|  |  |  |
| --- | --- | --- |
| logo_ec_17_colors_300dpi | EUROPEAN COMMISSION  EUROSTAT  Directorate F: Social statistics  **Unit F-4: Income and living conditions; Quality of life** |  |

|  |
| --- |
| **EU-SILC Validation Software**  **USER GUIDE – R-Studio users** |

Table of contents

[Introduction 3](#_Toc212136083)

[1. Preliminary information 4](#_Toc212136084)

[1.1 Methodological references 4](#_Toc212136085)

[1.2 SILC operation years 4](#_Toc212136086)

[1.3 SILC Data Files 4](#_Toc212136087)

[1.4 Useful operational notations 5](#_Toc212136088)

[2 Running the pre-validation in R-Studio 5](#_Toc212136089)

[2.1 HTML report 6](#_Toc212136090)

[2.2 The results in R-Studio 7](#_Toc212136091)

[2.3 Results datasets 8](#_Toc212136092)

[ANNEX 1: The R-Studio pre-validation programs structure 11](#_Toc212136093)

[ANNEX 2: comparison of checks implemented in each process 12](#_Toc212136094)

# Introduction

Following the entry into force of IESS [Regulation (EU) 2019/1700 establishing a common framework for European statistics relating to persons and households, based on data at individual level collected from samples](https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R1700&from=EN), in 2021 Eurostat has re-engineered the internal IT tool to validate the data of the European Union Survey on Income and living conditions (EU-SILC). This validation software is based on SAS technologies (SAS enterprise guide or SAS foundation).

However, as many NSIs do not or cannot use SAS, an alternative process, albeit less complete, has been developed to enable them to pre-validate their SILC data.

This process is **two-stepped**:

* A first part of the pre-validation checks is implemented in STRUVAL
* The remaining part of the pre-validation checks is implemented in R programs

This User Guide aims at providing EU-SILC data validators (internal Eurostat’s EU-SILC production team members and colleagues from the NSIs) with practical, operational information on the second step of the EU-SILC alternative pre-validation with R-Studio.

The document presents the following outline:

1. Information on methodological references, operation years, SILC data files and notation to be followed to properly execute the pre-validation.
2. Instruction on how to run the pre-validation in R-Studio.
3. Instruction on how to analyse and interpret the pre-validation results.
4. Annex 1 reports an overview of the R programs that perform the pre-validation process.
5. Annex 2 reports a comparison of the checks performed in the SAS programs with the checks performed when using R-Studio and STRUVAL.

Eurostat invites colleagues from NSIs to contact the EU-SILC production team ([ESTAT-ILC@ec.europa.eu](mailto:ESTAT-ILC@ec.europa.eu)) for any information needed on the presented pre-validation process.

# Preliminary information

# **Methodological references**

For the User’s information, the pre-validation process is based on a number of validation rules checking the compliance of SILC variables, values and flags reported by data providers in the SILC files with the reference variables/flags and relative modalities contained in an XML file. Such file is annually configured according to the [EU-SILC Methodological guidelines (DOC065)](https://circabc.europa.eu/ui/group/853b48e6-a00f-4d22-87db-c40bafd0161d/library/334d943f-6f71-4f4b-9c7e-a6767a3fe164?p=1&n=-1&sort=name_DESC).

# **SILC operation years**

As SILC provides cross-sectional and longitudinal information[[1]](#footnote-1), the process is configured to refer to the proper validation rules for each operation year.

In other terms, if a country in December 2025 provides a 4-year panel SILC data, its data file will report information for the last 4 years (2022, 2023, 2024 and 2025 – or back to 2020 in case of 6 year panel).

# **SILC Data Files**

NSIs send EU-SILC data in four separate files:

* Household register (D)
* Personal register (R)
* Household data (H)
* Personal data (P)

The household register file (D) contains every household selected, including those where the address could not be contacted and households that could not be interviewed.

In the other files, records associated with a household will exist only if the household has been contacted and has a completed household interview in the household data file (H) and at least one member has complete data in the personal data file (P). This member must be the selected respondent if this mode of selection is used.

The personal register file (R) contains a record for every person currently living in the household or temporarily absent. As regards the longitudinal component, it must also contain a record for every person registered in the previous year’s R-file.

Finally, the personal data file (P) contains a record for every eligible person for whom the information could be completed from interview and/or registers.

# **Useful operational notations**

As mentioned in the installation guide, **&eusilc/main/&cc/csv** is the inbound directory where the SILC data files to be processed should be stored. They have to comply with the following naming rules:

* + SILC\_**R&F**\_A\_**&cc**\_**&yyyy**\_0000\_V**&FVER**.csv

Where “&FVER” is a 4 digit code. For example, the SILC 2025 D file version 1 for Austria is labelled as follows: SILC\_RD\_A\_AT\_2025\_0000\_V0001.csv.

Please bear in mind that all SILC data files (D, R, H, P) to be validated must have the same version number.

# Running the pre-validation in R-Studio

Open **DATACHECKSQL.Rmd** with R-Studio.

Figure 1 shows the execution parameters to set up before running the pre-validation.

**Figure 1: EU-SILC prevalidation main program in R-Studio**

A screenshot of a computer

AI-generated content may be incorrect.

Five parameters are required:

* The year of the survey: **Year: 2025**
* The country (identified by its bigram): **Country: "SE"**
* The version of the csv files: **File\_version: 1**
* The actual location of the SILC ROOT DIRECTORY on your server: **WorkingDrive: "V:/"**
* The path to which the report should be saved: **knitr::opts\_knit$set(root.dir = "V:/5.3\_Validation/R")**

After running the entire program, it will automatically generate tables in a SQLlite database containing the results.

To generate a report with the results, select option “Knit” and then “Knit to HTML”, as shown in Figure 2.

**Figure 2: Knit menu in R-Studio**

A screenshot of a computer

AI-generated content may be incorrect.

The report will be generated in the directory defined in the parameters.

# **HTML report**

The first section of the report contains a summary of all checks and the number of records found for each check ID, as shown in Figure 3.

**Figure 3: HTML validation report – Summary**

A screenshot of a computer

AI-generated content may be incorrect.

The following sections contain the results for the different types of structural checks and for the logical checks. For each check, the first 5 observations are shown in the report. To see all observations, the contents of the detailed results datasets should be extracted as described in section 2.2. Results datasets.

**Figure 4: HTML validation report – detail. Example for logical validation**

A screenshot of a computer

AI-generated content may be incorrect.

# **The results in R-Studio**

Once launched, the program will automatically generate tables in a SQLlite database with the different results. These tables are then exported as R datasets. This export is not exhaustive as explained later on in paragraph 2.3 and Annex 2.

**Figure 5: R-datasets created by the pre-validation program**

A screenshot of a computer

AI-generated content may be incorrect.

The datasets produced by the R pre-validation can be classified as follows:

|  |  |  |
| --- | --- | --- |
| ***Category*** | ***Name*** | ***Content*** |
| Overview | SUMMARY\_SQL\_PREVALID | Counting per check and year |
| Detailed datasets | SVAL\_D\_VALVSFL | Value vs flag inconsistences in D file |
| SVAL\_H\_VALVSFL | Value vs flag inconsistences in H file |
| SVAL\_R\_VALVSFL | Value vs flag inconsistences in R file |
| SVAL\_P\_VALVSFL | Value vs flag inconsistences in P file |
| SVAL\_VAR\_NA2\_M | Expected -2 flags for variable VAR.  *User must add an instruction to create the dataset* |
| SVAL\_VAR\_NA2\_W | Unexpected -2 flags for variable VAR  *User must add an instruction to create the dataset* |
| SVAL\_VAR\_NA4\_M | Expected -4 flags for variable VAR  *User must add an instruction to create the dataset* |
| SVAL\_VAR\_NA4\_W | Unexpected -4 flags for variable VAR  *User must add an instruction to create the dataset* |
| SVAL\_P\_NA3\_M | Expected -3 flags for unselected respondents |
| SVAL\_P\_NA3\_W | Unexpected -3 flags for selected respondents |
| LVAL\_NNN | Logical check n° NNN.  *User must add an instruction to create the dataset* |

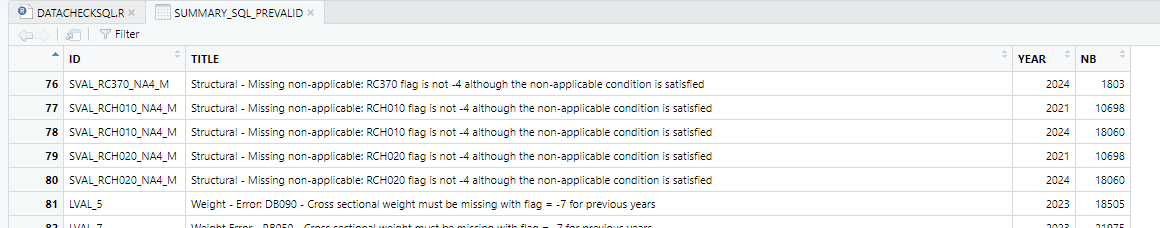
Section 2.3. Results datasets describes the main content of each dataset.

# **Results datasets**

* ***SUMMARY\_SQL\_PREVALID***

It provides a summary of the checks performed by the program.

**Figure 6: summary dataset**

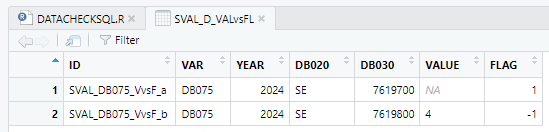


ID column contains the check identifier. This is also the dataset name for which detailed results can be found. The table also indicates, for each year, the count of the detected errors.

* **SVAL\_D\_VALVSFL**: test inconsistencies between values and respective flags in D file.
* **SVAL\_H\_VALVSFL**: test inconsistencies between values and respective flags in H file.
* **SVAL\_R\_VALVSFL**: test inconsistencies between values and respective flags in R file.
* **SVAL\_P\_VALVSFL**: test inconsistencies between values and respective flags in P file.

There is one detailed dataset for each file D, H, R and P. Figure 7 gives an example with D file.

**Figure 7: value vs flag dataset for D file**



This detailed dataset specifies the variable in error (column VAR), the observation ID (column DB030) and displays the value and flag that are not consistent.

* **SVAL\_VAR\_NA2\_M**: test expected -2 flag according to routing condition.
* **SVAL\_VAR\_NA2\_W**: test unexpected -2 flag when routing condition is not satisfied.
* **SVAL\_VAR\_NA4\_M**: test expected -4 flag according to routing condition.
* **SVAL\_VAR\_NA4\_W**: test unexpected -4 flag when routing condition is not satisfied.

There are two datasets for each variable that has a routing condition involving a -2 or -4 flag.

If it is needed to display the detailed results, new instructions need to be added at the end of the main pgm.

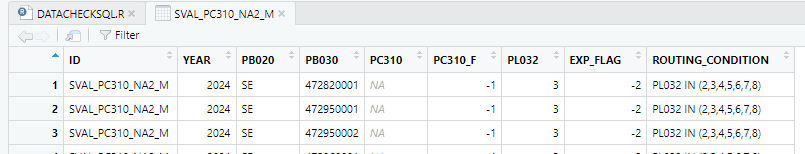
Figure 8 shows an example for flag -2 and variable PC310.

**Figure 8: examples of instructions to display -2 flag routing errors for variable PC310**



As Figure 9 shows, the detailed datasets specify the observations in error, the actual and expected flag and the routing condition.

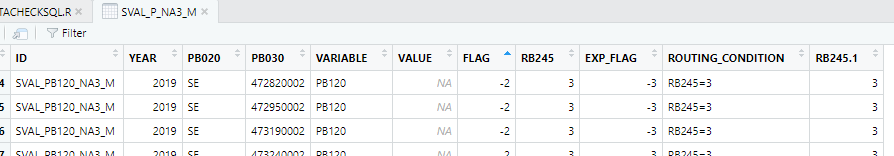
**Figure 9: dataset with -2 flag routing errors for variable PC310**



* **SVAL\_P\_NA3\_M**: test expected -3 flag for variables concerned by non-selected respondents.
* **SVAL\_P\_NA3\_W**: test unexpected -3 flag for selected respondents or variables non concerned by non-selected respondents.

The detailed results follow a similar logic as for the -2 and -4 flag errors. But contrary to -2 and -4 flags, there is only one routing condition for -3 flag (RB245=3) thus results are grouped in one single dataset.

**Figure 10: dataset with -3 flag non-selected respondents’ errors**



* **LVAL\_NNN**: test logical validation based on LVAL SQL queries.

There is one dataset for each logical check. If it is needed to display the detailed results, new instructions need to be added at the end of the main pgm.

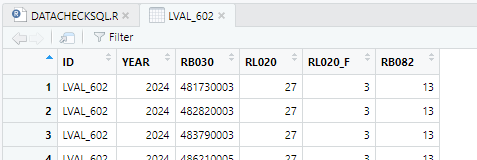
Figure 11 shows an example for logical check 602.

**Figure 11: example of instructions to display logical check 602 errors**



This dataset contains the result of logical checks carried out between SILC files and, for each detected error, informs on the year, the concerned variables and the affected observations, as shown in Figure 12.

**Figure 12: dataset with logical check 602 errors**



# ANNEX 1: The R-Studio pre-validation programs structure

Please consider this Annex as informative as it does not contain any indication of actions to be taken on User’s side.

For the User’s information, as said, the tool consists of a R main program calling several R sub-programs that treat and analyse different aspects of EU-SILC data. These subprograms are specific every year and their names are suffixed by the operation year YYYY:

* **DATACHECKSQL**: main program in which results are displayed.
* **YYYY/sql\_prevalid\_sval\_D\_valvsflag\_YYYY**: test inconsistencies between values and respective flags in D file.
* **YYYY/sql\_prevalid\_sval\_H\_valvsflag\_YYYY**: test inconsistencies between values and respective flags in H file.
* **YYYY/sql\_prevalid\_sval\_R\_valvsflag\_YYYY**: test inconsistencies between values and respective flags in R file.
* **YYYY/sql\_prevalid\_sval\_P\_valvsflag\_YYYY**: test inconsistencies between values and respective flags in P file.
* **YYYY/sql\_prevalid\_sval\_missing\_NA2\_YYYY**: test expected -2 flag according to routing condition.
* **YYYY/sql\_prevalid\_sval\_wrong\_NA2\_YYYY**: test unexpected -2 flag when routing condition is not satisfied.
* **YYYY/sql\_prevalid\_sval\_missing\_NA3\_YYYY**: test expected -3 flag for variables concerned by non-selected respondents.
* **YYYY/sql\_prevalid\_sval\_wrong\_NA3\_YYYY**: test unexpected -3 flag for selected respondents or variables non concerned by non-selected respondents.
* **YYYY/sql\_prevalid\_sval\_missing\_NA4\_YYYY**: test expected -4 flag according to routing condition.
* **YYYY/sql\_prevalid\_sval\_wrong\_NA4\_YYYY**: test unexpected -4 flag when routing condition is not satisfied.
* **YYYY/sql\_prevalid\_lval\_YYYY**: test logical validation based on LVAL SQL queries
* **YYYY/sql\_prevalid\_summary\_YYYY**: generates the summary table synthetizing the checks computed above.

# ANNEX 2: comparison of checks implemented in each process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Check** | | **SAS** | **STRUVAL** | **R-studio** |
| Expected variables | | **✓** | **✓** | **X** |
| Unexpected variables | | **✓** | **✓** | **X** |
| S  V  A  L | IDs uniqueness | **✓** | **✓** | **X** |
| Values | **✓** | **✓** | **X** |
| Flags | **✓** | **✓** | **X** |
| Imputation factors | **✓** | **!** | **X** |
| Value vs flag | **✓** | **X** | **✓** |
| Routing conditions (-2 and -4 flags) | **✓** | **X** | **✓** |
| Non-selected respondents (-3 flags) | **✓** | **X** | **✓** |
| Discontinued and created variables (-7 flags) | **✓** | **X** | **X** |
| Household grid | **✓** | **X** | **✓** |
| Logical checks | | **✓** | **X** | **✓** |
| Sample-size, Weights, Outliers, Counting, Comparison, Missing rates | | **✓** | **X** | **X** |
| Summary | | **✓** | **!** | **!** |

Legend

|  |
| --- |
| **✓ : checks fully implemented** |
| **! : checks partially implemented** |
| **X : checks not implemented** |

1. Cross-sectional data refer to a given time or a certain time period with variables on income, poverty, social exclusion and other living conditions. Longitudinal data refer to individual/household changes over time, observed periodically over a four-year period (or more years if a longer duration panel is used). [↑](#footnote-ref-1)