

Swiss Household Panel (SHP)

Data management with SPSS

SHP Workshop, June 8, 2010

By

Valérie-Anne Ryser, Florence Lebert, and Ursina Kuhn

| | | |
|----------|--|-----------|
| 1 | DATA MANIPULATION | 2 |
| 1.1 | VARIABLES SELECTION COMMAND | 2 |
| 2 | MERGE FILES | 4 |
| 2.1 | IDENTIFIERS IN THE SHP | 4 |
| 2.2 | THE MERGE COMMAND IN SPSS: ONE TO ONE MERGE | 5 |
| 2.3 | THE MERGE COMMAND IN SPSS: ONE TO MANY MERGE | 6 |
| 3 | ADD INFORMATION FROM OTHER HOUSEHOLD MEMBERS | 9 |
| 3.1 | CREATE A FILE WITH PARTNER INFORMATION | 9 |
| | <i>Duplicate the individual file</i> | 9 |
| | <i>Create a common key for matching with individual file W10 2008</i> | 10 |
| | <i>Match partner file on individual file</i> | 10 |
| 3.2 | CREATE A FILE WITH INFORMATION OF MOTHERS AND FATHERS WITH THE MERGE COMMAND | 11 |
| 4 | CREATING LONG DATA FILES (PERSON PERIOD FILES) | 12 |
| 4.1 | WIDE AND LONG DATA FORMATS | 12 |
| | <i>The varstocases command</i> | 12 |
| | <i>The loop command</i> | 14 |
| 4.2 | LAG VARIABLE | 17 |
| 5 | SPSS TIPS | 18 |
| 6 | REFERENCES | 18 |
| | <i>The SPSS Help menu</i> | 18 |
| | <i>Online Resources</i> | 18 |
| | <i>Book</i> | 19 |

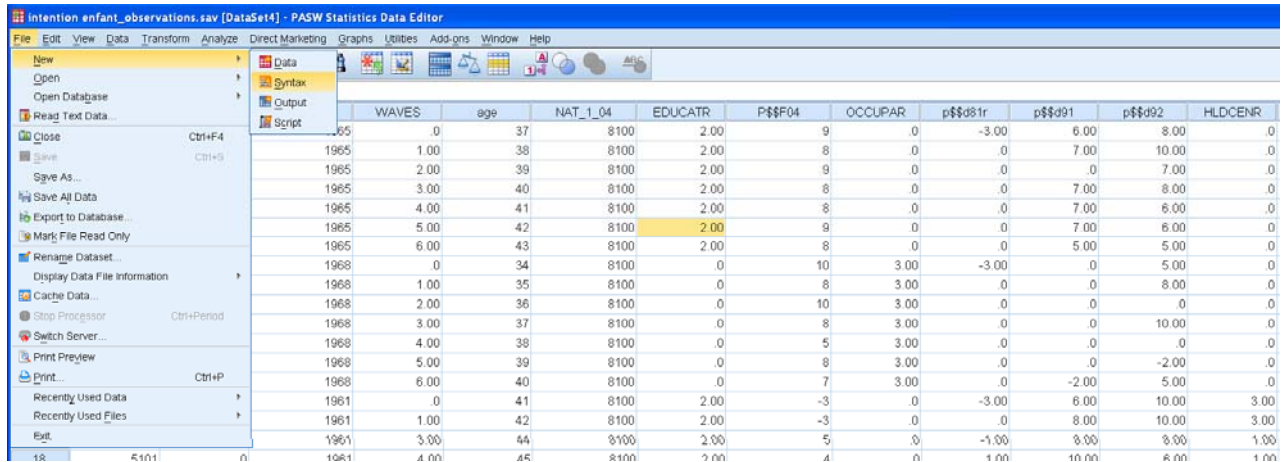


1 Data manipulation

First of all a lot of data manipulation is possible through buttons and SPSS dialog box. However, here we will favor the use of the **syntax**. The use of syntax while using SHP data has three advantages:

1. There are many variables in 11 different files, it is easier and quicker to write the variables to keep instead of searching in a dialog box for what you need.
2. You have a trace of what you did and which can be reused or transmitted
3. Very easy to adapt for similar research questions or addition of variables.

With SHP data this is a gain of time and it is a way to organize your analysis. There is no need to keep data file but it is much more important to organize your syntaxes.



| | WAVES | age | NAT_1_04 | EDUCATR | P\$\$\$F04 | OCCUPAR | p\$\$\$d81r | p\$\$\$d91 | p\$\$\$d92 | HLDCENR |
|------|-------|-----|----------|---------|------------|---------|-------------|------------|------------|---------|
| 1965 | .0 | 37 | 8100 | 2.00 | 9 | .0 | -3.00 | 6.00 | 8.00 | .0 |
| 1965 | 1.00 | 38 | 8100 | 2.00 | 8 | .0 | .0 | 7.00 | 10.00 | .0 |
| 1965 | 2.00 | 39 | 8100 | 2.00 | 9 | .0 | .0 | .0 | 7.00 | .0 |
| 1965 | 3.00 | 40 | 8100 | 2.00 | 8 | .0 | .0 | 7.00 | 8.00 | .0 |
| 1965 | 4.00 | 41 | 8100 | 2.00 | 8 | .0 | .0 | 7.00 | 6.00 | .0 |
| 1965 | 5.00 | 42 | 8100 | 2.00 | 9 | .0 | .0 | 7.00 | 6.00 | .0 |
| 1965 | 6.00 | 43 | 8100 | 2.00 | 8 | .0 | .0 | 5.00 | 5.00 | .0 |
| 1968 | .0 | 34 | 8100 | 0 | 10 | 3.00 | -3.00 | .0 | 5.00 | .0 |
| 1968 | 1.00 | 35 | 8100 | 0 | 8 | 3.00 | .0 | .0 | 8.00 | .0 |
| 1968 | 2.00 | 36 | 8100 | 0 | 10 | 3.00 | .0 | .0 | .0 | .0 |
| 1968 | 3.00 | 37 | 8100 | 0 | 8 | 3.00 | .0 | .0 | 10.00 | .0 |
| 1968 | 4.00 | 38 | 8100 | 0 | 5 | 3.00 | .0 | .0 | .0 | .0 |
| 1968 | 5.00 | 39 | 8100 | 0 | 8 | 3.00 | .0 | .0 | -2.00 | .0 |
| 1968 | 6.00 | 40 | 8100 | 0 | 7 | 3.00 | .0 | -2.00 | 5.00 | .0 |
| 1961 | .0 | 41 | 8100 | 2.00 | -3 | .0 | -3.00 | 6.00 | 10.00 | 3.00 |
| 1961 | 1.00 | 42 | 8100 | 2.00 | -3 | .0 | .0 | 8.00 | 10.00 | 3.00 |
| 1961 | 3.00 | 44 | 8100 | 2.00 | 5 | .0 | -1.00 | 8.00 | 8.00 | 1.00 |

Several ways exist to select a subsample of the data set you have. There are also several possibilities to build a temporary file with reduced observations.

However, this documentation focuses on the selection of variables. If you work with a longitudinal file, it is not convenient to work with ten thousand variables.

Before the construction of a longitudinal file, we propose to make a selection of the variables you need to answer your research question.

1.1 Variables selection command

It is possible to select only the variables you need, without deleting variables which is very laborious with big files such as the SHP data files.

To create a smaller file use the command **/keep** to select only the variables you are interested in.

In order to create a smaller file by deleting the variables you do not need, it is possible to use the command **/drop**. If you have many variables, it is easier to keep the variables you need instead of dropping variables you do not want in your smaller file.

The keep variables command: example w2_2000

```
get file = 'N:\Workshop\SHP00_P_USER.sav'  
/keep idpers idhous00 age00 status00 NAT_1_00 EDUCAT00  
P00P19 P00P02 P00F04 OCCUPA00 P00F08 CIVSTA00 P00C44  
P00L25 PAUG00R4 P00N15 P00C01 I00PTOTG WEIP00TS WEIP00LS.  
SORT CASES BY  
idpers (A) .  
save outfile='N:\Workshop\Temp\P00temp.sav'.
```

A sorted file is essential to a successful merge file. This is why we propose saving the file sorted by the identifier in order to facilitate the next step which is the merge file.

2 Merge files

The purpose is to have a longitudinal file which allows the understanding of changes. A merge of two or more data sets means the combination of these to a new data set. Variables from the second data set are added to the cases of the first data set.

To do so we need an identifying variable that is present in both data sets. The identifier variable assures that the values of the same individual are on the same line.

2.1 Identifiers in the SHP

When merging two data sets, we require an identifier which is present in all files you want to merge. In the SHP we have the following identifiers.

| | | |
|--|--|--|
| Individual master file | shp_mp | idpers, idhous\$\$ ¹ , idfath\$\$, idmoth\$\$ |
| Individual annual files | shp\$\$_p_user | idpers, idint, idhous\$\$, idspou\$\$, reppers |
| Additional ind. files (Social origin, last job, calendar, biographic) | shp\$\$_so, shp\$\$_lj shp\$\$_ca, shp0_* | idpers |
| Last job | shp\$\$_lj | idpers |
| Interviewer data | | idint |
| Household annual files | shp\$\$_h_user | idhous\$\$, reppers, canton\$\$, (gdenr ²) |
| Biographic files | | idpers |

¹ \$\$ represent the year of interview

² Municipality number of the Swiss federal office of statistic

2.2 The Merge command in SPSS: one to one merge

When you have at least two data files, you can combine them by merging them side by side, matching up observations based on an identifier.

| IDPERS | AGE02 |
|---------------|--------------|
| 4101 | 37 |
| 4102 | 34 |
| 4103 | 11 |
| 4104 | 9 |
| 4105 | 6 |

| IDPERS | AGE03 |
|---------------|--------------|
| 4101 | 38 |
| 4102 | 35 |
| 4103 | 12 |
| 4104 | 10 |
| 4105 | 7 |

For example, do we have a file SHP02_P_USER.sav containing **IDPERS** and a file SHP03_P_USER.sav also containing **IDPERS**. We would like to merge the files together in order to have the SHP02_P_USER.sav observations on the same line with the SHP03_P_USER.sav observations based on the key variable **IDPERS**.

To merge two files, follow the next steps:

1. Use **SORT CASES**³ to sort **IDPERS** from the SHP02_P_USER.sav and save that file.
2. Use **SORT CASES** to sort **IDPERS** on SHP03_P_USER.sav and save that file.
3. Use **MATCH FILES** to merge the SHP02_P_USER.sav and SHP03_P_USER.sav files based on **IDPERS**.

The sort variables command: example w4 2002 – w5 2003

```
get file = 'N:\Workshop\SHP02_P_USER.sav'.
SORT CASES by idpers.
exe.
save outfile = 'N:\Workshop\SHP02_P_USER.sav'.

get file = 'N:\Workshop\SHP03_P_USER.sav'.
SORT CASES by idpers.
exe.
save outfile = 'N:\Workshop\SHP03_P_USER.sav'.
```

³ Note that SPSS sorts the cases in an ascending order (A). You can also sort them in a descending order. Therefore write sort BY idpers (D).

The match file command: example w4 2002 – w5 2003

```
MATCH FILES
/FILE = 'N:\Workshop\SHPO2_P_USER.sav'
/FILE = 'N:\Workshop\SHPO3_P_USER.sav'
/BY idpers.
exe.
```

Output

| IDPERS | AGE02 | AGE03 |
|--------|-------|-------|
| 4101 | 37 | 38 |
| 4102 | 34 | 35 |
| 4103 | 11 | 12 |
| 4104 | 9 | 10 |
| 4105 | 6 | 7 |

2.3 The Merge command in SPSS: one to many merge

The next example considers a **one to many** merge where one observation in a file may have multiple matching records in another file. For instance for the SHP we have a **Household** file, and we have an **individual** file. Each individual lives in a **household** and a household can be composed by more than one individual.

Here we want to merge the household information for all household members.

We call this merge a **one to many** merge since we are matching **one household** observation to **one or many individual** observations. The household file called **SHPO2_H_USER.sav** is the file with **one** observation, and the **SHPO2_P_USER.sav** file is the one with **many** observations within a household.

Individual file

| IDPERS | IDHOUS02 | AGE02 |
|--------|----------|-------|
| 4101 | 41 | 37 |
| 4102 | 41 | 34 |
| 4103 | 41 | 11 |
| 4104 | 41 | 9 |
| 4105 | 41 | 6 |

Household file

| IDHOUS02 | HDATE02 | HLINGU02 | CANTON02 |
|----------|------------|----------|----------|
| 41 | 17.09.2002 | 1 | 8 |
| 51 | 19.09.2002 | 1 | 8 |
| 111 | 03.10.2002 | 2 | 24 |
| 131 | 14.11.2002 | 2 | 8 |

As you see below, the steps for doing a **one to many** merge is quite similar to the **one to one merge** that we saw above.

1. Use **SORT CASES BY** to sort **IDHOUS02** on **household** file and save that file (with the same name or with an other name)
2. Use **SORT CASES BY** to sort **IDHOUS02** on individual file and save that file
3. Use **MATCH FILES** to merge the **household** and **individual** files. However, since the **SHP02_H_USER.sav** file is the file with **one** observation, use **/TABLE= 'SHP02_H_USER.sav'** , not **/FILE = 'SHP02_H_USER'** to specify the *household file*.

The key difference between a **one to one** merge and a **one to many** merge is that you need to use **/TABLE= 'SHP02_H_USER.sav'** instead of **/FILE = 'SHP02_H_USER.sav'**. For the construction of your own merged data file, ask yourself two questions:

1. Do I have a one to one merge?

If no:

2. Which file plays the role of **one**? For that file, use **/TABLE=** instead of **/FILE=**.

The match table command: example w4 - 2002

```
MATCH FILES
  /FILE= 'N:\Workshop\SHP02_P_USER.sav'
  /TABLE = 'N:\Workshop\ SHP02_H_USER.sav'
  /BY idhous02.
EXECUTE.
```

Output

| IDPERS | IDHOUS | AGE02 | HLINGU02 |
|---------------|---------------|--------------|-----------------|
| 4101 | 41 | 37 | 1 |
| 4102 | 41 | 34 | 1 |
| 4103 | 41 | 11 | 1 |
| 4104 | 41 | 9 | 1 |
| 4105 | 41 | 6 | 1 |

Mismatching records in one-to-one merge

In the SHP we have every year new household members who join the survey. This means that observations for an extra individual arrived in 2000 will not match to data in the individual file in 1999.

SPSS recommends using **/IN=fromind_99** in order to create a 0/1 variable that indicates whether the resulting file contains a record with data from the SHP99_P_USER.sav file or from the SHP00_P_USER.sav. Similarly, we use **/IN=fromind_00** to indicate if the resulting file has a record from this file.

/IN command: example w1 1999 – w2 2000

```
MATCH FILES
/FILE = 'N:\Workshop\SHP99_P_USER.sav' /IN fromind_99
/FILE = 'N:\Workshop\SHP00_P_USER.sav' /IN fromind_00
/BY idpers.
LIST.
```

```
CROSSTABS
/TABLES= fromind_99 by fromind_00.
```

It is recommended to use the **LIST** and **CROSSTABS** to show the mismatching records. For the SHP, considering the numbers of variables, the command **LIST** is interesting to use after a selection of variables.

If you want to have individuals or households followed for a certain number of years, you can use the command **SELECT IF** to eliminate the non-matching records.

A select if command: example w1 1999 – w2 2000

```
SELECT IF fromind_99 AND fromind_00.
LIST.
```


3 Add information from other household members

The SHP interviews all individuals who live in the same household. This opens many analytical possibilities, such as looking at partners or influences between parents and children.

Individual file

| IDPERS | IDHOUS99 | SEX | IDSPOU99 | EDUCAT99 | P99F08 |
|--------|----------|-----|----------|----------|--------|
| 4102 | 41 | 1 | 4101 | 8 | 2 |
| 4101 | 41 | 2 | 4102 | 1 | 19 |

Output

| IDPERS | IDHOUS99 | IDSPOU99 | EDUCAT99 | EDUCAT99_P | P99F08 | P99F08_P |
|--------|----------|----------|----------|------------|--------|----------|
| 4102 | 41 | 4101 | 8 | 1 | 2 | 19 |
| 4101 | 41 | 4102 | 1 | 8 | 19 | 2 |

For such analyses you need to follow 3 steps:

1. Duplication of the individual file
2. Creation of a common key
3. Match file

3.1 Create a file with partner information

The file SHP08_P_USER.sav has to be sorted by `ids pou08` which is the key variable for a partner file. You already have selected the variables you are interested in.

Duplicate the individual file

For instance duplicate the individual file of W10 and change the variable names to create a partner file.

Select the variables you need with the `/keep` command.

Rename command: example w10 - 2008

```
rename var idpers = idpers_p.  
rename var P08C44 = P08C44_p.  
rename var p08d91 = p08d91_p.  
rename var p08d92 = p08d92_p.  
rename var EDUCAT08 = EDUCAT08_p.  
execute.
```

Create a common key for matching with individual file W10 2008

We have seen that idspou\$\$ is a key variable. We need to create this variable in the new partner file in order to merge this file with the individual file.

Recode command: example w10 - 2008

```
compute idspou08=idpers_p.  
sort cases by idspou08.  
  
SAVE OUTFILE='C:\PARTNER_W10.sav' .
```

Match partner file on individual file

Use the already presented command for a one to one or many merge.

Match file command: example w10 – 2008 partner file

```
GET FILE='C:\INDIVIDUAL_W10.sav' .  
  
sort cases by idspou08.  
exe.  
  
MATCH FILES /FILE=*  
  /TABLE='C:\PARTNER_W10.sav'  
  /BY IDSPOU08.  
EXECUTE .
```

Select only cases having a partner

```
select if not missing(idspou08).  
sort cases by idpers.  
SAVE OUTFILE='C:\PARTNERS_FILE_W10.sav'.
```

3.2 Create a file with information of mothers and fathers with the merge command

The approach to add information of fathers and mothers is equivalent to the approach 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.

With this approach we can only add information of parents who responded to the individual questionnaire. Additional information on the parents, for almost all respondents, is available in the social origin file **SHP_SO.sav**.

The information of the social origin file is however very different in nature to the information in the annual file in two respects.

1. First, in the annual files are actual answers by the parents themselves, whereas the social origin variables contain information by the individual about their parents.
2. Second, the annual file refers to the present situation at the time of the interview, whereas the social origin variables are retrospective questions on the situation of the parents when the respondent was 15 years old.

4 Creating long data files (person period files)

4.1 Wide and long data formats

When we have a file with several observation for an individual interviewed yearly, it is called a wide format file. For instance with the SHP, on one line we have the individual with every year observation which makes the data set wide.

Wide file

| IDPERS | AGE02 | AGE03 | AGE04 | AGE05 | AGE06 | AGE07 |
|--------|-------|-------|-------|-------|-------|-------|
| 4101 | 37 | 38 | 39 | 40 | 41 | 42 |
| 4102 | 34 | 35 | 36 | 37 | 38 | 39 |
| 4103 | 11 | 12 | 13 | 14 | 15 | 16 |
| 4104 | 9 | 10 | 11 | 12 | 13 | 14 |
| 4105 | 6 | 7 | 8 | 9 | 10 | 11 |

But for several analyses like multilevel model (also called MLM hierarchical, random effects, mixed effects models) or survival analysis for instance, we need to reshape the data into a long format or a person period file.

A long format or a person period file means that each individual is represented by as many lines as the number of the SHP waves, which makes the data long.

In order to construct such file, two possibilities are available.

1. The first one uses the SPSS command **/varstocases** to reshape the data from wide to long format.
2. The second one uses a **/LOOP** command after the creation of vectors.

The varstocases command

First of all organize your data in order to have a clear file where you understand what you have.

The keep command: longitudinal individual and household file

```
get file = 'N:\Workshop\individual.sav'  
/keep idpers sex birthy nat_1_04 idhous00 idhous01 idhous02  
idhous03 idhous04 idhous05 idhous06 idhous07 idhous08 age00  
age01 age02 age03 age04 age05 age06 age07 age08 status00  
status01 status02 status03 status04 status05 status06  
status07 status08 P00C44 P01C44 P02C44 P03C44 P04C44 P05C44  
P06C44 P07C44 P08C44 p00d91 p01d91 p02d91 p03d91 p04d91  
p05d91 p06d91 p07d91 p08d91 p00d92 p01d92 p02d92 p03d92  
p04d92 p05d92 p06d92 p07d92 p08d92 EDUCAT00 EDUCAT01  
EDUCAT02 EDUCAT03 EDUCAT04 EDUCAT05 EDUCAT06 EDUCAT07
```

```

EDUCAT08  HLDCE00  HLDCE01  HLDCE02  HLDCE03  HLDCE04
HLDCE05  HLDCE06  HLDCE07  HLDCE08  NBKID00  NBKID01  NBKID02
NBKID03  NBKID04  NBKID05  NBKID06  NBKID07  NBKID08.
SORT CASES BY
idpers (A) .
save outfile='N:\Workshop\Temp\selected_var.sav'.

```

The next command is used to reshape the data. We can display the same data in a different form, the so called long form. The data are broken out so that an observation is for a distinct individual – year pair. One individual may have several observations in the data set.

Varstocases command: person period file, individual and household file

```

get file = 'N:\Workshop\selected_var.sav'.

VARSTOCASES /ID = id "vague"
/MAKE idhous FROM idhous00 idhous01 idhous02 idhous03
idhous04 idhous05 idhous06 idhous07 idhous08
/MAKE age FROM age00 age01 age02 age03 age04 age05 age06
age07 age08
/MAKE status FROM status00 status01 status02 status03
status04 status05 status06 status07 status08
/MAKE P$$C44 FROM P00C44 P01C44 P02C44 P03C44 P04C44 P05C44
P06C44 P07C44 P08C44
/MAKE p$$d91 FROM p00d91 p01d91 p02d91 p03d91 p04d91 p05d91
p06d91 p07d91 p08d91
/MAKE p$$d92 FROM p00d92 p01d92 p02d92 p03d92 p04d92 p05d92
p06d92 p07d92 p08d92
/MAKE EDUCAT FROM EDUCAT00 EDUCAT01 EDUCAT02 EDUCAT03
EDUCAT04 EDUCAT05 EDUCAT06 EDUCAT07 EDUCAT08
/MAKE HLDCE0 FROM HLDCE00 HLDCE01 HLDCE02 HLDCE03
HLDCE04 HLDCE05 HLDCE06 HLDCE07 HLDCE08
/MAKE NBKID FROM NBKID00 NBKID01 NBKID02 NBKID03 NBKID04
NBKID05 NBKID06 NBKID07 NBKID08
/INDEX = Index1(9)
/KEEP = idpers sex birthy nat_1_04
/NULL = KEEP.
save outfile = 'N:\Workshop\Person_Periode.sav'.

```

The command syntax is easy to interpret: it collapses all the variables into a "new" variable.

At the same time, a new variable, "Index", is created to index the five new cases within each subject.

The last subcommand means that variables constant within a subject, in other words variables which are not time dependant, should be kept.

There is a shorter way to transfer the data into a long file, but for this you have to make sure that your data file is sorted by variable and wave:

| Name |
|----------|
| IDPERS |
| SEX |
| AGE00 |
| AGE01 |
| AGE02 |
| AGE03 |
| AGE04 |
| IDHOUS00 |
| IDHOUS01 |
| IDHOUS02 |
| IDHOUS03 |
| IDHOUS04 |

Instead of listing all the variables, you can treat them as a serial using TO to tell SPSS which variables it has to account for.

```
VARSTOCASES
  /ID = id
  /MAKE age FROM age00 TO age04
  /MAKE house FROM idhous00 TO idhous04
  /MAKE educat FROM educat00 TO educat04
  /MAKE hldcen FROM hldcen00 TO hldcen04
  /KEEP = IDPERS SEX.
```

The loop command

The second method to reshape the data used loop and vectors. According to De Coster (2004, p54)

“A vector is a set of variables that are linked together because they represent similar things. The purpose of the vector is to provide a single name that can be used to access any of the entire set of variables. A loop is used to tell the computer to perform a set of procedures a specified number of times. Often times we need to perform the same transformation on a large number of variables. By using a loop, we only need to define the transformation once, and can then tell the computer to do the same thing to all the variables using a loop.”

De Coster (2004, p54) add that

“what SPSS calls a .vector. is typically referred to as an .array. in most programming languages. If you are familiar with arrays and loops from a computer programming

course, you are a step ahead. Vectors and loops are used in data manipulation in more or less the same way that arrays and loops are used in standard computer programming”.

Before using this method you need first to have sorted data like shown above. Second, you must **/compute** a variable wave `sex birthy` and `nat_1_04` for each wave. Here is the example for wave and sex only.

Compute variables command

```
compute w00=2.  
compute w01=3.  
compute w02=4.  
compute w03=5.  
compute w04=6.  
compute w05=7.  
compute w06=8.  
compute w07=9.  
compute w08=10.
```

```
compute sex00=sex.  
compute sex01=sex.  
compute sex02=sex.  
compute sex03=sex.  
compute sex04=sex.  
compute sex05=sex.  
compute sex06=sex.  
compute sex07=sex.  
compute sex08=sex.  
execute.
```

Then construct a vector for each variable (remember that your variables need to be sorted).

Vector and loop command

```
vector ww = w00 to w08  
/sx = sex00 to sex08  
/ho = idhous00 to idhous08  
/ag = age00 to age08  
/st = status00 to status08  
/pc = P00C44 to P08C44  
/id = p00d91 to p08d91  
/cd= p00d92 to p08d92  
/ed = EDUCAT00 to EDUCAT08  
/hl = HLDcen00 to HLDcen08  
/kd= NBKID00 to NBKID08.
```

```

loop #i = 1 to 9.
compute vague =ww(#i).
compute sex = sx(#i).
compute idhous = ho(#i).
compute age = ag(#i).
compute status = st(#i).
compute P$$C44 = pc(#i).
compute p$$d91 = id(#i).
compute p$$d92=cd(#i).
compute EDUCAT =ed(#i).
compute HLDCEN = hl(#i).
compute NBKID =kd(#i).

xsave out = "observations.sav"
/keep = idpers vague sex idhous canton age status P$$C44
p$$d91 p$$d92 EDUCAT HLDCEN NBKID.
end loop.
execute.

```

If you use this method run this command all together, do not try to run the vector syntax and the loop syntax separately.

Output

| IDPERS | SEX | age |
|--------|-----|-----|
| 4101 | 1 | 37 |
| 4101 | 1 | 38 |
| 4101 | 1 | 39 |
| 4101 | 1 | 40 |
| 4101 | 1 | 41 |
| 4101 | 1 | 42 |
| 4102 | 2 | 34 |
| 4102 | 2 | 35 |
| 4102 | 2 | 36 |
| 4102 | 2 | 37 |
| 4102 | 2 | 38 |
| 4102 | 2 | 39 |

The CASESTOVARS command

In case you have a data file in the long format and you want to reshape it into the wide format, use the CASESTOVARS command. By the /FIXED subcommand you specify which variables do not change in time.

```
CASESTOVARS
  /ID = IDPERS
  /FIXED = SEX.
```

The long file is shaped back into the wide format:

| IDPERS | SEX | AGE.1 | AGE.2 | AGE.3 | AGE.4 | AGE.5 | AGE.6 |
|--------|-----|-------|-------|-------|-------|-------|-------|
| 4101 | 1 | 37 | 38 | 39 | 40 | 41 | 42 |
| 4102 | 2 | 34 | 35 | 36 | 37 | 38 | 39 |
| 4103 | 1 | 15 | 16 | 17 | 18 | 19 | 20 |

4.2 Lag variable

While you have a longitudinal file, depending on your research question, it can be useful to have the variable which reports the value at the previous wave.

Output

| IDPERS | SEX | EDUC | P\$\$F04 | OCCUP | P\$\$D81 | P\$\$D91 | PD91_la | P\$\$D92 | PD92_la |
|--------|-----|------|----------|-------|----------|----------|---------|----------|---------|
| 4101 | 0 | 2 | 9 | 0 | -3.00 | 6 | - | 8 | - |
| 4101 | 0 | 2 | 8 | 0 | 0 | 7 | 6 | 10 | 8 |
| 4101 | 0 | 2 | 9 | 0 | 0 | 0 | 7 | 7 | 10 |
| 4101 | 0 | 2 | 8 | 0 | 0 | 7 | 0 | 8 | 7 |
| 4101 | 0 | 2 | 8 | 0 | 0 | 7 | 7 | 6 | 8 |
| 4101 | 0 | 2 | 9 | 0 | 0 | 7 | 7 | 6 | 6 |
| 4101 | 0 | 2 | 8 | 0 | 0 | 5 | 7 | 5 | 6 |

Lag command: person period file

```
do if idpers=lag(idpers).
  compute p$c44_r =lag(p$$c44).
  compute p$d91_r =lag(p$$d91).
  compute p$d92_r=lag(p$$d92).
  compute EDUC_r =lag(EDUCAT).
  compute HLDCEN_r =lag(HLDCEN).
end if.
execute.
```

5 SPSS Tipps

- sort your cases by the identifier
- be careful with the missing (missing declaration)
- write path without any mistake
- do not forget the dots, comas, etc.

6 References

The SPSS help menu

- Topics
- Command Syntax Reference
- Tutorial

| | Name | Type | Width | Decimals | Label | Values | | | | Measure | Role |
|----|----------|---------|-------|----------|----------------------|-----------------|------------|----|-------|---------|-------|
| 1 | IDPERS | Numeric | 10 | 0 | Identification n... | {-8, other e... | L | | | Scale | Input |
| 2 | SEXR | Numeric | 8 | 2 | Sex recoded | {-8,00, oth... | L | | | Scale | Input |
| 3 | BIRTHY | Numeric | 12 | 0 | Date of birth: ... | {-8, other e... | L | | | Scale | Input |
| 4 | WAVES | Numeric | 8 | 2 | vagues | {0, vague ... | L | | | Scale | Input |
| 5 | age | Numeric | 10 | 0 | Age in year of... | {-8, other e... | L | | | Scale | Input |
| 6 | NAT_1_04 | Numeric | 12 | 0 | First nationality | {-8, other e... | L | | | Scale | Input |
| 7 | EDUCATR | Numeric | 8 | 2 | Level of educa... | {-8,00, oth... | L | | | Scale | Input |
| 8 | OCCUPAR | Numeric | 8 | 2 | Actual occupat... | {-8,00, oth... | L | | | Scale | Input |
| 9 | P\$F08 | Numeric | 10 | 0 | Time for hous... | {-8, other e... | L | | | Scale | Input |
| 10 | CIVSTAR | Numeric | 8 | 2 | Civil status in t... | {-8,00, oth... | L | | | Scale | Input |
| 11 | HLDCEMR | Numeric | 8 | 2 | Family typolog... | {-8,00, oth... | LO - -1.00 | 10 | Right | Scale | Input |
| 12 | NBKID | Numeric | 12 | 0 | Number of chil... | {-8, other e... | LO - -1 | 12 | Right | Nominal | Input |
| 13 | | | | | | | | | | | |

Online resources

Introduction to SPSS. UCLA: Academic Technology Services, Statistical Consulting Group. from <http://www.ats.ucla.edu/stat/spss/modules/merge.htm> (accessed june 2, 2010) <http://www.ats.ucla.edu/stat/spss/modules/>

<http://faculty.chass.ncsu.edu/garson/PA765/statnote.htm>

DeCoster, J. (2004). Data Analysis in SPSS. Retrieved 06, 03, 2010 from <http://www.stat-help.com/notes.html>

<http://www.uk.sagepub.com/field3e/>

<http://www.statisticshell.com/>

Books

Pallant, J. (2007). *SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS for Windows*. Sydney: Open University Press.

Field, A, (2009). *Discovering Statistics Using SPSS*. Sussex: Sage Publication.