Stata SHP Data Management (Stata 13 or 14)

Ursina Kuhn and Oliver Lipps, FORS, Lausanne
Version August 2016

Content

Aim of this documentation

This documentation looks at tasks or problems which are specific to analyse panel data in general, and data from the Swiss Household Panel (SHP) in particular using Stata. Some familiarity with opening and saving data, basic data manipulation and working with do files is assumed. Most sections contain exercises which are similar to the Stata syntax examples.

1 Missing values in the SHP

Missing data in the SHP have the following values

-1	don't know
-2	no answer
-3	not available (question has not been asked)
-4 to -9	variable specific missing values

In Stata, (system) missing values are defined as "." (a period). Negative values are treated as any other value (Stata e.g. treats an income of -3 as a true income of -3 Sfr.).[1] In analyses of the SHP data, the missing values have to be taken care of appropriately (this is a very frequent source of error). There are different strategies you might want to use.
A first strategy is to recode the negative values into system missing values (.). E.g. for yearly net income from employment in 2008, the Stata syntax is

```
recode i08empyn -8/-1=.
```

or

```
mvdecode i08empyn, mv(-8/-1)
```

Such a recode is of course a loss of information (which may or may not be relevant to you for further analysis). In the example of income from employment, we don't know any more whether an individual has missing income from employment because he or she did not participate in that year (-3), does not work (-4), does not want to report his or her income (-2), does not know the income (-1) or whether values were implausible (-8). To keep this information we can create a new variable i08empynr and leave the original variable i08empyn unchanged (strategy 2).

```
recode i08empyn -8/-1=., gen(i08empynr)
```

A third strategy is to designate separate missing values for each negative code. In the syntax example below, this is done for all variables together

```
recode _all(-1=.a)(-2=.b)(-3=.c)(-4=.d)(-5=.e)(-6=.f)(-7=.g)(-8=.h)
```

Finally, we may leave the values unchanged, but restrict the analysis to positive values when analysing the data.

[1] This is an important difference to SPSS, where specific values (such as negative values) can be declared as missing. In the SPSS version of the SHP, this has already been done for all negative values.

```
sum i08empyn if i08empyn > 0
```

!!! Regardless of how missing values are defined and which strategy you adopt, it is important to know that missing values in Stata are defined as the largest possible numbers. Furthermore, "." < ".a" < ".b" etc.
Here is an example to demonstrate the importance of taking account of missing values. With the syntax

```
count if i08empyn > 500000
```

Stata counts not only individuals who earn more than 500'000 Sfr. per year, but also individuals with a missing income from employment. If instead, we used the syntax

```
count if i08empyn > 500000 & i08empyn<.
```

we get the correct value.
To account for the different missing values (., `.a, .b, .c etc)`, write `<.` instead of `!=.` (not equal missing). Writing `!=.`, would include all values with `.a, .b, .c` etc., because their values are not equal to " .".

2 Merge files

2.1 Illustration of data merge

Suppose you want to use variables from two datasets in the same statistical model. For example, we are interested in the satisfaction with life in 2007 and 2008, which are included in different files. The (horizontal) combination of two data sets (or parts of two datasets) into a new data set is called a merge and illustrated in Figure 1. Data set 1 and data set 2 of Figure 1 contain information on the satisfaction with life in 2008 and 2007, respectively (variable p$$c44 of the SHP). There are three hypothetical individuals (idpers 43, 44 and 45) who are in both data sets, and two individuals who are only in one of the data sets (idpers 42 and 46). The variable idpers serves as an identifier variable and must be present in both data sets. The identifier enables correctly combining values for observations that are in both data sets. In the merged data set, individuals with idpers 43, 44 and 45 have values on their satisfaction with life in both 2008 and 2007.

idpers	p08c44
42	8
43	7
44	9
45	10

Data set 1

idpers	p07c44
43	7
44	8
45	6
46	10

Data set 2

idpers	p08c44	p07c44
42	8	.
43	7	7
44	9	8
45	10	6
46	.	10

Merged data set

Figure 1: Illustration of a 1:1 merge

In this example (of Figure 1), the variable idpers is a unique identifier, meaning that within each data set, every observation (individual) has a different idpers. This implies that each idpers in data set 2 is merged with the *same* idpers in data set 1. In Stata, this is called a one-to-one (1:1) merge. Another type of merge is presented in Figure 2. Here we want to add the household income to the observations in data set 1. Individuals 42 and 43 live in the same household (idhous08 101) and individuals 44 and 45 live in the same household (idhous08 102). Here, the identification variable is the variable idhous08, which is in both files. In data set 1, idhous08 is not a unique identifier, because there are some observations which have the same identifier. In data set 2, the variable idhous08 uniquely identifies the observations. In Stata, this is called a many-to-one ("m:1") merge, because several observations in data set 1 are merged to one observation in data set 2.

idpers	idhous08	p08c44
42	101	8
43	101	7
44	102	9
45	102	10

Data set 1

idhous08	i08htyn
101	50000
102	80000

Data set 2

idpers	idhous08	p08c44	i08htyn
42	101	8	50000
43	101	7	50000
44	102	9	80000
45	102	10	80000

Merged data set

Figure 2: Illustration of a m:1 merge

2.2 Identifiers in the SHP

When merging two data sets, we usually use an identifier variable. Again, identifier(s) need to be present in both files to be merged.[2] The different data sets of the SHP each contain identifier variables which allow combining these data sets. The different identifier variables in the SHP data files are listed in Table 1.

[2] If no identifier is indicated in the merge command, Stata combines the first lines of both data sets, the second lines of both data sets and so on.

	File name	identifiers
Individual master file	shp_mp	idpers, idhous$$, idfath__, idmoth__
Individual annual files	shp$$_p_user	idpers, idint, idhous$$, idspou$$, refpers$$
Other individual files (Social origin, last job, calendar, biographic)	shp_so, shp_lj shp_ca, shp0_*[3]	idpers
Interviewer data	shp$$_v_user	idint
Household annual files[4]	shp$$_h_user	idhous$$, refpers$$, canton$$, (no_ofs)[5]
CNEF files	shpequiv_$$$$	x11101ll (=idpers)

Note: the characters $$ are placeholders for indications of the panel year (e.g. 99 for the first wave in 1999, 00 for the second wave in 2000).

Table 1 : Files and identifier variables in the SHP

2.3 The Merge command in Stata

The merge command

To merge two files, we need a data set to be open in Stata. This data set is called the master data set. With the `merge` command, we tell Stata which other file (the using file) we want to merge.[6] Additionally, we indicate which variable(s) serve as identifier, so that the variables can be added correctly.

After a merge, the new data set contains observations from any of the original data sets (the master and the using data sets). Stata automatically creates the variable _merge, which indicates the origin of each observation with respect to the master and the using data set. The variable _merge has the following values[7]:

_merge=1 obs. from master data only (data set open before the merge)
_merge=2 obs. from the using dataset only (data set added with the merge command)
_merge=3 obs. from both master and using dataset

Table 2 presents the merged data set of Figure 1 including the _merge variable created by Stata. The value 1 of the _merge variable indicates individuals of which we have information from the 2008 data set only (the master data). The value 2 indicates cases which are in the

[3] The asterisk (*) serves as a placeholder for any combination of characters. The files of the biographic interviews all start with shp0_ (e.g. SHP0_BVLA_USER, SHP0_BLSA_USER, SHP0_BH_USER). The file SHP0_BH_USER is in the wide format (or horizontal format), all others are in the long data format (or vertical format) format.

[4] There is also a household master file (shp_mh), which contains the original household number given in the first wave (1999 or 2004). This file contains all households in the panel.

[5] The municipality identifiers of the Swiss Federal Statistical office (no_ofs) are not in the SHP user files for reasons of data confidentiality. If you want to add municipality information to the SHP data, please send us an email.

[6] The syntax presented here work with Stata versions 11 and above. The syntax for the merge command has changed considerably with version 11 of Stata. The examples presented below do not work with Stata10 and previous versions.

[7] There are in addition the codes 4 and 5 which only apply if the options update and/or replace are used

2007 data set only (the using data). The value 3 indicates individuals surveyed in both 2007 and 2008.

idpers	p08c44	p07c44	_merge
42	8	.	1
46	.	10	2
44	9	8	3
45	10	6	3
43	7	7	3

Table 2 : Example of the _merge variable in Stata

The _merge variable is useful to select cases for the analysis. In the example of Table 2, we delete cases where _merge equals 2 if we only want individuals who participated in 2007. Or if we want only individuals who were surveyed both in 2007 and in 2008 in the example of, we keep only cases where _merge is equal to 3.
Note that the _merge variable has to be deleted (or renamed) before an additional merge is possible. If not, Stata returns an error message.[8]
There are various options that can be used with the merge command (see examples below for some of them or type `help merge` in the Stata command window).

2.4 Merging two files of the SHP

Stata Example 1: Merging two annual files: personal annual files from 2008 and 2000
First, we open the 2008 individual file (the master file). Second, we add the 2000 individual file (this is the using file). Our identifier variable is idpers.[9]

```
use shp08_p_user, clear // the master file
merge 1:1 idpers using shp00_p_user // the using file
tab _merge
```

Note that in this example, we add all variables from the 2000 individual file to the 2008 individual file, which creates a large data set. See example 2 for the use of the keepusing option. Stata automatically prints the frequencies of the _merge variable.

[8] It may be useful to type `capture drop _merge` before a merge command to prevent the error message stopping the program. But be sure that you have selected the observations of interest previously. Alternatively the option nogenerate prevents Stata from producing the variable _merge.
[9] Stata automatically sorts the variable according to the identifier variable. In Stata versions previous to Stata 11, data must be manually sorted before or within the merge command.

```
Result                           # of obs.
-------------------------------------------
not matched                          2,625
    from master                      1,256  (_merge==1)
    from using                       1,369  (_merge==2)

matched                              9,633  (_merge==3)
```

9'633 individuals are both in the 2008 individual file and in the 2000 individual file. For 1'256 individuals, there is data from 2008 only (which was our master file). For 1'369 individuals, there is data from 2000 only. It depends on our research design or research question, which individuals we want to keep in our data set and which we delete. If, for instance, we are interested in how satisfaction with life of the same individual changed between 2000 and 2008, we keep only individuals with values in both years (`keep if _merge==3` or use the option `keep(match)` within the `merge` command). But if, for instance, we want to compare the group of individuals who stayed in the panel with individuals who dropped out of the panel in order to analyse attrition, we keep observations with values 2 and 3 of the _merge variable.
Additionally, we may want to delete individuals who neither in 2000 nor in 2008 answered the individual questionnaire which is indicated by the status$$ variable. Finally, we delete the _merge variable (in order to be able to do other merges later).

```
drop if status00!=0 & status08!=0  /* do this only if you don't
     need information from proxy and grid interviews !! */
drop _merge
```

Take time to select your observations according to the _merge variable, because it is crucial to know which observations are in your data set!

Stata Example 2: Merging an annual individual file with the individual master file
In this example, we want to add the variables `idmoth__` and `idfath__` to the 2008 individual annual file. The identifiers for the mother and the father are not in the annual file, but in the individual master file of the SHP (because they do not vary over time).

```
use shp08_p_user, clear // master file
merge 1:1 idpers using shp_mp, keepusing(idmoth__ idfath__)
```

With the option keepusing after the merge command, we specify which variables from the file "shp_mp" (the using file) we want to add to the 2008 individual file (the master file). With this option we add only the variables idmoth__ and idfath__ to our data set. Without the keepusing option, we would add all variables from the file "shp_mp".
Stata prints the frequencies of the _merge variable:

```
Result                           # of obs.
-------------------------------------------
not matched                         11,111
    from master                          0  (_merge==1)
    from using                      11,111  (_merge==2)

matched                             10,889  (_merge==3)
```

10'889 observations (individuals) are in both the individual Master File of the SHP (shp_mp) and the 2008 individual file (shp08_p_user). 11'111 observations are in the individual master

file but not in the 2008 individual file. These are individuals who were in the panel in any year other than 2008. They have missing values in all variables other than idpers and idmoth__ and idfath__. Note that there are no observations where the _merge variable equals 1. This is correct, since – by definition – the individual master file of the SHP (shp_mp) contains information on all individuals ever in the panel.
If we only want to work with individuals present in 2008, we delete observations which are only in the using file.

```
drop if _merge==2 // if we only want ind. from 2008
drop _merge // we don't need this variable any more
```

You can avoid doing these steps with the options "keep (match master)" and "nogenerate". With the "keep" option you can select the observations. With the "nogenerate" option, the _merge variable is not generated.

```
use shp08_p_user, clear // master file
merge 1:1 idpers using shp_mp, keepusing(idmoth__ idfath__) keep(match
master) nogen
```

Stata Example 3: Merging an annual individual file with the household file
To merge the 2008 individual file with the 2008 household file, we use idhous08 as the identifier. The identifier idhous08 does not uniquely determine observations in the master file, because several individuals may live in the same household and therefore have the same idhous08. This is why we specify a many-to-1 (m:1) merge.[10]

```
use shp08_p_user, clear // master file
merge m:1 idhous08 using shp08_h_user // using file
```

Stata prints the frequencies of the _merge variable.

```
    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                            10,889  (_merge==3)
```

All individuals have corresponding variables at the household level. We see that it was possible to add the household variable to all 10'889 individuals.
We delete (or rename) the _merge variable, if we plan additional merges.

```
drop _merge
```

2.5 Merging many files of the SHP

In a panel, we often want information on all (or more than two) panel waves (because this is what panels are about). In order to avoid typing the merge command many times, it is convenient to use a loop. For repetitive procedures, working with loops not only saves time when you write the syntax (at least once you are familiar with loops), but particularly if you need to change the syntax later. If, for instance, you want to add a variable to your data set (keepusing option), you only have to make one change in the code instead of repeating the change in each wave separately.

[10] If the h file is the master, and the p file the using file, we would have to specify a 1:m (one to many) merge.

Stata Example 4: Merge all individual files from 1999 to 2008
In this example, we are interested in the political left-right positions (variable p$$p10 in the SHP) of respondents to all waves between 1999 and 2008. We also add the status$$ variable of each year¨

```
use idpers p99p10 status99 using shp99_p_user if status99!=0,clear
foreach y in 00 01 02 03 04 05 06 07 08 {
  merge 1:1 idpers using shp`y'_p_user, keepusing(p??p10 status??)
  keep if _merge==3 & status`y'==0
  drop _merge
}
```

The "??" are used as placeholders, where each ? stands for any single character. After each merge, stata prints the frequencies of the _merge variable. We select individuals who answered all individual questionnaires from 1999-2008.
The keepusing option in the merge command allows the selection of variables we want to include into our data file. Without this option, we would end up with a huge data file with all the variables from all the data files added in the loop. Depending on the version of Stata, there may not be enough space for all the variables (merging the individual user files from the first 10 years of the SHP, for instance, results in a data set with about 3'700 variables). Additionally, Stata runs more slowly with a huge data set and makes it much more difficult to keep an overview. It is therefore advisable to select the variables needed for the analysis with the keepusing option in the merge command. However, the keepusing option works only when the variables are present in all the using files of the loop. While some variables of the SHP are present in each annual file, others are only available in some years due to changes in the questionnaire.

2.6 Exercises: merge files

Exercise 1: Life satisfaction and gender
(Exercise to merge individual annual file and individual master file)

1. Open the 2008 individual user file of the SHP
   How many individuals have completed the individual questionnaire?
2. Select respondents who completed the individual questionnaire
3. Add the variable sex (from the individual master file). Look at the _merge variable and select the cases needed to analyse life satisfaction. Delete the _merge variable.
4. Descriptive analysis of life satisfaction (variable p08c44)
   What is the median and mean life satisfaction in 2008? What is its standard deviation?
   How many individuals have missing values for satisfaction with life?
   Is there a difference in life satisfaction between men and women?

Exercise 2: Life satisfaction and household size
(Exercise to merge individual annual file and annual household file)

1. Open the 2008 individual user file of the SHP
2. Select respondents who completed the individual questionnaire
3. Add the variable nbpers08 (from the household file). Interpret the _merge variable and select the cases you need to analyse life satisfaction.
4. Descriptive analysis of life satisfaction (variable p08c44)

Compare individuals in single households, two person households and larger households. Is there a difference in life satisfaction between these groups?

Exercise 3: Correlation of life satisfaction across time
(Exercise to merge several individual annual files)

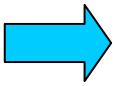
1. Open the 2008 individual user file of the SHP
2. Add the variable for life satisfaction (p$$c44) of all previous waves (where possible)
3. Keep individuals who responded in at least one year.
   How many cases do we have in the data set?
4. Descriptive analysis of life satisfaction (variable p08c44)
   How does life satisfaction correlate across time?
5. Keep the variables idpers, status$$ and p$$c44 in the data set and save the data (for example, name the file "satisfaction_allwaves.dta")

3 Add information from other household members

The fact that the SHP interviews all individuals within a household opens many possibilities to analyse household contexts, such as looking at influences between couples or influences between parents and children. For such analyses, we first have to create a dataset which adds variables from other (currently present or former) household members.
In the SHP, partners, fathers, and mothers are identified by the variables idspou$$, idfath__ and idmoth__, respectively.[11] An exemplary partner file is illustrated in Figure 3. The data set on the left is (part of) the 2007 individual file. The partner data set on the right contains not only the satisfaction with life of the individual in 2007 (variable p07c44), but also the satisfaction with life of the partner (variable p07c44_p). Such data sets can be created with the merge command.

idpers	idspou07	p07c44
42	43	8
43	42	7
44	45	9
45	44	-2
46	-3	10

Base file: shp07_p_user



idpers	idspou07	p07c44	p07c44_p
42	43	8	7
43	42	7	8
44	45	9	-2
45	44	-2	9
46	-3	10	.

Partner file (after merge)

Figure 3: Illustration of a partner file

[11] Partners, fathers and mothers are only identified if they lived together with the individual analysed at least once in the course of the SHP and are listed in the grid questionnaire. Other relationships are specified through their relation with the reference person (identifier refpers$$). Since the father and the mother of an individual cannot change over time, specifying a year is not necessary.

3.1 Create a partner file with the merge command

To create the partner file, several steps are needed, which are illustrated in Figure 4.

1. We open our data set (the base file in Figure 3 and Figure 4) with the necessary variables.
2. We make the following changes in the base and save it as a temporary file, using another name:
    a. We delete the variable idspou07.
    b. We rename the variable idpers to (a new) idspou07.
    c. We rename the variables we want to add to the individual file (e.g. with a suffix _p to denote the partner).
    d. We save this as a temporary file.
3. We merge the base file with the temporary file (saved in 2.), using idspou07 as the identifier.

idpers	idspou07	p07c44
42	43	8
43	42	7
44	45	9
45	44	-2
46	-3	10

Base file

steps 2. a., b., c., d.

idspou07 (former idpers)	p07c44_p (former p07c44)
42	8
43	7
44	9
45	-2

Temporary file (changed base file)

Step 3.

idpers	idspou07	p07c44	p07c44_p
42	43	8	7
43	42	7	8
44	45	9	-2
45	44	-2	9

Partner file

Figure 4: Creation of a partner file by merging with a temporary file

Stata Example 5: Creating a partner file

This is the Stata syntax used to construct the partner file in Figure 3 and Figure 4. First, we create the temporary file.

```
use idpers idspou07 p07c44 using shp07_p_user if idspou07>0, clear
// open individ. File; only people with a partner
drop idspou07 // otherwise the next step would not work
rename idpers idspou07 // rename idpers to idspou07
rename p07c44 p07c44_p // rename variable we want to add
save temp_partner07, replace
```

Second, we merge the individual file with this temporary file. Although normally, the idspou07 is a unique identifier, there may be some cases where it is not (if two individuals report to be the partner of the same other individuals). That's why we may have to use a m:1 merge.

```
use idpers idspou07 p07c44 using shp07_p_user, clear
merge m:1 idspou07 using temp_partner07
```

Look at the frequencies of the _merge variable:

```
    Result                           # of obs.
    -----------------------------------------
    not matched                        10,356
        from master                     5,178  (_merge==1)
        from using                      0      (_merge==2)

    matched                             5,824  (_merge==3)
```

The value 1 of the _merge variable (only present in individual file) designates individuals who do not currently live with a partner (no idspou07 defined). There are no people with a value of 2 (only present in partner file), because we dropped the individuals without a partner from the partner file. The value 3 of the _merge variable designates individuals who live with their partner (idspou07 validly defined). If we only want individuals with a partner in the file, we keep only observations where _merge equals 3.

```
keep if _merge==3
```

Finally, we delete the _merge variable.

```
drop _merge
```

Note that there are now two observations for the same couple. If we want to look at couples as units of analysis, we need to drop one of the two observations.

Stata Example 6: Renaming many variables when creating a partner file
In example 5, we only added one variable of the partner (p07c44_p) to the individual file. If we want to merge many variables of the partner, variable names can be renamed more efficiently with the `renvars` command. `renvars` is a user written command that has to be downloaded and installed by the data user. If you don't have the renvars command installed yet, type `search renvars, net` and install the corresponding package.
This is a syntax example to construct a partner file, which adds all variables of the 2007 individual file from the partner.

```
use shp07_p_user, clear
renvars _all, postfix(_p) // rename all variables: postfix with _p
drop idspou07_p           // delete the variable idspou07
rename idpers_p idspou07  // create a new variable idspou07
save temp_partner07, replace

use shp07_p_user, clear
merge m:1 idspou07 using temp_partner07, keep(master match)
drop _merge
```

3.2 Create a file with information of mothers and fathers

The procedure to add information of fathers and mothers is equivalent to that with partners. One small additional step is necessary, because the identifiers of the father and mother are in the individual master file and not in the annual individual file (see Stata example 2). The reason is that fathers and mothers cannot change over time, in contrast to partners.
Note that with this approach, we can only add information of parents who responded to the individual questionnaire. For almost all respondents, there is additional information on the parents in the social origin file (shp_so). The information of the social origin file differs in two respects from information in the annual file. First, the annual files contain responses by the parents themselves, whereas the social origin variables contain information by the individual about their parents. Second, the annual files refer to the situation at the time of the interview, whereas the social origin variables are retrospective questions about the situation when the respondent was 15 years old.

Stata Example 7: Adding variables from the mother and father
First, we add identifiers of parents from the individual master file to the annual individual file.[12]

```
use status07 p07c44 idpers using shp07_p_user if status==0,clear
/* only individuals who responded to the ind. questionnaire */
merge 1:1 idpers using shp_mp, keepusing (idfath__ idmoth__) ///
keep (match master) nogen // we do not want to add other individ.
save temp_parents, replace
```

Second, we create a temporary file containing the variables of the mother. We also drop the variable idfath__ because it is not needed.

```
use temp_parents, clear
drop idmoth__ idfath__
rename idpers idmoth__
rename p07c44 p07c44_m
save temp_mother, replace
```

Third, we create a temporary file containing the variables of the father (same as for mother).

```
use temp_parents, clear
drop idfath__ idmoth__
rename idpers idfath__
rename p07c44 p07c44_f
save temp_father, replace
```

Fourth, we merge the variables of the parents to the individual file

```
use temp_parents, clear
merge m:1 idmoth__ using temp_mother, ///
keepusing (p07c44_m) keep (match master) nogen
merge m:1 idfath__ using temp_father, ///
```

[12] When typing, note that the variable idfath__ and idmoth__ have two underscores.

```
keepusing (p07c44_f) keep (match master) nogen
save parents2007, replace
```

3.3 Exercise: add information from other household members

Exercise 4: Correlations with life satisfaction of partner and parents
(Exercise to add information from household members)

1. Create a data set with individuals who completed the individual questionnaire in 2008 (n=6904). The data set should contain the following variables:
   idpers
   p08c44 satisfaction with life
   p08c44_p partner's satisfaction with life (create this variable)
   p08c44_f father's satisfaction with life (create this variable)
   p08c44_m mother's satisfaction with life (create this variable)
2. Look at the correlations of life satisfaction
   How many couples are in the data? Is life satisfaction between partners correlated?
   How many mother-child pairs are in the data? Is life satisfaction between mothers and children correlated?
   How many father-child pairs are in the data? Is life satisfaction between fathers and children correlated?

4 Creating long data files (person period files)

4.1 Wide and long data formats

When merging different waves of a panel (as in examples 1 and 4), we add variables of other years to the open data set in a "horizontal" way. One observation (one line) in the data set then contains information of several panel waves. This format is called wide form. Table 3 shows an example of a wide form data set, with three individuals and net income from employment (i$$empyn) of several years.

idpers	i04empyn	i05empyn	i06empyn	i07empyn
4101	103190	107730	113400	122470
42101	63180	69500	.	.
56102	35473	.	41400	45500

Table 3: Example of a data set in wide format

We can present the same data in a different form, the so called long form, as in Table 4. In this data set, one individual may have several observations (from different time points). The information about the year of data collection is not anymore contained in the variable names of income from employment but in a newly created variable "year". The idpers is no longer a unique identifier. Each observation is now uniquely identified through the combination of idpers and year.

idpers	year	iempyn
4101	2004	103190
4101	2005	107730
4101	2006	113400
4101	2007	122470
42101	2004	63180
42101	2005	69500
56102	2004	35473
56102	2006	41400
56102	2007	45500

Table 4: Example of a data set in long format

Data in the long form have more observations and fewer variables than data in the wide form. In the long form, there is no single unique identifier variable. Rather, each observation is uniquely identified by the combination of idpers and year. The two-level structure of the data in the long form is clearly visible. There is a first level of individuals (indicated by idpers) and a second level of the year (indicated by the variable year).
Long data files are very convenient, and often necessary for longitudinal analysis. Multilevel models and all xt commands in Stata require the data to be in long format. In order to work with the different levels, we have to tell Stata what the levels are using the xtset command.

```
xtset idpers year
```

4.2 Create a long file with the append command

A long data set with different waves of the SHP can be constructed with the append command. In a first step, we modify the data sets of each wave. This involves removing the year indication in the variable names and creating a new variable year. We store these files as temporary files. In a second step, we stack the data of each year into the same data set. The Stata syntax is shown in example 8.

Stata Example 8: creating a long file with append
First step: create a temporary file for each wave

```
local wave = 1 // this is a macro, it has the value 1

foreach y in 00 01 02 03 04 05 06 07 08 {
   use idpers  status`y' i??wyn p??w32 p??w39 p??p10 p??c44 ///
   educat??  civsta?? p??w46 age?? using shp`y'_p_user if ///
   status`y'==0, clear
   renvars age?? educat??  status??  civsta??, postsub(`y')
   renvars  p`y'* , presub (p`y' p)
   renvars  i`y'* , presub (i`y' i)
   gen wave=`wave' // replaces `wave' with the value of the macro
   save tempp`y', replace
   local wave = `wave' + 1 // increases the value of the macro by 1
}
```

The `renvars` command is very useful to rename variable names (see also example 6 or type `help renvars` in Stata for details). The `postsub` option replaces the suffix of the variable names. The presub option replaces the prefix of the variable names in the same way than the postsub option. The brackets inform about the details of the rename: we want for instance to replace p08 by p or i06 by i. The local macro "wave" is a counter. It starts with the value 1. In each loop, the counter is increased by one, so that it is 1 for 00, 2 for 01, 3 for 02 etc.

Second step: append the files

```
use tempp08, clear
foreach y in 07 06 05 04 03 02 01 00 {
   append using tempp`y'
}
```

4.3 Create a long file with the merge and reshape command

With the reshape command, it is possible to change between the wide and the long data form. The Stata command `reshape long` converts data from wide form to long form. We indicate the two identifier variables (here idpers and year). Depending on the variable names and size of the data set, the reshape command can be complex to use and may take some time.
With the data of the SHP, it is not possible to reshape data automatically to the long format. This is, because the year information (e.g. 99, 01, 08) in the variable names is sometime at the end of the variable name (e.g. wstat$$, idspou$$) and sometimes in the middle of the variable name (e.g. p$$p19, p$$c44). The reshape command requires some additional specifications. An example of the reshape command is presented in example 9. This syntax can be copied and adapted to other variables.
An alternative way to reshape is to rename the variables before the reshape command, so that all variables have the year identifier at the end of the variable name (as in educat00-educat06). This takes additional steps but makes the use of the reshape command substantially easier.

Stata Example 9: Creating a long file with reshape
First step: create a wide data set (merge command, equivalent to example 4)

```
use idpers idhous?? p??p10 i??wyn p??w32 p??w39 age?? ///
    status99 p??c01 p??c02 p??c03 p??c08 p??c11 ///
    using shp99_p_user if status99==0, clear

foreach y in 00 01 02 03 04 05 06 07 08 {
    merge 1:1 idpers using shp`y'_p_user, ///
    keepusing(idhous?? i??wyn p??w32 p??w39 age?? status?? ///
    p??c44 p??c01 p??c02 p??c03 p??c08 p??c11)
    drop if _merge==2 & status`y'!=0
    drop _merge
}
```

Second step: reshape to the long format

```
reshape long idhous@ p@p10 i@wyn p@w32 p@w39 p@c44 ///
    p@c01 p@c02  p@c03 p@c08 p@c11 age@ status@ , ///
    i(idpers) j(year ///
    "99" "00" "01" "02" "03" "04" "05" "06" "07" "08") atwl ()
```

We have to declare for every variable manually where the year indication is placed within the variable name using the @ character. With the option i we indicate the higher level variable (here idpers), with the option j the lower level variable (here year), which is constructed during the transformation into the long form. We also specify which characters represent the year. The atwl () option indicates that the @ characters are not part of the variable names but stand for the j-variable (here year).
Stata informs us about the transformations made, which involve increasing the number of observations, decreasing the number of variables, creating the variable years and changing the variable names. Also variables that were not found in the dataset (such as the not existing variable p99c44) are reported.

```
Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                    14786   ->  147860
Number of variables                 121   ->      15
j variable (10 values)                    ->   yearxij variables:
         idhous99 idhous00 ... idhous08   ->   idhous
               p99p10 p00p10 ... p08p10   ->   pp10
               i99wyn i00wyn ... i08wyn   ->   iwyn
               p99w32 p00w32 ... p08w32   ->   pw32
               p99w39 p00w39 ... p08w39   ->   pw39
               p99c44 p00c44 ... p08c44   ->   pc44
               p99c01 p00c01 ... p08c01   ->   pc01
               p99c02 p00c02 ... p08c02   ->   pc02
               p99c03 p00c03 ... p08c03   ->   pc03
               p99c08 p00c08 ... p08c08   ->   pc08
               p99c11 p00c11 ... p08c11   ->   pc11
                  age99 age00 ... age08   ->   age
         status99 status00 ... status08   ->   status
-----------------------------------------------------------------------------
```

The variable year that has been created looks like this:

```
tab year
```

```
      year |      Freq.     Percent        Cum.
-----------+-----------------------------------
         0 |     14,786       10.00       10.00
         1 |     14,786       10.00       20.00
         2 |     14,786       10.00       30.00
         3 |     14,786       10.00       40.00
         4 |     14,786       10.00       50.00
         5 |     14,786       10.00       60.00
         6 |     14,786       10.00       70.00
         7 |     14,786       10.00       80.00
         8 |     14,786       10.00       90.00
        99 |     14,786       10.00      100.00
-----------+-----------------------------------
     Total |    147,860      100.00
```

We see firstly, that the leading zeros (e.g. in 00 01 02 03) are omitted from the variable year and that 99 appears as the last year instead of the first year. We therefore recode the variable year:

```
recode year 99=1999 0=2000 1=2001 2=2002 3=2003 4=2004 5=2005 ///
6=2006 7=2007 8=2008
```

We also see that Stata created observations for each individual and each time period. If individuals did not participate in that year, observations contain only missing values. If we only want individuals who were in the annual files, we type

```
keep if status<.
```

Or, if we only want individuals who responded to the individual questionnaire, we type

```
keep if status==0
```

Tabulating the year variable again, we get

```
  tab year
```

```
      year |      Freq.     Percent        Cum.
-----------+-----------------------------------
      1999 |      7,799       11.55       11.55
      2000 |      7,073       10.47       22.02
      2001 |      6,601        9.77       31.79
      2002 |      5,700        8.44       40.23
      2003 |      5,220        7.73       47.96
      2004 |      8,067       11.94       59.91
      2005 |      6,537        9.68       69.58
      2006 |      6,659        9.86       79.44
      2007 |      6,980       10.33       89.78
      2008 |      6,904       10.22      100.00
-----------+-----------------------------------
     Total |     67,540      100.00
```

4.4 Exercise: creating a long file

Exercise 5: Create a long file with append
Create a long file with the variables life satisfaction (p$$c44) and the languages of personal use (p$$e16 and p$$e17).

1. Create temporary files for each wave
   - Keep individuals who answered the individual questionnaire
   - Keep variables p$$c44 p$$e16 p$$e17
   - Remove the year identifier in the variables p$$c44 p$$e16 p$$e17
   - Create a variable wave
   - Save the temporary files
2. Append the temporary files
   - Verify the file (browse, codebook, tabulate the year variable)
   - Define the data to be a panel (xtset command)
3. Save the file (name it longfile1.dta)

Exercise 6: Create a long file with reshape

1. Open the data set created in example 3 (satisfaction_allwaves.dta)
2. Transform it into the long data format.

5 Working in the long file

5.1 The bysort prefix

When working with clustered data, the bysort prefix is very useful. In the SHP, you will often work with clustered data, for instance with households and persons in the individual files, with persons and time points in the long data format, or with spatial clustering if you include municipality or canton characteristics in your data. The commands `generate` and `egen` are useful in combination with the `bysor`t prefix. We demonstrate the use of the `bysort` prefix with an example.

Stata example 10: the number of household members 65 years or older
Assume that we need a variable which indicates the number of household members who are older than 64 years. For this we open an annual individual file

```
use shp08_p_user, clear
```

We keep all observations, irrespectively if they have been interviewed individually or not, but flag individuals who are at least 65 year old.

```
gen flag=age08>=65 & age08<.
tab flag
```

There are 1601 individuals who are at least 65 years old. Now we analyse the flag variables within a household.

```
bysort idhous08: egen nbpers65=total(flag)
tab nbpers65
```

```
   nbpers65 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      8,936       82.06       82.06
          1 |      1,042        9.57       91.63
          2 |        908        8.34       99.97
          3 |          3        0.03      100.00
------------+-----------------------------------
      Total |     10,889      100.00
```

There are 1042 individuals who live in a household with 1 person above 64 years. 908 individuals live in a household with 2 persons above 64 years. 3 individuals live in a household with 3 persons above 64 years. If, however, we need the number of households with persons above 64 years, we have to count every household only once. We can do this with another flag variable, which indicates the first observation per household.

```
egen firstobs=tag(idhous08)
tab nbpers65 if firstobs==1   //or: if idhous[_n]==1
```

```
   nbpers65 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      3,299       74.04       74.04
          1 |        714       16.02       90.06
          2 |        442        9.92       99.98
          3 |          1        0.02      100.00
------------+-----------------------------------
```

5.2 Working with lags

Once a panel has been defined with the xtset command, you can use lags in the stata syntax. For instance, "l.wstat" is the value of the variable wstat (working status) of an individual in the previous year and "l2.wstat" is the value of the variable wstat of an individual two years ago. Examples 11 and 12, as well as exercises 7 and 8 present the use of lags. In addition to lags, leads (f.) are possible.

Stata example 11: Change scores
Using the long file created in example 3, which includes life satisfaction of all years from 2000-2008, we want to create a variable, which indicates the change of life satisfaction since the last year.

```
xtset idpers year // be sure your data is xtset
gen diffsat = pc44-l.pc44 if pc44>=0 & l.pc44>=0
```

Stata example 12: impute height from first wave and calculate BMI from 2004 to 2008
Height is asked of the individuals in their first wave only (from 2004 on), under the assumption that this variable (p$$c45) remains constant. In contrast, weight (p$$c46) is asked each year from 2004 on). This example presents the syntax to impute height in all years until 2008 and calculate the body mass index (bmi)

```
foreach y in 04 05 06 07 08 {
```

```
 use idpers p`y'c45 p`y'c46 using shp`y'_p_user, clear
 rename p`y'c45 height
 rename p`y'c46 weight
 gen year=20`y'
 save temp_20`y', replace
}

 use temp_2004
 forval y = 2005/2008 {
  append using temp_`y'
 }
 xtset idpers year
 save temp, replace

 use temp, replace
 recode ?eight (-8/-1=.)
 replace height=height/100 // was in cm, now in m
 forval y = 2005/2008 {
  replace height=l.height if height<1 | height>=.
 }
 gen bmi = weight / height^2
 histogram bmi  // check
```

5.3 Exercises working in the long file

Exercise 7: Create variable "language of personal use"
(Working with lags)
Respondents are asked about their first and second language in their first interview only (because this is considered as stable over time). This implies that only one annual file per person - when the first interview has taken place - contains positive values for the variables p$$e16 and p$$e17. If, for instance, we want to analyse the first and second language of individuals who responded in 2008, we have to merge this information from the data files of earlier waves.
In this exercise, we are interested in the first and second language of personal use of the individuals with a personal interview in 2008.

1. Create a long file containing the variables p$$e16 and p$$e17 or open the file longfile1 (from exercise 5)
2. Define the panel variables (xtset)
3. Create the variables firstlan (first language) and secondlan (second language), which are identical to p08e16 and p08e17.
4. Replace the values of firstlan and secondlan by p$$e16 p$$e17 of previous waves, if firstlan and secondlan are missing, but p$$e16 p$$e17 of a previous waves is positive.
5. Cross tabulate first and second language in 2008. How many of the 2008 are bilingual (create a dummy variable bilingual)?

Exercise 8: Create variable "number of personal interviews"
(Practise the use of the bysort prefix)

1. Create a long file or open the file created in exercise 5 (longfile 1).
2. Create a variable which indicates the number of individual interviews a person has responded to between 1999 and 2008.
3. How many persons have responded 10 times? How many persons have responded 9 times?
4. Is there an easier way to check the number of individual interviews using one file only?

Exercise 9: assigning social class
(uses long files, partner files, parent files, data management in the long file)
(This is a time consuming exercise intended for experienced Stata users)
Variables of social stratification (e.g. the European socio-economic class, esec) are constructed at the basis of job characteristics. This implies that we can only attribute an esec (European socio-economic class) to respondents who are working. For example, about 33 percent of respondents do have a missing esec (variable esec08mj, esec of main job) in the SHP individual file from 2008. To attribute an esec to the maximum number of respondents, we impute missing values with the esec of earlier panel waves, of the last job[13], of the partner or of the parents.
The aim of exercise 9 is to assign an esec to the maximum number of respondents in 2008, by imputing missing values successively from earlier waves, the last job, from partners, from fathers and from mothers.

- Impute esec in 2008 with esec from earlier panel waves
- to merge the esec of the last job and try to replace still missing values
- to add the esec of the partner and try to replace still missing values
- to add the esec of the father and try to replace still missing values
- to add the esec of the mother and try to replace still missing values

[13] Question about the last job are asked at the time of the first interview to individuals who were not economically active then. The variables are in the file shp_lj.