InGRID

Supporting expertise in inclusive growth

www.inclusivegrowth.eu

Deliverable 8.9

COMBINATION OF EUROMOD AND LIAM2 TOOLS FOR THE DEVELOPMENT OF DYNAMIC MICROSIMULATION MODELS

FEASIBILITY, EXAMPLE AND CONDITIONS FOR SUSTAINABILITY OF THE LINKAGE

Philippe Liégeois

October 2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 730998

Abstract

LISER (https://www.liser.lu/) has implemented, within Work Package 8 of the EU-'HORIZON 2020'-funded 'InGRID-2'project (http://www.inclusivegrowth.eu/), a combination of tools for the development of dynamic microsimulation models which objective is 'to establish a linkage between two important microsimulation platforms: EUROMOD and LIAM2'.EUROMOD is an efficient and well-known platform for the development of static microsimulation models. It is also a model in itself. LIAM2 is a free, open source and user-friendly modelling and simulation framework, mainly implemented for the development of dynamic discrete-time microsimulation models with cross-sectional population ageing. LIAM2 has strong similarities in its built-in logic and grammar to EUROMOD. It is not a model but has been adopted by modellers in at least nine countries (AR, BE, CN, FR, IT, HU, LU, PT and SK) and has led to the creation of several dynamic microsimulation models with a similar structure (based on MiDAS_BE, Federal Planning Bureau, Brussels).In examining the possible avenues for implementing a link between the two platforms, one track was favoured, known as 'Dynamic in Static/DiS'. It aims at dynamising to a certain extent a static model such as EUROMOD in order to carry out short-term projections, as is already done in the well-known developments of 'nowcasting'. But our aim is to derive a device that is both as transparent as possible, and therefore more easily reproducible and improvable, and that links EUROMOD to another innovative development platform targeting the development and simulation of dynamic models. This could open up avenues for a more sophisticated implementation of the dynamic side of the combination, which might be valuable in a particularly volatile context. This implementation may also generate positive by-products of interest for the development of dynamic microsimulation models in a broader sense. This includes a 'Static in Dynamic/SiD' approach in which EUROMOD could become the tax-benefit module of a larger dynamic model operating over the long term. While proceeding, we emphasise the 'infrastructural' nature of the InGRID-2 project, hence the need to draw synergies from existing experiences to innovate at the level of the development and simulation tools mobilised in the background (EUROMOD-LIAM2 linkage), more than to rewrite from scratch or on a large scale, or even to deepen, the short-term projection procedures. What is at stake here is a method, considered as an intermediate good, rather than applications, considered as final goods, yet keeping in mind that the latter remain of utmost importance with regard to the sustainability of the path proposed here. The methodology of combination is described, implemented and first outcomes reported, including possible drawbacks.

This report constitutes Deliverable 8.9, for Work Package 8 of the InGRID-2 project.

October 2021

General contact: inclusive.growth@kuleuven.be
p.a. InGRID
HIVA - Research Institute for Work and Society
Parkstraat 47 box 5300, 3000 LEUVEN, Belgium

For more information philippe.Liegeois@liser.lu

Please refer to this publication as follows:
Type the author(s), title & subtitle, Deliverable n°, Liégeois, P. (2021), Combination of EUROMOD and LIAM2 tools for the development of dynamic microsimulation models, Feasibility, example and conditions for sustainability of the linkage, Deliverable 8.9, Leuven, InGRID-2 project 730998 – H2020

This publication is part of the InGRID-2 project, this project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 730998.

InGRID
Supporting expertise in inclusive growth

Contents

1. The origin of the proposal of a static-dynamic linkage and background motivations

The objective of Work Package 8 (hereafter WP8) in InGRID-2, entitled '*Data harmonisation and integration regarding poverty & living conditions'*,[1] is to harmonise and integrate various research infrastructures and thereby create new possibilities for European research on poverty, living conditions and social policy, as well as more effective policymaking.

TARKI takes the lead of this work package which includes several tasks, among which:

> Task 8 (LISER (lead), UEssex, UA): Combining tools for the development of dynamic microsimulation models. The purpose of this task is to establish linkages between two important microsimulation platforms: EUROMOD and LIAM2. EUROMOD is an efficient platform for the development of static microsimulation models. LIAM2 is a free, open source, user-friendly modelling and simulation framework mainly set up for the development of discrete-time dynamic microsimulation models with cross-sectional dynamic ageing. We will organise EUROMOD to act as a tax-benefit module from within a dynamic framework. We will address the technical questions of establishing a linkage between EUROMOD and LIAM2, and based on practical examples we will show how this objective can be realised.
> We will also establish conditions for guaranteeing sustainability of the linkage, for example, in relation to future updates of the models.

As mentioned above, EUROMOD (Sutherland & Figari, 2013) is an efficient and well-known platform for the development of static microsimulation models. It also embeds tax-benefit models for the European Union and UK. LIAM2 is a free, open source, user-friendly modelling and simulation framework mainly implemented for the development of discrete-time microsimulation models with cross-sectional dynamic ageing. LIAM2 is a platform that does not provides models as such. The latter are developed on a free basis by 'local' teams.

Whereas the EUROMOD package was developed a long time ago, a similar platform for the development and simulation of *dynamic* microsimulation models, the 'Life-cycle income analysis model'/LIAM (O'Donoghue, Hynes & Lennon, 2009) was created a few years later by Cathal O'Donoghue, originally a member of the EUROMOD team. The objects studied - microsimulation models - and this 'parental' proximity led to strong similarities between the embedded logics and grammars of LIAM and EUROMOD. This prompted Philippe Liégeois and Gijs Dekkers (2014) to evaluate a few years ago the feasibility and limitations of a combination between EUROMOD and a LIAM-based dynamic model, with the former serving as a tax-benefit module in a larger dynamic framework.[2]

Since then, LIAM has been extensively revised by the Federal Planning Bureau in Brussels, in the framework of the EU-funded 'MiDaL project', conceptualised and managed by LISER,[3] leading to this completely renewed package: 'LIAM2' (de Menten *et al.* [2014]; http://liam2.plan.be/). LIAM2

1 See 'Grant Agreement-730998-InGRID-2 - 12.6.17.pdf', page 132.

2 This research was part of the MiDaL project ('Towards the development of a dynamic microsimulation toolbox LIAM-II and the complementary implementation of administrative Data needed for dynamic microsimulation of pensions in Luxembourg'), supported by the European Community Programme for Employment and Social Solidarity - PROGRESS (2007-2013), under the Grant VS/2009/0569.

3 See previous footnote.

is still being developed continuously and efficiently in the Federal Planning Bureau in Brussels and has been adopted by modellers in at least nine countries[4] (AR, BE, CN, FR, IT, HU, LU, PT and SK), to which others may be added in the near future. In addition, LIAM2 has led to the creation of dynamic discrete-time microsimulation models with dynamic cross-aging with a similar structure to MIDAS_BE (Federal Planning Bureau, Brussels, historically the first of the 'MIDAS' family), MIDAS_LU for Luxembourg, MIDAS_HU, MIDAS_PT, ... In the meantime, the EUROMOD package has also evolved significantly[5] and has become a leading standard in the field of microsimulation (static and behavioural).

This underlying framework involving a certain complementarity and proximity between EUROMOD and LIAM2, on the one hand, and a significant evolution of both platforms over time, on the other hand, has led to the present proposal. Our main purpose is to address the *technical* issue of a *linkage* between recent versions of EUROMOD and LIAM2, more than the analysis of policy-oriented results, which is strictly speaking out of the scope of the present task.

In the core of the report, we clarify and motivate the perimeter chosen for this exercise (*Section 2*) before designing its implementation (*Section 3*), having a look on possible outcomes (*Section 4*) and concluding (*Section 5*).

4 See the Special Interest Group on Dynamic microsimulation/SIG, Work Package 6 of InGRID-2 project.

5 The version adopted in Liégeois and Dekkers (2014) was the '31A'.

2. Clarifying the perimeter of the present task 8

While performing task 8 described in *Section 1*, and before implementing the static/EUROMOD-dynamic/LIAM2 linkage itself, it is appropriate to raise a triple set of questions that help us to conceive the scope of our current work. These issues are now discussed in *Sections 3.1* to *3.3*.

2.1 Deciding upon a Dynamic in Static ('DiS') or a Static in Dynamic ('SiD') approach

The description of task 8 (hereafter 'task' for simplicity) in *Section 1* raises a first question that might be perceived as ambiguous.

On the one hand, we can focus on a dynamic framework and model in which several calculations of tax benefits are performed through the EUROMOD module. This is precisely what has been done with EUROMOD and LIAM in Liégeois and Dekkers (2014).

On the other hand, EUROMOD and its static approach based most often on EU-SILC microdata provided with some delay (2 or 3 years between the collection of raw data and the provision of the SILC set to the end-users) could be extended with the desire to 'dynamise' somewhat the environment from the recent past to present or even the near future. This can be done for example by organising transitions on the labour market due to exogenous changes during this short period (behavioural contents) or by changing the demographic content of the population a little (structural concerns).[6]

The first perspective, which can be named *'Static in Dynamic/SiD'* and which builds for example on Liégeois and Dekkers (2014) aims at making EUROMOD a basis for the tax-benefit module of dynamic microsimulation models. Its underlying motivation is to reduce, through better synergies between static and dynamic tools, the development costs of dynamic microsimulation models that are the cornerstone of this approach.

The second approach, *'Dynamic in Static/DiS'*, has been operationalised recently and is constantly evolving, through experiments in 'nowcasting' or 'flash estimates'. 'Nowcasting' is a contraction of 'now' and 'forecasting'. As O'Donoghue and Loughrey (2014) point out: '*The process of nowcasting refers to the estimation of current indicators using data on the past income distribution combined with other current statistics which may include the latest available macroeconomic statistics* (Navicke *et al.* 2013). ... *These nowcasting methods differ in a number of respects to the more established methods of forecasting. The main distinction is that while forecasting extrapolates from current data to estimate the future, the methods of nowcasting extrapolate from data of the recent past to reflect the present situation.*' The reasons for the development of such a track are well defined in the documents made available on nowcasting works:[7] for example, *providing timelier social statistics, especially indicators on income poverty and inequality, ... an essential tool in order to prepare the European semester ... and to monitor progress towards the Europe 2020 poverty and social exclusion target.*

6 In EUROMOD, the 'population' is unchanged compared to the year of data collection/survey (2 or 3 years, compared to 'now'); the gross income only is 'uprated' to take into account the evolution of average incomes, but this remains approximate, especially in times of crisis when the underlying structure of the population changes significantly: unemployment, etc. Despite more recent 'Add-ons' developments introducing the possibility to impact some labour market-related variables, those adaptations and possibilities are still limited.

7 Cf. EUROSTAT (2018).

The usefulness of both tracks for the current period was discussed in the '*Special Interest Group on dynamic microsimulation'* set up in the framework of the WP6 of InGRID-2.[8]

It appeared that an SiD effort seems to be of rather limited interest at the moment, which might contradict one of the objectives of the task which is to provide a sustainable tool, as far as possible (see *Section 1*). In comparison, the DiS track is now more widely developed and continuously improved by reference organisations such as the EUROMOD network and EUROSTAT. Focusing on the DiS track will therefore allow for a wider prior audience to be targeted, giving the present task a better chance of sustainability. This advantage is even stronger if one considers the large and growing number of dynamic applications based on LIAM2 (see *Section* 1).

A by-product resulting from the implementation of a DiS approach could be *methodological improvements in the very early years of simulation of dynamic models*. The transition from the input microdata year to the early simulation years is critical and demanding, as we suddenly move from 'observed' to 'simulated'- and then modelled - information. This involves many methodological choices, including several econometric analyses in the background. However, this initial period is not always favoured when developing a full dynamic model that raises so many questions and is generally aimed at broader medium- to long-term concerns. This is especially true if the development team is small and there is little time to proceed, which is often the case. Short-term results may then be given a lower priority and thus have some irrelevant discontinuity from the initial 'data' period. The present task, if it focuses on a DiS approach, which requires controlling such jumps (if irrelevant) from one period to the next, could provide a basis for a better apparatus to mobilise on the dynamic side for this early simulation period. We will return to this issue in *Section 3*.

Another positive side effect of such an approach could be to *feed EUROMOD with relevant intertemporal information* (e.g. past income and employment status, useful for the calculation of unemployment and parental leave benefits). More generally, the generation of less rough changes in income or population than implied by static ageing or a coarse reweighting procedure deserves consideration, at least to examine the extent to which such a less ad hoc environment can be reflected in EUROMOD results.

Moreover and not the least interest, *the dynamic side is well shaped to rely on an expanded weighted population* (see *Section 3*), making it easier to implement selections (e.g. of the unemployed) or changes in gross income, if they are based on external controls.

In summary, the dynamic side can help to provide useful information that would be difficult to generate on the static side.

Nevertheless, the SiD and DiS approaches have similar implementation requirements, as far as the present Task of technically linking EUROMOD with a dynamic model based on LIAM2 is concerned. Given the similarity of requirements, focusing on a DiS approach, as decided in this Task, will be an effective starting point for a possible future interest in a SiD implementation.

Let us note that keeping in mind, once again, the infrastructural nature of the InGRID-2 project, our objective is more to innovate at the level of the development and simulation tools mobilised in the background (EUROMOD-LIAM2 link) than to rewrite from scratch or on a large scale, or even to deepen, the procedures of short-term projections. What is at stake here is a method, considered as an intermediate good, rather than applications, considered as final goods. However, we are aware of these ultimate 'objectives', a dimension that may condition the sustainability of the proposed pathway, an explicit concern in this task.

2.2 The relevance of the LIAM2 platform for such a linkage

Another preliminary concern is the relevance of LIAM2 as a basis for development of the 'dynamic' side. Ahead the growing number of close applications presently developed with this platform (see

8 Liégeois *et al.* (2021).

Section 1), one may wonder whether the type of dynamics implied by LIAM2, namely discrete-time with cross-sectional population ageing, is of specific interest, especially taking into account the demanding effort involved in the development of such a framework. Although our main objective is to show the feasibility of a linkage between EUROMOD and LIAM2, we give here an overview of possible answers.

As evoked in *Section 2.1*, the exercise undertaken in the present Task is close to a 'nowcasting' exercise which requires a dynamic evolution of the individual socio-economic status (for example 'being at work' or not) and, possibly of the population structure itself (for example in case of rapid ageing or significant immigration). This may involve, as summarised by O'Donoghue and Loughrey (2014), the '*uprating of market incomes, the updating of a model to reflect policy changes* [in particular parameters], ... *the re-weighting of population sample (or static ageing) to account for changed population structure, ... without modelling the processes that drive these changes, ... or some dynamic ageing ... allowing individuals to change their characteristics due to endogenous factors from within the model, ... [simulating] changes to the population structure*'.

However and as O'Donoghue and Loughrey (2014) underline, 'static ageing cannot be used where there are no individuals in the sample in a particular state' and 'static ageing procedures are relatively well suited to short to medium term forecasts where changes in the structure of the population are small'. Therefore, 'in periods of in periods of significant volatility, reweighting may not be appropriate as it may rely excessively on small groups (Klevmarken, 1997)'.

It is clear that (full) dynamic modelling is well equipped for effective implementation of changes in such circumstances. Without entering into the debate on the best framework for short-term projections in such a context, which would be partly out of the scope of this task, this is a first reason to implement dynamic ageing, despite some drawbacks that will be discussed in *Section 5*.

A second benefit lies in the existence of already well-developed modules in LIAM2-based dynamic models that may be of interest for the implementation of the dynamic side of a simulation device for short term projections[9] and easily transferred between developers given the 'proximity' between the languages mobilised.

2.3 A EUROMOD-LIAM2 linkage among other implementations of short-term dynamics nowadays

Finally and before entering the implementation phase of this retained specific DiS approach, it is worth giving some indication of how this planned development fits into the whole sphere of applications performed through microsimulation for short term projections. *Table 1* summarises some of these applications. It is only a sample of possibilities, which could be extended, chosen here for their relevance to our objective of a short-term projection tool based on the instruments linked to EUROMOD and LIAM2.

Table 1 shows that several experiments cover part of the objectives we have in mind, yet none of them fits all criteria (basically EUROMOD and LIAM2).

As mentioned earlier, Liégeois and Dekkers (2014) combine EUROMOD and LIAM2 but with a somewhat different SiD-like objective than ours, besides being based on rather outdated versions of both platforms.

Our DiS approach also partially departs from the 'nowcasting' exercises undertaken by both the EUROMOD network and EUROSTAT (Gasior & Rastrigina, 2017; EUROSTAT, 2018 and 2019) which incorporate a series[10] of re-weighting (or calibration) of the distributions of employment status

9 Unemployment and work status, ageing, pensions, etc.

10 Each country team is choosing the most suitable method, given the information at its disposal and in consultation with its national agencies.

and work intensity as well as some basic technology[11] for calculating the employment income induced by these changes in statuses and work effort.

O'Donoghue *et al.* (2020) go a step further with an aligned or calibrated microsimulation approach extending the components of household disposable income considered in the adjustments (for example capital income) and proposing a sophisticated household income generation (IGM) model, developed by Sologon *et al.* (2018), to predict the distributional impact of the recent COVID-19 in Ireland. Overall, this approach could be seen as an in-between approach between static and dynamic ageing.

Table 1: Short-term micro-based projections summarising a selection of existing tracks, with a view on their relation to underlying modelling platforms

Ref	Source	Header	Static side	Dynamic side	Remarks
1	Liégeois and Dekkers (2014)	Combining EUROMOD and LIAM tools for the development of dynamic cross-sectional micro-simulation models: a sneak preview	EUROMOD (former version)	MIDAS_LU (former version)	- EUROMOD and LIAM2 at stake, yet former versions - More a SiD track
2	Gasior and Rastrigina (2017), grounding on several prior developments and papers (including by Lulescu, Leventi, Lopez-Vilaplana, Rastagrina and Sutherland [2016]), not reminded here	Nowcasting: timely indicators for monitoring risk of poverty in 2014-2016	EUROMOD	Now (ex post the publication mentioned here) possible in EUROMOD via 'LMA' Add-On: ad hoc routines for changes in 'population'(mainly employment status) and gross income (that may be country-specific), grounding for example on EU-SILC and LFS microdata, register data and macro-level statistics	- EUROMOD at stake - No LIAM2-related tools
3	EUROSTAT (2018-2019)	Flash estimates of income inequalities and poverty indicators for 2018 (FE 2018) - Experimental results	EUROMOD		- Closely related to the previous track (see ref. '2') - EUROMOD at stake, but no LIAM2-related tools
4	Dekkers, Tarantchenko and Van den Bosch (2019)	Medium-term projection for Belgium of the at-risk-of-poverty and social exclusion indicators based on EU-SILC	Dynamic model (inspired from MIDAS_BE) as a starter, together with other models of the Federal Planning Bureau in Brussels, and grounding on BE/EU-SILC data, with numerous ad hoc adaptations, for example for dealing with the specific short term indicators targeted		- LIAM2-based (partially) - No relation with EUROMOD
5	O'Donoghue, Sologon. Kyzyma and McHale (2020)	Modelling the distributional Impact of the COVID-19 crisis	NUI Galway microsimulation model (O'Donoghue *et al.*, 2018) as a starter and implementation of a dynamic ageing-related income-generation model (IGM) for changes in status and income		- Dynamic ageing-related method - No relation with EUROMOD* or LIAM2

* A previous implementation of the IGM tool was involving EUROMOD (Sologon *et al.*, 2018).

Dekkers *et al.* (2019) produce nowcasts for the BE/EU-SILC data, based on a full dynamic micro-simulation model developed through LIAM2 and inspired by MIDAS_BE, hence by dynamic ageing,

11 Basically, imputation.

as well as *ad hoc* adaptations for short-term projections. It also innovates by addressing the issue of the real divergence in SILC data between the year of income and the year of survey, by finding a way to reconcile them.

More broadly, previous developments undertaken with contributions from partners at the University of Antwerp and the University of Essex can feed into the present Task. In addition to the now-casting exercise already mentioned, we are thinking of the Hypothetical Household Tool (HHoT) in EUROMOD. This application allows the user to construct in a structured way and simulate *ad hoc* cases. Some variables needed for the simulations are then generated automatically or their values can be suggested to the user. Thus, some know-how on how to proceed in creating relevant and consistent information for 'new' cases has been accumulated, which is beneficial in our Task where the same kind of choices regarding several variables to be initialised, sometimes in an *ad hoc* way, are at stake.

3. Implementation of a Dynamic in Static (DiS) approach

We are now entering the implementation phase of a DiS approach, grounding on EUROMOD and a LIAM2-based dynamic model (here-after 'LIAM2 model' for simplicity). The objective is to dynamise some aspects of the population, while keeping in line with EUROMOD as our corner stone.

3.1 General guidelines

The dynamic LIAM2 model is inspired from MiDAS_LU 2020[12] and we are running EUROMOD version 3.2.4 as publicly available for Luxembourg on early 2021, although the present technical implementation may be applicable to another country. The corner stone of the DiS approach being the static side, most variables and the basic logic governing the developments are coming from EUROMOD. MiDAS_LU has then to be amended in order to make it EUROMOD-compatible with those respects while deriving the LIAM2 model.

The input microdata ('`LU_2018_A2.TXT`') refer to the survey year 2018 and income year 2017. The whole linkage is then simulated for the years 2018-2023, encompassing the recent past as well as the very short term to come, for illustration purposes.

We leave aside in the present implementation the difficulty arising from a **discrepancy between the survey and income years** in the input. Dekkers *et al.* (2019) address this issue with regard to the SILC data in an explicit and comprehensive proposal. For the time being, we stick to the EUROMOD practice and consider that 'year', 'period' or 'system year' are all referring to the *income* year and that the structural background information (demography, statuses, etc), despite mostly referring to the year after, are considered applicable to the income year as well.[13]

The first question arising is to decide **how the tasks should be distributed between the dynamic and static environments**, respectively. EUROMOD will be run over the chosen simulation period, 2018-2023, based on the annual 'populations'(structure and gross earnings) generated largely by the LIAM2 model. The latter is therefore expected to provide a population picture for each simulation year.

Basically, each side is supposed to generate what it is best suited for, with some priority given to the EUROMOD already built-in functionalities if these are relevant in the present combination. Therefore, the uprate of monetary variables (see also *Section 3.4*) and the tax-benefit computations can be left to EUROMOD, the LIAM2 model inheriting from unmet needs. As a starting point, the latter involve the question of a change in status on the labour market and impacts on gross earnings (out of the uprates).

Nevertheless and since years, EUROMOD was equipped with add-ons, one of them ('LMA') giving the possibility to generate some changes in unemployment and working statuses. Even more recently,

12 See Liégeois and Genevois (2015) for a comprehensive description of a former version of the model. Liégeois (2021) is referring to the more recent MiDAS_LU 2020 version that is the starter for the present development. MiDAS_LU was updated under MIGAPE project, co-funded by the Rights, Equality and Citizenship Programme of the European Union (2014-2020) via Grant Agreement no. 820798.

13 In this sense, relying on EUROMOD microdata is an advantage, as these are generated from SILC data through a series of STATA procedures that assess the compatibility of structural information with the observed income. For example, if some employment income is recorded when there is no month worked or a status not compatible with such income, this is treated as a matter of principle.

due to the COVID-19, some possibility to shock the employees' gross income (through changes in the number of months and hours worked during the year), together with a partial financial compensation by the government or the firms, was introduced. Therefore, we should stick to our basic principle and keep on the shoulders of EUROMOD what is actually provided on this side.

However, EUROMOD relies on a sample (based on EU-SILC) of residents, which may lead to some difficulties when changing the population (including statuses) and imputing earnings over time. These may need to be controlled for their averages or other characteristics, through parameters derived from external statistics referring to subgroups (see 'alignments' in *Section 3.2*). If so, it may become demanding to undertake a selection of individuals (facing a transition in the labour market, for example) and an adaptation of monetary variables on a non-expanded sample. The latter might be mistakenly considered, by the routines governing change, as a weighted population.

To avoid such a possible source for distortion, we should organise the simulation on an **expanded population**. 'Clones' are therefore created, their number depending on the weight attached to each observation in the sample feeding EUROMOD. Such an 'augmentation' task is easily manageable on the dynamic side and, therefore, we transfer these processes involving variations controlled through time for their means or several characteristics per subgroup, from EUROMOD to the LIAM2 model.[14]

In particular, **changes in employment status** (including from employment to unemployment and vice versa) and subsequent provisions on gross employment income, a priori achievable by the LMA module as actually provided in EUROMOD, are rather realised on the dynamic side, and a large population. On this side, the parameters are more easily derived for such a configuration and the implementation procedures are better controlled.

The same type of concern could be raised by the new EUROMOD instrument implementing an additional **COVID-19 shock on gross income** as mentioned above (policies 'YEMCOMPTIME_LU' and 'YEMCOMP_LU'). In their simplest implementation, these policies impose a fixed loss in terms of months worked per year per employee ('$MC_MY'parameter = '3'in 2020), with partial monetary compensation ('$MC_RRATE'= '0.8') identical for all impacted workers.

However, it was not clear what specific considerations regarding sub-groups or overall averages were governing the choice of parameter values underlying this instrument and we could not afford to pursue this specific issue, which would certainly be relevant if broader considerations were at play. However, this is not essential here given our objective of establishing a technical linkage rather than a policy-oriented exercise, and we can decide to defer the issue. Therefore, ignoring the issues arising from possible controls at this level and due to time constraints, we have opted to maintain these policies on the EUROMOD side, imposing an *ad hoc* reduction in months worked per worker of maximum three months in 2020, then only two months in 2021, one month left out in 2022 and back to the original work intensity in 2023.

More generally, we **distinguish** in the present implementation **the technology relating to the linkage itself**, which is at stake here, **from a fine modelling of policies or socio-economic changes**, which serves here only to illustrate our main technical concern. Insofar as a methodological choice on the policies implemented does not significantly affect the way a link is to be organised, we focus our efforts on the latter, which becomes a priority in such a limited time investment.

3.2 An overview of how the linkage is implemented

Table 2 below shows briefly how the linkage, our main purpose, is organised.

14 We could also start by expanding the sample, then turn to EUROMOD and run it on the weighted population. However, we are not sure of the background information on which some processes and calibrations are built in EUROMOD and we had little time to overcome this problem. Therefore, we decided to go for the dynamic side, where we have more control over the underpinnings.

After some preparatory work (*Block I*), the dynamic envelope is taken into account and allows the LIAM2 model to generate a series of **images of the population** (its structure and some gross incomes or benefits), one per simulated period. These images are then fed into EUROMOD for more detailed calculations on income, taxes and benefits (*Block II*). Finally, some analytical work is undertaken on the simulation results (*Block III*). The last column indicates the steps (hereafter '#') referred to in the next Tables 3 and 4.

As mentioned in *Section 3.1*, we derive from MiDAS_LU some routines that are interesting for the linkage. These are obviously simplified, compared to a full dynamic model, and must be made compliant with EUROMOD in terms of the **variables** mobilised (name, coverage), at least for the dimensions that play a role on both sides. Other variables may also be required on one side (for example capital income in EUROMOD) that are generally not taken into account in the dynamic sphere. If this is the case, careful monitoring of these differentiated needs is at stake. This may also imply partial rewriting of some modules, in particular 'policies' or 'systems' in EUROMOD and 'processes' in the LIAM2 model.

However, one of the first functionalities mobilised on the dynamic side is new and realises the **expansion** of the population inherited from the EUROMOD input dataset (see *Section 3.1*): the 10,493 observations of the sample become 574,336 for the income year 2017 and the resident population in Luxembourg. Thereafter, this population will comprise, after several and progressive demographic adjustments, 645,435 individuals in 2023.

Table 2: An overview of tasks involved in a EUROMOD-LIAM2 model linkage general structure

Blocks	Contents	Steps # as referenced in other *Tables* below
I	PREPARATORY WORK	
	Inputs for the dynamic side / LIAM2 model (EUROMOD set of variables, alignments, other parameters and methodological choices)	#1 - #2 (*Table 4*)
II	MAIN CORPUS	
	Write and run the dynamic envelope / LIAM2 model (input 2017, simulation 2018-2023)	#3 - #4 (*Table 3*)
	Expansion of input microdata (2017), Ageing, Migrations, Education/Unemployment/Working/Retirement statuses, Gross income (out of COVID-19 loss on months worked and induced compensations), Gross pension rights, Reporting (including yearly exhaustive populations as inputs for the static side)	(#4.1 – #4.10, *Table 3*)
	Static ground (EUROMOD)	#5 (*Table 4*)
	Uprating of monetary variables, COVID-19 loss on months worked and induced reductions in gross employment income (mitigated by compensations), Taxes and Benefits, Reporting (including yearly exhaustive populations as inputs for the analysis)	
III	ANALYSIS OF OUTCOMES	#6 (*Table 4*)

As discussed in *Section 3.1*, a model can be designed in such a way that the key dimensions follow at the macro level and for successive years (sometimes also by subgroups) the assumptions made by a group of experts or the results derived from external sources, a procedure called '**alignment**'. (Liégeois, 2021; Dekkers *et al.*, 2015, Liégeois & Genevois, 2015). For example, MiDAS_LU 2020, hence the LIAM2 model considered here, is based on projections, for Luxembourg, produced for the '2018 Ageing report'(European Commission, 2017) by the 'Working group on Ageing populations and sustainability'(AWG). These figures also involve EUROSTAT demographic projections (fertility rates, life tables, etc.). On top of this, national sources may be mobilised, for example the unemployment rates by age group and gender, as observed until recently; we then have to decide on some extrapolation of these figures for the coming years, 2022 and 2023.

Among the 'other methodological choices' mentioned in the *Block I* of *Table 2*, we can cite this **share of tasks between EUROMOD and the LIAM2** model addressed in *Section 3.1*.

Going further, we then have to decide which **processes to consider on the dynamic side** (see *#4*). Apart from unemployment and labour status, discussed in *Section 3.1*, these involve, on the demographic side, population ageing (including deaths and births) and migrations. As soon as these dimensions are taken into account and in order to create a coherent overall picture, we are confronted with changes in education and retirement status, as well as with the appearance of new retirement benefits (old age and survivors, without taking into account the new, less numerous, invalids in the current implementation of the link).

It should be noted that the analyst may want to limit the set of processes to be executed, for example by running only the 'unemployment and working' process, while leaving the population unchanged (that is, no ageing, no migration, etc.). This may help to determine the impact of a specific factor on the overall outcome of the simulation.

3.3 The comprehensive path for implementation

We are now deepening those contents and developing in *Table 3* all tasks implied by this EUROMOD-LIAM2 linkage, with most important aspects commented. More explanations may also come later while discussing the dynamic side (see *Table 4*).

Block I referenced in *Table 2* is operationalised through the Stata 'PRIOR.DO' routine essentially (*#2*). This combines the EUROMOD input data 'LU_2018_A2.TXT'and useful EUROMOD output variables for the year 2017 (coming from *#1*) for **generating raw input microdata** for the LIAM2 model: 'HOUSEHOLDS.CSV' and 'PERSONS.CSV'. As we proceed, an important step is to rearrange the **individual identifiers** so that these follow a continuous and regular path, starting at '0', then '1', etc. `Prior.do` also generates variables and statistics that will feed tables of interest on the dynamic side later on, for example **alignment** (see *Section 3.2*) or **imputation tables**, the latter providing average employment earnings by subgroups and the share of self-employed among the employed (cf. *#4.8* in *Table 4*), but also a series of parameters dealing with pensions (#4.6 and #4.10, see for example *Section 3.4*).

Appendix a1.1 is listing the topics covered in `Prior.do`.

The LIAM2 model is entering the dance in *steps #3* and *#4*. The routine 'CREATE_INPUT.YML' allows us to **create the input microdata** that will be used by the LIAM2 model downstream. This data file must be in HDF5[15] format (with extension '*.h5'). Then our core business, the LIAM2-based 'MODEL_WP8.YML', performs **all tasks to be run on the dynamic side**. These have already been mentioned in *Section 3.2* and will be further specified by *Table 4* below (see *Section 3.4*).

Once the populations have been generated through the LIAM2 model (one microdata file per simulation year), we are ready to take advantage of them through the EUROMOD package (#5).

15 'HDF5'stems for Hierarchical Data Format 5. This is a freely available set of libraries designed to store and organise large amounts of numerical data (de Menten *et al.*, 2014) and appeared to be quite efficient in terms of a reduction in the running time of large models. See HDF group (https://www.hdfgroup.org) for more information.

Table 3 All tasks involved in a EUROMOD-LIAM2 linkage

#	Tool	Basic Actions	More information	Remark
1	EUROMOD	Running with last input microdata available, where survey year is 2018 and income year is 2017 (LU_2018_a2.txt)	'LU_2018_a2.txt'has been created by EUROMOD team from EU-SILC and complementary national information (10,493 individuals)	The income year is privileged, considering that all info relating to that year (cf. EUROMOD and see Section 3.1)
2	STATA	- Preparing microdata for LIAM2 model input (#3) (Prior.do) - Files needed for creation of a LIAM2 input created (Households.csv and Persons.csv) - Preparing information for main LIAM2 model (#4) (Prior.do)	- Variables renamed and identifiers made continuous (0, 1, 2, ...) - Statistics useful for further developments below generated	- Some preparatory LIAM2 code elaborated (automatised) in EXCEL, from present STATA outcomes (WP8.xlsm) - Idem for alignment/imputation tables (births, deaths, migrations, un/employment, education attainments, average earnings and pensions), from present Stata outcomes, AWG projections and external macro sources
3	LIAM2	Creating input microdata for LIAM2 (Create_Input.yml)	- Inputs: Persons.csv and Households.csv files - Output: in HDF5 format ('*.h5')	- List of households and persons as in initial microdata, hence not expanded yet
4	LIAM2	Adapting the model from MiDAS_LU and run for the 'dynamisation' of population, including a change in unemployment and working status (hence gross earnings resulting from this adjustment) (Model_WP8.yml)	- Processes: input, expansion, initialisation, ageing/deaths, births, education, immigration, retirement, unemployment/ working (including impact on gross earnings) and pension rights - Output created (LU_2018/…/24_a4.txt)	- Some preparatory LIAM2 code elaborated (automatised) in EXCEL (WP8.xlsm) - Therefore, no change in unemployment/working status will be done on the EUROMOD side downstream - 'LU_2018_a4.txt'is an expanded population (574,336 persons)
5	EUROMOD	- (Additional) COVID-19 related shock on gross earnings on 2020-2022, for employees - Taxes & benefits (if simulated)	- Reduction in time worked (from 3 months over the year in 2020, up to 1 month in 2022), but loss in gross earnings partially compensated by the government (EUROMOD basic implementation) - Generates OUTPUTS files, where 'years' going from 2017 up to 2023)	- We could implement changes in unemployment through the EUROMOD LMA Add-On, but choose to do it here in step #4 (hence through LIAM2), rather, to work on an expanded population for selections
6	STATA (and EXCEL)	Analysis ex post and showing up outcomes (Post.do, decomposed in Post_one_wave.do, Post_all_waves.do and Post_assembling.do)	- Deriving several variables useful for analysis downstream - Poverty (by subgroups) and inequality indices - Net incomes as resulting both from LIAM2 (#4) and EUROMOD (#5)	Then, EXCEL for deriving outcome tables and graphs (WP8.xlsm)

This involves **uprating the monetary variables** (from the year 2017, which is the base year for the valuation of monetary amounts in the dynamic model; see also *Section 3.4*), **having an additional impact on the employment income** as a result of COVID-19 (see *Section 3.1*), and calculating **taxes and benefits,** taking into account all those aspects. The outcomes are generated, that is one file 'LU_<YEAR>_STD.TXT' per simulation year.[16]

16 And by EUROMOD 'system' implemented, if several systems are designed for the same year, if there is a need to adjust the calculations while changing some aspects in the policies.

Finally, an **analysis of the results** is performed and any desired tables or graphs generated (#6), through the software of your choice, in this implementation Stata ('POST.DO') and EXCEL ('WP8.XLSM'). We will give an overview of such outcomes in *Section 4*. *Appendices a1.2* and *a1.3* summarise the topics covered in 'POST_ONE_WAVE.DO' and 'POST_ASSEMBLING.DO'. *Appendix a1.4* lists the sheets involved in 'WP8.XLSM'. This file is generic to the current implementation as a whole and also deals with other aspects of previous processing (#1), including some code writing for the LIAM2 model which is facilitated though the EXCEL file.

3.4 A focus on the dynamic side of the linkage

As the dynamic model, which is part of our EUROMOD-LIAM2 link, is a less common activity, we now clarify in *Table 4* the tasks involved in *step #4*. This report does not aim to be exhaustive, but if more details are needed, the interested reader can examine the LIAM2 model that is part of this deliverable. Only a few important methodological aspects are highlighted below.

Before looking at *Table 4*, let us first point out that a LIAM2 model is basically structured as shown in *Figure 1*, which may help to situate the list of tasks mentioned in this new table.

The heart of the simulation lies in the third part, '**simulation:**'. This is where the input and output files are referenced and the number of periods for the simulation is set. The simulation: part also lists all processes to be run and their order of execution. Therefore and referring to *Table 4*, the 'BIRTH' process will follow the 'SURVIVAL' one which itself comes before the 'EDUCATION' process, with perhaps intermediary technical steps not mentioned here.

Each process is described in detail in the '**entities:**' part which contains a declaration of the types of entities mobilised in the simulation (here 'PERSONS' and 'HOUSEHOLDS'), including their characteristics. Then comes on top the '**globals:**' part which introduces and references (if external files) a series of macro parameters and alignment tables (see *Section 3.2*), such as 'AL_P_UNEMPLOYMENT' indicating what proportion of active persons is unemployed, by gender, age group and year.

With this framework set, the LIAM2 model, when run, has to initialise the **education attainments**[17] (*#4.1* in *Table 4*, completed by *#4.5* for subsequent periods in the simulation) which affect for example employment earnings as soon as the person is working. For individuals having left school, the education attainment is known already. Otherwise, each person may be imputed a level of education that will be reached in the future, based on the distribution of education attainments observed for the [30-34] age group in 2017 (microdata input year). It is worth to mention that, given the proportions to be imposed by subgroups, the imputation procedure is building on an expanded population (see *Section 3.1*). Then, for each level of education attainment, a school-leaving age is determined, with a possibility left in the model for a part of the students to prolong their studies a little beyond this ordinary maximum age corresponding to their level of education attainment. These ages and proportions have been calibrated in order to maintain at school a proportion for the whole population not too far from that observed in 2017.

17 EUROMOD variables 'DEH_EVER', built on 'DEH'.

Figure 1 The basic structure of a model elaborated trough LIAM2

```
globals:
    periodic:
        - AGE_LOWEST_IN_EDUCATION : int
        - ...
    AL_P_UNEMPLOYMENT:
        path: AL_P_UNEMPLOYED_EUROMOD_FROM_EUROSTAT_WEIGHTED.csv
        ...

entities:
    household:
        ...
    person:
        fields:
            - dag: int              # age
            - dgn: bool             # gender : "0" if female, "1" if male
            - m_id: int             # mother's identifier
            - ...
        macros:
            FEMALE:   dgn == 0
            ...
        links:                      # first one ("mc") : Mother to Children
            mc: {type: one2many, target: person, field: m_id}
            ...
        processes:
            init_block_person_arrangements_post_expand():   # cf. Task 4.1
                - deh_ever: ...
            ...
            unemployment_and_working_process():             # cf. Task 4.8
                ...
                # the process is here DECLARED/SPECIFIED only (see also Figure 2)
            ...

simulation:
    processes:
        - person: [
              init_block_person_arrangements_post_expand,
              ...
              immigration_person_process,       # cf. Task 4.6
              ...
              unemployment_and_working_process, # cf. Task 4.8
              ...
              ]
    input:
        path:  "INPUT_DATA"
        file:  "EUROMOD_LU_2018_A2_ENTRY_for_WP8.h5"        # "HDF5" format
    output:
        path:  "OUTPUT_DATA"
        file:  "FINAL_OUTPUT_EUROMOD_FROM_LU_2018_A2_ENTRY_for_WP8.h5"  # "HDF5" format
    start_period: 2018              # first simulated period
    periods: 6                      # 2018 - 2023
```

Table 4 Processes implemented on the dynamic side (LIAM2 model, inspired from MiDAS_LU 2020)

#	Process or Block	Action	More details
4.1	Input & Init Blocks	Reading input micro data and initial computations (income period 2017)	- Expansion of initial dataset, taking into account individual weights (coming from SILC/EUROMOD data), through the cloning of persons and households, including adapted links between those clones - Introduction and computation of relevant variables, in particular in relation with education attainments (achieved already or imputed, if still a student) and pension rights
⇓ Core of the simulation, evolving through time (2018 – 2023, in the present implementation) ⇓			
4.2	Ageing procedure	Age and incremental variables updated	Incremental: number of years of residence in Luxembourg
4.3	Survival	Going on alive of dying	- Probabilities aligned on AWG-2018 figures (by gender, age and year) - If a death, inter-personal links and marital status updated, on top of a transfer of information about possible survival pension benefits to the partner still alive
4.4	Birth	New births and creation of persons	- Probabilities aligned on AWG-2018 figures (by gender, age and year) - If a birth, new person with relevant characteristics (not necessarily inherited from the parents)
4.5	Education	Education attainments and status	- Status based on the maximum education level to be reached by this person (distributed based on gender) and age - If has just left school, the young graduated is given 'another status'(to be fixed downstream)
4.6	Migration	Creating new immigrants (persons and households)	- Number aligned on AWG-2018 figures (by gender, age and year) - We consider net migrations (immigration – emigration, fortunately positive for most cells), in conformity with AWG-2018 projections - Migrants cloned from existing households, with a selection rule based on mean age of adults, citizenship and origin (migrants or not) and time since the household having been a clone upstream if ever - Characteristics of the new clones not necessarily purely 'inherited', especially with respect to pension rights and statuses
4.7	Retirement	Updating the status	If this person is in position to become retired
4.8	Unemployment and Work	Updating the status and fixing earnings	- Unemployment based on a risk function and proportions by gender, age and year, derived from external statistics (up to 2021) or hypotheses - Working if was active the period before or is young graduated (#4.5), while being neither unemployed nor retired (#4.7) - If a new worker, proportion of self-employed and civil servants as well as gross labour earnings based on statistics from input data 2017 (no uprate of earnings, to be done on EUROMOD side downstream) - Other characteristics updated, most often in conformity with EUROMOD treatments with that respect
4.9	Pension rights	Fixing benefits	Old-age and survival benefits (no new disabled)
4.10	Output Block	Outputting outcomes	Summarising info (1 line par year) and comprehensive info (1 file per year, all persons) to be served as input for EUROMOD simulations downstream)
5	EUROMOD to be run, for COVID-19-related shock on work intensity and gross incomes (out of unemployment and working status, fixed on the dynamic side), uprate of monetary amounts (when relevant) and taxes and benefits computations (out of gross pensions)		

It should be mentioned that such a **'smoothing' principle** has been retained at different levels in the implementation. Our objective is to link a static model, hence a structure in population and earnings *observed* for the input year, and a dynamic one *simulated* for short term projections. The simulated outcomes depend on a series of methodological choices, modelling approximations and lack of information for a perfect fit with the 'reality'. Such a configuration with a static-dynamic linkage and a short term dimension altogether requires that the simulation results be anchored as much as possible on previous outcomes or observations: **the transition from an observed to a simulated population should be as smooth as possible**, a concern already mentioned in *Section 2.1*. This is particularly relevant for variables that should not reasonably be expected to change too quickly or strongly in the short term, such as the level of education and the proportion of students in the population.

The initialisation block of the LIAM2 model is also involving some **preparatory work with respect to pension rights** (#4.1). These depend on periods (months) worked or assimilated, among which the time spent at school. Since this school information is missing from the input database or generated by the model as soon as initialised (see above), we have to derive and update these pensionable periods. Another initialisation concerns certain aspects of future survivor pensions if a spouse dies during the simulation period. The survivor's pension of the still living partner will depend on the situation of the deceased spouse which has to be registered somewhere 'in case of'. This aspect of survivor's pensions taken into account through *steps #4.1* up to *#4-3*.

In case of a **birth** (*#4.4*), the baby inherits from many characteristics of its parents, but some of them are obviously imputed *ex nihilo*: the gender (randomly attributed), the age (set to '0'), the education attainment (not yet fixed, therefore 'missing'), the number of years of residence '`drsyy`' in Luxembourg (set to '0') which may become important for the immigration process, the time spent on work so far, etc.

This is followed by the **migration** process (*#4.6*) which consists of adding migrants, based on a selection rule for determining which households and individuals will be cloned in this respect. However, although the new migrants are clones of current residents, they cannot be civil servants and their employment income will be reduced accordingly if the 'original' person has such a status. This is done by applying to employment income a ratio of employment income of civil servants to that of non-civil servants, as derived from the input data (*#2*). Similarly for pension entitlements of the cloned person downstream, if 'inherited' from the 'original' person upstream, those are also limited. The ratio applied is derived from the input data, again by comparing the pension income of residents who have lived abroad for a certain time or not (`drsyy` == the age).

Next, the **retirement process** (*#4.7*) determines whether an individual meets the condition to be retired. Here again, we adopt a smoothing approach for limiting the number of new retirees in the very first years, if any (assumed) irrelevant jump in the simulation between the entry year (2017) to the first simulated one (2018) is observed. The input data show that if the general pension rules were applied, a significant share of the population would 'suddenly' retire in the 2018 period. Determining why so many individuals in a position to be retired, given our limited knowledge, are not, is beyond the scope of the present implementation and we limit this drawback by making it possible to retire later than the legal retirement age (65 years in Luxembourg) for a certain period of time.

Step #4.8 operationalises a change in **unemployment and work statuses**, as discussed in *Section 3.1* and illustrated in *Figure 2* below. Among members of the active population, a 'risk' of being unemployed is calculated based on a prior econometric prior analysis ('`employment_score:`', which shows the ease with which such a functionality is implemented). Retirees, including the new pensioners who are known at this stage of the simulation (*#4.7* executed earlier), since they are no longer part of the labour force, hence their socio-economic status is not equal to 1 (farmer), 2 (self-employed), 3 (employed) or 5 (unemployed). On the other side, new graduates, who are also known at this stage of the simulation (*#4.5*), are incorporated into the active population, which is certainly a strong and simplifying assumption.

Figure 2 **Implementation of the 'UNEMPLOYMENT_AND_WORKING' process (#4.8)**

```
unemployment_and_working_process():

    - unemployed: False
    - unemployment score: "-1"
    - working: False

    # "DAG" in the risk function below is age, "DAG2" is dag * dag,
    # "NCH_011" is the number of children aged [0-11]
    # The "LOGIT_SCORE" function is a logistic with a random part,
    # which then returns a probability (see LIAM2UserGuide.pdf)

    - unemployment_score:
        if( FEMALE and WORKING_1 and not(PENSIONER or
            YOUNG_GRADUATED or NEW_IMMIGRANT) and ...,
            logit_score( 0.5476218 * dag -0.0079123 * dag2
                         + 0.5241817 * nch_011 -7.079126
                       ),
            unemployment_score
          )
        ...

    - unemployed: if( (les==1 or les==2 or les==3 or les==5
                       or YOUNG GRADUATED)
                       and ...,
                       align( unemployment score,
                              AL P UNEMPLOYED,
                              leave = CIVIL_SERVANT_WORKING_1
                            ),
                       unemployed
                    )

    - working: if( (les==1 or les==2 or les==3 or les==5
                    or YOUNG_GRADUATED) and not(UNEMPLOYED),
```

Then, a standard alignment procedure is followed: the 'active' individuals are sorted according to their risk level and a proportion of them, derived from the alignment table 'AL_P_UNEMPLOYMENT' alignment Table is marked as unemployed for that period.

A by-product of this change in unemployment status is that the working population is now identified. Then, if an individual was not working already in the previous year, it has to be determined whether this person is self-employed or employed, and among the employed, who is a civil servant. In the present implementation, all these calculations are based on proportions determined by gender and education level for the year of the input data (2017, #2) and assumed constant over time. We also assume that former civil servants who are still active retain their status for sure. These are simplifying assumptions, but we are less interested in the current implementation of technologies with respect to the socio-economic dimensions than in the static-dynamic link which is virtually unaffected by these technologies, also given the short time period considered (1 + 6 years). In the case of a 'newly unemployed' person, the number of months in this status is considered to be 12 or less, depending on the situation observed for this person in the previous year.

These status considerations also have a clear impact on the derivation of gross employment earnings. In the case of a new worker, earnings are imputed based on the gender and education averages as observed for the year of entry data (2017, #2).

One last important task is left for the end of the dynamic simulation, which objective is to calculate **new pension entitlements** for newly retired persons (after *#4.7*) or new widows (after ageing in *#4.2*).

Note that all monetary amounts are valuated here with reference to a common period which is the year of the input data (2017). These amounts, whether employment income or pension benefits, will be uprated when running downstream the EUROMOD simulation (*#4.10* and *#5*). The static aspect will also involve a calculation of unemployment benefits, where applicable.

This concludes our main methodological reporting report on the linkage between EUROMOD and LIAM2. We now propose a few outcomes in *Section 5*, before concluding in *Section 6*.

4. A few initial outcomes

We briefly illustrate some results of the static/EUROMOD-dynamic/LIAM2 coupling. This implementation is at the heart of the present Task and our aim here is to give an overview of what such an instrument can offer in terms of analyses complementary to the usual 'static' results.

A key objective of a microsimulation model is to explore the distribution of income and well-being within a population, identifying and examining the main factors that play a role in such a distribution. The tool then allows for a comparative analysis of alternative policies, with a focus on population subgroups that may be particularly affected by these changes.

We first look at the overall averages and indices, before moving on to examine cross-sectional effects on subgroups (that is, some types of households).

4.1 Monetary averages and overall inequality indices

Figure 3 and *4* show the evolution of monetary variables on the short-term (2017-2023) and their implications in terms of inequalities, in general through the 'Gini' index (increasing with the degree of inequality) or when focusing on the left side of the income distribution, with the 'at-risk-of-poverty rate'. The 'well-being' is here represented by the so-called 'equivalised income' which looks at both income and needs assessed at the household level.[18]

18 The equivalised income is derived from the residence household's net total disposable income which is divided by a coefficient taking into account economies of scale and then attributing to each household member a weight corresponding to the household composition (following the OECD modified scale, '1' is the weight given to 1st adult in the household, '0.5' is attributed to each supplementary adult and '0.3' to each child). Finally, the same equivalised income, as fixed at the household level, is attributed to each member of the household.

Figure 3 **Evolution of several indicators on the short term if a CHANGE IN UNEMPLOYMENT and work statuses only are considered, together with a shock (and compensation) on gross income**

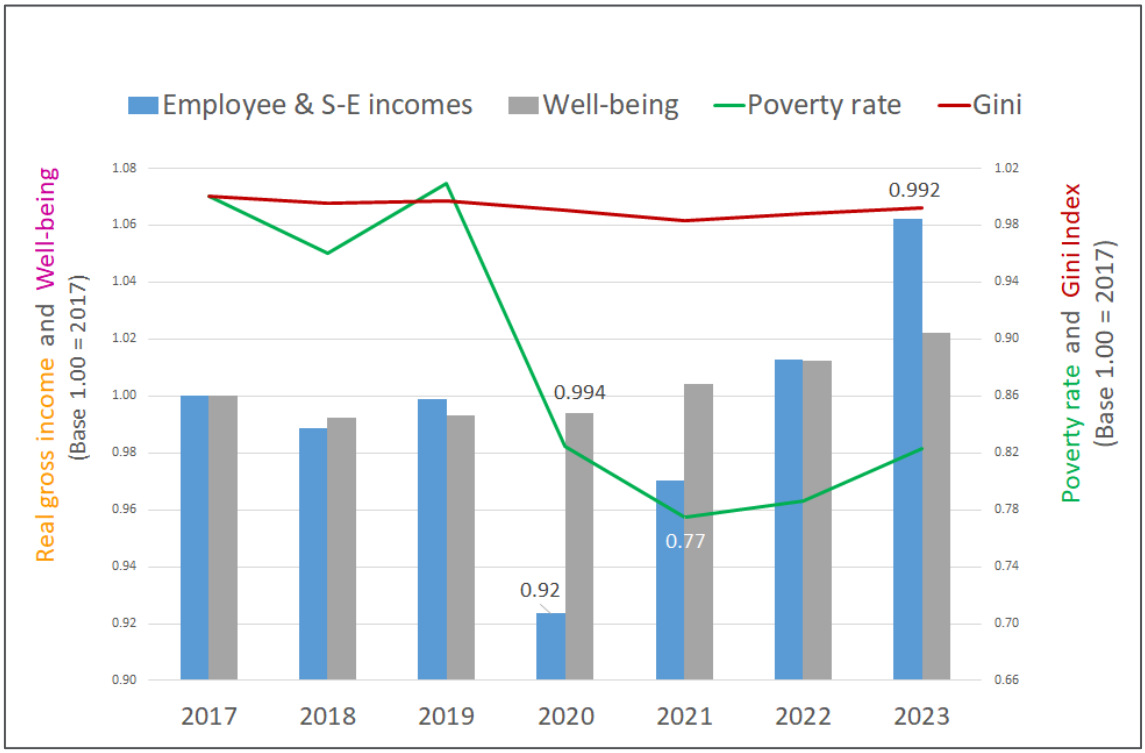
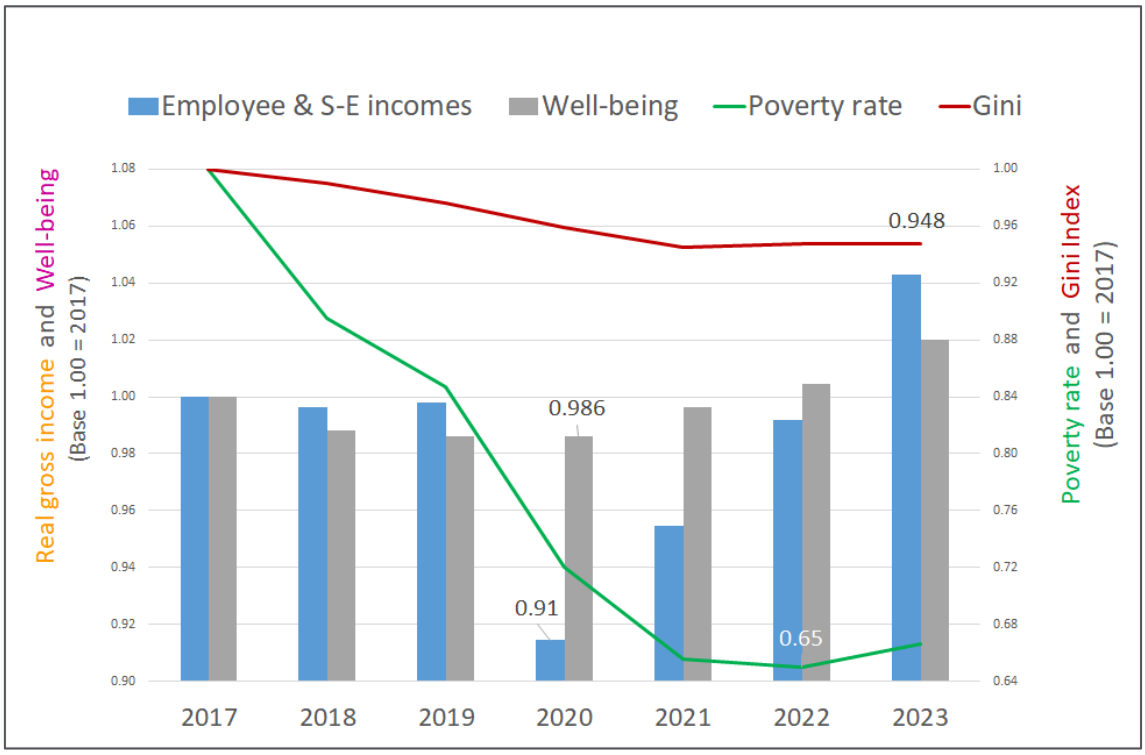


Figure 4 **Evolution of several indicators if ALL PROCESSES from *Table 4* are activated, including demographic changes and their downstream impacts at large (for example on gross pensions)**



The standard poverty rate here represented is telling which percentage of the population benefits from an equivalised income lower than 60% of the median of individual equivalised incomes.

Figure 3 shows the results when only unemployment and work status are implemented in the simulation, in addition to the usual static content of EUROMOD, including an additional COVID-19 shock on gross income and some monetary compensations (see *Section 3.1*) and the usual uprate of monetary amounts. At this stage, no demographic change (ageing, education, migrations and pensions, in short) is considered over the period. With reference to *Table 4* in *Section 3.4*, this implies that steps #4.1, #4.8 and #4.10 are performed on the dynamic side, with the other steps disabled for this intermediate review.

The gross employment income is facing a significant drop in 2020, as expected, while recovering progressively, in conformity with hypotheses formulated in *Section 3.1* and despite the compensation of income losses considered. Thanks to the tax-benefit system at large, which is classically damping the fluctuations in the original gross income, the well-being does not seem that much affected. We underline here that EUROMOD for Luxembourg is considering, in its standard implementation, a full take-up of social assistance benefits. Perhaps surprisingly, inequality appears to show some reduction over the period, on average, whether measured by the Gini inequality index or the poverty rate. This would certainly merit further analysis, beyond the scope of this technical implementation, to better understand the underlying forces that generate such a picture.

Figure 4 reproduces the same type of view when demographic changes (ageing, education, migration and pensions, in short) are considered over the period. Therefore, *steps #4.2* up to *#4.7* and *#4.9* of *Table 4* are also activated.

In spite of a largely completed set-up in terms of additional simulations and modelling choices, we observe that the fluctuations of gross income and well-being, although slightly affected (the minimum in 2020 is more pronounced), do not differ much from the simple framework. The well-being in 2020 is 98.6% of the 2017 level when all processes are taken into account, whereas it reaches 99.4% if unemployment, labour and gross income only are 'dynamised'. The relative loss is more than doubled (from 0.6% to 1.4%), but it is still quite limited when considering absolute levels.

4.2 A transversal view on the structure of inequalities

An indication of more significant changes in income distribution (compared to averages) can be seen in *Figures 3* and *4*. The Gini index is more affected if all processes are activated (0.948 in 2017 base, compared to 0.992 in *Figure 3*) and, even more strikingly, the poverty rate falls more significantly (minimum 0.65 in 2017 base, 0.77 in *Figure 3*). This will definitely merit further investigation as well, in order to better understand the effect of the modelling assumptions made regarding the structural effects resulting from changes in status, income and population.

Tables 4 and *5* (for the well-being), and *6* and *7* (for the poverty rates) are rapidly illustrating the point, while completing information on averages shown previously. We are now comparing the initial year of simulation (2017) and the year showing up the strongest impacts, due to the COVID-19 (2020).

In all *tables* are considered on top (and in blue) the cumulative effect of factors affecting the overall variables. Of course, the progression thus highlighted may path dependent, but we can already get a sense of what is going on in the background in this way.

The first message is that the impact of a change in unemployment and work status alone (from [a] to [b]), which is here distinguished from the additional shock induced by COVID-19 on gross incomes and compensations, is quite limited in terms of both the well-being and the overall poverty rate.

Table 4: **WELL-BEING (real equivalised income) - 2020 versus 2017 Impact through time, by factor influencing and by household category *NO DEMOGRAPHIC CHANGES* (cf. *Figure 3*)**

	2017	2020
Impact of successive factors		
[a] Uprate only (real terms)	1.000	1.007
[b] = [a] + change in Unemployment		1.006
[c] = [b] + change in Gross Income and Compensation		0.994
Impact of Household Composition	% whole	2020/2017
Whole Population	*1.000*	*-0.6%*
Single (<65)	0.984	-1.1%
Single (65+)	0.947	0.4%
Single with dependent(s)	0.730	0.7%
Couple+ 0 dep	1.162	-0.2%
Couple+ 1-2 dep	0.925	-1.0%
Couple+ 3+ dep	0.800	-2.0%

Table 5: **WELL-BEING (real equivalised income) - 2020 versus 2017 Impact through time, by factor influencing and by household category *ALL PROCESSES* (cf. *Figure 4*)**

	2017	2020 / 2017	***"Factor" impact*** *(2020/2017, then b/a, c/b and d/c)*
Impact of successive factors			
[a] Uprate only (real terms)	1.000	1.007	*100.7%*
[b] = [a] + change in Unemployment		1.006	*-0.2%*
[c] = [b] + change in Gross Income and Compensation		0.994	*-1.1%*
[d] = [c] + Demography		0.986	*-0.8%*
Impact of Household Composition	% whole	2020/2017	
Whole Population	*1.000*	*-0.6%*	
Single (<65)	0.984	-0.2%	
Single (65+)	0.947	-4.6%	
Single with dependent(s)	0.730	1.1%	
Couple+ 0 dep	1.162	-2.2%	
Couple+ 1-2 dep	0.925	0.4%	
Couple+ 3+ dep	0.800	-6.1%	

Table 6: **POVERTY RATES - 2020 versus 2017 Impact through time, by factor influencing and by household category *NO DEMOGRAPHIC CHANGES* (cf. *Figure 3*)**

	2017	2020
Impact of successive factors		
[a] Uprate only (real terms)	11.3	10.4
[b] = [a] + change in Unemployment		10.2
[c] = [b] + change in Gross Income and Compensation		9.0
Impact of Household Composition		2020 - 2017
Whole Population	*11.3*	*-2.3*
Single (<65)	15.7	-0.9
Single (65+)	8.7	1.7
Single with dependent(s)	31.8	-10.7
Couple+ 0 dep	6.3	-0.6
Couple+ 1-2 dep	11.3	-2.9
Couple+ 3+ dep	14.1	-3.5

Table 7 **POVERTY RATES - 2020 versus 2017 Impact through time, by factor influencing and by household category *ALL PROCESSES* (cf. *Figure 4*)**

	2017	2020	***"Factor" impact*** *(2020-2017, then b-a, c-b and d-c)*
Impact of successive factors			
[a] Uprate only (real terms)	11.3	10.4	*-0.9*
[b] = [a] + change in Unemployment		10.2	*-0.2*
[c] = [b] + change in Gross Income and Compensation		9.0	*-1.2*
[d] = [c] + Demography		7.8	*-1.1*
Impact of Household Composition		2020 - 2017	
Whole Population	*11.3*	*-3.5*	
Single (<65)	15.7	-3.6	
Single (65+)	8.7	6.3	
Single with dependent(s)	31.8	-18.3	
Couple+ 0 dep	6.3	-1.4	
Couple+ 1-2 dep	11.3	-4.7	
Couple+ 3+ dep	14.1	-1.4	

The second main observation is that the gain in well-being between 2017 and 2020 due to the uprate of employment income, i.e. +0.7% ([a]), is doubled... but in the negative direction, if we consider the simplified framework without demographic changes (-1.3% in *Table 4*, [a] to [c]), and even tripled if demographic changes are added (-2.1% in *Table 5*, [a] to [d]). However, those impacts remain limited in absolute terms, as mentioned in *Section 4.1*. On the contrary, poverty rates are systematically reduced, whether considering uprates of incomes alone or other dimensions on top. They fall by

2.3 p.p. if no demographic change is considered (*Table 6*, [a] to [c]), 3.5 p.p. otherwise (*Table 7*, [a] to [d]).

Looking at the different types of households, we see the usual differences between subgroups. Residents living in a single household with dependants are worse off in terms of well-being (0.730 compared to the general average in 2017) and much more at risk of poverty than if they belong to other household types (+20.5 p.p. compared to the general situation). On the other hand, couples without dependants have a higher standard of living (on average).

What is striking here are the large differences in the changes from 2017 to 2020 if the 'full' dynamic tool is mobilised (that is. with demographic adjustments), or not. For example, considering all processes, singles with dependants lose 18.3 p.p. in terms of poverty rate (*Table 7*), whereas the reduction in poverty is 10.7 p.p. in the latter configuration (no demographic change considered, *Table 6*). Other differences also appear, particularly for single people over 65. This is our third finding, which will also merit further consideration beyond this technical note, particularly with regard to the methodological choices that may have resulted in such significant impacts or discrepancies.

5. Conclusions: where we are, limits and sustainability

The current task of Work Package 8 of the EU-InGRID-2 project has achieved its main objective. An **operational link between the EUROMOD static model and the LIAM2 development and simulation platform for dynamic microsimulation** has been designed and implemented, and first results obtained. These developments could from now certainly benefit further from external experiments, including the avenues indicated by Dekkers *et al.* (2019).

The present coupling has been carried out in the Luxembourg context, in a '**Dynamic in Static**' (DiS) perspective and for mainly short-term projections (nowcasting or a little more). However, the whole procedure should be easily extended to a '**Static in Dynamic**' (SiD) configuration if desired, where EUROMOD would act as a static module in a complete model designed for a longer period.

The prospects for these approaches are real, given the broad partnership established by EUROMOD over the years, the need for appropriate (and perhaps more sophisticated) tools to dynamise aspects of static simulations, the closeness of the underlying logic architectures of EUROMOD and LIAM2, and the extension of the use of LIAM2 as a dynamic microsimulation platform over the years. In addition, the consistent and value-added input microdata built by EUROMOD from EU-SILC and complementary national data could also become a relevant and standard basis for dynamic (or combined) microsimulation developers.

With regard the static-dynamic linkage, it appears that if the complementary effects of a global approach, integrating demographic changes and downstream adaptations, are not always striking on average, at least in the context taken into account in the present implementation, they still merit some additional analyses, particularly with regard to the **distributional impact of changes**. Moreover, the **methods** used here to deal with several aspects in the simulation of the dynamic side (for example the selection of individuals to undergo a transition), would deserve to be used more generally, possibly in a more restricted implementation.

However, these outcomes, when they happen to fluctuate little, should be linked to the sampling issue underlying the constitution of our input databases. Given this initial sampling operation, we know that our **results** are **delivered under uncertainty** up to a certain point, thus with confidence intervals, identified or not. Therefore, the question of whether our dynamic fluctuations fall within (hypothetical) confidence intervals, and are therefore 'dynamically significant' , or not, remains open, a concern that is by no means specific to the present implementation.

On the other side, outcomes showing more significant impacts or divergences would also merit a more in-depth examination beyond this technical note, particularly with regard to the **methodological choices**, linked either to the construction of the model or to the external (macro) data considered, which may have led to such results.

Before closing this Task, it is important to point out some **drawbacks** that might imply further developments in the future, even if the needs mentioned here are not necessarily of immediate interest.

The LIAM2 based models involve discrete time dynamic ageing frameworks with most often a periodicity of one year for the results. In this sense, some useful **sub-periodic information** embedded in EUROMOD and derived from EU-SILC data may be more difficult to take into account in a dynamic (LIAM2) environment. Social policies often refer to events or quantities relating to a part of

the year (e.g. employment income or unemployment status), rather than to a whole period. If the generation of such fine-grid information is required on the dynamic side, this may involve complex additional processing, unless this dimension is ignored and annual results or roughly imputed sub-period quantities are provided to EUROMOD. The latter option has been chosen in the present implementation.

The **type of households** considered may also be a concern. If a dynamic model is built on nuclear households only (a single person or an adult couple with their children), careful consideration of the implications for EUROMOD (dealing with residential households, possibly with several 'generations' living together) may be necessary.

The final weakness highlighted here is the often unavoidable **loops** that occur in processing: EUROMOD may generate an outcome (for example, unemployment benefit) which should be an input for the dynamic model which in turn will provide some information (for example, a means-tested pension benefit) useful for EUROMOD to make its own computations. The exercise carried out in the present task did not avoid such loops,[19] fortunately with limited implications. But we could not fully address the issue, which arose late.

19 A first loop may appear if we need to target the head of a household (any type of household) on the dynamic side, for example, in *steps #4.1* to *#4.10* of *Table 4*. A decision rule for such a designation is available in EUROMOD which will produce this information, but only in *step #5*. We could overcome this problem by coding the EUROMOD decision rule mentioned above on the dynamic side. However, if this algorithm involves complex calculations or intermediate results to be provided by the static side (benefits, taxes, etc.), we are back to the original question.
Another loop has arisen when calculating gross survivors' pensions on the dynamic side. It may be necessary to combine them with other types of incomes in order to respect the maximum and minimum regulations. Again, if such information (in our case the gross employment income resulting from the income shock treated on this side, see *Section 3.1*) is to be provided by EUROMOD, we should consider reverting to the dynamic module after the static simulation.
Fortunately, the expected impact of these (unprocessed) loops in our implementation, as far as we could see, is very limited. However, we may have unintentionally ignored other more problematic configurations of this type.

appendix 1 Topics covered through the Stata routines and excel sheets, in brief

a1.1 'Prior.do' – Topics

```
I. GENERALITIES
***************

      1.    INIT

      2.    DEFINING CONSTANTS

      3.    CREATING '.DTA'& FIRST '.CSV'FROM 'LU_2018_A2.TXT'

                  1.    REORDERING VARIABLE NAMES
                  2.    SECONDLY, LOADING EUROMOD OUTPUT DATA, SAME YEAR
                  3.    EMPHASISING VARIABLES WHICH ARE PRESENT IN EUROMOD
                        INPUT but NOT IN THIS EUROMOD OUTPUT FILE
                  4.    ADDING FLAG VARIABLES
                        (for variables in INPUT EUROMOD)
                  5.    REORDERING VARIABLE NAMES
                  6.    SAVING
                  7.    THIRDLY, LOADING INFO FROM EUROMOD OUTPUT DATA
                        for SYSTEM 2020, and SAVING AGAIN (for taking into
                        account shock on YEM and YEMMY due to crisis)
                  8.    BASIC STATISTICS and COUNTING NUMBER OF VARIABLES

      4.    ADDING FIELD 'PERIOD' and VARIABLE 'BCA01'

      5.    RE-ASSESSIG IDENTIFIERS SO THAT STARTING FROM '0'
            AND NO UNUSED VALUES (0, 1, 2, 3, ...)

      6.    'IDHH'REARRANGED => 'H_ID'CREATED

      7.    'IDPERSON'REARRANGED => 'ID'CREATED

      8.    'IDPARTNER'REARRANGED => 'S_ID'CREATED

      9.    'IDMOTHER'REARRANGED => 'M_ID'CREATED

      10.   'IDFATHER'REARRANGED => 'F_ID'CREATED

      11.   'TU_HOUSEHOLD_LU_HEAD_ID'REARRANGED => 'HEAD_HH_ID'CREATE

II. ADAPTATIONS and GENERATING USEFUL INFO & TABLES FOR ALIGNMENTS
*****************************************************************

      100. RENAMING <vars>_IN_RAW to INITIAL (EUROMOD) NAMES

      101. WEIGHTS

      102. LABELS, FOR EXISTING VARIABLES

      103. NEW VARIABLES & TYPOLOGIES

                  1. CLASS_AGE_EUROMOD_LIAM2(_FINE)
                  2. ADULT_OECD (>=14) & CHILD_OECD (<14)
                  3. EQUIVALENT WEIGHT
                  4. EQUIVALENT (DISPOSABLE) INCOME FOR RESIDENCE HOUSEHOLDS
                  5. CREATING USEFUL THRESHOLDS AND DECILES

      104. POVERTY

      105. INEQUALITY
```

106. SUMMING UP OUTCOMES BY HOUSEHOLD &
CREATING HOUSEHOLD-RELATED VARIABLES

III. LAST ADAPTATIONS and OUTPUTTING FOR LIAM2 / YML PROGRAMS
****

150. LISTING AND CREATING => PERSONS.CSV

151. LISTING AND CREATING => HOUSEHOLDS.CSV

IV. DERIVING SOME USEFUL PRIOR RESULTS
****

200. ABOUT UNEMPLOYMENT (USEFUL FOR 'ALIGNMENTS'=>
by 'DGN'& 'CLASS_AGE_EUROMOD_LIAM2(_FINE')

201. ABOUT WORKERS

1. MONTHS WORKED ('CLASS_MONTHS_YEMMY/YSEMY',
by 'DGN' and 'CLASS_AGE_EUROMOD_LIAM2')
2. EARNINGS (BY 'DGN', 'CLASS_AGE_EUROMOD_LIAM2_FINE',
'DEH'& 'LCS')
3. OLD-AGE PENSIONS (BY 'DGN', 'DEH' and AGES)
4. SURVIVAL PENSIONS (BY 'DGN', 'DEH' and AGES)
5. 'LCS'

202. OTHERS

1. 'DRGMD'(DEMOGRAPHIC: Region: Middle density)
2. 'DCZ'(DEMOGRAPHIC: Citizenship)

a1.2 'Post_One_Wave.do'(= a Single Year Simulation) – Topics

```
I. GENERALITIES
***************

       1.  INIT
       2.  DEFINING CONSTANTS
       3.  LOADING EUROMOD OUTPUT DATA (FROM LIAM2_BASED INPUT) FOR INPUT YEAR
       4.  WEIGHTS
       5.  LABELS, FOR EXISTING VARIABLES

II. GENERATING USEFUL INFO & TYPOLOGIES FOR ANALYSIS
****************************************************

     10. NEW VARIABLES & TYPOLOGIES

            1. CLASS_AGE_EUROMOD_LIAM2_FINE
            2. CLASS_AGE_EUROMOD_LIAM2
            3. ABOUT 'UNEMPLOYMENT'

     11. 'OECD'EQUIVALENT DIMENSIONS

            1. ADULT_OECD (>=14) & CHILD_OECD (<14)
            2. NUMBER OF OECD-ADULTS (>=14) IN RESIDENCE HOUSEHOLDS
            3. NUMBER OF OECD-CHILDREN (<14) IN RESIDENCE HOUSEHOLDS
            4. OECD EQUIVALENT WEIGHTS
            5. DISPOSABLE INCOME FOR RESIDENCE HOUSEHOLDS
            6. EQUIVALENT (DISPOSABLE) INCOME FOR RESIDENCE HOUSEHOLDS
            7. POVERTY THRESHOLDS
            8. DECILES OF EQUIVALENT INCOME

     12. TYPES OF HOUSEHOLDS

            1. VARIABLES 'IS_PARTNER'& 'NB_PARTNERS'
            2. VARIABLE 'PARTNERSHIP'
              (= '1'WHEN 'SINGLE'[WHATEVER # OF DEPENDENTS] &
                   '2'WHEN 'COUPLE')
            3. NUMBER OF FISCALLY DEPENDENTS (CHILDREN OR NOT)
              IN RESIDENCE HOUSEHOLDS
                   'IS_DEPENDENT'(individual) & 'NB_DEPENDENTS'
                   (/ RESIDENCE household)
                   'CLASS_NB_DEPENDENTS'
            4. 'WORKER'/ 'NOT WORKER'
            5. 'NB_WORKERS'& 'CLASS_NB_WORKERS'
               WITHIN THE RESIDENCE HOUSEHOLD
            6. HOUSEHOLD TYPOLOGY BY FUCHS (AUSTRIA),
              pp 21 - BUT WHILE SPLITTING 'SINGLE'(MORE OR LESS 65)

III. ANALYSIS OF OUTCOMES
*************************

     20. POVERTY

            1.     PROPORTION OF POOR BY GENDER
            2.     PROPORTION OF POOR BY AGE GROUP
            3.     PROPORTION OF POOR BY NB_WORKERS & 'FUCHS'TYPOLOGY


     21. INEQUALITY

            1.     GINI OF EQUIVALENT (DISPOSABLE) INCOME
            2.     ATKINSON FOR EQUIVALENT (DISPOSABLE) INCOME
```

a1.3 'Post_Assembling.do' – Topics

```
I. GENERALITIES
***************

     1.     INIT

     2.     DEFINING CONSTANTS

                  1.     FIRST, TO BE CHANGED SOMETIMES
                  2.     SECOND, 'INVARIANT'NAMES
                  3.     FINALLY, SHOW UP

II. MERGING USEFUL INFO & TYPOLOGIES FOR ANALYSIS
************************************************

     10. LOADING BASIC OUTPUT ANALYSIS FILE

                  0.     BASICS
                  1.     LOADING BASIC FILE
                  2.     RENAMING RELEVANT VARIABLES AND SAVING



     11. IMPORTING RELEVANT VARIABLES FROM OTHER
         'EUROMOD_OUTPUT_ANALYSIS ... . dta' FILES
            (as from POST_ONE_WAVE.DO, FOR ALL SIMULATIONS)

                  1.& 2. IMPORTING => 'MERGING'& RENAMING RELEVANT VARIABLES

                          ... => ALL IN ONE

                  3.     THEN SAVING

III. ANALYSIS
*************

     20. SIMPLE EVOLUTION THROUGH TIME

                  1.     MONETARY VARIABLES
                  2.     CATEGORICAL VARIABLES
                  3.     POVERTY LINE

     21. INEQUALITY THROUGH TIME

                  1.     GINI & Co
                  2.     POVERTY, BY GENDER

     22. CROSSING CATEGORIES

                  1.     UNEMPLOYMENT, BY GENDER (over ACTIVE POPULATION)
                  2.     POVERTY, BY GENDER
                  3.     POVERTY, BY TYPE OF HOUSEHOLD (INDIVIDUAL LEVEL)
                  4.     WELL-BEING, BY TYPE OF HOUSEHOLD (INDIVIDUAL LEVEL)
```

a1.4 'WP8.xlsm'– List of Sheets and their Functionalities in a Word

Sheet Name	What's in ?
Analysing Outcomes	
EVOLUT THROUGH TIME - MIGRA - G	All processes ⇒ *Graphs*
EVOL THROUGH TIME - UNEMP - G	unemployment_and_working process only ⇒ *Graphs*
EVOLUTION THROUGH TIME - MIGRAT	*Analysis of outcomes* from Post_Assembling.do (A.3) if *all processes* on the dynamic side
EVOLUTION THROUGH TIME - UNEMP	*Analysis of outcomes* from Post_Assembling.do (A.3) if *unemployment_and_working process* only, on the dynamic side
OUT - STATA_ASSEMB - MIG - RAW	Raw outcomes from Post_Assembling.do (A.3) if *all processes* on the dynamic side
OUT - STAT_ASSEMB - UNEMP - RAW	Raw outcomes from Post_Assembling.do (A.3) if *unemployment_and_working process* only, on the dynamic side
OUTPUT - LIAM2 - AGGREG - RAW	Synthetic outcomes (1 line per year) from LIAM2 model
From Raw to LIAM2	
INIT VARS after CLONE - MIGRAT	*Supporting code* for variable update if new clone
INIT VARS after NEW - BIRTH	*Supporting code* for variable update if new birth
AL_NET_MIGRATIONS_LU18_MiDAS_21	*Alignment* Table for net migrations
TAB_PENS_OLD_SILC_2018_NOT_UPR	*Imputation* Table for poapups
TAB_YSE_WEIGHTD_SILC_18_NOT_UPR	*Imputation* Table for yse
TAB_YEM_WEIGHTD_SILC_18_NOT_UPR	*Imputation* Table for yem
AL_P_UNEMPLOYED_WEIGHTED	*Alignment* Table for unemployment rates
YML - CSV_OUT as EUR_INP – HEAD / YML - CSV_OUT as EUR_INP – CONT	*Generating code* for output from LIAM2 model – Headers/-Contents
LIAM CREAT from STATA for LU_18	*Generating code* for Creation of microdata input
DICTIONARY & DRD for LU_2018_A2	EUROMOD DRD (Dictionary info)
EUROMOD Info	
UPRATES, PENSIONS	
EARNINGS	YEM, YSE
UN-EMPLOYMENT_RAW	Macro info on unemployment
POP - from MiDAS_2021	Population structure
MIGRATIONS _NET - from MiDAS_21	Net migrations
EDUCATION	Education attainments
Docu	
SOURCES - LARGE LIST	List of papers
METHODS & OUTCOMES	Short-term dynamics

Bibliography

Dekkers G., Tarantchenko K. and K. Van den Bosch (2019), *Medium-term projection for Belgium of the at-risk-of-poverty and social exclusion indicators based on EU-SILC*, Working Paper 3-19, Federal Planning Bureau:Brussels, https://www.plan.be/uploaded/documents/201903110928260.WP_1903_11838.pdf

Dekkers Gijs, Raphaël Desmet, Nicole Fasquelle and Saskia Weemaes (2015), The social and budgetary impacts of recent social security reform in Belgium, in Ioana Salagean, Catalina Lomos & Anne Hartung, '*The young and the elderly at risk: Individual outcomes and contemporary policy challenges in European societies*', Intersentia. ISBN 978-1-78068-343-0. Chapter 6, pp. 129-158.

de Menten G., Dekkers G., Bryon G., Liégeois Ph. and C. O'Donoghue (2014), *LIAM2: a New Open Source Development Tool for Discrete-Time Dynamic Microsimulation Models*, Journal of Artificial Societies and Social Simulation, 17 (3) 9, http://jasss.soc.surrey.ac.uk/17/3/9/9.pdf

European Commission, 2017, The 2018 Ageing Report: Underlying Assumptions and Projection Methodologies. Economic and Social Affairs Institutional Paper 065. November 2017. doi:10.2765/286359 (online), https://ec.europa.eu/info/sites/info/files/economy-finance/ip065_en.pdf [21/03/2021]

EUROSTAT (2019), Flash estimates of income inequalities and poverty indicators for 2018 (FE 2018) - Experimental results, Luxembourg, https://ec.europa.eu/eurostat/documents/7894008/8256843/Flash-estimate-of-income-inequalities-and-poverty-indicators-experimental-results-2018.pdf.

EUROSTAT (2019), Flash estimates of income inequalities and poverty indicators for 2018 (FE 2018) - Methodological note, Luxembourg, https://ec.europa.eu/eurostat/documents/7894008/8256843/Methodological-note-2018.pdf.

EUROSTAT (2018), Flash estimates of income inequalities and poverty indicators for 2017 (FE 2017) - Experimental results, Luxembourg, https://ec.europa.eu/eurostat/documents/7894008/8256843/Flash-estimates-of-income-inequalities-and-poverty-indicators-experimental-results-2017.pdf.

EUROSTAT (2018), Flash estimates of income inequalities and poverty indicators for 2017 (FE 2017) - Methodological note, Luxembourg, https://ec.europa.eu/eurostat/documents/7894008/8256843/Methodological-note-2017.pdf.

Gasior K. and O. Rastrigina (2017), Nowcasting: timely indicators for monitoring risk of poverty in 2014-2016, EUROMOD Working Paper Series, EM 8/16, https://www.iser.essex.ac.uk/research/publications/working-papers/euromod/em8-16.pdf.

Klevmarken, N. A. (1997). Modelling Behavioural Response in EUROMOD, Cambridge Working Papers in Economics 9720, Faculty of Economics, University of Cambridge.

Philippe Liégeois, Mahdi Benjelloul, Stefano Boscolo, Leonardo Calcagno, Riccardo Conti, Gijs Dekkers, Nataša Kump, Jinjing Li, Boris Majcen, Tomáš Miklošovič, Amilcar Moreira, Ingrid Schockaert, Miroslav Štefánik, Krisztián Tóth, Vincent Vergnat and Peng Zhan (2021), Spotlight Report on Dynamic Microsimulation, WP6, Task 2, D6.9, Leuven, InGRID-2 project 730998 – H2020.

Liégeois Ph. (2021), Projections of the Gender Pension Gap in Luxembourg using MIDAS_LU 2020, Mind the GAp in PEnsions (MIGAPE) Project, LISER: Luxembourg, http://www.migape.eu/pubs/Preliminary_FINAL_VERSION_23MAR2021_MIGAPE_WP3_GPG_projections_LU_AWG18_HEADER_reviewed.pdf.

Liégeois Ph. and Genevois A-So. (2015), 'MiDLAS - WP-B - 1 - 27 AUG 2015 - DYNAMIC MICROSIMULATION MODEL - LuDMi_I - TECHNICAL REPORT.pdf', LISER:Luxembourg.

Liégeois Ph. and G. Dekkers, (2014), *Combining EUROMOD and LIAM Tools for the Development of Dynamic Cross-sectional Microsimulation Models: a Sneak Preview*, in G. Dekkers, M. Keegan and C. O'Donoghue (eds), New Pathway in Microsimulation, Burlington: Ashgate.

Leulescu A., Leventi C., Lopez-Vilaplana C., Rastrigina O. and H. Sutherland (2016), Flash estimates of income distribution indicators for the European Union: methods, assessment and future prospects), 34th General Conference, Dresden, Germany.

Navicke J., Rastrigina O. and H. Sutherland (2013), Nowcasting Indicators of Poverty Risk in the European Union: A Microsimulation Approach, Paper presented to EU-SILC Conference, Vienna.

O'Donoghue C., Sologon D. Kyzyma I. and J. McHale (2020), Modelling the Distributional Impact of the COVID-19 Crisis, IZA Discussion Paper No 13235, https://www.iza.org/publications/dp/13235/modelling-the-distributional-impact-of-the-covid-19-crisis.

O'Donoghue C., Loughrey J. and D. M. Sologon (2018). Decomposing the drivers of changes in inequality during the great recession in Ireland using the fields approach, *The Economic and Social Review*, 49 (2, Summer), 173-200.

O'Donoghue C. and J. Loughrey (2014), *Nowcasting in Microsimulation Models: A Methodological Survey*, Journal of Artificial Societies and Social Simulation 17 (4) 12, https://www.researchgate.net/publication/267667488_Nowcasting_in_Microsimulation_Models_A_Methodological_Survey.

O'Donoghue C., Hynes S. and J. Lennon (2009), *The Life-Cycle Income Analysis Model (LIAM): A Study of a Flexible Dynamic Microsimulation Modelling Computing Framework*. International Journal of Microsimulation 2 (1): 16-31, https://microsimulation.org/IJM/V2_1/IJM_2_1_2.pdf.

Sologon D.M., Van Kerm P., Li J. and C. O'Donoghue (2018), Accounting for differences in income inequality across countries: Ireland and the United Kingdom, LISER Working Paper No 2018-01.

Sutherland, H. and Figari, F. (2013), *EUROMOD: the European Union tax-benefit microsimulation model*, International Journal of Microsimulation, 6(1): 4-26, https://microsimulation.org/IJM/V6_1/2_IJM_6_1_Sutherland_Figari.pdf.

InGRID-2
Integrating Research Infrastructure for European expertise on Inclusive Growth from data to policy

Referring to the increasingly challenging EU2020-ambitions of Inclusive Growth, the objectives of the InGRID-2 project are to advance the integration and innovation of distributed social sciences research infrastructures (RI) on 'poverty, living conditions and social policies' as well as on 'working conditions, vulnerability and labour policies'. InGRID-2 will extend transnational on-site and virtual access, organise mutual learning and discussions of innovations, and improve data services and facilities of comparative research. The focus areas are (a) integrated and harmonised data, (b) links between policy and practice, and (c) indicator-building tools.
Lead users are social scientist involved in comparative research to provide new evidence for European policy innovations. Key science actors and their stakeholders are coupled in the consortium to provide expert services to users of comparative research infrastructures by investing in collaborative efforts to better integrate microdata, identify new ways of collecting data, establish and improve harmonised classification tools, extend available policy databases, optimise statistical quality, and set-up micro-simulation environments and indicator-building tools as important means of valorisation. Helping scientists to enhance their expertise from data to policy is the advanced mission of InGRID-2. A new research portal will be the gateway to this European science infrastructure.

This project is supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 730998.

More detailed information is available on the website: www.inclusivegrowth.eu

Co-ordinator
Monique Ramioul

KU LEUVEN HIVA
RESEARCH INSTITUTE FOR WORK AND SOCIETY

Partners

TÁRKI Social Research Institute Inc. (HU)
Amsterdam Institute for Advanced Labour Studies – AIAS, University of Amsterdam (NL)
Swedish Institute for Social Research - SOFI, Stockholm University (SE)
Economic and Social Statistics Department, Trier University (DE)
Centre for Demographic Studies – CED, University Autonoma of Barcelona (ES)
Luxembourg Institute of Socio-Economic Research – LISER (LU)
Herman Deleeck Centre for Social Policy – CSB, University of Antwerp (BE)
Institute for Social and Economic Research - ISER, University of Essex (UK)
German Institute for Economic Research – DIW (DE)
Centre for Employment and Work Studies – CEET, National Conservatory of Arts and Crafts (FR)
Centre for European Policy Studies – CEPS (BE)
Department of Economics and Management, University of Pisa (IT)
Department of Social Statistics and Demography – SOTON, University of Southampton (UK)
Luxembourg Income Study – LIS, asbl (LU)
School of Social Sciences, University of Manchester (UK)
Central European Labour Studies Institute – CELSI (SK)
Panteion University of Social and Political Sciences (GR)
Central Institute for Labour Protection – CIOP, National Research Institute (PL)

InGRID-2

Integrating Research Infrastructure for European expertise on Inclusive Growth from data to policy Contract N° 730998

For further information about the InGRID-2 project, please contact
inclusive.growth@kuleuven.be
www.inclusivegrowth.eu
p/a HIVA – Research Institute for Work and Society
Parkstraat 47 box 5300
3000 Leuven
Belgium

```
#            ##########################################################
#            #  LUXEMBOURG "EUROMOD_PLUS" MODEL - VERSION 5 (2021)  #
#            ##########################################################
#
#                 ##############################################
#                 # Initially from EU-funded MiDLAS PROJECT  #
#                 #            AS ON 24 AUG 2015             #
#                 #                                            #
#                 #      Further elaborated and completed,    #
#                 #          for EU-InGRID-2 - WP8          #
#                 #                                            #
#                 #       THIS VERSION AS ON 9 OCT 2021       #
#                 #      ******************************        #
#                 #                                            #
#                 #              FOR "LU_2018_A2"              #
#                 #              ----------------              #
#                 #                                            #
#                 #            Philippe Liégeois           #
#                 #                LISER                       #
#                 ##############################################
#
#
#
#  ********************************************************************
#  * CAREFUL => ITEMS BELOW MIGHT BE CHANGED FROM ONE RUN TO ANOTHER !!! *
#  *            ========================================================= *
#  ********************************************************************
#
#  - MACRO "HEADERS_FOR_EXHAUSTIVE_OUTPUT" in ENTITY "PERSON"   (look for "MAY_BE_CHANGED_A")
#  - CSV OUTPUT FOR FULL POPULATION                             (look for "MAY_BE_CHANGED_B")
#
#
#  SOURCE
#  ======
#   DERIVED FROM
#      A] VERSION "CONVENTION IGSS" - 14 MAR 2012
#      B] MIDAS_BE - 18 DEC 2014
#      C] EU-MiDLAS PROJECT (2013-2015)
#      D) Then completed (2016-2018), hence VERSIONS 526_B and 603 (starting point here)
#      E) ... and augmented through EU-MIGAPE (2019-2021)
#      F) Now a basis for the WP8 MODEL for AUGMENTING ("DYNAMIZING") EUROMOD (EU-InGRID-2 Project)
#
#
#  TARGET
#  ======
```

```
#  - RESIDENT POPULATION
#
#
#  DATA
#  ====
#  - INPUT FROM EUROMOD, INCOME 2017 (SILC 2018 / "LU_2018.A2.TXT") => 1st YEAR OF SIMULATION = 2018
#
#
#  PARAMETERS
#  ==========
#  - AWG 2018 if needed
#
```

```
globals:
```

```
    periodic:
```

```
        path: "MACRO_AV (with INFLATION)  -  From MIDAS_2020  -  XRX621_MIG19_ 3 MAR 2021  -  COMMA SEP  -  WITH EUROMOD
UPRATES as on 21 AUG 2021.csv"
        transposed: true
```

```
        fields:
```

```
        # period is implicit
        - AGE_LOWEST_IN_EDUCATION : int
```

```
        - EUROMOD_IPCN : float
        - EUROMOD_NO_UPR : float
        - EUROMOD_UPR_2 : float
        - EUROMOD_UPR_3 : float
        - EUROMOD_UPR_5 : float
        - EUROMOD_UPR_6 : float
        - EUROMOD_UPR_7 : float
        - NI_B16 : float
        - PEN_ANTICIPATED_AGE_HIGH_F : int
        - PEN_ANTICIPATED_AGE_HIGH_M : int
        - PEN_ANTICIPATED_AGE_LOW_F : int
        - PEN_ANTICIPATED_AGE_LOW_M : int
        - PEN_DUR_ANTICIPATED_HIGH_COMPULS : int
        - PEN_DUR_ANTICIPATED_HIGH_TOT : int
        - PEN_DUR_ANTICIPATED_LOW_COMPULS : int
        - PEN_DUR_LEGAL_MIN : int
        - PEN_DUR_LEGAL_TOT : int
        - PEN_DUR_MIN_BENEFIT: int
        - PEN_LEGAL_AGE_F : int
        - PEN_LEGAL_AGE_M : int
        - PEN_REFER_AMOUNT_YEAR_PROD_N : float



    AL_P_BIRTH:
        path: AL_P_BIRTH_MiDAS_2020_AWG_2018.csv
        type: float


    # Migration
    AL_NET_MIGRATION:
        path: AL_NET_MIGRATION_WP8_FROM_MiDAS_2021_LU_2018.csv
        type: float

    AL_P_SURVIVAL_F:
        path: AL_P_ALIVE_F_MiDAS_2020_AWG_2018.csv
        type: float

    AL_P_SURVIVAL_M:
        path: AL_P_ALIVE_M_MiDAS_2020_AWG_2018.csv
        type: float

    AL_P_UNEMPLOYED:
```

```
        path: AL_P_UNEMPLOYED_EUROMOD_FROM_EUROSTAT_WEIGHTED.csv
        type: float

    TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED:
        path: TAB_PENSIONS_OLD_AGE_SILC_2018_NOT_UPRATED.csv
        type: float

    TAB_YEM_BASE_2017_NOT_UPRATED:
        path: TAB_YEM_WEIGHTED_SILC_2018_INCOME_2017_NOT_UPRATED.csv
        type: float

    TAB_YSE_BASE_2017_NOT_UPRATED:
        path: TAB_YSE_WEIGHTED_SILC_2018_INCOME_2017_NOT_UPRATED.csv
        type: float









entities:




    household:



        fields:


            # [WP8] EMPTY (out of ID and PERIOD) in this version

            # period and id are implicit

            - dwt_int: int
```

```
        #
        # THEN, ADDED (not in INPUT)
        #

        - bidon:                                {type: int, initialdata: false}
        - cloned_expand_upstream_hh_id:         {type: int, initialdata: false}
        - cloned_immigration_upstream_hh_id:    {type: int, initialdata: false}
        - dwt_final:                            {type: int, initialdata: false}
        - dwt_int_in_raw:                       {type: int, initialdata: false}
        - has_a_clone_expand_upstream:          {type: bool, initialdata: false}
        - has_a_clone_immigration_upstream:     {type: bool, initialdata: false}
        - hh_age_max:                           {type: int, initialdata: false}
        - hh_citizenship:                       {type: int, initialdata: false}
        - hh_clone_expand_child_id:             {type: int, initialdata: false}
        - hh_clone_immigration_child_id:        {type: int, initialdata: false}
        - hh_head_id:                           {type: int, initialdata: false}
        - hh_ilsorigy_max:                      {type: float, initialdata: false}
        - hh_immigrant:                         {type: bool, initialdata: false}
        - hh_last_clone_immigration_period:     {type: int, initialdata: false}
        - hh_nb_copies_remaining:               {type: int, initialdata: false}
        - hh_now_to_be_cloned_for_immigration:  {type: bool, initialdata: false}
        - hh_to_be_clone_expand:                {type: bool, initialdata: false}
        - hh_wave_migration:                    {type: int, initialdata: false}
        - immigration_candidate:                {type: bool, initialdata: false}
        - immigration_score:                    {type: float, initialdata: false}
        - impact_immigration_age:               {type: float, initialdata: false}
        - nb_adults:                            {type: int, initialdata: false}
        - nb_persons:                           {type: int, initialdata: false}
        - year:                                 {type: int, initialdata: false}
        - year_of_expansion:                    {type: int, initialdata: false}
        - year_of_immigration:                  {type: int, initialdata: false}



macros:



    # Input data YEAR
    BASE_PERIOD: "2017"
    BASE_PERIOD_IN_AL_NET_MIGRATION: "2016"
```

```
            # Now about "scoring" for IMMIGRATION =>
            # (NB : as used here, 10 years younger for adults (on avg in HH) makes us "gaining" 1.0 in score,
            #       which is equivalent to 1 more year since last clonage (additive scoring up to there)
            FACTOR_AGE: "1.0"
            # Impact of "CITIZENSHIP" (see variable "HH_CITIZENSHIP") on MIGRATION SCORE
            FACTOR_CITIZENSHIP: "10.0"
            # This factor below used in LU ONLY if EXPAND not mobilized (hence POP and need for MIG lower)
            FACTOR_INFLATOR_POP: "1.0"
            # Impact of "INTENSITY" of MIGRATION (both/sum Nb of years in LU and Nb of non-"LU citizens"
            FACTOR_MIGRANTS_IN_HH: "1.0"
            # Random part in IMMIGRATION SCORES
            FACTOR_MIGRANTS_RANDOM: "1.0"
            # MAXIMUM "WEIGHT", used for "WHILE" LOOP below
            MAX_DWT_INT: "600"
            # Max population considered for CSV OUTPUTS, now or later (273.749 as on 29 SEPT 2021)
            MAX_NUMBER_OF_RECORDS: "500000"
            # DURATION of MIGRATION considered as "moderate" (in years)
            MIGRATION_SHORT_WAVE_DURATION : "15"
            # % of population selected for candidacy to clonage, taking into account that
            #   all most "FOREIGN-CITIZENs" with RECENT WAVE (< 15= years) AND NOT SELECTED BEFORE are selected
            # => little room left for other HHs
            PROPORTION_SELECTION: "0.5"



        links:



            persons: {type: one2many, target: person, field: h_id}

            # Added for EXPANSION & IMMIGRATION
            hh_clone_expand: {type: many2one, target: household, field: hh_clone_expand_child_id}
            hh_clone_immigration: {type: many2one, target: household, field: hh_clone_immigration_child_id}



        # HOUSEHOLDS's LEVEL
        processes:
```

```
271
272
273
274
275             ##############################################
276             # "INIT:" BLOCK => PREPARATION OF SIMULATION  #
277             ##############################################
278
279
280
281             init_block_hh_generalities():
282
283                 - bidon: "0"
284                 - cloned_expand_upstream_hh_id: "-1"
285                 - cloned_immigration_upstream_hh_id: "-1"
286                 - dwt_final: dwt_int
287                 - dwt_int_in_raw: dwt_int
288                 - has_a_clone_expand_upstream: "False"
289                 - has_a_clone_immigration_upstream: "False"
290                 - hh_age_max: persons.max( dag )
291                 - nb_adults: persons.count( ADULT )
292                 # "DCZ" : 1/ LU  -  2/ Other EU  -  3) Other
293                 # Citizenship :  1/  "2" if All ADULTS are "LU-citizens"
294                 #                2/  "3" if 1 ADULT at least if NOT "LU" and 1 at least is "LU" (=> "MIX HH)
295                 #                3/  "4" if all ADULTS not-"LU"  =>  "FOREIGN"
296                 #                4/  "1" otherwise
297                 - hh_citizenship: if( (nb_adults > 0)  and  ( persons.count( ADULT and (dcz == 1) ) == nb_adults ),
298                                       2,
299                                       if( (nb_adults > 0)
300                                            and  ( persons.count( ADULT and (dcz == 1) ) < nb_adults )
301                                            and  ( persons.count( ADULT and (dcz == 1) ) > 0 ),
302                                            3,
303                                            if( (nb_adults > 0)  and  ( persons.count( ADULT and (dcz == 1) ) == 0 ),
304                                                 4,
305                                                 1
306                                              )
307                                         )
308                                   )
309                 - hh_clone_expand_child_id: "-1"
310                 - hh_clone_immigration_child_id: "-1"
311                 - hh_head_id: persons.max( id, dag == ph.hh_age_max )
312                 - hh_ilsorigy_max: persons.max( ILS_ORIGY )
313                 - hh_wave_migration: if( ( persons.count(drsyy < (dag-1)) > 0 )
314                                          and ( persons.max(drsyy) <= MIGRATION_SHORT_WAVE_DURATION ),
315                                            3,
```

```
                                    if( persons.count(drsyy < (dag-1)) > 0,
                                        2,
                                        1
                                       )
                                )
        - hh_immigrant: hh_wave_migration > 0
        - hh_last_clone_immigration_period: "-1"
        - hh_nb_copies_remaining: dwt_int - 1
        - hh_now_to_be_cloned_for_immigration: "False"
        - hh_to_be_clone_expand: "False"
        - immigration_candidate: "False"
        - immigration_score: "0.0"
        - impact_immigration_age: "0.0"
        - nb_persons: persons.count()
        - year: period
        - year_of_expansion: "-1"
        - year_of_immigration:: "-1"



    #
    # OUTPUTTING RUSEFUL INFO for HOUSEHOLDS - ANTE EXPAND
    ######################################################


    init_block_household_output_ante_expand():


        # FIRST [A]  -  "EXPANSION"  -  HEADERS
        ########################################

        - csv( 'PERIOD',
               '',
               '',
               '',
               'NB_HHs',
               'NB_PERSONS_AVG',
               '',
               'HAS_CLONE_EXPAND_UPSTREAM_P',
               'NOT_HAS_CLONE_EXPAND_UPSTREAM_P',
               '',
               'HAS_CLONE_IMMIGRATION_UPSTREAM_P',
               'NOT_HAS_CLONE_IMMIGRATION_UPSTREAM_P',
               '',
               'IMMIGRANTS_P',
```

```
                        '',
                        'HH_CITIZENSHIP_NONE_P',
                        'HH_CITIZENSHIP_LU_P',
                        'HH_CITIZENSHIP_MIX_P',
                        'HH_CITIZENSHIP_FOREIGN_P',
                        '',
                        'HH_WAVE_MIGRATION_NONE_P',
                        'HH_WAVE_MIGRATION_>0_<=9_P',
                        'HH_WAVE_MIGRATION_OTHER_>0_>9_P',
                        '',
                        fname='OUTPUT_HOUSEHOLD_AGGREGATE.csv')


                # FIRST [B]  -  "EX ANTE EXPANSION"  -  VALUES
                ##############################################

                - csv( period,
                        '',
                        'ANTE',
                        '',
                        count(),
                        avg(nb_persons),
                        '',
                        count( has_a_clone_expand_upstream ) / count() * 100,
                        count( not(has_a_clone_expand_upstream) ) / count() * 100,
                        '',
                        count( has_a_clone_immigration_upstream ) / count() * 100,
                        count( not(has_a_clone_immigration_upstream) ) / count() * 100,
                        '',
                        count( hh_immigrant ) / count() * 100,
                        '',
                        count( hh_citizenship == 0 ) / count() * 100,
                        count( hh_citizenship == 1 ) / count() * 100,
                        count( hh_citizenship == 2 ) / count() * 100,
                        count( hh_citizenship == 3 ) / count() * 100,
                        '',
                        count( hh_wave_migration == 0 ) / count() * 100,
                        count( hh_wave_migration == 2 ) / count() * 100,
                        count( hh_wave_migration == 1 ) / count() * 100,
                        '',
                        fname='OUTPUT_HOUSEHOLD_AGGREGATE.csv',
                        mode='a' )
```

```
406             #
407             # CLONING HOUSEHOLDS
408             ####################
409
410
411             init_block_household_clone_expand():
412
413
414                 # SETTING BASIC VARIABLES FOR ALL, FIRST
415                 - dwt_final: dwt_int
416                 - has_a_clone_expand_upstream: False
417                 - year_of_expansion: -1
418                 - cloned_expand_upstream_hh_id: -1
419                 - hh_to_be_clone_expand: False
420                 - hh_clone_expand_child_id: -1
421
422                 # THEN CLONING
423                 - hh_nb_copies_remaining: dwt_int - 1
424
425                 - show('EXPAND HH - STEP_DWT')
426
427                 - i: 1
428                 - while ( i < (MAX_DWT_INT + 5) ) :
429                     - hh_to_be_clone_expand: hh_nb_copies_remaining > 0
430                     - hh_clone_expand_child_id:  if( hh_nb_copies_remaining > 0,
431                                                      clone(   filter = hh_to_be_clone_expand,
432                                                               hh_clone_expand_child_id = -1,
433                                                               has_a_clone_expand_upstream = True,
434                                                               cloned_expand_upstream_hh_id = id,
435                                                               hh_nb_copies_remaining = max( hh_nb_copies_remaining - 1,
436 0),
437                                                               year_of_expansion = period
438                                                          ),
439                                                    hh_clone_expand_child_id
440                                                 )
441                     - hh_nb_copies_remaining: if( hh_clone_expand_child_id > -1, 0, hh_nb_copies_remaining )
442                     - show('\n (EXPAND HH  -  I=', i, ' => COUNT=', count(),')')
443                     - i: i + 1
444
445                 - dwt_final: 1
446
447
448             #
449             # OUTPUTTING RUSEFUL INFO for HOUSEHOLDS - POST EXPAND
450             ######################################################
```

```
451
452
453             init_block_household_output_post_expand():
454
455
456
457
458                 # FIRST [C]  -  "EX POST EXPANSION"  -  VALUES
459                 ##############################################
460
461                 - csv( period,
462                        '',
463                        'POST',
464                        '',
465                        count(),
466                        avg(nb_persons),
467                        '',
468                        count( has_a_clone_expand_upstream ) / count() * 100,
469                        count( not(has_a_clone_expand_upstream) ) / count() * 100,
470                        '',
471                        count( has_a_clone_immigration_upstream ) / count() * 100,
472                        count( not(has_a_clone_immigration_upstream) ) / count() * 100,
473                        '',
474                        count( hh_immigrant ) / count() * 100,
475                        '',
476                        count( hh_citizenship == 0 ) / count() * 100,
477                        count( hh_citizenship == 1 ) / count() * 100,
478                        count( hh_citizenship == 2 ) / count() * 100,
479                        count( hh_citizenship == 3 ) / count() * 100,
480                        '',
481                        count( hh_wave_migration == 0 ) / count() * 100,
482                        count( hh_wave_migration == 2 ) / count() * 100,
483                        count( hh_wave_migration == 1 ) / count() * 100,
484                        '',
485                        fname='OUTPUT_HOUSEHOLD_AGGREGATE.csv',
486                        mode='a' )
487
488
489                 # SECOND [A]  -  "EXPANSION"  -  "HOUSEHOLDS" / EXHAUSTIVE  -  HEADERS
490                 ######################################################################
491
492                 - csv( 'ID', 'PERIOD', 'DWT_FINAL', 'DWT_INT_IN_RAW', 'DWT_INT', 'HH_CLONE_EXPAND_CHILD_ID',
493 'CLONED_EXPAND_UPSTREAM_HH_ID',
494                        suffix='OUTPUT_ALL_DEMOGRAPHICS')
495
```

```
                # SECOND [B]  -  "HOUSEHOLDS" / EXHAUSTIVE  -  VALUES
                ################################################

                - csv( dump( id, period, dwt_final, dwt_int_in_raw, dwt_int, hh_clone_expand_child_id,
cloned_expand_upstream_hh_id , header = False ),
                       suffix='OUTPUT_ALL_DEMOGRAPHICS' ,
                       mode='a' )




            ###############################################
            #  GENERALITIES (IN "HH => PROCESSES:" BLOCK) #
            ###############################################


            year_setting():
                - year: "period"



            #
            # COMPOSITION
            ##############

            household_characteristics():


                # BASIC
                #######

                - hh_age_max: persons.max( dag )
                - nb_adults: persons.count( ADULT )
                # "DCZ" : 1/ LU  -  2/ Other EU  -  3) Other
                # Citizenship :  1/  All ADULTS "LU"  =  "LU"
                #                2/  NOT ALL "LU" BUT ONE at least == "MIX"
                #                3/  All not-"LU"  =  "FOREIGN"
                - hh_citizenship: if( (nb_adults > 0)  and  ( persons.count( ADULT and (dcz == 1) ) == nb_adults ),
                                      1,
                                      if( (nb_adults > 0)
                                           and  ( persons.count( ADULT and (dcz == 1) ) < nb_adults )
```

```
                                        and  ( persons.count( ADULT and (dcz == 1) ) > 0 ),
                                        2,
                                        if( (nb_adults > 0)  and  ( persons.count( ADULT and (dcz == 1) ) == 0 ),
                                             3,
                                             0
                                          )
                                    )
                                )
    - hh_head_id: persons.max( id, dag == ph.hh_age_max )
    - hh_wave_migration: if( ( (persons.count(drsyy < dag)) > 0 ) and ( persons.max(dag - drsyy) <= 9),
                                        2,
                                        if( persons.count(drsyy < dag) > 0,
                                            1,
                                            0
                                        )
                                )
    - nb_persons: persons.count()


clean_empty():


    # INCOMES
    #########

    - remove( nb_persons == 0 )



#######################################
# IMMIGRATION  -  PARTIM "HOUSEHOLDS" #
#######################################


#
# IMMIGRATION THROUGH CLONING
#############################

immigration_hh_process:


    # 1) INITIALIZING
```

```
586
587                 - immigration_score: "0.0"
588                 - impact_immigration_age: "0.0"
589                 - hh_now_to_be_cloned_for_immigration : False
590                 - immigration_candidate: False
591
592                 # Migration flow wanted this year (by age & gender)
593                 - IMMIGRATION_THIS_PERIOD: AL_NET_MIGRATION[:,:, period - BASE_PERIOD_IN_AL_NET_MIGRATION]
594                 - immigration_target: trunc( IMMIGRATION_THIS_PERIOD / FACTOR_INFLATOR_POP )
595
596
597                 # 2) SCORING FOR CANDIDACY
598
599                 # Main criteria, after "number of persons in HH" => "last_clone_period"
600                 #   => If first year of simulation => reset to "-1" ("reset" because was set to "-1" in "init block" already)
601                 - hh_last_clone_immigration_period: if((period == BASE_PERIOD + 1), -1, hh_last_clone_immigration_period)
602
603                 # Favorizing "young" HHs : component is "2" if avg age of adults is in [20-29], "1" if [30-39], then "0"
604                 # (but if NO "adults" in HH, not favouring clonage => "0" for the present age component)
605                 - impact_immigration_age: if( persons.count(ADULT) > 0,
606                                               max (4 - trunc( (persons.sum(ADULT * dag)/persons.count(ADULT)) / 10 ) , 0),
607                                               0
608                                             )
609
610                 # Building "IMMIGRATION_SCORE" for candidacy
611                 # MAX 2023 - 2017 = "6" for weight of "LAST CLONE PERIOD" in the present exercise
612                 - immigration_score: if( (hh_last_clone_immigration_period == -1), period - BASE_PERIOD, period -
613 hh_last_clone_immigration_period )
614                 # As implemented here, 10 years younger for adults (on avg in HH) makes us "gaining" 1.0 in score,
615                 #    which is equivalent to 1 more year since last clonage (additive scoring up to there)
616                 # MAX reachable up to there : 1 + 1.0 * 0 = 1  /  MAX reachable up to there : 6 + 1.0 * 4 = 10
617                 - immigration_score: immigration_score + ( FACTOR_AGE * impact_immigration_age )
618                 # Citizenship :  1/  "2" if All ADULTS are "LU-citizens"
619                 #                2/  "3" if 1 ADULT at least if NOT "LU" and 1 at least is "LU" (=> "MIX HH)
620                 #                3/  "4" if all ADULTS not-"LU"  =>  "FOREIGN"
621                 #                4/  "1" otherwise
622                 # Otherwise, a "preference" (multiplicative impact) in selection is given to more "recent" waves (<= 15 years
623 in LU, e.g.)
624                 #   => hh_wave_migration = "3" (if there are migrants, but max less than 15 years in LU), "2" (there are
625 migrants) or "1"
626                 # On the whole for the score => MIN = 1 * 1.0 * (1 + 1) = 3  /  MAX = 10 * 1.0 * (4 + 3) = 17
627                 # (all this + [0-1] "at random")
628                 - immigration_score: immigration_score * FACTOR_MIGRANTS_IN_HH * ( hh_citizenship + hh_wave_migration )
629                                      + FACTOR_MIGRANTS_RANDOM * uniform()
630
```

```
631
632                 # 3) SELECTION OF CANDIDATES (HHs) FOR CLONING & UPDATING "last_clone_period"
633
634                 # Candidacy (taking for sure all most "FOREIGN-CITIZENs" with RECENT WAVE (< 15= years) AND NOT SELECTED
635 BEFORE)
636                 - immigration_candidate: align( immigration_score, PROPORTION_SELECTION,
637                                                 take = ( (hh_citizenship == 4)  and  (hh_wave_migration == 3 )
638                                                          and (hh_last_clone_immigration_period == -1))
639                                               )
640
641                 # Selection
642                 - hh_now_to_be_cloned_for_immigration: align_abs(nb_persons, immigration_target, filter =
643 immigration_candidate, link = persons, secondary_axis = dgn, errors='carry')
644
645                 # Updating "last-clone_period"
646                 - hh_last_clone_immigration_period: if(hh_now_to_be_cloned_for_immigration, period,
647 hh_last_clone_immigration_period)
648
649
650                 # 4) CLONING HHs
651
652                 - hh_clone_immigration_child_id: clone( hh_now_to_be_cloned_for_immigration,
653                                                         hh_clone_immigration_child_id = -1,
654                                                         has_a_clone_immigration_upstream = True,
655                                                         hh_immigrant = True,
656                                                         hh_now_to_be_cloned_for_immigration = False,
657                                                         cloned_immigration_upstream_hh_id = id,
658                                                         year_of_immigration = period
659                                                       )
660
661
662
663
664             ################################
665             # OUTPUTTING EXCEL (.CSV) FILES #
666             ################################
667
668
669
670             output_household_final_csv():
671
672
673
674
675
```

```
# FIRST [D]  -  "EX POST EXPANSION"  -  VALUES
##############################################

- csv( period,
       '',
       '----',
       '',
       count(),
       avg(nb_persons),
       '',
       count( has_a_clone_expand_upstream ) / count() * 100,
       count( not(has_a_clone_expand_upstream) ) / count() * 100,
       '',
       count( has_a_clone_immigration_upstream ) / count() * 100,
       count( not(has_a_clone_immigration_upstream) ) / count() * 100,
       '',
       count( hh_immigrant ) / count() * 100,
       '',
       count( hh_citizenship == 0 ) / count() * 100,
       count( hh_citizenship == 1 ) / count() * 100,
       count( hh_citizenship == 2 ) / count() * 100,
       count( hh_citizenship == 3 ) / count() * 100,
       '',
       count( hh_wave_migration == 0 ) / count() * 100,
       count( hh_wave_migration == 2 ) / count() * 100,
       count( hh_wave_migration == 1 ) / count() * 100,
       '',
       fname='OUTPUT_HOUSEHOLD_AGGREGATE.csv',
       mode='a' )


# THIRD [A]  -  "HOUSEHOLDS" / EXHAUSTIVE  -  HEADERS
#####################################################

- csv( 'ID', 'PERIOD',
       suffix='OUTPUT_ALL_DEMOGRAPHICS')


# SECOND [B]  -  "HOUSEHOLDS" / EXHAUSTIVE  -  VALUES
#####################################################

- csv( dump( id, period, header = False),
       suffix='OUTPUT_ALL_DEMOGRAPHICS', mode='a' )
```

```
person:


    fields:


        #
        # FIRST, FROM INPUT
        #

        # period and id are implicit

        # Derived from "VERSION XX  -  DD MMM 2021  -  EUROMOD-LIAM for MID-TERM DYNAMICS.xlsm"
        #   => SHEET "CREATE_DECLA  -  2016_OUT_FLAGS"

        # A FEW EXPLICITATIONS, FIRST
        #
        #
        # ******************************
        # * CLASS_AGE_EUROMOD_LIAM2_FINE *
        # ******************************
        #
        #   1 "Age<18"
        #   2 "18<=Age<25"
        #   3 "25<=Age<50"
        #   4 "50<=Age<65"
        #   5 "Age>=65"
        #
        #
        # *************************
        # * CLASS_AGE_EUROMOD_LIAM2 *
        # *************************
        #
        #   1 "Age<15"
        #   2 "15<=Age<20"
```

```
#   3 "20<=Age<65"
#   4 "Age>=65"
#
#
# ***********************************
# * DEC (EDUCATION - CURRENT STATUS) *
# ***********************************
#
#   0: Not in Education
#   1: Pre-primary
#   2: Primary
#   3: Lower Secondary
#   4: Upper Secondary
#   5: Post Secondary
#   6: Tertiary"
#
#
# ************************************************************
# * DEH_EVER (EDUCATION - HIGHEST STATUS - [PhL] EVER REACHED) *
# ************************************************************
#
#   0: Not completed Primary
#   1: Primary
#   2: Lower Secondary
#   3: Upper Secondary
#   4: Post Secondary
#   5: Tertiary
#
#
# ************************
# * DMS ("MARITAL STATUS" *
# ************************
#
#   1 "SINGLE"
#   2 "MARRIED"
#   3 "SEPERATED"
#   4 "DIVORCED"
#   5 "WIDOWED"
#
#
# *********************************************
# * LCS  -  CIVIL SERVANT, based in "OCCUPATION" *
# *********************************************
#
#   0. False
```

```
#   1. True
#
#
# *******
# * LES *
# *******
#   0 "Pre-School"
#   1 "Farmer"
#   2 "Employer/Self-Employed"
#   3 "Employee"
#   4 "Pensioner"
#   5 "Unemployed"
#   6 "Student"
#   7 "Inactive"
#   8 "Sick/Disabled"
#   9 "Other"
#   10 "Family worker"
#
#

- id: int
- h_id: int
- period: int
- s_id: int
- f_id: int
- m_id: int
- head_hh_id: int
- idperson: int
- idhh: int
- idpartner: int
- idfather: int
- idmother: int
- aca: int
- aco: int
- adult_oecd: bool
- afc: float
- amrrm: int
- amrtn: int
- amrtn00: int
- ate: int
- bacpm: float
- bca01: float
- bca02: float
- bched01: float
- bched01_s: float
```

```
            - bched02: float
            - bched03: float
            - bched04: float
            - bched04_s: float
            - bched05: float
            - bdisv: float
            - bed: float
            - bed_s: float
            - bfa: float
            - bfapl: float
            - bfauc: float
            - bfauc_s: float
            - bhl: float
            - bho: float
            - bho_s: float
            - bmaba: float
            - bmact_s: float
            - bmals: float
            - bmawk: float
            - boni: float
            - bpact_s: float
            - bplct_s: float
            - bsa: float
            - bsacm: float
            - bsacm_s: float
            - bsaho_s: float
            - bsaht: float
            - bsaht_s: float
            - bsaot: float
            - bun: float
            - bunmy: int
            - bunmy_s: int
            - bunss: float
            - bunss_s: float
            - bunssmy: int
            - bwkmcee_s_LU_2020_STD: float
            - bwkmcee_s_LU_2020_STD_BASE: float
            - bwkmcee_s_LU_2021_STD: float
            - bwkmcee_s_LU_2021_STD_BASE: float
            - bwkmcee_s_LU_2022_STD: float
            - bwkmcee_s_LU_2022_STD_BASE: float
            - bwkmcee_s_LU_2023_STD: float
            - bwkmcee_s_LU_2023_STD_BASE: float
            - bwkmceemy_s_LU_2020_STD: int
            - bwkmceemy_s_LU_2021_STD: int
```

```
        - bwkmceemy_s_LU_2022_STD: int
        - bwkmceemy_s_LU_2023_STD: int
        - byr: float
        - child_oecd: bool
        - class_age_euromod_liam2: int
        - class_age_euromod_liam2_fine: int
        - dag: int
        - dag00: int
        - dct: int
        - dcz: int
        - ddi: int
        - ddilv: int
        - ddp_s: bool
        - ddt: int
        - dec: int
        - deciles_categories_rh_equiv_inc: int
        - deciles_rh_equiv_inc: float
        - decrg: bool
        - deh: int
        - dew: int
        - dey: int
        - dgn: bool
        - dmb: int
        - dms: int
        - dmsyy03: bool
        - dmsyy04: bool
        - dncsy: int
        - drgmd: bool
        - drgn1: int
        - drgru: bool
        - drgur: bool
        - drsyy: int
        - dsu00: int
        - dsu01: int
        - dsu02: int
        - dta_s: int
        - dwt: float
        - dwt_int: int
        - e20ps_o: int
        - e20pslw_o: int
        - e20psmd_o: bool
        - e20pspo_o: bool
        - freq_weights: int
        - hh_amrrm: int
        - hh_ils_dispy: float
```

```
            - idorighh: int
            - idorigperson: int
            - il_bho_disposable: float
            - il_earncouple: float
            - il_heating: float
            - il_repl: float
            - il_sadisregard: float
            - il_saother: float
            - il_tax_inc_bef: float
            - il_taxdedsic: float
            - il_taxinc: float
            - il_taxpen: float
            - ils_b1_bcb: float
            - ils_b1_bdi: float
            - ils_b1_bed: float
            - ils_b1_bfa: float
            - ils_b1_bhl: float
            - ils_b1_bho: float
            - ils_b1_boa: float
            - ils_b1_bsa: float
            - ils_b1_bsu: float
            - ils_b1_bun: float
            - ils_b2_bfaed: float
            - ils_b2_bsaho: float
            - ils_b2_penhl: float
            - ils_base_tin: float
            - ils_ben: float
            - ils_benmt: float
            - ils_bennt: float
            - ils_bensim: float
            - ils_dispy: float
            - ils_earns: float
            - ils_origrepy: float
            - ils_origy: float
            - ils_pen: float
            - ils_sicct: float
            - ils_sicdy: float
            - ils_sicee: float
            - ils_sicer: float
            - ils_sicot: float
            - ils_sicse: float
            - ils_tax: float
            - ils_taxsim: float
            - ils_udb_bdi: float
            - ils_udb_bed: float
```

```
            - ils_udb_bfa: float
            - ils_udb_bhl: float
            - ils_udb_bho: float
            - ils_udb_boa: float
            - ils_udb_bsa: float
            - ils_udb_bsu: float
            - ils_udb_bun: float
            - ils_udb_kfbcc: float
            - ils_udb_tis: float
            - ils_udb_tpr: float
            - ils_udb_xmp: float
            - ils_udb_yds: float
            - ils_udb_yem: float
            - ils_udb_yiy: float
            - ils_udb_yot: float
            - ils_udb_ypp: float
            - ils_udb_ypr: float
            - ils_udb_ypt: float
            - ils_udb_yse: float
            - is_poor_40: bool
            - is_poor_50: bool
            - is_poor_60: bool
            - is_poor_70: bool
            - kfb: float
            - kfbcc: float
            - kfbmy: int
            - kivho: float
            - lcb_a: float
            - lcl: int
            - lcs: bool
            - lcs01: bool
            - lcs02: bool
            - les: int
            - lfs: int
            - lhw: int
            - lhw_a: int
            - lhw_f: int
            - lhwsr_s_LU_2020_STD: int
            - lhwsr_s_LU_2021_STD: int
            - lhwsr_s_LU_2022_STD: int
            - lhwsr_s_LU_2023_STD: int
            - lindi: int
            - liwftmy: int
            - liwmy: int
            - liwmy_a: int
```

```
            - liwmy_f: int
            - liwmy_s: int
            - liwptmy: int
            - liwwh: int
            - liwwh00: int
            - liwwh_f: int
            - lma20_LU_2020_STD: int
            - lma21_LU_2021_STD: int
            - lma22_LU_2022_STD: int
            - lma23_LU_2023_STD: int
            - lmcee_s_LU_2020_STD: int
            - lmcee_s_LU_2021_STD: int
            - lmcee_s_LU_2022_STD: int
            - lmcee_s_LU_2023_STD: int
            - lnu: int
            - lnu_LU_2020_STD: int
            - lnu_LU_2021_STD: int
            - lnu_LU_2022_STD: int
            - lnu_LU_2023_STD: int
            - loc: int
            - lowas: bool
            - lpemy: int
            - lse: int
            - ltr: int
            - lunmy: int
            - lunmy_f: int
            - lunmy_s: int
            - new_f_id: int
            - new_h_id: int
            - new_head_hh_id: int
            - new_id: int
            - new_m_id: int
            - new_s_id: int
            - pdi: float
            - pdi00: float
            - pdimy: int
            - poa: float
            - poacc: float
            - poacm: float
            - poamr: float
            - poamy: int
            - poaps: float
            - poapu: float
            - poapups: float
            - poawr: float
```

```
            - poaxp: float
            - psu: float
            - psumy: int
            - psups: float
            - psupu: float
            - psupups: float
            - rh_eq_ils_dispy_oecd: float
            - rh_ils_dispy: float
            - rh_nb_adults_oecd: int
            - rh_nb_children_oecd: int
            - rh_weight_oecd: float
            - sel_s: float
            - sin01_s: float
            - sin02_s: float
            - sin03_s: float
            - sin04_s: float
            - sin20_s: float
            - sin21_s: float
            - sin22_s: float
            - sin23_s: float
            - sin24_s: float
            - sin25_s: float
            - sin26_s: float
            - sin27_s: float
            - sin28_s: float
            - tad: float
            - tin_s: float
            - tinta_s: float
            - tintace_s: float
            - tintadt_s: float
            - tintami_s: float
            - tintamp02_s: float
            - tintaox_s: float
            - tintape_s: float
            - tintapesn_s: float
            - tintapf_s: float
            - tintapv_s: float
            - tintb_s: float
            - tintceent_s: float
            - tintclpnt_s: float
            - tintcmp_s: float
            - tintcot_s: float
            - tintcpent_s: float
            - tintcrt_s: float
            - tintcsent_s: float
```

```
            - tinty_s: float
            - tinui_s: float
            - tinxt_s: float
            - tis: float
            - tpr: float
            - tscct_s: float
            - tsccthl_s: float
            - tscctpi_s: float
            - tscctsi_s: float
            - tscee_s: float
            - tsceebeot_s: float
            - tsceehl_s: float
            - tsceeot_s: float
            - tsceepbpi_s: float
            - tsceepi_s: float
            - tsceesi_s: float
            - tscer: float
            - tscer_s: float
            - tscerac_s: float
            - tscerhl_s: float
            - tscerpi_s: float
            - tscersi_s: float
            - tscse_s: float
            - tscseac_s: float
            - tscsehl_s: float
            - tscseot_s: float
            - tscsepi_s: float
            - tscsesi_s: float
            - tu_bho_lu_headid: int
            - tu_bho_lu_isdepchild: bool
            - tu_bho_lu_isloneparent: bool
            - tu_bho_lu_ispartner: bool
            - tu_bsa_lu_headid: int
            - tu_bsa_lu_isdepchild: bool
            - tu_bsa_lu_isloneparent: bool
            - tu_bsa_lu_ispartner: bool
            - tu_cb_lu_headid: int
            - tu_cb_lu_isdepchild: bool
            - tu_cb_lu_isloneparent: bool
            - tu_cb_lu_ispartner: bool
            - tu_edugrant_lu_headid: int
            - tu_edugrant_lu_isdepchild: bool
            - tu_edugrant_lu_isloneparent: bool
            - tu_edugrant_lu_ispartner: bool
            - tu_household_lu_headid: int
```

```
            - tu_household_lu_isdepchild: bool
            - tu_household_lu_isloneparent: bool
            - tu_household_lu_ispartner: bool
            - tu_tin_lu_headid: int
            - tu_tin_lu_isdepchild: bool
            - tu_tin_lu_isloneparent: bool
            - tu_tin_lu_ispartner: bool
            - value_deciles_rh_equiv_inc: int
            - xhc: float
            - xhcmomi: float
            - xhcot: float
            - xhcrt: float
            - xmp: float
            - xpp: float
            - yds: float
            - ydses_o: float
            - yem: float
            - yem_LU_2020_STD: float
            - yem_LU_2020_STD_BASE: float
            - yem_LU_2021_STD: float
            - yem_LU_2021_STD_BASE: float
            - yem_LU_2022_STD: float
            - yem_LU_2022_STD_BASE: float
            - yem_LU_2023_STD: float
            - yem_LU_2023_STD_BASE: float
            - yem_a: float
            - yemmw_s_LU_2020_STD: float
            - yemmw_s_LU_2020_STD_BASE: float
            - yemmw_s_LU_2021_STD: float
            - yemmw_s_LU_2021_STD_BASE: float
            - yemmw_s_LU_2022_STD: float
            - yemmw_s_LU_2022_STD_BASE: float
            - yemmw_s_LU_2023_STD: float
            - yemmw_s_LU_2023_STD_BASE: float
            - yemmwmy_s_LU_2020_STD: int
            - yemmwmy_s_LU_2021_STD: int
            - yemmwmy_s_LU_2022_STD: int
            - yemmwmy_s_LU_2023_STD: int
            - yemmy: int
            - yemmy_LU_2020_STD: int
            - yemmy_LU_2021_STD: int
            - yemmy_LU_2022_STD: int
            - yemmy_LU_2023_STD: int
            - yempv: float
            - yempv_a: float
```

```
1216            - yempv_s: float
1217            - yivwg: float
1218            - yiy: float
1219            - ymwdt: float
1220            - yot: float
1221            - ypp: float
1222            - ypr: float
1223            - ypt: float
1224            - yptmp: float
1225            - yse: float
1226            - yse_LU_2020_STD: float
1227            - yse_LU_2020_STD_BASE: float
1228            - yse_LU_2021_STD: float
1229            - yse_LU_2021_STD_BASE: float
1230            - yse_LU_2022_STD: float
1231            - yse_LU_2022_STD_BASE: float
1232            - yse_LU_2023_STD: float
1233            - yse_LU_2023_STD_BASE: float
1234            - ysemy: int
1235            - ysemy_LU_2020_STD: int
1236            - ysemy_LU_2021_STD: int
1237            - ysemy_LU_2022_STD: int
1238            - ysemy_LU_2023_STD: int
1239            - ysv: float
1240
1241
1242            #
1243            # THEN, ADDED, GENERAL (not in INPUT)
1244            #
1245
1246            - active_in_input:                  {type: bool, initialdata: false}
1247            - alive:                            {type: bool, initialdata: false}
1248            - bidon:                            {type: int, initialdata: false}
1249            - bidon_1:                          {type: int, initialdata: false}
1250            - dag_input:                        {type: int, initialdata: false}
1251            - dag2:                             {type: int, initialdata: false}
1252            - dead:                             {type: bool, initialdata: false}
1253            - deh_ever:                         {type: int, initialdata: false}
1254            - deh_ever_10:                      {type: bool, initialdata: false}
1255            - deh_ever_11:                      {type: bool, initialdata: false}
1256            - deh_ever_12:                      {type: bool, initialdata: false}
1257            - deh_ever_13:                      {type: bool, initialdata: false}
1258            - deh_ever_14:                      {type: bool, initialdata: false}
1259            - deh_ever_15:                      {type: bool, initialdata: false}
1260            - deh_try:                          {type: bool, initialdata: false}
```

```
1261            - deh_try_1:                          {type: bool, initialdata: false}
1262            - deh_try_l1:                         {type: bool, initialdata: false}
1263            - deh_try_l2:                         {type: bool, initialdata: false}
1264            - deh_try_l3:                         {type: bool, initialdata: false}
1265            - deh_try_l5:                         {type: bool, initialdata: false}
1266            - disabled:                           {type: bool, initialdata: false}
1267            - dms_1:                              {type: int, initialdata: false}
1268            - dwt_final:                          {type: int, initialdata: false}
1269            - dwt_in_raw:                         {type: float, initialdata: false}
1270            - dwt_int_in_raw:                     {type: int, initialdata: false}
1271            - employee:                           {type: bool, initialdata: false}
1272            - family_worker:                      {type: int, initialdata: false}
1273            - farmer:                             {type: bool, initialdata: false}
1274            - feed_age_unemp_M:                   {type: int, initialdata: false}
1275            - feed_age_unemp_F:                   {type: int, initialdata: false}
1276            - has_a_clone_expand_upstream:        {type: bool, initialdata: false}
1277            - has_a_clone_immigration_upstream:   {type: bool, initialdata: false}
1278            - inactive:                           {type: bool, initialdata: false}
1279            - is_new_clone_expand_downstream:     {type: bool, initialdata: false}
1280            - is_new_clone_immigration_downstream: {type: bool, initialdata: false}
1281            - les_1:                              {type: int, initialdata: false}
1282            - in_pre_school:                      {type: bool, initialdata: false}
1283            - is_age_60:                          {type: bool, initialdata: false}
1284            - is_child_011:                       {type: bool, initialdata: false}
1285            - months_in_statuses:                 {type: int, initialdata: false}
1286            - nch_011:                            {type: int, initialdata: false}
1287            - new_pensioner:                      {type: bool, initialdata: false}
1288            - new_unemployed:                     {type: bool, initialdata: false}
1289            - new_worker:                         {type: bool, initialdata: false}
1290            - original_clone_expand_pp_id:        {type: int, initialdata: false}
1291            - original_clone_immigration_pp_id:   {type: int, initialdata: false}
1292            - other_status:                       {type: bool, initialdata: false}
1293            - pensioner:                          {type: bool, initialdata: false}
1294            - pensioner_statutory:                {type: bool, initialdata: false}
1295            - pp_clone_expand_child_id:           {type: int, initialdata: false}
1296            - pp_clone_immigration_child_id:      {type: int, initialdata: false}
1297            - pp_immigrant:                       {type: bool, initialdata: false}
1298            - pp_nb_copies_remaining:             {type: int, initialdata: false}
1299            - pp_to_be_cloned_expand:             {type: bool, initialdata: false}
1300            - present_birth_in_model_child_id:    {type: int, initialdata: false}
1301            - ps_dead_id:                         {type: int, initialdata: false}
1302            - ps_dead_les:                        {type: int, initialdata: false}
1303            - to_give_birth:                      {type: bool, initialdata: false}
1304            - self_employed:                      {type: bool, initialdata: false}
1305            - student:                            {type: bool, initialdata: false}
```

```
        - student_1:                        {type: bool, initialdata: false}
        - temp1_bool:                       {type: bool, initialdata: false}
        - temp1_int:                        {type: int, initialdata: false}
        - temp2_int:                        {type: int, initialdata: false}
        - temp1_float:                      {type: float, initialdata: false}
        - temp2_float:                      {type: float, initialdata: false}
        - unemployed:                       {type: bool, initialdata: false}
        - unemployed_1:                     {type: bool, initialdata: false}
        - unemployed_in_input:              {type: bool, initialdata: false}
        - unemployment_score:               {type: float, initialdata: false}
        - working_in_input:                 {type: bool, initialdata: false}
        - working:                          {type: bool, initialdata: false}
        - working_1:                        {type: bool, initialdata: false}
        - year:                             {type: int, initialdata: false}
        - year_of_expansion:                {type: int, initialdata: false}
        - year_of_immigration:              {type: int, initialdata: false}



        #
        # THEN, ADDED, MORE SEPCIFIC ABOUT "PENSIONS" (not in INPUT)
        #

        # Some of the variables below are not necessary in the present context but we keep them
        # for CONFORMITY to NAMING OF VARIABLES in MAIN/FULL "MiDAS_LU" MODEL
        - pen_dur_coef_max:                 {type: float, initialdata: false}
        - pen_dur_compl_FO:                 {type: int, initialdata: false}
        - pen_dur_compl_FO_1:               {type: int, initialdata: false}
        - pen_dur_compl_LU:                 {type: int, initialdata: false}
        - pen_dur_compl_LU_1:               {type: int, initialdata: false}
        - pen_dur_compl_school_FO:          {type: int, initialdata: false}
        - pen_dur_compl_school_FO_1:        {type: int, initialdata: false}
        - pen_dur_compl_school_LU:          {type: int, initialdata: false}
        - pen_dur_compl_school_LU_1:        {type: int, initialdata: false}
        - pen_dur_eff_FO:                   {type: int, initialdata: false}
        - pen_dur_eff_FO_1:                 {type: int, initialdata: false}
        - pen_dur_eff_LU:                   {type: int, initialdata: false}
        - pen_dur_eff_LU_1:                 {type: int, initialdata: false}
        - pen_dur_eff_inwork_FO:            {type: int, initialdata: false}
        - pen_dur_eff_inwork_FO_1:          {type: int, initialdata: false}
        - pen_dur_eff_inwork_LU:            {type: int, initialdata: false}
        - pen_dur_eff_inwork_LU_1:          {type: int, initialdata: false}
        - pen_dur_eff_repl_FO:              {type: int, initialdata: false}
```

```
            - pen_dur_eff_repl_FO_1:            {type: int, initialdata: false}
            - pen_dur_eff_repl_LU:              {type: int, initialdata: false}
            - pen_dur_eff_repl_LU_1:            {type: int, initialdata: false}
            - pen_dur_tot:                      {type: int, initialdata: false}
            - pen_dur_tot_1:                    {type: int, initialdata: false}
            - pen_dur_tot_FO:                   {type: int, initialdata: false}
            - pen_dur_tot_FO_1:                 {type: int, initialdata: false}
            - pen_dur_tot_LU:                   {type: int, initialdata: false}
            - pen_dur_tot_LU_1:                 {type: int, initialdata: false}
            - pen_surv_base_LU:                 {type: float, initialdata: false}
            - pen_surv_base_LU_1:               {type: float, initialdata: false}
            - pen_surv_tr_LU:                   {type: float, initialdata: false}
            - pen_surv_tr_LU_1:                 {type: float, initialdata: false}
            - pen_surv_tr_nolimit_LU_alone:     {type: float, initialdata: false}
            - surv_personal_reference_income:   {type: float, initialdata: false}
            - surv_professional_based_income:   {type: float, initialdata: false}
            - to_new_pen_surv:                  {type: bool, initialdata: false}
            - ps_pen_surv_tr_LU:                {type: float, initialdata: false}



        macros:


            # PREPARING HEADERS FOR CSV OUTPUT FOR ALL POPULATION
            HEADERS_A_OUTPUT_ALL_PERSONS_FOR_EUROMOD: "'id', 'h_id', 'period', 'idhh', 'idperson', 'idpartner', 'idfather',
'idmother', 'idorighh', 'idorigperson', 'aca', 'aco', 'afc', 'amrrm', 'amrtn', 'amrtn00', 'ate', 'bca01', 'bca02', 'bched01',
'bched04', 'bched05', 'bed', 'bfa', 'bfapl', 'bfauc', 'bhl', 'bho', 'bmaba', 'bmals', 'bmawk', 'boni', 'bsa', 'bsacm',
'bsaht', 'bsaot', 'bun', 'bunmy', 'bunss', 'bunssmy', 'byr', 'dag', 'dag00', 'dct', 'dcz', 'ddi', 'ddt', 'dec', 'decrg',
'deh', 'dew', 'dey', 'dgn', 'dmb', 'dms', 'dmsyy03', 'dmsyy04', 'dncsy', 'drgmd', 'drgn1', 'drgru', 'drgur', 'drsyy', 'dsu00',
'dsu01', 'dsu02', 'dwt', 'e20ps_o', 'e20pslw_o', 'e20psmd_o', 'e20pspo_o', 'kfb', 'kfbcc', 'kfbmy', 'kivho', 'lcl', 'lcs',
'lcs01', 'lcs02', 'les', 'lfs', 'lhw', 'lhw_f', 'lindi', 'liwftmy', 'liwmy', 'liwmy_f', 'liwptmy', 'liwwh', 'liwwh00',
'liwwh_f', 'loc', 'lowas', 'lpemy', 'lse', 'ltr', 'lunmy', 'lunmy_f', 'pdi', 'pdi00', 'pdimy', 'poa', 'poacc', 'poacm',
'poamy', 'poapups', 'poaxp', 'psu', 'psumy', 'psupups', 'tad', 'tis', 'tpr', 'tscer', 'xhc', 'xhcmomi', 'xhcot', 'xhcrt',
'xmp', 'xpp', 'yds', 'ydses_o', 'yem', 'yemmy', 'yempv', 'yivwg', 'yiy', 'ymwdt', 'yot', 'ypp', 'ypr', 'ypt', 'yptmp', 'yse',
'ysemy', 'ysv', 'active_in_input', 'alive', 'dag2', 'deh_ever', 'deh_try', 'disabled', 'dms_1', 'employee', 'family_worker',
'farmer', 'has_a_clone_expand_upstream', 'has_a_clone_immigration_upstream', 'inactive',
'is_new_clone_expand_downstream', 'is_new_clone_immigration_downstream', 'les_1', 'in_pre_school', 'is_age_60',
'is_child_011', 'months_in_statuses', 'nch_011', 'new_pensioner', 'new_unemployed', 'new_worker',
'original_clone_expand_pp_id', 'original_clone_immigration_pp_id', 'other_status', 'pensioner', 'pensioner_statutory',
'pp_clone_expand_child_id', 'pp_clone_immigration_child_id', 'pp_immigrant', 'pp_nb_copies_remaining',
'pp_to_be_cloned_expand', 'present_birth_in_model_child_id', 'to_give_birth', 'self_employed', 'student', 'student_1',
```

```
'unemployed', 'unemployed_1', 'unemployed_in_input', 'unemployment_score', 'working_in_input', 'working', 'working_1',
'year', 'year_of_expansion', 'year_of_immigration', 'pen_dur_coef_max', 'pen_dur_compl_FO', 'pen_dur_compl_FO_1',
'pen_dur_compl_LU', 'pen_dur_compl_LU_1', 'pen_dur_compl_school_FO', 'pen_dur_compl_school_FO_1',
'pen_dur_compl_school_LU', 'pen_dur_compl_school_LU_1', 'pen_dur_eff_FO', 'pen_dur_eff_FO_1', 'pen_dur_eff_LU',
'pen_dur_eff_LU_1', 'pen_dur_eff_inwork_FO', 'pen_dur_eff_inwork_FO_1', 'pen_dur_eff_inwork_LU',
'pen_dur_eff_inwork_LU_1', 'pen_dur_eff_repl_FO', 'pen_dur_eff_repl_FO_1', 'pen_dur_eff_repl_LU',
'pen_dur_eff_repl_LU_1', 'pen_dur_tot', 'pen_dur_tot_1', 'pen_dur_tot_FO', 'pen_dur_tot_FO_1', 'pen_dur_tot_LU',
'pen_dur_tot_LU_1', 'pen_surv_base_LU', 'pen_surv_base_LU_1', 'pen_surv_tr_LU', 'pen_surv_tr_LU_1',
'pen_surv_tr_nolimit_LU_alone', 'surv_personal_reference_income', 'surv_professional_based_income', 'to_new_pen_surv',
'ps_pen_surv_tr_LU'"


            ACTIVE: les == 1 or les == 2 or les == 3 or les == 5
            ACTIVE_1: les_1 == 1 or les_1 == 2 or les_1 == 3 or les_1 == 5
            ADULT: if( (id == ph.hh_head_id) or (ps.id == ph.hh_head_id),
                        True,
                        False )
            AGE_ACTIVE_SMOOTHED: ( dag >= 16 ) and
                                 ( dag < if( dgn,
                                             trunc(PEN_LEGAL_AGE_M) + max( min( 3 - ( period - 2018 ), 3 ), 0 ),
                                             trunc(PEN_LEGAL_AGE_F) + max( min( 3 - ( period - 2018 ), 3 ), 0 )
                                           )
                                       )
            AGE_HIGHEST_COMPULSORY_EDUCATION: "15"
            AGE_HIGHEST_LOWER_SECONDARY: "15"
            AGE_HIGHEST_PRIMARY: "11"
            AGE_HIGHEST_UPPER_SECONDARY: "19"
            AGE_HIGHEST_TERTIARY: "26"
            AGE_MAX_DEH_EVER: "34"
            AGE_LOW_LIMIT_PENSION_SCHOOL: "18"
            AGE_RETIREMENT_LEGAL: dag >= if(dgn, trunc(PEN_LEGAL_AGE_M), trunc(PEN_LEGAL_AGE_F))
            AGE_RETIREMENT_LEGAL_SMOOTHED: dag >= if( dgn,
                                                      trunc(PEN_LEGAL_AGE_M) + max( min( 3 - ( period - 2018 ), 3 ), 0 ),
                                                      trunc(PEN_LEGAL_AGE_F) + max( min( 3 - ( period - 2018 ), 3 ), 0 )
                                                    )
            AGE_RETIREMENT_ANTICIPATED_HIGH_SMOOTHED: dag >= if( dgn,
                                                                 trunc(PEN_ANTICIPATED_AGE_HIGH_M) + max( min( 3 - ( period
- 2018 ), 3 ), 0 ),
                                                                 trunc(PEN_ANTICIPATED_AGE_HIGH_F) + max( min( 3 - ( period
- 2018 ), 3 ), 0 )
                                                               )
            AGE_RETIREMENT_ANTICIPATED_LOW_SMOOTHED: dag >= if( dgn,
                                                                trunc(PEN_ANTICIPATED_AGE_LOW_M) + max( min( 3 - ( period
- 2018 ), 3 ), 0 ),
                                                                trunc(PEN_ANTICIPATED_AGE_LOW_F) + max( min( 3 - ( period
```

```
- 2018 ), 3 ), 0 )
                                                                    )
            AGE_UNEMPLOYMENT_SMOOTHED: ( dag >= 20 ) and ( dag < if( dgn, feed_age_unemp_M, feed_age_unemp_F ) )
            BASE_PERIOD: "2017"
            BASE_YEAR_FOR_AL_TABLES: "2016"
            BUNSS_HYPOTHETICAL_VALUE: "1000"
            CIVIL_SERVANT_STATUS: (lcs == 1)
            CIVIL_SERVANT_STATUS_1: (lag(lcs) == 1)
            CIVIL_SERVANT_WORKING: (les == 1 or les == 2 or les == 3) and (lcs == 1)
            CIVIL_SERVANT_WORKING_1: ( lag(les) == 1 or lag(les) == 2 or lag(les == 3) ) and (lag(lcs) == 1)
            CLASS_AGE_EUROMOD_LIAM2_FINE_ORDER: if( dag >= 65,
                                                    4,
                                                    if( dag >= 50,
                                                        3,
                                                        if( dag >= 25,
                                                            2,
                                                            if( dag >= 15,
                                                                1,
                                                                0
                                                              )
                                                          )
                                                      )
                                                )

            # Proportions of Citizenships in 2017, in % of PERSONS
            DCZ_LU_P: "54"
            DCZ_EU_P: "39"
            DCZ_NON_EU_P: "7"
            # FOR "DEC" (current education level) : Not in education, Pre-/Primary, Lower/Upper/Post Secondary, Tertiary
            DEC_NONE: "0"
            DEC_PRE_P: "1"
            DEC_P: "2"
            DEC_LS: "3"
            DEC_US: "4"
            DEC_PS: "5"
            DEC_T: "6"
            # PROPORTIONS OF STUDENTS IN SEVERAL LEVELS OF EDUCATION
            # (WEIGTHED PROPORTIONS based on 2017 for [30-34] - must sum up to 100)
            DEH_EVER_0_P: "2"
            DEH_EVER_1_P: "3"
            DEH_EVER_2_P: "10"
            DEH_EVER_3_P: "33"
            DEH_EVER_4_P: "1"
            DEH_EVER_5_P: "51"
            # FOR "DEH" (highest diploma) : Nothing, Primary, Lower/Upper/Post Secondary, Tertiary
```

```
DEH_NONE: "0"
DEH_LS: "2"
DEH_P: "1"
DEH_PS: "4"
DEH_T: "5"
DEH_US: "3"
# PROPORTIONS OF STUDENTS OF ONE LEVEL THAT TRY THE NEXT LEVEL YET UNSUCCESFULLY
DEH_TRY_1_P: "0"
DEH_TRY_2_P: "80"
DEH_TRY_3_P: "100"
DEH_TRY_5_P: "70"
DISABLED: les == 8
EMPLOYEE: employee == 1
EMPLOYEE_1: lag(employee) == 1
FACTOR_PENSION_BROADLY_IMPUTED_FEMALE: "0.89"
FACTOR_PENSION_BROADLY_IMPUTED_MALE: "1.19"
# To convert pensions for new immigrants (as coming from OUTSIDE),
FACTOR_PENSION_FOREIGN_LU_RATIO: "0.82"
# CORRECTIVE FACTOR for making YEM as if SIMULATED for NOT "NEW_WORKERS" & YEM>0
FACTOR_YEM_BROADLY_IMPUTED_FEMALE: "0.99"
FACTOR_YEM_BROADLY_IMPUTED_MALE: "0.99"
# Change in YEM is cloned for MIGRATION but CS "originally"
FACTOR_YEM_MIGRANTS_NOW_NOT_CS:  "0.78"
# Ratio between YEM(PV) as known ABROAD and in LU
FACTOR_YEMPV_MIG: "0.80"
# CORRECTIVE FACTOR for making YSE as if SIMULATED for NOT "NEW_S-E" & YSE>0
FACTOR_YSE_BROADLY_IMPUTED_FEMALE: "1.05"
FACTOR_YSE_BROADLY_IMPUTED_MALE: "1.11"
FEMALE: dgn == 0
# In conformity with EUROMOD INCOME LISTS (which indeed adds to this "YEMMC_S", not present here !)
ILS_ORIGY: yem + yse + yiy + yot + ypp + xmp + ypr + ypt
# We distinguish "IN_COUPLE" from "MARRIED" given that 537/10493 persons are with a PARTNER yet not MARRIED
IN_COUPLE: (dms == 2) or (s_id != -1)
LCS_IN_EMPLOYEES_P: "0.07"
LOWER_SECONDARY_DEH_EVER: deh_ever == 2
MALE: dgn == 1
MARRIED: dms == 2
# MAXIMUM "WEIGHT", used for "WHILE" LOOP below
MAX_DWT_INT: "600"
# Max population considered for CSV OUTPUTS, now or later (645.674 in 2023, as on 29 SEPT 2021)
MAX_NUMBER_OF_RECORDS: "1000000"
NEW_IMMIGRANT: year_of_immigration == period
NEW_PENSIONER: (les == 4) and (les_1 != 4)
NO_PRIMARY_DEH_EVER: deh_ever == 0
PENS_OLD_AGE_PRIVATE_AVG_RATIO: "2772/3347"
```

```
PENS_OLD_AGE_PUBLIC_AVG_RATIO: "5441/3347"
PEN_SURV_PRORATA_OLD_AVG_PERC: "62.0"
PRIMARY_DEH_EVER: deh_ever == 1
PENSIONER: les == 4
PENSIONER_STATUTORY: pensioner_statutory
POST_SECONDARY_DEH_EVER: deh_ever == 4
SELF_EMPLOYED: self_employed == 1
SELF_EMPLOYED_1: lag(self_employed) == 1
# Below, linked to "LES" & "DMS"
SET_DIVORCED: "4"
SET_FARMER: "1"
SET_DISABLED: "8"
SET_FAMILY_WORKER: "10"
SET_EMPLOYEE: "3"
SET_INACTIVE: "7"
SET_IN_PRE_SCHOOL: "0"
SET_OTHER_STATUS: "9"
SET_PENSIONER: "4"
SET_SELF_EMPLOYED: "2"
SET_SINGLE: "1"
SET_STUDENT: "6"
SET_UNEMPLOYED: "5"
SET_WIDOW: "5"
STUDENT: "les == 6"
STUDENT_1: "les_1 == 6"
TERTIARY_DEH_EVER: deh_ever == 5
UNEMPLOYED: unemployed
UNEMPLOYED_1: unemployed_1
UNEMPLOYED_IN_INPUT: unemployed_in_input ==  1
UNEMP_TO_STILL_FIX: (unemployment_score == -1)
                     and ( les==1 or les==2 or les==3 or les==5
                           or ( (les_1 == 6) and not(les == 6)  )
                         )
UPPER_SECONDARY_DEH_EVER: deh_ever == 3
YOUNG_GRADUATED: not(les == 6) and (les_1 == 6)
WIDOW: dms == 5
WORKING: working
WORKING_1: working_1
YEAR_INPUT: "2017"
YEAR_FOR_CHECK_INIT: "2015"
YEAR_FOR_CHECK_THROUGH: "2016"
```

```
    links:


        mc: {type: one2many, target: person, field: m_id}
        fc: {type: one2many, target: person, field: f_id}
        ph: {type: many2one, target: household, field: h_id}
        pf: {type: many2one, target: person, field: f_id}
        ps: {type: many2one, target: person, field: s_id}
        pm: {type: many2one, target: person, field: m_id}
        person_clone_expand: {type: many2one, target: person, field: pp_clone_expand_child_id}



    processes:



        ################################################
        # "INIT:" BLOCK => PREPARATION OF SIMULATION  #
        ################################################


        #
        # GENERALITIES about variables while initialisation
        ####################################################

        init_block_person_general_variables_ante_expand():


            # GENERAL, BENEFITS, SOC CONTRIBUTIONS, TAXES & INCOMES
            ######################################################

            - active_in_input: les == 1 or les == 2 or les == 3 or les == 5
            - alive: True
            - bidon: "0"
            - class_age_euromod_liam2_fine: if( dag >= 65,
```

```
                                                    65,
                                                    if( dag >= 50,
                                                        50,
                                                        if( dag >= 25,
                                                            25,
                                                            if( dag >= 15,
                                                                15,
                                                                0
                                                              )
                                                          )
                                                      )
                                                  )
                - dag_input: dag
                - dag: dag00
                - dag2: dag ** 2
                - dead: not( alive )

                # EDUCATION - HIGHEST STATUS - [PhL] EVER REACHED
                - deh_ever: "-1"
                - deh_ever_l0: "False"
                - deh_ever_l1: "False"
                - deh_ever_l2: "False"
                - deh_ever_l3: "False"
                - deh_ever_l4: "False"
                - deh_ever_l5: "False"
                - deh_try: "False"
                - deh_try_l1: "False"
                - deh_try_l2: "False"
                - deh_try_l3: "False"
                - deh_try_l5: "False"

                - drsyy: if( drsyy > dag, dag, drsyy )
                - feed_age_unemp_M: trunc(PEN_LEGAL_AGE_M) + max( min( 3 - ( period - 2018 ), 3 ), 0 )
                - feed_age_unemp_F: trunc(PEN_LEGAL_AGE_M) + max( min( 3 - ( period - 2018 ), 3 ), 0 )
                - has_a_clone_immigration_upstream: "False"
                - is_age_60: if( dag00 >= 60, True, False )
                - is_new_clone_immigration_downstream: "False"
                - liwwh: max( liwwh, 0 )
                - liwwh00: max( liwwh00, 0 )# I change the condition below given that we encounter bizarre input with ages
around 10 NOT in education
                - original_clone_immigration_pp_id: "-1"
                - pp_immigrant: "False"
                - student: if( (dec > 1) or (dag >= AGE_LOWEST_IN_EDUCATION and dag <= AGE_HIGHEST_COMPULSORY_EDUCATION),
                               True,
                               False
```

```
                    )
            - les: if( student, SET_STUDENT, les)
            - disabled: "les == SET_DISABLED"
            - dwt_in_raw: dwt
            - dwt_int_in_raw: max(trunc(round(dwt,0)),1)
            - employee: "les == SET_EMPLOYEE"
            - family_worker: "les == SET_FAMILY_WORKER"
            - farmer: "les == SET_FARMER"
            - inactive: "les == SET_INACTIVE"
            - in_pre_school: "les == SET_IN_PRE_SCHOOL"
            - is_child_011: "if( (dag >= 0) and (dag <= 11),
                                True,
                                False)"
            - months_in_statuses: "0"
            - nch_011: "if( MALE,
                          fc.count(is_child_011),
                          mc.count(is_child_011))"
            - present_birth_in_model_child_id: "-1"
            - new_unemployed: False
            - new_worker: False
            - other_status: "les == SET_OTHER_STATUS"
            - pp_clone_expand_child_id: "-1"
            - pp_clone_immigration_child_id: "-1"
            - pp_immigrant: "False"
            - ps_dead_id: "-1"
            - ps_dead_les: "-1"
            - self_employed: "les == SET_SELF_EMPLOYED"
            - to_give_birth: False
            - student: "les == SET_STUDENT"
            - temp1_bool: "False"
            - temp1_int: "0"
            - temp2_int: "0"
            - temp1_float: "0.0"
            - temp2_float: "0.0"
            - unemployed: les == 5
            - unemployed_in_input: unemployed
            - working: les == SET_FARMER or les == SET_SELF_EMPLOYED or les == SET_EMPLOYEE
            - working_in_input: working
            - year: period
            - year_of_expansion: "-1"
            - year_of_immigration: "-1"


        #
```

```
1711            # PENSIONS
1712            ##########
1713
1714
1715            init_block_person_pension_variables_ante_expand():
1716
1717
1718                # GENERAL PENSIONS
1719                ###################
1720
1721
1722                # DURATIONS
1723                ############
1724
1725
1726                - pen_dur_eff_inwork_FO: max( (liwwh - drsyy * 12) , 0 )
1727                - pen_dur_eff_inwork_FO_1: "0"
1728                - pen_dur_eff_inwork_LU: max( liwwh - pen_dur_eff_inwork_FO, 0 )
1729                - pen_dur_eff_inwork_LU_1: "0"
1730                - pen_dur_eff_repl_FO: "0"
1731                - pen_dur_eff_repl_FO_1: "0"
1732                - pen_dur_eff_repl_LU: bunmy
1733                - pen_dur_eff_repl_LU_1: "0"
1734
1735
1736                # BASIC BENEFITS & RELATED INCOMES
1737                ##################################
1738
1739                - pen_surv_base_LU: "0.0"
1740                - pen_surv_tr_LU: "0.0"
1741                - surv_personal_reference_income: "0.0"
1742                - surv_professional_based_income: "0.0"
1743                - ps_pen_surv_tr_LU: "0.0"
1744                - to_new_pen_surv: "False"
1745
1746
1747                # STATUSES & IDENTIFIERS
1748                ########################
1749
1750                - new_pensioner: "False"
1751                - pensioner: les == SET_PENSIONER
1752                - pensioner_statutory: if( PENSIONER and CIVIL_SERVANT_STATUS,
1753                                           True,
1754                                           False )
1755
```

```
#
# OUTPUTTING RUSEFUL INFO for HOUSEHOLDS - ANTE EXPAND
#######################################################


init_block_person_output_ante_expand():


    # FIRST [A]  -  "EXPANSION"  -  HEADERS
    ######################################


    - csv('PERIOD',
          '',
          '',
          '',
          'POPULATION',
          'POP_FEMALES',
          'POP_MALES',
          '',
          '*** DEMOG ***',
          '',
          'HAS_A_CLONE_EXPAND_UPSTREAM',
          '',
          'HAS_A_CLONE_IMMIGRATION_UPSTREAM',
          '',
          'TO_GIVE_BIRTH',
          '',
          'AGE_FEM_AVG',
          'AGE_MAL_AVG',
          '',
          'IS_AGE_0_FEM',
          'IS_AGE_0_MAL',
          '',
          'IS_AGE_>=60_FEM_P',
          'IS_AGE_>=60_MAL_P',
          '',
          'CLASS_AGE_FEM_0_P',
          'CLASS_AGE_FEM_15_P',
          'CLASS_AGE_FEM_20_P',
          'CLASS_AGE_FEM_65_P',
          '',
          'CLASS_AGE_MAL_0_P',
          'CLASS_AGE_MAL_15_P',
```

```
                    'CLASS_AGE_MAL_20_P',
                    'CLASS_AGE_MAL_65_P',
                    '',
                    'CLASS_AGE_FINE_FEM_0_P',
                    'CLASS_AGE_FINE_FEM_15_P',
                    'CLASS_AGE_FINE_FEM_25_P',
                    'CLASS_AGE_FINE_FEM_50_P',
                    'CLASS_AGE_FINE_FEM_65_P',
                    '',
                    'CLASS_AGE_FINE_MAL_0_P',
                    'CLASS_AGE_FINE_MAL_15_P',
                    'CLASS_AGE_FINE_MAL_25_P',
                    'CLASS_AGE_FINE_MAL_50_P',
                    'CLASS_AGE_FINE_MAL_65_P',
                    '',
                    'DEC_NONE_P',
                    'DEC_PRE_P',
                    'DEC_P_P',
                    'DEC_LS_P',
                    'DEC_US_P',
                    'DEC_PS_P',
                    'DEC_T_P',
                    '',
                    'DEH_NONE_P',
                    'DEH_P_P',
                    'DEH_LS_P',
                    'DEH_US_P',
                    'DEH_PS_P',
                    'DEH_T_P',
                    '',
                    'DEH_EVER_NONE_P',
                    'DEH_EVER_P_P',
                    'DEH_EVER_LS_P',
                    'DEH_EVER_US_P',
                    'DEH_EVER_PS_P',
                    'DEH_EVER_T_P',
                    '',
                    'DEH_TRY_0_in_1_P',
                    'DEH_TRY_1_in_2_P',
                    'DEH_TRY_2_in_3_P',
                    'DEH_TRY_3_4_in_5_P',
                    '',
                    'DEY_Y_G_FEM_AVG',
                    'DEY_Y_G_MAL_AVG',
                    '',
```

```
                'DMS_0_FEM_P',
                'DMS_SINGLE_FEM_P',
                'DMS_MARRIED_FEM_P',
                'DMS_SEPERATED_FEM_P',
                'DMS_DIVORCED_FEM_P',
                'DMS_WIDOW_FEM_P',
                '',
                'DMS_0_MAL_P',
                'DMS_SINGLE_MAL_P',
                'DMS_MARRIED_MAL_P',
                'DMS_SEPERATED_MAL_P',
                'DMS_DIVORCED_MAL_P',
                'DMS_WIDOW_MAL_P',
                '',
                '*** WORKING STATUS ***',
                '',
                'LCS_0_P',
                'LCS_1_P',
                '',
                'LCS_EMPL_0_F_P',
                'LCS_EMPL_1_F_P',
                '',
                'LCS_EMPL_0_M_P',
                'LCS_EMPL_1_M_P',
                '',
                'LES_PRE_SCHOOL_FEM_P',
                'LES_FARM_FEM_P',
                'LES_S-E_FEM_P',
                'LES_EMPL_FEM_P',
                'LES_PENS_FEM_P',
                'LES_UNEMP_FEM_P',
                'LES_STUD_FEM_P',
                'LES_INACT_FEM_P',
                'LES_DISAB_FEM_P',
                'LES_OTHER_FEM_P',
                'LES_FAM_WORK_FEM_P',
                '',
                'LES_PRE_SCHOOL_MAL_P',
                'LES_FARM_MAL_P',
                'LES_S-E_MAL_P',
                'LES_EMPL_MAL_P',
                'LES_PENS_MAL_P',
                'LES_UNEMP_MAL_P',
                'LES_STUD_MAL_P',
                'LES_INACT_MAL_P',
```

```
                    'LES_DISAB_MAL_P',
                    'LES_OTHER_MAL_P',
                    'LES_FAM_WORK_MAL_P',
                    '',
                    'STUDENT_FEM_P',
                    'STUDENT_MAL_P',
                    '',
                    'WORKING_FEM_P',
                    'WORKING_MAL_P',
                    '',
                    'EMPLOYEE_FEM_P',
                    'EMPLOYEE_MAL_P',
                    '',
                    'CS_WORKING_FEM_P',
                    'CS_WORKING_MAL_P',
                    '',
                    'S-E_FEM_P',
                    'S-E_MAL_P',
                    '',
                    'UNEMPLOYED_FEM_P',
                    'UNEMPLOYED_MAL_P',
                    '',
                    'PENSIONER_FEM_P',
                    'PENSIONER_MAL_P',
                    '',
                    'PENS_STAT_FEM_P',
                    'PENS_STAT_MAL_P',
                    '',
                    'DISABLED_FEM_P',
                    'DISABLED_MAL_P',
                    '',
                    'TO_NEW_PENS_SURV_FEM_P',
                    'TO_NEW_PENS_SURV_FEM_M',
                    '',
                    '*** INCOMES ***',
                    '',
                    'YEM>0_P',
                    '',
                    'YEM>0_FEM_P',
                    'YEM>0_MAL_P',
                    '',
                    'YEM_if>0_AVG',
                    '',
                    'YEM_if>0_FEM_AVG',
                    'YEM_if>0_MAL_AVG',
```

```
                '',
                'YEM_NEW_W_EMPL_FEM_AVG',
                'YEM_NEW_W_EMPL_MAL_AVG',
                '',
                'YEM_NOT_NEW_W_EMPL_FEM_AVG',
                'YEM_NOT_NEW_W_EMPL_MAL_AVG',
                '',
                'YSE>0_P',
                '',
                'YSE>0_FEM_P',
                'YSE>0_MAL_P',
                '',
                'YSE_if>0_AVG',
                '',
                'YSE_if>0_FEM_AVG',
                'YSE_if>0_MAL_AVG',
                '',
                'YSE_NEW_W_S-E_FEM_AVG',
                'YSE_NEW_W_S-E_MAL_AVG',
                '',
                'YSE_NOT_NEW_W_S-E_FEM_AVG',
                'YSE_NOT_NEW_W_S-E_MAL_AVG',
                '',
                'POAPUPS>0_P',
                '',
                'POAPUPS>0_FEM_P',
                'POAPUPS>0_MAL_P',
                '',
                'POAPUPS_if>0_AVG',
                '',
                'POAPUPS_if>0_FEM_AVG',
                'POAPUPS_if>0_MAL_AVG',
                '',
                'POA_if>0_FEM_AVG',
                'POA_if>0_MAL_AVG',
                '',
                'TO_NEW_SURV_FEM_P',
                'TO_NEW_SURV_MAL_P',
                '',
                'PSUPUPS>0_P',
                '',
                'PSUPUPS>0_FEM_P',
                'PSUPUPS>0_MAL_P',
                '',
                'PSUPUPS_if>0_AVG',
```

```
                    '',
                    'PSUPUPS_if>0_FEM_AVG',
                    'PSUPUPS_if>0_MAL_AVG',
                    '',
                    'PDI00>0_FEM_P',
                    'PDI00>0_MAL_P',
                    '',
                    'PDI00_if>0_FEM_AVG',
                    'PDI00_if>0_MAL_AVG',
                    '',
                    'PDI>0_FEM_P',
                    'PDI>0_MAL_P',
                    '',
                    'PDI_if>0_FEM_AVG',
                    'PDI_if>0_MAL_AVG',
                    '',
                    fname='OUTPUT_PERSONS_AGGREGATE.csv')



# FIRST [B]  -  "EX ANTE EXPANSION"  -  VALUES
###############################################

- csv( period,
       '',
       'ANTE',
       '',
       count(),
       count( FEMALE ),
       count( MALE ),
       '',
       '',
       '',
       count( has_a_clone_expand_upstream ),
       '',
       count( has_a_clone_immigration_upstream ),
       '',
       count( to_give_birth ),
       '',
       avg( dag00, filter = FEMALE ),
       avg( dag00, filter = MALE ),
       '',
       count( dag==0 and FEMALE ),
```

```
                    count( dag==0 and MALE ),
                    '',
                    avg( is_age_60, filter = FEMALE ) * 100,
                    avg( is_age_60, filter = MALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2 == 0) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2 == 15) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2 == 20) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2 == 65) and FEMALE ) / count( FEMALE ) * 100,
                     '',
                    count( (class_age_euromod_liam2 == 0) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 15) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 20) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 65) and MALE ) / count( MALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2_fine == 0) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 15) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 25) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 50) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 65) and FEMALE ) / count( FEMALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2_fine == 0) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 15) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 25) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 50) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 65) and MALE ) / count( MALE ) * 100,
                    '',
                    avg( dec == 0 ) * 100,
                    avg( dec == 1 ) * 100,
                    avg( dec == 2 ) * 100,
                    avg( dec == 3 ) * 100,
                    avg( dec == 4 ) * 100,
                    avg( dec == 5 ) * 100,
                    avg( dec == 6 ) * 100,
                    '',
                    avg( deh == 0 ) * 100,
                    avg( deh == 1 ) * 100,
                    avg( deh == 2 ) * 100,
                    avg( deh == 3 ) * 100,
                    avg( deh == 4 ) * 100,
                    avg( deh == 5 ) * 100,
                    '',
                    avg( deh_ever == 0 ) * 100,
                    avg( deh_ever == 1 ) * 100,
                    avg( deh_ever == 2 ) * 100,
```

```
                    avg( deh_ever == 3 ) * 100,
                    avg( deh_ever == 4 ) * 100,
                    avg( deh_ever == 5 ) * 100,
                    '',
                    avg( deh_try_l1 ) * 100,
                    avg( deh_try_l2 ) * 100,
                    avg( deh_try_l3 ) * 100,
                    avg( deh_try_l5 ) * 100,
                    '',
                    avg( dey, filter = YOUNG_GRADUATED and FEMALE),
                    avg( dey, filter = YOUNG_GRADUATED and MALE),
                    '',
                    avg( dms == 0 , filter = FEMALE ) * 100,
                    avg( dms == 1 , filter = FEMALE ) * 100,
                    avg( dms == 2 , filter = FEMALE ) * 100,
                    avg( dms == 3 , filter = FEMALE ) * 100,
                    avg( dms == 4 , filter = FEMALE ) * 100,
                    avg( dms == 5 , filter = FEMALE ) * 100,
                    '',
                    avg( dms == 0 , filter = MALE ) * 100,
                    avg( dms == 1 , filter = MALE ) * 100,
                    avg( dms == 2 , filter = MALE ) * 100,
                    avg( dms == 3 , filter = MALE ) * 100,
                    avg( dms == 4 , filter = MALE ) * 100,
                    avg( dms == 5 , filter = MALE ) * 100,
                    '',
                    '',
                    '',
                    avg( lcs == 0 ) * 100,
                    avg( lcs == 1 ) * 100,
                    '',
                    avg( lcs == 0 , filter = EMPLOYEE and FEMALE ) * 100,
                    avg( lcs == 1 , filter = EMPLOYEE and FEMALE ) * 100,
                    '',
                    avg( lcs == 0 , filter = EMPLOYEE and MALE ) * 100,
                    avg( lcs == 1 , filter = EMPLOYEE and MALE ) * 100,
                    '',
                    avg( les == 0 , filter = FEMALE ) * 100,
                    avg( les == 1 , filter = FEMALE  ) * 100,
                    avg( les == 2 , filter = FEMALE  ) * 100,
                    avg( les == 3 , filter = FEMALE  ) * 100,
                    avg( les == 4 , filter = FEMALE  ) * 100,
                    avg( les == 5 , filter = FEMALE  ) * 100,
                    avg( les == 6 , filter = FEMALE  ) * 100,
                    avg( les == 7 , filter = FEMALE  ) * 100,
```

```
                avg( les == 8 , filter = FEMALE  ) * 100,
                avg( les == 9 , filter = FEMALE  ) * 100,
                avg( les == 10 , filter = FEMALE  ) * 100,
                '',
                avg( les == 0 , filter = MALE ) * 100,
                avg( les == 1 , filter = MALE  ) * 100,
                avg( les == 2 , filter = MALE  ) * 100,
                avg( les == 3 , filter = MALE  ) * 100,
                avg( les == 4 , filter = MALE  ) * 100,
                avg( les == 5 , filter = MALE  ) * 100,
                avg( les == 6 , filter = MALE  ) * 100,
                avg( les == 7 , filter = MALE  ) * 100,
                avg( les == 8 , filter = MALE  ) * 100,
                avg( les == 9 , filter = MALE  ) * 100,
                avg( les == 10 , filter = MALE  ) * 100,
                '',
                avg( STUDENT , filter = FEMALE  ) * 100,
                avg( STUDENT , filter = MALE  ) * 100,
                '',
                avg( WORKING , filter = FEMALE  ) * 100,
                avg( WORKING , filter = MALE  ) * 100,
                '',
                avg( EMPLOYEE , filter = FEMALE  ) * 100,
                avg( EMPLOYEE , filter = MALE  ) * 100,
                '',
                avg( CIVIL_SERVANT_WORKING , filter = FEMALE  ) * 100,
                avg( CIVIL_SERVANT_WORKING , filter = MALE  ) * 100,
                '',
                avg( SELF_EMPLOYED , filter = FEMALE  ) * 100,
                avg( SELF_EMPLOYED , filter = MALE  ) * 100,
                '',
                avg( UNEMPLOYED , filter = FEMALE  ) * 100,
                avg( UNEMPLOYED , filter = MALE  ) * 100,
                '',
                avg( PENSIONER , filter = FEMALE  ) * 100,
                avg( PENSIONER , filter = MALE  ) * 100,
                '',
                avg( PENSIONER_STATUTORY , filter = FEMALE  ) * 100,
                avg( PENSIONER_STATUTORY , filter = MALE  ) * 100,
                '',
                avg( DISABLED , filter = FEMALE  ) * 100,
                avg( DISABLED , filter = MALE  ) * 100,
                '',
                avg( to_new_pen_surv, filter = FEMALE) * 100,
                avg( to_new_pen_surv, filter = MALE) * 100,
```

```
2161                    '',
2162                    '',
2163                    '',
2164                    avg( yem > 0 ) * 100,
2165                    '',
2166                    avg( yem > 0 , filter = FEMALE ) * 100,
2167                    avg( yem > 0 , filter = MALE ) * 100,
2168                    '',
2169                    avg( yem, filter = (yem > 0) ),
2170                    '',
2171                    avg( yem, filter = (yem > 0) and FEMALE ),
2172                    avg( yem, filter = (yem > 0) and MALE ),
2173                    '',
2174                    avg( yem, filter = (yem > 0) and new_worker and EMPLOYEE and FEMALE ),
2175                    avg( yem, filter = (yem > 0) and new_worker and EMPLOYEE and MALE ),
2176                    '',
2177                    avg( yem, filter = (yem > 0) and not(new_worker) and EMPLOYEE and FEMALE ),
2178                    avg( yem, filter = (yem > 0) and not(new_worker) and EMPLOYEE and MALE ),
2179                    '',
2180                    avg( yse > 0 ) * 100,
2181                    '',
2182                    avg( yse > 0 , filter = FEMALE ) * 100,
2183                    avg( yse > 0 , filter = MALE ) * 100,
2184                    '',
2185                    avg( yse, filter = (yse > 0) ),
2186                    '',
2187                    avg( yse, filter = (yse > 0) and FEMALE ),
2188                    avg( yse, filter = (yse > 0) and MALE ),
2189                    '',
2190                    avg( yse, filter = (yse > 0) and new_worker and SELF_EMPLOYED and FEMALE ),
2191                    avg( yse, filter = (yse > 0) and new_worker and SELF_EMPLOYED and MALE ),
2192                    '',
2193                    avg( yse, filter = (yse > 0) and not(new_worker) and SELF_EMPLOYED and FEMALE ),
2194                    avg( yse, filter = (yse > 0) and not(new_worker) and SELF_EMPLOYED and MALE ),
2195                    '',
2196                    avg( poapups > 0 ) * 100,
2197                    '',
2198                    avg( poapups > 0, filter = FEMALE ) * 100,
2199                    avg( poapups > 0, filter = MALE ) * 100,
2200                    '',
2201                    avg( poapups, filter = poapups > 0 ),
2202                    '',
2203                    avg( poapups, filter = FEMALE and poapups > 0 ),
2204                    avg( poapups, filter = MALE and poapups > 0 ),
2205                    '',
```

```
                    avg( poa, filter = FEMALE and poa > 0 ),
                    avg( poa, filter = MALE and poa > 0 ),
                    '',
                    avg( to_new_pen_surv, filter = FEMALE ) * 100,
                    avg( to_new_pen_surv, filter = MALE ) * 100,
                    '',
                    avg( psupups > 0 ) * 100,
                    '',
                    avg( psupups > 0, filter = FEMALE ) * 100,
                    avg( psupups > 0, filter = MALE ) * 100,
                    '',
                    avg( psupups, filter = psupups > 0 ),
                    '',
                    avg( psupups, filter = FEMALE and psupups > 0 ),
                    avg( psupups, filter = MALE and psupups > 0 ),
                    '',
                    avg( pdi00 > 0, filter = FEMALE ) * 100,
                    avg( pdi00 > 0, filter = MALE ) * 100,
                    '',
                    avg( pdi00, filter = FEMALE and pdi00 > 0 ),
                    avg( pdi00, filter = MALE and pdi00 > 0 ),
                    '',
                    avg( pdi > 0, filter = FEMALE ) * 100,
                    avg( pdi > 0, filter = MALE ) * 100,
                    '',
                    avg( pdi, filter = FEMALE and pdi > 0 ),
                    avg( pdi, filter = MALE and pdi > 0 ),
                    '',
                    fname='OUTPUT_PERSONS_AGGREGATE.csv',
                    mode='a' )


        #
        # CLONING HOUSEHOLDS
        ####################


        init_block_person_clone_expand():


            - dwt_final: dwt_int
            - has_a_clone_expand_upstream: False
            - year_of_expansion: "-1"
            - original_clone_expand_pp_id: "-1"
```

```
            - pp_to_be_cloned_expand: False
            - pp_clone_expand_child_id: "-1"

            # THEN CLONING
            - pp_nb_copies_remaining: dwt_int - 1
            - i: 1
            - while ( i < (MAX_DWT_INT + 5) ) :
                - is_new_clone_expand_downstream: False
                - pp_to_be_cloned_expand: pp_nb_copies_remaining > 0
                - pp_clone_expand_child_id:  if( pp_nb_copies_remaining > 0,
                                                 clone( filter = pp_to_be_cloned_expand,
                                                        has_a_clone_expand_upstream = True,
                                                        h_id = ph.hh_clone_expand.id,
                                                        is_new_clone_expand_downstream = True,
                                                        original_clone_expand_pp_id = id,
                                                        pp_clone_expand_child_id = -1,
                                                        pp_nb_copies_remaining = max( pp_nb_copies_remaining - 1, 0),
                                                        year_of_expansion = period,
                                                      ),
                                            pp_clone_expand_child_id
                                          )

                - s_id: if(is_new_clone_expand_downstream, ps.pp_clone_expand_child_id, s_id)

                - m_id: if(is_new_clone_expand_downstream, pm.pp_clone_expand_child_id, m_id)

                - f_id: if(is_new_clone_expand_downstream, pf.pp_clone_expand_child_id, f_id)

                - pp_nb_copies_remaining: if( pp_clone_expand_child_id > -1, 0, pp_nb_copies_remaining )
                - show('\n (EXPAND PP  -  I=', i, ' => COUNT=', count(),')')
                - i: i + 1
            - show('\n')

            - dwt_final: 1




        #
        # ARRANGING VARIABLES FOR EUROMOD
        ################################
```

```
2296            init_block_person_arrangements_post_expand():
2297
2298
2299
2300
2301                # FILLING "ID" VARIABLES WITH VALUES RELEVANT FOR EUROMOD
2302                ########################################################
2303
2304                - idperson: id
2305                - idhh: h_id
2306                - idpartner: s_id
2307                - idfather: f_id
2308                - idmother: m_id
2309
2310
2311                # EDUCATION (put here given that ALIGNMENTS => PROPORTIONS TO BE RESPECTED)
2312                ########################################################################
2313
2314                # EDUCATION - HIGHEST STATUS - [PhL] EVER REACHED
2315                #   0: Not completed Primary
2316                #   1: Primary
2317                #   2: Lower Secondary
2318                #   3: Upper Secondary
2319                #   4: Post Secondary
2320                #   5: Tertiary
2321
2322                - deh_ever: "-1"
2323                - deh_ever: if( (deh >= 0) and not(student) and (dag >= AGE_LOWEST_IN_EDUCATION),
2324                                deh,
2325                                deh_ever
2326                              )
2327
2328                - deh_ever_l0: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
2329                                   align( uniform(), DEH_EVER_0_P/100,
2330                                          take = deh_ever==0,
2331                                          leave = deh_ever >0 or deh > 0
2332                                        ),
2333                                   False
2334                                 )
2335                - deh_ever: if( deh_ever_l0, DEH_NONE, deh_ever)
2336
2337                - deh_ever_l1: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
2338                                   align( uniform(), DEH_EVER_1_P/100,
2339                                          take = deh_ever==1,
2340                                          leave = (deh_ever >=0 and deh_ever!=1) or deh > 1
```

```
                                ),
                            False
                        )
    - deh_ever: if( deh_ever_l1, DEH_P, deh_ever)

    - deh_ever_l2: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                            align( uniform(), DEH_EVER_2_P/100,
                                   take = deh_ever==2,
                                   leave = (deh_ever >=0 and deh_ever!=2) or deh > 2
                                ),
                            False
                        )
    - deh_ever: if( deh_ever_l2, DEH_LS, deh_ever)

    - deh_ever_l3: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                            align( uniform(), DEH_EVER_3_P/100,
                                   take = deh_ever==3,
                                   leave = (deh_ever >=0 and deh_ever!=3) or deh > 3
                                ),
                            False
                        )
    - deh_ever: if( deh_ever_l3, DEH_US, deh_ever)

    - deh_ever_l4: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                            align( uniform(), DEH_EVER_4_P/100,
                                   take = deh_ever==4,
                                   leave = (deh_ever >=0 and deh_ever!=4) or deh > 4
                                ),
                            False
                        )
    - deh_ever: if( deh_ever_l4, DEH_PS, deh_ever)

    - deh_ever_l5: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                            align( uniform(), DEH_EVER_5_P/100,
                                   take = deh_ever==5,
                                   leave = (deh_ever >=0 and deh_ever!=5)
                                ),
                            False
                        )
    - deh_ever: if( deh_ever_l5, DEH_T, deh_ever)

    # EDUCATION : INTRODUCING THE POSSIBILITY TO "TRY" REACHING A LEVEL BUT UNSUCESSFUL
    - deh_try_l1: if( deh_ever_l0, choice([True, False], [DEH_TRY_1_P/100, 1-DEH_TRY_1_P/100]), False)
    - deh_try_l2: if( deh_ever_l1, choice([True, False], [DEH_TRY_2_P/100, 1-DEH_TRY_2_P/100]), False)
    - deh_try_l3: if( deh_ever_l2, choice([True, False], [DEH_TRY_3_P/100, 1-DEH_TRY_3_P/100]), False)
```

```
                - deh_try_l5: if( deh_ever_l3 or deh_ever_l4, choice([True, False], [DEH_TRY_5_P/100, 1-DEH_TRY_5_P/100]),
False)
                - deh_try: False
                - deh_try: deh_try_l1 or deh_try_l2 or deh_try_l3 or deh_try_l5

                - show('\nAFTER "DEH_TRY_5" in ARRANGEMENTS_POST_EXPAND\n')


                # PENSION-RELATED VARIABLES, WHICH CAN BE INITIALIZED AFTER "EXPAND" ONLY
                ########################################################################
                #
                # -------------
                # DURATIONS ...
                # -------------
                #

                - pen_dur_compl_school_FO: "0"
                - pen_dur_compl_school_FO_1: "0"
                - pen_dur_compl_school_LU: "0"
                - temp1_int: ( dew - (year - dag) - AGE_LOW_LIMIT_PENSION_SCHOOL ) * 12
                - temp2_int: AGE_HIGHEST_UPPER_SECONDARY - AGE_LOW_LIMIT_PENSION_SCHOOL
                - pen_dur_compl_school_LU: if( dew <= 0,
                                            if( LOWER_SECONDARY_DEH_EVER and (dag >= AGE_LOW_LIMIT_PENSION_SCHOOL)
                                                and (deh_try) and (dag >= AGE_HIGHEST_UPPER_SECONDARY),
                                                max( temp2_int, 0 ) * 12,
                                                if( (UPPER_SECONDARY_DEH_EVER or POST_SECONDARY_DEH_EVER) and (dag >=
AGE_LOW_LIMIT_PENSION_SCHOOL),
                                                      if( deh_try and (dag >= AGE_HIGHEST_TERTIARY),
                                                          (AGE_HIGHEST_TERTIARY - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12,
                                                          if( not(deh_try) and (dag >= AGE_HIGHEST_UPPER_SECONDARY),
                                                              max( temp2_int, 0 ) * 12,
                                                              pen_dur_compl_school_LU
                                                            )
                                                        ),
                                                      if( TERTIARY_DEH_EVER and (dag >= AGE_LOW_LIMIT_PENSION_SCHOOL)
                                                          and (dag >= AGE_HIGHEST_TERTIARY),
                                                          (AGE_HIGHEST_TERTIARY - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12,
                                                          pen_dur_compl_school_LU
                                                        )
                                                    )
                                                ),
                                            if( ( LOWER_SECONDARY_DEH_EVER and (deh_try) )
                                                or UPPER_SECONDARY_DEH_EVER or POST_SECONDARY_DEH_EVER,
                                                min( max( temp1_int, 0), (AGE_HIGHEST_UPPER_SECONDARY -
AGE_LOW_LIMIT_PENSION_SCHOOL) * 12 ),
```

```
                                                    if( TERTIARY_DEH_EVER,
                                                        min( max( temp1_int, 0), (AGE_HIGHEST_TERTIARY -
AGE_LOW_LIMIT_PENSION_SCHOOL) * 12 ),
                                                        pen_dur_compl_school_LU
                                                    )
                                                )
                                            )
                - pen_dur_compl_school_LU_1: "0"

                - pen_dur_compl_school_LU: if( (pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU + pen_dur_compl_school_LU)
                                                > (dag - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12,
                                                max( pen_dur_compl_school_LU -
                                                     ( (pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU +
pen_dur_compl_school_LU)
                                                        - (dag - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12
                                                     ),
                                                     0
                                                ),
                                                pen_dur_compl_school_LU
                                            )

                - pen_dur_compl_FO: pen_dur_compl_school_FO
                - pen_dur_compl_FO_1: pen_dur_compl_school_FO_1
                - pen_dur_compl_LU: pen_dur_compl_school_LU
                - pen_dur_compl_LU_1: pen_dur_compl_school_LU_1
                - pen_dur_eff_FO: pen_dur_eff_inwork_FO + pen_dur_eff_repl_FO
                - pen_dur_eff_FO_1: pen_dur_eff_inwork_FO_1 + pen_dur_eff_repl_FO_1
                - pen_dur_eff_LU: pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU
                - pen_dur_eff_LU_1: pen_dur_eff_inwork_LU_1 + pen_dur_eff_repl_LU_1
                - pen_dur_tot_FO: pen_dur_eff_FO + pen_dur_compl_FO
                - pen_dur_tot_FO_1: pen_dur_eff_FO_1 + pen_dur_compl_FO_1
                - pen_dur_tot_LU: pen_dur_eff_LU + pen_dur_compl_LU
                - pen_dur_tot_LU_1: pen_dur_eff_LU_1 + pen_dur_compl_LU_1
                - pen_dur_tot: pen_dur_tot_LU + pen_dur_tot_FO
                - pen_dur_tot_1: pen_dur_tot_LU_1 + pen_dur_tot_FO_1

                - pen_dur_coef_max: "0.0"
                - pen_dur_coef_max:
                    min( 1.0 - (1/trunc(PEN_DUR_LEGAL_TOT/12)) * trunc( ( PEN_DUR_LEGAL_TOT - pen_dur_tot ) / 12 ), 1.0)

                #
                # ------------------------
                # ... THEN PENSION AMOUNTS
                # ------------------------
                #
```

```
2476
2477                - pen_surv_tr_nolimit_LU_alone: "0.0"
2478                - pen_surv_tr_nolimit_LU_alone:
2479                    if( PENSIONER or DISABLED,
2480                        ( poapups + pdi00 ) * PEN_SURV_PRORATA_OLD_AVG_PERC / 100,
2481                        if( CIVIL_SERVANT_STATUS,
2482                            PENS_OLD_AGE_PUBLIC_AVG_RATIO,
2483                            PENS_OLD_AGE_PRIVATE_AVG_RATIO
2484                          )
2485                          *
2486                          if( FEMALE,
2487                              if( not( is_age_60 ),
2488                                  TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 0, deh_ever],
2489                                  TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 1, deh_ever]
2490                                ),
2491                              if( not( is_age_60 ),
2492                                  TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 0, deh_ever],
2493                                  TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 1, deh_ever]
2494                                )
2495                            )
2496                            *
2497                            PEN_SURV_PRORATA_OLD_AVG_PERC / 100
2498                      )
2499                # MIN
2500                - pen_surv_tr_LU:
2501                    if( ( pen_surv_tr_nolimit_LU_alone > 0 ) and ( pen_dur_tot >= PEN_DUR_MIN_BENEFIT ),
2502                          max( pen_surv_tr_nolimit_LU_alone,
2503                               0.9 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12 )
2504                               * pen_dur_coef_max
2505                             ),
2506                          pen_surv_tr_nolimit_LU_alone
2507                      )
2508                # MAX
2509                - pen_surv_tr_LU:
2510                    if( pen_surv_tr_LU > 0,
2511                        if( CIVIL_SERVANT_STATUS_1,
2512                            pen_surv_tr_LU,
2513                            min( pen_surv_tr_LU,
2514                                 5/6 * 5 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12 )
2515                               )
2516                          ),
2517                        pen_surv_tr_LU
2518                      )
2519
2520
```

```
#
# OUTPUTTING USEFUL INFO for HOUSEHOLDS - POST EXPAND
#####################################################


init_block_person_output_post_expand():



    # FIRST [C]  -  "EX POST EXPANSION"  -  VALUES
    ##############################################

    - csv( period,
           '',
           'POST',
           '',
           count(),
           count( FEMALE ),
           count( MALE ),
           '',
           '',
           '',
           count( has_a_clone_expand_upstream ),
           '',
           count( has_a_clone_immigration_upstream ),
           '',
           count( to_give_birth ),
           '',
           avg( dag00, filter = FEMALE ),
           avg( dag00, filter = MALE ),
           '',
           count( dag==0 and FEMALE ),
           count( dag==0 and MALE ),
           '',
           avg( is_age_60, filter = FEMALE ) * 100,
           avg( is_age_60, filter = MALE ) * 100,
           '',
           count( (class_age_euromod_liam2 == 0) and FEMALE ) / count( FEMALE ) * 100,
           count( (class_age_euromod_liam2 == 15) and FEMALE ) / count( FEMALE ) * 100,
           count( (class_age_euromod_liam2 == 20) and FEMALE ) / count( FEMALE ) * 100,
           count( (class_age_euromod_liam2 == 65) and FEMALE ) / count( FEMALE ) * 100,
            '',
           count( (class_age_euromod_liam2 == 0) and MALE ) / count( MALE ) * 100,
```

```
                    count( (class_age_euromod_liam2 == 15) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 20) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 65) and MALE ) / count( MALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2_fine == 0) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 15) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 25) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 50) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 65) and FEMALE ) / count( FEMALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2_fine == 0) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 15) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 25) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 50) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 65) and MALE ) / count( MALE ) * 100,
                    '',
                    avg( dec == 0 ) * 100,
                    avg( dec == 1 ) * 100,
                    avg( dec == 2 ) * 100,
                    avg( dec == 3 ) * 100,
                    avg( dec == 4 ) * 100,
                    avg( dec == 5 ) * 100,
                    avg( dec == 6 ) * 100,
                    '',
                    avg( deh == 0 ) * 100,
                    avg( deh == 1 ) * 100,
                    avg( deh == 2 ) * 100,
                    avg( deh == 3 ) * 100,
                    avg( deh == 4 ) * 100,
                    avg( deh == 5 ) * 100,
                    '',
                    avg( deh_ever == 0 ) * 100,
                    avg( deh_ever == 1 ) * 100,
                    avg( deh_ever == 2 ) * 100,
                    avg( deh_ever == 3 ) * 100,
                    avg( deh_ever == 4 ) * 100,
                    avg( deh_ever == 5 ) * 100,
                    '',
                    avg( deh_try_l1 ) * 100,
                    avg( deh_try_l2 ) * 100,
                    avg( deh_try_l3 ) * 100,
                    avg( deh_try_l5 ) * 100,
                    '',
                    avg( dey, filter = YOUNG_GRADUATED and FEMALE),
                    avg( dey, filter = YOUNG_GRADUATED and MALE),
```

```
                    '',
                    avg( dms == 0 , filter = FEMALE ) * 100,
                    avg( dms == 1 , filter = FEMALE ) * 100,
                    avg( dms == 2 , filter = FEMALE ) * 100,
                    avg( dms == 3 , filter = FEMALE ) * 100,
                    avg( dms == 4 , filter = FEMALE ) * 100,
                    avg( dms == 5 , filter = FEMALE ) * 100,
                    '',
                    avg( dms == 0 , filter = MALE ) * 100,
                    avg( dms == 1 , filter = MALE ) * 100,
                    avg( dms == 2 , filter = MALE ) * 100,
                    avg( dms == 3 , filter = MALE ) * 100,
                    avg( dms == 4 , filter = MALE ) * 100,
                    avg( dms == 5 , filter = MALE ) * 100,
                    '',
                    '',
                    '',
                    avg( lcs == 0 ) * 100,
                    avg( lcs == 1 ) * 100,
                    '',
                    avg( lcs == 0 , filter = EMPLOYEE and FEMALE ) * 100,
                    avg( lcs == 1 , filter = EMPLOYEE and FEMALE ) * 100,
                    '',
                    avg( lcs == 0 , filter = EMPLOYEE and MALE ) * 100,
                    avg( lcs == 1 , filter = EMPLOYEE and MALE ) * 100,
                    '',
                    avg( les == 0 , filter = FEMALE ) * 100,
                    avg( les == 1 , filter = FEMALE  ) * 100,
                    avg( les == 2 , filter = FEMALE  ) * 100,
                    avg( les == 3 , filter = FEMALE  ) * 100,
                    avg( les == 4 , filter = FEMALE  ) * 100,
                    avg( les == 5 , filter = FEMALE  ) * 100,
                    avg( les == 6 , filter = FEMALE  ) * 100,
                    avg( les == 7 , filter = FEMALE  ) * 100,
                    avg( les == 8 , filter = FEMALE  ) * 100,
                    avg( les == 9 , filter = FEMALE  ) * 100,
                    avg( les == 10 , filter = FEMALE  ) * 100,
                    '',
                    avg( les == 0 , filter = MALE ) * 100,
                    avg( les == 1 , filter = MALE  ) * 100,
                    avg( les == 2 , filter = MALE  ) * 100,
                    avg( les == 3 , filter = MALE  ) * 100,
                    avg( les == 4 , filter = MALE  ) * 100,
                    avg( les == 5 , filter = MALE  ) * 100,
                    avg( les == 6 , filter = MALE  ) * 100,
```

```
                    avg( les == 7 , filter = MALE  ) * 100,
                    avg( les == 8 , filter = MALE  ) * 100,
                    avg( les == 9 , filter = MALE  ) * 100,
                    avg( les == 10 , filter = MALE  ) * 100,
                    '',
                    avg( STUDENT , filter = FEMALE  ) * 100,
                    avg( STUDENT , filter = MALE  ) * 100,
                    '',
                    avg( WORKING , filter = FEMALE  ) * 100,
                    avg( WORKING , filter = MALE  ) * 100,
                    '',
                    avg( EMPLOYEE , filter = FEMALE  ) * 100,
                    avg( EMPLOYEE , filter = MALE  ) * 100,
                    '',
                    avg( CIVIL_SERVANT_WORKING , filter = FEMALE  ) * 100,
                    avg( CIVIL_SERVANT_WORKING , filter = MALE  ) * 100,
                    '',
                    avg( SELF_EMPLOYED , filter = FEMALE  ) * 100,
                    avg( SELF_EMPLOYED , filter = MALE  ) * 100,
                    '',
                    avg( UNEMPLOYED , filter = FEMALE  ) * 100,
                    avg( UNEMPLOYED , filter = MALE  ) * 100,
                    '',
                    avg( PENSIONER , filter = FEMALE  ) * 100,
                    avg( PENSIONER , filter = MALE  ) * 100,
                    '',
                    avg( PENSIONER_STATUTORY , filter = FEMALE  ) * 100,
                    avg( PENSIONER_STATUTORY , filter = MALE  ) * 100,
                    '',
                    avg( DISABLED , filter = FEMALE  ) * 100,
                    avg( DISABLED , filter = MALE  ) * 100,
                    '',
                    avg( to_new_pen_surv, filter = FEMALE) * 100,
                    avg( to_new_pen_surv, filter = MALE) * 100,
                    '',
                    '',
                    '',
                    avg( yem > 0 ) * 100,
                    '',
                    avg( yem > 0 , filter = FEMALE ) * 100,
                    avg( yem > 0 , filter = MALE ) * 100,
                    '',
                    avg( yem, filter = (yem > 0) ),
                    '',
                    avg( yem, filter = (yem > 0) and FEMALE ),
```

```
                    avg( yem, filter = (yem > 0) and MALE ),
                    '',
                    avg( yem, filter = (yem > 0) and new_worker and EMPLOYEE and FEMALE ),
                    avg( yem, filter = (yem > 0) and new_worker and EMPLOYEE and MALE ),
                    '',
                    avg( yem, filter = (yem > 0) and not(new_worker) and EMPLOYEE and FEMALE ),
                    avg( yem, filter = (yem > 0) and not(new_worker) and EMPLOYEE and MALE ),
                    '',
                    avg( yse > 0 ) * 100,
                    '',
                    avg( yse > 0 , filter = FEMALE ) * 100,
                    avg( yse > 0 , filter = MALE ) * 100,
                    '',
                    avg( yse, filter = (yse > 0) ),
                    '',
                    avg( yse, filter = (yse > 0) and FEMALE ),
                    avg( yse, filter = (yse > 0) and MALE ),
                    '',
                    avg( yse, filter = (yse > 0) and new_worker and SELF_EMPLOYED and FEMALE ),
                    avg( yse, filter = (yse > 0) and new_worker and SELF_EMPLOYED and MALE ),
                    '',
                    avg( yse, filter = (yse > 0) and not(new_worker) and SELF_EMPLOYED and FEMALE ),
                    avg( yse, filter = (yse > 0) and not(new_worker) and SELF_EMPLOYED and MALE ),
                    '',
                    avg( poapups > 0 ) * 100,
                    '',
                    avg( poapups > 0, filter = FEMALE ) * 100,
                    avg( poapups > 0, filter = MALE ) * 100,
                    '',
                    avg( poapups, filter = poapups > 0 ),
                    '',
                    avg( poapups, filter = FEMALE and poapups > 0 ),
                    avg( poapups, filter = MALE and poapups > 0 ),
                    '',
                    avg( poa, filter = FEMALE and poa > 0 ),
                    avg( poa, filter = MALE and poa > 0 ),
                    '',
                    avg( to_new_pen_surv, filter = FEMALE ) * 100,
                    avg( to_new_pen_surv, filter = MALE ) * 100,
                    '',
                    avg( psupups > 0 ) * 100,
                    '',
                    avg( psupups > 0, filter = FEMALE ) * 100,
                    avg( psupups > 0, filter = MALE ) * 100,
                    '',
```

```
2746                    avg( psupups, filter = psupups > 0 ),
2747                    '',
2748                    avg( psupups, filter = FEMALE and psupups > 0 ),
2749                    avg( psupups, filter = MALE and psupups > 0 ),
2750                    '',
2751                    avg( pdi00 > 0, filter = FEMALE ) * 100,
2752                    avg( pdi00 > 0, filter = MALE ) * 100,
2753                    '',
2754                    avg( pdi00, filter = FEMALE and pdi00 > 0 ),
2755                    avg( pdi00, filter = MALE and pdi00 > 0 ),
2756                    '',
2757                    avg( pdi > 0, filter = FEMALE ) * 100,
2758                    avg( pdi > 0, filter = MALE ) * 100,
2759                    '',
2760                    avg( pdi, filter = FEMALE and pdi > 0 ),
2761                    avg( pdi, filter = MALE and pdi > 0 ),
2762                    '',
2763                    fname='OUTPUT_PERSONS_AGGREGATE.csv',
2764                    mode='a' )
2765
2766
2767                # SECOND [A]  -  "PERSONS" / EXHAUSTIVE  -  HEADERS
2768                ###################################################
2769
2770                - csv( HEADERS_A_OUTPUT_ALL_PERSONS_FOR_EUROMOD,
2771                      suffix='OUTPUT_ALL_PERSONS_FOR_EUROMOD_0_IDMAX_COMMA_SEP')
2772
2773                # SECOND [B]  -  "PERSONS" / EXHAUSTIVE  -  VALUES
2774                ###################################################
2775
2776                # MAY_BE_CHANGED_B
2777
2778                - csv( dump( id, h_id, period, idhh, idperson, idpartner, idfather, idmother, idorighh, idorigperson, aca,
2779                             aco, afc, amrrm, amrtn, amrtn00, ate, bca01, bca02, bched01, bched04, bched05, bed, bfa,
2780                             bfapl, bfauc, bhl, bho, bmaba, bmals, bmawk, boni, bsa, bsacm, bsaht, bsaot, bun, bunmy,
2781                             bunss, bunssmy, byr, dag, dag00, dct, dcz, ddi, ddt, dec, if(decrg,1,0), deh, dew, dey,
2782                             if(dgn,1,0), dmb, dms, if(dmsyy03,1,0), if(dmsyy04,1,0), dncsy, if(drgmd,1,0), drgn1,
2783                             if(drgru,1,0), if(drgur,1,0), drsyy, dsu00, dsu01, dsu02, dwt, e20ps_o, e20pslw_o,
2784                             if(e20psmd_o,1,0), if(e20pspo_o,1,0), kfb, kfbcc, kfbmy, kivho, lcl, if(lcs,1,0),
2785                             if(lcs01,1,0), if(lcs02,1,0), les, lfs, lhw, lhw_f, lindi, liwftmy, liwmy, liwmy_f,
2786                             liwptmy, liwwh, liwwh00, liwwh_f, loc, if(lowas,1,0), lpemy, lse, ltr, lunmy, lunmy_f,
2787                             pdi, pdi00, pdimy, poa, poacc, poacm, poamy, poapups, poaxp, psu, psumy, psupups, tad,
2788                             tis, tpr, tscer, xhc, xhcmomi, xhcot, xhcrt, xmp, xpp, yds, ydses_o, yem, yemmy, yempv,
2789                             yivwg, yiy, ymwdt, yot, ypp, ypr, ypt, yptmp, yse, ysemy, ysv, if(active_in_input,1,0),
2790                             if(alive,1,0), dag2, deh_ever, if(deh_try,1,0), if(disabled,1,0), dms_1, if(employee,1,0),
```

```
                                family_worker, if(farmer,1,0), if(has_a_clone_expand_upstream,1,0),
                                if(has_a_clone_immigration_upstream,1,0), if(inactive,1,0),
                                if(is_new_clone_expand_downstream,1,0), if(is_new_clone_immigration_downstream,1,0), les_1,
                                if(in_pre_school,1,0), if(is_age_60,1,0), if(is_child_011,1,0), months_in_statuses, nch_011,
                                if(new_pensioner,1,0), if(new_unemployed,1,0), if(new_worker,1,0),
                                original_clone_expand_pp_id, original_clone_immigration_pp_id, if(other_status,1,0),
                                if(pensioner,1,0), if(pensioner_statutory,1,0), pp_clone_expand_child_id,
                                pp_clone_immigration_child_id, if(pp_immigrant,1,0), pp_nb_copies_remaining,
                                if(pp_to_be_cloned_expand,1,0), present_birth_in_model_child_id, if(to_give_birth,1,0),
                                if(self_employed,1,0), if(student,1,0), if(student_1,1,0), if(unemployed,1,0),
                                if(unemployed_1,1,0), if(unemployed_in_input,1,0), unemployment_score,
                                if(working_in_input,1,0), if(working,1,0), if(working_1,1,0), year, year_of_expansion,
                                year_of_immigration, pen_dur_coef_max, pen_dur_compl_FO, pen_dur_compl_FO_1,
pen_dur_compl_LU,
                                pen_dur_compl_LU_1, pen_dur_compl_school_FO, pen_dur_compl_school_FO_1,
                                pen_dur_compl_school_LU, pen_dur_compl_school_LU_1, pen_dur_eff_FO, pen_dur_eff_FO_1,
                                pen_dur_eff_LU, pen_dur_eff_LU_1, pen_dur_eff_inwork_FO, pen_dur_eff_inwork_FO_1,
                                pen_dur_eff_inwork_LU, pen_dur_eff_inwork_LU_1, pen_dur_eff_repl_FO, pen_dur_eff_repl_FO_1,
                                pen_dur_eff_repl_LU, pen_dur_eff_repl_LU_1, pen_dur_tot, pen_dur_tot_1, pen_dur_tot_FO,
                                pen_dur_tot_FO_1, pen_dur_tot_LU, pen_dur_tot_LU_1, pen_surv_base_LU, pen_surv_base_LU_1,
                                pen_surv_tr_LU, pen_surv_tr_LU_1, pen_surv_tr_nolimit_LU_alone,
                                surv_personal_reference_income, surv_professional_based_income, if(to_new_pen_surv,1,0),
                                ps_pen_surv_tr_LU
                                , filter = ( id >= 0) and ( id <= (MAX_NUMBER_OF_RECORDS-1) )
                                , header = False
                            ),
                        suffix='OUTPUT_ALL_PERSONS_FOR_EUROMOD_0_IDMAX_COMMA_SEP', mode='a'
                    )



            ##################################################
            #  GENERALITIES (IN "PERSON => PROCESSES:" BLOCK) #
            ##################################################



            break():
                - breakpoint()
            break_init():
                - breakpoint(YEAR_FOR_CHECK_INIT)
            break_through():
                - breakpoint(YEAR_FOR_CHECK_THROUGH)

```

```
year_setting():
    - year: "period"



#
# LAGGED DIRECT VARIABLES PROCESS
#


lagged_vars_person_process():


    - bidon_1: if(dag == 0, 0, lag(bidon))
    - deh_try_1: if(dag == 0, False, lag(deh_try))
    - dms_1: if(dag == 0, 1, lag(dms))
    - les_1: if(dag == 0, 0, lag(les))
    - pen_dur_compl_FO_1: if(dag == 0, 0, lag(pen_dur_eff_FO))
    - pen_dur_compl_LU_1: if(dag == 0, 0, lag(pen_dur_eff_LU))
    - pen_dur_compl_school_FO_1: if(dag == 0, 0, lag(pen_dur_eff_FO))
    - pen_dur_compl_school_LU_1: if(dag == 0, 0, lag(pen_dur_eff_LU))
    - pen_dur_eff_FO_1: if(dag == 0, 0, lag(pen_dur_eff_FO))
    - pen_dur_eff_LU_1: if(dag == 0, 0, lag(pen_dur_eff_LU))
    - pen_dur_eff_inwork_FO_1: if(dag == 0, 0, lag(pen_dur_eff_inwork_FO))
    - pen_dur_eff_inwork_LU_1: if(dag == 0, 0, lag(pen_dur_eff_inwork_LU))
    - pen_dur_eff_repl_FO_1: if(dag == 0, 0, lag(pen_dur_eff_repl_FO))
    - pen_dur_eff_repl_LU_1: if(dag == 0, 0, lag(pen_dur_eff_repl_LU))
    - pen_dur_tot_1: if(dag == 0, 0, lag(pen_dur_tot))
    - pen_dur_tot_FO_1: if(dag == 0, 0, lag(pen_dur_tot_FO))
    - pen_dur_tot_LU_1: if(dag == 0, 0, lag(pen_dur_tot_LU))
    - pen_surv_base_LU_1: if(dag == 0, 0, lag(pen_surv_base_LU))
    - pen_surv_tr_LU_1: if(dag == 0, 0, lag(pen_surv_tr_LU))
    - student_1: if(dag == 0, False, lag(student))
    - unemployed_1: if(dag == 0, False, lag(unemployed))
    - working_1: if(dag == 0, False, lag(working))

#
# INSURING CONSISTENT STATUSES - FROM OVERALL "SOC_STATUS" TOWARDS BOOLEANS
#


from_overall_les_status_to_booleans():

    # FROM "LES" TO SIMPLE BOOLEANS
```

```
        - disabled: "les == SET_DISABLED"
        - employee: "les == SET_EMPLOYEE"
        - family_worker: "les == SET_FAMILY_WORKER"
        - farmer: "les == SET_FARMER"
        - inactive: "les == SET_INACTIVE"
        - in_pre_school: "les == SET_IN_PRE_SCHOOL"
        - other_status: "les == SET_OTHER_STATUS"
        - pensioner: "les == SET_PENSIONER"
        - self_employed: "les == SET_SELF_EMPLOYED"
        - student: "les == SET_STUDENT"
        - unemployed: "les == SET_UNEMPLOYED"
        - working: "les == SET_FARMER or les == SET_SELF_EMPLOYED or les == SET_EMPLOYEE"


    #
    # UPDATING "DURATIONS" AND MAKING THEM CONSISTENT
    #


    durations_consolidated():


        # "PENSION" DURATION VARIABLES
        ##############################


        - pen_dur_compl_FO: pen_dur_compl_school_FO
        - pen_dur_compl_FO_1: pen_dur_compl_school_FO_1
        - pen_dur_compl_LU: pen_dur_compl_school_LU
        - pen_dur_compl_LU_1: pen_dur_compl_school_LU_1
        - pen_dur_eff_FO: pen_dur_eff_inwork_FO + pen_dur_eff_repl_FO
        - pen_dur_eff_FO_1: pen_dur_eff_inwork_FO_1 + pen_dur_eff_repl_FO_1
        - pen_dur_eff_LU: pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU
        - pen_dur_eff_LU_1: pen_dur_eff_inwork_LU_1 + pen_dur_eff_repl_LU_1
        - pen_dur_tot_FO: pen_dur_eff_FO + pen_dur_compl_FO
        - pen_dur_tot_FO_1: pen_dur_eff_FO_1 + pen_dur_compl_FO_1
        - pen_dur_tot_LU: pen_dur_eff_LU + pen_dur_compl_LU
        - pen_dur_tot_LU_1: pen_dur_eff_LU_1 + pen_dur_compl_LU_1
        - pen_dur_tot: pen_dur_tot_LU + pen_dur_tot_FO
        - pen_dur_tot_1: pen_dur_tot_LU_1 + pen_dur_tot_FO_1
```

```
2926
2927                #
2928                # COMPUTING USEFUL INCOME VARIABLES, AT PERSONAL LEVEL
2929                #
2930
2931
2932                personal_revenue_variables_process():
2933
2934                    # USEFUL COMPOSED INCOME VARIABLES
2935
2936
2937                    # PENSIONS
2938                    ##########
2939
2940                    - pdi: if( pdi == 0.0,
2941                               if( dag < 65,
2942                                   ( pdi00 + bca01 + bca02 + bdisv + bacpm ),
2943                                   ( 0.0 )
2944                                 ),
2945                               pdi
2946                             )
2947                    - poa: if( new_pensioner,
2948                               if( dag < 65,
2949                                   ( poapups + poacm + poamr + poawr ) + poacc + poaxp,
2950                                   ( poaps + poapu + poacm + psups + psupu + pdi00 + poamr + bca01
2951                                     + bca02 + bdisv + poawr + bacpm + poacc + poaxp )
2952                                 ),
2953                               poa
2954                             )
2955
2956
2957                    # INCOMES
2958                    #########
2959
2960                    # Cf. EUROMOD INCOME LIST (useful for determining who is HAED of HOUSEHOLD in EUROMOD, hence to update
2961    "HEAD_HH_ID")
2962                    #   => This introduces a possibility of LOOP "EUROMOD <-> LIAM2", through "yemmc_s", a simulated compensation
2963    revenue
2964                    - ils_origy: ILS_ORIGY
2965
2966                    - surv_professional_based_income:
2967                        max( (yem + yse) + (bunss + byr) - 2/3 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12
2968    ),
2969                             0.0
2970                           )
```

```
    - surv_personal_reference_income: surv_professional_based_income + poapups + pdi00



################
#  DEMOGRAPHY  #
################



#
# AGES & CLASS_AGE PROCEDURE
#


ageing_and_incremental_procedure():

    - dag: "dag + 1"
    - dag00: "dag00 + 1"
    - dag2: dag ** 2
    - class_age_euromod_liam2: if( dag >= 65,
                                   65,
                                   if( dag >= 20,
                                       20,
                                       if( dag >= 15,
                                           15,
                                           0
                                         )
                                     )
                                 )

    - class_age_euromod_liam2_fine: if( dag >= 65,
                                        65,
                                        if( dag >= 50,
                                            50,
                                            if( dag >= 25,
                                                25,
                                                if( dag >= 15,
                                                    15,
                                                    0
                                                  )
                                              )
                                          )
                                      )
```

```
        - is_age_60: if( dag00 >= 60, True, False )

        - drsyy: drsyy + 1



    #
    # SURVIVAL & DEATH PROCEDURES
    #


    survival_procedure():

        - alive: "False"
        - alive: if( FEMALE,
                     align(uniform(), (AL_P_SURVIVAL_F)[:,period - BASE_YEAR_FOR_AL_TABLES]),
                     alive
                   )
        - alive: if( MALE,
                     align(uniform(), (AL_P_SURVIVAL_M)[:,period - BASE_YEAR_FOR_AL_TABLES]),
                     alive
                   )


    death_procedure():

        - dead: "not alive"

        # IMPACT ON PARTNER
        - dms: if( ps.dead,
                   if( MARRIED,
                       SET_WIDOW,
                       dms
                     ),
                   dms
                 )
        - ps_dead_id: "if(ps.dead, ps.id, 0)"
        - ps_dead_les: "if(ps.dead, lag(ps.les), 0)"
        - ps_pen_surv_tr_LU:
            if( ps.dead and WIDOW,
                ps.pen_surv_tr_LU_1,
                0.0
              )
        - to_new_pen_surv: "if(ps.dead and WIDOW, True, False)"
```

```
3061                # Updating LINKS
3062                - s_id: if( ps.dead, -1, s_id )
3063                - f_id: if(pf.dead, -1, f_id)
3064                - m_id: if(pm.dead, -1, m_id)
3065
3066                # Killing the dead person
3067                - remove(dead)
3068
3069            #
3070            # BIRTHS
3071            #
3072
3073
3074            birth_process():
3075
3076
3077                # WHO FOR GIVING BIRTH
3078                - to_give_birth: False
3079                - to_give_birth: if( FEMALE and (dag >= 15) and (dag <= 50),
3080                                     align(uniform(), AL_P_BIRTH),
3081                                     False
3082                                   )
3083
3084                # NEW BIRTHS
3085                - present_birth_in_model_child_id:
3086                        new( 'person', filter=to_give_birth,
3087                              amrrm = amrrm,
3088                              amrtn = amrtn,
3089                              amrtn00 = amrtn00,
3090                              ate = ate,
3091                              dag = 0,
3092                              dct = dct,
3093                              dcz = dcz,
3094                              dmb = dmb,
3095                              drgmd = drgmd,
3096                              drgn1 = drgn1,
3097                              drgru = drgru,
3098                              dsu00 = dsu00,
3099                              dsu01 = dsu01,
3100                              dsu02 = dsu02,
3101                              dwt = dwt,
3102                              dwt_int = dwt_int,
3103                              e20ps_o = e20ps_o,
3104                              e20pslw_o = e20pslw_o,
3105                              e20psmd_o = e20psmd_o,
```

```
                    e20pspo_o = e20pspo_o,
                    hh_amrrm = hh_amrrm,
                    hh_ils_dispy = hh_ils_dispy,
                    f_id = ps.id,
                    freq_weights = freq_weights,
                    head_hh_id = head_hh_id,
                    h_id = h_id,
                    idhh = h_id,
                    idfather = ps.id,
                    idmother = id,
                    il_bho_disposable = il_bho_disposable,
                    il_earncouple = il_earncouple,
                    il_heating = il_heating,
                    il_repl = il_repl,
                    il_sadisregard = il_sadisregard,
                    il_saother = il_saother,
                    il_tax_inc_bef = il_tax_inc_bef,
                    il_taxdedsic = il_taxdedsic,
                    il_taxinc = il_taxinc,
                    il_taxpen = il_taxpen,
                    ils_b1_bcb = ils_b1_bcb,
                    ils_b1_bdi = ils_b1_bdi,
                    ils_b1_bed = ils_b1_bed,
                    ils_b1_bfa = ils_b1_bfa,
                    ils_b1_bhl = ils_b1_bhl,
                    ils_b1_bho = ils_b1_bho,
                    ils_b1_boa = ils_b1_boa,
                    ils_b1_bsa = ils_b1_bsa,
                    ils_b1_bsu = ils_b1_bsu,
                    ils_b1_bun = ils_b1_bun,
                    ils_b2_bfaed = ils_b2_bfaed,
                    ils_b2_bsaho = ils_b2_bsaho,
                    ils_b2_penhl = ils_b2_penhl,
                    ils_base_tin = ils_base_tin,
                    ils_ben = ils_ben,
                    ils_benmt = ils_benmt,
                    ils_bennt = ils_bennt,
                    ils_bensim = ils_bensim,
                    ils_dispy = ils_dispy,
                    ils_earns = ils_earns,
                    ils_origrepy = ils_origrepy,
                    ils_origy = ils_origy,
                    ils_pen = ils_pen,
                    ils_sicct = ils_sicct,
                    ils_sicdy = ils_sicdy,
```

```
                                ils_sicee = ils_sicee,
                                ils_sicer = ils_sicer,
                                ils_sicot = ils_sicot,
                                ils_sicse = ils_sicse,
                                ils_tax = ils_tax,
                                ils_taxsim = ils_taxsim,
                                ils_udb_bdi = ils_udb_bdi,
                                ils_udb_bed = ils_udb_bed,
                                ils_udb_bfa = ils_udb_bfa,
                                ils_udb_bhl = ils_udb_bhl,
                                ils_udb_bho = ils_udb_bho,
                                ils_udb_boa = ils_udb_boa,
                                ils_udb_bsa = ils_udb_bsa,
                                ils_udb_bsu = ils_udb_bsu,
                                ils_udb_bun = ils_udb_bun,
                                ils_udb_kfbcc = ils_udb_kfbcc,
                                ils_udb_tis = ils_udb_tis,
                                ils_udb_tpr = ils_udb_tpr,
                                ils_udb_xmp = ils_udb_xmp,
                                ils_udb_yds = ils_udb_yds,
                                ils_udb_yem = ils_udb_yem,
                                ils_udb_yiy = ils_udb_yiy,
                                ils_udb_yot = ils_udb_yot,
                                ils_udb_ypp = ils_udb_ypp,
                                ils_udb_ypr = ils_udb_ypr,
                                ils_udb_ypt = ils_udb_ypt,
                                ils_udb_yse = ils_udb_yse,
                                is_poor_40 = is_poor_40,
                                is_poor_50 = is_poor_50,
                                is_poor_60 = is_poor_60,
                                is_poor_70 = is_poor_70,
                                m_id = id,
                                period = period,
                                rh_eq_ils_dispy_oecd = rh_eq_ils_dispy_oecd,
                                rh_ils_dispy = rh_ils_dispy,
                                rh_nb_adults_oecd = rh_nb_adults_oecd,
                                rh_nb_children_oecd = rh_nb_children_oecd,
                                rh_weight_oecd = rh_weight_oecd,
                                year = year
                            )


        after_birth_process():


```

```
3196                # INITIALIZING OTHER VARIABLES, if not done below in further processes
3197
3198                # FIRSTY, as LIST OF VARIABLES in INPUT DATASET
3199                # =============================================
3200
3201                - aca: if( dag == 0, 3, aca )
3202                - aco: if( dag == 0, 3, aco )
3203                - afc: if( dag == 0, 0.0, afc )
3204                - bacpm: if( dag == 0, 0.0, bacpm )
3205                - bca01: if( dag == 0, 0.0, bca01 )
3206                - bca02: if( dag == 0, 0.0, bca02 )
3207                - bched01: if( dag == 0, 0.0, bched01 )
3208                - bched01_s: if( dag == 0, 0.0, bched01_s )
3209                - bched02: if( dag == 0, 0.0, bched02 )
3210                - bched03: if( dag == 0, 0.0, bched03 )
3211                - bched04: if( dag == 0, 0.0, bched04 )
3212                - bched04_s: if( dag == 0, 0.0, bched04_s )
3213                - bched05: if( dag == 0, 0.0, bched05 )
3214                - bdisv: if( dag == 0, 0.0, bed )
3215                - bed: if( dag == 0, 0.0, bed )
3216                - bed_s: if( dag == 0, 0.0, bed_s )
3217                - bfa: if( dag == 0, 0.0, bfa )
3218                - bfapl: if( dag == 0, 0.0, bfapl )
3219                - bfauc: if( dag == 0, 0.0, bfauc )
3220                - bfauc_s: if( dag == 0, 0.0, bfauc_s )
3221                - bhl: if( dag == 0, 0.0, bhl )
3222                - bho: if( dag == 0, 0.0, bho )
3223                - bho_s: if( dag == 0, 0.0, bho_s )
3224                - bmaba: if( dag == 0, 0.0, bmaba )
3225                - bmact_s: if( dag == 0, 0.0, bmact_s )
3226                - bmals: if( dag == 0, 0.0, bmals )
3227                - bmawk: if( dag == 0, 0.0, bmawk )
3228                - boni: if( dag == 0, 0.0, boni )
3229                - bpact_s: if( dag == 0, 0.0, bpact_s )
3230                - bplct_s: if( dag == 0, 0.0, bplct_s )
3231                - bsa: if( dag == 0, 0.0, bsa )
3232                - bsacm: if( dag == 0, 0.0, bsacm )
3233                - bsacm_s: if( dag == 0, 0.0, bsacm_s )
3234                - bsaho_s: if( dag == 0, 0.0, bsaho_s )
3235                - bsaht: if( dag == 0, 0.0, bsaht )
3236                - bsaht_s: if( dag == 0, 0.0, bsaht_s )
3237                - bsaot: if( dag == 0, 0.0, bsaot )
3238                - bun: if( dag == 0, 0.0, bun )
3239                - bunmy: if( dag == 0, 0, bunmy )
3240                - bunmy_s: if( dag == 0, 0, bunmy_s )
```

```
                - bunss: if( dag == 0, 0.0, bunss )
                - bunss_s: if( dag == 0, 0.0, bunss_s )
                - bunssmy: if( dag == 0, 0, bunssmy )
                - bwkmcee_s_LU_2020_STD: if( dag == 0, 0.0, bwkmcee_s_LU_2020_STD )
                - bwkmcee_s_LU_2020_STD_BASE: if( dag == 0, 0.0, bwkmcee_s_LU_2020_STD_BASE )
                - bwkmcee_s_LU_2021_STD: if( dag == 0, 0.0, bwkmcee_s_LU_2021_STD )
                - bwkmcee_s_LU_2021_STD_BASE: if( dag == 0, 0.0, bwkmcee_s_LU_2021_STD_BASE )
                - bwkmcee_s_LU_2022_STD: if( dag == 0, 0.0, bwkmcee_s_LU_2022_STD )
                - bwkmcee_s_LU_2022_STD_BASE: if( dag == 0, 0.0, bwkmcee_s_LU_2022_STD_BASE )
                - bwkmcee_s_LU_2023_STD: if( dag == 0, 0.0, bwkmcee_s_LU_2023_STD )
                - bwkmcee_s_LU_2023_STD_BASE: if( dag == 0, 0.0, bwkmcee_s_LU_2023_STD_BASE )
                - bwkmceemy_s_LU_2020_STD: if( dag == 0, 0, bwkmceemy_s_LU_2020_STD )
                - bwkmceemy_s_LU_2021_STD: if( dag == 0, 0, bwkmceemy_s_LU_2021_STD )
                - bwkmceemy_s_LU_2022_STD: if( dag == 0, 0, bwkmceemy_s_LU_2022_STD )
                - bwkmceemy_s_LU_2023_STD: if( dag == 0, 0, bwkmceemy_s_LU_2023_STD )
                - byr: if( dag == 0, 0.0, byr )
                - class_age_euromod_liam2: if( dag == 0, 0, class_age_euromod_liam2 )
                - class_age_euromod_liam2_fine: if( dag == 0, 0, class_age_euromod_liam2_fine )
                - dag00: if( dag == 0, 0, dag00 )
                - ddi: if( dag == 0, 0, ddi )
                - ddt: if( dag == 0, -1, ddt )
                # Not standard treatment for "DGN" : NOT JUST "0" !
                - dgn : if( dag == 0, choice([True, False], [0.51, 0.49]), dgn )
                # Not standard treatment ("0")
                - dms: if( dag == 0, 1, dms )
                - dmsyy03: if( dag == 0, False, dmsyy03 )
                - dmsyy04: if( dag == 0, False, dmsyy04 )
                - dncsy: if( dag == 0, 0, dncsy )
                - drsyy: if( dag == 0, 0, drsyy )
                # Not standard treatment ("0") for "IDs" below
                - idpartner: if( dag == 0, -1, idpartner )
                - idorighh: if( dag == 0, -1, idorighh )
                - idorigperson: if( dag == 0, -1, idorigperson )
                # Not standard treatment ("0/-1/False") for "IDs" below
                - idperson: if( dag == 0, id, idperson )
                - kfb: if( dag == 0, 0.0, kfb )
                - kfbcc: if( dag == 0, 0.0, kfbcc )
                - kfbmy: if( dag == 0, 0, kfbmy )
                - kivho: if( dag == 0, 0, kivho )
                # Not standard treatment ("0")
                - lcl: if( dag == 0, -1, lcl )
                - lcs: if( dag == 0, False, lcs )
                - lcs01: if( dag == 0, False, lcs01 )
                - lcs02: if( dag == 0, False, lcs02 )
                - les: if( dag == 0, 0, les )
```

```
3286                - lfs: if( dag == 0, 0, lfs )
3287                - lhw_a: if( dag == 0, 0, lhw_a )
3288                - lhw_f: if( dag == 0, 0, lhw_f )
3289                - lhwsr_s_LU_2020_STD: if( dag == 0, 0, lhwsr_s_LU_2020_STD )
3290                - lhwsr_s_LU_2021_STD: if( dag == 0, 0, lhwsr_s_LU_2021_STD )
3291                - lhwsr_s_LU_2022_STD: if( dag == 0, 0, lhwsr_s_LU_2022_STD )
3292                - lhwsr_s_LU_2023_STD: if( dag == 0, 0, lhwsr_s_LU_2023_STD )
3293                - lindi: if( dag == 0, 0, lindi )
3294                - liwftmy: if( dag == 0, 0, liwftmy )
3295                - liwmy: if( dag == 0, 0, liwmy )
3296                - liwmy_a: if( dag == 0, 0, liwmy_a )
3297                - liwmy_f: if( dag == 0, 0, liwmy_f )
3298                - liwmy_s: if( dag == 0, 0, liwmy_s )
3299                - liwptmy: if( dag == 0, 0, liwptmy )
3300                - liwwh: if( dag == 0, 0, liwwh )
3301                - liwwh_f: if( dag == 0, 0, liwwh_f )
3302                - lma20_LU_2020_STD: if( dag == 0, 0, lma20_LU_2020_STD )
3303                - lma21_LU_2021_STD: if( dag == 0, 0, lma21_LU_2021_STD )
3304                - lma22_LU_2022_STD: if( dag == 0, 0, lma22_LU_2022_STD )
3305                - lma23_LU_2023_STD: if( dag == 0, 0, lma23_LU_2023_STD )
3306                - lmcee_s_LU_2020_STD: if( dag == 0, 0, lmcee_s_LU_2020_STD )
3307                - lmcee_s_LU_2021_STD: if( dag == 0, 0, lmcee_s_LU_2021_STD )
3308                - lmcee_s_LU_2022_STD: if( dag == 0, 0, lmcee_s_LU_2022_STD )
3309                - lmcee_s_LU_2023_STD: if( dag == 0, 0, lmcee_s_LU_2023_STD )
3310                - lnu: if( dag == 0, 0, lnu )
3311                - lnu_LU_2020_STD: if( dag == 0, 0, lnu_LU_2020_STD )
3312                - lnu_LU_2021_STD: if( dag == 0, 0, lnu_LU_2021_STD )
3313                - lnu_LU_2022_STD: if( dag == 0, 0, lnu_LU_2022_STD )
3314                - lnu_LU_2023_STD: if( dag == 0, 0, lnu_LU_2023_STD )
3315                # Not standard treatment ("0")
3316                - loc: if( dag == 0, -1, loc )
3317                - lowas: if( dag == 0, False, lowas )
3318                - lpemy: if( dag == 0, 0, lpemy )
3319                - lse: if( dag == 0, 0, lse )
3320                - ltr: if( dag == 0, -1, ltr )
3321                - lunmy: if( dag == 0, 0, lunmy )
3322                - lunmy_f: if( dag == 0, 0, lunmy_f )
3323                - lunmy_s: if( dag == 0, 0, lunmy_s )
3324                # Not standard treatment ("0") => probably "NEW_..." variables unused / irrelevant
3325                - new_f_id: if( dag == 0, -1, new_f_id )
3326                - new_h_id: if( dag == 0, -1, new_h_id )
3327                - new_head_hh_id: if( dag == 0, -1, new_head_hh_id )
3328                - new_id: if( dag == 0, -1, new_id )
3329                - new_m_id: if( dag == 0, -1, new_m_id )
3330                - new_s_id: if( dag == 0, -1, new_s_id )
```

```
                - pdi: if( dag == 0, 0.0, pdi )
                - pdi00: if( dag == 0, 0.0, pdi00 )
                - pdimy: if( dag == 0, 0, pdimy )
                - pen_dur_coef_max: if( dag == 0, 0.0, pen_dur_coef_max )
                - pen_surv_base_LU: if( dag == 0, 0.0, pen_surv_base_LU )
                - pen_surv_tr_LU: if( dag == 0, 0.0, pen_surv_tr_LU )
                - pen_surv_tr_nolimit_LU_alone: if( dag == 0, 0.0, pen_surv_tr_nolimit_LU_alone )
                - poa: if( dag == 0, 0.0, poa )
                - poacc: if( dag == 0, 0.0, poacc )
                - poacm: if( dag == 0, 0.0, poacm )
                - poamr: if( dag == 0, 0.0, poamr )
                - poamy: if( dag == 0, 0, poamy )
                - poaps: if( dag == 0, 0.0, poaps )
                - poapu: if( dag == 0, 0.0, poapu )
                - poapups: if( dag == 0, 0.0, poapups )
                - poawr: if( dag == 0, 0.0, poawr )
                - poaxp: if( dag == 0, 0.0, poaxp )
                - ps_dead_id: if( dag == 0, -1, ps_dead_id )
                - ps_dead_les: if( dag == 0, -1, ps_dead_les )
                - ps_pen_surv_tr_LU: if( dag == 0, 0.0, ps_pen_surv_tr_LU )
                - psu: if( dag == 0, 0.0, psu )
                - psumy: if( dag == 0, 0, psumy )
                - psups: if( dag == 0, 0.0, psups )
                - psupu: if( dag == 0, 0.0, psupu )
                - psupups: if( dag == 0, 0.0, psupups )
                - sel_s: if( dag == 0, 0.0, sel_s )
                # Not standard treatment ("0")
                - s_id: if( dag == 0, -1, s_id )
                - sin01_s: if( dag == 0, 0.0, sin01_s )
                - sin02_s: if( dag == 0, 0.0, sin02_s )
                - sin03_s: if( dag == 0, 0.0, sin03_s )
                - sin04_s: if( dag == 0, 0.0, sin04_s )
                - sin20_s: if( dag == 0, 0.0, sin20_s )
                - sin21_s: if( dag == 0, 0.0, sin21_s )
                - sin22_s: if( dag == 0, 0.0, sin22_s )
                - sin23_s: if( dag == 0, 0.0, sin23_s )
                - sin24_s: if( dag == 0, 0.0, sin24_s )
                - sin25_s: if( dag == 0, 0.0, sin25_s )
                - sin26_s: if( dag == 0, 0.0, sin26_s )
                - sin27_s: if( dag == 0, 0.0, sin27_s )
                - sin27_s: if( dag == 0, 0.0, sin27_s )
                - surv_personal_reference_income: if( dag == 0, 0.0, surv_personal_reference_income )
                - surv_professional_based_income: if( dag == 0, 0.0, surv_professional_based_income )
                - tad: if( dag == 0,  0.0, tad )
                - tin_s: if( dag == 0,  0.0, tin_s )
```

```
3376                - tinta_s: if( dag == 0,  0.0, tinta_s )
3377                - tintace_s: if( dag == 0,  0.0, tintace_s )
3378                - tintadt_s: if( dag == 0,  0.0, tintadt_s )
3379                - tintami_s: if( dag == 0,  0.0, tintami_s )
3380                - tintamp02_s: if( dag == 0,  0.0, tintamp02_s )
3381                - tintaox_s: if( dag == 0,  0.0, tintaox_s )
3382                - tintape_s: if( dag == 0,  0.0, tintape_s )
3383                - tintapesn_s: if( dag == 0,  0.0, tintapesn_s )
3384                - tintapf_s: if( dag == 0,  0.0, tintapf_s )
3385                - tintapv_s: if( dag == 0,  0.0, tintapv_s )
3386                - tintb_s: if( dag == 0,  0.0, tintb_s )
3387                - tintceent_s: if( dag == 0,  0.0, tintceent_s )
3388                - tintclpnt_s: if( dag == 0,  0.0, tintclpnt_s )
3389                - tintcmp_s: if( dag == 0,  0.0, tintcmp_s )
3390                - tintcot_s: if( dag == 0,  0.0, tintcot_s )
3391                - tintcpent_s: if( dag == 0,  0.0, tintcpent_s )
3392                - tintcrt_s: if( dag == 0,  0.0, tintcrt_s )
3393                - tintcsent_s: if( dag == 0,  0.0, tintcsent_s )
3394                - tinty_s: if( dag == 0,  0.0, tinty_s )
3395                - tinui_s: if( dag == 0,  0.0, tinui_s )
3396                - tinxt_s: if( dag == 0,  0.0, tinxt_s )
3397                - tis: if( dag == 0,  0.0, tis )
3398                - to_new_pen_surv: if( dag == 0, False, to_new_pen_surv )
3399                - tpr: if( dag == 0,  0.0, tpr )
3400                - tscct_s: if( dag == 0,  0.0, tscct_s )
3401                - tsccthl_s: if( dag == 0,  0.0, tsccthl_s )
3402                - tscctpi_s: if( dag == 0,  0.0, tscctpi_s )
3403                - tscctsi_s: if( dag == 0,  0.0, tscctsi_s )
3404                - tscee_s: if( dag == 0,  0.0, tscee_s )
3405                - tsceebeot_s: if( dag == 0,  0.0, tsceebeot_s )
3406                - tsceehl_s: if( dag == 0,  0.0, tsceehl_s )
3407                - tsceeot_s: if( dag == 0,  0.0, tsceeot_s )
3408                - tsceepbpi_s: if( dag == 0,  0.0, tsceepbpi_s )
3409                - tsceepi_s: if( dag == 0,  0.0, tsceepi_s )
3410                - tsceesi_s: if( dag == 0,  0.0, tsceesi_s )
3411                - tscer: if( dag == 0,  0.0, tscer )
3412                - tscer_s: if( dag == 0,  0.0, tscer_s )
3413                - tscerac_s: if( dag == 0,  0.0, tscerac_s )
3414                - tscerhl_s: if( dag == 0,  0.0, tscerhl_s )
3415                - tscerpi_s: if( dag == 0,  0.0, tscerpi_s )
3416                - tscersi_s: if( dag == 0,  0.0, tscersi_s )
3417                - tscse_s: if( dag == 0,  0.0, tscse_s )
3418                - tscseac_s: if( dag == 0,  0.0, tscseac_s )
3419                - tscsehl_s: if( dag == 0,  0.0, tscsehl_s )
3420                - tscseot_s: if( dag == 0,  0.0, tscseot_s )
```

```
                - tscsepi_s: if( dag == 0,  0.0, tscsepi_s )
                - tscsesi_s: if( dag == 0,  0.0, tscsesi_s )
                # Not standard treatment ("0")
                - tu_bho_lu_headid: if( dag == 0, -1, tu_bho_lu_headid )
                - tu_bho_lu_isdepchild: if( dag == 0, False, tu_bho_lu_isdepchild )
                - tu_bho_lu_isloneparent: if( dag == 0, False, tu_bho_lu_isloneparent )
                - tu_bho_lu_ispartner: if( dag == 0, False, tu_bho_lu_ispartner )
                # Not standard treatment ("0")
                - tu_bsa_lu_headid: if( dag == 0, -1, tu_bsa_lu_headid )
                - tu_bsa_lu_isdepchild: if( dag == 0, False, tu_bsa_lu_isdepchild )
                - tu_bsa_lu_isloneparent: if( dag == 0, False, tu_bsa_lu_isloneparent )
                - tu_bsa_lu_ispartner: if( dag == 0, False, tu_bsa_lu_ispartner )
                # Not standard treatment ("0")
                - tu_cb_lu_headid: if( dag == 0, -1, tu_cb_lu_headid )
                - tu_cb_lu_isdepchild: if( dag == 0, False, tu_cb_lu_isdepchild )
                - tu_cb_lu_isloneparent: if( dag == 0, False, tu_cb_lu_isloneparent )
                - tu_cb_lu_ispartner: if( dag == 0, False, tu_cb_lu_ispartner )
                # Not standard treatment ("0")
                - tu_edugrant_lu_headid: if( dag == 0, -1, tu_edugrant_lu_headid )
                - tu_edugrant_lu_isdepchild: if( dag == 0, False, tu_edugrant_lu_isdepchild )
                - tu_edugrant_lu_isloneparent: if( dag == 0, False, tu_edugrant_lu_isloneparent )
                - tu_edugrant_lu_ispartner: if( dag == 0, False, tu_edugrant_lu_ispartner )
                # Not standard treatment ("0")
                - tu_household_lu_headid: if( dag == 0, -1, tu_household_lu_headid )
                - tu_household_lu_isdepchild: if( dag == 0, False, tu_household_lu_isdepchild )
                - tu_household_lu_isloneparent: if( dag == 0, False, tu_household_lu_isloneparent )
                - tu_household_lu_ispartner: if( dag == 0, False, tu_household_lu_ispartner )
                # Not standard treatment ("0")
                - tu_tin_lu_headid: if( dag == 0, -1, tu_tin_lu_headid )
                - tu_tin_lu_isdepchild: if( dag == 0, False, tu_tin_lu_isdepchild )
                - tu_tin_lu_isloneparent: if( dag == 0, False, tu_tin_lu_isloneparent )
                - tu_tin_lu_ispartner: if( dag == 0, False, tu_tin_lu_ispartner )
                - xhc: if( dag == 0, 0.0, xhc )
                - xhcmomi: if( dag == 0, 0.0, xhcmomi )
                - xhcot: if( dag == 0, 0.0, xhcot )
                - xhcrt: if( dag == 0, 0.0, xhcrt )
                - xmp: if( dag == 0, 0.0, xmp )
                - xpp: if( dag == 0, 0.0, xpp )
                - yds: if( dag == 0, 0.0, yds )
                - ydses_o: if( dag == 0, 0.0, ydses_o )
                - yem: if( dag == 0, 0.0, yem )
                - yem_LU_2020_STD: if( dag == 0, 0.0, yem_LU_2020_STD )
                - yem_LU_2020_STD_BASE: if( dag == 0, 0.0, yem_LU_2020_STD_BASE )
                - yem_LU_2021_STD: if( dag == 0, 0.0, yem_LU_2021_STD )
                - yem_LU_2021_STD_BASE: if( dag == 0, 0.0, yem_LU_2021_STD_BASE )
```

```
3466                - yem_LU_2022_STD: if( dag == 0, 0.0, yem_LU_2022_STD )
3467                - yem_LU_2022_STD_BASE: if( dag == 0, 0.0, yem_LU_2022_STD_BASE )
3468                - yem_LU_2023_STD: if( dag == 0, 0.0, yem_LU_2023_STD )
3469                - yem_LU_2023_STD_BASE: if( dag == 0, 0.0, yem_LU_2023_STD_BASE )
3470                - yem_a: if( dag == 0, 0.0, yem_a )
3471                - yemmw_s_LU_2020_STD: if( dag == 0, 0.0, yemmw_s_LU_2020_STD )
3472                - yemmw_s_LU_2020_STD_BASE: if( dag == 0, 0.0, yemmw_s_LU_2020_STD_BASE )
3473                - yemmw_s_LU_2021_STD: if( dag == 0, 0.0, yemmw_s_LU_2021_STD )
3474                - yemmw_s_LU_2021_STD_BASE: if( dag == 0, 0.0, yemmw_s_LU_2021_STD_BASE )
3475                - yemmw_s_LU_2022_STD: if( dag == 0, 0.0, yemmw_s_LU_2022_STD )
3476                - yemmw_s_LU_2022_STD_BASE: if( dag == 0, 0.0, yemmw_s_LU_2022_STD_BASE )
3477                - yemmw_s_LU_2023_STD: if( dag == 0, 0.0, yemmw_s_LU_2023_STD )
3478                - yemmw_s_LU_2023_STD_BASE: if( dag == 0, 0.0, yemmw_s_LU_2023_STD_BASE )
3479                - yemmwmy_s_LU_2020_STD: if( dag == 0, 0,yemmwmy_s_LU_2020_STD )
3480                - yemmwmy_s_LU_2021_STD: if( dag == 0, 0,yemmwmy_s_LU_2021_STD )
3481                - yemmwmy_s_LU_2022_STD: if( dag == 0, 0,yemmwmy_s_LU_2022_STD )
3482                - yemmwmy_s_LU_2023_STD: if( dag == 0, 0,yemmwmy_s_LU_2023_STD )
3483                - yemmy: if( dag == 0, 0,yemmy )
3484                - yemmy_LU_2020_STD: if( dag == 0, 0,yemmy_LU_2020_STD )
3485                - yemmy_LU_2021_STD: if( dag == 0, 0,yemmy_LU_2021_STD )
3486                - yemmy_LU_2022_STD: if( dag == 0, 0,yemmy_LU_2022_STD )
3487                - yemmy_LU_2023_STD: if( dag == 0, 0,yemmy_LU_2023_STD )
3488                - yempv: if( dag == 0, 0.0, yempv )
3489                - yempv_a: if( dag == 0, 0.0, yempv_a )
3490                - yempv_s: if( dag == 0, 0.0, yempv_s )
3491                - yivwg: if( dag == 0, 0.0, yivwg )
3492                - yiy: if( dag == 0, 0.0, yiy )
3493                - ymwdt: if( dag == 0, 0.0, ymwdt )
3494                - yot: if( dag == 0, 0.0, yot )
3495                - ypp: if( dag == 0, 0.0, ypp )
3496                - ypr: if( dag == 0, 0.0, ypr )
3497                - ypt: if( dag == 0, 0.0, ypt )
3498                - yptmp: if( dag == 0, 0.0, yptmp )
3499                - yse: if( dag == 0, 0.0, yse )
3500                - yse_LU_2020_STD: if( dag == 0, 0.0, yse_LU_2020_STD )
3501                - yse_LU_2020_STD_BASE: if( dag == 0, 0.0, yse_LU_2020_STD_BASE )
3502                - yse_LU_2021_STD: if( dag == 0, 0.0, yse_LU_2021_STD )
3503                - yse_LU_2021_STD_BASE: if( dag == 0, 0.0, yse_LU_2021_STD_BASE )
3504                - yse_LU_2022_STD: if( dag == 0, 0.0, yse_LU_2022_STD )
3505                - yse_LU_2022_STD_BASE: if( dag == 0, 0.0, yse_LU_2022_STD_BASE )
3506                - yse_LU_2023_STD: if( dag == 0, 0.0, yse_LU_2023_STD )
3507                - yse_LU_2023_STD_BASE: if( dag == 0, 0.0, yse_LU_2023_STD_BASE )
3508                - ysemy: if( dag == 0, 0,ysemy )
3509                - ysemy_LU_2020_STD: if( dag == 0, 0,ysemy_LU_2020_STD )
3510                - ysemy_LU_2021_STD: if( dag == 0, 0, ysemy_LU_2021_STD )
```

```
3511                - ysemy_LU_2022_STD: if( dag == 0, 0,ysemy_LU_2022_STD )
3512                - ysemy_LU_2023_STD: if( dag == 0, 0,ysemy_LU_2023_STD )
3513                - ysv: if( dag == 0, 0.0, ysv )
3514
3515
3516                # SECONDLY, VARIABLES ADDED IN MODEL (cf. "..., initialdata: false")
3517                # ================================================================
3518
3519                - active_in_input: if( dag == 0, False, active_in_input )
3520                # Not standard treatment ("False")
3521                - alive: if( dag == 0, True, alive )
3522                - bidon: if( dag == 0, 0, bidon )
3523                # Not standard treatment ("0")
3524                - dag_input: if( dag == 0, -1, dag_input )
3525                - dag2: if( dag == 0, 0, dag2 )
3526                - dead: if( dag == 0, not( alive ), dead )
3527                # Not standard treatment for "DEH_EVER" : NOT JUST "0" !
3528                - deh_ever: if( dag == 0, -1, deh_ever )
3529                - deh_ever_l0: if( dag == 0, False, deh_ever_l0)
3530                - deh_ever_l1: if( dag == 1, False, deh_ever_l1)
3531                - deh_ever_l2: if( dag == 2, False, deh_ever_l2)
3532                - deh_ever_l3: if( dag == 3, False, deh_ever_l3)
3533                - deh_ever_l4: if( dag == 4, False, deh_ever_l4)
3534                - deh_ever_l5: if( dag == 5, False, deh_ever_l5)
3535                - deh_try: if( dag == 0, False, deh_try )
3536                - deh_try_l1: if( dag == 1, False, deh_try_l1)
3537                - deh_try_l2: if( dag == 2, False, deh_try_l2)
3538                - deh_try_l3: if( dag == 3, False, deh_try_l3)
3539                - deh_try_l5: if( dag == 5, False, deh_try_l5)
3540                - disabled: if( dag == 0, False, disabled )
3541                - dwt_final: if( dag == 0, 0, dwt_final )
3542                - dwt_in_raw: if( dag == 0, 0.0, dwt_in_raw )
3543                - dwt_int_in_raw: if( dag == 0, 0, dwt_int_in_raw )
3544                - employee: if( dag == 0, False, employee )
3545                - family_worker: if( dag == 0, 0, family_worker )
3546                - farmer: if( dag == 0, False, farmer )
3547                - has_a_clone_expand_upstream: if( dag == 0, False, has_a_clone_expand_upstream )
3548                - inactive: if( dag == 0, False, inactive )
3549                - is_age_60: if( dag == 0, False, is_age_60 )
3550                - is_new_clone_expand_downstream: if( dag == 0, False, is_new_clone_expand_downstream )
3551                # Not standard treatment ("False")
3552                - in_pre_school: if( dag == 0, True, in_pre_school )
3553                - months_in_statuses: if( dag == 0, 0, months_in_statuses )
3554                - new_pensioner: if( dag == 0, False, new_pensioner )
3555                - new_unemployed: if( dag == 0, False, new_unemployed )
```

```
3556                - new_worker: if( dag == 0, False, new_worker )
3557                - original_clone_expand_pp_id: if( dag == 0, -1, original_clone_expand_pp_id )
3558                - other_status: if( dag == 0, False, other_status )
3559                - pensioner: if( dag == 0, False, pensioner )
3560                - pensioner_statutory: if( dag == 0, False, pensioner_statutory )
3561                # Not standard treatment ("False") : "id" of a clone downstream
3562                - pp_clone_expand_child_id: if( dag == 0, -1, pp_clone_expand_child_id )
3563                - pp_nb_copies_remaining: if( dag == 0, 0, pp_nb_copies_remaining )
3564                - pp_to_be_cloned_expand: if( dag == 0, False, pp_to_be_cloned_expand )
3565                - to_give_birth: if( dag == 0, False, to_give_birth )
3566                - self_employed: if( dag == 0, False, self_employed )
3567                - student: if( dag == 0, False, student )
3568                - temp1_bool: if( dag == 0, False, temp1_bool )
3569                - temp1_int: if( dag == 0, 0, temp1_int )
3570                - temp2_int: if( dag == 0, 0, temp2_int )
3571                - temp1_float: if( dag == 0, 0.0, temp1_float )
3572                - temp2_float: if( dag == 0, 0.0, temp2_float )
3573                - unemployed: if( dag == 0, False, unemployed )
3574                - unemployed_in_input: if( dag == 0, False, unemployed_in_input )
3575                - unemployment_score: if( dag == 0, 0.0, unemployment_score )
3576                - working_in_input: if( dag == 0, False, working_in_input )
3577                - working: if( dag == 0, False, working )
3578                # Not standard treatment ("0")
3579                - year_of_expansion: if( dag == 0, -1, year_of_expansion )
3580
3581
3582
3583            #
3584            # CHILDHOOD
3585            #
3586
3587
3588            childhood():
3589
3590                - is_child_011: if( (dag >= 0) and (dag <= 11),
3591                                    True,
3592                                    False
3593                                  )
3594
3595                - nch_011: "if( MALE,
3596                                fc.count(is_child_011),
3597                                mc.count(is_child_011))"
3598
3599
3600
```

```
3601            #
3602            # EDUCATION PROCEDURE
3603            #
3604
3605
3606            education_process():
3607
3608
3609                #
3610                # EDUCATION - HIGHEST STATUS - [PhL] EVER REACHED
3611                #   0: Not completed Primary
3612                #   1: Primary
3613                #   2: Lower Secondary
3614                #   3: Upper Secondary
3615                #   4: Post Secondary
3616                #   5: Tertiary
3617
3618                - deh_ever_l0: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
3619                                   align( uniform(), DEH_EVER_0_P/100,
3620                                          take = deh_ever==DEH_NONE,
3621                                          leave = deh_ever > DEH_NONE or deh > DEH_NONE or
3622                                                  ( not(student_1) and dag > AGE_LOWEST_IN_EDUCATION )
3623                                        ),
3624                                   False
3625                                 )
3626                - deh_ever: if( deh_ever_l0, DEH_NONE, deh_ever)
3627
3628                - deh_ever_l1: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
3629                                   align( uniform(), DEH_EVER_1_P/100,
3630                                          take = deh_ever==DEH_P,
3631                                          leave = (deh_ever >=DEH_NONE and deh_ever!=DEH_P) or deh > DEH_P or
3632                                                  ( not(student_1) and dag > AGE_LOWEST_IN_EDUCATION )
3633                                        ),
3634                                   False
3635                                 )
3636                - deh_ever: if( deh_ever_l1, DEH_P, deh_ever)
3637
3638                - deh_ever_l2: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
3639                                   align( uniform(), DEH_EVER_2_P/100,
3640                                          take = deh_ever==DEH_LS,
3641                                          leave = (deh_ever >=DEH_NONE and deh_ever!=DEH_LS) or deh > DEH_LS or
3642                                                  ( not(student_1) and dag > AGE_LOWEST_IN_EDUCATION )
3643                                        ),
3644                                   False
3645                                 )
```

```
                - deh_ever: if( deh_ever_l2, DEH_LS, deh_ever)

                - deh_ever_l3: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                                   align( uniform(), DEH_EVER_3_P/100,
                                          take = deh_ever==DEH_US,
                                          leave = (deh_ever >=DEH_NONE and deh_ever!=DEH_US) or deh > DEH_US or
                                                  ( not(student_1) and dag > AGE_LOWEST_IN_EDUCATION )
                                       ),
                                   False
                                 )
                - deh_ever: if( deh_ever_l3, DEH_US, deh_ever)

                - deh_ever_l4: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                                   align( uniform(), DEH_EVER_4_P/100,
                                          take = deh_ever==DEH_PS,
                                          leave = (deh_ever >=DEH_NONE and deh_ever!=DEH_PS) or deh > DEH_PS or
                                                  ( not(student_1) and dag > AGE_LOWEST_IN_EDUCATION )
                                       ),
                                   False
                                 )
                - deh_ever: if( deh_ever_l4, DEH_PS, deh_ever)

                - deh_ever_l5: if( (dag >= AGE_LOWEST_IN_EDUCATION) and (dag <= AGE_MAX_DEH_EVER),
                                   align( uniform(), DEH_EVER_5_P/100,
                                          take = deh_ever==DEH_T,
                                          leave = (deh_ever >=DEH_NONE and deh_ever!=DEH_T) or
                                                  ( not(student_1) and dag > AGE_LOWEST_IN_EDUCATION )
                                       ),
                                   False
                                 )
                - deh_ever: if( deh_ever_l5, DEH_T, deh_ever)

                # MARKING THOSE WHO "TRY" REACHING A LEVEL, YET UNSUCESSFULLY
                # (<> "EVER" which implies "success")
                - deh_try_l1: if( deh_ever_l0, align( uniform(), DEH_TRY_1_P/100, take = deh_try_1 ), False )
                - deh_try_l2: if( deh_ever_l1, align( uniform(), DEH_TRY_2_P/100, take = deh_try_1 ), False )
                - deh_try_l3: if( deh_ever_l2, align( uniform(), DEH_TRY_3_P/100, take = deh_try_1 ), False )
                - deh_try_l5: if( deh_ever_l3 or deh_ever_l4, align( uniform(), DEH_TRY_5_P/100, take = deh_try_1 ), False
)
                - deh_try: False
                - deh_try: deh_try_l1 or deh_try_l2 or deh_try_l3 or deh_try_l5

                # EDUCATION - CURRENT STATUS
                #   0: Not in Education
                #   1: Pre-primary
```

```
#   2: Primary
#   3: Lower Secondary
#   4: Upper Secondary
#   5: Post Secondary
#   6: Tertiary"
#

- dec: if( dag < AGE_LOWEST_IN_EDUCATION,
           DEC_PRE_P,
           if( STUDENT_1 or dag == AGE_LOWEST_IN_EDUCATION,
                if( ( dag <= AGE_HIGHEST_PRIMARY ) and ( deh_ever >= DEH_NONE ),
                      DEC_P,
                      if( dag <= AGE_HIGHEST_COMPULSORY_EDUCATION
                          and (deh_ever == DEH_NONE or ( deh_ever == DEH_P and not(deh_try) ) ),
                          DEC_P,
                          if( ( dag <= AGE_HIGHEST_LOWER_SECONDARY )
                                and ( (deh_ever >= DEH_LS) or (deh_ever == DEH_P and deh_try) ),
                                DEC_LS,
                                if( ( dag <= AGE_HIGHEST_UPPER_SECONDARY )
                                      and ( (deh_ever >= DEH_US) or (deh_ever == DEH_LS and deh_try) ),
                                      DEC_US,
                                      if( (dag <= AGE_HIGHEST_TERTIARY) and
                                          ( (deh_ever == DEH_T) or
                                          ( (deh_ever == DEH_US or deh_ever == DEC_PS) and deh_try ) ),
                                           DEC_T,
                                           DEC_NONE
                                        )
                                    )
                                )
                            ),
                        DEC_NONE
                      )
                )

# STUDY AND LIVE ABROAD
- decrg: if( dec == DEC_T,
             align( uniform(), 0.65,
                    take = ( lag(dec) == DEC_T ) and lag(decrg)
                  ),
             False
           )

# EDUCATION - HIGHEST STATUS
# (SILC PE040 => highest level the person has successfully completed)
```

```
3736                #   0: Not completed Primary
3737                #   1: Primary
3738                #   2: Lower Secondary
3739                #   3: Upper Secondary
3740                #   4: Post Secondary
3741                #   5: Tertiary
3742
3743                - deh: if( student_1,
3744                           if( (dag > AGE_HIGHEST_TERTIARY) and deh_ever == DEH_T,
3745                               max( deh, DEH_T ),
3746                               if( (dag > AGE_HIGHEST_UPPER_SECONDARY) and deh_ever == DEH_PS,
3747                                   max( deh, DEH_PS ),
3748                                   if( (dag > AGE_HIGHEST_UPPER_SECONDARY) and deh_ever == DEH_US,
3749                                        max( deh, DEH_US ),
3750                                        if( (dag > AGE_HIGHEST_LOWER_SECONDARY) and deh_ever == DEH_LS,
3751                                             max( deh, DEH_LS ),
3752                                             if( (dag > AGE_HIGHEST_PRIMARY) and deh_ever == DEH_P,
3753                                                  max( deh, DEH_P ),
3754                                                  max( deh, DEH_NONE )
3755                                               )
3756                                          )
3757                                     )
3758                                )
3759                             ),
3760                           if( dag < AGE_LOWEST_IN_EDUCATION,
3761                               DEH_NONE,
3762                               deh
3763                             )
3764                         )
3765
3766                - student: if( (dec > 1) or (dag >= AGE_LOWEST_IN_EDUCATION and dag <= AGE_HIGHEST_COMPULSORY_EDUCATION),
3767                               True,
3768                               False
3769                             )
3770
3771                # EDUCATION - (YEAR) WHEN ACHIEVED HIGHEST STATUS
3772                - dew: if( dag < AGE_LOWEST_IN_EDUCATION,
3773                           -1,
3774                           if( student_1 and not(student),
3775                               year - 1,
3776                               min( dew, period )
3777                             )
3778                         )
3779
3780                # EDUCATION - NUMBER OF YEARS (REALLY GONE THROUGH, WHATEVER SUCCESSFUL or NOT)
```

```
                - dey: if( dew > 0,
                           ( dew - (year - dag) - 6 ) + 1,
                           -1
                         )

                - les: if( student,
                           SET_STUDENT,
                           if( dag < AGE_LOWEST_IN_EDUCATION,
                               SET_IN_PRE_SCHOOL,
                               if( student_1,
                                   SET_OTHER_STATUS,
                                   les
                                 )
                             )
                         )



            ###################################
            #  IMMIGRATION  -  PARTIM "PERSONS" #
            ###################################


            #
            # CLONING, INCLUDING UPDATE OF LINKS
            #


            immigration_person_process():


                - is_new_clone_immigration_downstream: "False"

                - pp_clone_immigration_child_id:
                    clone(   filter = ph.hh_now_to_be_cloned_for_immigration,
                             dcz = if( trunc( ph.get(persons.avg( dcz, id == ph.hh_head_id) ) ) ==  1,
                                       choice([2, 3], [ DCZ_EU_P/(DCZ_EU_P+DCZ_NON_EU_P) , DCZ_NON_EU_P/(DCZ_EU_P+
DCZ_NON_EU_P) ]),
                                       trunc( ph.get(persons.avg( dcz, id == ph.hh_head_id) ) )),
                             pp_clone_immigration_child_id = -1,
                             h_id = ph.hh_clone_immigration.id,
                             pp_immigrant = True,
```

```
                        has_a_clone_immigration_upstream = True,
                        is_new_clone_immigration_downstream = True,
                        original_clone_immigration_pp_id = id,
                        year_of_immigration = period
                    )

        - s_id: if(is_new_clone_immigration_downstream, ps.pp_clone_immigration_child_id, s_id)

        - m_id: if(is_new_clone_immigration_downstream, pm.pp_clone_immigration_child_id, m_id)

        - f_id: if(is_new_clone_immigration_downstream, pf.pp_clone_immigration_child_id, f_id)



    #
    # ADAPTING VARIABLES AFTER "PERSON" IMMIGRATION
    #



    after_immigration_person_process():


        - idperson: if( is_new_clone_immigration_downstream, id, idperson )
        - idhh: if( is_new_clone_immigration_downstream, h_id, idhh )
        - idpartner: if( is_new_clone_immigration_downstream, s_id, idpartner )
        - idfather: if( is_new_clone_immigration_downstream, f_id, idfather )
        - idmother: if( is_new_clone_immigration_downstream, m_id, idmother )

        - adult_oecd: if( is_new_clone_immigration_downstream, False, adult_oecd )
        - bched01_s: if( is_new_clone_immigration_downstream, 0, bched01_s )
        - bched04_s: if( is_new_clone_immigration_downstream, 0, bched04_s )
        - bed_s: if( is_new_clone_immigration_downstream, 0, bed_s )
        - bfauc_s: if( is_new_clone_immigration_downstream, 0, bfauc_s )
        - bho_s: if( is_new_clone_immigration_downstream, 0, bho_s )
        - bpact_s: if( is_new_clone_immigration_downstream, 0, bpact_s )
        - bplct_s: if( is_new_clone_immigration_downstream, 0, bplct_s )
        - bsacm_s: if( is_new_clone_immigration_downstream, 0, bsacm_s )
        - bsaho_s: if( is_new_clone_immigration_downstream, 0, bsaho_s )
        - bsaht_s: if( is_new_clone_immigration_downstream, 0, bsaht_s )
        - bunmy_s: if( is_new_clone_immigration_downstream, -1, bunmy_s )
        - bunss_s: if( is_new_clone_immigration_downstream, 0, bunss_s )
        - child_oecd: if( is_new_clone_immigration_downstream, False, child_oecd )
        - ddp_s: if( is_new_clone_immigration_downstream, False, ddp_s )
        - deciles_categories_rh_equiv_inc: if( is_new_clone_immigration_downstream, -1,
```

```
3871  deciles_categories_rh_equiv_inc )
3872                  - deciles_rh_equiv_inc: if( is_new_clone_immigration_downstream, 0, deciles_rh_equiv_inc )
3873                  - drsyy: if( is_new_clone_immigration_downstream, 0, drsyy )
3874                  - dta_s: if( is_new_clone_immigration_downstream, -1, dta_s )
3875                  - dwt_int: if( is_new_clone_immigration_downstream, -1, dwt_int )
3876                  - e20ps_o: if( is_new_clone_immigration_downstream, -1, e20ps_o )
3877                  - e20pslw_o: if( is_new_clone_immigration_downstream, -1, e20pslw_o )
3878                  - e20psmd_o: if( is_new_clone_immigration_downstream, False, e20psmd_o )
3879                  - e20pspo_o: if( is_new_clone_immigration_downstream, False, e20pspo_o )
3880                  - freq_weights: if( is_new_clone_immigration_downstream, -1, freq_weights )
3881                  - hh_ils_dispy: if( is_new_clone_immigration_downstream, 0, hh_ils_dispy )
3882                  - idorighh: if( is_new_clone_immigration_downstream, -1, idorighh )
3883                  - idorigperson: if( is_new_clone_immigration_downstream, -1, idorigperson )
3884                  - il_bho_disposable: if( is_new_clone_immigration_downstream, 0, il_bho_disposable )
3885                  - il_earncouple: if( is_new_clone_immigration_downstream, 0, il_earncouple )
3886                  - il_heating: if( is_new_clone_immigration_downstream, 0, il_heating )
3887                  - il_repl: if( is_new_clone_immigration_downstream, 0, il_repl )
3888                  - il_sadisregard: if( is_new_clone_immigration_downstream, 0, il_sadisregard )
3889                  - il_saother: if( is_new_clone_immigration_downstream, 0, il_saother )
3890                  - il_tax_inc_bef: if( is_new_clone_immigration_downstream, 0, il_tax_inc_bef )
3891                  - il_taxdedsic: if( is_new_clone_immigration_downstream, 0, il_taxdedsic )
3892                  - il_taxinc: if( is_new_clone_immigration_downstream, 0, il_taxinc )
3893                  - il_taxpen: if( is_new_clone_immigration_downstream, 0, il_taxpen )
3894                  - ils_b1_bcb: if( is_new_clone_immigration_downstream, 0, ils_b1_bcb )
3895                  - ils_b1_bdi: if( is_new_clone_immigration_downstream, 0, ils_b1_bdi )
3896                  - ils_b1_bed: if( is_new_clone_immigration_downstream, 0, ils_b1_bed )
3897                  - ils_b1_bfa: if( is_new_clone_immigration_downstream, 0, ils_b1_bfa )
3898                  - ils_b1_bhl: if( is_new_clone_immigration_downstream, 0, ils_b1_bhl )
3899                  - ils_b1_bho: if( is_new_clone_immigration_downstream, 0, ils_b1_bho )
3900                  - ils_b1_boa: if( is_new_clone_immigration_downstream, 0, ils_b1_boa )
3901                  - ils_b1_bsa: if( is_new_clone_immigration_downstream, 0, ils_b1_bsa )
3902                  - ils_b1_bsu: if( is_new_clone_immigration_downstream, 0, ils_b1_bsu )
3903                  - ils_b1_bun: if( is_new_clone_immigration_downstream, 0, ils_b1_bun )
3904                  - ils_b2_bfaed: if( is_new_clone_immigration_downstream, 0, ils_b2_bfaed )
3905                  - ils_b2_bsaho: if( is_new_clone_immigration_downstream, 0, ils_b2_bsaho )
3906                  - ils_b2_penhl: if( is_new_clone_immigration_downstream, 0, ils_b2_penhl )
3907                  - ils_base_tin: if( is_new_clone_immigration_downstream, 0, ils_base_tin )
3908                  - ils_ben: if( is_new_clone_immigration_downstream, 0, ils_ben )
3909                  - ils_benmt: if( is_new_clone_immigration_downstream, 0, ils_benmt )
3910                  - ils_bennt: if( is_new_clone_immigration_downstream, 0, ils_bennt )
3911                  - ils_bensim: if( is_new_clone_immigration_downstream, 0, ils_bensim )
3912                  - ils_dispy: if( is_new_clone_immigration_downstream, 0, ils_dispy )
3913                  - ils_earns: if( is_new_clone_immigration_downstream, 0, ils_earns )
3914                  - ils_origrepy: if( is_new_clone_immigration_downstream, 0, ils_origrepy )
3915                  - ils_origy: if( is_new_clone_immigration_downstream, 0, ils_origy )
```

```
3916                - ils_pen: if( is_new_clone_immigration_downstream, 0, ils_pen )
3917                - ils_sicct: if( is_new_clone_immigration_downstream, 0, ils_sicct )
3918                - ils_sicdy: if( is_new_clone_immigration_downstream, 0, ils_sicdy )
3919                - ils_sicee: if( is_new_clone_immigration_downstream, 0, ils_sicee )
3920                - ils_sicer: if( is_new_clone_immigration_downstream, 0, ils_sicer )
3921                - ils_sicot: if( is_new_clone_immigration_downstream, 0, ils_sicot )
3922                - ils_sicse: if( is_new_clone_immigration_downstream, 0, ils_sicse )
3923                - ils_tax: if( is_new_clone_immigration_downstream, 0, ils_tax )
3924                - ils_taxsim: if( is_new_clone_immigration_downstream, 0, ils_taxsim )
3925                - ils_udb_bdi: if( is_new_clone_immigration_downstream, 0, ils_udb_bdi )
3926                - ils_udb_bed: if( is_new_clone_immigration_downstream, 0, ils_udb_bed )
3927                - ils_udb_bfa: if( is_new_clone_immigration_downstream, 0, ils_udb_bfa )
3928                - ils_udb_bhl: if( is_new_clone_immigration_downstream, 0, ils_udb_bhl )
3929                - ils_udb_bho: if( is_new_clone_immigration_downstream, 0, ils_udb_bho )
3930                - ils_udb_boa: if( is_new_clone_immigration_downstream, 0, ils_udb_boa )
3931                - ils_udb_bsa: if( is_new_clone_immigration_downstream, 0, ils_udb_bsa )
3932                - ils_udb_bsu: if( is_new_clone_immigration_downstream, 0, ils_udb_bsu )
3933                - ils_udb_bun: if( is_new_clone_immigration_downstream, 0, ils_udb_bun )
3934                - ils_udb_kfbcc: if( is_new_clone_immigration_downstream, 0, ils_udb_kfbcc )
3935                - ils_udb_tis: if( is_new_clone_immigration_downstream, 0, ils_udb_tis )
3936                - ils_udb_tpr: if( is_new_clone_immigration_downstream, 0, ils_udb_tpr )
3937                - ils_udb_xmp: if( is_new_clone_immigration_downstream, 0, ils_udb_xmp )
3938                - ils_udb_yds: if( is_new_clone_immigration_downstream, 0, ils_udb_yds )
3939                - ils_udb_yem: if( is_new_clone_immigration_downstream, 0, ils_udb_yem )
3940                - ils_udb_yiy: if( is_new_clone_immigration_downstream, 0, ils_udb_yiy )
3941                - ils_udb_yot: if( is_new_clone_immigration_downstream, 0, ils_udb_yot )
3942                - ils_udb_ypp: if( is_new_clone_immigration_downstream, 0, ils_udb_ypp )
3943                - ils_udb_ypr: if( is_new_clone_immigration_downstream, 0, ils_udb_ypr )
3944                - ils_udb_ypt: if( is_new_clone_immigration_downstream, 0, ils_udb_ypt )
3945                - ils_udb_yse: if( is_new_clone_immigration_downstream, 0, ils_udb_yse )
3946                - is_poor_40: if( is_new_clone_immigration_downstream, False, is_poor_40 )
3947                - is_poor_50: if( is_new_clone_immigration_downstream, False, is_poor_50 )
3948                - is_poor_60: if( is_new_clone_immigration_downstream, False, is_poor_60 )
3949                - is_poor_70: if( is_new_clone_immigration_downstream, False, is_poor_70 )
3950                - lcl: if( is_new_clone_immigration_downstream, if(lcl==3, 2, lcl), lcl )
3951                - lcs: if( is_new_clone_immigration_downstream, False, lcs )
3952                - lcs01: if( is_new_clone_immigration_downstream, False, lcs01 )
3953                - lcs02: if( is_new_clone_immigration_downstream, False, lcs02 )
3954                - lhwsr_s_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, lhwsr_s_LU_2020_STD )
3955                - lhwsr_s_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, lhwsr_s_LU_2021_STD )
3956                - lhwsr_s_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, lhwsr_s_LU_2022_STD )
3957                - lhwsr_s_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, lhwsr_s_LU_2023_STD )
3958                - liwwh00: if( is_new_clone_immigration_downstream, -1, liwwh00 )
3959                - liwwh_f: if( is_new_clone_immigration_downstream, -1, liwwh_f )
3960                - lma20_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, lma20_LU_2020_STD )
```

```
3961                - lma21_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, lma21_LU_2021_STD )
3962                - lma22_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, lma22_LU_2022_STD )
3963                - lma23_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, lma23_LU_2023_STD )
3964                - lmcee_s_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, lmcee_s_LU_2020_STD )
3965                - lmcee_s_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, lmcee_s_LU_2021_STD )
3966                - lmcee_s_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, lmcee_s_LU_2022_STD )
3967                - lmcee_s_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, lmcee_s_LU_2023_STD )
3968                - lnu: if( is_new_clone_immigration_downstream, -1, lnu )
3969                - lnu_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, lnu_LU_2020_STD )
3970                - lnu_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, lnu_LU_2021_STD )
3971                - lnu_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, lnu_LU_2022_STD )
3972                - lnu_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, lnu_LU_2023_STD )
3973                - ltr: if( is_new_clone_immigration_downstream, -1, ltr )
3974                - lunmy_s: if( is_new_clone_immigration_downstream, -1, lunmy_s )
3975                - new_f_id: if( is_new_clone_immigration_downstream, -1, new_f_id )
3976                - new_h_id: if( is_new_clone_immigration_downstream, -1, new_h_id )
3977                - new_head_hh_id: if( is_new_clone_immigration_downstream, -1, new_head_hh_id )
3978                - new_id: if( is_new_clone_immigration_downstream, -1, new_id )
3979                - new_m_id: if( is_new_clone_immigration_downstream, -1, new_m_id )
3980                - new_s_id: if( is_new_clone_immigration_downstream, -1, new_s_id )
3981                - pdi00: if( is_new_clone_immigration_downstream, FACTOR_PENSION_FOREIGN_LU_RATIO * pdi00, pdi00 )
3982                - poapups: if( is_new_clone_immigration_downstream, FACTOR_PENSION_FOREIGN_LU_RATIO * poapups, poapups )
3983                - psu: if( is_new_clone_immigration_downstream, FACTOR_PENSION_FOREIGN_LU_RATIO * psupups, psu )
3984                - psupups: if( is_new_clone_immigration_downstream, FACTOR_PENSION_FOREIGN_LU_RATIO * psupups, psupups )
3985                - rh_eq_ils_dispy_oecd: if( is_new_clone_immigration_downstream, 0, rh_eq_ils_dispy_oecd )
3986                - rh_ils_dispy: if( is_new_clone_immigration_downstream, 0, rh_ils_dispy )
3987                - rh_nb_adults_oecd: if( is_new_clone_immigration_downstream, -1, rh_nb_adults_oecd )
3988                - rh_nb_children_oecd: if( is_new_clone_immigration_downstream, -1, rh_nb_children_oecd )
3989                - rh_weight_oecd: if( is_new_clone_immigration_downstream, 0, rh_weight_oecd )
3990                - sel_s: if( is_new_clone_immigration_downstream, 0, sel_s )
3991                - sin01_s: if( is_new_clone_immigration_downstream, 0, sin01_s )
3992                - sin02_s: if( is_new_clone_immigration_downstream, 0, sin02_s )
3993                - sin03_s: if( is_new_clone_immigration_downstream, 0, sin03_s )
3994                - sin04_s: if( is_new_clone_immigration_downstream, 0, sin04_s )
3995                - sin20_s: if( is_new_clone_immigration_downstream, 0, sin20_s )
3996                - sin21_s: if( is_new_clone_immigration_downstream, 0, sin21_s )
3997                - sin22_s: if( is_new_clone_immigration_downstream, 0, sin22_s )
3998                - sin23_s: if( is_new_clone_immigration_downstream, 0, sin23_s )
3999                - sin24_s: if( is_new_clone_immigration_downstream, 0, sin24_s )
4000                - sin25_s: if( is_new_clone_immigration_downstream, 0, sin25_s )
4001                - sin26_s: if( is_new_clone_immigration_downstream, 0, sin26_s )
4002                - sin27_s: if( is_new_clone_immigration_downstream, 0, sin27_s )
4003                - sin28_s: if( is_new_clone_immigration_downstream, 0, sin28_s )
4004                - tad: if( is_new_clone_immigration_downstream, 0, tad )
4005                - tin_s: if( is_new_clone_immigration_downstream, 0, tin_s )
```

```
4006                - tinta_s: if( is_new_clone_immigration_downstream, 0, tinta_s )
4007                - tintace_s: if( is_new_clone_immigration_downstream, 0, tintace_s )
4008                - tintadt_s: if( is_new_clone_immigration_downstream, 0, tintadt_s )
4009                - tintami_s: if( is_new_clone_immigration_downstream, 0, tintami_s )
4010                - tintamp02_s: if( is_new_clone_immigration_downstream, 0, tintamp02_s )
4011                - tintaox_s: if( is_new_clone_immigration_downstream, 0, tintaox_s )
4012                - tintape_s: if( is_new_clone_immigration_downstream, 0, tintape_s )
4013                - tintapesn_s: if( is_new_clone_immigration_downstream, 0, tintapesn_s )
4014                - tintapf_s: if( is_new_clone_immigration_downstream, 0, tintapf_s )
4015                - tintapv_s: if( is_new_clone_immigration_downstream, 0, tintapv_s )
4016                - tintb_s: if( is_new_clone_immigration_downstream, 0, tintb_s )
4017                - tintceent_s: if( is_new_clone_immigration_downstream, 0, tintceent_s )
4018                - tintclpnt_s: if( is_new_clone_immigration_downstream, 0, tintclpnt_s )
4019                - tintcmp_s: if( is_new_clone_immigration_downstream, 0, tintcmp_s )
4020                - tintcot_s: if( is_new_clone_immigration_downstream, 0, tintcot_s )
4021                - tintcpent_s: if( is_new_clone_immigration_downstream, 0, tintcpent_s )
4022                - tintcrt_s: if( is_new_clone_immigration_downstream, 0, tintcrt_s )
4023                - tintcsent_s: if( is_new_clone_immigration_downstream, 0, tintcsent_s )
4024                - tinty_s: if( is_new_clone_immigration_downstream, 0, tinty_s )
4025                - tinui_s: if( is_new_clone_immigration_downstream, 0, tinui_s )
4026                - tinxt_s: if( is_new_clone_immigration_downstream, 0, tinxt_s )
4027                - tis: if( is_new_clone_immigration_downstream, 0, tis )
4028                - tscct_s: if( is_new_clone_immigration_downstream, 0, tscct_s )
4029                - tsccthl_s: if( is_new_clone_immigration_downstream, 0, tsccthl_s )
4030                - tscctpi_s: if( is_new_clone_immigration_downstream, 0, tscctpi_s )
4031                - tscctsi_s: if( is_new_clone_immigration_downstream, 0, tscctsi_s )
4032                - tscee_s: if( is_new_clone_immigration_downstream, 0, tscee_s )
4033                - tsceebeot_s: if( is_new_clone_immigration_downstream, 0, tsceebeot_s )
4034                - tsceehl_s: if( is_new_clone_immigration_downstream, 0, tsceehl_s )
4035                - tsceeot_s: if( is_new_clone_immigration_downstream, 0, tsceeot_s )
4036                - tsceepbpi_s: if( is_new_clone_immigration_downstream, 0, tsceepbpi_s )
4037                - tsceepi_s: if( is_new_clone_immigration_downstream, 0, tsceepi_s )
4038                - tsceesi_s: if( is_new_clone_immigration_downstream, 0, tsceesi_s )
4039                - tscer: if( is_new_clone_immigration_downstream, 0, tscer )
4040                - tscer_s: if( is_new_clone_immigration_downstream, 0, tscer_s )
4041                - tscerac_s: if( is_new_clone_immigration_downstream, 0, tscerac_s )
4042                - tscerhl_s: if( is_new_clone_immigration_downstream, 0, tscerhl_s )
4043                - tscerpi_s: if( is_new_clone_immigration_downstream, 0, tscerpi_s )
4044                - tscersi_s: if( is_new_clone_immigration_downstream, 0, tscersi_s )
4045                - tscse_s: if( is_new_clone_immigration_downstream, 0, tscse_s )
4046                - tscseac_s: if( is_new_clone_immigration_downstream, 0, tscseac_s )
4047                - tscsehl_s: if( is_new_clone_immigration_downstream, 0, tscsehl_s )
4048                - tscseot_s: if( is_new_clone_immigration_downstream, 0, tscseot_s )
4049                - tscsepi_s: if( is_new_clone_immigration_downstream, 0, tscsepi_s )
4050                - tscsesi_s: if( is_new_clone_immigration_downstream, 0, tscsesi_s )
```

```
                - tu_bho_lu_headid: if( is_new_clone_immigration_downstream, -1, tu_bho_lu_headid )
                - tu_bho_lu_isdepchild: if( is_new_clone_immigration_downstream, False, tu_bho_lu_isdepchild )
                - tu_bho_lu_isloneparent: if( is_new_clone_immigration_downstream, False, tu_bho_lu_isloneparent )
                - tu_bho_lu_ispartner: if( is_new_clone_immigration_downstream, False, tu_bho_lu_ispartner )
                - tu_bsa_lu_headid: if( is_new_clone_immigration_downstream, -1, tu_bsa_lu_headid )
                - tu_bsa_lu_isdepchild: if( is_new_clone_immigration_downstream, False, tu_bsa_lu_isdepchild )
                - tu_bsa_lu_isloneparent: if( is_new_clone_immigration_downstream, False, tu_bsa_lu_isloneparent )
                - tu_bsa_lu_ispartner: if( is_new_clone_immigration_downstream, False, tu_bsa_lu_ispartner )
                - tu_cb_lu_headid: if( is_new_clone_immigration_downstream, -1, tu_cb_lu_headid )
                - tu_cb_lu_isdepchild: if( is_new_clone_immigration_downstream, False, tu_cb_lu_isdepchild )
                - tu_cb_lu_isloneparent: if( is_new_clone_immigration_downstream, False, tu_cb_lu_isloneparent )
                - tu_cb_lu_ispartner: if( is_new_clone_immigration_downstream, False, tu_cb_lu_ispartner )
                - tu_edugrant_lu_headid: if( is_new_clone_immigration_downstream, -1, tu_edugrant_lu_headid )
                - tu_edugrant_lu_isdepchild: if( is_new_clone_immigration_downstream, False, tu_edugrant_lu_isdepchild )
                - tu_edugrant_lu_isloneparent: if( is_new_clone_immigration_downstream, False, tu_edugrant_lu_isloneparent
)
                - tu_edugrant_lu_ispartner: if( is_new_clone_immigration_downstream, False, tu_edugrant_lu_ispartner )
                - tu_household_lu_headid: if( is_new_clone_immigration_downstream, -1, tu_household_lu_headid )
                - tu_household_lu_isdepchild: if( is_new_clone_immigration_downstream, False, tu_household_lu_isdepchild
)
                - tu_household_lu_isloneparent: if( is_new_clone_immigration_downstream, False,
tu_household_lu_isloneparent )
                - tu_household_lu_ispartner: if( is_new_clone_immigration_downstream, False, tu_household_lu_ispartner )
                - tu_tin_lu_headid: if( is_new_clone_immigration_downstream, -1, tu_tin_lu_headid )
                - tu_tin_lu_isdepchild: if( is_new_clone_immigration_downstream, False, tu_tin_lu_isdepchild )
                - tu_tin_lu_isloneparent: if( is_new_clone_immigration_downstream, False, tu_tin_lu_isloneparent )
                - tu_tin_lu_ispartner: if( is_new_clone_immigration_downstream, False, tu_tin_lu_ispartner )
                - value_deciles_rh_equiv_inc: if( is_new_clone_immigration_downstream, -1, value_deciles_rh_equiv_inc )
                - yds: if( is_new_clone_immigration_downstream, 0, yds )
                - ydses_o: if( is_new_clone_immigration_downstream, 0, ydses_o )
                - yem: if( is_new_clone_immigration_downstream, if(lcs == 1, FACTOR_YEM_MIGRANTS_NOW_NOT_CS * yem, yem),
yem )
                - yem_LU_2020_STD: if( is_new_clone_immigration_downstream, 0, yem_LU_2020_STD )
                - yem_LU_2020_STD_BASE: if( is_new_clone_immigration_downstream, 0, yem_LU_2020_STD_BASE )
                - yem_LU_2021_STD: if( is_new_clone_immigration_downstream, 0, yem_LU_2021_STD )
                - yem_LU_2021_STD_BASE: if( is_new_clone_immigration_downstream, 0, yem_LU_2021_STD_BASE )
                - yem_LU_2022_STD: if( is_new_clone_immigration_downstream, 0, yem_LU_2022_STD )
                - yem_LU_2022_STD_BASE: if( is_new_clone_immigration_downstream, 0, yem_LU_2022_STD_BASE )
                - yem_LU_2023_STD: if( is_new_clone_immigration_downstream, 0, yem_LU_2023_STD )
                - yem_LU_2023_STD_BASE: if( is_new_clone_immigration_downstream, 0, yem_LU_2023_STD_BASE )
                - yemmw_s_LU_2020_STD: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2020_STD )
                - yemmw_s_LU_2020_STD_BASE: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2020_STD_BASE )
                - yemmw_s_LU_2021_STD: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2021_STD )
                - yemmw_s_LU_2021_STD_BASE: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2021_STD_BASE )
                - yemmw_s_LU_2022_STD: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2022_STD )
```

```
                - yemmw_s_LU_2022_STD_BASE: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2022_STD_BASE )
                - yemmw_s_LU_2023_STD: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2023_STD )
                - yemmw_s_LU_2023_STD_BASE: if( is_new_clone_immigration_downstream, 0, yemmw_s_LU_2023_STD_BASE )
                - yemmwmy_s_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, yemmwmy_s_LU_2020_STD )
                - yemmwmy_s_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, yemmwmy_s_LU_2021_STD )
                - yemmwmy_s_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, yemmwmy_s_LU_2022_STD )
                - yemmwmy_s_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, yemmwmy_s_LU_2023_STD )
                - yemmy_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, yemmy_LU_2020_STD )
                - yemmy_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, yemmy_LU_2021_STD )
                - yemmy_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, yemmy_LU_2022_STD )
                - yemmy_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, yemmy_LU_2023_STD )
                - yempv: if( is_new_clone_immigration_downstream, FACTOR_YEMPV_MIG * yempv, yempv )
                - yempv_s: if( is_new_clone_immigration_downstream, 0, yempv_s )
                - yse_LU_2020_STD: if( is_new_clone_immigration_downstream, 0, yse_LU_2020_STD )
                - yse_LU_2020_STD_BASE: if( is_new_clone_immigration_downstream, 0, yse_LU_2020_STD_BASE )
                - yse_LU_2021_STD: if( is_new_clone_immigration_downstream, 0, yse_LU_2021_STD )
                - yse_LU_2021_STD_BASE: if( is_new_clone_immigration_downstream, 0, yse_LU_2021_STD_BASE )
                - yse_LU_2022_STD: if( is_new_clone_immigration_downstream, 0, yse_LU_2022_STD )
                - yse_LU_2022_STD_BASE: if( is_new_clone_immigration_downstream, 0, yse_LU_2022_STD_BASE )
                - yse_LU_2023_STD: if( is_new_clone_immigration_downstream, 0, yse_LU_2023_STD )
                - yse_LU_2023_STD_BASE: if( is_new_clone_immigration_downstream, 0, yse_LU_2023_STD_BASE )
                - ysemy_LU_2020_STD: if( is_new_clone_immigration_downstream, -1, ysemy_LU_2020_STD )
                - ysemy_LU_2021_STD: if( is_new_clone_immigration_downstream, -1, ysemy_LU_2021_STD )
                - ysemy_LU_2022_STD: if( is_new_clone_immigration_downstream, -1, ysemy_LU_2022_STD )
                - ysemy_LU_2023_STD: if( is_new_clone_immigration_downstream, -1, ysemy_LU_2023_STD )

                - active_in_input: if( is_new_clone_immigration_downstream, False, active_in_input )
                - bidon: if( is_new_clone_immigration_downstream, -1, bidon )
                - bidon_1: if( is_new_clone_immigration_downstream, -1, bidon_1 )
                - dag_input: if( is_new_clone_immigration_downstream, -1, dag_input )
                - dwt_in_raw: if( is_new_clone_immigration_downstream, 0, dwt_in_raw )
                - dwt_int_in_raw: if( is_new_clone_immigration_downstream, -1, dwt_int_in_raw )
                - feed_age_unemp_M: if( is_new_clone_immigration_downstream, -1, feed_age_unemp_M )
                - feed_age_unemp_F: if( is_new_clone_immigration_downstream, -1, feed_age_unemp_F )
                - has_a_clone_expand_upstream: if( is_new_clone_immigration_downstream, False, has_a_clone_expand_upstream
)
                - is_new_clone_expand_downstream: if( is_new_clone_immigration_downstream, False,
is_new_clone_expand_downstream )
                - original_clone_expand_pp_id: if( is_new_clone_immigration_downstream, -1, original_clone_expand_pp_id )
                - pp_clone_expand_child_id: if( is_new_clone_immigration_downstream, -1, pp_clone_expand_child_id )
                - pp_nb_copies_remaining: if( is_new_clone_immigration_downstream, -1, pp_nb_copies_remaining )
                - pp_to_be_cloned_expand: if( is_new_clone_immigration_downstream, False, pp_to_be_cloned_expand )
                - present_birth_in_model_child_id: if( is_new_clone_immigration_downstream, -1,
present_birth_in_model_child_id )
                - temp1_bool: if( is_new_clone_immigration_downstream, False, temp1_bool )
```

```
                - temp1_int: if( is_new_clone_immigration_downstream, -1, temp1_int )
                - temp2_int: if( is_new_clone_immigration_downstream, -1, temp2_int )
                - temp1_float: if( is_new_clone_immigration_downstream, 0, temp1_float )
                - temp2_float: if( is_new_clone_immigration_downstream, 0, temp2_float )
                - unemployed_in_input: if( is_new_clone_immigration_downstream, False, unemployed_in_input )
                - working_in_input: if( is_new_clone_immigration_downstream, False, working_in_input )
                - year_of_expansion: if( is_new_clone_immigration_downstream, -1, year_of_expansion )
                - pen_dur_compl_school_FO: if( is_new_clone_immigration_downstream, pen_dur_compl_school_LU,
pen_dur_compl_school_FO )
                - pen_dur_compl_school_FO_1: if( is_new_clone_immigration_downstream, pen_dur_compl_school_LU_1,
pen_dur_compl_school_FO_1 )
                - pen_dur_compl_school_LU: if( is_new_clone_immigration_downstream, 0, pen_dur_compl_school_LU )
                - pen_dur_compl_school_LU_1: if( is_new_clone_immigration_downstream, 0, pen_dur_compl_school_LU_1 )
                - pen_dur_eff_inwork_FO: if( is_new_clone_immigration_downstream, pen_dur_eff_inwork_LU +
pen_dur_eff_inwork_FO, pen_dur_eff_inwork_FO )
                - pen_dur_eff_inwork_FO_1: if( is_new_clone_immigration_downstream, pen_dur_eff_inwork_LU_1 +
pen_dur_eff_inwork_FO_1, pen_dur_eff_inwork_FO_1 )
                - pen_dur_eff_inwork_LU: if( is_new_clone_immigration_downstream, 0, pen_dur_eff_inwork_LU )
                - pen_dur_eff_inwork_LU_1: if( is_new_clone_immigration_downstream, 0, pen_dur_eff_inwork_LU_1 )
                - pen_dur_eff_repl_FO: if( is_new_clone_immigration_downstream, pen_dur_eff_repl_LU, pen_dur_eff_repl_FO
)
                - pen_dur_eff_repl_FO_1: if( is_new_clone_immigration_downstream, pen_dur_eff_repl_LU_1,
pen_dur_eff_repl_FO_1 )
                - pen_dur_eff_repl_LU: if( is_new_clone_immigration_downstream, 0, pen_dur_eff_repl_LU )
                - pen_dur_eff_repl_LU_1: if( is_new_clone_immigration_downstream, 0, pen_dur_eff_repl_LU_1 )



            #############
            #  LABOUR  #
            #############


            # A FEW INITIALIZATION



            #
            # RETIRING OR NOT ( NB : IS CONSIDERED AS "ABSORBING" PRIOR STATUS)
            #

            retirement_process():
```

```
4186
4187
4188                - temp1_bool: ( AGE_RETIREMENT_LEGAL_SMOOTHED and ( ( pen_dur_eff_LU + pen_dur_eff_FO ) >= PEN_DUR_LEGAL_MIN
4189    ) )
4190                                    or ( AGE_RETIREMENT_ANTICIPATED_HIGH_SMOOTHED and (pen_dur_tot >=
4191    PEN_DUR_ANTICIPATED_HIGH_TOT)
4192                                         and ( (pen_dur_eff_LU + pen_dur_eff_FO) >= PEN_DUR_ANTICIPATED_HIGH_COMPULS) )
4193                                    or ( AGE_RETIREMENT_ANTICIPATED_LOW_SMOOTHED
4194                                         and ( (pen_dur_eff_LU + pen_dur_eff_FO) >= PEN_DUR_ANTICIPATED_LOW_COMPULS) )
4195                - temp1_float: if( CIVIL_SERVANT_STATUS_1,
4196                                   PENS_OLD_AGE_PUBLIC_AVG_RATIO,
4197                                   PENS_OLD_AGE_PRIVATE_AVG_RATIO
4198                                 )
4199                                 *
4200                               if( FEMALE,
4201                                   if( not( is_age_60 ),
4202                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 0, deh_ever],
4203                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 1, deh_ever]
4204                                     ),
4205                                   if( not( is_age_60 ),
4206                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 0, deh_ever],
4207                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 1, deh_ever]
4208                                     )
4209                                 )
4210
4211                - les: if( ( AGE_RETIREMENT_LEGAL_SMOOTHED and ( ( pen_dur_eff_LU + pen_dur_eff_FO ) >= PEN_DUR_LEGAL_MIN
4212    ) )
4213                               or ( AGE_RETIREMENT_ANTICIPATED_HIGH_SMOOTHED and (pen_dur_tot >=
4214    PEN_DUR_ANTICIPATED_HIGH_TOT)
4215                                    and ( (pen_dur_eff_LU + pen_dur_eff_FO) >= PEN_DUR_ANTICIPATED_HIGH_COMPULS) )
4216                               or ( AGE_RETIREMENT_ANTICIPATED_LOW_SMOOTHED
4217                                    and ( (pen_dur_eff_LU + pen_dur_eff_FO) >= PEN_DUR_ANTICIPATED_LOW_COMPULS) ),
4218                                    SET_PENSIONER,
4219                               les)
4220
4221                - new_pensioner: if( ( les == SET_PENSIONER )
4222                                       and ( not( les_1 == SET_PENSIONER ) ),
4223                                       True,
4224                                       False
4225                                   )
4226
4227                - pensioner: les == SET_PENSIONER
4228                - pensioner_statutory: if( new_pensioner and CIVIL_SERVANT_STATUS_1 ,
4229                                           True,
4230                                           pensioner_statutory
```

```
                                        )
4231
4232
4233
4234
4235
4236            ##############################################
4237            #  AFTER LABOR & BEFORE PENSIONS  -  STATUSES #
4238            ##############################################
4239
4240
4241
4242            #
4243            # UNEMPLOYMENT OR WORK
4244            ######################
4245
4246
4247            unemployment_and_working_process():
4248
4249
4250                # GENERALITIES
4251                # ------------
4252
4253                # INITIALIZING UNEMPLOYMENT
4254                # -------------------------
4255
4256                # "UNEMP_TO_STILL_FIX" : "unemployement_score==-1"
4257                # - unemployed: if(AGE_UNEMPLOYMENT and active_in_input, False, unemployed_in_input)
4258                # ACTIVE_1: lag(les) == 1 or lag(les) == 2 or lag(les) == 3 or lag(les) == 5
4259                # "AGE_UNEMPLOYMENT": formerly "( dag >= 20 )  and  ( dag <= 64 )"
4260                #   NOW => AGE_UNEMPLOYMENT_SMOOTHED: ( dag >= 20 ) and ( dag < if( dgn, feed_age_unemp_M, feed_age_unemp_F
4261    ) )
4262                # "WORKING": working ==  1
4263                # "YOUNG_GRADUATED": not(STUDENT) and lag(STUDENT)
4264
4265                - feed_age_unemp_M: trunc(PEN_LEGAL_AGE_M) + max( min( 3 - ( period - 2018 ), 3 ), 0 )
4266                - feed_age_unemp_F: trunc(PEN_LEGAL_AGE_M) + max( min( 3 - ( period - 2018 ), 3 ), 0 )
4267                - unemployed: False
4268                - unemployment_score: "-1"
4269                - working: False
4270
4271
4272                # FEMALES
4273
4274                - unemployment_score: "if( FEMALE and UNEMPLOYED_1 and not(PENSIONER or YOUNG_GRADUATED or NEW_IMMIGRANT)
4275                                        and AGE_UNEMPLOYMENT_SMOOTHED and UNEMP_TO_STILL_FIX,
```

```
4276                                        logit_score(1000),
4277                                        unemployment_score)"
4278
4279                - unemployment_score: "if( FEMALE and WORKING_1 and not(PENSIONER or YOUNG_GRADUATED or NEW_IMMIGRANT)
4280                                        and AGE_UNEMPLOYMENT_SMOOTHED and UNEMP_TO_STILL_FIX,
4281                                        logit_score(0.5476218 * dag -0.0079123 * dag2 + 0.5241817 * nch_011 -7.079126),
4282                                        unemployment_score)"
4283
4284                # YOUNG GRADUATED & NEW IMMIGRANTS
4285                - unemployment_score: "if( FEMALE and (YOUNG_GRADUATED or NEW_IMMIGRANT) and AGE_UNEMPLOYMENT_SMOOTHED and
4286    UNEMP_TO_STILL_FIX,
4287                                        logit_score(-1000),
4288                                        unemployment_score)"
4289
4290
4291                # MALES
4292
4293                - unemployment_score: "if( MALE and UNEMPLOYED_1 and not(PENSIONER or YOUNG_GRADUATED or NEW_IMMIGRANT)
4294                                        and AGE_UNEMPLOYMENT_SMOOTHED and UNEMP_TO_STILL_FIX,
4295                                        logit_score(1000),
4296                                        unemployment_score)"
4297
4298                - unemployment_score: "if( MALE and WORKING_1 and not(PENSIONER or YOUNG_GRADUATED or NEW_IMMIGRANT)
4299                                        and AGE_UNEMPLOYMENT_SMOOTHED and UNEMP_TO_STILL_FIX ,
4300                                        logit_score(0.2063094 * dag -0.0029645 * dag2 -1.326891 * IN_COUPLE -2.223461),
4301                                        unemployment_score)"
4302
4303                # YOUNG GRADUATED
4304                - unemployment_score: "if( MALE and (YOUNG_GRADUATED or NEW_IMMIGRANT) and AGE_UNEMPLOYMENT_SMOOTHED and
4305    UNEMP_TO_STILL_FIX,
4306                                        logit_score(-1000),
4307                                        unemployment_score)"
4308
4309                # SETTING UNEMPLOYMENT STATUS FOR ALL
4310
4311                - unemployed: if( (les==1 or les==2 or les==3 or les==5 or YOUNG_GRADUATED) and AGE_UNEMPLOYMENT_SMOOTHED,
4312                                  align( unemployment_score,
4313                                         AL_P_UNEMPLOYED,
4314                                         leave = CIVIL_SERVANT_WORKING_1
4315                                       ),
4316                                  unemployed
4317                                )
4318
4319                - working: if( (les==1 or les==2 or les==3 or les==5 or YOUNG_GRADUATED) and not(UNEMPLOYED),
4320                                True,
```

```
                                    working
                                )

                - new_worker: if( WORKING and not(WORKING_1), True, False )

                # UPDATING STATUS (based on observations in INPUT DATA SET = INCOME YEAR 2017)

                - lcl: if( new_worker and loc > 0,
                           1,
                           lcl )

                - les: if( UNEMPLOYED,
                           SET_UNEMPLOYED,
                           if( new_worker,
                               if( MALE,
                                   if( deh == 0, choice([2, 3], [39/(39+2388), 2388/(39+2388)]),
                                       if( deh == 1, choice([2, 3], [379/(379+12548), 12548/(379+12548)]),
                                           if( deh == 2, choice([2, 3], [688/(688+17936), 17936/(688+17936)]),
                                               if( deh == 3, choice([2, 3], [3653/(3653+46743), 46743/(3653+46743)]),
                                                   if( deh == 4, choice([2, 3], [318/(318+3421), 3421/(318+3421)]),
                                                       if( deh == 5, choice([2, 3], [4309/(4309+41782),
41782/(4309+41782)]), les )
                                                   )
                                               )
                                           )
                                       )
                                   ),
                                   if( deh == 0, 3,
                                       if( deh == 1, choice([2, 3], [364/(364+10890), 10890/(364+10890)]),
                                           if( deh == 2, choice([2, 3], [412/(412+14283), 14283/(412+14283)]),
                                               if( deh == 3, choice([2, 3], [1313/(1313+36969), 36969/(1313+36969)]),
                                                   if( deh == 4, choice([2, 3], [70/(70+679), 679/(70+679)]),
                                                       if( deh == 5, choice([2, 3], [3040/(3040+37488),
37488/(3040+37488)]), les )
                                                   )
                                               )
                                           )
                                       )
                                   )
                               ),
                               les
                           )
                       )

                - employee: "les == SET_EMPLOYEE"
```

```
    - new_unemployed: if( UNEMPLOYED and not(UNEMPLOYED_1), True, False )
    - self_employed: "les == SET_SELF_EMPLOYED"

    # END OF "UNEMPLOYMENT_AND_WORKING_PROCESS"



#
# POST  "UNEMPLOYENT & WORK" TREATMENT
#####################################


post_unemployment_and_working_process():



    # PARTIALLY INSPIRED FROM "LMA ADD-ON" in EUROMOD

    # "BHL" (Health Benefits) will feed "LMA Add-on"
    - bhl: if( new_worker or NEW_PENSIONER,
               0,
               bhl
             )

    # BUNSS_HYPOTHETICAL_VALUE: "1000" (the idea here is just to give a POSITIVE value to BUNSS,
    - bunss: if( new_unemployed,
                 if( liwwh > 0,
                     BUNSS_HYPOTHETICAL_VALUE,
                     0
                   ),
                 if( new_worker or NEW_PENSIONER,
                     0,
                     bunss
                   )
               )

    # "KFB" (Fringe Benefits) & "KFBMY" will feed "LMA Add-on"
    - kfb: if( new_unemployed or NEW_PENSIONER,
               0,
               kfb
             )

    - kfbmy: if( new_unemployed or NEW_PENSIONER,
                 0,
```

```
                kfbmy
              )

  # "KFBCC" (Company Car) will feed "LMA Add-on"
  - kfbcc: if( new_unemployed or NEW_PENSIONER,
              0,
              kfbcc )

  - lcs: if( EMPLOYEE,
              align( uniform(), 0.0626,
                     take = CIVIL_SERVANT_STATUS_1
                   ),
              lcs
            )

  - yemmy: if( new_unemployed or NEW_PENSIONER,
                0,
                if( new_worker,
                    if( SELF_EMPLOYED, 0, 12 ),
                    yemmy
                  )
              )

  - ysemy: if( new_unemployed or NEW_PENSIONER,
                0,
                if( new_worker,
                    if( SELF_EMPLOYED, 12 , 0 ),
                    ysemy
                  )
              )

  - liwmy: yemmy + ysemy

  - liwwh: if( new_unemployed or NEW_PENSIONER,
                liwwh,
                liwwh + liwmy
                )
  - liwwh00: liwwh00 + liwwh - lag(liwwh)

  # "LMA20" will feed "LMA" in LMA Add-on : "0" if no change,
  #                                         "1" if new worker (not employed => worker),
  #                                         "2" if short-term unemployed,
  #                                         "3" if short-term to long-term unemployed, etc
  - lma20: if( UNEMPLOYED,
                if( UNEMPLOYED_1,
```

```
                        3,
                        2
                    ),
                  if( new_worker,
                      1,
                      0
                    )
                )

    # "LNU" (New unemployed)will feed "LMA" in LMA Add-on
    - lnu: if( new_unemployed,
                1,
                0
              )

    # "LOWAS" (Out of work & Actively seeking job)
    - lowas: if( UNEMPLOYED,
                  True,
                  if( new_worker or NEW_PENSIONER,
                      False,
                      lowas
                    )
                )

    # Months "working" transferred to "unemployed"
    - lunmy: if( new_unemployed,
                  min( lag(lunmy) + lag(liwmy), 12 ),
                  if( new_worker or NEW_PENSIONER,
                      0,
                      lunmy
                    )
                )


    # "BUNMY" will feed "LMA Add-on"
    - bunmy: if( new_unemployed,
                  lunmy,
                  if( new_worker or NEW_PENSIONER,
                      0,
                      bunmy
                    )
                )

    # NB => The strategy differs from the one in "LMA Add-on" from EUROMOD
    - bunssmy: if( new_unemployed,
```

```
                        if( bunss > 0,
                            lunmy,
                            0
                          ),
                        if( new_worker or NEW_PENSIONER,
                            0,
                            bunssmy
                          )
                      )

    # Not necessary given that I can just keep former YEMPV on
    - yempv: if( NEW_PENSIONER,
                  0,
                  if( new_unemployed,
                      max( lag(yem) + lag(yse), yempv ),
                      yempv
                    )
                )

    - yem: if( new_unemployed or NEW_PENSIONER,
                0,
                if( new_worker and EMPLOYEE,
                    if( yempv > 0,
                        yempv,
                        if( FEMALE,
                            FACTOR_YEM_BROADLY_IMPUTED_FEMALE *
                             TAB_YEM_BASE_2017_NOT_UPRATED[0, CLASS_AGE_EUROMOD_LIAM2_FINE_ORDER, deh_ever],
                            FACTOR_YEM_BROADLY_IMPUTED_MALE *
                             TAB_YEM_BASE_2017_NOT_UPRATED[1, CLASS_AGE_EUROMOD_LIAM2_FINE_ORDER, deh_ever]
                          )
                      ),
                    if( new_worker and SELF_EMPLOYED,
                        0,
                        yem
                      )
                  )
              )

    - yse: if( new_unemployed or NEW_PENSIONER,
                0,
                if( new_worker,
                    if( SELF_EMPLOYED,
                        if( yempv > 0,
                            yempv,
                            if( FEMALE,
```

```
                                        FACTOR_YSE_BROADLY_IMPUTED_FEMALE
                                         * TAB_YSE_BASE_2017_NOT_UPRATED[0, CLASS_AGE_EUROMOD_LIAM2_FINE_ORDER,
deh_ever],
                                        FACTOR_YSE_BROADLY_IMPUTED_MALE
                                         * TAB_YSE_BASE_2017_NOT_UPRATED[1, CLASS_AGE_EUROMOD_LIAM2_FINE_ORDER,
deh_ever]
                                    )
                                ),
                            0
                        ),
                    yse
                )
            )

        # "YSV" (Severance Payment = Montants indemnités de préavis et/ou de licenciement),
        - ysv: if( new_unemployed,
                   ysv,
                   if( new_worker or NEW_PENSIONER,
                       0,
                       ysv
                     )
                 )

        - show("\n TABULATE LIMWY", groupby(liwmy), "\n TABULATE LUNMY", groupby(lunmy), "\n TABULATE BUNSSMY",
groupby(bunssmy) )

        # END OF POST TREATMENT



    ##############################################
    # BENEFITS MODULE - PART A : BEFORE PENSIONS #
    ##############################################



    ######################################
    # BENEFITS MODULE - PART B : PENSIONS #
    ######################################


    #
```

```
4591            # OLD-AGE PENSIONS
4592            ###################
4593
4594
4595
4596            #*****************************
4597            # ACCUMULATION OF PENSION RIGHTS
4598            #*****************************
4599
4600
4601            old_age_pensions_rights_process:
4602
4603
4604                # DURATIONS
4605
4606                # TOTAL DURATION = EFFECTIVE + COMPLEMENTARY
4607                #       EFFECTIVE = COMPULSORY + VOLUNTARY
4608                #           COMPULSORY = INWORK + REPLACEMENT (Unemployment + "pré-retraite"/not taken into account)
4609                #           VOLUNTARY : not taken here into account yet
4610                #       COMPLEMENTARY = DISABLEDITY + SCHOOL [18-27] + CHILDREN (not yet taken into account here)
4611
4612                # "LUxemburgish" CAREER
4613
4614                # BASEMENTS FIRST...
4615                - pen_dur_eff_inwork_LU: max( liwwh - pen_dur_eff_inwork_FO, 0 )
4616                - pen_dur_eff_repl_LU: pen_dur_eff_repl_LU_1 + bunmy
4617                - temp1_int: ( dew - (year - dag) - AGE_LOW_LIMIT_PENSION_SCHOOL ) * 12
4618                - temp2_int: AGE_HIGHEST_UPPER_SECONDARY - AGE_LOW_LIMIT_PENSION_SCHOOL
4619                # if "DEW" > 0, we impose a MAX to "pen_dur_compl_school_LU" because DEW (year of diploma) can be very late
4620    in life
4621                - pen_dur_compl_school_LU: if( dew <= 0,
4622                                               if( LOWER_SECONDARY_DEH_EVER and (dag >= AGE_LOW_LIMIT_PENSION_SCHOOL)
4623                                                   and (deh_try) and (dag >= AGE_HIGHEST_UPPER_SECONDARY),
4624                                                   max( temp2_int, 0 ) * 12,
4625                                                   if( (UPPER_SECONDARY_DEH_EVER or POST_SECONDARY_DEH_EVER) and (dag >=
4626    AGE_LOW_LIMIT_PENSION_SCHOOL),
4627                                                        if( deh_try and (dag >= AGE_HIGHEST_TERTIARY),
4628                                                            (AGE_HIGHEST_TERTIARY - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12,
4629                                                            if( not(deh_try) and (dag >= AGE_HIGHEST_UPPER_SECONDARY),
4630                                                                max( temp2_int, 0 ) * 12,
4631                                                                pen_dur_compl_school_LU
4632                                                            )
4633                                                        ),
4634                                                        if( TERTIARY_DEH_EVER and (dag >= AGE_LOW_LIMIT_PENSION_SCHOOL)
4635                                                            and (dag >= AGE_HIGHEST_TERTIARY),
```

```
                                                                    (AGE_HIGHEST_TERTIARY - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12,
                                                                    pen_dur_compl_school_LU
                                                                )
                                                            )
                                                        ),
                                                    if( ( LOWER_SECONDARY_DEH_EVER and (deh_try) )
                                                        or UPPER_SECONDARY_DEH_EVER or POST_SECONDARY_DEH_EVER,
                                                        min( max( temp1_int, 0), (AGE_HIGHEST_UPPER_SECONDARY -
AGE_LOW_LIMIT_PENSION_SCHOOL) * 12 ),
                                                        if( TERTIARY_DEH_EVER,
                                                            min( max( temp1_int, 0), (AGE_HIGHEST_TERTIARY -
AGE_LOW_LIMIT_PENSION_SCHOOL) * 12 ),
                                                            pen_dur_compl_school_LU
                                                        )
                                                    )
                                                )
                # Complementary ADJUSTEMENT on "PEN_DUR_COMPL_SCHOOL..." if the expected TOTAL DURATION
                #   AFTER "AGE_LOW_LIMIT_PENSION_SCHOOL" > what is feasible based on age ("DAG")
                - pen_dur_compl_school_LU: if( (pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU + pen_dur_compl_school_LU)
                                               > (dag - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12,
                                               max( pen_dur_compl_school_LU -
                                                    ( (pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU +
pen_dur_compl_school_LU)
                                                       - (dag - AGE_LOW_LIMIT_PENSION_SCHOOL) * 12
                                                    ),
                                                    0
                                               ),
                                               pen_dur_compl_school_LU
                                           )
                # ... THEN, COMPOSED DURATION VARs
                - pen_dur_compl_LU: pen_dur_compl_school_LU
                - pen_dur_eff_LU: pen_dur_eff_inwork_LU + pen_dur_eff_repl_LU
                - pen_dur_tot_LU: pen_dur_eff_LU + pen_dur_compl_LU
                - pen_dur_tot: pen_dur_tot_LU + pen_dur_tot_FO

                # ... and DOWNSTREAM VARIABLE
                - pen_dur_coef_max:
                    min( 1.0 - (1/trunc(PEN_DUR_LEGAL_TOT/12))
                                * trunc( ( PEN_DUR_LEGAL_TOT - pen_dur_tot ) / 12 ),
                         1.0
                       )


            #*************************************
```

```
4681            # NEW PENSION BENEFITS FOR NEW PENSIONERS
4682            #*************************************
4683
4684
4685            old_age_pensions_new_benefits_process():
4686
4687
4688
4689                # "FACTOR_PENSION_BROADLY_IMPUTED_FEMALE" are CORRECTIVE FACTORS for making PENSIONS
4690                #   as if SIMULATED for NOT NEW_PENSIONERS in 2018
4691                #   reaching AVG(PENSION for NOT NEW_PENSIONERS) as "observed" (coming +/- from 2017)
4692                - poapups: if( new_pensioner,
4693                               if( FEMALE,
4694                                   FACTOR_PENSION_BROADLY_IMPUTED_FEMALE,
4695                                   FACTOR_PENSION_BROADLY_IMPUTED_MALE
4696                                 )
4697                                 *
4698                               if( pensioner_statutory,
4699                                   PENS_OLD_AGE_PUBLIC_AVG_RATIO,
4700                                   PENS_OLD_AGE_PRIVATE_AVG_RATIO
4701                                 )
4702                                 *
4703                               if( FEMALE,
4704                                   if( not( is_age_60 ),
4705                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 0, deh_ever],
4706                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 1, deh_ever]
4707                                     ),
4708                                   if( not( is_age_60 ),
4709                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 0, deh_ever],
4710                                       TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 1, deh_ever]
4711                                     )
4712                                 ),
4713                               poapups
4714                             )
4715                # "forfait d'éducation" or mammenrent
4716                - poacc: if( new_pensioner, 0.0, poacc )
4717                # "2nd pilier"
4718                - poacm: if( new_pensioner, 0.0, poacm )
4719                # Complementary pension for MINERS and metal workers
4720                - poamr: if( new_pensioner, 0.0, poamr )
4721                # Now integrated in "poapups"
4722                - poamy: if( new_pensioner, 12, poamy )
4723                # Now integrated in "poapups"
4724                - poaps: if( new_pensioner, 0.0, poaps )
4725                # Now integrated in "poapups"
```

```
4726                - poapu: if( new_pensioner, 0.0, poapu )
4727                # END-OF-YEAR ALLOWANCE
4728                - poawr: if( new_pensioner, 0.0, poawr )
4729                # END-OF-YEAR ALLOWANCE
4730                - poaxp: if( new_pensioner, 0.0, poaxp )
4731
4732                #  ... And COMPOSITIONS
4733                #     (inspired from "POAEF" in "06b_Income_do" for Y9 [2015-2018]
4734                - poa: if( new_pensioner,
4735                           if( dag < 65,
4736                               ( poapups + poacm + poamr + poawr ) + poacc + poaxp,
4737                               ( poaps + poapu + poacm + psups + psupu + pdi00 + poamr + bca01
4738                                 + bca02 + bdisv + poawr + bacpm + poacc + poaxp )
4739                             ),
4740                           poa
4741                         )
4742                # Also inspired from "06b_Income_do" => if dag >= 65, included in "POA"
4743                - pdi: if( pdi == 0.0,
4744                           if( dag < 65,
4745                               ( pdi00 + bca01 + bca02 + bdisv + bacpm ),
4746                               ( 0.0 )
4747                             ),
4748                           pdi
4749                         )
4750
4751
4752
4753            #
4754            # ABOUT SURVIVAL PENSIONS
4755            ##########################
4756
4757
4758
4759            #************************************************
4760            # BUILDING RIGHTS TO TRANSFER (e.g. TO WIDOW) IF DEATH
4761            #************************************************
4762
4763
4764            # COMPUTING HERE THE BASIC AMOUNT TO BE "TRANSFERED" TO WIDOW IF EVER DYING EARLY NEXT PERIOD
4765            surv_pensions_to_transfer_process():
4766
4767
4768                # STEP 0 : INIT & COMPUTING RIGHT IF "FO_LU" PART OF EVALUATION
4769                #          (the latter not DONE in WP8)
4770                - pen_surv_tr_LU: "0.0"
```

```
4771                - pen_surv_tr_nolimit_LU_alone: "0.0"
4772
4773                # STEP 1
4774
4775                # 1 / 1st SUB-CASE : NEWLY PENSIONER
4776                #    Pension right "limited" through a PROPORTION of "Income-Related" part transfered, only
4777                - pen_surv_tr_nolimit_LU_alone:
4778                    if( new_pensioner,
4779                        ( poapups + pdi00 ) * PEN_SURV_PRORATA_OLD_AVG_PERC / 100,
4780                      pen_surv_tr_nolimit_LU_alone
4781                    )
4782
4783                # 1 / 2nd SUB-CASE : DISABLED or PENSIONER but 0-value registered for "pen_surv_tr_nolimit_LU_alone"
4784                #                    (e.g. due to those retired BEFORE first period of simulation)
4785                - pen_surv_tr_nolimit_LU_alone:
4786                    if( ( PENSIONER or DISABLED ) and ( pen_surv_tr_nolimit_LU_alone <= 0 ),
4787                          ( poapups + pdi00 ) * PEN_SURV_PRORATA_OLD_AVG_PERC / 100,
4788                            pen_surv_tr_nolimit_LU_alone
4789                      )
4790
4791                # 1 / 3rd SUB-CASE : Still ALIVE and NEITHER PENSIONER NOR DISABLED yet
4792                #                    (e.g. due to those retired BEFORE first period of simulation)                K
4793                - pen_surv_tr_nolimit_LU_alone:
4794                    if( not( PENSIONER or DISABLED ) and ( pen_surv_tr_nolimit_LU_alone <= 0 ),
4795                        if( CIVIL_SERVANT_STATUS,
4796                            PENS_OLD_AGE_PUBLIC_AVG_RATIO,
4797                            PENS_OLD_AGE_PRIVATE_AVG_RATIO
4798                          )
4799                          *
4800                        if( FEMALE,
4801                            if( not( is_age_60 ),
4802                                TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 0, deh_ever],
4803                                TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[0, 1, deh_ever]
4804                              ),
4805                            if( not( is_age_60 ),
4806                                TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 0, deh_ever],
4807                                TAB_PENSIONS_OLD_AGE_BASE_2017_NOT_UPRATED[1, 1, deh_ever]
4808                              )
4809                          )
4810                          *
4811                          PEN_SURV_PRORATA_OLD_AVG_PERC / 100,
4812                        pen_surv_tr_nolimit_LU_alone
4813                      )
4814
4815
```

```
4816                # STEP 2 : "MIN-MAX"
4817
4818                # MIN
4819                - pen_surv_tr_LU:
4820                    if( ( pen_surv_tr_nolimit_LU_alone > 0 ) and ( pen_dur_tot >= PEN_DUR_MIN_BENEFIT ),
4821                          max( pen_surv_tr_nolimit_LU_alone,
4822                               0.9 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12 )
4823                               * pen_dur_coef_max
4824                             ),
4825                          pen_surv_tr_nolimit_LU_alone
4826                        )
4827
4828                # MAX
4829                - pen_surv_tr_LU:
4830                    if( pen_surv_tr_LU > 0,
4831                        if( CIVIL_SERVANT_STATUS_1,
4832                            pen_surv_tr_LU,
4833                            min( pen_surv_tr_LU,
4834                                 5/6 * 5 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12 )
4835                               )
4836                          ),
4837                          pen_surv_tr_LU
4838                        )
4839
4840                # COPY
4841                - pen_surv_tr_LU:
4842                    if( pen_surv_tr_LU > 0,
4843                        pen_surv_tr_LU,
4844                        pen_surv_tr_LU_1
4845                        )
4846
4847
4848            #*****************************************
4849            # COMPUTING SURVIVAL BENEFIT, INCLUDING UPRATE
4850            #*****************************************
4851
4852
4853            surv_pensions_benefits_process():
4854
4855                # STEP 1 : UNCONSTRAINED AMOUNT, either "new" right or "former"
4856                - pen_surv_base_LU: if( to_new_pen_surv,
4857                                        ps_pen_surv_tr_LU,
4858                                        if( WIDOW,
4859                                            pen_surv_base_LU_1,
4860                                            0.0 )
```

```
                                        )
                # STEP 2 : COMPUTING "PROFESSIONAL_BASED" INCOME
                - surv_professional_based_income:
                    max( (yem + yse) + (bunss + byr) - 2/3 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12
),
                           0.0
                        )

                - surv_personal_reference_income: surv_professional_based_income + poapups + pdi00

                # STEP 3 : COMPUTING THE BENEFIT IN ALL CASES (including uprating of "former" survival pensions
                - psupups: if( pen_surv_base_LU + surv_personal_reference_income
                               > ( 1.5 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] ) / 12 ) ) ,
                                   max( pen_surv_base_LU -
                                            0.3 * max( surv_professional_based_income -
                                                       max( 1.5 * ( lag( PEN_REFER_AMOUNT_YEAR_PROD_N, (year - 2017)[0] )
/ 12 ) - pen_surv_base_LU,
                                                            0.0
                                                       ),
                                                  0.0
                                              ),
                                        0.0
                                    ),
                                  pen_surv_base_LU
                                )



            ####################################
            # BENEFITS MODULE - PART C : OTHERS #
            ####################################



            ##############################
            # SOCIAL CONTRIBUTION MODULE #
            ##############################



            #################
            # TAX ON INCOME #
```

```
4906            ##################
4907
4908
4909
4910
4911            ##############
4912            # INDICATORS #
4913            ##############
4914
4915
4916
4917
4918            ##############################
4919            # OUTPUTTING INFO FOR CHECKS #
4920            ##############################
4921
4922
4923
4924            #
4925            # EXCEL (.CSV) FILES
4926            ######################
4927
4928
4929            output_person_final_csv():
4930
4931
4932                # FIRST, GENERAL "SYNTHESIS" / AGGREGATE (1 line per period)
4933                # ----------------------------------------------------------
4934
4935                - csv( period,
4936                       '',
4937                       '----',
4938                       '',
4939                       count(),
4940                       count( FEMALE ),
4941                       count( MALE ),
4942                       '',
4943                       '',
4944                       '',
4945                       count( has_a_clone_expand_upstream ),
4946                       '',
4947                       count( has_a_clone_immigration_upstream ),
4948                       '',
4949                       count( to_give_birth ),
4950                       '',
```

```
                    avg( dag00, filter = FEMALE ),
                    avg( dag00, filter = MALE ),
                    '',
                    count( dag==0 and FEMALE ),
                    count( dag==0 and MALE ),
                    '',
                    avg( is_age_60, filter = FEMALE ) * 100,
                    avg( is_age_60, filter = MALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2 == 0) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2 == 15) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2 == 20) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2 == 65) and FEMALE ) / count( FEMALE ) * 100,
                     '',
                    count( (class_age_euromod_liam2 == 0) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 15) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 20) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2 == 65) and MALE ) / count( MALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2_fine == 0) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 15) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 25) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 50) and FEMALE ) / count( FEMALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 65) and FEMALE ) / count( FEMALE ) * 100,
                    '',
                    count( (class_age_euromod_liam2_fine == 0) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 15) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 25) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 50) and MALE ) / count( MALE ) * 100,
                    count( (class_age_euromod_liam2_fine == 65) and MALE ) / count( MALE ) * 100,
                    '',
                    avg( dec == 0 ) * 100,
                    avg( dec == 1 ) * 100,
                    avg( dec == 2 ) * 100,
                    avg( dec == 3 ) * 100,
                    avg( dec == 4 ) * 100,
                    avg( dec == 5 ) * 100,
                    avg( dec == 6 ) * 100,
                    '',
                    avg( deh == 0 ) * 100,
                    avg( deh == 1 ) * 100,
                    avg( deh == 2 ) * 100,
                    avg( deh == 3 ) * 100,
                    avg( deh == 4 ) * 100,
                    avg( deh == 5 ) * 100,
```

```
                    '',
                    avg( deh_ever == 0 ) * 100,
                    avg( deh_ever == 1 ) * 100,
                    avg( deh_ever == 2 ) * 100,
                    avg( deh_ever == 3 ) * 100,
                    avg( deh_ever == 4 ) * 100,
                    avg( deh_ever == 5 ) * 100,
                    '',
                    avg( deh_try_l1 ) * 100,
                    avg( deh_try_l2 ) * 100,
                    avg( deh_try_l3 ) * 100,
                    avg( deh_try_l5 ) * 100,
                    '',
                    avg( dey, filter = YOUNG_GRADUATED and FEMALE),
                    avg( dey, filter = YOUNG_GRADUATED and MALE),
                    '',
                    avg( dms == 0 , filter = FEMALE ) * 100,
                    avg( dms == 1 , filter = FEMALE ) * 100,
                    avg( dms == 2 , filter = FEMALE ) * 100,
                    avg( dms == 3 , filter = FEMALE ) * 100,
                    avg( dms == 4 , filter = FEMALE ) * 100,
                    avg( dms == 5 , filter = FEMALE ) * 100,
                    '',
                    avg( dms == 0 , filter = MALE ) * 100,
                    avg( dms == 1 , filter = MALE ) * 100,
                    avg( dms == 2 , filter = MALE ) * 100,
                    avg( dms == 3 , filter = MALE ) * 100,
                    avg( dms == 4 , filter = MALE ) * 100,
                    avg( dms == 5 , filter = MALE ) * 100,
                    '',
                    '',
                    '',
                    avg( lcs == 0 ) * 100,
                    avg( lcs == 1 ) * 100,
                    '',
                    avg( lcs == 0 , filter = EMPLOYEE and FEMALE ) * 100,
                    avg( lcs == 1 , filter = EMPLOYEE and FEMALE ) * 100,
                    '',
                    avg( lcs == 0 , filter = EMPLOYEE and MALE ) * 100,
                    avg( lcs == 1 , filter = EMPLOYEE and MALE ) * 100,
                    '',
                    avg( les == 0 , filter = FEMALE ) * 100,
                    avg( les == 1 , filter = FEMALE  ) * 100,
                    avg( les == 2 , filter = FEMALE  ) * 100,
                    avg( les == 3 , filter = FEMALE  ) * 100,
```

```
                avg( les == 4 , filter = FEMALE  ) * 100,
                avg( les == 5 , filter = FEMALE  ) * 100,
                avg( les == 6 , filter = FEMALE  ) * 100,
                avg( les == 7 , filter = FEMALE  ) * 100,
                avg( les == 8 , filter = FEMALE  ) * 100,
                avg( les == 9 , filter = FEMALE  ) * 100,
                avg( les == 10 , filter = FEMALE  ) * 100,
                '',
                avg( les == 0 , filter = MALE ) * 100,
                avg( les == 1 , filter = MALE  ) * 100,
                avg( les == 2 , filter = MALE  ) * 100,
                avg( les == 3 , filter = MALE  ) * 100,
                avg( les == 4 , filter = MALE  ) * 100,
                avg( les == 5 , filter = MALE  ) * 100,
                avg( les == 6 , filter = MALE  ) * 100,
                avg( les == 7 , filter = MALE  ) * 100,
                avg( les == 8 , filter = MALE  ) * 100,
                avg( les == 9 , filter = MALE  ) * 100,
                avg( les == 10 , filter = MALE  ) * 100,
                '',
                avg( STUDENT , filter = FEMALE  ) * 100,
                avg( STUDENT , filter = MALE  ) * 100,
                '',
                avg( WORKING , filter = FEMALE  ) * 100,
                avg( WORKING , filter = MALE  ) * 100,
                '',
                avg( EMPLOYEE , filter = FEMALE  ) * 100,
                avg( EMPLOYEE , filter = MALE  ) * 100,
                '',
                avg( CIVIL_SERVANT_WORKING , filter = FEMALE  ) * 100,
                avg( CIVIL_SERVANT_WORKING , filter = MALE  ) * 100,
                '',
                avg( SELF_EMPLOYED , filter = FEMALE  ) * 100,
                avg( SELF_EMPLOYED , filter = MALE  ) * 100,
                '',
                avg( UNEMPLOYED , filter = FEMALE  ) * 100,
                avg( UNEMPLOYED , filter = MALE  ) * 100,
                '',
                avg( PENSIONER , filter = FEMALE  ) * 100,
                avg( PENSIONER , filter = MALE  ) * 100,
                '',
                avg( PENSIONER_STATUTORY , filter = FEMALE  ) * 100,
                avg( PENSIONER_STATUTORY , filter = MALE  ) * 100,
                '',
                avg( DISABLED , filter = FEMALE  ) * 100,
```

```
                    avg( DISABLED , filter = MALE  ) * 100,
                    '',
                    avg( to_new_pen_surv, filter = FEMALE) * 100,
                    avg( to_new_pen_surv, filter = MALE) * 100,
                    '',
                    '',
                    '',
                    avg( yem > 0 ) * 100,
                    '',
                    avg( yem > 0 , filter = FEMALE ) * 100,
                    avg( yem > 0 , filter = MALE ) * 100,
                    '',
                    avg( yem, filter = (yem > 0) ),
                    '',
                    avg( yem, filter = (yem > 0) and FEMALE ),
                    avg( yem, filter = (yem > 0) and MALE ),
                    '',
                    avg( yem, filter = (yem > 0) and new_worker and EMPLOYEE and FEMALE ),
                    avg( yem, filter = (yem > 0) and new_worker and EMPLOYEE and MALE ),
                    '',
                    avg( yem, filter = (yem > 0) and not(new_worker) and EMPLOYEE and FEMALE ),
                    avg( yem, filter = (yem > 0) and not(new_worker) and EMPLOYEE and MALE ),
                    '',
                    avg( yse > 0 ) * 100,
                    '',
                    avg( yse > 0 , filter = FEMALE ) * 100,
                    avg( yse > 0 , filter = MALE ) * 100,
                    '',
                    avg( yse, filter = (yse > 0) ),
                    '',
                    avg( yse, filter = (yse > 0) and FEMALE ),
                    avg( yse, filter = (yse > 0) and MALE ),
                    '',
                    avg( yse, filter = (yse > 0) and new_worker and SELF_EMPLOYED and FEMALE ),
                    avg( yse, filter = (yse > 0) and new_worker and SELF_EMPLOYED and MALE ),
                    '',
                    avg( yse, filter = (yse > 0) and not(new_worker) and SELF_EMPLOYED and FEMALE ),
                    avg( yse, filter = (yse > 0) and not(new_worker) and SELF_EMPLOYED and MALE ),
                    '',
                    avg( poapups > 0 ) * 100,
                    '',
                    avg( poapups > 0, filter = FEMALE ) * 100,
                    avg( poapups > 0, filter = MALE ) * 100,
                    '',
                    avg( poapups, filter = poapups > 0 ),
```

```
                ',
                avg( poapups, filter = FEMALE and poapups > 0 ),
                avg( poapups, filter = MALE and poapups > 0 ),
                ',
                avg( poa, filter = FEMALE and poa > 0 ),
                avg( poa, filter = MALE and poa > 0 ),
                ',
                avg( to_new_pen_surv, filter = FEMALE ) * 100,
                avg( to_new_pen_surv, filter = MALE ) * 100,
                ',
                avg( psupups > 0 ) * 100,
                ',
                avg( psupups > 0, filter = FEMALE ) * 100,
                avg( psupups > 0, filter = MALE ) * 100,
                ',
                avg( psupups, filter = psupups > 0 ),
                ',
                avg( psupups, filter = FEMALE and psupups > 0 ),
                avg( psupups, filter = MALE and psupups > 0 ),
                ',
                avg( pdi00 > 0, filter = FEMALE ) * 100,
                avg( pdi00 > 0, filter = MALE ) * 100,
                ',
                avg( pdi00, filter = FEMALE and pdi00 > 0 ),
                avg( pdi00, filter = MALE and pdi00 > 0 ),
                ',
                avg( pdi > 0, filter = FEMALE ) * 100,
                avg( pdi > 0, filter = MALE ) * 100,
                ',
                avg( pdi, filter = FEMALE and pdi > 0 ),
                avg( pdi, filter = MALE and pdi > 0 ),
                ',
                fname='OUTPUT_PERSONS_AGGREGATE.csv',
                mode='a' )

        # SECOND, FULL POPULATION / EXHAUSTIVE (1 file per period) - VALUES
        # -----------------------------------------------------------------

        - csv( HEADERS_A_OUTPUT_ALL_PERSONS_FOR_EUROMOD,
               suffix="OUTPUT_ALL_PERSONS_FOR_EUROMOD_0_IDMAX_COMMA_SEP")

        - csv( dump( id, h_id, period, idhh, idperson, idpartner, idfather, idmother, idorighh, idorigperson, aca,
                     aco, afc, amrrm, amrtn, amrtn00, ate, bca01, bca02, bched01, bched04, bched05, bed, bfa,
                     bfapl, bfauc, bhl, bho, bmaba, bmals, bmawk, boni, bsa, bsacm, bsaht, bsaot, bun, bunmy,
```

```
                                bunss, bunssmy, byr, dag, dag00, dct, dcz, ddi, ddt, dec, if(decrg,1,0), deh, dew, dey,
                                if(dgn,1,0), dmb, dms, if(dmsyy03,1,0), if(dmsyy04,1,0), dncsy, if(drgmd,1,0), drgn1,
                                if(drgru,1,0), if(drgur,1,0), drsyy, dsu00, dsu01, dsu02, dwt, e20ps_o, e20pslw_o,
                                if(e20psmd_o,1,0), if(e20pspo_o,1,0), kfb, kfbcc, kfbmy, kivho, lcl, if(lcs,1,0),
                                if(lcs01,1,0), if(lcs02,1,0), les, lfs, lhw, lhw_f, lindi, liwftmy, liwmy, liwmy_f,
                                liwptmy, liwwh, liwwh00, liwwh_f, loc, if(lowas,1,0), lpemy, lse, ltr, lunmy, lunmy_f,
                                pdi, pdi00, pdimy, poa, poacc, poacm, poamy, poapups, poaxp, psu, psumy, psupups, tad,
                                tis, tpr, tscer, xhc, xhcmomi, xhcot, xhcrt, xmp, xpp, yds, ydses_o, yem, yemmy, yempv,
                                yivwg, yiy, ymwdt, yot, ypp, ypr, ypt, yptmp, yse, ysemy, ysv, if(active_in_input,1,0),
                                if(alive,1,0), dag2, deh_ever, if(deh_try,1,0), if(disabled,1,0), dms_1, if(employee,1,0),
                                family_worker, if(farmer,1,0), if(has_a_clone_expand_upstream,1,0),
                                if(has_a_clone_immigration_upstream,1,0), if(inactive,1,0),
                                if(is_new_clone_expand_downstream,1,0), if(is_new_clone_immigration_downstream,1,0), les_1,
                                if(in_pre_school,1,0), if(is_age_60,1,0), if(is_child_011,1,0), months_in_statuses, nch_011,
                                if(new_pensioner,1,0), if(new_unemployed,1,0), if(new_worker,1,0),
                                original_clone_expand_pp_id, original_clone_immigration_pp_id, if(other_status,1,0),
                                if(pensioner,1,0), if(pensioner_statutory,1,0), pp_clone_expand_child_id,
                                pp_clone_immigration_child_id, if(pp_immigrant,1,0), pp_nb_copies_remaining,
                                if(pp_to_be_cloned_expand,1,0), present_birth_in_model_child_id, if(to_give_birth,1,0),
                                if(self_employed,1,0), if(student,1,0), if(student_1,1,0), if(unemployed,1,0),
                                if(unemployed_1,1,0), if(unemployed_in_input,1,0), unemployment_score,
                                if(working_in_input,1,0), if(working,1,0), if(working_1,1,0), year, year_of_expansion,
                                year_of_immigration, pen_dur_coef_max, pen_dur_compl_FO, pen_dur_compl_FO_1,
pen_dur_compl_LU,
                                pen_dur_compl_LU_1, pen_dur_compl_school_FO, pen_dur_compl_school_FO_1,
                                pen_dur_compl_school_LU, pen_dur_compl_school_LU_1, pen_dur_eff_FO, pen_dur_eff_FO_1,
                                pen_dur_eff_LU, pen_dur_eff_LU_1, pen_dur_eff_inwork_FO, pen_dur_eff_inwork_FO_1,
                                pen_dur_eff_inwork_LU, pen_dur_eff_inwork_LU_1, pen_dur_eff_repl_FO, pen_dur_eff_repl_FO_1,
                                pen_dur_eff_repl_LU, pen_dur_eff_repl_LU_1, pen_dur_tot, pen_dur_tot_1, pen_dur_tot_FO,
                                pen_dur_tot_FO_1, pen_dur_tot_LU, pen_dur_tot_LU_1, pen_surv_base_LU, pen_surv_base_LU_1,
                                pen_surv_tr_LU, pen_surv_tr_LU_1, pen_surv_tr_nolimit_LU_alone,
                                surv_personal_reference_income, surv_professional_based_income, if(to_new_pen_surv,1,0),
                                ps_pen_surv_tr_LU
                                , filter = ( id >= 0) and ( id <= (MAX_NUMBER_OF_RECORDS-1) )
                                , header = False
                            ),
                      suffix='OUTPUT_ALL_PERSONS_FOR_EUROMOD_0_IDMAX_COMMA_SEP', mode='a'
                    )
```

```
simulation:

    #
    # INITIALIZATION
    # (Applying to data BEFORE simulation year,
    #  hence 2015 if "start_period" is 2016)
    ##########################################

    init:

        - household: [ init_block_hh_generalities,
                       init_block_household_output_ante_expand,

                       ############
                       # EXPAND_HH
                       # =========

                       init_block_household_clone_expand,

                       init_block_household_output_post_expand
                     ]

        - person: [

            init_block_person_general_variables_ante_expand,
            init_block_person_pension_variables_ante_expand,
            init_block_person_output_ante_expand,

            ############
            # EXPAND_PP
            # =========

            init_block_person_clone_expand,

            init_block_person_arrangements_post_expand,
            init_block_person_output_post_expand,

            ]

    processes:


           #
           # INITITAL TREATMENTS
           # (for each simulated year <> "init" block above)
```

```
5266            ################################################
5267
5268        - household: [year_setting, household_characteristics, clean_empty]
5269
5270        - person: [
5271
5272            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES
5273
5274            from_overall_les_status_to_booleans,
5275            lagged_vars_person_process,
5276            durations_consolidated,
5277            personal_revenue_variables_process,
5278
5279
5280
5281            #
5282            # DEMOGRAPHY
5283            #############
5284
5285
5286            # AGE & INCREMENTS
5287            ##################
5288
5289            # "YEAR"
5290            year_setting,
5291            # "DAG/2", "CLASS_AGE_EUROMOD_LIAM2", "DRSYY"
5292            ageing_and_incremental_procedure,
5293
5294
5295            # SURVIVAL PROCEDURE (needing "age" only)
5296            ########################################
5297
5298            survival_procedure,
5299            death_procedure,
5300
5301
5302            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES
5303
5304            from_overall_les_status_to_booleans,
5305            durations_consolidated,
5306            personal_revenue_variables_process,
5307
5308
5309            # BIRTH & CHILDREN PROCEDURE
5310            ############################
```

```
            # GIVING BIRTH & CHILHOOD

            birth_process,
            after_birth_process,
            # "IS_CHILD_011", "NCH_011"
            childhood,

            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES

            from_overall_les_status_to_booleans,
            lagged_vars_person_process,
            durations_consolidated,
            personal_revenue_variables_process,


            # EDUCATION PROCEDURE
            ######################

            education_process,

            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES

            from_overall_les_status_to_booleans,
            lagged_vars_person_process,
            durations_consolidated,
            personal_revenue_variables_process,

            ]

            # CLEANING & REASSESSING COMPOSITION

        - household: [household_characteristics, clean_empty]

        - household: [

            # CLEANING & REASSESSING COMPOSIITON

            household_characteristics,
            clean_empty,

            # IMMIGRATION  -  PARTIM "HOUSEHOLDS"
            ####################################
```

```
            # IMMIGRATION - HH
            # ----------------

            immigration_hh_process,

            ]

        - person: [

            # IMMIGRATION -  PARTIM "PERSONS"


            # IMMIGRATION - PP
            # ----------------

            immigration_person_process,
            after_immigration_person_process,

            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES

            from_overall_les_status_to_booleans,
            durations_consolidated,
            personal_revenue_variables_process,



            ]

        - person: [



            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES

            durations_consolidated,


            # PENSIONER OR NOT (if considered as an "absorbing" state)
            ######################################################


            retirement_process,

            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES
```

```
5401
5402            from_overall_les_status_to_booleans,
5403            durations_consolidated,
5404            personal_revenue_variables_process,
5405
5406
5407            # UNEMPLOYED OR WORKING
5408            ########################
5409
5410            unemployment_and_working_process,
5411            post_unemployment_and_working_process,
5412
5413            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES
5414
5415            from_overall_les_status_to_booleans,
5416            durations_consolidated,
5417            personal_revenue_variables_process,
5418
5419
5420            # PENSIONS => OLD-AGE
5421            ######################
5422
5423            old_age_pensions_rights_process,
5424            old_age_pensions_new_benefits_process,
5425
5426            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES
5427
5428            from_overall_les_status_to_booleans,
5429            durations_consolidated,
5430            personal_revenue_variables_process,
5431
5432
5433            # PENSIONS => SURVIVAL
5434            ######################
5435
5436            surv_pensions_to_transfer_process,
5437            surv_pensions_benefits_process,
5438
5439            # OVERALL CONSISTENCY & REASSESSMENT OF STATUS & MONETARY VARIABLES
5440
5441            from_overall_les_status_to_booleans,
5442            durations_consolidated,
5443            personal_revenue_variables_process,
5444
5445            ]
```

```
        - household: [

            # RECOMPUTING MAYBE USEFUL VARIABLES (INCOME), BESORE SOCIAL ASSISTANCE
            household_characteristics,

            ]

        - person: [



            #
            # OUTPUT FOR CHECK
            ####################


            output_person_final_csv,


            ]

        - household: [

            # OUTPUT FOR CHECK

            output_household_final_csv,

            ]

    input:

        path:  "INPUT_DATA"
        file:  "EUROMOD_LU_2018_A2_ENTRY_for_WP8.h5"

    output:

        path:  "OUTPUT_DATA"
        file:  "FINAL_OUTPUT_EUROMOD_FROM_LU_2018_A2_ENTRY_for_WP8.h5"


    start_period: 2018    # first simulated period

    periods: 6            # MAX 6 (2018 [for SYSTEM 2017] up to 2023 [for SYSTEM 2024]), as done on 4 SEPT 2021
```

```
5491
5492
5493        default_entity: person
5494
5495        skip_shows: False
5496
5497        random_seed: 10000    #  optional
```