Data Handling in EU-SILC

Mack, Alexander

Veröffentlichungsversion / Published Version
Arbeitspapier / working paper

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:
GESIS - Leibniz-Institut für Sozialwissenschaften

Empfohlene Zitierung / Suggested Citation:
Mack, A. (2016). *Data Handling in EU-SILC.* (GESIS Papers, 2016/10). Köln: GESIS - Leibniz-Institut für Sozialwissenschaften. https://doi.org/10.21241/ssoar.47123

gesis Leibniz-Institut für Sozialwissenschaften

GESIS *Papers* 2016|10

Data Handling in EU-SILC

Alexander Mack

GESIS Papers 2016|10

Data Handling in EU-SILC

Alexander Mack

GESIS – Leibniz-Institut für Sozialwissenschaften 2016

GESIS Papers

GESIS – Leibniz-Institut für Sozialwissenschaften
Dauerbeobachtung der Gesellschaft
Postfach 12 21 55
68072 Mannheim
Telefon: (0621) 1246 - 133
Telefax: (0621) 1246 - 100
E-Mail: alexander.mack@gesis.org

ISSN: 2364-3781 (Online)
Herausgeber,
Druck und Vertrieb: GESIS – Leibniz-Institut für Sozialwissenschaften
Unter Sachsenhausen 6-8, 50667 Köln

Summary

This report aims to assist researchers who conduct analysis on the basis of the European Statistics on Income and Living Conditions (EU-SILC). The main objective is to explain the structure of the EU-SILC data and aid researchers in compiling information from the different data files. To this aim, the report includes a number of practical examples which explain how to merge the different files and how to aggregate information from different levels of analysis. All practical examples include code for both SPSS and Stata which can be used or customized to construct data files for different types of analyses. The primary target audience of this report is quantitative researchers with little experience with complex datasets. However it might also prove useful to more experienced analysts unfamiliar with the specifics of the EU-SILC.

1 Introduction

The European Statistics on Income and Living Conditions (EU-SILC) is a yearly data collection effort conducted by Eurostat in cooperation with the National Statistical Institutes (NSIs) of the European Union, European Free Trade Association (EFTA) and candidate countries. The primary objective of the EU-SILC is to provide comparable data on income, poverty, social exclusion and living conditions (Eurostat 2013: 13). Eurostat employs the EU-SILC as an important data source for indicators on income, poverty and living conditions in the EU and to evaluate progress towards EU policy objectives. The first round of the EU-SILC was carried out in 2003 on the basis of a gentlemen's agreement in 6 states (Eurostat 2016). Since 2004 the EU-SILC is conducted on the basis of EU legislature and microdata is made available free of cost to accredited researchers from 2004 onward in form of the EU-SILC User Database (UDB) (Eurostat 2015). Figure 1 below provides an overview over the availability of the EU-SILC UDB over time and by country up until the 2013 release.

The EU-SILC is devised as an output-harmonized data collection effort by Eurostat and National Statistical Institutes (NSIs). Output harmonization means that Eurostat defines a set of target variables, which are described in the document "Methodological guidelines and description of target variables" commonly referred to as "Guidelines" (Eurostat 2013) and defines a number of quality criteria in regards to data collection (defined in Commission Regulation No 28/2004). The NSIs are responsible for the data collection efforts in their country. While in most countries data for the EU-SILC is collected via a survey in a number of countries a large part of data is collected from registers (Jäntti et al. 2013).

The EU-SILC includes both a longitudinal and cross- sectional component. This is achieved by realizing the EU-SILC as a rotating panel study in which households are interviewed in four[1] consecutive years (Eurostat 2013: 19ff.). From this data a longitudinal and a cross sectional data set are derived and delivered to researchers who request access. This report will focus mainly on the cross sectional microdata but most of the information contained in sections 2 and 3 also applies to the longitudinal data set.

The EU-SILC UDB is delivered to researchers as comma separated values (csv). While most statistical packages can read data in .csv format the data delivered by Eurostat include neither variable nor value labels. The German Microdata Lab (GML) at GESIS provides setup files which can read .csv data into SPSS or Stata and attach variable and value labels[2]. All of the examples presented in this report were tested on the data imported using these tools.

This report will be structured as follows. Section 2 provides an overview of the structure of the EU-SILC UDB, highlights the different data files, their relationship to each other and provides some instructions on how to best combine these files for analysis including sample syntax for SPSS and Stata.[3] Section 3 explores the relationship between households and individuals in the EU-SILC, highlighting the different identifiers which can be employed to identify family relationships within households. It also provides a number of hands on examples on how to aggregate information to the household level and how to combine families in one data file. These topics are covered as they highlight specifics of the EU-SILC data and should provide valuable information to most data users.

---

[1] A few countries have longer panel durations

[2] The GESIS setup files for EU-SILC are available at http://www.gesis.org/missy/eu/setups/EU-SILC

[3] These files are available at http://www.gesis.org/en/missy/materials/EU-SILC/tools/datahandling

Figure 1: Countries included in the EU-SILC User Database

2 Structure of EU-SILC

The data delivered to researchers in the EU-SILC UDB includes both longitudinal and cross-sectional data files. The cross sectional data files (file names start with the letters UDB_c) include only data collected for the indicated year while the longitudinal files (UDB_l) include data for all households which were included in the indicated year and at least one previous round. The longitudinal file includes data for the current and all previous survey rounds in which a household was included (for most countries this means the current and up to 3 previous rounds).

Both the cross-sectional and longitudinal data are delivered as four separate files, these files are identified by a letter after the year identifier (e.g. the file UDB_c10d… indicates the 2010 cross sectional household register). Here is a short summary of these different data files and their contents:

- Household Register (d-file): Includes information on weights, sampling, regional identifiers and degree of urbanization. Contains only information at the household level. This is by far the smallest of the four files.
- Household Data (h-file): This file contains information on the interview, household income, subjective economic situation, household level poverty and employment indicators as well as information on household assets and housing. All variables refer to the household level.
- Personal Register (r-files): This file is special as it is the only file that includes information on persons under sixteen years of age. It mainly includes identifiers which can be used to analyze family relations, basic demographic information and variables on childcare usage. All variables refer to individuals.
- Personal Data (p-files): This file contains the largest number of variables all of which refer to individuals. However no information is contained on persons under sixteen years of age. It includes information on demographics, income, work, unemployment, health, nationality, migration, and work intensity as well as person weights, identifiers and information on the interview.

Data in the EU-SILC are structured hierarchically with individuals' nested in households. In the household level files one observation corresponds to a household. Households can be distinguished with the help of the household ID. In the personal data files one observation corresponds to an individual. Each individual is assigned a personal ID as well as household ID for the household it lives in. While the personal ID and the household ID are unique within countries the same ID can be assigned in different countries which means that the identifiers on their own do not distinctly identify cases within a data file. Thus, whenever one is conducting comparisons between countries it is mandatory to use the information on country and year in combination with the respective identifiers in order to ensure that identifiers distinctly identify cases. Another issue to consider is that each of the four data files detailed above assigns different names to the identifiers, thus a first step of any analysis with the EU-SILC should be to harmonize the identifier variables. In this report and the adjoining syntax documents the following abbreviations will be used `country`, `year`, `hh_id` and `person_id` (see Figure 2 below for a schematic representation).

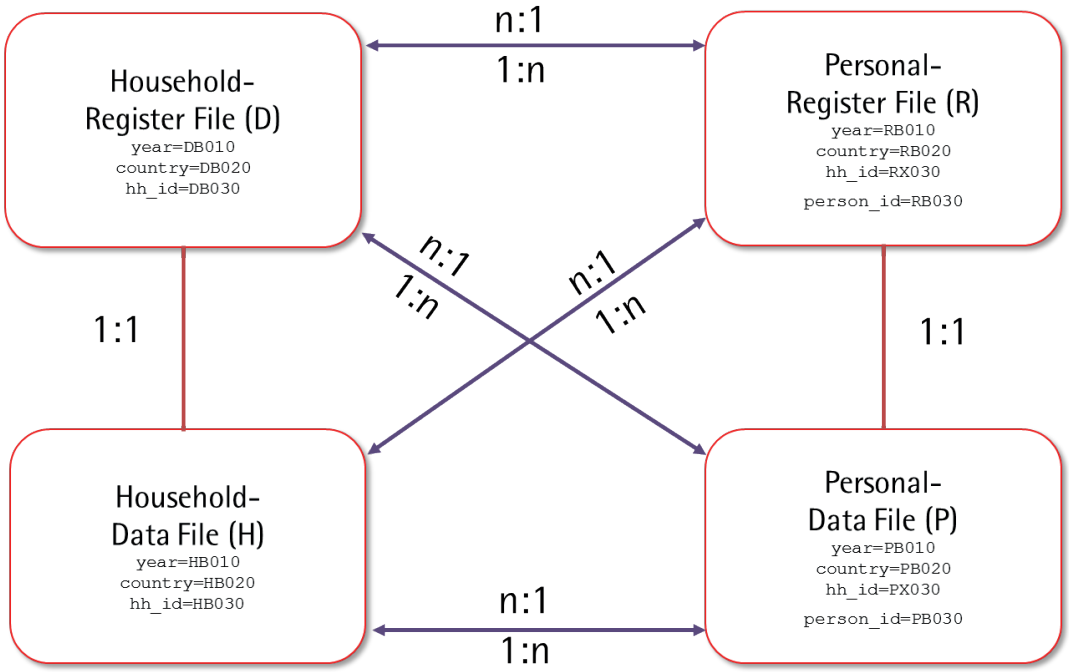

Figure 2: Identifiers in the EU-SILC data files (Source: Heike Wirth)

The example below illustrates how to add information from the household data (in our example the tenure status variable `HX070`) to the personal data files using the `hh_id`. This results in a n:1 merge as there can be multiple persons living in one household. In a first step the household data file is opened and the identifier variables are renamed. Note that in all examples abbreviations of the actual file names are used. When using the code presented here you will have to modify the reference to file names accordingly. In the example, we are interested in the tenure status (`HX070`) and thus, we drop all other variables (save for the identifiers) and save the dataset under a new file name (`tenurestatus.dta`). Before saving this file we have to sort it based on the variables which will be used for merging. Then we open the personal data file and likewise rename the identifiers before merging. As detailed above the combination of `country`, `year` and `hh_id` is used for the merge in order to ensure distinct identification of households.

Stata Code:

```
***Change path to working directory and open household register ***
cd "C:\SILC\"
use UDB_c10d.dta, clear

***Rename ID variables, drop unneeded variables, sort and save***
rename HB010 year
rename HB020 country
rename HB030 hh_id
```

```
keep year country hh_id HX070
sort by year country hh_id
save tenurestatus.dta, replace

***Open personal data and rename identifiers***
***Identifiers are renamed so that they are identical in both data
files to be merged***
use UDB_c10p.dta, clear

rename PB010 year
rename PB020 country
rename PX030 hh_id
rename PB030 pers_id
sort year country hh_id

***Merge tenurestatus to personal data file***
***Save as new file***
merge m:1 year country hh_id using tenurestatus.dta
tab _merge
keep if _merge==3
drop _merge
save personHH.dta, replace
```

SPSS Syntax:

```
***Change path to working directory and open household register.
FILE HANDLE data_path / NAME='C:\SILC\'.
GET FILE='data_path/UDB_c10d.sav'.

***Open personal data and rename identifiers.
***Identifiers are renamed so that they are identical in both data
files to be merged.

rename variables (HB010 HB020 HB030 = year country hh_id).
autorecode country /into country_num.
```

```
execute.
sort cases by year country_num hh_id.
save outfile = 'data_path/ tenurestatus.sav '
/keep year country_num hh_id HX070.

***Open personal data and rename identifiers.
GET FILE='data_path/UDB_c10p.sav'.
rename variables (PB010 PB020 PX030 PB030 = year country hh_id
pers_id).

***Merge tenurestatus to personal data file.
***Save as new file.
sort cases by year country hh_id.
MATCH FILES FILE= *
  /file ='data_path/DB100.sav'
  /BY year country_num hh_id.
  execute.

save outfile = 'data_path/personHH.sav '.
```

3 Households in the EU-SILC

The EU-SILC household data file contains extensive information about characteristics of the household on income, housing or the subjective economic position. The example presented in section 2 described how to add such household characteristics to the individual level data. However if you are interested in household level information not included in the household files such as the number of children living in the household or the equivalence weight according to the old OECD scale you will have to generate such variables yourself from the individual level data. Remember that whenever you wish to generate variables which refer to children you should always employ the personal register as information on children under 16 years of age is not included in the personal data file.

Example 1: Aggregating individual level information to the household level

year	country	hh_id	pers_id	RX010	child	childc	hhrank
2010	AT	1	11	47	0	0	1
2010	AT	1	12	45	0	0	2
2010	AT	2	21	34	0	3	1
2010	AT	2	22	9	1	3	2
2010	AT	2	23	5	1	3	3
2010	AT	2	24	3	1	3	4
2010	BE	1	11	48	0	2	1
2010	BE	1	12	25	0	2	2
2010	BE	1	13	24	0	2	3
2010	BE	1	14	4	1	2	4

Figure 3: Counting children in EU-SILC (fictitious data)

In order to illustrate how household level variables can be generated from individual data the following example calculates the number of children in the household. In a first step a dummy variable needs to be generated which identifies whether a person is under 18 years of age (`child`). In a second step the number of children in each household is counted and written into a new variable (`childc`). This step requires a command which can aggregate individual level information to the household level. In Stata this can be done with the help of the `egen` command, in SPSS the `aggregate` command is employed. Figure 3 illustrates a fictitious personal register file which includes 2 households from Austria and one from Belgium. The figure also demonstrates why it is so important to not only use identifiers but also the year and country information as there exist a number of overlaps in the `hh_id` and `pers_id` between Austria and Belgium. Thus the `aggregate` and `egen` commands refer to the combination of `year`, `country` and `hh_id`.

When analyzing the `childc` variable remember that you have generated a household level variable at the individual level. When examining it descriptively you either need to merge it to a household level data file or generate a variable which ensures that each household is counted only once. This is achieved in the example below by generating the variable `hhrank` and only selecting individuals with `hhrank`=1 for the descriptive analysis.

Stata Code:

```
***Open personal register file and generate uniform identifiers***
use UDB_c10r.dta, clear
rename RB010 year
rename RB020 country
rename RX030 hh_id
rename RB030 pers_id

***Generate a dummy variable identifying children under 18***
gen child=.
recode child .=1 if RX010<18

***Count the number of children for each household***
egen childc =count(child) , by(year country hh_id)
label var childc "Number of children under 18 in HH"

***generate hhrank variable for descriptives***
egen hhrank=rank(RX010), by(year country hh_id) unique
tab childc if hhrank==1
```

SPSS Syntax:

```
***Open personal register file and generate uniform identifiers.
use UDB_c10r.dta, clear
rename RB010 year
rename RB020 country
rename RX030 hh_id
rename RB030 pers_id
autorecode country /into country_num.
execute.

***Count the number of children for each household.
compute child=0.
if RX010<18 child=1.
```

```
aggregate outfile=* mode=add
 /break=year country_num hh_id
 / childc=SUM(child).
var lab childc "Number of children under 18 in HH".

***generate hhrank variable for descriptives.
rank var=pers_id by year country_num hh_id
 /rank into hhrank.
filter by hhrank.
fre childc.
cross country by childc /cel cou row.
filter off.
```

Example 2: Utilizing pointer variables to identify family relations

In Example 1 we calculated the number of children living in a household. However, in some cases you might be interested in more precise information on family relations. In such cases it is not sufficient to merely look at the household context. One must also consider family relations. To this aim the EU-SILC contains a number of so called pointer variables with which relationships between household members can be identified. This includes the father ID, mother ID and the spouse/partner ID. To illustrate how to work with such pointer variables in the EU-SILC we will recur to the example dataset employed above.

year	country	hh_id	pers_id	hhrank	RB220	RB230	RB240
2010	AT	1	11	1	.	.	12
2010	AT	1	12	2	.	.	11
2010	AT	2	21	1	.	.	.
2010	AT	2	22	2	.	21	.
2010	AT	2	23	3	.	21	.
2010	AT	2	24	4	.	21	.
2010	BE	1	11	1	.	.	.
2010	BE	1	12	2	.	11	13
2010	BE	1	13	3	.	.	12
2010	BE	1	14	4	13	12	.

Figure 4: Family relations in EU-SILC (fictitious data)

Figure 4 includes the three pointer variables RB220, RB230 and RB240. These variables include the personal ID of the mother/father/spouse. Adding theses to the above dataset provides us with more detailed information on the household. Household 1 in Austria is a couple household with no children. Household 2 in Austria is a single mother household with three children while our Belgian household is a multigenerational household. Person 11 is the mother of person 12 who lives together with her spouse (person 13) and their child (person 14). From this information we can also deduce that person

14 is the grandchild of person 11. In the below example we will calculate the number of own children living in the household. The syntax required to do so is a bit more complex and the different steps will be illustrated graphically.

Step 1

For each person in the household generate a new variable (`mtemp`) which includes the personal id of the mother. This is done with a loop command which runs over the household number. In the example dataset the largest household size is four thus only 4 variables are generated however, most EU-SILC rounds include households with 20 or more individuals[4]. This is merely an intermediate step however as the `mtemp` variable needs to be aggregated to the household level.

year	country	hh_id	pers_id	hhrank	RB220	RB230	mtemp1	mtemp2	mtemp3	mtemp4
2010	AT	1	11	1	.	.	.	.	.	.
2010	AT	1	12	2	.	.	.	.	.	.
2010	AT	2	21	1	.	.	.	.	.	.
2010	AT	2	22	2	.	21	.	21	.	.
2010	AT	2	23	3	.	21	.	.	21	.
2010	AT	2	24	4	.	21	.	.	.	21
2010	BE	1	11	1	.	.	.	.	.	.
2010	BE	1	12	2	.	11	.	11	.	.
2010	BE	1	13	3	.	.	.	.	.	.
2010	BE	1	14	4	13	12	.	.	.	12

*Figure 5: Generate variables with mother ID for each member of the household ((*fictitious *data))*

Step 2

In this step the `mtemp` variables are aggregated to the household level similarly to the last example. The new household level variables `mpers1` through `mpers4` are generated. These variables allow for counting each individual's children. As the `mtemp` variables are no longer needed they can be deleted.

Step 3

In the final step the number of own children is counted for each person in the household. This is achieved by comparing the `person_id` with each `mpers` variable. If they are identical the value of the `kidcount` variable is increased by 1. The `foreach` command in Stata and the `loop` command in SPSS repeat this operation for each person in the household. In order to simplify the example the `kidcount` variable was only calculated for mothers. The syntax included in the Appendix also calculates it for men on the basis of the father id. Additionally the Appendix also includes Syntax which calculates the age of the youngest child for each person.

[4] Thus when running the code provided below with real data you will have to adjust the number of iterations in the foreach command in Stata or the vector/loop/aggregate commands in SPSS.

year	country	hh_id	pers_id	hhrank	RB220	RB230	mpers1	mpers2	mpers3	mpers4	kidcount
2010	AT	1	11	1	.	.	.	.	.	.	0
2010	AT	1	12	2	.	.	.	.	.	.	0
2010	AT	2	21	1	.	.	.	21	21	21	3
2010	AT	2	22	2	.	21	.	21	21	21	0
2010	AT	2	23	3	.	21	.	21	21	21	0
2010	AT	2	24	4	.	21	.	21	21	21	0
2010	BE	1	11	1	.	.	.	11	.	12	1
2010	BE	1	12	2	.	11	.	11	.	12	1
2010	BE	1	13	3	.	.	.	11	.	12	0
2010	BE	1	14	4	13	12	.	11	.	12	0

*Figure 6: Aggregating to household level and counting number of children (*fictitious *data)*

Stata Code:

```
***Tab the hhrank to define the upper bound in foreach***
tab hhrank

***Generate kidcount as described above***
gen kidcount=0
foreach i of numlist 1/4{
gen long mtemp`i'=RB230 if hhrank==`i' //Step1
egen long mompers`i'=mean(mtemp`i'), by(year country hh_id)//Step2
replace kidcount=kidcount+1 if mompers`i'==pers_id//Step3
}

***Drop temporary variables and label kidcount***
drop mtemp* dtemp*
label var kidcount "Number of own children in HH"
tab kidcount
```

SPSS Syntax:

```
***Tab the hhrank to define the upper bound in loop.
fre hhrank.

***Step 1.
VECTOR mtemp(4).
LOOP #i=1 TO 4.
```

```
   IF hhrank=#i  mtemp(#i) =RB230.
END LOOP.
EXECUTE.

***Step 2.
aggregate outfile=* mode=add
 /break= year country hh_id
 /mompers1 TO mompers4=MEAN(mtemp1 TO mtemp4)
EXECUTE.
DELETE VARIABLES mtemp1 TO mtemp4.

***Step 3.
COMPUTE kidcount=0.
EXECUTE.
VECTOR mompers= mompers1 TO mompers4.
LOOP #i=1 TO 4.
      IF mompers(#i)=pers_id kidcount=kidcount+1.
END LOOP.
EXECUTE.
label var kidcount "Number of own children in HH".
FRE kidcount.
```

Example 3: Generating a partner data file

The pointer variables can not only be employed to aggregate information, but can also be used for merging. The simplest application is generating a partner file which includes information for couples. Assume we are interested in examining differences in regards to educational assortative mating between married and cohabiting couples. The personal data file includes the requisite variables Marital Status (PB190), Highest ISCED level attained (PE040) and age (PX020). The syntax below utilizes the personHH dataset which was generated above.

In a first step the personal data file is opened and the variables mentioned above as well as the identifiers are selected. From this file partnered women are selected and saved into a new file (personal_woman). Additionally variable names have to be recoded. In order to be able to merge partnered women to their partners all variables that will be merged (this excludes the identifiers) need to have names that are distinct from variable names in the original data. Thus PB190, PE040 and PX020 were renamed into PB190_w, PE040_w and PX020_w. The names of the identifier variables remain unchanged with one exception pers_id is renamed to PB180. This allows merging of the personal_woman file to the personHH file. Once the personal_woman file is saved the per-

sonHH file can be reopend[5] and data can be merged. The combination of country year hh_id and PB180 is used to merge.

Stata Code:

```
***Generate a file which includes only partnered women***
***All variables in this file are renamed "varname_w"***
use personHH.dta, clear

keep if PB150==2 & PB180!=.
keep  country year hh_id pers_id PX020 PE040 PB190
rename pers_id PB180
rename PX020 PX020_w
rename PE040 PE040_w
rename PB190 PB190_w
save personal_woman.dta, replace

***Reopen personal data file***
***Only select partnered indivduals***
use personHH.dta, clear
keep if PB180!=.

***Merge the file with the separate file of partnered women***
merge 1:1 year country hh_id PB180 using personal_woman.dta
tab _merge
keep if _merge==3
drop _merge
save partner.dta, replace
```

SPSS Syntax:

```
***Generate a partner file which includes only partnered women.
***All variables in this file are renamed "varname_w".
GET FILE='data_path/personHH.sav'.
```

[5] This step is not required with SPSS as personal_woman.sav was written out from the original file and all selections made were only temporary.

```
SORT CASES BY year country pers_id.
TEMPORARY.
SELECT IF PB150=2 AND NOT(SYSMIS(PB180)).
SAVE OUTFILE='data_path/personal_women.sav'
   / KEEP=country year hh_id pers_id PX020 PE040 PB190
   /RENAME(pers_id PX020 PE040 PB190=PB180 PX020_w PE040_w PB190_w).

***Only select partnered indivduals.
***Merge the file with the separate file of partnered women.
SELECT IF NOT (SYSMIS(PB180)).
EXECUTE.
SORT CASES BY year country hh_id PB180.
MATCH FILES / FILE=*
   / FILE='data_path//personal_women.sav'
   / BY year country hh_id PB180.
EXECUTE.
save outfile = 'data_path/partner.sav '.
```

4 References

Commission Regulation No 28/2004. Available at: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32004R0028&from=EN

Eurostat (2013). Methodological Guidelines and Description of EU-SILC Target Variables. 2014 operation (Version September 2013).

Eurostat (2015). Access to Microdata. EUROPEAN UNION STATISTICS ON INCOME AND LIVING CONDITIONS (EU-SILC) [Online]. Available at: http://ec.europa.eu/eurostat/web/microdata/european-union-statistics-on-income-and-living-conditions

Eurostat (2016). EU statistics on income and living conditions (EU-SILC) methodology – data collection [Online]. Available at: http://ec.europa.eu/eurostat/statistics-explained/index.php/EU_statistics_on_income_and_living_conditions_(EU-SILC)_methodology_-_data_collection

Jäntti, M., Törmälehto, V.-M., & Marlier, E. (Eds) (2013). The use of registers in the context of EU–SILC: challenges and opportunities 2013 edition. Eurostat statistical working papers.