



SWINS

Sustainable Well-being
through INvestment
in Social Services

Strategies to integrate longitudinal household data to EUROMOD

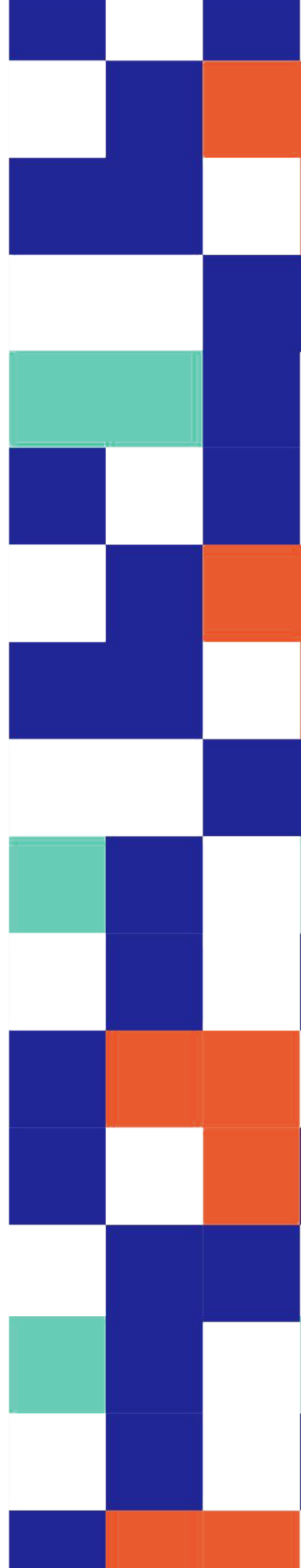
Technical notes for
merging longitudinal EU-
SILC 2022 with EUROMOD
2022 data files

January 2026



Funded by
the European Union

Funded by European Union's Horizon Europe
Programme under Grant Agreement No. 101177566





Authors

Réka Branyiczki - Researcher of the SWINS project, TARKI Social Research Institute

András Gábos - Researcher of the SWINS project, TARKI Social Research Institute

Peer Reviewers

Marko Vladislavljević, Faculty of Economics and Business (EKOF), University of Belgrade;
Nemanja Vuksanovic, Faculty of Economics and Business (EKOF), University of Belgrade; Jelena
Žarković, Faculty of Economics and Business (EKOF), University of Belgrade

Cite as

Branyiczki, R., Gábos, A. (2026), “*Strategies to integrate longitudinal household data to EUROMOD. Technical notes for merging longitudinal EU-SILC 2022 with EUROMOD 2022 data files*”

SWINS Working Paper n°4.1. SWINS – Sustainable Well-being through Investment in Social Services. ARCO, research centre at PIN Foundation.

Acknowledgements

The authors would like to thank the Fiscal Policy Analysis team at the Joint Research Centre (JRC) of the European Commission for the valuable input and all SWINS partners for their inputs provided in several SWINS meetings and communications.

Disclaimer

This Working Paper is part of Task 4.2, “Ways to integrate longitudinal household data into EUROMOD”. This task explores further possibilities of using longitudinal EU-SILC in combination with EUROMOD for the study of economic and social returns of investments in social services. The task will serve as an input for Tasks 6.1 and 6.4.

Funded by the European Union under Grant Agreement No.101177566. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Agency. Neither the European Union nor the granting authority can be held responsible for them.

Table of contents

1. Introduction	4
2. Description of the merged datasets.....	4
Attrition.....	6
Checking the matched individuals.....	11
3. Conclusion	13
Annex: Stata syntaxes	14
Preparation of the longitudinal EU-SILC file	14
Preparation of the EUROMOD data file	18
Merging longitudinal EU-SILC with EUROMOD	24



1. Introduction

Our aim in Task 4.2 of Work package 4 of the SWINS (Sustainable Well-Being Through INvestment in Social Services) project, to which this deliverable belongs, was to explore possibilities of linking longitudinal EU-SILC with EUROMOD input data files to extend opportunities to study economic and social returns of investments in social services. The benefit of the longitudinal EU-SILC is that its rotational panel design allows following individuals over a 4-year period in most countries, which enables us to model transitions. On the other hand, EUROMOD includes valuable information on benefits and taxes apart from the socio-economic variables. This report will serve as an input for Tasks 6.1 and 6.4 of SWINS. Furthermore, we hope to provide useful input for further research that aims to leverage the integration of the two data sources.

We identified two approaches for integrating longitudinal EU-SILC and EUROMOD input data sets.

- 1) Directly match individuals in the longitudinal EU-SILC and the EUROMOD input data, based on common identifiers.
- 2) Align the longitudinal EU-SILC data with the EUROMOD input data structure by generating the variables present in the EUROMOD data files in the longitudinal EU-SILC datasets, based on the variable definitions provided by the country-specific EUROMOD Documented Reference Dataset (DRD) files.

From the two approaches we preferred the first one, as in the case of direct matching we do not lose variables and the process is more transparent and less time-consuming. Furthermore, we may not be able to reconstruct all EUROMOD variables based on longitudinal EU-SILC. Accordingly, we investigated the possibility of direct linkage based on personal identifiers. We found that linking the cross-sectional and longitudinal EU-SILC files is possible from 2022 onwards for most EU countries, and accordingly EUROMOD data files may also be linked to longitudinal EU-SILC from 2022.

The rest of the document provides a technical description of merging longitudinal EU-SILC 2022 with EUROMOD 2022 data files. Section 2 includes descriptive statistics about the matched datasets. Section 3 briefly concludes. In an annex, we provide the Stata codes used for linking the datasets.

2. Description of the merged datasets

Linking longitudinal EU-SILC with cross-sectional EU-SILC and accordingly with EUROMOD data files based on common personal identifiers is possible since 2022 for most of the EU member states. To be able to include as many countries as possible, we link the longitudinal EU-SILC 2022 – version 2025, release 1 with EUROMOD input data, UDB (C22_release_23_09). We list the EUROMOD input data files used in the linking process in Table 1.

Overall, the matched dataset (called “LongiSILC_Euromod_2022.dta”) includes 325 897 matched individuals from 23 EU countries (AT, BE, BG, CY, CZ, DK, EE, EL, ES, FI, FR, HR, IE, IT, LT, LU, LV, NL, PT, RO, SE, SI, SK). Germany (DE), Hungary¹ (HU), and Poland (PL) are not included in

¹ Hungarian microdata from 2019 to 2024 have been removed from release 1 in 2025 of EU-SILC UDB. A methodological revision is planned in June 2025 for Hungarian EU-SILC data that will concern 2019 to 2024 data collections. The income and living conditions (ILC) indicators have already been updated after the

the panel EU-SILC 2022 (in release 1 of 2025). EUROMOD input data files are not available either for Hungary (HU) and Poland (PL) in 2022. While Malta (MT) is present both in the panel EU-SILC and the EUROMOD dataset, the merger is not possible based on personal identifiers.

Table 1 EUROMOD input data file for each country

Nr.	Country	EM input data name	The basis of the EM data (original data)
1	AT	AT_2022_b1_2015_03_n2	EU-SILC UDB (C22_release_23_09)
2	BE	BE_2022_c1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
3	BG	BG_2022_c1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
4	CY	CY_2022_b2_2015_03_e2	EU-SILC UDB (C22_release_23_09)
5	CZ	CZ_2022_b1_2015_03_e2	EU-SILC UDB (C21_release_23_09)
6	DK	DK_2022_c1_2015_03_e2	EU-SILC DK_EMSD2_2022-rev140224
7	EE	EE_2022_f1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
8	EL	EL_2022_c2_2015_03_e2	EU-SILC 2022; -
9	ES	ES_2022_b2_2015_03_e2	EU-SILC UDB (C22_release_23_09)
10	FI	FI_2022_b1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
11	FR	FR_2022_c1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
12	HR	HR_2022_b2_2015_03_e2	EU-SILC UDB (C22_release_23_09)
13	IE	IE_2022_b1_2015_03_e2	EU-SILC IE_EMSD2_2022
14	IT	IT_2022_b2	EMSD, 2022 released in 2023
15	LT	LT_2022_c1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
16	LU	LU_2022_b1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
17	LV	LV_2022_b2_2015_03_e2	EU-SILC UDB (C22_release_23_09)
18	NL	NL_2022_b1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
19	PT	PT_2022_b2_2015_03_e2	EU-SILC UDB (C22_release_23_09)
20	RO	RO_2022_b1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
21	SE	SE_2022_b1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
22	SI	SI_2022_c1_2015_03_e2	EU-SILC UDB (C22_release_23_09)
23	SK	SK_2022_b1_2015_03_e2	EU-SILC UDB (C22_release_23_09)

Notes. Names of the EUROMOD input data files used in the linking process and the exact EU-SILC release, which served as a basis for the EUROMOD file.

methodological revision of the Hungarian EU-SILC data (2019 to 2024 data collections), and the HU microdata are part of release 2 since December 2025, but they have not yet been used for the EUROMOD input datafile.



Attrition

When focusing on the 23 countries where mergers are possible in 2022, we see that overall, approximately 61% of the individuals were matched, and around 30% of them are present only in EUROMOD, and 9% of them are present only in the panel EU-SILC (see Table 2). The high share of unmatched individuals may be present due to the differences in cross-sectional and longitudinal EU-SILC files. We provide the number of matched individuals by country as well (see Table 3).

Table 2 Overall number and percentage of individuals matched and unmatched

	Frequency	Percent
Only in panel EU-SILC	47 295	8.92
Only in EM	156 982	29.61
Matched individuals	325 897	61.47
Total	530 174	100.00

Notes. Number and percentage of individuals matched and unmatched from each dataset (longitudinal EU-SILC 2022 – version 2025, release 1 and EUROMOD input data, UDB (C22_release_23_09) for AT, BE, BG, CY, CZ, DK, EE, EL, ES, FI, FR, HR, IE, IT, LT, LU, LV, NL, PT, RO, SE, SI, SK).

Table 3 Number and percentage of individuals matched and unmatched by country.

Nr.	Country	Nr. of matched individuals	Only in panel EU-SILC	Only in EUROMOD	Total	Attrition from the panel EU-SILC sample	Attrition from the EM sample
1	AT	8 368	964	3701	13033	10%	31%
2	BE	12 356	1959	2639	16954	14%	18%
3	BG	13 493	1032	3926	18451	7%	23%
4	CY	7 268	562	3082	10912	7%	30%
5	CZ	13 245	815	4906	18966	6%	27%
6	DK	8 373	2231	4871	15475	21%	37%
7	EE	9 321	1554	3656	14531	14%	28%
8	EL	16 281	689	5861	22831	4%	26%
9	ES	40 536	4494	19136	64166	10%	32%
10	FI	14 972	1607	6594	23173	10%	31%
11	FR	25 386	2814	13457	41657	10%	35%
12	HR	13 344	1165	5713	20222	8%	30%
13	IE	7 150	2062	4227	13439	22%	37%
14	IT	23 316	5470	21307	50093	19%	48%
15	LT	8 431	720	3667	12818	8%	30%
16	LU	4 834	1817	4204	10855	27%	47%
17	LV	8 111	1004	4236	13351	11%	34%
18	NL	24 863	4633	5466	34962	16%	18%
19	PT	18 563	2258	11539	32360	11%	38%
20	RO	12 344	191	4254	16789	2%	26%
21	SE	13 227	7243	7221	27691	35%	35%
22	SI	12 518	1730	9748	23996	12%	44%
23	SK	9 597	281	3571	13449	3%	27%
	Total	325 897	47 295	156 982	530 174	13%	33%

Notes. Number and percentage of individuals matched and unmatched from each dataset (longitudinal EU-SILC 2022 – version 2025, release 1 and EUROMOD input data, UDB (C22_release_23_09) for AT, BE, BG, CY, CZ, DK, EE, EL, ES, FI, FR, HR, IE, IT, LT, LU, LV, NL, PT, RO, SE, SI, SK).



We compare the matched and the unmatched individuals of both the panel EU-SILC samples and the EUROMOD samples to see whether the attrition is random or systemic. In case of random attrition, we may still consider the matched sample representative, however if the attrition is systemic, then statistical corrections, such as reweighting the sample may be necessary before using the linked dataset for analysis.

We found that most unmatched individuals coming from the EU-SILC sample have only a household identifier and otherwise there are no observations for them in 2022, which may be due to the loss of participants over the waves of the longitudinal EU-SILC.

In the case of matched and unmatched individuals of the EUROMOD samples, we compared the shares of males, the average age, the average equivalised disposable income and the average years spent in education in the two groups to see whether attrition from the original EUROMOD data file is random. We report these descriptive statistics together with a t-test in Table 3 and 4. The share of males is similar in the matched and unmatched samples in all countries. The average age was also similar in many countries, however in some cases matched individuals tend to be older. The difference ranges between less than a year and more than five years (in Ireland), but in most countries the difference is less than two years (see Table 4).

Table 4 Share of males and average age in the matched and unmatched EM samples by country.

				Share of males		Mean age	
Nr.	Country	Only EM	Matched	t	Only EM	Matched	t
1	AT	0.47	0.48	-0.75	43.04	45.05	-4.43
		0.50	0.50		23.00	22.97	
2	BE	0.49	0.49	0.15	39.08	42.66	-7.18
		0.50	0.50		22.89	23.38	
3	BG	0.47	0.47	-0.29	49.11	49.47	-0.88
		0.50	0.50		22.41	22.38	
4	CY	0.48	0.48	0.58	42.98	44.19	-2.42
		0.50	0.50		23.22	23.17	
5	CZ	0.47	0.47	-0.15	47.13	47.40	-0.67
		0.50	0.50		24.38	24.05	
6	DK	0.49	0.48	0.84	44.79	49.48	-10.97
		0.50	0.50		23.89	23.62	
7	EE	0.47	0.46	0.09	42.22	43.49	-2.76
		0.50	0.50		23.30	23.64	
8	EL	0.48	0.48	-0.17	49.19	49.89	-1.97
		0.50	0.50		23.23	23.40	
9	ES	0.48	0.49	-0.48	43.07	43.81	-3.75

		<i>0.50</i>	<i>0.50</i>		22.38	22.50	
10	FI	0.51	0.50	0.73	39.62	42.58	-8.73
		<i>0.50</i>	<i>0.50</i>		22.53	23.14	
11	FR	0.48	0.48	-1.02	42.74	43.19	-1.77
		<i>0.50</i>	<i>0.50</i>		23.95	23.99	
12	HR	0.47	0.47	-0.23	49.77	49.68	0.27
		<i>0.50</i>	<i>0.50</i>		21.84	21.96	
13	IE	0.50	0.48	1.33	39.61	44.98	-11.81
		<i>0.50</i>	<i>0.50</i>		22.74	23.82	
14	IT	0.48	0.47	1.52	48.66	51.69	-14.10
		<i>0.50</i>	<i>0.50</i>		22.63	22.70	
15	LT	0.42	0.43	-1.04	48.26	48.75	-1.07
		<i>0.49</i>	<i>0.50</i>		23.70	22.92	
16	LU	0.51	0.51	-0.02	42.44	41.24	2.62
		<i>0.50</i>	<i>0.50</i>		21.71	21.66	
17	LV	0.45	0.45	0.55	44.29	44.83	-1.20
		<i>0.50</i>	<i>0.50</i>		23.49	23.34	
18	NL	0.50	0.49	0.24	44.17	47.72	-10.21
		<i>0.50</i>	<i>0.50</i>		23.21	23.25	
19	PT	0.47	0.47	-1.24	47.51	49.18	-6.38
		<i>0.50</i>	<i>0.50</i>		22.17	22.06	
20	RO	0.48	0.48	0.09	48.78	50.33	-4.12
		<i>0.50</i>	<i>0.50</i>		20.92	21.25	
21	SE	0.50	0.50	-0.48	40.73	40.89	-0.48
		<i>0.50</i>	<i>0.50</i>		23.55	23.63	
22	SI	0.50	0.49	0.87	42.11	46.59	-14.46
		<i>0.50</i>	<i>0.50</i>		22.56	23.23	
23	SK	0.45	0.46	-0.49	47.06	47.17	-0.25
		<i>0.50</i>	<i>0.50</i>		22.05	22.51	

Notes. We report the average, the standard deviation in italics and the t value of a two-sample t-test with equal variances. Age is measured in years.



When it comes to mean disposable equivalised monthly income, there is a significant difference between the matched and unmatched EUROMOD samples in fifteen countries; the difference is above 200 euros only in four countries and its sign differs across countries. Ireland is an outlier with a large, more than 1000-euro difference. We find a significant difference in the average number of years spent in education between matched and unmatched individuals in 11 countries, though the differences tend to be small. The highest difference is one year, in Luxembourg.

Table 5 Mean disposable income and mean years of education in the matched and unmatched EM samples by country.

Nr.	Country			Mean income		Mean years of education	
		Only EM	Matched	t	Only EM	Matched	t
1	AT	2605.51	2766.04	-4.33	10.65	11.03	-3.94
		1695.18	1953.56		4.95	4.83	
2	BE	2393.59	2430.02	-1.20	10.28	10.60	-2.74
		1358.78	1433.71		5.70	5.39	
3	BG	1220.50	1019.06	11.51	10.59	10.53	0.71
		1280.03	851.91		4.94	4.86	
4	CY	1699.03	1743.21	-1.65	9.62	9.71	-0.84
		1345.95	1199.58		4.81	4.70	
5	CZ	28178.39	27786.41	1.66	11.92	11.92	0.07
		218.86	118.50		5.09	5.03	
6	DK	23251.70	23790.64	-1.89	13.04	13.43	-3.60
		205.08	181.22		6.12	5.88	
7	EE	1339.51	1399.58	-3.81	10.33	10.39	-0.60
		798.10	810.34		5.22	5.26	
8	EL	869.82	898.15	-2.91	9.82	9.73	1.20
		580.40	659.98		4.92	4.98	
9	ES	1627.40	1662.25	-3.90	9.79	10.06	-4.76
		1030.59	1013.40		6.49	6.44	
10	FI	2809.93	2932.55	-1.60	10.31	10.55	-3.16
		1835.15	6097.47		5.12	5.07	
11	FR	2187.28	2225.76	-2.59	12.88	12.93	-0.73
		1585.03	1281.37		6.35	6.38	
12	HR	5695.89	5604.33	1.75	11.16	11.14	0.26

		3400.74	3268.54		4.25	4.22	
13	IE	4223.76	2884.16	20.10	11.85	11.02	6.48
		4630.36	2470.30		7.12	6.39	
14	IT	1345.09	1391.19	-1.99	10.14	10.19	-1.07
		2778.68	2086.74		5.20	5.16	
15	LT	996.02	1031.62	-2.23	11.77	12.21	-3.91
		784.41	816.51		5.98	5.56	
16	LU	4211.06	4665.69	-8.53	10.75	11.79	-7.43
		2494.67	2558.03		6.79	6.51	
17	LV	983.63	994.86	-0.84	11.54	11.67	-1.23
		749.54	686.15		5.61	5.52	
18	NL	2516.36	2681.41	-7.08	11.77	12.71	-11.33
		1417.83	1591.39		5.94	5.46	
19	PT	1007.21	1104.81	-10.63	8.26	8.68	-7.83
		768.77	777.88		4.59	4.45	
20	RO	2774.71	2538.19	8.68	11.33	11.39	-0.86
		1579.79	1515.18		3.84	3.79	
21	SE	26725.21	27903.66	-4.27	10.85	10.94	-1.30
		239.10	156.87		4.78	4.86	
22	SI	1458.61	1590.03	-12.97	11.35	12.12	-10.60
		729.16	765.71		5.34	5.41	
23	SK	8788.73	8611.92	2.54	12.13	12.00	1.33
		3610.72	3524.02		4.58	4.70	

Notes. We report the average, the standard deviation in italics and the t value of a two-sample t-test with equal variances. Income refers to the equivalized disposable income, based on variable *ydses_o* in the EUROMOD data files, in most cases the income is in euros, however some countries report in their own currency (DK in DKK, HR in HRK, RO in RON, SE in SEK). Years of education is based on variable *dey* in the EUROMOD data files.

Checking the matched individuals

We check the matched sample for each country to see whether the merger based on personal ID is feasible. We check whether the age and the sex of the matched individuals are the same based on the panel EU-SILC variables and the EUROMOD variables that define age and sex for the individuals. Overall, we find that 2% of the matched individuals are “suspicious” as the age is different according to EU-SILC and EUROMOD. The same share is 0.1% when looking at the sex of the individuals. We flagged those individuals where the difference between age based on EU-SILC and EUROMOD was larger than 2.



Table 6. Number matched individuals with different age and sex based on EU-SILC and EUROMOD, by country.

Nr.	Country	Different age		Different sex		Different age or sex	
		Nr. of individuals	Share of individuals	Nr. of individuals	Share of individuals	Nr. of individuals	Share of individuals
1	AT	0	0.00	0	0.00	0	0.00
2	BE	152	0.01	133	0.01	168	0.01
3	BG	15	0.00	10	0.00	15	0.00
4	CY	420	0.06	46	0.01	426	0.06
5	CZ	744	0.06	2	0.00	744	0.06
6	DK	467	0.06	0	0.00	467	0.06
7	EE	68	0.01	60	0.01	76	0.01
8	EL	1485	0.09	10	0.00	1486	0.09
9	ES	0	0.00	0	0.00	0	0.00
10	FI	6	0.00	0	0.00	6	0.00
11	FR	1115	0.04	72	0.00	1123	0.04
12	HR	13	0.00	12	0.00	13	0.00
13	IE	0	0.00	0	0.00	0	0.00
14	IT	-	-	14	0.00	14	0.00
15	LT	0	0.00	0	0.00	0	0.00
16	LU	61	0.01	0	0.00	61	0.01
17	LV	2	0.00	2	0.00	2	0.00
18	NL	0	0.00	0	0.00	0	0.00
19	PT	109	0.01	102	0.01	126	0.01
20	RO	708	0.06	6	0.00	708	0.06
21	SE	0	0.00	8	0.00	8	0.00
22	SI	0	0.00	1	0.00	1	0.00
23	SK	11	0.00	8	0.00	11	0.00
	Total	5376	0.02	486	0.00	5455	0.02

Notes. Number and percentage of individuals from the matched dataset, where the age, sex or either age or sex is different according to the variable defined by EUROMOD and EU-SILC 2022 (longitudinal EU-SILC 2022 – version 2025, release 1 and EUROMOD input data, UDB (C22_release_23_09) for AT, BE, BG, CY, CZ, DK, EE, EL, ES, FI, FR, HR, IE, IT, LT, LU, LV, NL, PT, RO, SE, SI, SK).

3. Conclusion

Overall, we found that matching longitudinal EU-SILC and EUROMOD data files is possible from 2022 onwards based on personal identifiers, which opens up possibilities to study the relationship between transitions (i.e. on the labour market), transfers and income.

There are a few caveats to keep in mind when using the linked datasets. There is considerable attrition from the original EUROMOD sample, which is often random, when looking at sex, age, income and education level of the individuals, but in some countries, differences may bias the matched sample. As the differences, both the magnitude and the sign are different across countries, we suggest checking the representativeness of the matched sample for each country that is part of the planned analysis. We also suggest considering reweighting the matched sample to account for non-random attrition.



Annex: Stata syntaxes

We provide Stata syntaxes (do files written in Stata/BE 18.0) to support users

- in the preparation of a longitudinal EU-SILC file including H-, D-, P- and R-files for all available countries in 2022,
- in the preparation of a EUROMOD data file that may be linked to EU-SILC, including all the available countries in 2022,
- and in merging the prepared longitudinal EU-SILC file with the prepared EUROMOD data file.

Preparation of the Longitudinal EU-SILC file

We are preparing a longitudinal EU-SILC file with all the available countries in 2022. We use the Stata-syntaxes provided by Pforr and Riemann (2025), available at <https://www.gesis.org/gml/european-microdata/eu-silc/> for transforming EU-SILC csv data into a Stata-System file. Then we put together the P-, H-, R-, and D-files – we provide the syntax below.

This Stata syntax below merges the above prepared P-, H-, R- and D-files and renames a few variables (used for linking) to match EUROMOD variable names. The output of the syntax is the data file “LongiSILC_all_2022_DHRP_euromod.dta”, which includes all available countries and their P-m H-, R-, D-files from the EU-SILC Panel 2022 - release 2025_release1 / DOI: 10.2907/EUSILC2004-2024.

```
*****
* MERGING P, R, H, D files of longitudinal SILC 2025, release 1 *****
*****

/* Initialization commands */
clear
capture log close
set more off
set linesize 250
set varabbrev off
* -----
* CONFIGURATION SECTION - Start
* The following command should contain the complete path and name of the Stata log file.
* Change LOG_FILENAME to your filename
*local log_file "LOG_FILENAME"
local log_file "[YOURPATH]/LOG_LongiSILC_2022_DHRP"
```

```

*global "DTA_PATH" - Add the path to your DTA folder
*[YOURPATH] refers to path to reach the folder on your computer where the set of folders used for
this specific exercise are placed.
global path "[YOURPATH]/LongiSILC_2022_allcountries/"

* Change STATA_FILENAME.dta to your final filename
local stata_file "${path}LongiSILC_all_2022_DHRP.dta"

*****
*** MERGING D, R, P, H FILES *****
*****

*** 1. PREPARATIONS - RENAMING VARIABLES *****
*** D-FILE
use "${path}LongiSILC_2022_D.dta", clear

rename DB010 year
rename DB020 country
rename DB030 idhh
rename DB040 region
rename DB095 hhcsweight_longi
rename DB060 primsu //primary sampling units
rename DB062 secsu //secondary sampling units

sort year country idhh
duplicates drop // We delete dupliptes here.

save "${path}LongiSILC_2022_D_m.dta", replace
clear
clear matrix

*** R-FILE
use "${path}LongiSILC_2022_R.dta"

rename RB010 year
rename RB020 country
rename RB030 idperson
rename RB040 idhh

```



```
sort year country idhh idperson
```

```
save "${path}LongiSILC_2022_R_m.dta", replace  
clear  
clear matrix
```

```
**** H-FILE  
use "${path}LongiSILC_2022_H.dta"
```

```
rename HB010 year  
rename HB020 country  
rename HB030 idhh
```

```
sort year country idhh  
duplicates drop
```

```
save "${path}LongiSILC_2022_H_m.dta", replace  
clear  
clear matrix
```

```
**** P-FILE  
use "${path}LongiSILC_2022_P.dta", clear
```

```
rename PB010 year  
rename PB020 country  
rename PB030 idperson  
rename PX030 idhh
```

```
sort year country idhh idperson  
save "${path}LongiSILC_2022_P_m.dta", replace
```

```
**** 2. MERGING R-P-D-H files *****  
use "${path}LongiSILC_2022_R_m.dta", clear  
sort year country idhh idperson
```

```
merge 1:1 year country idhh idperson using "${path}LongiSILC_2022_P_m.dta"  
drop _merge
```

```
sort year country idhh
merge year country idhh using "${path}LongiSILC_2022_D_m.dta"
drop _merge
```

```
sort year country idhh
merge year country idhh using "${path}LongiSILC_2022_H_m.dta"
drop _merge
```

```
save "${path}LongiSILC_all_2022_DHRP_euromod.dta", replace
```



Preparation of the EUROMOD data file

We are preparing a data file, which includes all the available country data input files of EUROMOD, UDB (C22_release_23_09).

This Stata syntax below transforms the EUROMOD txt files (as released by EUROMOD) into Stata system files (dta) and pools all the available countries into one Stata system file (dta). The output of the syntax is the “EM_2022_all.dta”, which includes all the countries available for 2022.

```
*****
* Preparing EUROMOD_2022 files for merging them with longi-SILC 2022 *
*****

*****
* INITIALIZATION commands *
*****

clear
capture log close
set more off
set linesize 250
set varabbrev off

*****
* CONFIGURATION SECTION *
*****

* Set globals to have the paths to your data folders ready
global path_0 "[YOURPATH]/EUROMOD input data files/" // add the path to your folder with the
Euromod input data files (txts)
global path_1 "[YOURPATH]/Euromod_dta/" // add the path to your folder where you save the dta
Euromod files prepared for merging

*****
* 1. PREPARATIONS *
*****

* Preparing Euromod input data files for the merge - adding year and country variables to Euromod
input data files and save them as dta
* (We did not use a loop given the differences in the input file names for the countries. Nevertheless,
it could be possible with caution.)

* AT
import delimited "${path_0}AT/AT_2022_b1_2015_03_n2.txt", clear
gen year=2022
```

```
gen country="AT"  
sort year country idhh idperson  
save "${path_1}AT_2022_b1_2015_03_n2.dta", replace  
duplicates report year country idhh idperson // checking for duplicates -none
```

* BE

```
import delimited "${path_0}BE/BE_2022_c1_2015_03_e2.txt", clear  
gen year=2022  
gen country="BE"  
sort year country idhh idperson  
save "${path_1}BE_2022_c1_2015_03_e2.dta", replace  
duplicates report year country idhh idperson
```

* BG

```
import delimited "${path_0}BG/BG_2022_c1_2015_03_e2.txt", clear  
gen year=2022  
gen country="BG"  
sort year country idhh idperson  
save "${path_1}BG_2022_c1_2015_03_e2.dta", replace
```

* CY

```
import delimited "${path_0}CY/CY_2022_b2_2015_03_e2.txt", clear  
gen year=2022  
gen country="CY"  
sort year country idhh idperson  
save "${path_1}CY_2022_b2_2015_03_e2.dta", replace
```

* CZ

```
import delimited "${path_0}CZ/CZ_2022_b1_2015_03_e2.txt", clear  
gen year=2022  
gen country="CZ"  
sort year country idhh idperson  
save "${path_1}CZ_2022_b1_2015_03_e2.dta", replace
```

* DE - not included in longiSILC 2022

```
import delimited "${path_0}DE/DE_2022_b1_2015_03_e2.txt", clear  
gen year=2022  
gen country="DE"  
sort year country idhh idperson  
save "${path_1}DE_2022_b1_2015_03_e2.dta", replace
```

* DK

```
import delimited "${path_0}DK/DK_2022_c1_2015_03_e2.txt", clear  
gen year=2022
```



```
gen country="DK"
sort year country idhh idperson
save "${path_1}DK_2022_c1_2015_03_e2.dta", replace

* EE
import delimited "${path_0}EE/EE_2022_f1_2015_03_e2.txt", clear
gen year=2022
gen country="EE"
sort year country idhh idperson
save "${path_1}EE_2022_f1_2015_03_e2.dta", replace

* EL
import delimited "${path_0}EL/EL_2022_c2_2015_03_e2.txt", clear
gen year=2022
gen country="EL"
sort year country idhh idperson
save "${path_1}EL_2022_c2_2015_03_e2.dta", replace

* ES
import delimited "${path_0}ES/ES_2022_b2_2015_03_e2.txt", clear
gen year=2022
gen country="ES"
sort year country idhh idperson
save "${path_1}ES_2022_b2_2015_03_e2.dta", replace

* FI
import delimited "${path_0}FI/FI_2022_b1_2015_03_e2.txt", clear
gen year=2022
gen country="FI"
sort year country idhh idperson
save "${path_1}FI_2022_b1_2015_03_e2.dta", replace

* FR
import delimited "${path_0}FR/FR_2022_c1_2015_03_e2.txt", clear
gen year=2022
gen country="FR"
sort year country idhh idperson
save "${path_1}FR_2022_c1_2015_03_e2.dta", replace

* HR
import delimited "${path_0}HR/HR_2022_b2_2015_03_e2.txt", clear
gen year=2022
gen country="HR"
```

```
sort year country idhh idperson
save "${path_1}HR_2022_b2_2015_03_e2.dta", replace
```

* HU - not included in longiSILC 2022 and Euromod 2022 either (due to microdata issues)

* IE

```
import delimited "${path_0}IE/IE_2022_b1_2015_03_e2.txt", clear
gen year=2022
gen country="IE"
sort year country idhh idperson
save "${path_1}IE_2022_b1_2015_03_e2.dta", replace
```

* IT

```
import delimited "${path_0}IT/IT_2022_b2.txt", clear
gen year=2022
gen country="IT"
sort year country idhh idperson
save "${path_1}IT_2022_b2.dta", replace
```

* LT

```
import delimited "${path_0}LT/LT_2022_c1_2015_03_e2.txt", clear
gen year=2022
gen country="LT"
sort year country idhh idperson
save "${path_1}LT_2022_c1_2015_03_e2.dta", replace
```

* LU

```
import delimited "${path_0}LU/LU_2022_b1_2015_03_e2.txt", clear
gen year=2022
gen country="LU"
sort year country idhh idperson
save "${path_1}LU_2022_b1_2015_03_e2.dta", replace
```

* LV

```
import delimited "${path_0}LV/LV_2022_b2_2015_03_e2.txt", clear
gen year=2022
gen country="LV"
sort year country idhh idperson
save "${path_1}LV_2022_b2_2015_03_e2.dta", replace
```

* MT

```
import delimited "${path_0}MT/MT_2022_b1_2015_03_e2.txt", clear
gen year=2022
gen country="MT"
```



```
sort year country idhh idperson  
save "${path_1}MT_2022_b1_2015_03_e2.dta", replace
```

* NL

```
import delimited "${path_0}NL/NL_2022_b1_2015_03_e2.txt", clear  
gen year=2022  
gen country="NL"  
sort year country idhh idperson  
save "${path_1}NL_2022_b1_2015_03_e2.dta", replace
```

* PL is missing from Euromod 2022. Available for 2021, 2020 and 2019

* PT

```
import delimited "${path_0}PT/PT_2022_b2_2015_03_e2.txt", clear  
gen year=2022  
gen country="PT"  
sort year country idhh idperson  
save "${path_1}PT_2022_b2_2015_03_e2.dta", replace
```

* RO

```
import delimited "${path_0}RO/RO_2022_b1_2015_03_e2.txt", clear  
gen year=2022  
gen country="RO"  
sort year country idhh idperson  
save "${path_1}RO_2022_b1_2015_03_e2.dta", replace
```

* SE

```
import delimited "${path_0}SE/SE_2022_b1_2015_03_e2.txt", clear  
gen year=2022  
gen country="SE"  
sort year country idhh idperson  
save "${path_1}SE_2022_b1_2015_03_e2.dta", replace
```

* SI

```
import delimited "${path_0}SI/SI_2022_c1_2015_03_e2.txt", clear  
gen year=2022  
gen country="SI"  
sort year country idhh idperson  
save "${path_1}SI_2022_c1_2015_03_e2.dta", replace
```

* SK

```
import delimited "${path_0}SK/SK_2022_b1_2015_03_e2.txt", clear  
gen year=2022  
gen country="SK"
```

```
sort year country idhh idperson
save "${path_1}SK_2022_b1_2015_03_e2.dta", replace
```

* Appending Euromod files - to have a dataset for 2022 with all the available countries

```
use "${path_1}AT_2022_b1_2015_03_n2.dta", clear
```

```
append using "${path_1}BE_2022_c1_2015_03_e2.dta" // appending EM input data file one by one
append using "${path_1}BG_2022_c1_2015_03_e2.dta"
append using "${path_1}CY_2022_b2_2015_03_e2.dta"
append using "${path_1}CZ_2022_b1_2015_03_e2.dta"
append using "${path_1}DE_2022_b1_2015_03_e2.dta"
append using "${path_1}DK_2022_c1_2015_03_e2.dta"
append using "${path_1}EE_2022_f1_2015_03_e2.dta"
append using "${path_1}EL_2022_c2_2015_03_e2.dta"
append using "${path_1}ES_2022_b2_2015_03_e2.dta"
append using "${path_1}FI_2022_b1_2015_03_e2.dta"
append using "${path_1}FR_2022_c1_2015_03_e2.dta"
append using "${path_1}HR_2022_b2_2015_03_e2.dta"
append using "${path_1}IE_2022_b1_2015_03_e2.dta"
append using "${path_1}IT_2022_b2.dta"
append using "${path_1}LT_2022_c1_2015_03_e2.dta"
append using "${path_1}LU_2022_b1_2015_03_e2.dta"
append using "${path_1}LV_2022_b2_2015_03_e2.dta"
append using "${path_1}MT_2022_b1_2015_03_e2.dta"
append using "${path_1}NL_2022_b1_2015_03_e2.dta"
append using "${path_1}PT_2022_b2_2015_03_e2.dta"
append using "${path_1}RO_2022_b1_2015_03_e2.dta"
append using "${path_1}SE_2022_b1_2015_03_e2.dta"
append using "${path_1}SI_2022_c1_2015_03_e2.dta"
append using "${path_1}SK_2022_b1_2015_03_e2.dta"
```

```
save "${path_1}EM_2022_all.dta", replace // Euromod dataset for 2022 including all the available
countries
```



Merging longitudinal EU-SILC with EUROMOD

We are linking the above prepared longitudinal EU-SILC 2022 dataset and the above prepared EUROMOD 2022 dataset.

This Stata syntax below merges the “LongiSILC_all_2022_DHRP_euromod.dta” file with the “EM_2022_all.dta” file. Additionally, it includes checks for each country to see whether the linkage based on personal ID is feasible. We flag those individuals where the difference between age based on EU-SILC and EUROMOD was larger than 2. The output of the syntax is the “LongiSILC_Euromod_2022.dta”, which includes overall 325 897 matched individuals from 23 EU countries (AT, BE, BG, CY, CZ, DK, EE, EL, ES, FI, FR, HR, IE, IT, LT, LU, LV, NL, PT, RO, SE, SI, SK).

```
*****
* Merging longi-SILC 2022 with EUROMOD_2022 files *
*****
*****
* INITIALIZATION commands *
*****

clear
capture log close
set more off
set linesize 250
set varabbrev off

*****
* CONFIGURATION SECTION *
*****
* Set globals to have the paths to your data folders ready
global path_1 "[YOURPATH]/Euromod_dta/" // add the path to your folder where you save the dta
Euromod files prepared for merging
global path "[YOURPATH]/LongiSILC_2022_allcountries/" // add the path to your folder with the
LongiSILC-2022 file
*****
* 1. MERGING AND CHECKS *
*****
* Merging the longi-SILC 2022 dataset including all the countries with Euromod 2022 dataset
including all the countries
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson

merge 1:1 year country idhh idperson using "${path_1}EM_2022_all.dta"

* Checking whether the matched individuals are identical in terms of age and sex
```

```

* age - (Italy is dropped as there is no age in EU-SILC for Italy)
count if (_merge==3 & RB081!=dag & country!="IT") // 5376 individuals have a different age coded
in SILC and Euromod
count if (_merge==3 & abs(RB081 - dag) > 2 & country!="IT") // 3244 individuals - the age difference
in SILC and Euromod is larger than 2
gen flag_unequalage = (_merge==3 & RB081!=dag & country!="IT")
* sex
gen sex_silc=. // generating a sex variable that is coded as "dgn" (EM sex var.): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) // 486
gen flag_unequalsex = (_merge==3 & sex_silc!=dgn) // flagging individuals where the sex is different
in SILC and EM

* Flagging individuals where the age or the sex is different in SILC and EM:
gen flag_unequal = (_merge==3 & flag_unequalage==1 | flag_unequalsex==1) // 5455
* Checking the number of matches in 2022 for countries included both in longiSILC-2022 and
Euromod 2022:
tab _merge if ((country=="AT" | country=="BE" | country=="BG" | country=="CY" | country=="CZ" |
country=="DK" | country=="EE" | country=="EL" | country=="ES" |
country=="FI" | country=="FR" | country=="HR" |
country=="IE" | country=="IT" | country=="LT" | country=="LU" | country=="LV" | country=="NL" | country=="
"PT" | country=="RO" | country=="SE" | country=="SI" | country=="SK") & year==2022)

drop _merge

* Saving the full merged dataset
save "${path}LongiSILC_Euromod_2022.dta", replace

*****

* 2. CHECKS AND DESCRIPTIVES BY COUNTRY *
*****

* Merging longiSILC 2022 with the EUROMOD 2022 country files, country by country to see country-
level descriptives and anomalies

```



*AT

```
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}AT_2022_b1_2015_03_n2.dta"
tab _merge if country=="AT" & year==2022
list if (_merge==3 & RB081!=dag) // checking for differences - none
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) // 0
```

* Attrition - EM

* gender

```
tabstat dgn if year==2022 & country=="AT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="AT" & inlist(_merge,2,3), by(_merge)
```

* age

```
tabstat dag if year==2022 & country=="AT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="AT" & inlist(_merge,2,3), by(_merge)
```

* income - ydses_o (INCOME : Disposable: equivalized: original SILC)

```
tabstat ydses_o if year==2022 & country=="AT", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="AT" & inlist(_merge,2,3), by(_merge)
```

* education (years spent in education)

```
tabstat dey if year==2022 & country=="AT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="AT" & inlist(_merge,2,3), by(_merge)
```

*BE

```
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}BE_2022_c1_2015_03_e2.dta"
tab _merge if country=="BE" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 152 individuals' ages do not match
count if (_merge==3 & abs(RB081 - dag) > 2) // 104 individuals where the difference is larger than 2
list idperson if (_merge==3 & RB081!=dag) // listing the 152 individuals
```

```

list idperson if (_merge==3 & abs(RB081 - dag) > 2)
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) // 133
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="BE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="BE" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="BE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="BE" & inlist(_merge,2,3), by(_merge)
* income - ydses_o
tabstat ydses_o if year==2022 & country=="BE", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="BE" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="BE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="BE" & inlist(_merge,2,3), by(_merge)

*BG
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}BG_2022_c1_2015_03_e2.dta"
tab _merge if country=="BG" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 15 individuals' ages do not match
count if (_merge==3 & abs(RB081 - dag) > 2) // 13 where the diff. is larger than 2 years
list idperson if (_merge==3 & RB081!=dag)
list idperson if (_merge==3 & abs(RB081 - dag) > 2)
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female

```



```
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="BG", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="BG" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="BG", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="BG" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="BG", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="BG" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="BG", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="BG" & inlist(_merge,2,3), by(_merge)

*CY
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}CY_2022_b2_2015_03_e2.dta"
tab _merge if country=="CY" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 420
count if (_merge==3 & abs(RB081 - dag) > 2) // 39
list idperson if (_merge==3 & RB081!=dag)
list idperson if (_merge==3 & abs(RB081 - dag) > 2)
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
```

```

tabstat dgn if year==2022 & country=="CY", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="CY" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="CY", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="CY" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="CY", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="CY" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="CY", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="CY" & inlist(_merge,2,3), by(_merge)

*CZ
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}CZ_2022_b1_2015_03_e2.dta"
tab _merge if country=="CZ" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 744
count if (_merge==3 & abs(RB081 - dag) > 2) // 525
list idperson if (_merge==3 & RB081!=dag)
list idperson if (_merge==3 & abs(RB081 - dag) > 2)
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="CZ", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="CZ" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="CZ", by(_merge) stat(mean sd n) format(%6.3f)

```



```
ttest dag if year==2022 & country=="CZ" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="CZ", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="CZ" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="CZ", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="CZ" & inlist(_merge,2,3), by(_merge)

*DE is missing from longiSILC 2022

*DK
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}DK_2022_c1_2015_03_e2.dta"
tab _merge if country=="DK" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 467
count if (_merge==3 & abs(RB081 - dag) > 2) // 0
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* attrition - random?
tabstat sex_silc if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f) // only
10 unmatched SILC individual with age (otherwise empty rows)
tabstat RB090 if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f)
ttest sex_silc if year==2022 & country=="DK" & inlist(_merge,1,3), by(_merge)
tabstat RB081 if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f)
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="DK" & inlist(_merge,2,3), by(_merge)
```

```

* age
tabstat dag if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="DK" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="DK" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="DK", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="DK" & inlist(_merge,2,3), by(_merge)

*EE
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}EE_2022_f1_2015_03_e2.dta"
tab _merge if country=="EE" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 68
count if (_merge==3 & abs(RB081 - dag) > 2) // 47
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="EE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="EE" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="EE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="EE" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="EE", by(_merge) stat(mean sd n) format(%6.3f)

```



```
ttest ydses_o if year==2022 & country=="EE" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="EE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="EE" & inlist(_merge,2,3), by(_merge)

*EL
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}EL_2022_c2_2015_03_e2.dta"
tab _merge if country=="EL" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 1485
count if (_merge==3 & abs(RB081 - dag) > 2) // 1091
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) // 10
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="EL", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="EL" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="EL", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="EL" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="EL", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="EL" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="EL", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="EL" & inlist(_merge,2,3), by(_merge)
```

```

*ES
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}ES_2022_b2_2015_03_e2.dta"
tab _merge if country=="ES" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 0
count if (_merge==3 & abs(RB081 - dag) > 2) // 0
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="ES", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="ES" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="ES", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="ES" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="ES", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="ES" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="ES", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="ES" & inlist(_merge,2,3), by(_merge)

*FI
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}FI_2022_b1_2015_03_e2.dta"
tab _merge if country=="FI" & year==2022

```



* checking age

```
count if (_merge==3 & RB081!=dag) // 6
```

```
count if (_merge==3 & abs(RB081 - dag) > 2) // 0
```

* checking sex

```
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
```

```
replace sex_silc=0 if RB090==2 // female
```

```
replace sex_silc=1 if RB090==1 // male
```

```
count if (_merge==3 & sex_silc!=dgn) //
```

* Attrition - EM

* gender

```
tabstat dgn if year==2022 & country=="FI", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dgn if year==2022 & country=="FI" & inlist(_merge,2,3), by(_merge)
```

* age

```
tabstat dag if year==2022 & country=="FI", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dag if year==2022 & country=="FI" & inlist(_merge,2,3), by(_merge)
```

* income

```
tabstat ydses_o if year==2022 & country=="FI", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest ydses_o if year==2022 & country=="FI" & inlist(_merge,2,3), by(_merge)
```

* education (years spent in education)

```
tabstat dey if year==2022 & country=="FI", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dey if year==2022 & country=="FI" & inlist(_merge,2,3), by(_merge)
```

*FR

```
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
```

```
sort year country idhh idperson
```

```
merge 1:1 year country idhh idperson using "${path_1}FR_2022_c1_2015_03_e2.dta"
```

```
tab _merge if country=="FR" & year==2022
```

* checking age

```
count if (_merge==3 & RB081!=dag) // 1115
```

```
count if (_merge==3 & abs(RB081 - dag) > 2) // 802
```

* checking sex

```
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
```

```
replace sex_silc=0 if RB090==2 // female
```

```

replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="FR", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="FR" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="FR", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="FR" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="FR", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="FR" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="FR", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="FR" & inlist(_merge,2,3), by(_merge)

*HR
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}HR_2022_b2_2015_03_e2.dta"
tab _merge if country=="HR" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 13
count if (_merge==3 & abs(RB081 - dag) > 2) // 8
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="HR", by(_merge) stat(mean sd n) format(%6.3f)

```



```
ttest dgn if year==2022 & country=="HR" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="HR", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="HR" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="HR", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="HR" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="HR", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="HR" & inlist(_merge,2,3), by(_merge)

*HU is missing from longiSILC 2022

*IE
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}IE_2022_b1_2015_03_e2.dta"
tab _merge if country=="IE" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) //
count if (_merge==3 & abs(RB081 - dag) > 2) //
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* attrition - random?
tabstat sex_silc if year==2022 & country=="IE", by(_merge) stat(mean sd n) format(%6.3f) // only 8
unmatched SILC individual with age (otherwise empty rows)
tabstat RB081 if year==2022 & country=="IE", by(_merge) stat(mean sd n) format(%6.3f)
* Attrition - EM
* gender
```

```

tabstat dgn if year==2022 & country=="IE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="IE" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="IE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="IE" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="IE", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="IE" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="IE", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="IE" & inlist(_merge,2,3), by(_merge)

*IT
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}IT_2022_b2.dta"
tab _merge if country=="IT" & year==2022
* checking age - not possible
*count if (_merge==3 & RB081!=dag) // Age (RB081) is missing for Italy
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) // 14
* attrition - random?
tabstat sex_silc if year==2022 & country=="IT", by(_merge) stat(mean sd n) format(%6.3f) // only 90
unmatched SILC individual with age (otherwise empty rows)
tabstat RB081 if year==2022 & country=="IT", by(_merge) stat(mean sd n) format(%6.3f) // age is
not available
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="IT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="IT" & inlist(_merge,2,3), by(_merge)
* age

```



```
tabstat dag if year==2022 & country=="IT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="IT" & inlist(_merge,2,3), by(_merge)
* income
tabstat yds if year==2022 & country=="IT", by(_merge) stat(mean sd n) format(%6.3f)
ttest yds if year==2022 & country=="IT" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="IT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="IT" & inlist(_merge,2,3), by(_merge)

*LT
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}LT_2022_c1_2015_03_e2.dta"
tab _merge if country=="LT" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 0
count if (_merge==3 & abs(RB081 - dag) > 2) // 0
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="LT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="LT" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="LT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="LT" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="LT", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="LT" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
```

```

tabstat dey if year==2022 & country=="LT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="LT" & inlist(_merge,2,3), by(_merge)

*LU
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}LU_2022_b1_2015_03_e2.dta"
tab _merge if country=="LU" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 61
count if (_merge==3 & abs(RB081 - dag) > 2) //35
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - random?
tabstat sex_silc if year==2022 & country=="LU", by(_merge) stat(mean sd n) format(%6.3f) // only
40 unmatched SILC individual with age (otherwise empty rows)
tabstat RB081 if year==2022 & country=="LU", by(_merge) stat(mean sd n) format(%6.3f) // only 40
unmatched
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="LU", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="LU" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="LU", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="LU" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="LU", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="LU" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="LU", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="LU" & inlist(_merge,2,3), by(_merge)

```



*LV

```
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}LV_2022_b2_2015_03_e2.dta"
tab _merge if country=="LV" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 2
count if (_merge==3 & abs(RB081 - dag) > 2) // 2
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="LV", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="LV" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="LV", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="LV" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="LV", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="LV" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="LV", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="LV" & inlist(_merge,2,3), by(_merge)
```

*MT

```
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}MT_2022_b1_2015_03_e2.dta" // No
matches made, merging is not possible for MT
```

```

*NL
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}NL_2022_b1_2015_03_e2.dta"
tab _merge if country=="NL" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 0
count if (_merge==3 & abs(RB081 - dag) > 2) // 0
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="NL", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="NL" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="NL", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="NL" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="NL", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="NL" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="NL", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="NL" & inlist(_merge,2,3), by(_merge)

```

*PL - PL is missing from Euromod 2022.

```

*PT
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson

```



```
merge 1:1 year country idhh idperson using "${path_1}PT_2022_b2_2015_03_e2.dta"
tab _merge if country=="PT" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 109
count if (_merge==3 & abs(RB081 - dag) > 2) // 83
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="PT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="PT" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="PT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="PT" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="PT", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="PT" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="PT", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="PT" & inlist(_merge,2,3), by(_merge)

*RO
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}RO_2022_b1_2015_03_e2.dta"
tab _merge if country=="RO" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 708
count if (_merge==3 & abs(RB081 - dag) > 2) // 488
```

```

* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //

* Attrition - EM

* gender
tabstat dgn if year==2022 & country=="RO", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="RO" & inlist(_merge,2,3), by(_merge)

* age
tabstat dag if year==2022 & country=="RO", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="RO" & inlist(_merge,2,3), by(_merge)

* income
tabstat ydses_o if year==2022 & country=="RO", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="RO" & inlist(_merge,2,3), by(_merge)

* education (years spent in education)
tabstat dey if year==2022 & country=="RO", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="RO" & inlist(_merge,2,3), by(_merge)

*SE
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}SE_2022_b1_2015_03_e2.dta"
tab _merge if country=="SE" & year==2022

* checking age
count if (_merge==3 & RB081!=dag) // 0
count if (_merge==3 & abs(RB081 - dag) > 2) // 0

* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //

```



* Attrition - random?

```
tabstat sex_silc if year==2022 & country=="SE", by(_merge) stat(mean sd n) format(%6.3f) // only  
30 unmatched SILC individual with age (otherwise empty rows)
```

```
tabstat RB081 if year==2022 & country=="SE", by(_merge) stat(mean sd n) format(%6.3f) // only 30,  
no age observed (missing observations)
```

* Attrition - EM

* gender

```
tabstat dgn if year==2022 & country=="SE", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dgn if year==2022 & country=="SE" & inlist(_merge,2,3), by(_merge)
```

* age

```
tabstat dag if year==2022 & country=="SE", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dag if year==2022 & country=="SE" & inlist(_merge,2,3), by(_merge)
```

* income

```
tabstat ydses_o if year==2022 & country=="SE", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest ydses_o if year==2022 & country=="SE" & inlist(_merge,2,3), by(_merge)
```

* education (years spent in education)

```
tabstat dey if year==2022 & country=="SE", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dey if year==2022 & country=="SE" & inlist(_merge,2,3), by(_merge)
```

*SI

```
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
```

```
sort year country idhh idperson
```

```
merge 1:1 year country idhh idperson using "${path_1}SI_2022_c1_2015_03_e2.dta"
```

```
tab _merge if country=="SI" & year==2022
```

* checking age

```
count if (_merge==3 & RB081!=dag) // 0
```

```
count if (_merge==3 & abs(RB081 - dag) > 2) // 0
```

* checking sex

```
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
```

```
replace sex_silc=0 if RB090==2 // female
```

```
replace sex_silc=1 if RB090==1 // male
```

```
count if (_merge==3 & sex_silc!=dgn) //
```

* Attrition - EM

```

* gender
tabstat dgn if year==2022 & country=="SI", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="SI" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="SI", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="SI" & inlist(_merge,2,3), by(_merge)
* income
tabstat ydses_o if year==2022 & country=="SI", by(_merge) stat(mean sd n) format(%6.3f)
ttest ydses_o if year==2022 & country=="SI" & inlist(_merge,2,3), by(_merge)
* education (years spent in education)
tabstat dey if year==2022 & country=="SI", by(_merge) stat(mean sd n) format(%6.3f)
ttest dey if year==2022 & country=="SI" & inlist(_merge,2,3), by(_merge)

*SK
use "${path}LongiSILC_all_2022_DHRP_euromod.dta", clear
sort year country idhh idperson
merge 1:1 year country idhh idperson using "${path_1}SK_2022_b1_2015_03_e2.dta"
tab _merge if country=="SK" & year==2022
* checking age
count if (_merge==3 & RB081!=dag) // 11
count if (_merge==3 & abs(RB081 - dag) > 2) // 7
* checking sex
gen sex_silc=. // generating a sex variable that is coded as dgn (EM sex var): 0: female, 1: male
replace sex_silc=0 if RB090==2 // female
replace sex_silc=1 if RB090==1 // male
count if (_merge==3 & sex_silc!=dgn) //
* Attrition - EM
* gender
tabstat dgn if year==2022 & country=="SK", by(_merge) stat(mean sd n) format(%6.3f)
ttest dgn if year==2022 & country=="SK" & inlist(_merge,2,3), by(_merge)
* age
tabstat dag if year==2022 & country=="SK", by(_merge) stat(mean sd n) format(%6.3f)
ttest dag if year==2022 & country=="SK" & inlist(_merge,2,3), by(_merge)

```



* income

```
tabstat ydses_o if year==2022 & country=="SK", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest ydses_o if year==2022 & country=="SK" & inlist(_merge,2,3), by(_merge)
```

* education (years spent in education)

```
tabstat dey if year==2022 & country=="SK", by(_merge) stat(mean sd n) format(%6.3f)
```

```
ttest dey if year==2022 & country=="SK" & inlist(_merge,2,3), by(_merge)
```

SWINS
Sustainable Well-Being
through Investments
in Social Services
sustainablewellbeing.eu

