstätiger, der bereits eine Jobzusage hatte <input type="checkbox"/></p> <p>Als Ehegatte, Kind oder anderer Familienangehöriger <input type="checkbox"/></p> <p>Als Student, Schüler oder Auszubildender <input type="checkbox"/></p> <p>Als Arbeitssuchender..... <input type="checkbox"/></p> <p>Auf einem anderen Weg..... <input type="checkbox"/></p> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">und zwar:</div> <p>32. Sind Sie danach erneut in ein anderes Land gezogen oder zurück nach Deutschland?</p> <p><i>☞ Gemeint ist hier eine durchgehende Aufenthaltsdauer von mehr als 3 Monaten. Kürzere Aufenthalte, z.B. Urlaubsreisen oder Besuche bei Verwandten, sind hier nicht gemeint.</i></p> <p>In ein anderes Land..... <input type="checkbox"/> ➔ Frage 30!</p> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">und zwar:</div> <p>Direkt nach Deutschland <input type="checkbox"/></p> <p>32a Wann sind Sie nach Deutschland gezogen?</p> <p><i>☞ Wenn Sie es nicht mehr exakt sagen können, geben Sie bitte einen Schätzwert an.</i></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> </tr> <tr> <td colspan="4" style="text-align: center;">Jahr</td> <td colspan="4" style="text-align: center;">Monat</td> </tr> </table> <p>33. Wie war das bei Ihnen nachdem Sie einige Zeit in Deutschland gelebt haben: Sind Sie danach nochmals aus Deutschland weggezogen?</p> <p><i>☞ Gemeint ist hier eine durchgehende Aufenthaltsdauer in einem anderen Land von mehr als 3 Monaten. Kürzere Aufenthalte, z.B. Urlaubsreisen oder Besuche bei Verwandten, sind hier nicht gemeint.</i></p> <p>Ja, in ein anderes Land..... <input type="checkbox"/> ➔ Frage 30!</p> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">und zwar:</div> <p>Nein, bin bis heute in Deutschland geblieben <input type="checkbox"/></p>									Jahr				Monat												Jahr				Monat			
Jahr				Monat																												
Jahr				Monat																												

Englisch translation

(just used as a translation utility, not being used as a survey questionnaire)

How You Came to Germany

Bitte folgen Sie für Ihre Angaben der Filterführung im CAPI!

16. Many people live in several different countries over the course of their lives. What about you? We're interested in finding out which countries you have lived in for more than three months. First of all, when did you first move away from the country where you were born? If you can't remember exactly, please give an estimate.

What we are referring to here are stays in another country lasting more than three months. We are not referring to shorter stays such as vacations or visits to relatives.

Year				Month		

17. Which country did you move to?

To another country..... ⇨ please state: ⇨ Q. 22!

To Germany..... ⇩

18. There are different ways of moving to another country. How did you move to Germany?

As an employed person who already had a job offer ⇨ Q. 19!

As an ethnic German (*Aussiedler*) from an Eastern European country

As a spouse, child or other family member

As an asylum-seeker or refugee

As a student or vocational trainee ⇨ Q. 20!

As a job-seeker

In a different way

Please state:

19. When you moved to Germany as a job-seeker, what group did you belong to?

Self-employed / entrepreneur

Seasonal laborer / contractor

I was relocated to Germany by my employer

I was sent to Germany by my employer in my home country to carry out a task

Highly skilled worker / scientist / other specialist subject to special conditions for facilitated entry

Other employee

20. Did you move away from Germany again after that?

We are referring here to a continuous stay in another country lasting more than three months. We are not referring to shorter stays such as vacations or visits to relatives.

No, I have remained in Germany since then → Question 27!

Yes, back to the country where I was born.....

Yes, to another country..... please state:

21. When did you move back to your home country or to the other country?

If you don't remember exactly, please estimate!

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Year				Month	

22. How did you move to the other country?

- As an employed person who already had a job offer
- As a spouse, child, or other family member
- As an asylum-seeker or refugee
- As a student or vocational trainee
- As a job-seeker
- In a different way

Please state:

23. Did you move to another country after that or directly to Germany?

We are referring here to a continuous stay in another country lasting more than three months. We are not referring to shorter stays such as vacations or visits to relatives.

Yes, to another country.....

Please state: → Question 21!

Yes, back to the country where I was born.....

No, directly to Germany.....

24. When did you move to Germany?

If you don't remember exactly, please estimate!

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Year				Month	

25. How did you move to Germany?

As an employed person who already had a job offer in Germany..... ⇒ Q. 26!

As an ethnic German (Aussiedler) from an Eastern European country

As a spouse, child, or other family member

As an asylum-seeker or refugee

As a student or vocational trainee → Q. 20!

As a job-seeker

In a different way

Please state:

26. When you moved to Germany with a job, which group did you belong to?

Self-employed / entrepreneur

Seasonal laborer / contractor

I was relocated to Germany by my employer

I was sent to Germany by my employer in my home country to carry out a task → Question 20!

Highly skilled worker / scientist / other specialist subject to special conditions for facilitated entry

Other employee

27. When you moved to Germany, did you have the help of any relatives or friends who already lived in Germany?

☞ If you have moved to Germany more than once in your life, please refer to your most recent move.

Yes, relatives

Yes, friends

Yes, both

No

Skip now to Question 34!

Your Stays Abroad

Bitte folgen Sie für Ihre Angaben der Filterführung im CAPI!

**28. Many people live in different countries over the course of their lives. What about you?
 We're interested in finding out which countries you have lived in for more than three months.**
 ☞ *We are referring here to a continuous stay in another country lasting more than three months.
 We are not referring to shorter stays such as vacations or visits to relatives.*

Yes, I have always lived in Germany..... ➔ **Question 34!**
 No, I have also lived in another country

29. What country did you move to first?
 ☞ *If you have lived in another country more than once for a continuous period of more than three months, start with the country that you moved to first.*

30. When did you move to the other country?
 ☞ *If you can't remember exactly, please estimate.*

Year					Month	

**31. There are different ways of moving to another country.
 How did you move to another country?**

As an employed person who already had a job offer
 As a spouse, child, or other family member
 As a student or vocational trainee
 As a job-seeker
 In a different way

Please state:

32. Did you move to another country after that or directly to Germany?
 ☞ *We are referring here to a continuous stay in another country lasting more than three months.
 We are not referring to shorter stays such as vacations or visits to relatives.*

To another country..... ➔ **Question 30!**

Please state:

Directly to Germany.....

**33. What about after you had lived in Germany for a while:
 Did you move away again?**
 ☞ *We are referring here to a continuous stay in another country lasting more than three months.
 We are not referring to shorter stays such as vacations or visits to relatives.*

Yes, to another country..... ➔ **Question 30!**

und zwar:

No, I have stayed in Germany ever since