EUROMOD training course - exercises

Fiscal Policy Analysis Unit
Joint Research Centre
European Commission

Course based on
EUROMOD I4.0+, software v. 3.6.8
18-20 October 2023

The JRC is deeply indebted to colleagues from ISER - University of Essex who ran EUROMOD training until 2020. A number of exercises in this manual were originally developed by them and have been successively updated by the JRC.

Contents

Exercise 1: Running EUROMOD; Using the Statistics Presenter

The aim of this exercise is to learn how to run EUROMOD, produce micro-output and learn how to use the Statistics Presenter tool with Default and MultiSystem options. The exercise also introduces some basic options such as (un)hiding systems and expanding policies.

Steps:

1. Run EUROMOD for Austria (system: 2019);

2. Run EUROMOD for Austria, Finland, Italy and Spain together (system: 2020);

3. Use the Statistics Presenter Tool – Default option – to produce results for the income distribution in Austria for 2019 and 2020 and analyse the results.

4. Use the Statistics Presenter Tool – MultiSystem option – to produce results for the income distribution for these four countries for 2020 and analyse the results.

Description:

The **Statistics Presenter Tool** provided with EUROMOD allows computing a range of commonly used indicators and statistics for analysing EUROMOD micro-output. Amongst others, it computes basic poverty and inequality indices for the overall population and for selected groups; distribution of household income, taxes and benefits by income group; information on households by income group; and government revenue and expenditure.

The computations are based on EUROMOD standard micro-output, which comprises variables for disposable income and its components (original income, benefits, taxes and social insurance contributions).

Therefore, the first step is to produce this micro-output for the countries you would like to analyse.

Solution and further information

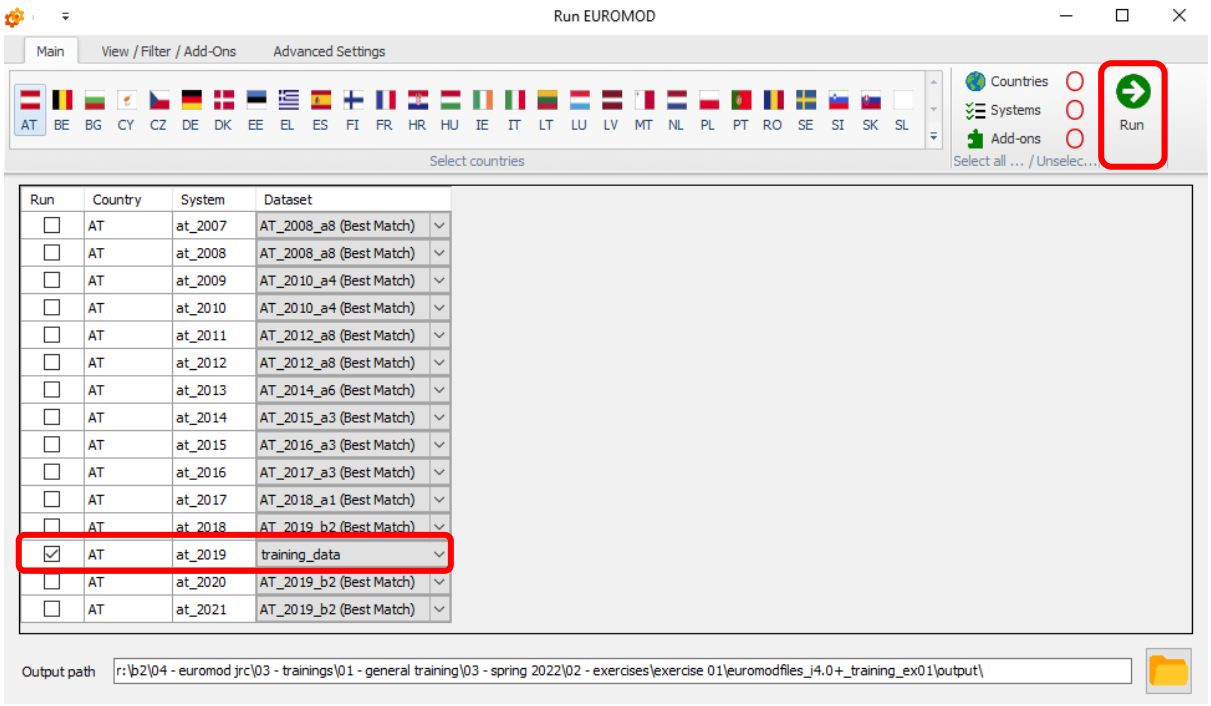
Task 1: Producing output micro-data for Austria for 2019

Run EUROMOD in order to produce micro-data output for Austria. The model can be run for one country at the time or the user can select several countries (and systems) for running all of them at once.

- In the Countries tab on the ribbon bar press the "Run EUROMOD" button. This will bring up a dialogue box.
- In the dialogue box, select Austria by pressing on the AT flag (if not already selected). This will display the available systems for Austria (AT) (see Figure 1.1).
- Select the AT_2019 system by clicking the corresponding boxes.
- For this system, select *training_data* as the dataset which EUROMOD should use (see Figure 1.2).
- Check (and if necessary, modify) the output path where EUROMOD will be writing the output files.
- Click on the Run button in the dialogue box.

Figure 1.1: EUROMOD ribbon – all countries displayed

Figure 1.2: Run EUROMOD dialogue, selecting Austria 2019 system and "training_data" for dataset



To see the Run or the Error logs (if there are any) click on the corresponding buttons.

Figure 1.3: Running AT_2019 system with training_data dataset – in progress

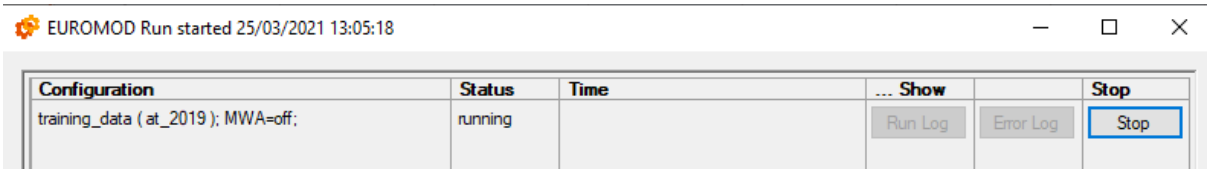
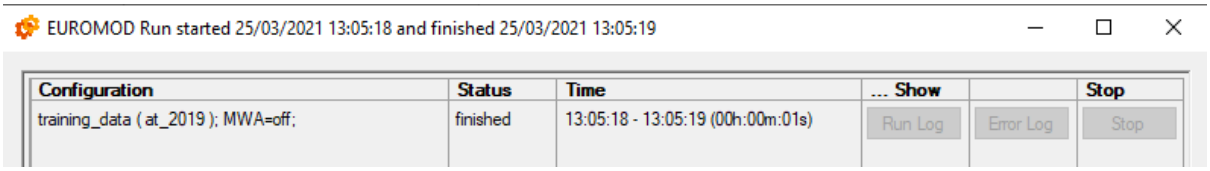


Figure 1.4: Running AT_2019 system with training_data dataset – finished with no errors



Well done! You have just run EUROMOD for the first time!

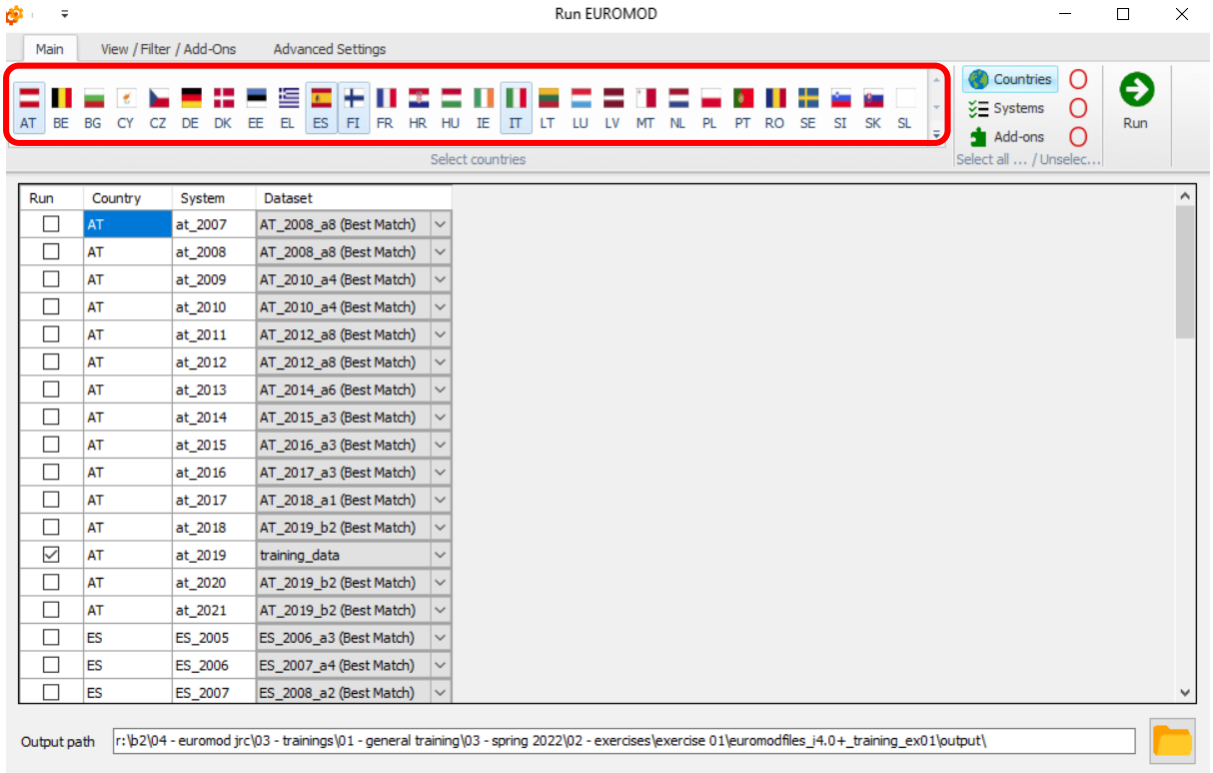
What happened is that EUROMOD simulated this country and year's policy system using 'training_data' as the input data. We will come back to this and have a look at the outputs of the run in Task 3. Let's try to do the same now for multiple countries and years.

Note: If you are stuck, a good place to look for help, is the detailed 'Help' section within the EUROMOD software. It is located in the ribbon menu under 'Help & Info' > Help.

Task 2: Running 2020 systems for Austria, Finland, Italy and Spain

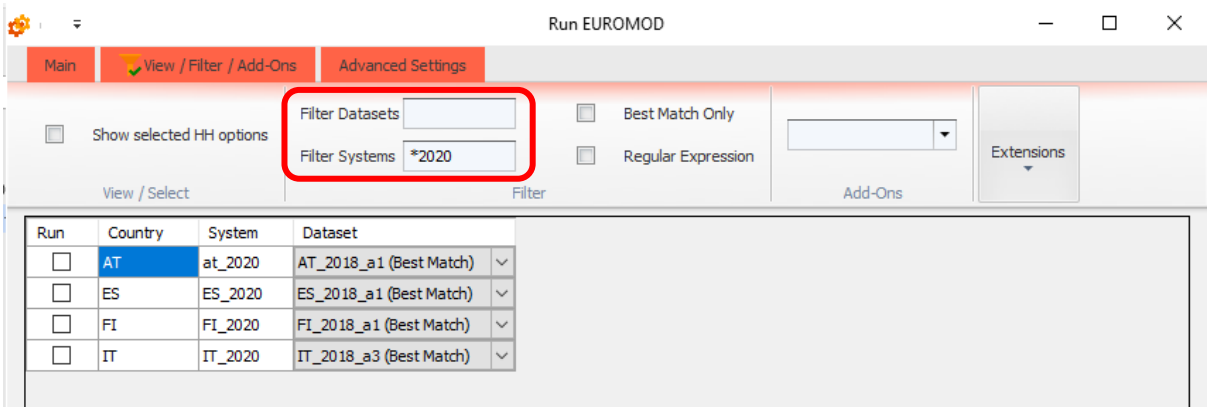
Click on the 'Run EUROMOD' button. Click on the flags for Austria (AT), Finland (FI), Italy (IT) and Spain (ES). A list of all available systems will appear (Figure 1.5).

Figure 1.5: EUROMOD run dialogue – selected countries: Austria, Finland, Italy and Spain



Click on the "View/Filter/Add-Ons" tab and using the 'Filters system' field filter the 2020 systems for these countries (Figure 1.6).

Figure 1.6: Advanced Run settings – filtering systems



Then type 'training*' in the 'Filter Datasets' field to select 'training_data' as the dataset (Figure 1.7). Make sure all boxes on the left of the systems are selected (ticked) (Figure 1.8) to ensure that they will be included in the EUROMOD run.

Figure 1.7: Advanced Run settings – filtering datasets and systems

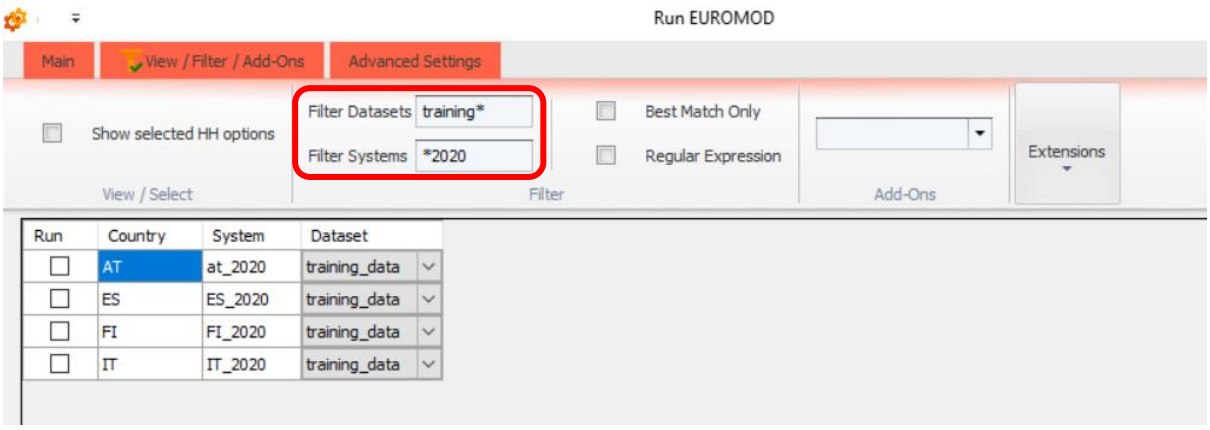
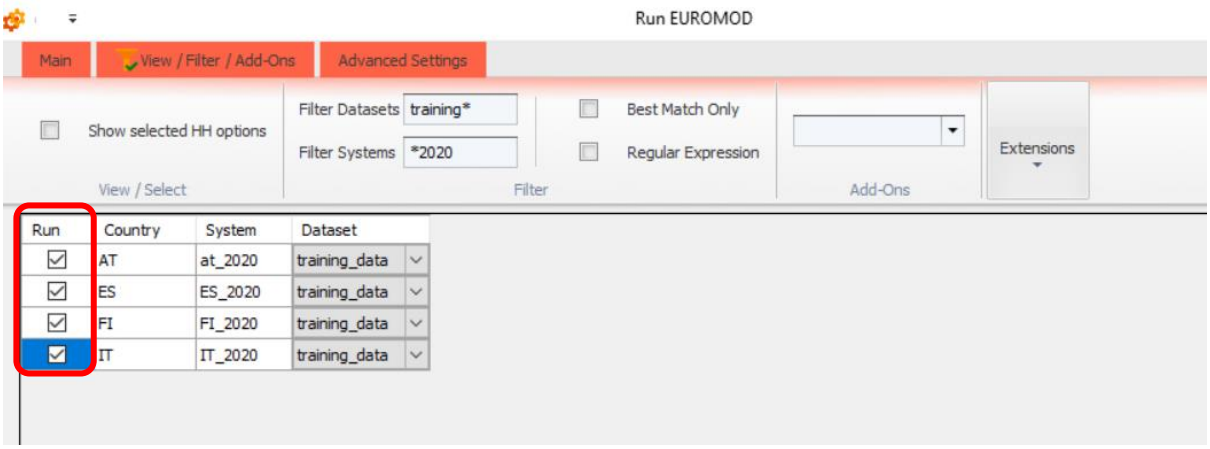


Figure 1.8: Advanced Run settings – selecting systems to run

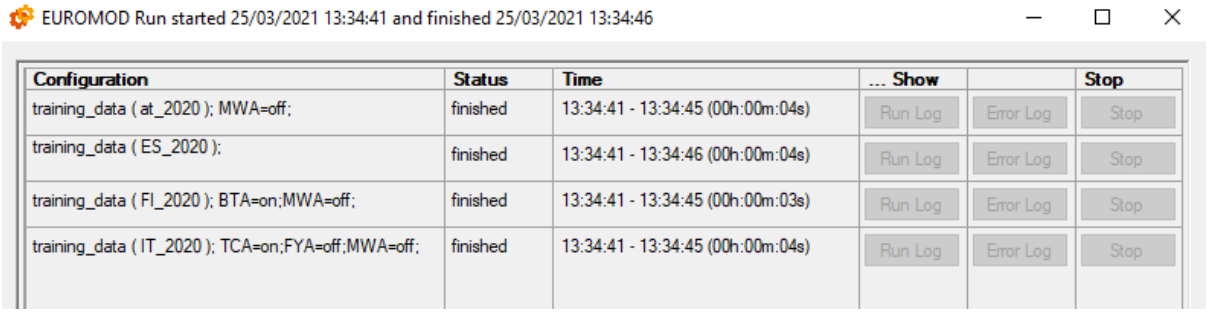


Click on the 'Main' tab to return to the main 'Run EUROMOD' window, then click on 'Run' button to run the systems (Figure 1.9).

Figure 1.9: Running the selected systems

The run window will appear as before and all systems should run without any problems (status: finished).

Figure 1.10: EUROMOD Run window – all systems and data combinations finished without errors



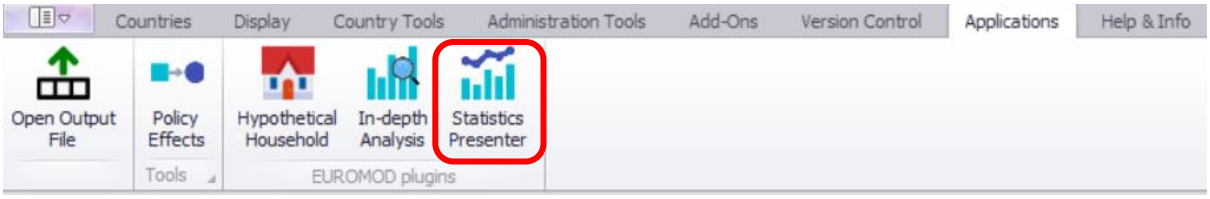
Task 3: Using the Statistics Presenter with the Default option

Let's now explore the outputs of the different runs that we set up in tasks 1 and 2. For each run, EUROMOD creates an output file, which includes the simulated values in addition to non-simulated variables. These are stored in the Output folder within your project folder.

An easy way to explore the output files is to use the Statistics Presenter ('Applications'> 'Statistics Presenter'):

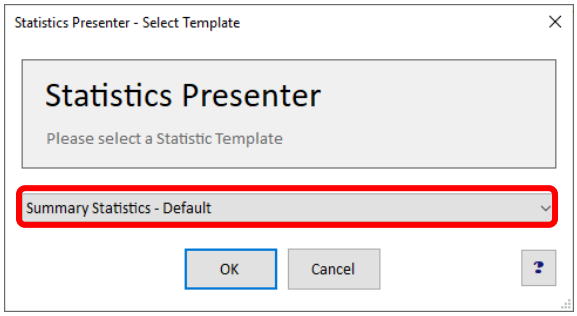
- To access the Statistics Presenter, click on the Applications tab. Click on the Statistics Presenter button to launch the tool (Figure 1.11)

Figure 1.11 Opening the Statistics Presenter



There are different types of options. Select the *Default* option and click ok (Figure 1.12)

Figure 1.12: Statistics Presenter: generating a Default report view

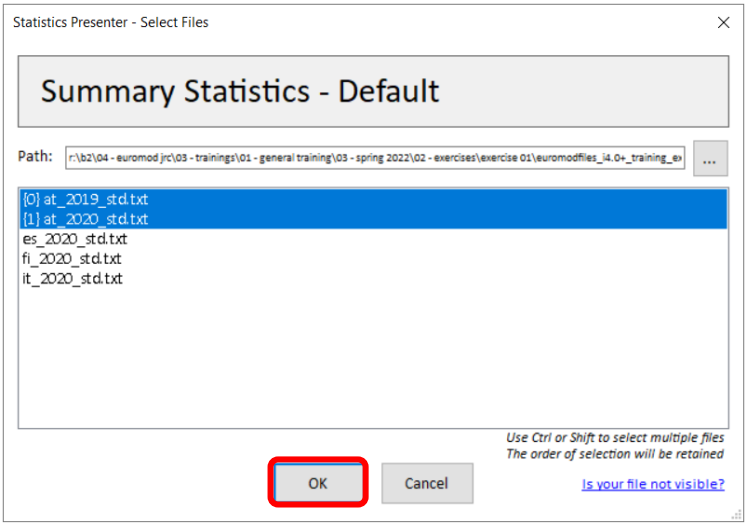


In the window that will appear (Figure 1.13), use Ctrl (select multiple individual elements) or Shift (select a range of multiple elements) to select the relevant EUROMOD output files that you would like to use and click OK. Note that the path where the output files are stored can be changed, if needed and that the order you select the files will be reflected on the order the systems will appear in the report.

Let's first compare the 2019 and 2020 Austrian systems like below:

- click on *at_2019_std.txt* first
- holding CTRL, click on *at_2020_std.txt* next

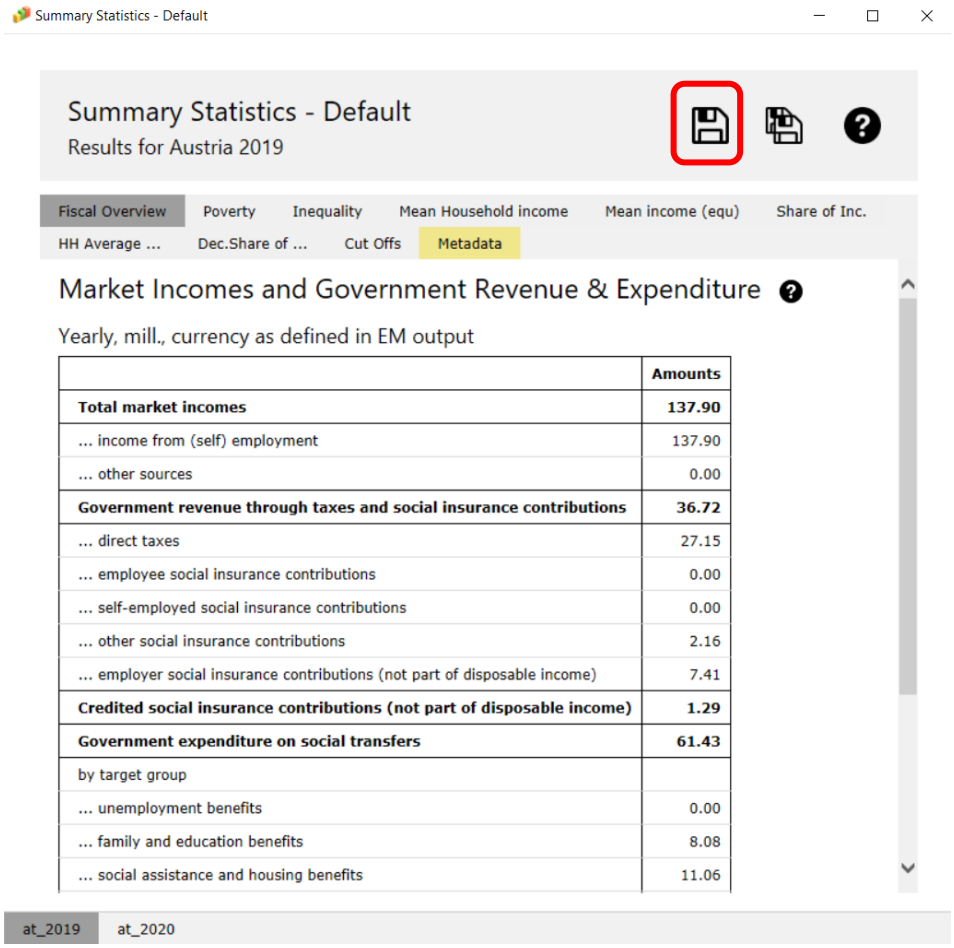
Figure 1.13: Statistics Presenter, Default report view: selecting systems to output



Click on OK.

A number of tables, as shown in Figure 1.14, will be generated, for each of the two systems. You can export the statistics in an excel file if you would like to further edit the tables, by clicking on the "disk" icon.

Figure 1.14: Statistics Presenter. Default report view; exporting to external file



Go through the different systems tabs (at_2019, at_2020) and different tables (top tab) to see the different types of statistics.

Alternatively, you can use the Statistics Presenter with the "MultiSystem" option to analyse the output micro-data. The next task explains how.

Task 4: Using the Statistics Presenter with the MultiSystem option

Open the Statistics Presenter and select the *MultiSystem* option. For this task, we will analyse the output micro-data for 2020 for Austria, Finland, Italy and Spain.

Select the output files for the four countries as in Figure 1.16. The output report should look like the one in Figure 1.17. Note that the different systems are now displayed next to each other (for easy comparison).

Figure 1.15: Statistics Presenter, Multisystem report

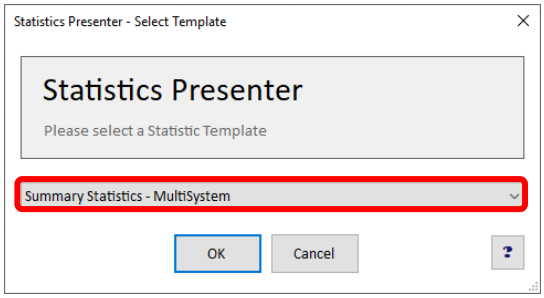


Figure 1.16: Statistics Presenter, Multisystem report: Selecting system output files

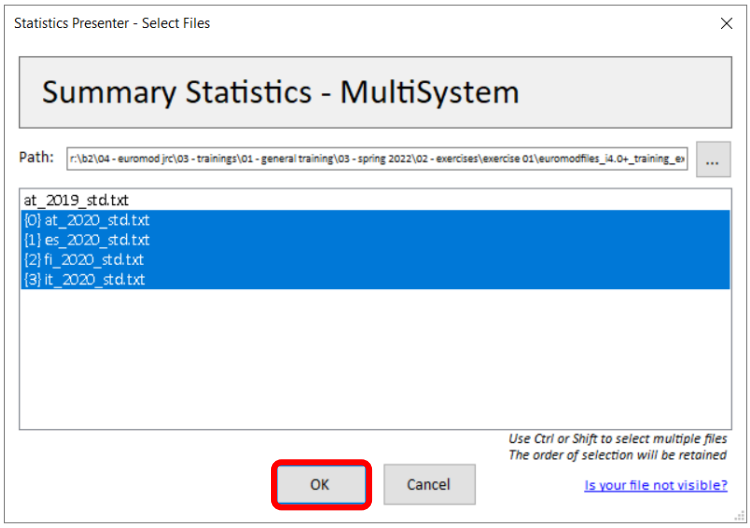
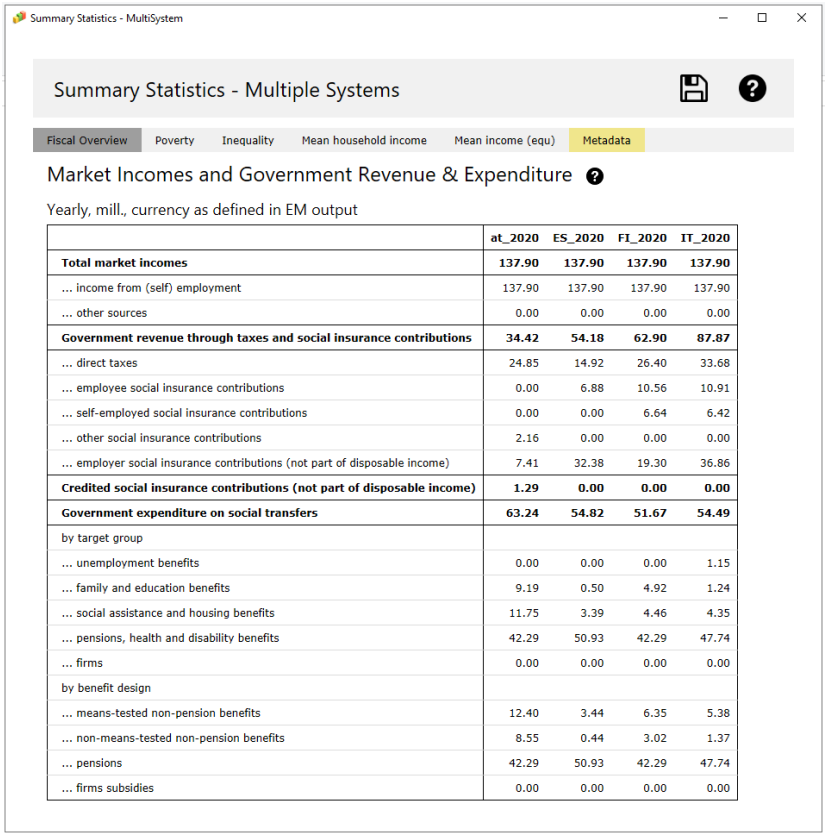
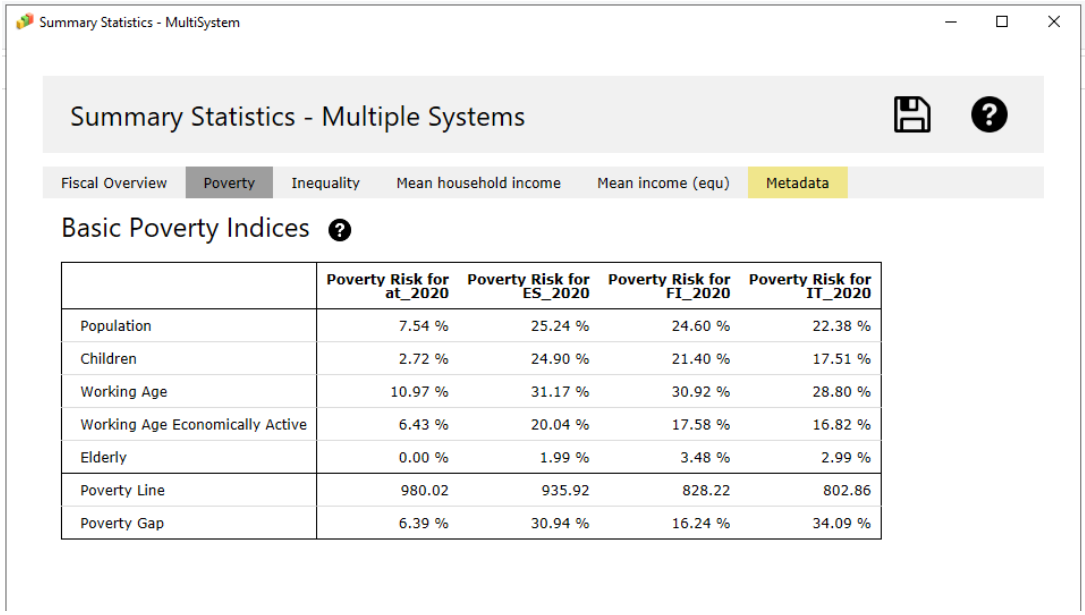


Figure 1.17: Statistics Presenter. Multiple system report view



	at_2020	ES_2020	FI_2020	IT_2020
Total market incomes	**137.90**	**137.90**	**137.90**	**137.90**
... income from (self) employment	137.90	137.90	137.90	137.90
... other sources	0.00	0.00	0.00	0.00
Government revenue through taxes and social insurance contributions	**34.42**	**54.18**	**62.90**	**87.87**
... direct taxes	24.85	14.92	26.40	33.68
... employee social insurance contributions	0.00	6.88	10.56	10.91
... self-employed social insurance contributions	0.00	0.00	6.64	6.42
... other social insurance contributions	2.16	0.00	0.00	0.00
... employer social insurance contributions (not part of disposable income)	7.41	32.38	19.30	36.86
Credited social insurance contributions (not part of disposable income)	**1.29**	**0.00**	**0.00**	**0.00**
Government expenditure on social transfers	**63.24**	**54.82**	**51.67**	**54.49**
by target group				
... unemployment benefits	0.00	0.00	0.00	1.15
... family and education benefits	9.19	0.50	4.92	1.24
... social assistance and housing benefits	11.75	3.39	4.46	4.35
... pensions, health and disability benefits	42.29	50.93	42.29	47.74
... firms	0.00	0.00	0.00	0.00
by benefit design				
... means-tested non-pension benefits	12.40	3.44	6.35	5.38
... non-means-tested non-pension benefits	8.55	0.44	3.02	1.37
... pensions	42.29	50.93	42.29	47.74
... firms subsidies	0.00	0.00	0.00	0.00



	Poverty Risk for at_2020	Poverty Risk for ES_2020	Poverty Risk for FI_2020	Poverty Risk for IT_2020
Population	7.54 %	25.24 %	24.60 %	22.38 %
Children	2.72 %	24.90 %	21.40 %	17.51 %
Working Age	10.97 %	31.17 %	30.92 %	28.80 %
Working Age Economically Active	6.43 %	20.04 %	17.58 %	16.82 %
Elderly	0.00 %	1.99 %	3.48 %	2.99 %
Poverty Line	980.02	935.92	828.22	802.86
Poverty Gap	6.39 %	30.94 %	16.24 %	34.09 %

Exercise 2: Implementing a reform to the model of Finland; Using the Statistics Presenter to analyse its impact on the income distribution

The aim of this exercise is to learn how to create (add) a new reform system and how to use the Statistics Presenter – Baseline/Reform option – to analyse the impact on the income distribution of the reform.

Steps:

1. Create a new system in Finland based on the 2021 system and name it FI_2021reform;

2. Increase the Child Benefit amount for the first child from €94.88 to €100 per month;

3. Run EUROMOD to produce micro-outputs for these two systems (2021 and 2021reform);

4. Analyse the results using the Statistics Presenter with Baseline/Reform option.

Description:

You are asked to implement a simple reform to FI_2021 policy system. You will increase the Child Benefit amount for the first child from €94.88 per month to 100€ per month.

In more detail, you will make a copy of the FI 2021 system and call it FI_2021reform. Then, you will modify the reform system FI_2021reform as follows:

- Go to the Child Benefit policy (policy name *bch_fi* in row 41) and open the functions DefConst (row 41.1) and BenCalc (row 41.3)
- Change the parameter value of $bch1 (row 41.1.1) from #94.88 (which is equal to €94.88 per month) to *100#m* (€100 per month).
- Save your changes.

When you are done with your changes to the reform system, run the model for the baseline system FI_2021 and reform system FI_2021reform. Use the *Statistics Presenter* – Baseline/Reform option – to analyse the distributional effect of the reform.

Hints:

- The variable for the simulated child benefit is *bch_s* (b: benefit; ch: child; _s: simulated)
- The suffix *#m* indicates that the benefit amount is paid on a monthly basis. By default, EUROMOD treats all values of parameters and variables as monthly amounts. The suffix #m is therefore not strictly required and could be omitted. Defining the parameter *$bch1* as *100*, instead of *100#m*, would therefore lead to exactly the same results.
- You can also assign amounts on a weekly basis (with suffix #w) or on a yearly basis (suffix #y). In such case, you need to specify the suffix.

Solution and further information

Task 1: Creating a new system

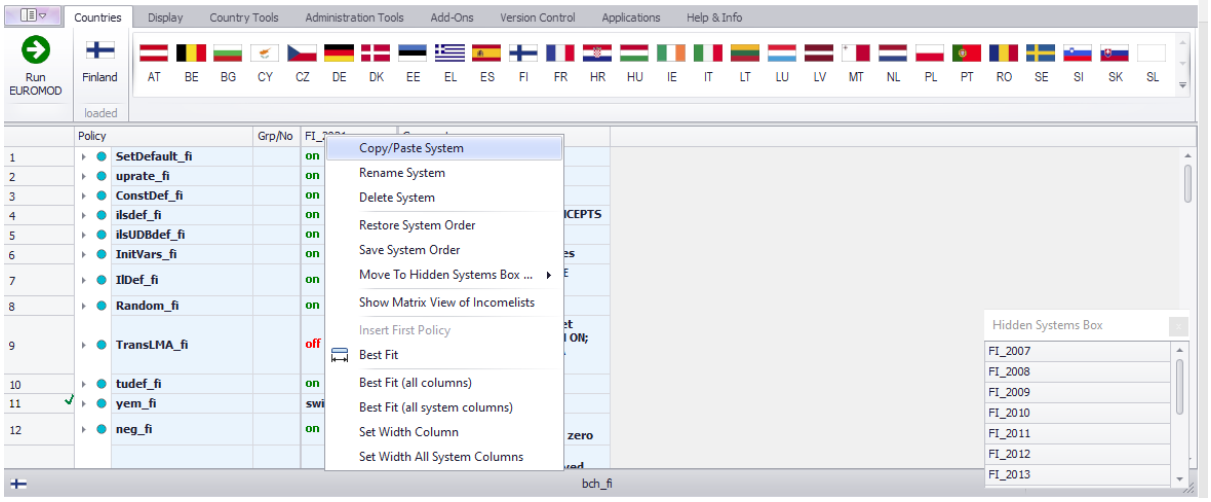
Open the FI model, by clicking on the Finnish country flag. To work more easily with the FI_2021 system, hide the rest of the systems, by right-clicking on the system name (FI_2021), selecting "*move to hidden system box*" and then selecting "*all systems but selected*" (see Figure 2.1).

Figure 2.1: Hiding a system

Add a new system. There are two ways of doing this, either:

1) right-click on the system heading FI_2021 and select the option 'Copy/Paste System' or
2) click on the button *Add System* in the Country Tools tab.

Figure 2.2: Adding a reform system using the copy/paste option



Let's use option 1). In the window that appears (Figure 2.3), specify a name for the reform system (type e.g. *FI_2021reform*). The new systems should appear on the right of the FI_2021 system (Figure 2.4). Note that columns can be expanded (by dragging the lines) if you cannot see all the contents.

Figure 2.3: Giving a name to the reform system

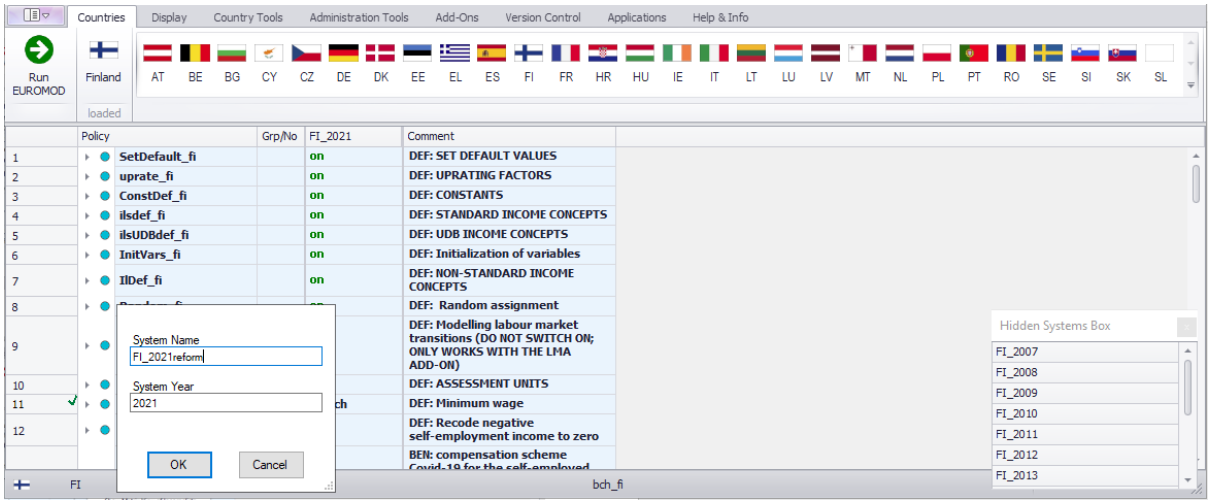
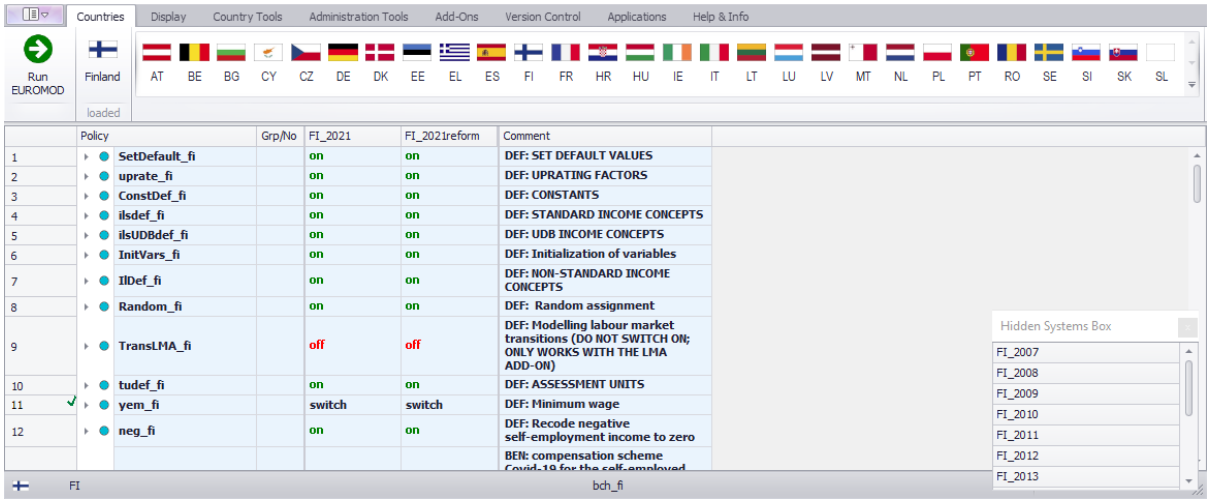


Figure 2.4: Baseline and new reform system



Task 2: Implementing the reform

You need to increase the Child Benefit amount for the first child from €94.88 per month to 100€ per month. To do that, we will need to edit the 'Child Benefit' policy on the country spine.

The Child Benefit is implemented in the policy called *bch_fi*. The amount for the first child is defined in the constant *$bch1* in the function *DefConst* (row 41.1.1). The constant *$bch1* is then used in the function BenCalc (row 41.3.17), which calculates the benefit amounts per number of children. If you hover your mouse over the *$bch1* constant in row 41.3.17 you can see its values ("$bch1 = 94.88#m" in Figure 2.5).

Figure 2.5: Hovering over the *$CBFirst* constant to see the current value

41	**bch_fi**		on	on	**BEN: Child benefit**
41.1	DefConst		on	on	Constants for Child Care Benefits
41.1.1	$bch1		94.88#m	94.88#m	Child benefit for the 1st child
41.1.2	$bch2		104.84#m	104.84#m	Child benefit for the 2nd child
41.1.3	$bch3		133.79#m	133.79#m	Child benefit for the 3rd child
41.1.4	$bch4		163.24#m	163.24#m	Child benefit for the 4th child
41.1.5	$bch5		182.69#m	182.69#m	Child benefit for the 5th child & subsequent
41.1.6	$bchsup		63.30#m	63.30#m	Child benefit supplement for a lone parent
41.2	Elig		on	on	Child benefit: eligibility
41.3	BenCalc		on	on	Child benefit: amount
41.3.1	who_must...		all	all	
41.3.2	comp_cond	1	IsNtoMchild#1=1	IsNtoMchild#1=1	
41.3.3	#_N	1	1	1	
41.3.4	#_M	1	1	1	
41.3.5	comp_cond	2	IsNtoMchild#2=1	IsNtoMchild#2=1	
41.3.6	#_N	2	2	2	
41.3.7	#_M	2	2	2	
41.3.8	comp_cond	3	IsNtoMchild#3=1	IsNtoMchild#3=1	
41.3.9	#_N	3	3	3	
41.3.10	#_M	3	3	3	
41.3.11	comp_cond	4	IsNtoMchild#4=1	IsNtoMchild#4=1	
41.3.12	#_N	4	4	4	
41.3.13	#_M	4	4	4	
41.3.14	comp_cond	5	IsNtoMchild#5=1	IsNtoMchild#5=1	
41.3.15	#_N	5	5	5	
41.3.16	#_M	5	99	99	
41.3.17	comp_perElig	1	$bch1	$bch1	Amount for the 1st child
41.3.18	comp_perElig	2	$bch2	$bch2	Amount for the 2nd child
41.3.19	comp_perElig	3	$bch3	$bch3	Amount for the 3rd child
41.3.20	comp_perElig	4	$bch4	$bch4	Amount for the 4th child
41.3.21	comp_perElig	5	$bch5	$bch5	Amount for the 5th & subsequent children
41.3.22	output_var		bch_s	bch_s	Child benefit (without lone parent supplement)
41.3.23	TAX_UNIT		tu_family17_fi	tu_family17_fi	
41.4	Elig		on	on	Eligibility for Lone parent supplement
41.5	ArithOp		on	on	Child benefit: single parent supplement

In the reform system FI_2021reform, change the parameter value of *$bch1* from 94.88#m to *100#m* (€100 per month).

Figure 2.6: New benefit amount for the first child

Policy	Grp/No	FI_2021	FI_2021reform	Comment
bch_fi		on	on	BEN: Child benefit
DefConst		on	on	Constants for Child Care Benefits
$bch1		94.88#m	100#m	Child benefit for the 1st child
$bch2		104.84#m	104.84#m	Child benefit for the 2nd child
$bch3		133.79#m	133.79#m	Child benefit for the 3rd child
$bch4		163.24#m	163.24#m	Child benefit for the 4th child
$bch5		182.69#m	182.69#m	Child benefit for the 5th child & subsequent
$bchsup		63.30#m	63.30#m	Child benefit supplement for a lone parent

Save your changes (Ctrl+S).

Step 3: Produce and analyse the micro-data output

Run the model to produce micro-data output for the baseline FI_2021 and reform FI_2021reform systems using *training_data* (see exercise 1 if you cannot remember how to do this) (Figure 2.7). Use the Statistics Presenter – option Baseline/Reform (Figure 2.8) – to produce a report (Figure 2.9) and analyse differences in government revenues and income poverty due to the changes in the Child Benefit.

Figure 2.7: Producing micro-data output for the baseline and reform systems

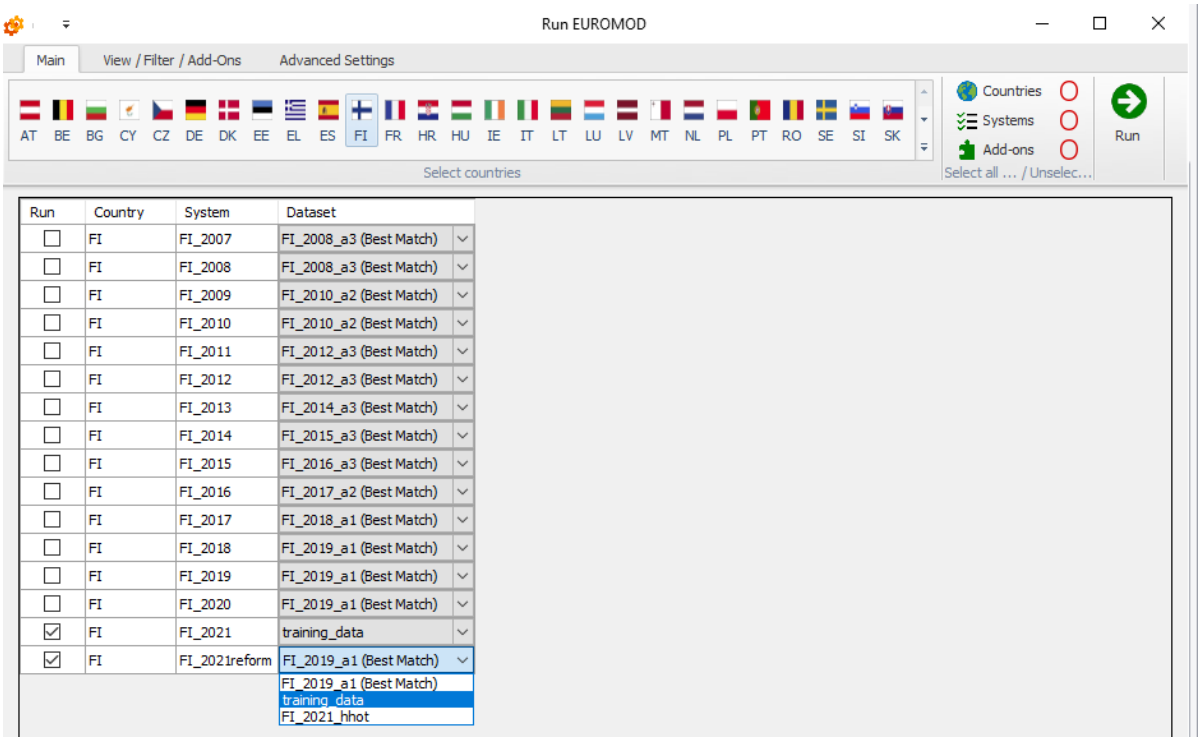


Figure 2.8: Opening Statistics Presenter – Baseline/Reform option

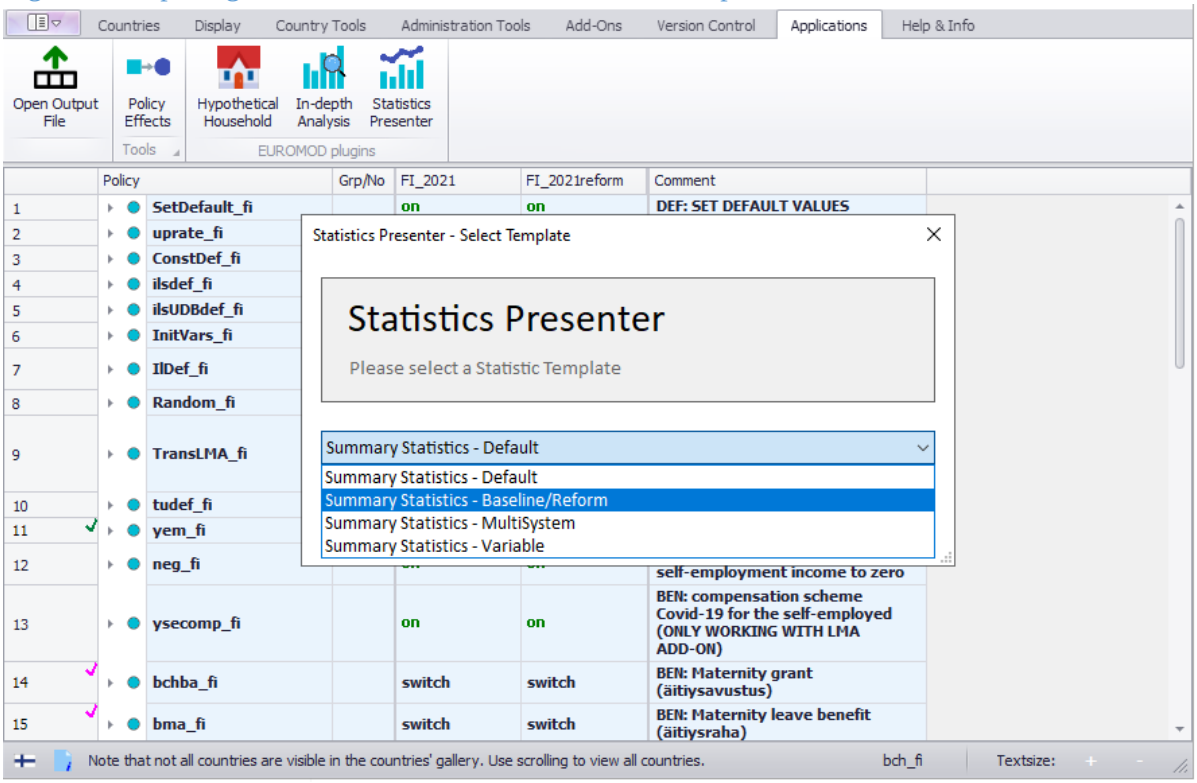
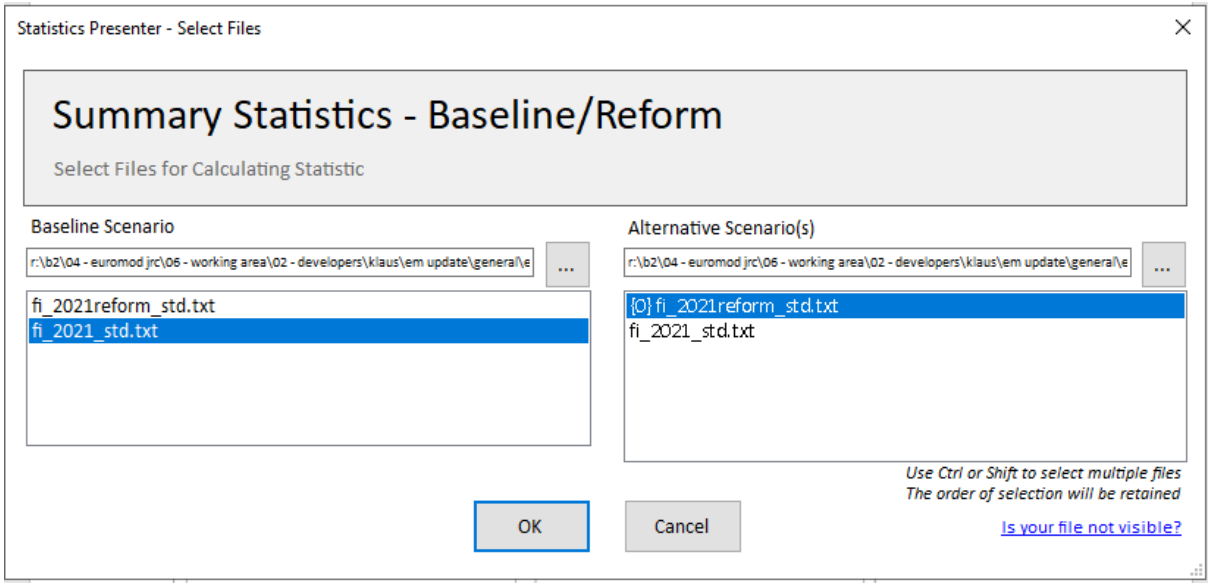


Figure 2.9: Selecting the relevant output files with the Statistics Presenter



Results

Table 2a: Government Expenditure

Yearly, mill., currency as defined in EM output

	FI_2021 (base)	*FI_2021reform*	*Difference to base*
Government expenditure on social transfers	**51.70**	**51.79**	**0.08**
... family and education benefits	4.93	5.02	0.09
... social assistance and housing benefits	4.48	4.47	-0.01

Table 2b: Mean household income by decile groups

	FI_2021 (base)	*FI_2021reform*	*Difference to base*
Decile 1	729.68	730.04	0.36
Decile 2	1,485.36	1,486.61	1.25
Decile 3	1,745.62	1,747.42	1.80
Decile 4	1,835.96	1,837.32	1.36
Decile 5	2,060.86	2,062.53	1.67
Decile 6	2,526.84	2,528.73	1.89
Decile 7	2,544.38	2,546.12	1.74
Decile 8	3,123.74	3,124.63	0.89
Decile 9	3,766.25	3,767.50	1.25
Decile 10	4,862.14	4,864.08	1.95
All	**2,437.55**	**2,438.91**	**1.37**
Poor	1,095.62	1,096.42	0.79

Exercise 3a: Reforming Child Allowance in Estonia

The aim of this exercise is to introduce functions Elig and ArithOp. It also introduces some basic options such as renaming and deleting systems, search and replace and queries.

Steps:

1. Create a new system in Estonia based on the 2012 system and name it EE_2012_E2;

2. Modify Child Allowance (CA) in way that access to the supplement should be restricted to families with at least two children and with earnings below 200 EUR per month. The supplement is equal to 1800 EUR per month;

3. Run EUROMOD to produce micro-outputs for these two systems (2012 and 2012_E2);

4. Analyse the results.

Description:

The aim of this exercise is to perform a reform of the Child Allowance (CA) benefit in Estonia in 2012. In that year, the CA was a monthly universal non-means-tested benefit paid to families with children below an age limit. The amount per child was two times the Child Allowance Rate (CAR equal to 9.59 EUR), resulting in 19.18 EUR per month for the first and second child, and 6 x CAR for the third and any consequent child.

Your task is to add a means-tested supplement to the universal CA using functions *Elig* and *ArithOp*.

Hints:

- The variable for earnings is *yem* (y: market income; em: employment).
- Use the same variable name for your calculations as the one used in the *BenCalc* function in the same policy, i.e. *bch00_s* (b: benefit, ch: child, 00: main, _s: simulated).
- Use the same tax/assessment unit for your calculations as the one defined in the *BenCalc* function in the same policy, i.e. *tu_CBfamily_ee*.
- The number of dependent children in a tax unit is calculated by the query *nDepChildrenInTu*.
- Monthly amounts need to be indicated with the suffix *#m*.

Solution and further information

Task 1: Creating a new system

When our task is to create (add) a new system, we should first think about which existing system the new system should be based on. In the case where we need to create a system for a new year, it is most likely that we will use the previous year as a starting point. If we want to change/modify an existing system year (but show it and analyse it as a separately), we will use that original system as a base. How do we do that?

Our first step is to open Estonia (country acronym EE) *(see exercise 1 for more information on how to load a country system)*. Hide all other systems except for EE_2012. If we go to the tab *Country Tools* in the ribbon bar, we can use the option *Add System* (Figure 3a.1) to add a new system based on an existing base system. Click on the 'Add System'. You will get a dialog box with base systems for the country that we are working with (i.e. currently active), in this case Estonia. As you can see, we are only allowed to choose one of them. In our case, it will be the 2012 system (Figure 3a.2).

Figure 3a.1: Country Tools menu

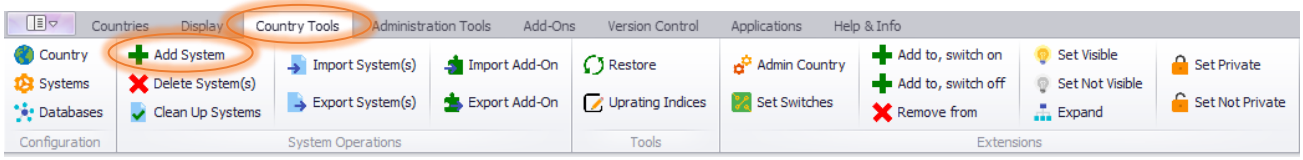
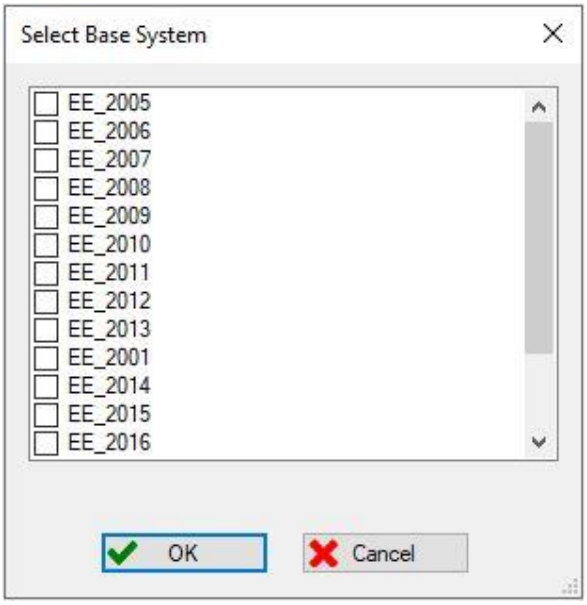
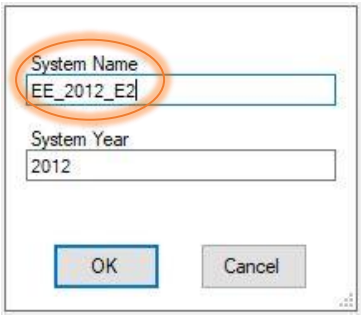


Figure 3a.2: Selecting the Base system



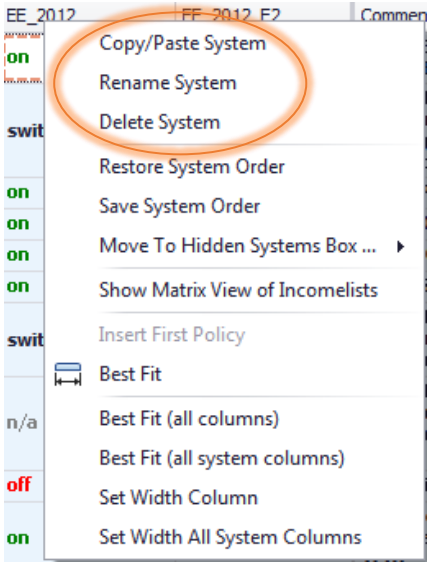
Rename the new system 'EE_2012_E2' (Figure 3a.3). Since we do not need other systems, we will hide all but these two (EE_2012 and EE_2012_E2).

Figure 3a.3: Renaming the System



You could also do the same by going to the system that we want to copy/paste/delete, right-click on it. This provides several options (Figure 3a.4). You can rename the system using the option *Rename System*.

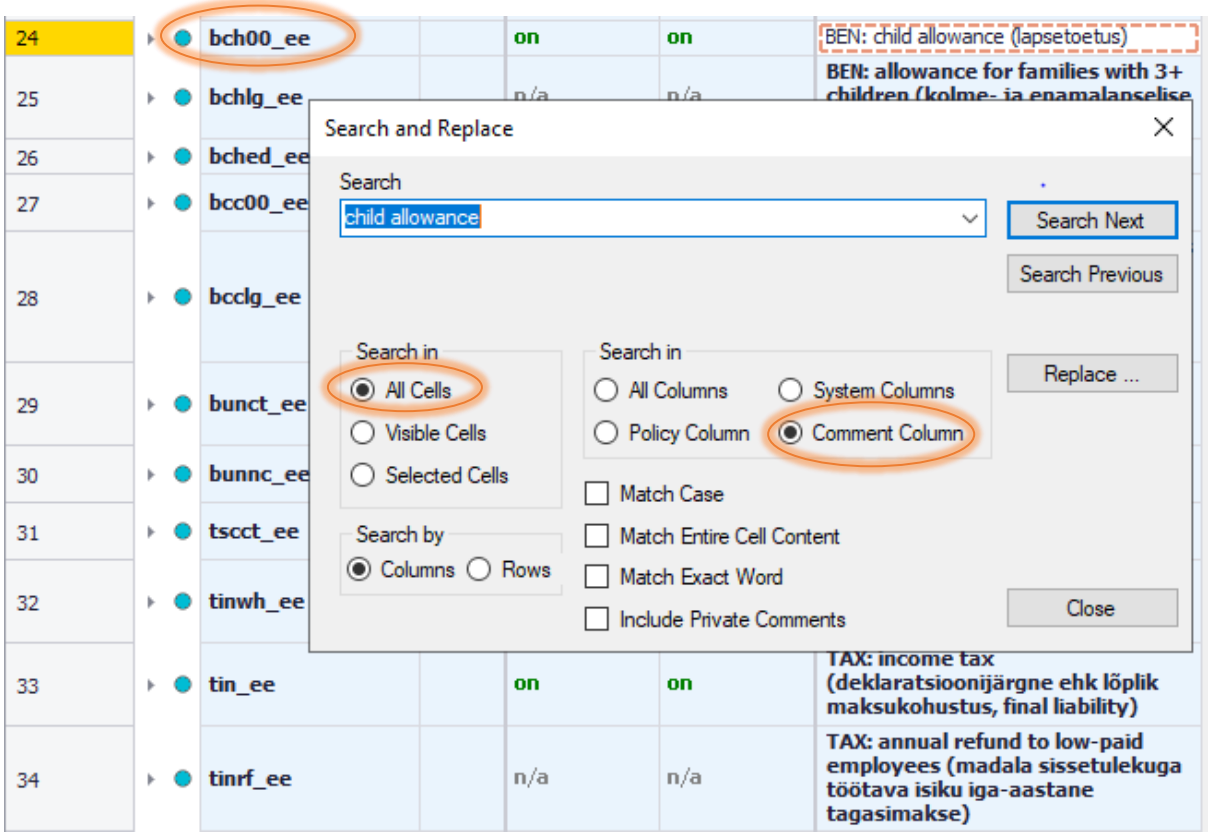
Figure 3a.4: Creating a copy of the existing system and renaming it



Task 2: Implementing the CA reform

After we have created a new system, our next task in implementing the reform is to find the Child Allowance policy in the policy spine. The easiest way to do this is to use command search and focus on the column *Comment* (Figure 3a.5). This should bring up the Child Allowance policy **bch00_ee.**

Figure 3a.5: Locating the Child Allowance policy



To see how it is calculated, we should use the expand option (click on the little triangle on the left of the policy) – Figure 3a.6. Functions with ***n/a*** (= not applicable) are not used for the calculations so we can collapse them. The Child Allowance amount is EUR 19.18 per month (twice the Child Allowance Rate of EUR 9.59) for the first and second child and EUR 57.54 per month (six times the Child Allowance Rate) for the third and any other child. These amounts are defined as constants in the policy *ConstDef_ee*.

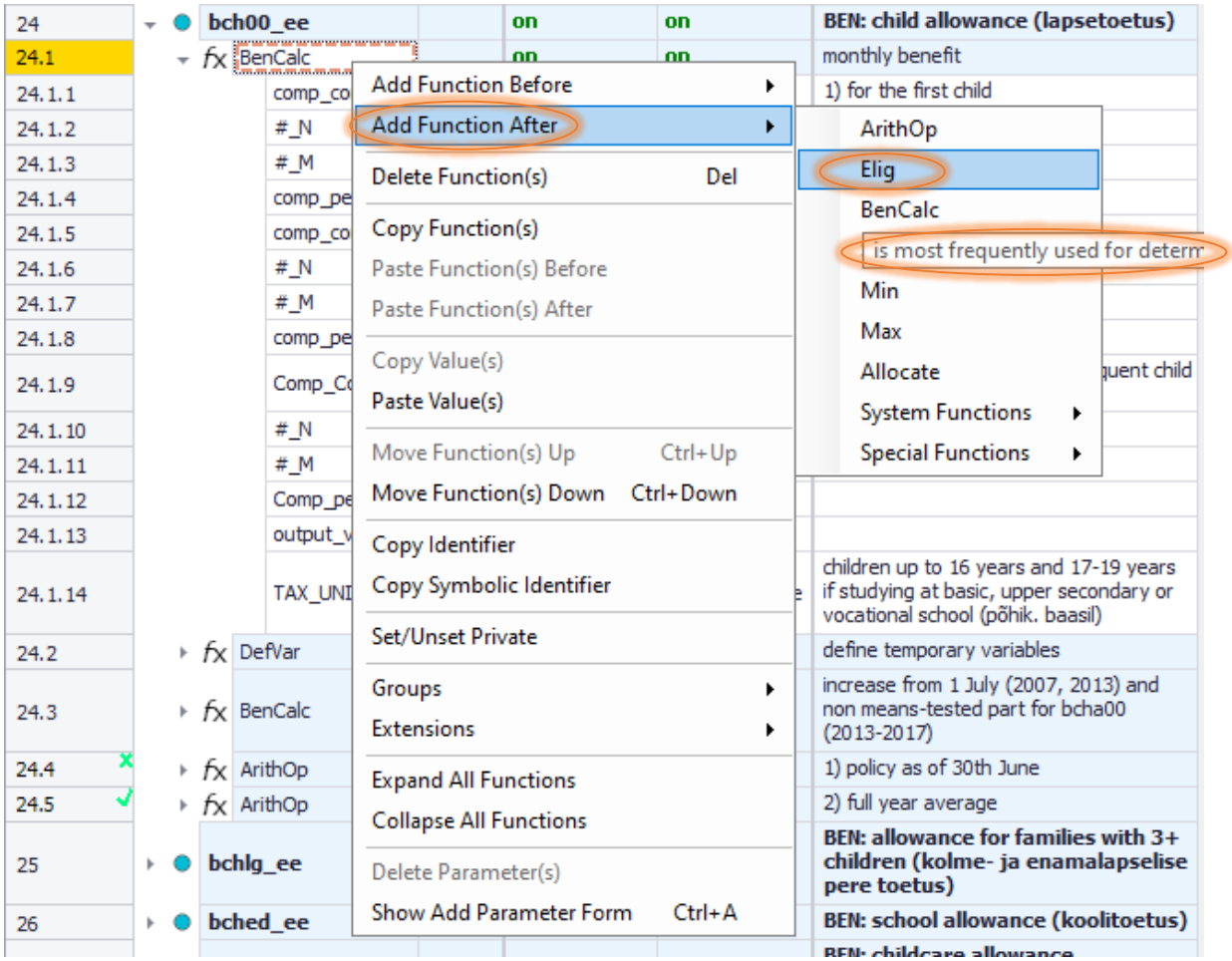
Figure 3a.6: Exploring the bch00_ee (Child Allowance) policy

24	**bch00_ee**		**on**	**on**	BEN: child allowance (lapsetoetus)
24.1	BenCalc		**on**	**on**	monthly benefit
24.1.1	comp_cond	1	IsNtoMchild#1	IsNtoMchild#1	1) for the first child
24.1.2	#_N	1	1	1	
24.1.3	#_M	1	1	1	
24.1.4	comp_perElig	1	$CB_Ch1	$CB_Ch1	
24.1.5	comp_cond	2	IsNtoMchild#2	IsNtoMchild#2	2) for the second child
24.1.6	#_N	2	2	2	
24.1.7	#_M	2	2	2	
24.1.8	comp_perElig	2	$CB_Ch2	$CB_Ch2	
24.1.9	Comp_Cond	3	IsNtoMchild#3	IsNtoMchild#3	3) for the third and each subsequent child
24.1.10	#_N	3	3	3	
24.1.11	#_M	3	99	99	
24.1.12	Comp_perElig	3	$CB_Ch3plus	$CB_Ch3plus	
24.1.13	output_var		bch00_s	bch00_s	
24.1.14	TAX_UNIT		tu_CBfamily_ee	tu_CBfamily_ee	children up to 16 years and 17-19 years if studying at basic, upper secondary or vocational school (põhik. baasil)
24.2	DefVar		n/a	n/a	define temporary variables
24.3	BenCalc		n/a	n/a	increase from 1 July (2007, 2013) and non means-tested part for bcha00 (2013-2017)
24.4	ArithOp		n/a	n/a	1) policy as of 30th June
24.5	ArithOp		switch	switch	2) full year average

Defining the Eligibility conditions

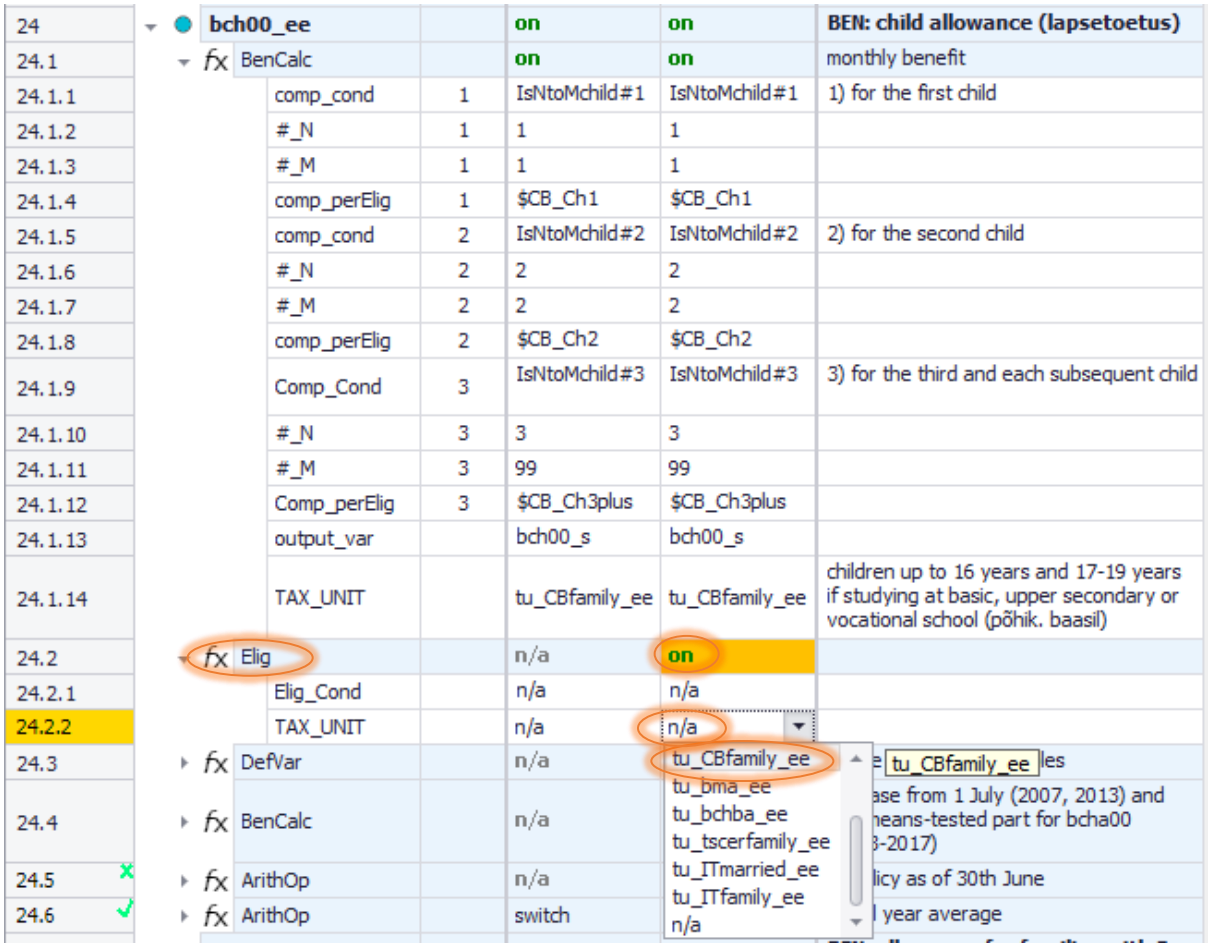
In order to give a supplement to families who have at least two children and satisfy the income test, we need to introduce an eligibility function (*Elig*) to define the families entitled to receive the supplement. Right-click on the function or a parameter name in the *BenCalc* function in *bch00_ee* to activate the context menu. Select 'Add Function After' and select function *Elig*. Note that via a 'hover over' tooltip, EUROMOD can provide a short explanation of the function (Figure 3a.7).

Figure 3a.7: Adding a new *Elig* function after the first *BenCalc*



This creates the new function with its two compulsory parameters - *Elig_Cond* and *TAX_UNIT* - with default values *n/a*. We should set our new function as *on* for the reform system (EE_2012_E2). The next very important thing is to choose the appropriate tax unit. In our case it will be *tu_CBfamily_ee (tax unit applicable to the child benefit family)* so we should click on the grey arrow and select it from the drop-down list (Figure 3a.8).

Figure 3a.8: Selecting the *tu_CBfamily_ee* tax unit



The next step is to create the eligibility condition. The supplement should be restricted to families i) with at least two children and ii) with earnings below 200 EUR per month. First, instead of generating a variable which contains the number of dependent children in each family, we can use the already-defined query *nDepChildrenInTu*. Queries contain the result of ready-made calculations: go to tab *Help&Info* and search for *Queries* for a description (Figure 3a.9). For a description of the query *nDepChildrenInTu* see Figure 3a.10. Second, the variable for earnings is called *yem*.

Figure 3a.9: EUROMOD Help for queries

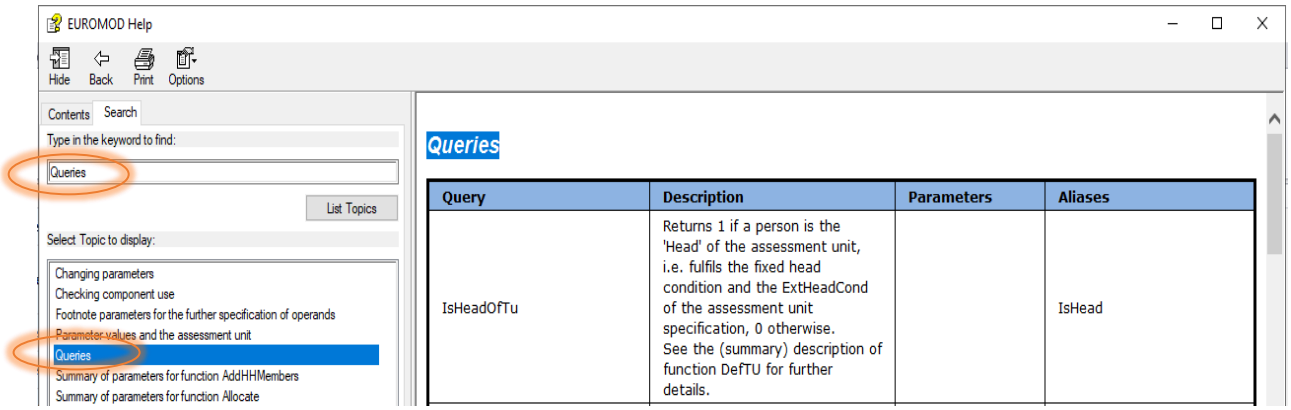


Figure 3a.10 EUROMOD Help entry for *nDepChildrenInTu* Query

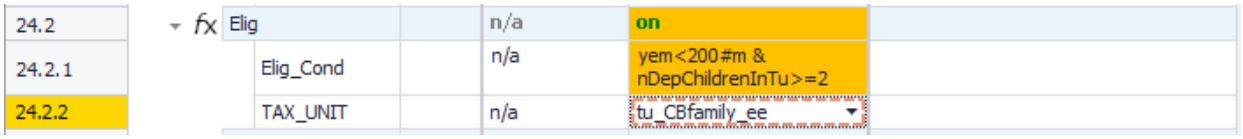
nDepChildrenInTu#x	Returns the number of dependent children in the assessment unit who fulfil dag >= parameter #_AgeMin and dag <= parameter #_AgeMax. For being counted as dependent child the query IsDepChild must apply.	#_AgeMin; optional #_AgeMax; optional	nDepChInTu nDepChildrenInTaxunit nDepChInTaxunit

So, we should type in *Elig_Cond*:

yem < 200#m & nDepChildrenInTU >= 2

(*#m* = per month). By doing this we have formulated a means test based on earnings below 200 EUR per month and number of dependent children equal to at least 2. Note that the means test is at the family level, i.e. the unit of assessment is tu_CBfamily_ee (Figure 3a.11).

Figure 3a.11: The modified new *Elig* function



Calculating the benefit amount

The other part of our task is to calculate the benefit amount of 1800 EUR per month. So, we need an arithmetic function *ArithOp after* the function *Elig*:

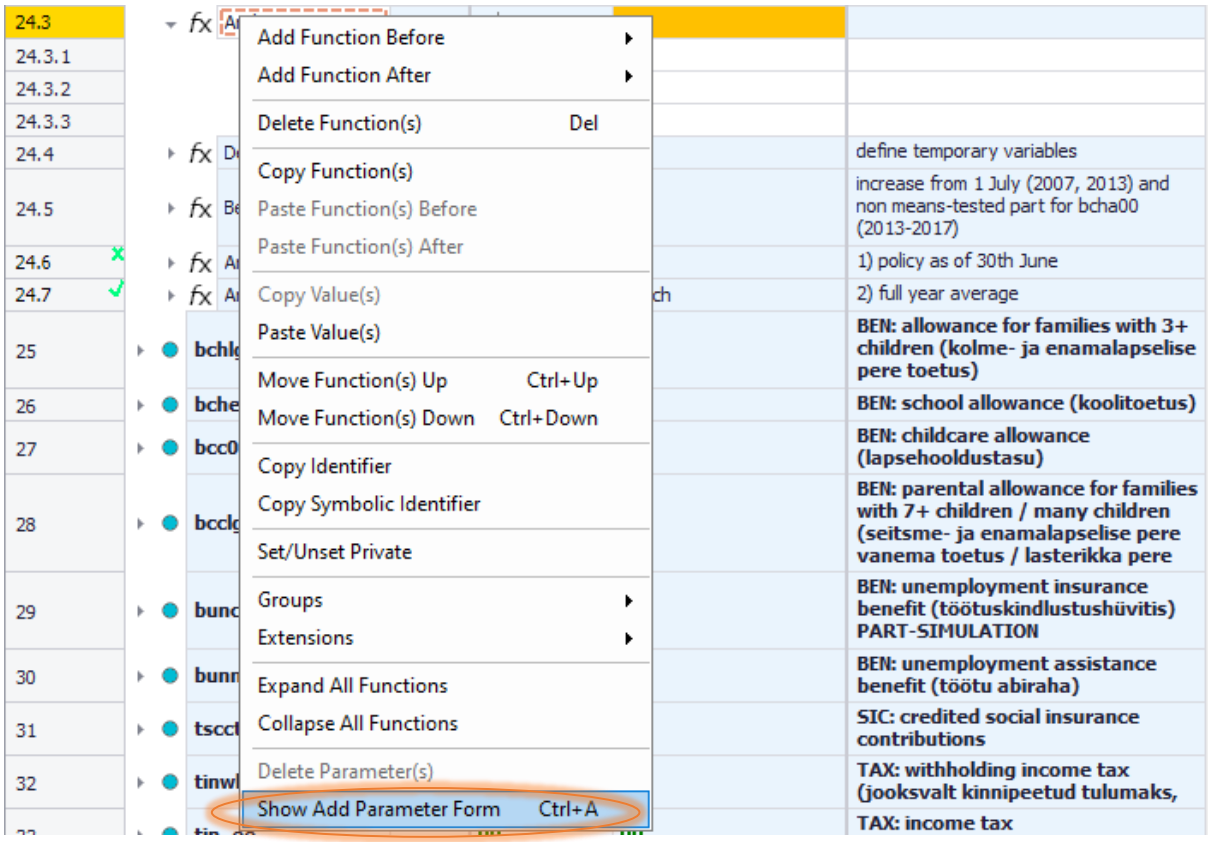
- Right-click on function *Elig* => *Add Function After* => *ArithOp*. This creates the new function with its three compulsory parameters: *Formula*, *Output_var* and *TAX_UNIT* with default values *n/a*.
- Set the function to *on* for the reform system (EE_2012_E2).
- Set the value of the *Output_Var* to the simulated benefit bch00_s.
- Define the tax unit: it will be the same as in the previous function - tu_CBfamily_ee.

- Set the value of the parameter *Formula* from n/a to 1800#m (1,800EUR per month - the amount of the supplement).

The important thing to remember is that we need to assign the benefit only to families who satisfy the eligibility condition. This is done by creating an interaction between the eligibility function (*Elig*) and the arithmetical function (*ArithOp*) by using the parameter *Who_Must_Be_Elig*. How do we do that?

Point the cursor at the function *ArithOp*, then right-click on the function or a parameter name in *ArithOp* and select *Show Add Parameter Form*; alternatively use the shortcut Ctrl+A (Figure 3a.12).

Figure 3a.12: Showing the *Add Parameter Form*



In the *Add Parameters* list we need to select the parameter *Who_Must_Be_Elig* and click the *Add* button (Figure 3a.13). You can drag the new parameter in order to have it at the beginning of the function. If EUROMOD warns you with "Please note that the action will have effect on the hidden systems as well", click ok. Set its value to *one* or any of the other values (all, all_adults, one_adult) – all will give the same result as the eligibility condition is at the family unit and takes, in this case, the same value for each member of the unit (Figure 3a.14).

Figure 3a.13: Inserting the *who_must_be_elig* parameter

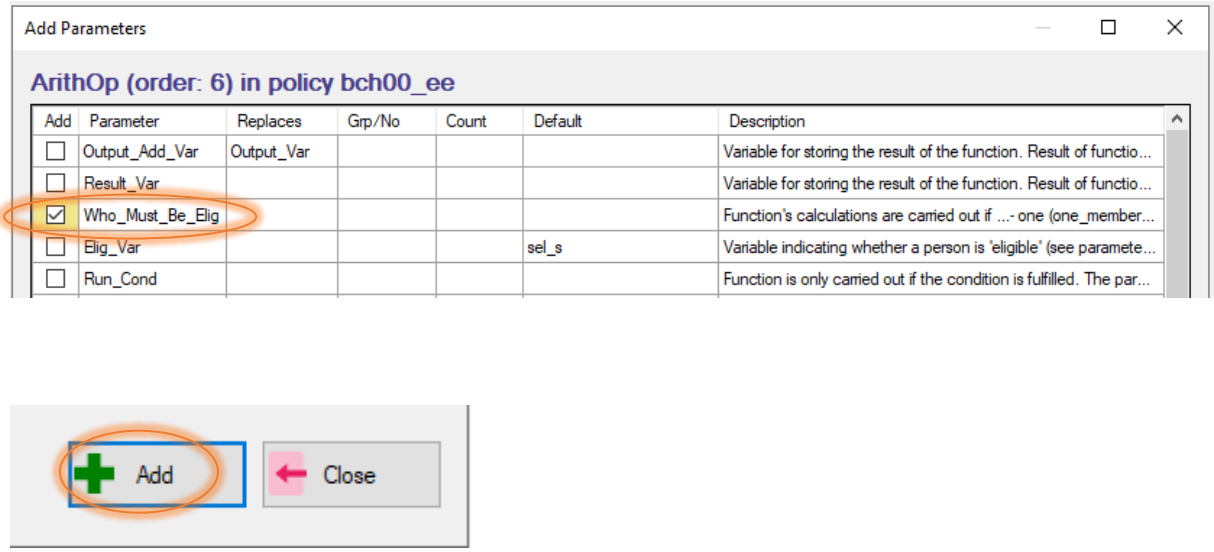
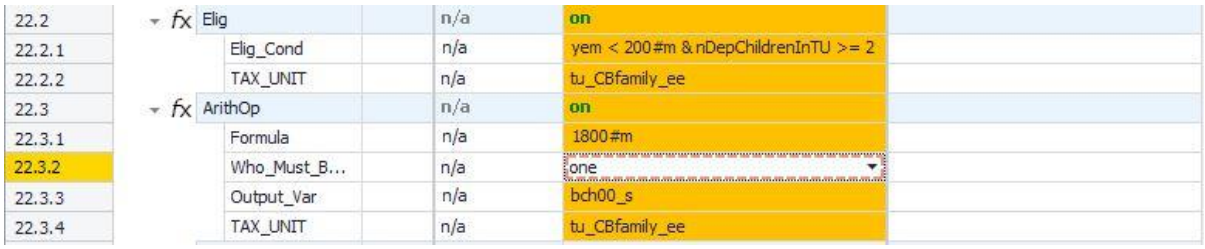


Figure 3a.14: Adding value "one" for the *who_must_be_elig* parameter



Next, we will need to replace parameter *Output_Var* with parameter *Output_Add_Var*, in order to add the supplement to (rather than replace the value of) the benefit calculated with the first *BenCalc* function. To do this, simply left-click on the parameter *Output_Var* and you will be asked if you would like to replace it (Figure 3a.15). Your reform is now ready to run as in Figure 3a.16.

Figure 3a.15: Replacing parameter *Output_Var* with parameter *Output_Add_Var*

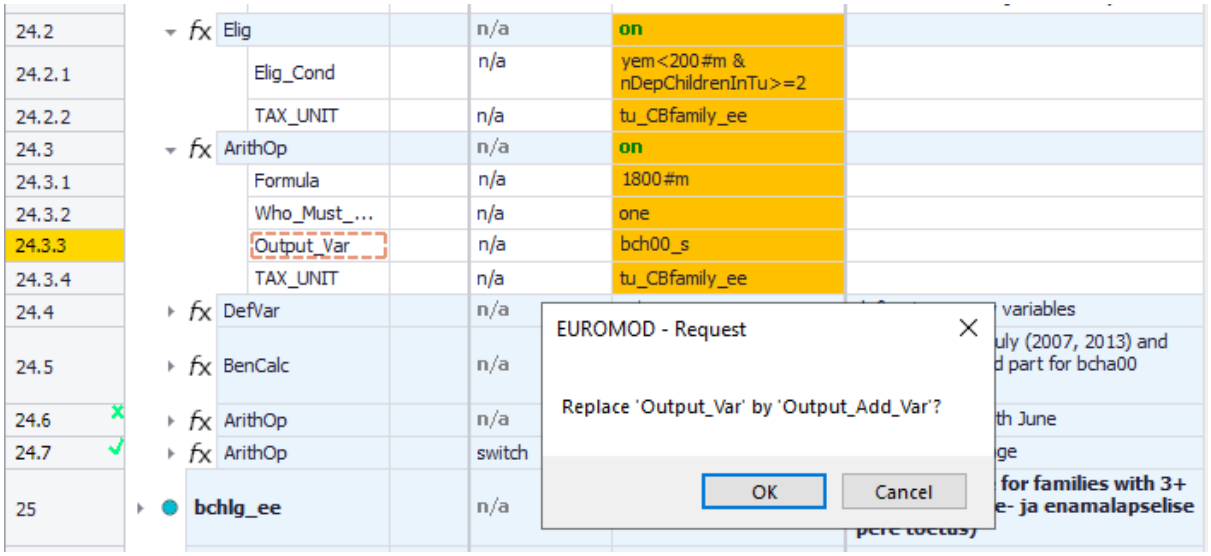


Figure 3a.16: The complete implementation

24	**bch00_ee**		on	on	**BEN: child allowance (lapsetoetus)**
24.1	BenCalc		on	on	monthly benefit
24.1.1	comp_cond	1	IsNtoMchild#1	IsNtoMchild#1	1) for the first child
24.1.2	#_N	1	1	1	
24.1.3	#_M	1	1	1	
24.1.4	comp_perElig	1	$CB_Ch1	$CB_Ch1	
24.1.5	comp_cond	2	IsNtoMchild#2	IsNtoMchild#2	2) for the second child
24.1.6	#_N	2	2	2	
24.1.7	#_M	2	2	2	
24.1.8	comp_perElig	2	$CB_Ch2	$CB_Ch2	
24.1.9	Comp_Cond	3	IsNtoMchild#3	IsNtoMchild#3	3) for the third and each subsequent child
24.1.10	#_N	3	3	3	
24.1.11	#_M	3	99	99	
24.1.12	Comp_perElig	3	$CB_Ch3plus	$CB_Ch3plus	
24.1.13	output_var		bch00_s	bch00_s	
24.1.14	TAX_UNIT		tu_CBfamily_ee	tu_CBfamily_ee	children up to 16 years and 17-19 years if studying at basic, upper secondary or vocational school (põhik. baasil)
24.2	Elig		n/a	on	eligibility for supplement
24.2.1	Elig_Cond		n/a	yem<200#m & nDepChildrenInTu>=2	earnings should be less than EUR 200 per month and there should be at least 2 children in the family
24.2.2	TAX_UNIT		n/a	tu_CBfamily_ee	
24.3	ArithOp		n/a	on	supplement
24.3.1	Formula		n/a	1800#m	amount of EUR 1800 per month
24.3.2	Who_Must_...		n/a	one	
24.3.3	Output_Add...		n/a	bch00_s	add amount to the results of the previous function
24.3.4	TAX_UNIT		n/a	tu_CBfamily_ee	

Save your changes.

Task 3: Producing the micro-output

Run EUROMOD for the baseline EE_2012 and reform EE_2012_E2 systems using *training_data*.

Task 4: Analyse the results

Use the *Statistics Presenter* (Baseline/Reform option) to analyse the fiscal and distributional impact of the reform.

Results:

Table 3a: Government Expenditure
Yearly, mill., currency as defined in EM output

	ee_2012 (base)	*ee_2012_e2*	*Difference to base*
Government expenditure on social transfers	43.74	47.14	3.40
... family and education benefits	0.87	4.32	3.46
... social assistance and housing benefits	0.58	0.53	-0.06

Table 3b: Basic Poverty Indices

	Poverty Risk for ee_2012 (base)	Poverty Risk for ee_2012_e2	Difference to base
Population	24.05%	21.59%	-2.46pp
Children	20.62%	14.79%	-5.84pp
Working Age	30.42%	28.55%	-1.87pp
Working Age Economically Active	18.71%	17.39%	-1.32pp
Elderly	2.99%	2.49%	-0.50pp
Fixed Poverty Line	811.44		

Exercise 3b: Reforming Child benefit in Greece

The aim of this exercise is to get familiar with functions DefVar, Elig and ArithOp.

Steps:

1. Create a new system in Greece based on the 2021 system and name it EL_2021_ref;

2. Add a supplement to the existing child benefit (bch_s). The supplement is given to recipient families with at least one child aged up to 6. Its aim is to increase the child benefit amount to 300 EUR per month (and not to reduce it, if it is higher than this amount);

3. Run EUROMOD to produce micro-outputs for these two systems (EL_2021 and EL_2021_ref);

4. Analyse the results with the statistics presenter.

Description:

The aim of this exercise is to perform a reform of the child benefit (bch_el) in Greece in 2021. In that year, the benefit was means-tested, paid to families with children below an age limit. The basic amount for the first and the second child was 70 EUR/month, and for the third and beyond 140 EUR/month.

Your task is to add a supplement to this benefit, using functions *DefVar*, *Elig* and *ArithOp*. The supplement is given to recipient families with at least one child up to age of 6, and increases the child benefit amount to 300 EUR per month. It does not reduce it, if it is higher than this amount.

Hints:

- Use DefVar to define a temporary variable for this top up. You can name it i_bch.
- Use the same tax/assessment unit for your calculations as the one already used in this policy, i.e. *tu_bch_el*.

Solution and further information

Task 1: Creating a new system

Our first step is to open Greece (country acronym EL). Hide all other systems except for EL_2021. We then go to this system and right-click on it. This provides several options (see Figure 3a.4). You can rename the system to *EL_2021_ref* using the option *Rename System*.

Task 2: Implementing the child benefit reform

The next task to implement the reform in the new system is to find the Child benefit policy in the spine. The easiest way to do this is to use the search option (Ctrl+F), i.e. search for **bch_el.**

Now we need to define the intermediate variable for the benefit supplement. We do that by adding a new *DefVar* function and include a parameter for the intermediate variable definition.
Adding the new function: right-click on the last function of the policy (policy 50.10) to activate the context menu. Select 'Add Function After' and then select the system function *DefVar*. Set the function as *on* for the reform system (EL_2021_ref).
Adding parameter to define the intermediate variable: point the cursor at the new function, right-click on it and select *Show Add Parameter Form*. In the *Add Parameters* list select the parameter *[Placeholder]* and click the *Add* button. Name the intermediate variable i_bch and set its original value equal to zero.

Figure 3b.1: Defining the intermediate variable

	Policy	Grp/No	EL_2021	EL_2021_ref	Comment
49	**yiv_el**		**on**	**on**	**DEF: imputed income (τεκμαρτό εισόδημα)**
50	**bch_el**		**on**	**on**	**BEN: child benefit (επίδομα παιδιού)**
50.1	BenCalc		**on**	**on**	equivalence scale for children
50.2	BenCalc		**on**	**on**	equivalence scale for household head
50.3	BenCalc		**on**	**on**	equivalence scale for non household head
50.4	BenCalc		**on**	**on**	additional equivalence scale for lone parents
50.5	ArithOp		**on**	**on**	define household equivalent weight
50.6	ArithOp		**on**	**on**	define household equivalent income
50.7	BenCalc		**on**	**on**	full amount
50.8	BenCalc		**on**	**on**	reduction ratios
50.9	ArithOp		**on**	**on**	apply reduction to benefit
50.10	BenCalc		switch	switch	non take-up adjustment
50.11	DefVar		n/a	**on**	define intermediate variables
50.11.1	i_bch	1	n/a	0	child benefit top up

To give the supplement to recipient families who have at least one child up to the age of 6, we need to introduce an eligibility function (*Elig*). Right-click on the *DefVar* function to activate the context menu again. Select 'Add Function After' and select function *Elig*. This creates the new function with its two compulsory parameters - *Elig_Cond* and *TAX_UNIT* - with default values *n/a*. Set our new function as *on* for the reform system.

The next step is to choose the appropriate tax unit. In our case it will be *tu_bch_el* (the tax unit applicable to the child benefit policy) so in the *TAX_UNIT* field, click on the grey arrow and select it from the drop-down list (similar to Figure 3a.8 in the previous exercise). To create the eligibility condition we fill in the Elig_Cond field with a specific expression able to capture eligible tax units. The supplement

should be restricted to families i) already receiving the child benefit and ii) with at least one child aged up to 6. The expression to be typed in *Elig_Cond*: bch_s > 0 & dag <= 6.

Please keep in mind that conditions related to monetary variables are assessed at the tax unit level, whereas conditions related to non-monetary variables (such as age) are assessed at the individual level.

Figure 3b.2: The modified new *Elig* function

	Policy	Grp/No	EL_2021	EL_2021_ref	Comment
49	**yiv_el**		**on**	**on**	**DEF: imputed income (τεκμαρτό εισόδημα)**
50	**bch_el**		**on**	**on**	**BEN: child benefit (επίδομα παιδιού)**
50.1	BenCalc		**on**	**on**	equivalence scale for children
50.2	BenCalc		**on**	**on**	equivalence scale for household head
50.3	BenCalc		**on**	**on**	equivalence scale for non household head
50.4	BenCalc		**on**	**on**	additional equivalence scale for lone parents
50.5	ArithOp		**on**	**on**	define household equivalent weight
50.6	ArithOp		**on**	**on**	define household equivalent income
50.7	BenCalc		**on**	**on**	full amount
50.8	BenCalc		**on**	**on**	reduction ratios
50.9	ArithOp		**on**	**on**	apply reduction to benefit
50.10	BenCalc		switch	switch	non take-up adjustment
50.11	DefVar		n/a	**on**	define intermediate variables
50.11.1	i_bch	1	n/a	0	child benefit top up
50.12	Elig		n/a	**on**	eligibility conditions for top up
50.12.1	Elig_Cond		n/a	bch_s > 0 & dag <= 6	
50.12.2	TAX_UNIT		n/a	tu_bch_el	

Our next task is to calculate the supplement amount. We do it with the arithmetic function *ArithOp* which is placed just after the function *Elig*. Right-click on function *Elig => Add Function After => ArithOp*. This creates the new function with its three compulsory parameters: *Formula*, *Output_var* and *TAX_UNIT* with default values *n/a*.

- Set the function to *on* for the reform system (EL_2021_ref)
- Set the value of the parameter *Formula* from n/a to 300#m – bch_s
- Set the value of the *Output_Var* to the simulated benefit bch_s
- Define the tax unit: it will be the same as in the previous function - tu_bch_el

Remember that we need to assign the top up only to families who satisfy the eligibility condition. This is done by creating an interaction between the eligibility function (*Elig*) and the arithmetical function (*ArithOp*) by using the parameter *Who_Must_Be_Elig*. Point the cursor at the function *ArithOp*, then right-click on the function or a parameter name in *ArithOp* and select *Show Add Parameter Form*. In the *Add Parameters* list we need to select the parameter *Who_Must_Be_Elig* and click the *Add* button. You can drag the new parameter in order to have it at the beginning of the function. Set its value to *one*, as we need at least one member of a child benefit-recipient family complying with the age criterion.

As far as the top up amount is concerned, our current implementation (300#m – bch_s) does not guarantee that the original child benefit is not reduced by the top up (as bch_s can be higher than 300#m). To do so, we need to add a new parameter to the *ArithOp* function: *LowLim*. We do that the same we introduced the *Who_Must_Be_Elig* parameter, and set it equal to zero.

Moreover, as we would like to have the (intermediate) child benefit top up variable included in our output file, and not just the final child benefit (this is very useful for checking purposes), we need to add another parameter to the function, namely *Result_Var*, and set it equal to i_bch.

Finally, we will need to replace parameter *Output_Var* with parameter *Output_Add_Var*, in order to add the supplement to the already calculated child benefit (and not replace its original value with the one of the top up). To do this, simply left-click on the parameter *Output_Var* and you will be asked if you would like to replace it. Your reform is now ready to run.

Figure 3b.3: The modified new *ArithOp* function

	Policy	Grp/No	EL_2021	EL_2021_ref	Comment
49	**yiv_el**		on	on	**DEF: imputed income (τεκμαρτό εισόδημα)**
50	**bch_el**		on	on	**BEN: child benefit (επίδομα παιδιού)**
50.1	BenCalc		on	on	equivalence scale for children
50.2	BenCalc		on	on	equivalence scale for household head
50.3	BenCalc		on	on	equivalence scale for non household head
50.4	BenCalc		on	on	additional equivalence scale for lone parents
50.5	ArithOp		on	on	define household equivalent weight
50.6	ArithOp		on	on	define household equivalent income
50.7	BenCalc		on	on	full amount
50.8	BenCalc		on	on	reduction ratios
50.9	ArithOp		on	on	apply reduction to benefit
50.10	BenCalc		switch	switch	non take-up adjustment
50.11	DefVar		n/a	on	define intermediate variables
50.11.1	i_bch	1	n/a	0	child benefit top up
50.12	Elig		n/a	on	eligibility conditions for top up
50.12.1	Elig_Cond		n/a	bch_s > 0 & dag <= 6	
50.12.2	TAX_UNIT		n/a	tu_bch_el	
50.13	ArithOp		n/a	on	top up amount
50.13.1	Who_Must_Be_Elig		n/a	one	
50.13.2	Formula		n/a	300#m - bch_s	
50.13.3	LowLim		n/a	0	
50.13.4	Result_Var		n/a	i_bch	
50.13.5	Output_Add_Var		n/a	bch_s	
50.13.6	TAX_UNIT		n/a	tu_bch_el	

Task 3: Producing the micro-output

Save your changes and run EUROMOD for the baseline EL_2021 and reform EL_2021_ref systems using *training_data*.

Task 4: Analyse the results

Use the *Statistics Presenter* (Baseline/Reform option) to analyse the fiscal and distributional impact of the reform.

Results:

Table 3c: Government Expenditure (yearly, mill. EUR)

Government expenditure on social transfers	**36.88**	**37.36**	**0.49**	**1.32 %**
by target group				
... unemployment benefits	0.00	0.00	0.00	0.00 %
... family and education benefits	0.44	0.92	0.49	111.25 %
... social assistance and housing benefits	0.00	0.00	0.00	0.00 %
... pensions, health and disability benefits	36.44	36.44	0.00	0.00 %
... firms	0.00	0.00	0.00	0.00 %
by benefit design				
... means-tested non-pension benefits	0.44	0.92	0.49	111.25 %
... non-means-tested non-pension benefits	0.00	0.00	0.00	0.00 %
... pensions	36.44	36.44	0.00	0.00 %
... firms subsidies	0.00	0.00	0.00	0.00 %

Table 3d: Basic inequality and poverty indices

3.1 Basic Inequality Indices

	Gini EL_2021	S80/S20 EL_2021	Gini EL_2021_ref	Diff. Gini EL_2021_ref/EL_2021	S80/S20 EL_2021_ref	Diff. S80/S20 EL_2021_ref/EL_2021
Original Income	0.4788	10.1067	0.4788	0.0000	10.3106	0.2038
Original Income after Taxes/SIC	0.4663	8.3544	0.4663	0.0000	8.5187	0.1643
Original Income incl. Public Pensions after Taxes/SIC	0.3277	7.0782	0.3277	0.0000	7.0211	-0.0571
Disposable Income	0.3244	6.8140	0.3208	-0.0036	6.6126	-0.2014

3.2.a Basic Poverty Indices

	Poverty Risk for EL_2021 (base)	Poverty Risk for EL_2021_ref	Difference to base
At-risk-of-poverty rate	22.06 %	20.79 %	-1.27pp
Median at-risk-of-poverty gap	37.22 %	38.88 %	1.66pp
Fixed Poverty line (monthly)	793.17		

Exercise 4: Introducing a supplement, withdrawn with earnings, to the Child Allowance in Estonia

The aim of this exercise is to learn how to use the function BenCalc.

Steps:

1. Create a new system in Estonia based on the 2021 system and name it EE_2021ref1;

2. Introduce a supplement for large families to the Child Allowance, which is withdrawn with earnings;

3. Run EUROMOD to produce micro-outputs for the baseline and reform systems;

4. Analyse the distributive impact of this reform.

Description:

Perform a reform of the Child Allowance in Estonia in 2021. Your task is to **add** a supplement to the Child Allowance for families with at least 2 dependent children below the age of 18, using the function *BenCalc*. The supplement is equal to €30 per week for the whole family but should be withdrawn with earnings: 10% should be withdrawn for each €1 that the family earns above €10,000 per year. Analyse the results after introducing the reform.

Hints:

- To calculate the earnings withdrawal, use *BenCalc*'s optional parameters *Withdraw_Base*, *Withdraw_Start* and *Withdraw_Rate*.
- Use the same tax/assessment unit for your calculations as the one defined in the *BenCalc* function in the same policy, i.e. *tu_CBfamily_ee* (the family/benefit unit).
- Use the same variable name for your calculations as the one used in the *BenCalc* function in the same policy, i.e. *bch00_s*.
- The variable for earnings is called *yem*.

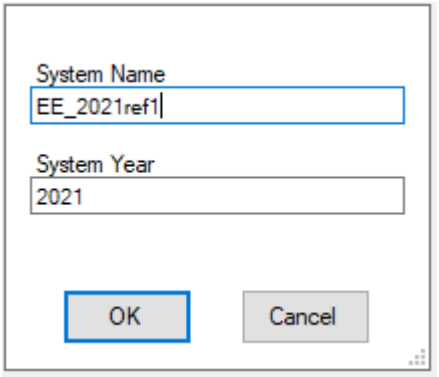
Solution and further information

Task 1: Creating a new system

Load the Estonian model. Hide all systems except for EE_2021. Add a new reform system based on EE_2021 and call it EE_2021ref1, as shown in Figure 4.1 and Figure 4.2.

Figure 4.1: Copying from the existing system (EE_2021)

Figure 4.2: Renaming the new system

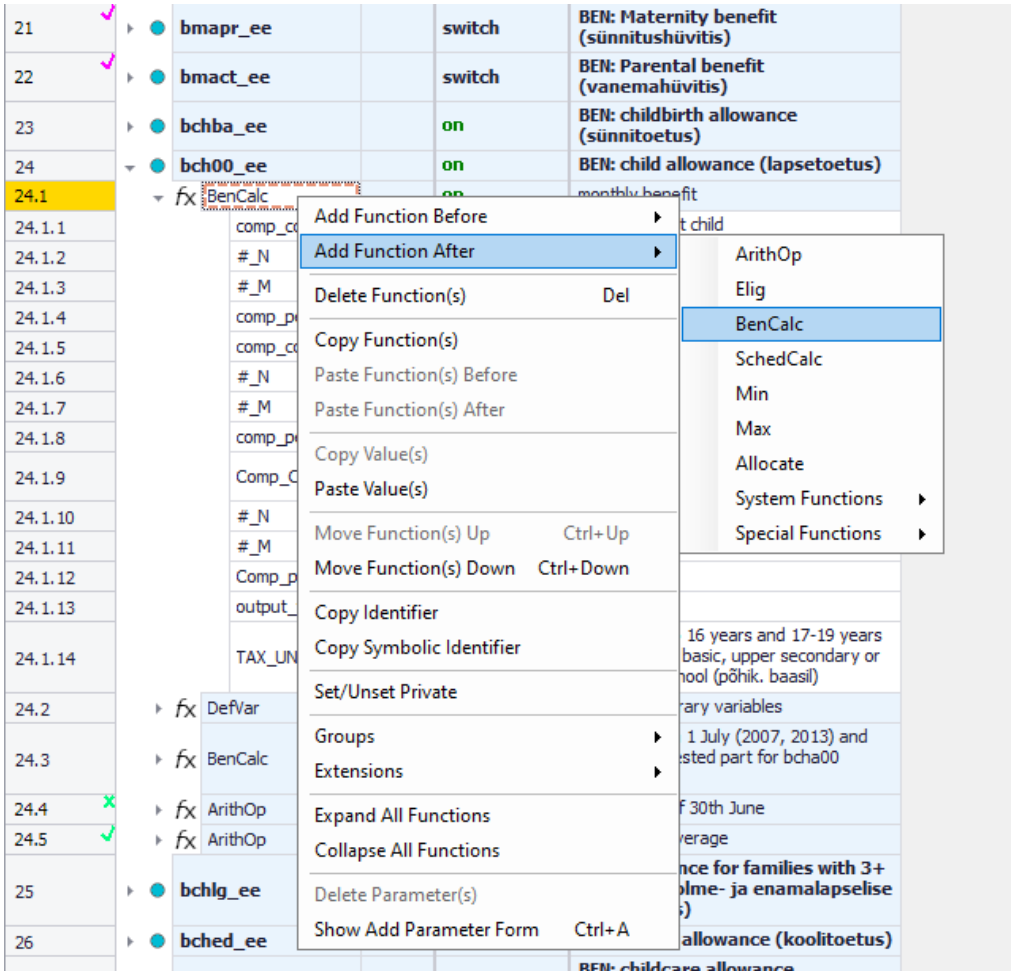


Task 2: Introduce a supplement to the Child Benefit

First, to introduce the supplement, we need a function which defines **who is entitled to it and how much the supplement amounts to**. We will use *BenCalc* (benefit calculator) function to help us with that, as it allows for modelling a wide range of policy instruments, particularly benefits.

- Locate the Child Benefit policy (bch00_ee) on the policy spine, right-click on the first function *BenCalc* and select from the option *Add Function After* the function *BenCalc*, as shown in Figure 4.3. This creates the new function with its compulsory parameters *Comp_Cond*, *Comp_perTU*, *output_var* and *TAX_UNIT* (with default values n/a). Note that the parameters *Comp_Cond* and *Comp_PerTU* are already grouped together into one component (column "Grp/No"), so you do not need to do anything. Set the function to **'on'** for the reform system (EE_2021ref1).

Figure 4.3: Adding a new BenCalc function after the current function in policy bch00_ee



- For the system we are reforming (EE_2021ref1), change the value of the parameter *Comp_Cond* to *nDepChildrenInTU#x>=2*. Define the age limits there. Thus, the eligibility condition says that there should be at least 2 dependent children below 18 in the benefit unit.
- Change the value of the parameter *Comp_perTU* to *30#*w i.e. €30 per week.
- Change the parameter value *Output_Var* from *n/a* to the simulated benefit *bch00_s*. You also need to use the parameter *Output_Add_Var* instead of *Output_Var*, to add the supplement to (rather than replace the value of) the benefit calculated with the first *BenCalc* function. To do this, simply click on the parameter *Output_Var* and you will be asked if you would like to replace it.

- Set the value of the parameter *TAX_UNIT* from *n/a* to *tu_CBfamily_ee* (i.e. the relevant benefit unit or the family used to calculate the Child Benefit).

Your reformed system should now look like Figure 4.4.

Figure 4.4: The modified function reform system – task 2

	Policy	Grp/No	EE_2021	EE_2021ref1	Comment
22	**bmact_ee**		**switch**	**switch**	**BEN: Parental benefit (vanemahüvitis)**
23	**bchba_ee**		**on**	**on**	**BEN: childbirth allowance (sünnitoetus)**
24	**bch00_ee**		**on**	**on**	**BEN: child allowance (lapsetoetus)**
24.1	fx BenCalc		**on**	**on**	monthly benefit
24.1.1	comp_cond	1	IsNtoMchild#1	IsNtoMchild#1	1) for the first child
24.1.2	#_N	1	1	1	
24.1.3	#_M	1	1	1	
24.1.4	comp_perElig	1	$CB_Ch1	$CB_Ch1	
24.1.5	comp_cond	2	IsNtoMchild#2	IsNtoMchild#2	2) for the second child
24.1.6	#_N	2	2	2	
24.1.7	#_M	2	2	2	
24.1.8	comp_perElig	2	$CB_Ch2	$CB_Ch2	
24.1.9	Comp_Cond	3	IsNtoMchild#3	IsNtoMchild#3	3) for the third and each subsequent child
24.1.10	#_N	3	3	3	
24.1.11	#_M	3	99	99	
24.1.12	Comp_perElig	3	$CB_Ch3plus	$CB_Ch3plus	
24.1.13	output_var		bch00_s	bch00_s	
24.1.14	TAX_UNIT		tu_CBfamily_ee	tu_CBfamily_ee	children up to 16 years and 17-19 years if studying at basic, upper secondary or vocational school (põhik. baasil)
24.2	fx BenCalc		n/a	**on**	
24.2.1	Comp_Cond	1	n/a	nDepChildrenInTu#1>=2	
24.2.2	Comp_perTU	1	n/a	30#w	
24.2.3	#_AgeMin	1	n/a	0	
24.2.4	#_AgeMax	1	n/a	18	
24.2.5	Output_Add_Var		n/a	bch00_s	
24.2.6	TAX_UNIT		n/a	tu_CBfamily_ee	

Second, we need to **calculate the benefit withdrawal**:

- Right-click on the function or a parameter name of the new *BenCalc* function and select *Show Add Parameter* Form. In the Add Parameters list select the parameters *Withdraw_Base*, *Withdraw_Start*, *Withdraw_Rate* and click the *Add* button, as shown in Figure 4.5. You can drag the parameters within the *BenCalc* function to change their position, e.g. after *Comp_perTU*. Set *Withdraw_Base* equal to *yem* (variable for earnings), *Withdraw_Start* to *10000#y* and *Withdraw_Rate* to *0.1*, as shown in Figure 4.6.

Figure 4.5: adding BenCalc parameters *Withdraw_Base, Withdraw_Start, Withdraw_Rate*

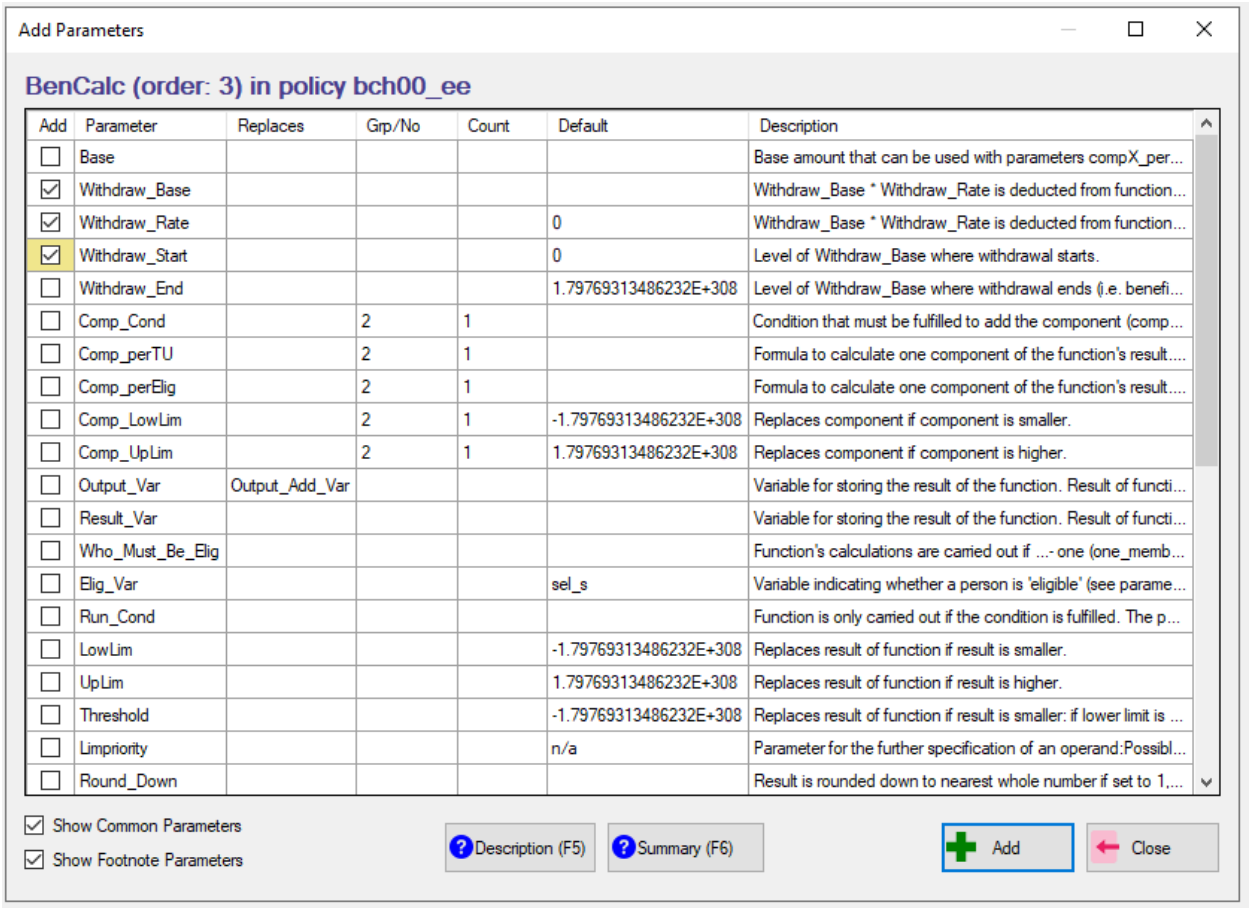


Figure 4.6: The completed modified function reform system

	Policy	Grp/No	EE_2021	EE_2021ref1	Comment
22	**bmact_ee**		**switch**	**switch**	**BEN: Parental benefit (vanemahüvitis)**
23	**bchba_ee**		**on**	**on**	**BEN: childbirth allowance (sünnitoetus)**
24	**bch00_ee**		**on**	**on**	**BEN: child allowance (lapsetoetus)**
24.1	fx BenCalc		**on**	**on**	monthly benefit
24.1.1	comp_cond	1	IsNtoMchild#1	IsNtoMchild#1	1) for the first child
24.1.2	#_N	1	1	1	
24.1.3	#_M	1	1	1	
24.1.4	comp_perElig	1	$CB_Ch1	$CB_Ch1	
24.1.5	comp_cond	2	IsNtoMchild#2	IsNtoMchild#2	2) for the second child
24.1.6	#_N	2	2	2	
24.1.7	#_M	2	2	2	
24.1.8	comp_perElig	2	$CB_Ch2	$CB_Ch2	
24.1.9	Comp_Cond	3	IsNtoMchild#3	IsNtoMchild#3	3) for the third and each subsequent child
24.1.10	#_N	3	3	3	
24.1.11	#_M	3	99	99	
24.1.12	Comp_perElig	3	$CB_Ch3plus	$CB_Ch3plus	
24.1.13	output_var		bch00_s	bch00_s	
24.1.14	TAX_UNIT		tu_CBfamily_ee	tu_CBfamily_ee	children up to 16 years and 17-19 years if studying at basic, upper secondary or vocational school (põhik. baasil)
24.2	fx BenCalc		n/a	**on**	
24.2.1	Comp_Cond	1	n/a	nDepChildrenInTu#1>=2	
24.2.2	Comp_perTU	1	n/a	30#w	
24.2.3	#_AgeMin	1	n/a	0	
24.2.4	#_AgeMax	1	n/a	18	
24.2.5	Withdraw_Base		n/a	yem	
24.2.6	Withdraw_Rate		n/a	0.1	
24.2.7	Withdraw_Start		n/a	10000#y	
24.2.8	Output_Add_Var		n/a	bch00_s	
24.2.9	TAX_UNIT		n/a	tu_CBfamily_ee	

Finally, save your changes.

Task 3: Producing the micro-output

Run EUROMOD for the baseline EE_2021 and reform EE_2021ref1 systems with *training_data* as the dataset.

Task 4: Analyse the results

Use the *Statistics Presenter* – option Baseline/Reform – to analyse the fiscal and poverty impact of the reform.

Results

Summary Statistics - Baseline vs Reforms

Results for Estonia: EE_2021 vs EE_2021ref1

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Market Incomes and Government Revenue & Expenditure

Yearly, mill., currency as defined in EM output

	EE_2021 (base)	EE_2021ref1	Difference to base
Total market incomes	**137.90**	**137.90**	**0.00**
... income from (self) employment	137.90	137.90	0.00
... other sources	0.00	0.00	0.00
Government revenue through taxes and social insurance contributions	**70.73**	**70.73**	**0.00**
... direct taxes	31.77	31.77	0.00
... employee social insurance contributions	1.73	1.73	0.00
... self-employed social insurance contributions	0.00	0.00	0.00
... other social insurance contributions	0.00	0.00	0.00
... employer social insurance contributions (not part of disposable income)	37.23	37.23	0.00
Credited social insurance contributions (not part of disposable income)	**0.79**	**0.79**	**0.00**
Government expenditure on social transfers	**46.13**	**46.41**	**0.29**
by target group			
... unemployment benefits	0.00	0.00	0.00
... family and education benefits	2.62	2.94	0.32
... social assistance and housing benefits	1.21	1.18	-0.03
... pensions, health and disability benefits	42.29	42.29	0.00
... firms	0.00	0.00	0.00
by benefit design			
... means-tested non-pension benefits	1.21	1.18	-0.03
... non-means-tested non-pension benefits	2.62	2.94	0.32
... pensions	42.29	42.29	0.00
... firms subsidies	0.00	0.00	0.00

Summary Statistics - Baseline vs Reforms

Results for Estonia: EE_2021 vs EE_2021ref1

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Basic Poverty Indices

	Poverty Risk for EE_2021 (base)	Poverty Risk for EE_2021ref1	Difference to base
Population	23.02 %	22.70 %	-0.32pp
Children	18.29 %	17.51 %	-0.78pp
Working Age	29.68 %	29.43 %	-0.25pp
Working Age Economically Active	17.58 %	17.39 %	-0.19pp
Elderly	2.49 %	2.49 %	0.00pp
Fixed Poverty Line	836.38		

Exercise 5a: Introducing a zero tax band to the flat income tax in Bulgaria

The aim of this exercise is to learn how to use the function SchedCalc.

Steps:

1. Create a new system in Bulgaria based on the 2021 system and name it BG_2021_reform;

2. Introduce a zero tax band to the flat income tax and modify the tax rate;

3. Run EUROMOD to produce micro-outputs for the baseline and reform systems;

4. Analyse the distributive impact of this reform.

Description:

Currently, taxable income in Bulgaria is taxed at 10% from the first BGN earned. You are asked to introduce a zero tax band of BGN 3,200 per year, i.e. individuals should pay tax only on taxable income above BGN 3,200 per year. Furthermore, you should raise the marginal rate of income tax from 10% to 15%.

Hints:

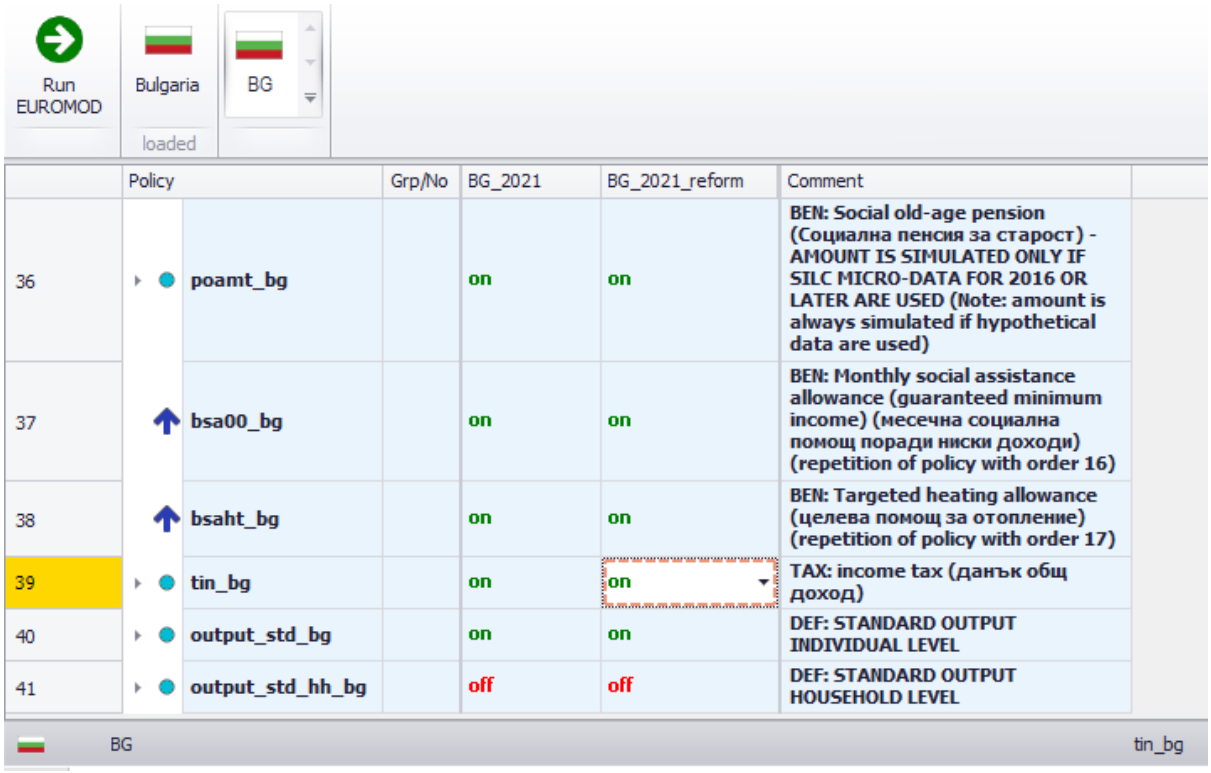
- Define the zero tax band as a constant in the *DefConst* function in the income tax policy (e.g. call it *$flat_zeroband*)
- The tax rate is defined as a constant (e.g. *$flat_rate*) so modify the value of that constant.
- Introduce the zero tax band to the *SchedCalc* function, where the flat tax is calculated (*tin_bg*) by adding/modifying parameters in a *SchedCalc* function.

Solution and further information

Task 1: Creating a new system

Open the Bulgarian model. Hide all systems except for BG_2021. Add a new reform system based on BG_2021 and call it BG_2021_reform and locate the Income Tax policy, as shown in Figure 5a.1.

Figure 5a.1: Locating the Income Tax policy (tin_bg)



	Policy	Grp/No	BG_2021	BG_2021_reform	Comment
36	poamt_bg		on	on	BEN: Social old-age pension (Социална пенсия за старост) - AMOUNT IS SIMULATED ONLY IF SILC MICRO-DATA FOR 2016 OR LATER ARE USED (Note: amount is always simulated if hypothetical data are used)
37	bsa00_bg		on	on	BEN: Monthly social assistance allowance (guaranteed minimum income) (месечна социална помощ поради ниски доходи) (repetition of policy with order 16)
38	bsaht_bg		on	on	BEN: Targeted heating allowance (целева помощ за отопление) (repetition of policy with order 17)
39	tin_bg		on	on	TAX: income tax (данък общ доход)
40	output_std_bg		on	on	DEF: STANDARD OUTPUT INDIVIDUAL LEVEL
41	output_std_hh_bg		off	off	DEF: STANDARD OUTPUT HOUSEHOLD LEVEL

BG tin_bg

Task 2: Modify the income tax

The income tax is implemented in the policy *tin_bg*. Since 2008, Bulgaria has had a flat tax of 10%, levied on taxable income (variable *tintb_s*) from the first BGN earned. The flat tax is calculated in the second *SchedCalc* function towards the end of the policy (policy spine row 39.10). The income tax parameters are defined as constants in the first function *DefConst* (policy spine row 39.1).

To implement the reform:

- First, increase the tax rate – constant *$flat_rate* – from 10% to 15% in the *DefConst* function in the reform system *BG_2021_reform*.
- Second, introduce a zero tax band of BGN 3,200 per year, by defining a new constant called *$flat_zeroband* in the *DefConst* function – to do so add a new *[Placeholder]* parameter and type the name for the constant in the *Policy* column. Type 3200#y in as value for the constant in the reform BG_2021_reform, as shown in Figure 5a.2.

Figure 5a.2: Introducing a zero tax band ($flat_zeroband) and new flat rate ($flat_rate)

	Policy	Grp/No	BG_2021	BG_2021_reform	Comment
39	**tin_bg**		on	on	**TAX: income tax (данък общ доход)**
39.1	DefConst		on	on	Parameters used in the income tax policy - defined as constants
39.1.1	$tintach1		200#y	200#y	Child tax deduction amount for the first child in 2007, 2015, 2016
39.1.2	$tintach2		400#y	400#y	Child tax deduction amount for the second child in 2007, 2015, 2016
39.1.3	$tintach3		600#y	600#y	Child tax deduction amount for the all other children in 2007, 2015, 2016
39.1.4	$tintapv		0.1	0.1	Tax deduction for private pension contributions - rate
39.1.5	$tintart		0.1	0.1	Tax deduction for income from rent - rate
39.1.6	$tintase		0.25	0.25	Tax deduction for freelancers
39.1.7	$tintadi		7920#y	7920#y	Tax deduction for disability - applied on the tax base (not taxable income!)
39.1.8	$band1_threshold		n/a	n/a	Zero tax band
39.1.9	$band2_rate		n/a	n/a	Rate of second band
39.1.10	$band2_threshold		n/a	n/a	Second tax threshold
39.1.11	$band3_rate		n/a	n/a	Rate of third band
39.1.12	$band3_amt		n/a	n/a	Second lump-sum tax
39.1.13	$band3_threshold		n/a	n/a	Third tax threshold
39.1.14	$band4_rate		n/a	n/a	Rate of fourth band in 2001 / Top marginal tax rate in 2007 - 2008
39.1.15	$band4_amt		n/a	n/a	Third lump-sum tax
39.1.16	$band4_threshold		n/a	n/a	Fourth tax threshold in 2001
39.1.17	$band5_rate		n/a	n/a	Top marginal tax rate in 2001
39.1.18	$flat_rate		0.1	0.15	Flat tax rate from 2008 onwards
39.1.19	$flat_zeroband	1	n/a	3200#y	Zero tax band

- Third, modify the calculation of the flat tax in the function *SchedCalc*, by adding a new parameter *Band_LowLim* with a group number 1. In the reform BG_2021_reform, type in the new constant *$flat_zeroband* against *Band_LowLim*. Thus, the tax rate of 15% is levied on taxable income equal or above the zero tax band of BGN 3,200 per year. See Figure 5a.3.

Figure 5a.3: Modifying the calculaton on *SchedCalc*, including the new *$flat_zeroband*

	Policy	Grp/No	BG_2021	BG_2021_reform	Comment
39	**tin_bg**		**on**	**on**	**TAX: income tax (данък общ**
39.1	DefConst		**on**	**on**	Parameters used in the income tax policy - defined as constants
39.2	BenCalc		**on**	**on**	child deduction (данъчно облекчение за деца)
39.3	ArithOp		**on**	**on**	tax deduction for private pension contributions (данъчно облекчение за частни пенсионни вноски)
39.4	ArithOp		**on**	**on**	tax deduction for income from rent (данъчно облекчение за доходи от наем)
39.5	ArithOp		**on**	**on**	tax deduction for income from freelancers (нормативно признати разходи за дейността)
39.6	BenCalc		**on**	**on**	tax deduction for disability (данъчно облекчение за лица с намалена работоспособност)
39.7	ArithOp		**on**	**on**	The sum of all tax deductions
39.8	ArithOp		**on**	**on**	Tax base = taxable income minus tax deductions
39.9	SchedCalc		n/a	n/a	Progressive income tax in 2007 - applying the marginal tax rates
39.10	SchedCalc		**on**	**on**	since 2008: flat income tax
39.10.1	base		tintb_s	tintb_s	tax base
39.10.2	Band_LowLim	1	n/a	$flat_zeroband	Zero tax band
39.10.3	band_rate	1	$flat_rate	$flat_rate	
39.10.4	output_var		tin_s	tin_s	income tax
39.10.5	TAX_UNIT		tu_individual...	tu_individual_bg	
39.11	Elig		n/a	n/a	eligibility for tax exemption for employees in 2014 only
39.12	ArithOp		n/a	n/a	tax exemption for min wage employees in 2014 only

Task 3: Producing the micro-output

Run EUROMOD for the baseline BG_2021 and reform BG_2021_reform systems with *training_data* as dataset.

Task 4: Analyse the results

Analyse the impact on the fiscal budget and the income distribution of the reform, you can use the *Statistics Presenter* with Baseline/Reform option.

Results:

Summary Statistics - Baseline vs Reforms

Results for Bulgaria: BG_2021 vs BG_2021_reform

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Market Incomes and Government Revenue & Expenditure

Yearly, mill., currency as defined in EM output

	BG_2021 (base)	BG_2021_reform	Difference to base
Total market incomes	**269.70**	**269.70**	**0.00**
... income from (self) employment	269.70	269.70	0.00
... other sources	0.00	0.00	0.00
Government revenue through taxes and social insurance contributions	**67.97**	**77.21**	**9.24**
... direct taxes	23.72	32.96	9.24

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Basic Poverty Indices

	Poverty Risk for BG_2021 (base)	Poverty Risk for BG_2021_reform	Difference to base
Population	29.52 %	30.48 %	0.95pp
Children	29.96 %	31.52 %	1.56pp
Working Age	36.78 %	37.78 %	1.00pp
Working Age Economically Active	25.33 %	26.28 %	0.95pp
Elderly	0.00 %	0.00 %	0.00pp
Fixed Poverty Line	2,015.97		

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Basic Inequality Indices

	Gini BG_2021	S80/S20 BG_2021	Gini BG_2021_reform	Diff. Gini BG_2021_reform/BG_2021	S80/S20 BG_2021_reform	Diff. S80/S20 BG_2021_reform/BG_2021
Original Income	0.4788	1.8904	0.4788	0.0000	1.7844	-0.1060
Original Income after Taxes/SIC	0.4831	2.0564	0.4805	-0.0026	1.8998	-0.1566
Original Income incl. Public Pensions after Taxes/SIC	0.5178	21.8576	0.5240	0.0062	22.6716	0.8140
Disposable Income	0.5158	20.8742	0.5220	0.0062	21.6117	0.7375

Exercise 5b: Reforming income tax in EL (using functions *SchedCalc* and *DefConst*)

The aim of this exercise is to learn how to modify and introduce new tax bands, using functions SchedCalc and DefConst.

Steps:

1. Create a new system based on the EL_2020 system and name it EL_2020_e5
2. Define the new tax band of 50,000 per year as a constant in policy tin00_el and add a new tax rate (48%) → function DefConst used to define the income tax parameters
3. Add the new tax band and modify the tax rates in the two SchedCalc functions used to calculate the income tax for employment, pension, self-employment & farming income
4. Run the 2020 system to produce micro-data outputs
5. Analyse the distributional impact of the reform using training_data

Description:

Income taxation is graduated, with progressively higher marginal tax rates applying to higher income brackets.

The table below summarises the income tax bands for tax year 2020:

Taxable income	**Rate of tax**
0 – 10,000	9%
10,001 – 20,000	22%
20,001 – 30,000	28%
30,001 – 40,000	36%
Over 40,000	44%

You are asked to modify the part of the income tax schedule. This change has to be applied both to the general tax base and to the farming income (which is taxed separately).

Keep the income tax schedule the same but on incomes above 50,000 euro apply a tax rate of 48%.

Analyse the impact of the tax reform on the government budget and income distribution.

Hints:

- Analyse the structure of the personal income tax in EL focusing on the policy (tin00_el)
- Perform a reform in EL that involves modifying the income tax, using the SchedCalc and DefConst functions
- Income tax band values and tax rates are defined with a DefConst function (see constants $tin_rate* and $tin_thres*). Define the new tax band as a constant (e.g. $tin_thres_ref) which has a value of 50,000 euro per year.
- Add parameters Band_Rate and Band_UpLim in each of the SchedCalc functions used to calculate the income tax on the general tax base (tintb_s) and on farming income (tinag_s)

Solution and further information

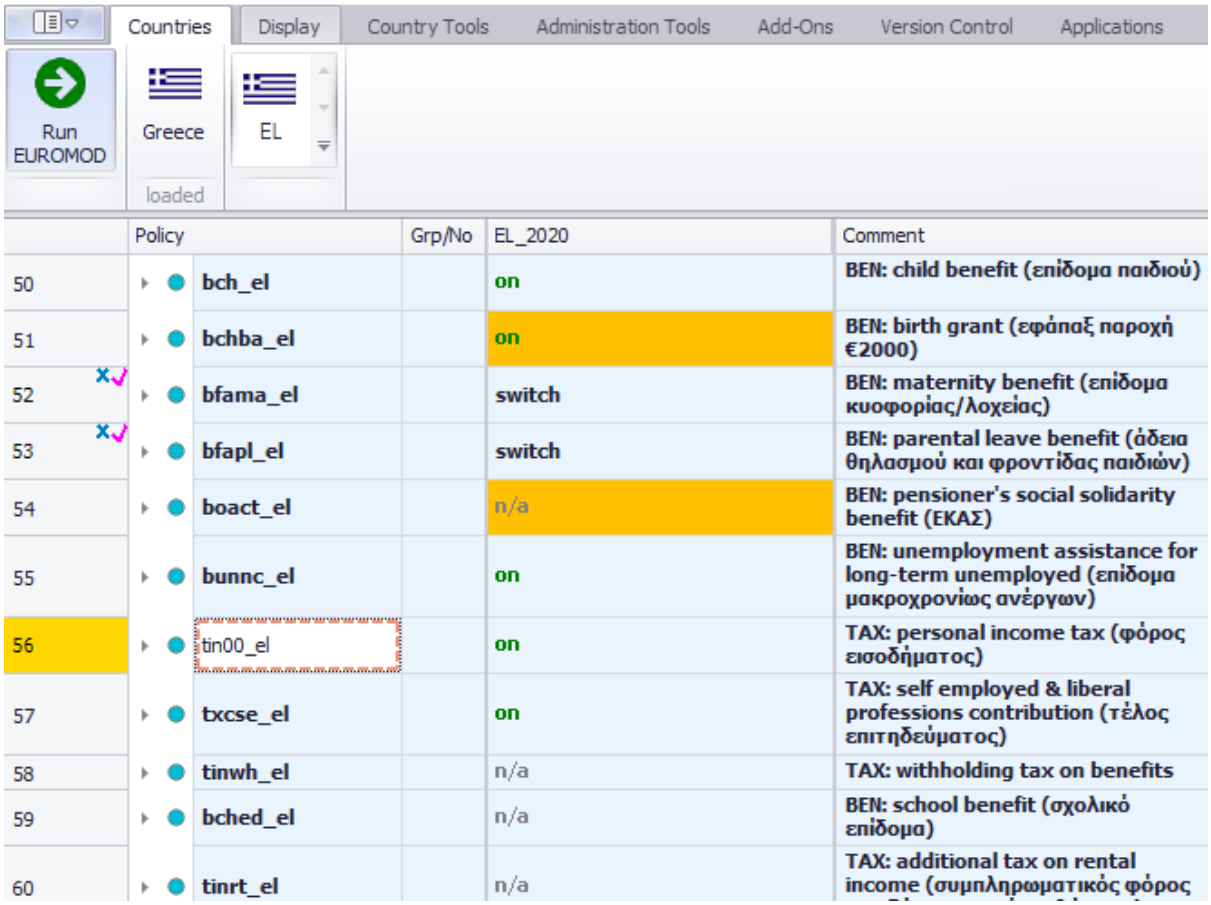
Task 1: Creating a new system

Our first task is to open EL and create a new system using the 2020 system as a base. Give the name EL_2020_e5 to the new system. Hide all systems but these two - 2020 and 2020_e5 (see exercise 1 for explanation).

Task 2: Analysing the current policy

Access the Income Tax policy in the EL (*tin00_el*). Consider year 2020. Hide other years (Figure 5b.1).

Figure 5b.1: Locating the Income Tax policy (*tin00_el*)



	Policy	Grp/No	EL_2020	Comment
50	bch_el		on	BEN: child benefit (επίδομα παιδιού)
51	bchba_el		on	BEN: birth grant (εφάπαξ παροχή €2000)
52	bfama_el		switch	BEN: maternity benefit (επίδομα κυοφορίας/λοχείας)
53	bfapl_el		switch	BEN: parental leave benefit (άδεια θηλασμού και φροντίδας παιδιών)
54	boact_el		n/a	BEN: pensioner's social solidarity benefit (ΕΚΑΣ)
55	bunnc_el		on	BEN: unemployment assistance for long-term unemployed (επίδομα μακροχρονίως ανέργων)
56	tin00_el		on	TAX: personal income tax (φόρος εισοδήματος)
57	txcse_el		on	TAX: self employed & liberal professions contribution (τέλος επιτηδεύματος)
58	tinwh_el		n/a	TAX: withholding tax on benefits
59	bched_el		n/a	BEN: school benefit (σχολικό επίδομα)
60	tinrt_el		n/a	TAX: additional tax on rental income (συμπληρωματικός φόρος

- Explore how personal income tax is calculated (policy *tin00_el*)

Further information can be found in the Country Report (see https://euromod-web.jrc.ec.europa.eu/using-euromod/country-reports/latest).

Locate the *SchedCalc* (Schedule Calculator) functions, used for the calculation of the income tax, by checking the comment column and the implementation. Expand it by clicking on the arrow on the left hand side of the function.

There are two SchedCalc functions: one for the general income tax and one for farming income tax. The general income tax base is defined by *tintb_s* (figure 5b.2), while the farming income tax base is defined by *tintbag_s* (figure 5b.3). For details of the components of the income list *tintb_s* and *tintbag_s*,

see policy tin00_el (you can also use the search option (Ctrl+F) to locate these variables in the model, e.g. search for tinbt_s).

Figure 5b.2: income tax calculation in 2020, general income tax (tin00_el, EL_2018)

56.22	SchedCalc		on	general tax schedule
56.22.1	who_must_...		n/a	
56.22.2	base		tintb_s	
56.22.3	band_uplim	1	$tin_thres1	
56.22.4	band_uplim	2	$tin_thres2	
56.22.5	band_uplim	3	$tin_thres3	
56.22.6	band_uplim	4	$tin_thres4	
56.22.7	band_uplim	5	n/a	
56.22.8	band_uplim	6	n/a	
56.22.9	band_uplim	7	n/a	
56.22.10	band_uplim	8	n/a	
56.22.11	band_rate	1	$tin_rate1	
56.22.12	band_rate	2	$tin_rate2	
56.22.13	band_rate	3	$tin_rate3	
56.22.14	band_rate	4	$tin_rate4	
56.22.15	band_rate	5	$tin_rate5	
56.22.16	band_rate	6	n/a	
56.22.17	band_rate	7	n/a	
56.22.18	band_rate	8	n/a	
56.22.19	band_rate	9	n/a	
56.22.20	output_var		tin00_s	
56.22.21	TAX_UNIT		tu_individual_el	

Figure 5b.3: income tax calculation in 2020, farming income tax (tin00_el, EL_2018)

56.27	SchedCalc		on	tax schedule for farming income
56.27.1	base		tintbag_s	taxed separately
56.27.2	band_uplim	1	$tin_thres1	
56.27.3	band_uplim	2	$tin_thres2	
56.27.4	band_uplim	3	$tin_thres3	
56.27.5	Band_UpLim	4	$tin_thres4	
56.27.6	band_rate	1	$tin_rate1	
56.27.7	band_rate	2	$tin_rate2	
56.27.8	band_rate	3	$tin_rate3	
56.27.9	band_rate	4	$tin_rate4	
56.27.10	Band_Rate	5	$tin_rate5	
56.27.11	LowLim		0	
56.27.12	Result_Var		tinag_s	
56.27.13	Output_Ad...		tin00_s	
56.27.14	TAX_UNIT		tu_individual_el	

The income tax bands and the tax rates are defined as constants – ($tin_thres1, $tin_thres2, $tin_thres3, $tin_thres4, $tin_rate1, $tin_rate2, $tin_rate3, $tin_rate4, $tin_rate5) – using a *DefConst* function in policy *tin00_el* (see Figure 5b.4) (you can also use the search option (Ctrl+F) to locate the constants in the model):

Figure 5b.4: Income tax constants in *tin00_el*

56	tin00_el		on	TAX: personal income tax (φόρος εισοδήματος)
56.1	DefConst		on	define constants
56.1.1	$tc_rate		0.1	rate for tax credits
56.1.2	$yiyit_rate		0.15	rate for interest income
56.1.3	$tax_extra		10000#y	threshold for extra taxable income
56.1.4	$tin_rate1		0.09	rate 1 for tin00_s
56.1.5	$tin_rate2		0.22	rate 2 for tin00_s
56.1.6	$tin_rate3		0.28	rate 3 for tin00_s
56.1.7	$tin_rate4		0.36	rate 4 for tin00_s
56.1.8	$tin_rate5		0.44	rate 5 for tin00_s
56.1.9	$tin_thres1		10000#y	threshold 1 for tin00_s
56.1.10	$tin_thres2		20000#y	threshold 2 for tin00_s
56.1.11	$tin_thres3		30000#y	threshold 3 for tin00_s
56.1.12	$tin_thres4		40000#y	threshold 4 for tin00_s
56.1.13	$tinag_rate		n/a	rate for tinag_s
56.1.14	$tinpr_rate1		0.15	rate 1 for tinpr_s
56.1.15	$tinpr_rate2		0.35	rate 2 for tinpr_s
56.1.16	$tinpr_rate3		0.45	rate 3 for tinpr_s
56.1.17	$tinpr_thres1		12000#y	threshold 1 for tinpr_s
56.1.18	$tinpr_thres2		35000#y	threshold 2 for tinpr_s

To find out the tax band values, you can also hover with the cursor over the constant names within the SchedCalc functions (see Figure 5b.5).

Figure 5b.5: Checking constant values by moving the cursor over the constant names

56.22	SchedCalc		on	general tax schedule
56.22.1	who_must_...		n/a	
56.22.2	base		tintb_s	
56.22.3	band_uplim	1	$tin_thres1	
56.22.4	band_uplim	2	$tin_thres2	
56.22.5	band_uplim	3	$tin_thres3	
56.22.6	band_uplim	4	$tin_thres4	
56.22.7	band_uplim	5	n/a	
56.22.8	band_uplim	6	n/a	
56.22.9	band_uplim	7	n/a	
56.22.10	band_uplim	8	n/a	
56.22.11	band_rate	1	$tin_rate1	
56.22.12	band_rate	2	$tin_rate2	
56.22.13	band_rate	3	$tin_rate3	
56.22.14	band_rate	4	$tin_rate4	
56.22.15	band_rate	5	$tin_rate5	
56.22.16	band_rate	6	n/a	
56.22.17	band_rate	7	n/a	
56.22.18	band_rate	8	n/a	
56.22.19	band_rate	9	n/a	
56.22.20	output_var		tin00_s	
56.22.21	TAX_UNIT		tu_individual_el	

$tin_thres3 = 30000#y

Step 3: Reforming the policy

In the reform system EL_2020_e5, add a new constant (i.e. parameter "placeholder") in the DefConst function where the income tax parameters are defined. The new constant will contain the value of the new income tax band – call it e.g. $tin_thres_ref. This constant should take the value of 50,000 euro per year which would be the default constant value and apply on all individuals. Note the constant should not have any group number (see column Grp/No) as no condition should apply on it (you may need to delete the group number if it has been automatically added).

To define the value of the tax rate, add a new constant (i.e. parameter "placeholder") with a new name (e.g. $tin_rate_ref), a value of 0.48. Once you are done, your implementation should look as in Figure 5b.6.

Figure 5b.6: The new income tax band and tax rate defined as a constant

	Policy	Grp/No	EL_2020	EL_2020_e5	Comment
56	**tin00_el**		**on**	**on**	**TAX: personal income tax (φόρος εισοδήματος)**
56.1	fx DefConst		**on**	**on**	define constants
56.1.1	$tc_rate		0.1	0.1	rate for tax credits
56.1.2	$yiyit_rate		0.15	0.15	rate for interest income
56.1.3	$tax_extra		10000#y	10000#y	threshold for extra taxable income
56.1.4	$tin_rate1		0.09	0.09	rate 1 for tin00_s
56.1.5	$tin_rate2		0.22	0.22	rate 2 for tin00_s
56.1.6	$tin_rate3		0.28	0.28	rate 3 for tin00_s
56.1.7	$tin_rate4		0.36	0.36	rate 4 for tin00_s
56.1.8	$tin_rate5		0.44	0.44	rate 5 for tin00_s
56.1.9	$tin_rate_ref		n/a	0.48	
56.1.10	$tin_thres1		10000#y	10000#y	threshold 1 for tin00_s
56.1.11	$tin_thres2		20000#y	20000#y	threshold 2 for tin00_s
56.1.12	$tin_thres3		30000#y	30000#y	threshold 3 for tin00_s
56.1.13	$tin_thres4		40000#y	40000#y	threshold 4 for tin00_s
56.1.14	$tin_thres_ref		n/a	50000#y	
56.1.15	$tinag_rate		n/a	n/a	rate for tinag_s
56.1.16	$tinpr_rate1		0.15	0.15	rate 1 for tinpr_s
56.1.17	$tinpr_rate2		0.35	0.35	rate 2 for tinpr_s
56.1.18	$tinpr_rate3		0.45	0.45	rate 3 for tinpr_s
56.1.19	$tinpr_thres1		12000#y	12000#y	threshold 1 for tinpr_s
56.1.20	$tinpr_thres2		35000#y	35000#y	threshold 2 for tinpr_s

To modify the income tax schedule of the general income tax, you can add the upper limit ($tin_thres_ref 50,000 euro) for tax rate 0.44 (group 5). Additionally, you have to add the new tax rate ($tin_rate_ref, 48%) for income above 50,000 euro. See figure 5b.7.

Figure 5b.7: Modified income tax schedule for the general income tax

	Policy		Grp/No	EL_2020	EL_2020_e5
56.22	fx	SchedCalc		on	on
56.22.1		who_must_be...		n/a	n/a
56.22.2		base		tintb_s	tintb_s
56.22.3		band_uplim	1	$tin_thres1	$tin_thres1
56.22.4		band_uplim	2	$tin_thres2	$tin_thres2
56.22.5		band_uplim	3	$tin_thres3	$tin_thres3
56.22.6		band_uplim	4	$tin_thres4	$tin_thres4
56.22.7		band_uplim	5	n/a	$tin_thres_ref
56.22.8		band_uplim	6	n/a	n/a
56.22.9		band_uplim	7	n/a	n/a
56.22.10		band_uplim	8	n/a	n/a
56.22.11		band_rate	1	$tin_rate1	$tin_rate1
56.22.12		band_rate	2	$tin_rate2	$tin_rate2
56.22.13		band_rate	3	$tin_rate3	$tin_rate3
56.22.14		band_rate	4	$tin_rate4	$tin_rate4
56.22.15		band_rate	5	$tin_rate5	$tin_rate5
56.22.16		band_rate	6	n/a	$tin_rate_ref
56.22.17		band_rate	7	n/a	n/a
56.22.18		band_rate	8	n/a	n/a
56.22.19		band_rate	9	n/a	n/a
56.22.20		output_var		tin00_s	tin00_s
56.22.21		TAX_UNIT		tu_individual_el	tu_individual_el

To modify the income tax schedule of the farming income tax, add parameters Band_Rate and Band_UpLim in the SchedCalc function used to calculate the income tax. You may need to adjust the group number of the parameters (see column Grp/No), so they apply on the correct parts of the tax schedule (see Figures 5b.8 and 5b.9).

Figure 5b.8: Adding new parameters to the *SchedCalc* function

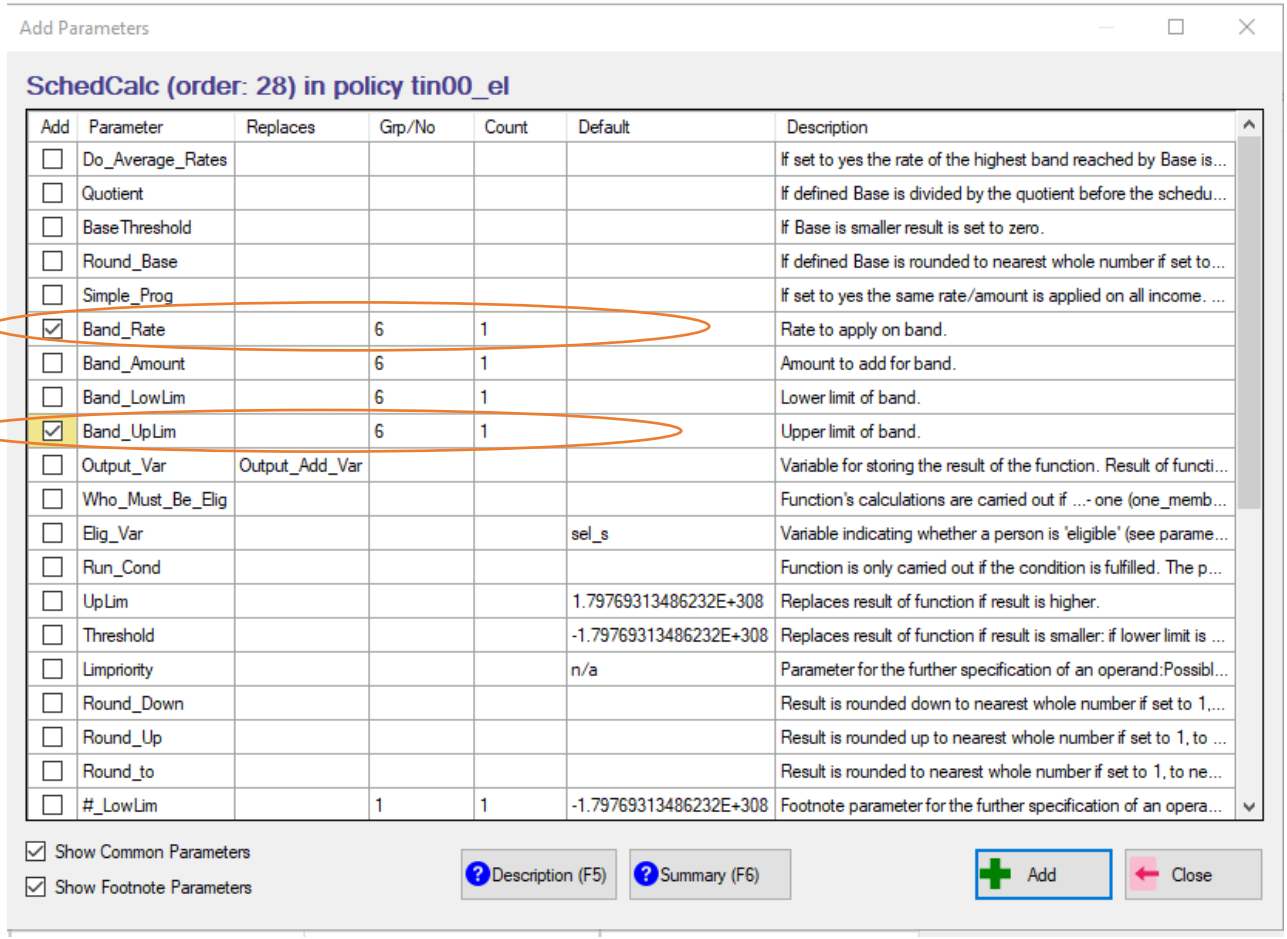


Figure 5b.9: Modified income tax schedule for the farming income tax

56.27	SchedCalc			on	on
56.27.1		base		tintbag_s	tintbag_s
56.27.2		band_uplim	1	$tin_thres1	$tin_thres1
56.27.3		band_uplim	2	$tin_thres2	$tin_thres2
56.27.4		band_uplim	3	$tin_thres3	$tin_thres3
56.27.5		Band_UpLim	4	$tin_thres4	$tin_thres4
56.27.6		Band_UpLim	5	n/a	$tin_thres_ref
56.27.7		band_rate	1	$tin_rate1	$tin_rate1
56.27.8		band_rate	2	$tin_rate2	$tin_rate2
56.27.9		band_rate	3	$tin_rate3	$tin_rate3
56.27.10		band_rate	4	$tin_rate4	$tin_rate4
56.27.11		Band_Rate	5	$tin_rate5	$tin_rate5
56.27.12		Band_Rate	6	n/a	$tin_rate_ref
56.27.13		LowLim		0	0
56.27.14		Result_Var		tinag_s	tinag_s
56.27.15		Output_Add_Var		tin00_s	tin00_s
56.27.16		TAX_UNIT		tu_individual_el	tu_individual_el

Step 4: Produce and analyse micro-data output

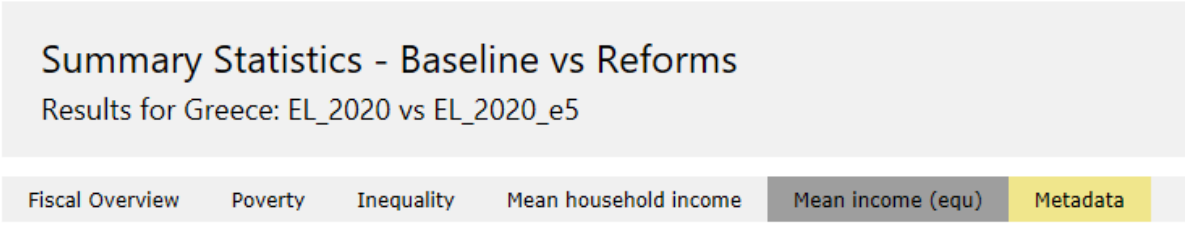
Run EUROMOD to produce micro-output for the baseline (EL_2020) and new system (EL_2020_e5) using the *training_data* dataset. Then use the Statistics Presenter Tool (EUROMOD Statistics – baseline/reform) to analyse the distributional impact of the reform.

***If you need any additional information regarding any of the things mentioned above, you can easily access the EUROMOD Help & Info (the last tab in the ribbon bar), click on the Help and type the word(s) in the search tab and/or you can continue with exercises.

Solution

Statistics Presenter - Baseline vs Reforms report

Results for Greece: EL_2020 vs EL_2020_e5



Mean equivalised income by decile groups

	EL_2020 (base)	EL_2020_e5	Difference to base
Decile 1	133.29	133.29	0.00
Decile 2	647.66	647.66	0.00
Decile 3	823.45	823.45	0.00
Decile 4	1,002.38	1,001.39	-0.99
Decile 5	1,236.04	1,236.04	0.00
Decile 6	1,417.20	1,416.27	-0.94
Decile 7	1,610.12	1,609.70	-0.42
Decile 8	1,836.87	1,834.43	-2.44
Decile 9	2,176.36	2,171.69	-4.67
Decile 10	2,910.19	2,881.36	-28.82
All	**1,377.30**	**1,373.49**	**-3.80**
Poor	444.99	444.99	0.00

Exercise 6: Introducing a benefit cap in Denmark

The aim of this exercise is to learn how to introduce a new policy, create a new income list and use the functions BenCalc and DefIL.

Steps:

1. Create a new system in Denmark based on the 2020 system and name it "dk_2020reform";

2. Create a new policy to limit the total amount of means tested benefits received by households in Denmark;

3. Run EUROMOD to produce micro-outputs for these two systems (DK_2020 and DK_2020reform);

4. Analyse the budgetary and distributive impact of this reform.

Description:

The aim of this exercise is the creation of a new policy that, from 2020, will include a limit on the total amount of means-tested benefit that people can receive in Denmark – a benefit cap. The benefits that will be affected by the cap are *Social Assistance Benefit* (bsa_s), *Housing Benefit* (bho01_s), *Housing Grant* (bho02_s) and *Green Check* (bhtuc_s).

Some working-age families, because of their special circumstances, will not be affected by the benefit cap. This is the case if anyone in the household qualifies for *Disability Benefit* (pdi), *Unemployment Benefit* (bunct_s) and *Survivor's benefit* (psu).

The maximum level of benefits is set to 3,000 Dkr per month for households with 4 or more dependent children and at 2,000 Dkr per month for households with by 3 or fewer dependent children.

Your task is to introduce this new policy – **ben_cap** using functions *DefIl*, *Elig* and *BenCalc* keeping in mind that this policy is a cap on other benefits, i.e. all these benefits need to be calculated first in order to set the cap

Hints:

- Use two *DefIL* functions to define 1) an income list for the benefits that will be capped and 2) another income list for the benefits that prevent the cap being applied.
- Using a *BenCalc* function, calculate the benefit cap as the benefit amount in excess of the cap. Use the optional parameter *LowLim*, equal to 0, to apply a minimum amount on the cap of 0 (i.e. to avoid negative values).
- Save the result of the benefit cap in the variable *brd_s* (b:benefit, rd_reduction, _s: simulated).
- Subtract the benefit cap from the sum of all means-tested benefits, i.e. income list *ils_benmt* in the policy *ILsDef_dk*.
- The household in the Danish model is defined as the tax unit *tu_household_dk*.

Solution and further information

Task 1: Creating a new system

Open Denmark, hide all systems except for dk_2020 and create a new system using the 2020 system as a base. Give the name "dk_2020reform" to the new system.

Task 2: Implementing a new policy

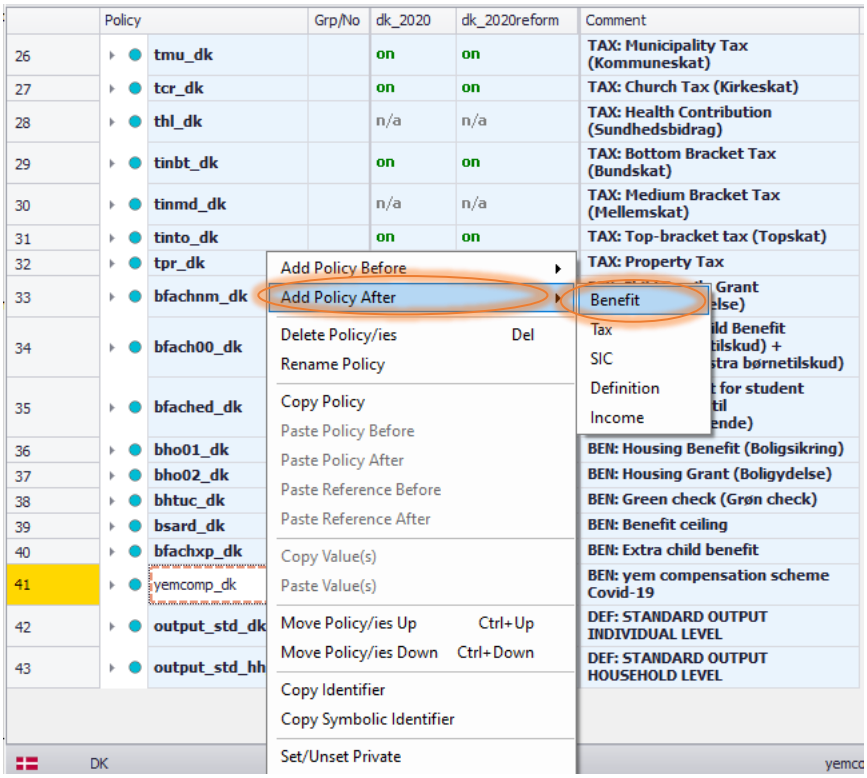
Before implementing a new policy, we should first analyse the tax-benefit system in Denmark – what are the policies, how are they constructed, how are they connected, which income list(s) do they use, which benefits will be affected by the cap, which tax units are used, what are the output variables, etc.

Based on the previous analysis, we can see that the new policy will be a cap on the *Social Assistance Benefit* (bsa_s), *Housing Benefit* (bho01_s), *Housing Grant* (bho02_s) and *Green Check* (bhtuc_s), i.e. all these benefits need to be calculated first in order to set the cap. That means that our benefit cap needs to be at the end of the spine.

Creating a new policy in the spine

As we can see from the policy spine, the last policy is *Benefit ceiling* (bsard_dk). Right-clicking on this policy, will give a list of different options. Choose *Add Policy After* and since we are creating a new benefit, choose *Benefit* (Figure 6.1).

Figure 6.1



When we choose the option to create a new benefit, we will be asked to give it a name. Naming new variables, components, policies, etc. should follow the *EUROMOD Modelling Conventions* document.

Based on those rules, we will name our benefit cap policy **bencap_dk** (Figure 6.2a) and set it as *on* instead of *n/a* in our reform system (dk_2020reform).

Figure 6.2a

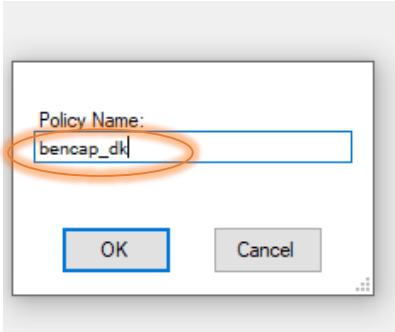
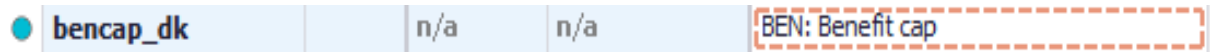


Figure 6.2b



When we create a new policy, function, etc. we should also make a comment (in column *Comment*) so that we have a record what this policy adjustment is for. In our case, we can simply type: 'BEN: *Benefit cap'* (Figure 6.2b)

Generating income lists

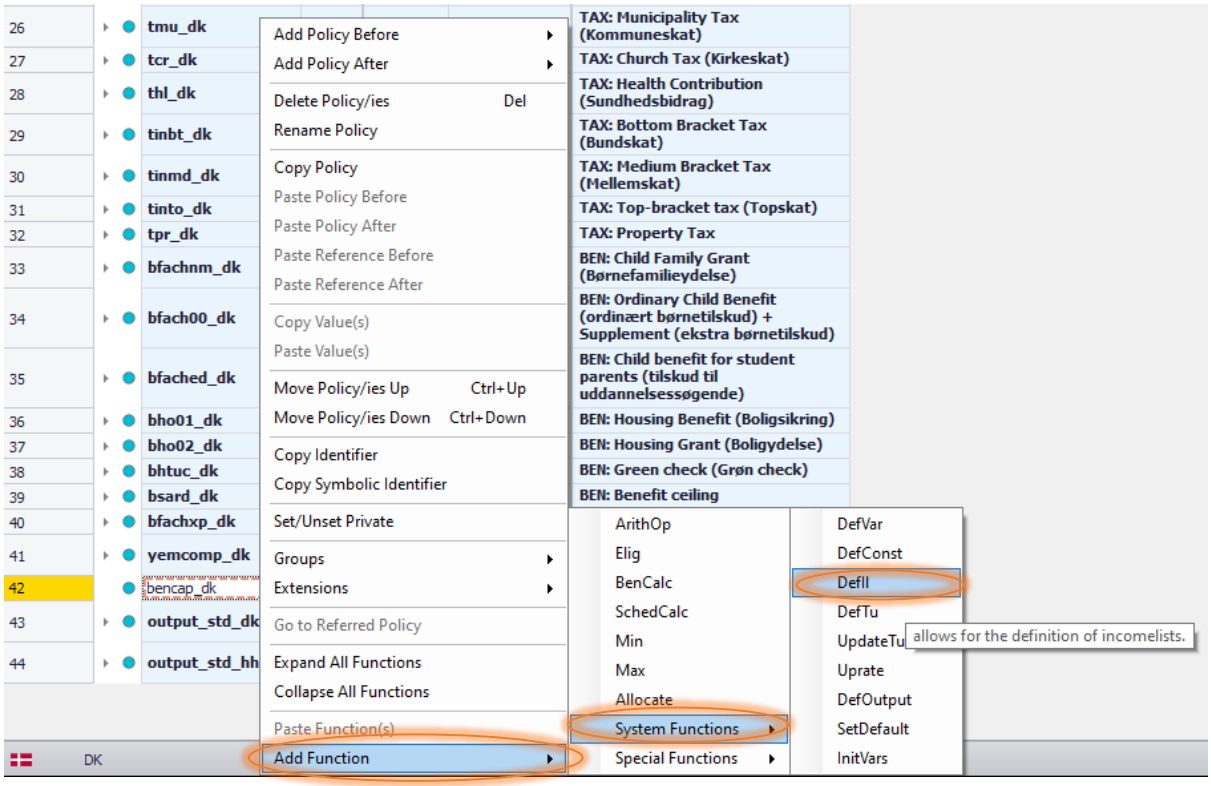
The next step is to generate two new income lists:

- one containing benefits to be capped (we could name it e.g. *il_bencap*)
- one containing benefits that prevent the cap to be applied (we could name it e.g. *il_bencap_dis*).

Income list *il_bencap* should include all the benefits that will be affected by the cap – bsa_s, bho01_s, bho02_s and bhtuc_s.

To define the new income lists, right-click on the *bencap_dk* policy. Select function *DefIl* under *System Functions* (Figure 6.3).

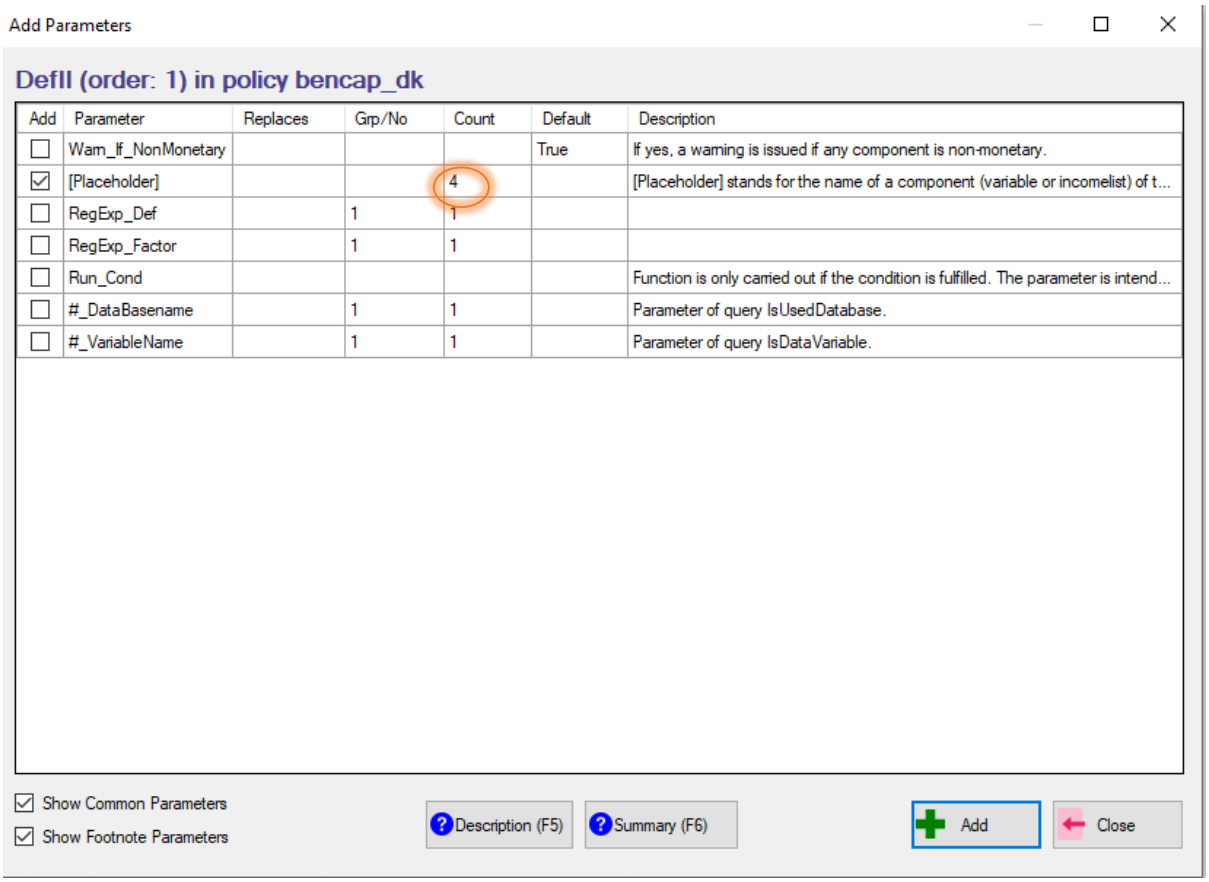
Figure 6.3



A parameter *Name* will be created automatically when we add function *DefIl*. Set *DefIl* to *on* instead of *n/a* and name the income list il_bencap in the reform system (dk_2020reform). Add an explanation in the *Comment* column. Then add all the previously mentioned benefits that will be affected by the cap. To do this, right-click on *Name*, and choose *Show Add Parameter Form*.

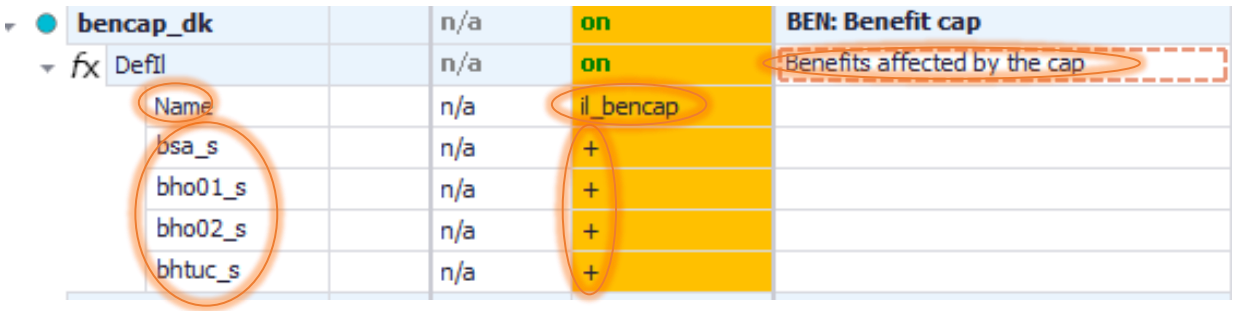
Click on [*Placeholder*] and type "4" in field *Count* (Figure 6.4). This will add 4 empty placeholder rows.

Figure 6.4



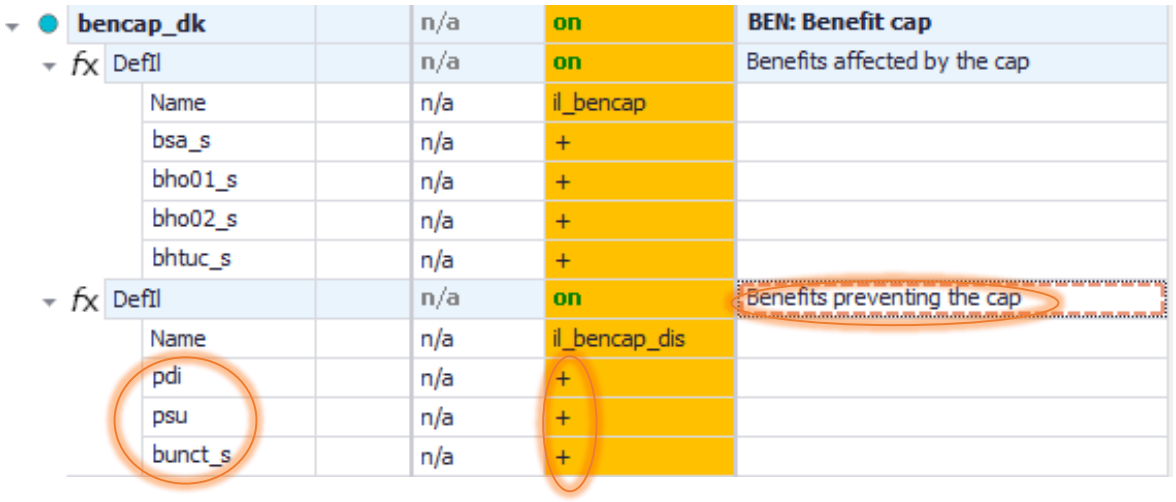
The next step is to rename the fields to include all the benefits that will be affected by the cap – bsa_s, bho01_s, bho02_s and bhtuc_s the and type sign "+" as the il_bencap should be the sum of their values (Figure 6.5).

Figure 6.5



Repeat the same steps to define the other income list - *il_bencap_dis* – which will include all benefits that prevent the cap being applied (pdi, psu, bunct_s) (Figure 6.6).

Figure 6.6

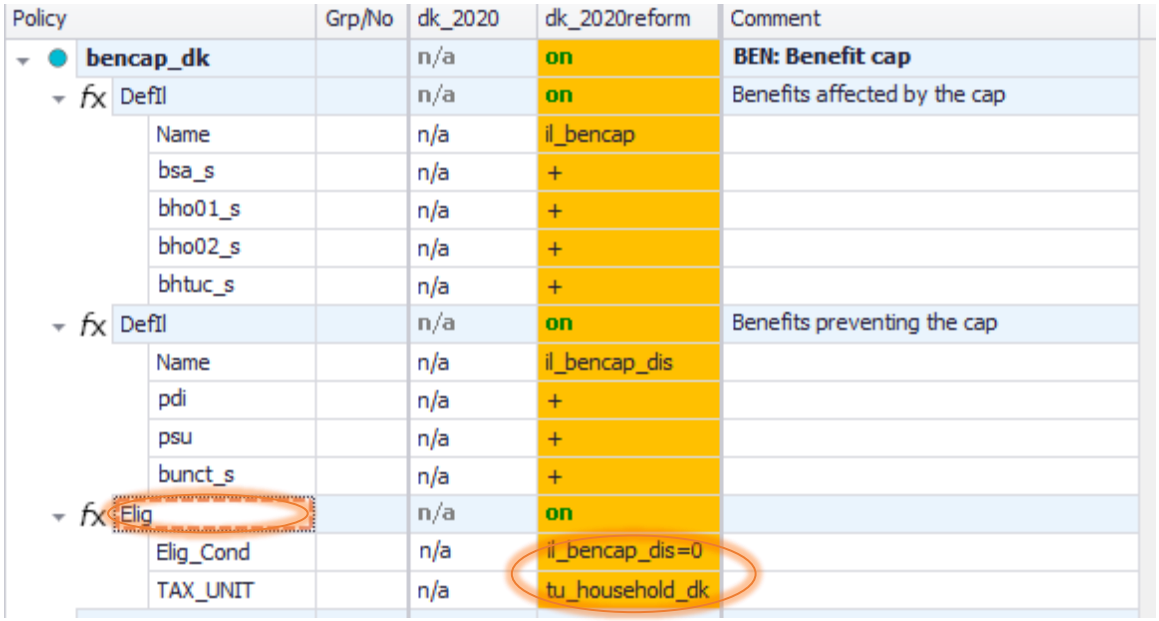


bencap_dk	n/a	on	**BEN: Benefit cap**
DefIl	n/a	on	Benefits affected by the cap
Name	n/a	il_bencap	
bsa_s	n/a	+	
bho01_s	n/a	+	
bho02_s	n/a	+	
bhtuc_s	n/a	+	
DefIl	n/a	on	Benefits preventing the cap
Name	n/a	il_bencap_dis	
pdi	n/a	+	
psu	n/a	+	
bunct_s	n/a	+	

Defining the eligibility condition for the cap

Now we need to define the eligibility condition for the cap: noone in the household should receive any of the benefits listed in the income list *il_bencap_dis* (i.e. these benefits should be zero), by using the function *Elig*. The relevant tax unit is tu_household_dk (Figure 6.7).

Figure 6.7



Policy	Grp/No	dk_2020	dk_2020reform	Comment
bencap_dk		n/a	on	**BEN: Benefit cap**
DefIl		n/a	on	Benefits affected by the cap
Name		n/a	il_bencap	
bsa_s		n/a	+	
bho01_s		n/a	+	
bho02_s		n/a	+	
bhtuc_s		n/a	+	
DefIl		n/a	on	Benefits preventing the cap
Name		n/a	il_bencap_dis	
pdi		n/a	+	
psu		n/a	+	
bunct_s		n/a	+	
Elig		n/a	on	
Elig_Cond		n/a	il_bencap_dis=0	
TAX_UNIT		n/a	tu_household_dk	

Calculating the benefit reduction for each affected household

After setting the eligibility condition, we need to calculate the benefit reduction for each affected household by using function *BenCalc*.

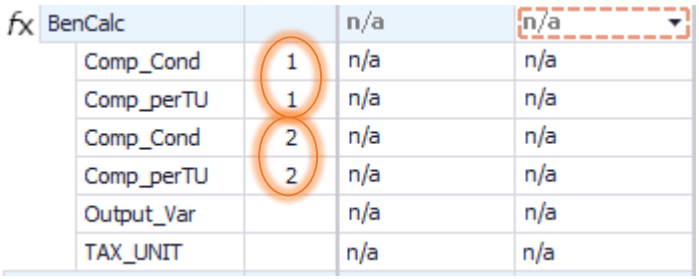
The maximum level of benefits needs to be set to 3,000 Dkr per month for households with 4 or more dependent children and at 2,000 Dkr per month for households with by 3 or less dependent children. This function will have, based on the previous information, two conditions – households with 4 or more and households with less than 4 dependent children.

To do that we add a new B*enCalc* function after the *Elig* function.

When we add function *BenCalc*, we will automatically get four components – *Comp_Cond*, *Comp_perTU*, *Output_Var* and *TAX_UNIT*. Components *Comp_Cond* and *Comp_perTU* are connected by the number from column *Grp/No* so that we know which condition is related to which value. Since we have to create two conditions, we will have to add new pair of components - *Comp_Cond* and *Comp_perTU* by using *Show Add Parameter Form* and connect them with group number "2".

Also, we can change the order of the components inside the policy simply by pressing the cursor, dragging the component to the place where we want it to be and releasing it when we see the blue arrow (Figure 6.8).

Figure 6.8

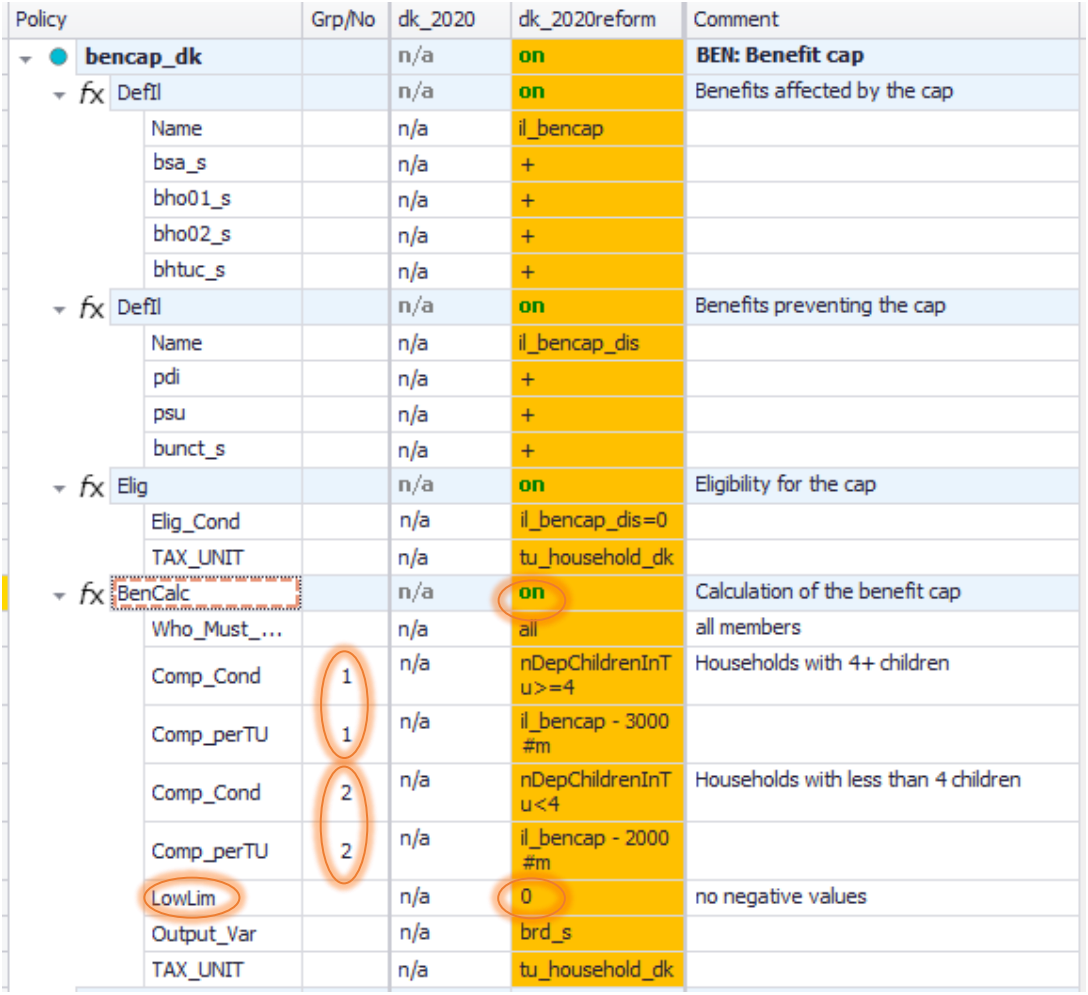


Let's change BenCalc now to implement the reform:

- The first condition refers to households with 4 or more dependent children. Set the value of the first *Comp_Cond* to *nDepChildrenInTu>=4*, where the variable *nDepChildrenInTu* is a query. The benefit cap equals the benefit amount in excess of 3,000 Dkr per month. Thus, set the value of the first *Comp_perTU* to *il_bencap-3000#m*.
- The second condition refers to households with less than 4 children. Set the value of the second *Comp_Cond* to *nDepChildrenInTu<4*. The benefit cap equals the benefit amount in excess of 2,000 Dkr per month. Thus, set the value of the second *Comp_perTU* to *il_bencap-2000#m*.
- It is very important to see if there should be some minimum/maximum values. In our case, we do not want to have negative values, so we will set up the minimum to be zero. In *Show Add Parameter Form* choose component *LowLim* (low limit) and this limit will apply on the result of the function. If our limit is different for different components, choose *#_LowLim* and connect it using the same group number in column *Grp/No*.
- To link the *Elig* and *BenCalc* function, use the parameter *Who_Must_Be_Elig* from *Show Add Parameter Form* box. All members need to fulfil the eligibility condition, so set the value of the parameter to *all*.
- Use the household as the tax unit - tu_household_dk - and call the output variable brd_s (Figure 6.9).

Our reform should now look like Figure 6.9:

Figure 6.9



Policy	Grp/No	dk_2020	dk_2020reform	Comment
bencap_dk		n/a	on	**BEN: Benefit cap**
fx DefIl		n/a	on	Benefits affected by the cap
Name		n/a	il_bencap	
bsa_s		n/a	+	
bho01_s		n/a	+	
bho02_s		n/a	+	
bhtuc_s		n/a	+	
fx DefIl		n/a	on	Benefits preventing the cap
Name		n/a	il_bencap_dis	
pdi		n/a	+	
psu		n/a	+	
bunct_s		n/a	+	
fx Elig		n/a	on	Eligibility for the cap
Elig_Cond		n/a	il_bencap_dis=0	
TAX_UNIT		n/a	tu_household_dk	
fx BenCalc		n/a	on	Calculation of the benefit cap
Who_Must_...		n/a	all	all members
Comp_Cond	1	n/a	nDepChildrenInTu>=4	Households with 4+ children
Comp_perTU	1	n/a	il_bencap - 3000 #m	
Comp_Cond	2	n/a	nDepChildrenInTu<4	Households with less than 4 children
Comp_perTU	2	n/a	il_bencap - 2000 #m	
LowLim		n/a	0	no negative values
Output_Var		n/a	brd_s	
TAX_UNIT		n/a	tu_household_dk	

Finally, before running the model, we need to subtract variable brd_s from the sum of all means-tested benefits – income list *ils_benmt* in policy *ILsDef_dk* – which in turn is part of household disposable income (income list ils_dispy) (Figure 6.10).

Figure 6.10

Policy	Grp/No	dk_2020	dk_2020reform	Comment
random_dk		**on**	**on**	**DEF: assign random numbers**
IlsDef_dk		**on**	**on**	**DEF: STANDARD INCOME CONCEPTS**
DefIl		**on**	**on**	earnings
DefIl		**on**	**on**	original income
DefIl		**on**	**on**	original and replacement income
DefIl		**on**	**on**	simulated benefits
DefIl		**on**	**on**	public pensions
DefIl		**on**	**on**	non means-tested benefits
DefIl		**on**	**on**	means-tested benefits
name		ils_benmt	ils_benmt	
bsa_s		+	+	social assistance benefit (Kontanthjælp; Aktivering af kontanthjælpsmodtagere og flygtninge; Integrationsydelse til flygtninge)
bsaot		+	+	other means-tested social assistance benefits
bho01_s		+	+	housing benefit (Boligsikring)
bho02_s		+	+	housing grant (Boligydelse)
bhtuc_s		+	+	green check (Grøn check)
bfachnm_s		+	+	child family grant (Børnefamilieydelse)
bsard_s		-	-	benefit ceiling
bfachxp_s		+	+	Extra child benefit
brd_s		n/a	-	Amount in excess of the cap

Task 3: Producing the micro-output

Run EUROMOD for the baseline dk_2020 and reform dk_2020reform systems using *training_data* as the dataset.

Task 4: Analyse the results

Use the *Statistics Presenter Tool* (Baseline/Reform option) to analyse the budgetary and distributional impact of the benefit cap.

Results:

Figure 6.11. Market Incomes and Government Revenue & Expenditure

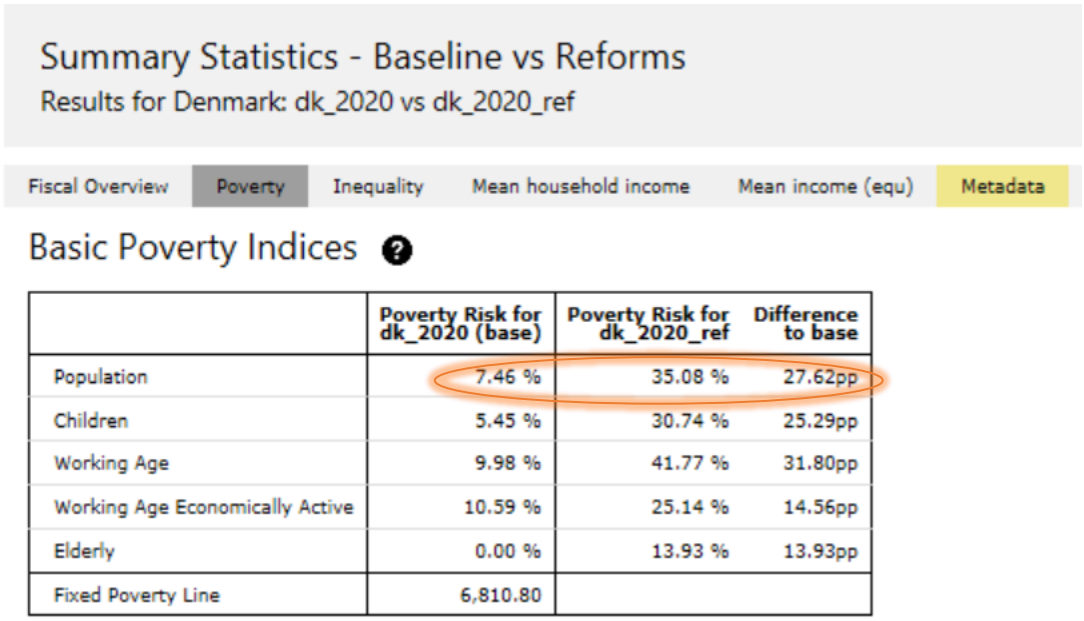
Summary Statistics - Baseline vs Reforms

Results for Denmark: dk_2020 vs dk_2020_ref

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

	dk_2020 (base)	dk_2020_ref	Difference to base
Total market incomes	**1,027.71**	**1,027.71**	**0.00**
... income from (self) employment	1,027.71	1,027.71	0.00
... other sources	0.00	0.00	0.00
Government revenue through taxes and social insurance contributions	**462.71**	**462.71**	**0.00**
... direct taxes	462.71	462.71	0.00
... employee social insurance contributions	0.00	0.00	0.00
... self-employed social insurance contributions	0.00	0.00	0.00
... other social insurance contributions	0.00	0.00	0.00
... employer social insurance contributions (not part of disposable income)	0.00	0.00	0.00
Credited social insurance contributions (not part of disposable income)	**0.00**	**0.00**	**0.00**
Government expenditure on social transfers	**615.76**	**387.06**	**-228.69**
by target group			
... unemployment benefits	0.00	0.00	0.00
... family and education benefits	34.79	34.79	0.00
... social assistance and housing benefits	277.39	277.39	0.00
... pensions, health and disability benefits	303.58	303.58	0.00
... firms	0.00	0.00	0.00
by benefit design			
... means-tested non-pension benefits	310.50	81.81	-228.69
... non-means-tested non-pension benefits	1.68	1.68	0.00
... pensions	303.58	303.58	0.00
... firms subsidies	0.00	0.00	0.00

Figure 6.12. Basic Poverty Indices.



Summary Statistics - Baseline vs Reforms

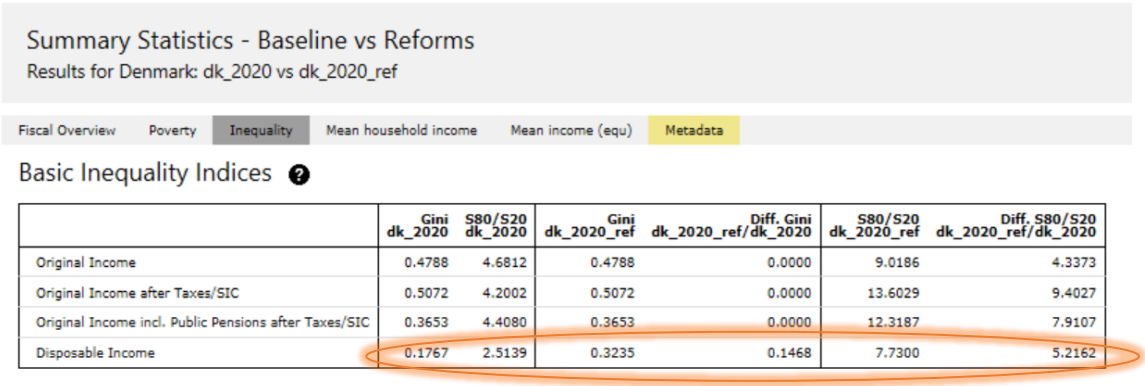
Results for Denmark: dk_2020 vs dk_2020_ref

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Basic Poverty Indices

	Poverty Risk for dk_2020 (base)	Poverty Risk for dk_2020_ref	Difference to base
Population	7.46 %	35.08 %	27.62pp
Children	5.45 %	30.74 %	25.29pp
Working Age	9.98 %	41.77 %	31.80pp
Working Age Economically Active	10.59 %	25.14 %	14.56pp
Elderly	0.00 %	13.93 %	13.93pp
Fixed Poverty Line	6,810.80		

Figure 6.13. Basic Inequality Indices



Summary Statistics - Baseline vs Reforms

Results for Denmark: dk_2020 vs dk_2020_ref

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Basic Inequality Indices

	Gini dk_2020	S80/S20 dk_2020	Gini dk_2020_ref	Diff. Gini dk_2020_ref/dk_2020	S80/S20 dk_2020_ref	Diff. S80/S20 dk_2020_ref/dk_2020
Original Income	0.4788	4.6812	0.4788	0.0000	9.0186	4.3373
Original Income after Taxes/SIC	0.5072	4.2002	0.5072	0.0000	13.6029	9.4027
Original Income incl. Public Pensions after Taxes/SIC	0.3653	4.4080	0.3653	0.0000	12.3187	7.9107
Disposable Income	0.1767	2.5139	0.3235	0.1468	7.7300	5.2162

Exercise 7: Reforming PT Child Benefit by narrowing its age condition

The aim of this exercise is to learn how to use the function DefTU.

Steps:

1. Create a new system in Portugal based on 2020 system and name it PT_2020refTU;

2. Reform the tax unit used to calculate the Family Benefit so that families with children aged 14+ no longer receive this benefit;

3. Run EUROMOD to produce micro-outputs for these two systems (2020 and 2020refTU);

4. Analyse the results.

Description:

The aim of this exercise is to limit the provision of the Child benefit in 2020 only to families with children aged up to 13, i.e. families with children aged 14+ will lose the benefit in the reform scenario. The idea of making the payments age-related is being studied as part of a Government shake-up.

Currently the benefit is paid up to the age of 16 or until 24 if a child is in full-time education. The Government poverty advisers have suggested that there should be a lower age limit for the child. Extra money could be given in the child's early years, and then taken away by the age of 14.

Your task is to implement the reform by creating and using a new Tax Unit.

Solution and further information

Task 1: Creating a new system

Our first task is to open the PT model and create a new system using the 2020 system as a base. Give the name PT_2020refTU to the new system.

Task 2: Implementing a new policy

Before implementing a new policy, we should first identify and analyse the policy where the 'Child Benefit' is implemented in the PT for system year 2020 - how the benefit works, which tax unit is used for the calculation, how is the tax unit defined, etc.

Our next step is, to create a new Tax Unit: Add a *DefTu* function after the function which defines tu_bch_fa_pt.

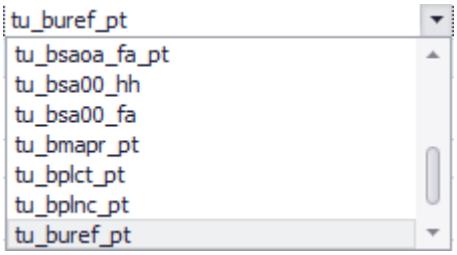
The new tax unit should only identify children under 14 as dependents. The easiest way to do this is by using the relevant tax unit (tu_bch_fa_pt) in the policy *TuDef_PT* as a starting point. Name the new tax unit as **tu_buref_pt** (Figure 7.1) and add the appropriate parameters and their values as shown. Remember to set the new *DefTu* to 'on'.

Figure 7.1

	Policy	Grp/No	PT_2020	PT_2020refTU	Comment
29	**bpInc_pt**		**switch**	**switch**	**BEN: Parental social allowance (Subsídio social parental)**
30	**bch_pt**		**on**	**on**	**BEN: Family benefit (Abono de família para crianças e jovens)**
30.1	fx DefConst		**on**	**on**	Defines the benefit amounts for every bracket
30.2	fx DefVar		**on**	**on**	Intermediate variables
30.3	fx DefTu		**on**	**on**	tu_bch_fa_pt: tu for child benefit
30.3.1	Name		tu_bch_fa_pt	tu_bch_fa_pt	tu for child benefit
30.3.2	Type		SUBGROUP	SUBGROUP	
30.3.3	Members		Partner & OwnDepChild & LooseDepChild	Partner & OwnDepChild & LooseDepChild	
30.3.4	DepChildCond		!IsParent & idpartner=0 & ils_earns#1 = 0 & (dag<=16 | (dag>=17 & dag<=18 & dec>=2) | (dag>=19 & dag<=20 & dec>=3) | (dag>=21 & dag<=24 & dec>=6) | (dag>=16 & dag<=24 & ddi>0))	!IsParent & idpartner=0 & ils_earns#1 = 0 & (dag<=16 | (dag>=17 & dag<=18 & dec>=2) | (dag>=19 & dag<=20 & dec>=3) | (dag>=21 & dag<=24 & dec>=6) | (dag>=16 & dag<=24 & ddi>0))	Not above 16. It may extend until 24 (not above...) under certain conditions:; a) 17 or 18: need to attend primary education (1st to 6th grade) or greater;; b) 19 or 20: need to attend secondary education (7th to 12th grade) or greater;; c) Until 24: need to attend superior school or greater;; d) Also until 24 if disabled children and receiving disability allowances (not simulated).;
30.3.5	#_level	1	tu_individual_pt	tu_individual_pt	
30.3.6	AssignDepC...		yes	yes	Make sure that dependet children who are parents are not separated from their children
30.3.7	LoneParent...		Default & !IsMarried	Default & !IsMarried	
30.4	fx DefTu		n/a	**on**	Reform: Benefit Unit
30.4.1	Name		n/a	tu_buref_pt	
30.4.2	Type		n/a	SUBGROUP	
30.4.3	Members		n/a	Partner & OwnDepChild & LooseDepChild	
30.4.4	DepChildCond		n/a	!IsParent & idpartner=0 & ils_earns#1 = 0 & (dag<14)	
30.4.5	#_Level	1	n/a	tu_individual_pt	
30.4.6	AssignDepC...		n/a	yes	
30.4.7	LoneParent...		n/a	Default & !IsMarried	
30.5	fx DefIl		**on**	**on**	Tested income

Now, in the reform system, we need to modify the tax unit used for the calculation of the child benefit by using the new one (Figure 7.2). Locate the *bch_pt* policy and modify the *TAX_UNIT* as in Figure 7.3.

Figure 7.2



Except for the TaxUnits, nothing else should be changed in the policy (Figure 7.3).

Figure 7.3

	Policy		Grp/No	PT_2020	PT_2020refTU	Comment
30.4	DefTu			n/a	on	Reform: Benefit Unit
30.5	DefIl			on	on	Tested income
30.6	ArithOp			on	on	Equivalised tested income
30.6.1		formula		il_bch / (nDepChildrenInTu + 1)	il_bch / (nDepChildrenInTu + 1)	
30.6.2		output_var		i_bch_eqinc	i_bch_eqinc	equivalised income
30.6.3		TAX_UNIT		tu_bch_fa_pt	tu_buref_pt	tu_bch_fa_pt: tu for child benefit
30.7	BenCalc			on	on	Income brackets
30.7.1		comp_cond	1	i_bch_eqinc <= ($SSI*14/12 * 0.5)	i_bch_eqinc <= ($SSI*14/12 * 0.5)	1) < 50% of minimum wage/SSI (14 pays per year)
30.7.2		comp_perTU	1	1	1	
30.7.3		comp_cond	2	i_bch_eqinc > ($SSI*14/12 * 0.5) & i_bch_eqinc <= ($SSI*14/12 * 1)	i_bch_eqinc > ($SSI*14/12 * 0.5) & i_bch_eqinc <= ($SSI*14/12 * 1)	2) <100% of minimum wage/SSI (14 pays per year)
30.7.4		comp_perTU	2	2	2	
30.7.5		comp_cond	3	i_bch_eqinc > ($SSI*14/12 * 1) & i_bch_eqinc <= ($SSI*14/12 * 1.5)	i_bch_eqinc > ($SSI*14/12 * 1) & i_bch_eqinc <= ($SSI*14/12 * 1.5)	3) <150% of minimum wage/SSI (14 pays per year)
30.7.6		comp_perTU	3	3	3	
30.7.7		comp_cond	4	i_bch_eqinc > ($SSI*14/12 * 1.5) & i_bch_eqinc <= ($SSI*14/12 * 2.5)	i_bch_eqinc > ($SSI*14/12 * 1.5) & i_bch_eqinc <= ($SSI*14/12 * 2.5)	4) <250% of minimum wage/SSI (14 pays per year)
30.7.8		comp_perTU	4	4	4	
30.7.9		comp_cond	5	i_bch_eqinc > ($SSI*14/12 * 2.5) & i_bch_eqinc <= ($SSI*14/12 * 5)	i_bch_eqinc > ($SSI*14/12 * 2.5) & i_bch_eqinc <= ($SSI*14/12 * 5)	5) <500% of minimum wage/SSI (14 pays per year)
30.7.10		comp_perTU	5	5	5	
30.7.11		output_var		i_bch_bracket	i_bch_bracket	income test bracket
30.7.12		TAX_UNIT		tu_bch_fa_pt	tu_buref_pt	tu for child benefit
30.8	BenCalc			on	on	No. of children aged 1 or 2
30.8.1		comp_cond	1	(dag#1=1|dag#1=2) & IsDepChild	(dag#1=1|dag#1=2) & IsDepChild	assigns the # of children aged 1 or 2 to the tax unit
30.8.2		comp_perElig	1	1	1	
30.8.3		#_level	1	tu_individual_pt	tu_individual_pt	
30.8.4		output_var		n_bch_children12	n_bch_children12	
30.8.5		TAX_UNIT		tu_bch_fa_pt	tu_buref_pt	
30.9	BenCalc			on	on	Benefit for children aged 0
30.9.1		comp_cond	1	i_bch_bracket = 1 & dag#1 < 1	i_bch_bracket = 1 & dag#1 < 1	1st income bracket ; child < 1 years-old
30.9.2		comp_perElig	1	$bch_ben_a1	$bch_ben_a1	
30.9.3		comp_cond	2	i_bch_bracket = 2 & dag#1 < 1	i_bch_bracket = 2 & dag#1 < 1	2nd income bracket ; child < 1 years-old
30.9.4		comp_perElig	2	$bch_ben_a2	$bch_ben_a2	
30.9.5		comp_cond	3	i_bch_bracket = 3 & dag#1 < 1	i_bch_bracket = 3 & dag#1 < 1	3rd income bracket ; child < 1 years-old
30.9.6		comp_perElig	3	$bch_ben_a3	$bch_ben_a3	
30.9.7		comp_cond	4	i_bch_bracket = 4 & dag#1 < 1	i_bch_bracket = 4 & dag#1 < 1	4th income bracket ; child < 1 years-old
30.9.8		comp_perElig	4	$bch_ben_a4	$bch_ben_a4	
30.9.9		comp_cond	5	n/a	n/a	5th income bracket ; child < 1 years-old
30.9.10		comp_perElig	5	n/a	n/a	
30.9.11		#_level	1	tu_individual_pt	tu_individual_pt	
30.9.12		output_var		bch_s	bch_s	
30.9.13		TAX_UNIT		tu_bch_fa_pt	tu_buref_pt	

Task 3: Producing the micro-output

Run EUROMOD for the baseline PT_2020 and reform PT_2020refTU systems using the *training_data* dataset.

Task 4: Analyse the results

Use the *Statistics Presenter* (Baseline/Reform option) to analyse the distributional impact of the reform.

Results:

Table 7a: Government Revenues

Yearly, mill., currency as defined in EM output

Government expenditure on social transfers	**48.89**	**48.68**	**-0.21**
by target group			
... unemployment benefits	0.00	0.00	0.00
... family and education benefits	0.78	0.57	-0.21
... social assistance and housing benefits	1.64	1.64	0.00
... pensions, health and disability benefits	46.47	46.47	0.00
... firms	0.00	0.00	0.00
by benefit design			
... means-tested non-pension benefits	2.42	2.21	-0.21
... non-means-tested non-pension benefits	0.00	0.00	0.00
... pensions	46.47	46.47	0.00
... firms subsidies	0.00	0.00	0.00

Table 7b: Basic Poverty Indices

	Poverty Risk for PT_2020 (base)	**Poverty Risk for PT_2020refTU**	**Difference to base**
Population	24.76 %	25.16 %	0.40pp
Children	22.57 %	23.74 %	1.17pp
Working Age	30.92 %	31.17 %	0.25pp
Working Age Economically Active	19.66 %	19.85 %	0.19pp
Elderly	2.99 %	2.99 %	0.00pp
Fixed Poverty Line	954.11		

Exercise 8: Introducing the Belgian social insurance contributions for pensioners & survival pensioners in Portugal

The aim of this exercise is to learn how to implement a policy swap between two countries.

Steps:

1. Create a new system in Portugal based on the 2019 system and name it PT_2019_E8;

2. Simulate what the impact of introducing a new social insurance contribution for pensioners & survival pensioners in Portugal would be, following the same structure as the one that exists in Belgium;

3. Run EUROMOD to produce micro-outputs for these two systems (PT_2019 and PT_2019_E8);

4. Analyse the results.

Description:

The high cost of nursing homes in Portugal has led the government to consider introducing a new social insurance contribution with the same structure as the one that exists in Belgium. This will then be used to help individuals to pay for the cost of care in old age. Perform this policy swap exercise by replicating, in the 2019 Portuguese system, the social insurance policy for pensioners and survival pensioners of Belgium in place in 2019.

Hints:

The aim of this exercise is to create a social insurance contribution – **tscpe_pt -** for 2019 which will have the same structure as the one in the Belgium model - **tscpe_be**.

- You will need to replace the tax unit **tu_family_be** used in Belgium with the unit **tu_bch_fa_pt** used in Portugal and also make an equivalent change for **tu_individual_be**.
- Instead of the income list **il_pension** used in Belgium you can use the income list **ils_pen** defined for Portugal.
- Remember to include the new contribution in one of the standard income lists (e.g. in the income list for "Other Social Insurance Contributions" **ils_sicot**).

Solution and further information

Task 1: Creating a new system

Our first task is to open the Portuguese model and create a new system using the 2019 system as a base. Give the name PT_2019_E8 to the new system.

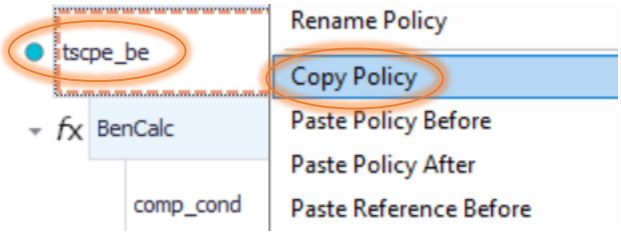
Then, we should open Belgium as well without making any changes.

Task 2: Implementing a new policy

Before implementing a new policy, we should first analyse the tax-benefit system in both countries – Belgium and Portugal. Find policy **tscpe_be** in the Belgium model and see how the social insurance contribution is created – where it is in the policy spine, how it is connected with other policies, what its elements, tax units, etc. are.

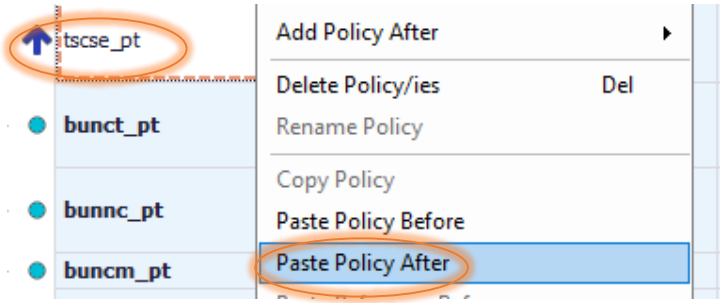
After familiarising yourselves with the different policy spines, start by implementing this contribution in the Portuguese system simply by copying the Belgian *tscpe_be* policy by right- clicking on it and using the option *Copy Policy* from the context (Figure 8.1).

Figure 8.1



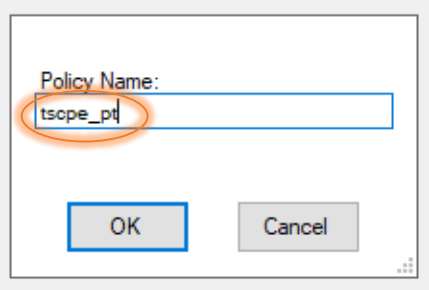
Based on the analyses of both models (Belgium and Portugal), we know that our new social contribution must be created between *Self-employed social insurance contribution (repetition of policy with order 23)* - **tscse_pt** - and *Unemployment insurance (Subsídio de desemprego)* - **bunct_pt** (Figure 8.2).

Figure 8.2



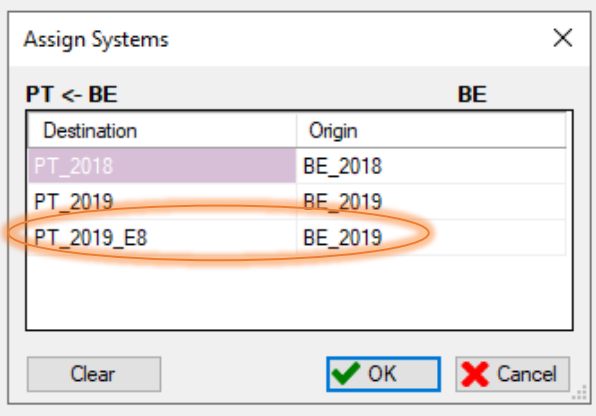
When we chose the option *Paste Policy After* we will be asked to give a (new) name to the policy. In our case, we want to keep the same name, but we will need to change the country acronym - **tscpe_pt** (Figure 8.3).

Figure 8.3



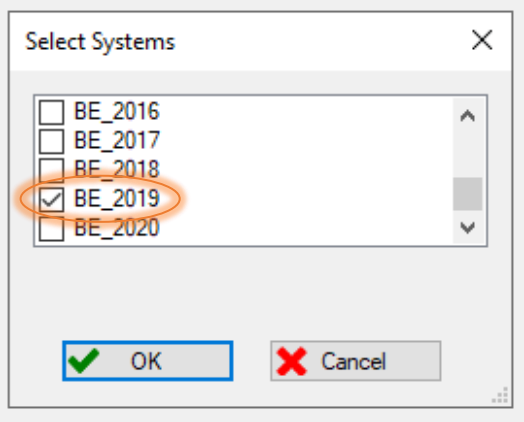
When we use *paste* we should do it in a way to link the BE_2019 and PT_2019_E8 systems. After clicking *ok*, we will get a small window where we will see how the systems are connected. If the systems are connected in a way that we want, we should choose option *ok* (Figure 8.4).

Figure 8.4



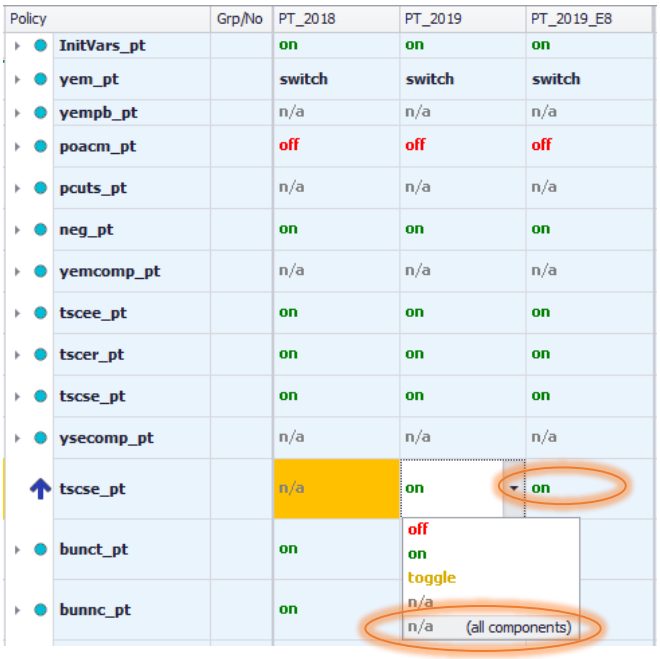
If they are not, then we should point our cursor to the Origin cell and with a left-click, we will get a window with all available systems. Simply by ticking the system which we want, we can create a combination which is the most suitable for what we are trying to do (Figure 8.5).

Figure 8.5



In our case, we are only interested in the PT_2019_E8 system, so we will keep this new policy *on* only for this system, and for all other systems we will set it as *n/a (all components)* (Figure 8.6). Also, we might wish to modify the *Comment* column, as it *currently* contains information related to the Belgium model and data.

Figure 8.6

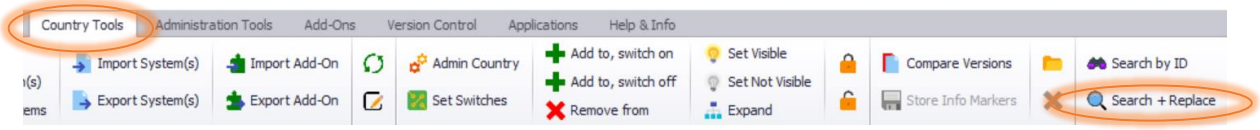


Then, work as follows:

- Make all systems visible. Make sure that the policy is set as n/a for all components for all systems except for PT_2019_E8. Then, you can either set it as n/a for each system individually or to avoid repeating this process for all systems, use hot-key ***Alt+s*** (spread). This option will copy the values to all cells in an observed row. Remember to set it to *on* for PT_2019_E8.

- Next, make all the necessary adjustments in tax units and income lists to make sure this new policy runs and affects disposable income in the correct way. The easiest way to do so is by using the function *Search and Replace* which you can find in the tab *Country Tools* in the ribbon bar – (Figure 8.7).

Figure 8.7



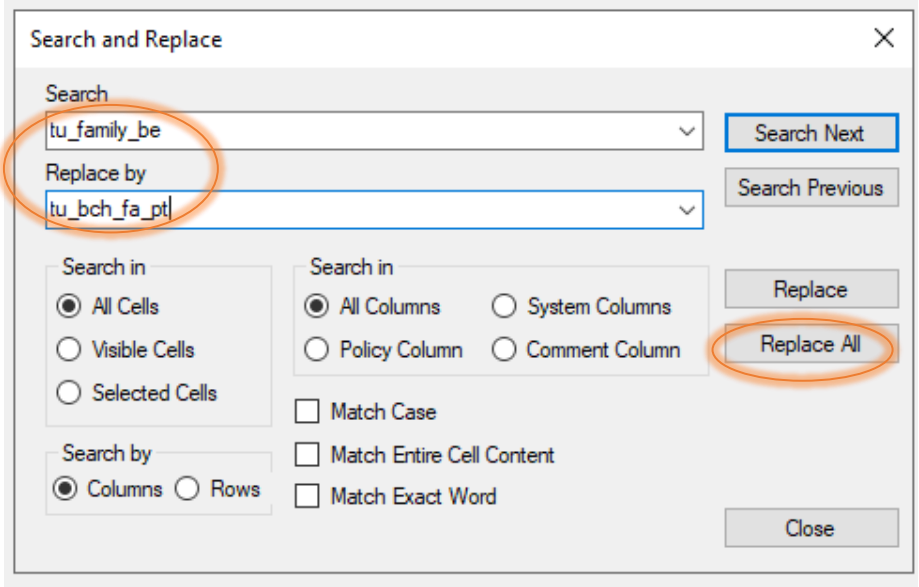
What do we need to *find and replace*?

- After analysing both models, we know that the main thing to do is to change tax units. Instead of the tax unit *family* in Belgium - **tu_family_be** - we should use the unit already defined in Portugal - **tu_bch_fa_pt** (Figure 8.8).

- The next tax unit that needs to be changed is **tu_individual_be**. Instead of this tax unit, we should use **tu_individual_pt**.

- Then, instead of the income list **il_pension** in Belgium you should use an income list already defined in Portugal - **ils_pen**.

Figure 8.8



The new calculated variable will only affect disposable income if it is included in the relevant standardized income lists. In our case it should be included in *Other Social Insurance Contributions* - **ils_sicot** (Figure 8.9).

Figure 8.9

Policy	Grp/No	PT_2019	PT_2019_E8	Comment
fx DefIl		on	on	Self-employed SIC
fx DefIl		on	on	other SIC
Name		ils_sicot	ils_sicot	
txcpe_s		+	+	extraordinary solidarity contribution on pensions
tscpe_s		n/a	+	

Task 3: Producing the micro-output

Run EUROMOD –for the baseline PT_2019 and reform PT_2019_E8 systems using *training_data* as the dataset.

Task 4: Analyse the results

Use the *Statistics Presenter Tool* (Baseline/Reform option) to analyse the distributional impact of the policy swap.

Basic Inequality Indices

	Gini PT_2019	S80/S20 PT_2019	Gini PT_2019_E8	Diff. Gini PT_2019_E8/PT_2019	S80/S20 PT_2019_E8	Diff. S80/S20 PT_2019_E8/PT_2019
Original Income	0.4788	6.6703	0.4788	0.0000	6.8888	0.2185
Original Income after Taxes/SIC	0.4767	6.6827	0.4792	0.0025	6.8666	0.1839
Original Income incl. Public Pensions after Taxes/SIC	0.3478	8.1343	0.3475	-0.0003	8.2031	0.0689
Disposable Income	0.3330	6.6377	0.3326	-0.0004	6.6925	0.0548

Mean equivalised income by decile groups

	PT_2019 (base)	PT_2019_E8	Difference to base
Decile 1	279.81	279.81	0.00
Decile 2	706.92	706.03	-0.89
Decile 3	954.09	950.00	-4.08
Decile 4	1,219.07	1,206.64	-12.44
Decile 5	1,465.12	1,456.07	-9.04
Decile 6	1,701.72	1,686.69	-15.03
Decile 7	1,959.66	1,949.29	-10.37
Decile 8	2,307.89	2,277.13	-30.76
Decile 9	2,741.86	2,702.09	-39.77
Decile 10	3,953.77	3,939.63	-14.14
All	**1,726.65**	**1,713.05**	**-13.60**
Poor	578.30	576.30	-2.00

Exercise 9: Reforming the health tax in Denmark

The aim of this exercise is to learn how to use the function SchedCalc.

Steps:

1. Create a new system in Denmark based on 2018 system and name it dk_2018ref;

2. Modify the Health Tax in order to make the tax allowance non-transferable between partners and increase the progressivity of the tax adding a 15% marginal tax rate for income above 100,000 Dkr per year

3. Run EUROMOD to produce micro-outputs for these two systems (2018 and 2018ref);

4. Analyse the results.

Description:

In the 2018 system, the Danish Health Tax consists of a 5% contribution paid by each individual. The tax base consists of the "Taxable Income" and the tax schedule includes a tax allowance equal to the General Personal Allowance. The unused individual allowance can be transferred between spouses.

The aim of this exercise is implementation and analysis of the distributional effect of a reform consisting of making the unused allowance non-transferable between spouses and including a 15% marginal tax rate for income above 100,000 Dkr per year.

The first step should be the analysis of the structure of the Health Tax in Denmark, focusing on transferability of the tax allowance between partners and tax schedule.

Solution and further information

Task 1: Creating a new system

Our first task is to open Denmark and create a new system using the 2018 system as a base. Give the name dk_2018ref to the new system.

Task 2: Implementing a new policy

Before implementing a new policy, we should first identify and analyse the policy where the 'Health Tax' is implemented in Denmark for system year 2018, with a special focus on the function(s) where the unused part of the allowance is transferred between spouses.

We should observe what the tax base is, where in the policy the unused tax allowance is calculated, where it is transferred, how the tax liability is computed, etc.

Our next step is to modify the policy in order to make the allowance non-transferable between spouses. The way to end the transfer of the unused allowance is simply by switching off the relevant function in the thl_dk policy (Figure 9.1).

Figure 9.1

	Policy	Grp/No	dk_2018	dk_2018ref	Comment
25	**bsa_dk**		**on**	**on**	**BEN: Social Assistance Benefit (Kontanthjælp; Aktivering af kontanthjælpsmodtagere og flygtninge; Integrationsydelse til flygtninge)**
26	**tmu_dk**		**on**	**on**	**TAX: Municipality Tax (Kommuneskat)**
27	**tcr_dk**		**on**	**on**	**TAX: Church Tax (Kirkeskat)**
28	**thl_dk**		**on**	**on**	**TAX: Health Contribution (Sundhedsbidrag)**
28.1	DefVar		**on**	**on**	
28.2	ArithOp		**on**	**on**	
28.3	ArithOp		**on**	**on**	Transferable unused personal allowance calculation
28.4	BenCalc		**on**	**off**	Tax base after the transfer of partner's unused allowance
28.4.1	comp_cond	1	i_thl<0 & IsMarried & GetPartnerIncome#1>0	i_thl<0 & IsMarried & GetPartnerIncome#1>0	If the partner does not use all the personal allowance…
28.4.2	comp_perTU	1	(GetPartnerIncome#1)*(-1)	(GetPartnerIncome#1)*(-1)	The tax base is reduced accordingly
28.4.3	#_income	1	i_thl	i_thl	
28.4.4	output_add_…		ttbhl_s	ttbhl_s	
28.4.5	TAX_UNIT		tu_individual_dk	tu_individual_dk	
28.5	SchedCalc		**on**	**on**	Health Contribution
29	**tinbt_dk**		**on**	**on**	**TAX: Bottom Bracket Tax (Bundskat)**

The second part of our task is to include a 15% marginal tax rate for income above 100,000 Dkr per year. How do we do that?

One of the most used functions in EUROMOD (mainly for progressive taxes) is *SchedCalc*. As we can see from the observed policy, this function usually contains a tax base (*base*), tax bands

(*band_upLim*/*band_lowlim*) and a tax rate (*band_rate*). Also, for fixed amounts, instead of rates we need to use *band_amount* and the component *simple_prog* is used to apply the highest marginal tax rate reached by base on the whole income.

In our case, we need to modify this function by adding new components so as to create the lower limit from which the new rate will be applied. By using *Add Parameter Form*, we can add additional components *band_lowlim* and *band_rate*, thus defining a new tax band (Figure 9.2).

Save your changes.

Figure 9.2

	Policy	Grp/No	dk_2018	dk_2018ref	Comment
25	**bsa_dk**		on	on	**BEN: Social Assistance Benefit (Kontanthjælp; Aktivering af kontanthjælpsmodtagere og flygtninge; Integrationsydelse til flygtninge)**
26	**tmu_dk**		on	on	**TAX: Municipality Tax (Kommuneskat)**
27	**tcr_dk**		on	on	**TAX: Church Tax (Kirkeskat)**
28	**thl_dk**		on	on	**TAX: Health Contribution (Sundhedsbidrag)**
28.1	DefVar		on	on	
28.2	ArithOp		on	on	
28.3	ArithOp		on	on	Transferable unused personal allowance calculation
28.4	BenCalc		on	off	Tax base after the transfer of partner's unused allowance
28.5	SchedCalc		on	on	Health Contribution
28.5.1	base		ttbhl_s	ttbhl_s	The tax base is the taxable income
28.5.2	band_lowlim	1	GenPersAllowance	GenPersAllowance	General Personal Allowance
28.5.3	band_rate	1	$thl	$thl	
28.5.4	Band_LowLim	2	n/a	100000#y	
28.5.5	Band_Rate	2	n/a	0.15	
28.5.6	output_var		thl_s	thl_s	
28.5.7	TAX_UNIT		tu_individual_dk	tu_individual_dk	

Task 3: Producing the micro-output

Run EUROMOD for the baseline DK_2018 and reform DK_2018ref systems using *training_data* as the dataset.

Task 4: Analyse the results

Use the *Statistics Presenter* (Baseline/Reform option) to analyse the fiscal and distributional impact of the reform.

Results:

Table 9a: Government Revenues

Yearly, mill., currency as defined in EM output

Government revenue through taxes and social insurance contributions	458.23	535.87	77.64
... direct taxes	458.23	535.87	77.64
... employee social insurance contributions	0.00	0.00	0.00
... self-employed social insurance contributions	0.00	0.00	0.00
... other social insurance contributions	0.00	0.00	0.00
... employer social insurance contributions (not part of disposable income)	0.00	0.00	0.00

Table 9b: Basic Poverty Indices

	Poverty Risk for dk_2018 (base)	Poverty Risk for dk_2018ref	Difference to base
Population	7.30 %	7.86 %	0.56pp
Children	5.45 %	6.61 %	1.17pp
Working Age	9.73 %	10.22 %	0.50pp
Working Age Economically Active	10.21 %	10.59 %	0.38pp
Elderly	0.00 %	0.00 %	0.00pp
Fixed Poverty Line	6,689.81		

Table 9c: Basic Inequality Indices

	Gini dk_2018	S80/S20 dk_2018	Gini dk_2018ref	Diff. Gini dk_2018ref/dk_2018
Original Income	0.4788	4.7441	0.4788	0.0000
Original Income after Taxes/SIC	0.5042	4.2699	0.5030	-0.0012
Original Income incl. Public Pensions after Taxes/SIC	0.3638	4.4487	0.3605	-0.0033
Disposable Income	0.1773	2.4985	0.1627	-0.0146

Exercise 10: New means-tested income support for families with children in education in Croatia

The aim of this exercise is to refresh existing knowledge about introducing new policies – benefits and taxes.

Steps:

1. Create a new system in Croatia based on the 2021 system and name it HR_2021_E10.

2. Implement the 'Income support to families with children in education'.

3. Run EUROMOD to produce micro-outputs for these two systems (2021 and 2021_E10).

4. Analyse the results.

Description:

The aim of this exercise is to create a new means-tested benefit for 2021 – **bched_hr** - which will be given to families with children in education (any level, not just the compulsory primary and lower secondary levels): 380 HRK per month per entitled dependent child.

The tax unit used to assess the means test will be that of the personal income tax, i.e. **tu_tin_hr**. Consistently, the definition of dependent children will be also the one used in this tax unit.

Educational level is stored in the variable **dec**. A value of 0 indicates that the person is not in education and a value higher than 0 indicates that the person is in education.

The household incomes used in the income test are earnings and public pensions. The income threshold is set to 60,000 HRK per year, increased by 3,800 HRK per year per dependent child.

The financing of this reform will come from introducing an 'Extraordinary contribution' (**txctx_hr**) at the end of the policy spine, where individuals with annual disposable incomes (**ils_dispy**) exceeding 100,000 HRK will pay an extraordinary contribution equal to 1.5% of disposable income.

Solution and further information

Task 1: Creating a new system

Our first task is to open Croatia and create new system using the 2021 system as a base. Give the name HR_2021_E10 to the new system.

Task 2: Implementing new policies

Before implementing a new policy, we should first analyse the tax-benefit system in Croatia – what the relevant policies are, how they are constructed, how they are connected, which income list(s) they use, which child benefits exist, which tax units are used, what variables are outputted, etc.

a) Introducing a new benefit – bched_hr

We should create a new benefit - **bched_hr -** at the end of the policy spine right after the *Compensation for electricity costs* (bhout_hr).

For the simulation of this benefit, you will have to create an income list (the household incomes used in means testing are earnings and public pensions) which will be used for the income test and name it **il_bched** (Figure 10.1).

Figure 10.1

	Policy	Gr...	HR_2021	HR_2021_E10	Comment
33	**bched_hr**		n/a	on	**DEF: income support to families with children in compulsory education**
33.1	fx DefIl		n/a	on	household incomes used in means-testing
33.1.1	Name		n/a	il_bched	
33.1.2	ils_earns		n/a	+	earnings
33.1.3	ils_pen		n/a	+	public pensions

In order to implement the means testing condition, you need to calculate the income threshold. As we said, the income threshold is set to 60,000 HRK per year, increased by 3,800 HRK per year per dependent child. The tax unit which will be used is **tu_tin_hr**. The function *ArithOp* will help you do that. However, you need to use a temporary variable (e.g. *i_bched*) to store the calculation, which needs to be initialised with a *DefVar* function, before *ArithOp*. See both functions in Figure 10.2.

Figure 10.2

	Policy	Gr...	HR_2021	HR_2021_E10	Comment
33	**bched_hr**		n/a	on	**DEF: income support to families with children in compulsory education**
33.1	fx DefIl		n/a	on	household incomes used in means-testing
33.2	fx DefVar		n/a	on	intermediate variable
33.2.1	i_bched	1	n/a	0	
33.3	fx ArithOp		n/a	on	the income threshold
33.3.1	Formula		n/a	60000#y + nDepChildrenInTu*3800#y	60,000 HRK per year, increased by 3,800 HRK per year per dependent child
33.3.2	Output_...		n/a	i_bched	intermediate variable
33.3.3	TAX_UNIT		n/a	tu_tin_hr	...like for personal income tax

The next step is to implement the eligibility conditions and set the value of the benefit. This benefit should be given to the families with children in any level of education. The amount is 380 HRK per month per entitled dependent child. All this can be done in a single *BenCalc* function (Figure 10.3).

Figure 10.3

	Policy	Gr...	HR_2021	HR_2021_E10	Comment
33	**bched_hr**		n/a	on	**DEF: income support to families with children in compulsory education**
33.1	fx DefIl		n/a	on	household incomes used in means-testing
33.2	fx DefVar		n/a	on	intermediate variable
33.3	fx ArithOp		n/a	on	the income threshold
33.4	fx BenCalc		n/a	on	benefit amount
33.4.1	Comp_C...	1	n/a	IsDepChild & dec!=0 & il_bched<i_bched	children in compulsory educatiion with family taxable income less than the income threshold
33.4.2	Comp_p...	1	n/a	380#m	380 HRK per month per dependent child
33.4.3	Output_...		n/a	bched_s	
33.4.4	TAX_UNIT		n/a	tu_tin_hr	...like for personal income tax

The new calculated variable bched_s will only affect disposable income if is included in the relevant standardized income lists. In our case it should be included in means-tested benefits (**ils_benmt**) for the reformed system (HR_2021_E10).

b) Introduce a new tax – txcxt_hr

The second part of this exercise is the introduction of a new tax ('Extraordinary contribution', **txcxt_hr**), where individuals with annual disposable incomes over 100,000 HRK will pay an extraordinary contribution equal to 1.5% of disposable income.

You should create a new policy by using the same steps as for the introduction of a new benefit, doing it right after the newly-created benefit. The reason for this is that our condition to apply the new tax includes disposable income. As you already know, disposable income includes all benefits, so we need to create and simulate all of them before simulating the new tax (Figure 10.4).

This tax should be also added to the list of simulated taxes (**ils_taxsim**), as a way to make it part of disposable income. As you may have realised, there is a circular reference here, since the extraordinary contribution is part of ils_dispy (through ils_taxsim), but ils_dispy is also used to compute the extraordinary contribution. EUROMOD solves this issue by giving value zero to all variables that hasn't been created yet when the income list is used, but will issue a warning. In order to avoid this warning, we may initialise the variable immediately before, as we do in row 34.1.

However, at the end of the spine all income lists are recomputed, so the extraordinary contribution will be finally deducted from ils_dispy, and this is the value that will be part of the output microdata.

Figure 10.4

	Policy	Gr...	HR_2021	HR_2021_E10	Comment
34	**txcxt_hr**		n/a	on	**TAX: extraordinary contribution**
34.1	*fx* ArithOp		n/a	on	initialise result variable as this itself is part of disposable income
34.1.1	Formula		n/a	0	
34.1.2	Output_...		n/a	txcxt_s	
34.1.3	TAX_UNIT		n/a	tu_individual_hr	... at the individual level
34.2	*fx* BenCalc		n/a	on	tax
34.2.1	Comp_C...	1	n/a	ils_dispy>100000#y	individuals with annual disposable incomes over 100,000HRK
34.2.2	Comp_p...	1	n/a	ils_dispy*1.5%	an extraordinary contribution equal to 1.5% of disposable income
34.2.3	Output_...		n/a	txcxt_s	
34.2.4	TAX_UNIT		n/a	tu_individual_hr	... at the individual level

Task 3: Producing the micro-output

Run EUROMOD for the baseline HR_2021 and reform HR_2021_E10 systems using *training_data*.

Task 4: Analyse the results

Use the *Statistics Presenter* (Baseline/Reform) to analyse the distributional impact of the reform.

Results:

Basic Poverty and Inequality Indices

Basic Poverty Indices

	Poverty Risk for HR_2021 (base)	Poverty Risk for HR_2021_E10	Difference to base
Population	28.97 %	29.13 %	0.16pp
Children	29.57 %	29.57 %	0.00pp
Working Age	36.03 %	36.28 %	0.25pp
Working Age Economically Active	24.76 %	24.95 %	0.19pp
Elderly	0.00 %	0.00 %	0.00pp
Fixed Poverty Line	8,815.54		

Basic Inequality Indices

	Gini HR_2021	S80/S20 HR_2021	Gini HR_2021_E10	Diff. Gini HR_2021_E10/HR_2021	S80/S20 HR_2021_E10	Diff. S80/S20 HR_2021_E10/HR_2021
Original Income	0.4788	4.7988	0.4788	0.0000	4.7988	0.0000
Original Income after Taxes/SIC	0.4844	4.7146	0.4857	0.0012	4.6668	-0.0479
Original Income incl. Public Pensions after Taxes/SIC	0.3518	7.1244	0.3518	0.0001	7.0495	-0.0749
Disposable Income	0.3630	8.9063	0.3611	-0.0019	8.7533	-0.1530

Exercise 11: Introducing an In-Work Benefit in Cyprus

The aim of this exercise is to recap your knowledge on introducing new policies – benefits and taxes.

Steps:

1. Create a new system in Cyprus based on the 2020 system and name it CY_2020_E11;

2. Introduce a new means-tested benefit for families with children under 16 years of age, where partners together need to earn at least 30,000 euros per year; use DefTU, Elig and BenCalc functions for that.

3. Run EUROMOD to produce micro-outputs for these two systems (CY_2020 and CY_2020_E11);

4. Analyse the results.

Description:

Cyprus is introducing a new 'In Work Benefit' for parents (**bwkmt_cy)**. The idea is to give a benefit to families with dependent children if both parents or lone parents are low or medium income earners. A child is dependent if it is younger than 16 (below compulsory school age). In order to be eligible, each parent needs to be liable to income tax (**tin_s>0**). The benefit amount depends on the joint earnings (**ils_earns**) of the parents and the number of dependent children. Parents with joint earnings:

- of €30,000 and below €40,000 per year are entitled to receive €600 per child per year;
- of €40,000 and below €50,000 per year are entitled to receive €500 per child per year;
- of €50,000 and below €60,000 per year are entitled to receive €400 per child per year;

The threshold is divided by 2 for lone parents. In addition, this benefit is not restrictive, i.e. receiving this benefit does not affect the amounts of other benefits.

Your task is to implement the new benefit by using functions **DefTU**, **Elig** and **BenCalc**. Remember to include the new benefit in the relevant income lists (**ils_bensim** and **ils_benmt**).

Solution and further information

Task 1: Creating a new system

Our first task is to open Cyprus and create a new system using the 2020 system as a base. Give the name CY_2020_E11 to the new system.

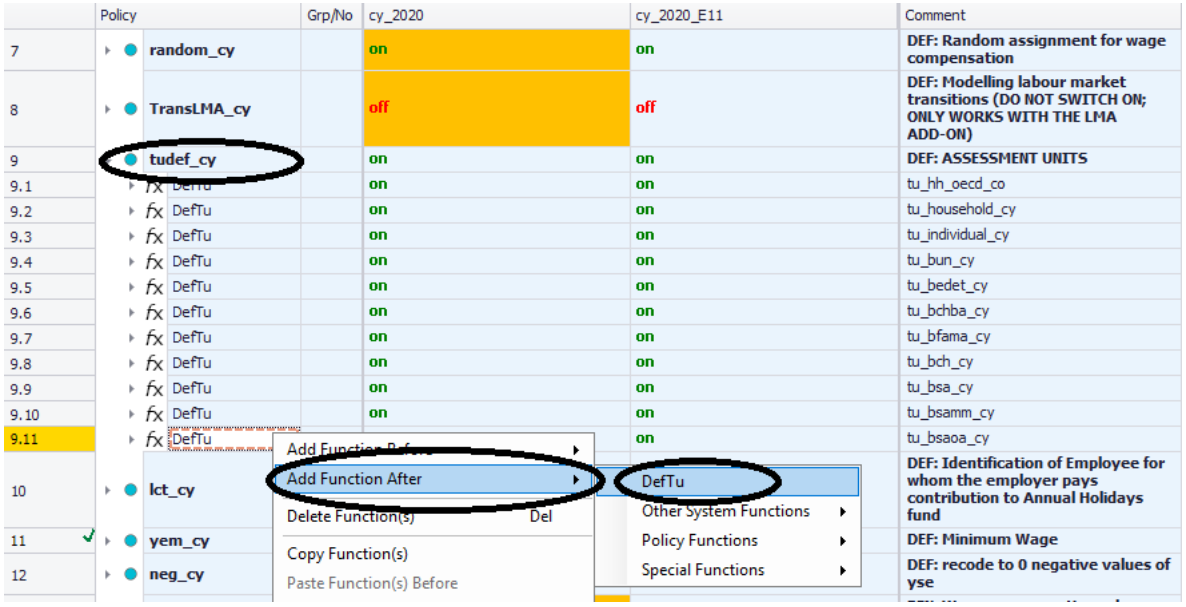
Task 2: Implementing a new policy

Before implementing a new policy, we should analyse the tax-benefit system in Cyprus.

One of the ways of implementing this benefit is to create a new Tax Unit and name it **tu_inwork_cy.** Eligible families are those with children under 16 years of age.

To get an idea of how to create a new Tax Unit you could have a look how other tax units are defined under the policy **tudef_cy**. The first step is to add a function *DefTu* by pointing the cursor at new policy, right-clicking and, among *System Functions*, selecting the one that we need (Figure 11.1).

Figure 11.1



By default, we only get two components - *Type* and *Name*. As we can see from the other tax units, most of them have the component *Members*. Our entitled family should have two parents or a lone parent who is not married with dependent children in the household (both own and loose children).

Also, when we create a new tax unit, we need to specify its type. In our case it will be a *SUBGROUP*. New components could be created/added by using the *Add Parameter Form*.

Another way of doing this is simply to copy-paste an existing tax unit - which should be similar to ours - and then modify it, e.g. **tu_bch_cy** (Figure 11.2). Remember to set the new *tu_inwork_cy* tax unit to 'on'.

Figure 11.2

Policy	Grp/No	cy_2020	cy_2020_E11	Comment
fx DefTu		n/a	on	tu_inwork_cy
Name		n/a	tu_inwork_cy	
Type		n/a	SUBGROUP	
Members		n/a	Partner & OwnDepChild & LooseDepChild	
PartnerCond		n/a	default	
DepChildCond		n/a	default & dag < 16	
LoneParentC...		n/a	default & !IsMarried	

After defining the tax unit, our next step should be to create the new benefit. Based on the information that the new benefit is not restrictive, we create it at the end of the policy spine to make sure that it does not enter the means test in other benefits and we name it **bwkmt_cy**.

The next step is to implement the eligibility conditions and amounts using functions *Elig* and *BenCalc*. Following the Modelling Convention rules, name the output variable as **bwkmt_s**. Please note that we need to do this for couple parents and for lone parent separately (Figure 11.3). Do not forget to use "Output_Add_Var" in the second BenCalc, and set the policy and its functions to on for the reform system.

Figure 11.3

Policy	Grp/No	cy_2020	cy_2020_E11	Comment
bwkmt_cy		n/a	on	**BEN: In-work benefit for parents**
fx Elig		n/a	on	Eligibility for couples
Elig_Cond		n/a	tin_s#1>0 & IsWithPartner#1	...liable to income tax
#_Level	1	n/a	tu_individual_cy	
TAX_UNIT		n/a	tu_inwork_cy	
fx BenCalc		n/a	on	
Who_Must_Be_Elig		n/a	all_adults	Both parents must be eligible
Comp_Cond	1	n/a	ils_earns >= 30000#y & ils_earns < 40000#y	earnings between...
Comp_perTU	1	n/a	nDepChildrenInTu*600#y	get this amount per year per child
Comp_Cond	2	n/a	ils_earns >= 40000#y & ils_earns < 50000#y	earnings between...
Comp_perTU	2	n/a	nDepChildrenInTu*500#y	get this amount per year per child
Comp_Cond	3	n/a	ils_earns >= 50000#y & ils_earns < 60000#y	earnings between...
Comp_perTU	3	n/a	nDepChildrenInTu*400#y	get this amount per year per child
Output_Var		n/a	bwkmt_s	
TAX_UNIT		n/a	tu_inwork_cy	
fx Elig		n/a	on	Eligibility for lone parents
Elig_Cond		n/a	tin_s#1>0 & IsLoneParent#1	...liable to income tax
#_Level	1	n/a	tu_individual_cy	
TAX_UNIT		n/a	tu_inwork_cy	
fx BenCalc		n/a	on	
Who_Must_Be_Elig		n/a	one	Lone parents must be eligible
Comp_Cond	1	n/a	ils_earns >= 15000#y & ils_earns < 20000#y	earnings between...
Comp_perTU	1	n/a	nDepChildrenInTu*600#y	get this amount per year per child
Comp_Cond	2	n/a	ils_earns >= 20000#y & ils_earns < 25000#y	earnings between...
Comp_perTU	2	n/a	nDepChildrenInTu*500#y	get this amount per year per child
Comp_Cond	3	n/a	ils_earns >= 25000#y & ils_earns < 30000#y	earnings between...
Comp_perTU	3	n/a	nDepChildrenInTu*400#y	get this amount per year per child
Output_Add_Var		n/a	bwkmt_s	
TAX_UNIT		n/a	tu_inwork_cy	

The new benefit is a means-tested benefit. Before running the model, we need to add this benefit to the appropriate income lists - **ils_bensim**, **ils_benmt** in our new system (Figure 11.4).

Figure 11.4

Policy		Grp/No	cy_2020	cy_2020_E11	Comment
fx DefIl			on	on	simulated benefits
	name		ils_bensim	ils_bensim	
	bsa_s		n/a	n/a	social assistance benefit (δημόσιο)
	bsamm_s		+	+	guaranteed minimum income benefit (ελάχιστο εγγυημένο εισόδημα)
	bch_s		+	+	child benefit (επίδομα τέκνου)
	bedet_s		+	+	student grant (φοιτητική χορηγία)
	bchlp_s		+	+	single parent child allowance (επίδομα μονογονιού)
	bchba_s		+	+	birth grant (Βοήθημα τοκετού)
	bunct_s		+	+	unemployment benefit from Social Insurance Fund (ανεργιακό επίδομα)
	bmact_s		+	+	Maternity allowance (PARBEN)
	bpact_s		+	+	Paternity allowance (PARBEN)
	bwkmcee_s		+	+	covid-19 special unemployment scheme
	bwkmcse_s		+	+	covid-19 special unemployment scheme for self-employed
	bwkmt_s		n/a	+	In-work benefit

Task 3: Producing the micro-output

Run EUROMOD for the baseline cy_2020 and reform cy_2020_E11 systems using *training_data*.

Task 4: Analyse the results

Use the *Statistics Presenter* to analyse the distributional impact of the reform.

Results:

Table 11a: Basic Poverty Indices

Summary Statistics - Baseline vs Reforms

Results for Cyprus: cy_2020 vs cy_2020_E11

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Basic Poverty Indices

	Poverty Risk for cy_2020 (base)	Poverty Risk for cy_2020_E11	Difference to base
Population	27.54 %	27.54 %	0.00pp
Children	10.51 %	10.51 %	0.00pp
Working Age	20.95 %	20.95 %	0.00pp
Working Age Economically Active	8.70 %	8.70 %	0.00pp
Elderly	75.62 %	75.62 %	0.00pp
Fixed Poverty Line	689.37		

Table 11b: Mean equivalised income by decile groups

Summary Statistics - Baseline vs Reforms

Results for Cyprus: cy_2020 vs cy_2020_E11

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Mean equivalised income by decile groups

	cy_2020 (base)	cy_2020_E11	Difference to base
Decile 1	428.68	428.68	0.00
Decile 2	536.15	536.15	0.00
Decile 3	644.20	644.20	0.00
Decile 4	854.00	854.00	0.00
Decile 5	1,031.00	1,031.00	0.00
Decile 6	1,298.31	1,299.30	0.99
Decile 7	1,547.82	1,551.87	4.05
Decile 8	1,834.30	1,836.02	1.72
Decile 9	2,316.57	2,318.46	1.89
Decile 10	3,296.74	3,296.74	0.00
All	**1,370.24**	**1,371.11**	**0.87**
Poor	497.57	497.57	0.00

Exercise 12: Revenue-neutral reform in Simpleland

The aim of this exercise is to learn about special functions such as Loop, ChangeParam and Totals, and create a revenue-neutral reform.

Steps:

1. Create a new system in Simpleland based on the SL_demo system and name it SL_demo_E12;

2. Set up a function Loop in the beginning of this block. Then, add relevant steps to the end of the loop:

- calculation of total (weighted) disposable income in the baseline and in each iteration,
- calculation of the difference of the two,
- adjustment of income tax rate,
- changing relevant policy parameters;

3. Run EUROMOD to produce micro-outputs for these two systems (SL_demo and SL_demo _E12);

4. Analyse the results.

Description:

Simpleland is an additional hypothetical tax-benefit system, whose purpose is to make learning and teaching EUROMOD easier. This has also served as a "sketch" for the New Member States' prototype models. SL provides a simple version of policies like social insurance contributions (distinguishing employer, employee and self-employed), income tax (single flat rate with a general allowance), universal child benefit and means-tested social assistance and is based on a synthetic database of hypothetical households.

A revenue neutral policy presents changes in the tax laws that result in no change in the amount of revenue coming into the government's coffers. For example, a provision may require individuals to pay less tax, but corporations will pay correspondingly more taxes. The aim of this exercise is to implement a budget neutral reform which doubles the child benefit amount in SL and funds it by raising the income tax rate, and to analyse its distributional effects.

Budget neutrality also takes into account interactions with the social assistance, i.e. the aggregate disposable income should remain the same. The new income tax rate is obtained through a loop which repeats the calculations until converging on a solution with required precision – total (weighted) disposable income should differ by less than 0.01% from its baseline value.

Solution and further information

Task 1: Creating new system

Our first task is to open Simpleland (SL in the list of countries) and create a new system using the SL_demo system as a base. Give the name SL_demo_E12 to the new system.

Task 2: Implementing a new policy

The looping functions allow repeating part (or all) of the tax-benefit calculations. As an example, for calculating marginal tax rates at least part of a policy needs to be calculated twice, once for original income and once for marginally increased income. Function *Loop* allows for such a loop over a group of policies.

The loop is carried out until the number of scheduled iterations is reached and/or the break condition is fulfilled. As certain calculations may depend on the current iteration a variable called *loopcount_loopid* is provided. If for example the identifier (parameter *loop_id*) of the loop for the marginal tax rate calculations is *mtr*, the variable *loopcount_mtr* will take a value of 1 in the first loop and 2 in the second.

This allows for a respective condition to increase income in the second loop. Moreover, if certain policies within the loop should not be repeated, they can be switched off after the first iteration, by using *run_cond*: *loopcount_loopid=1*.

Before implementing a new policy, we should first find relevant existing policies (child benefit, income tax) in the spine and parameters of interest (benefit amount, tax rate). Then we should see which policies need to be run repeatedly to achieve revenue neutrality in terms of (aggregate) disposable incomes. Simpleland has a child benefit (equal to 200 units per month) and a flat income tax with a 20% marginal tax rate.

Our next step is to add a new *Definition* policy, before the child benefit policy (bch_sl), to define the iterative loop and name it **loopdef_sl**. In this policy we need to define two new (intermediate) constants (Add function>System Functions>DefConst):

- adjusted income tax rate (**rate_pit**), and
- initial value of the child benefit (**amt_bch**).

The initial value for the income tax rate and child benefit should be the same as the existing rate and amount in the base system (Figure 12.1).

Figure 12.1: Defining the iterative loop

8	loopdef_sl		n/a	on	DEF: Constants needed for the loop
8.1	fx DefConst		n/a	on	
8.1.1	amt_bch	1	n/a	200#m	
8.1.2	rate_pit	2	n/a	0.2	
9	bch_sl		on	on	BEN: Child Benefit

Then we should replace the current values of the child benefit and the income tax rate with the newly created constants (Figure 12.2).

Figure 12.2: Adding the new constants

9	**bch_sl**		**on**	**on**	**BEN: Child Benefit**
9.1	ArithOp		**on**	**on**	benefit calculation
9.1.1	formula		nDepChildrenInTU*amount#1	nDepChildrenInTU*amount#1	
9.1.2	#_amount	1	200#m	amt_bch	
9.1.3	output_var		bch_s	bch_s	
9.1.4	TAX_UNIT		tu_sben_family_sl	tu_sben_family_sl	There are age limits set for dep. children in TU.
10	**tin_sl**		**on**	**on**	**TAX: Income Tax**
10.1	SchedCalc		**on**	**on**	
10.1.1	base		il_TaxableY	il_TaxableY	
10.1.2	band_lowlim	1	1000#m	1000#m	i.e. general tax exemption/allowance
10.1.3	band_rate	1	0.2	rate_pit	
10.1.4	output_var		tin_s	tin_s	
10.1.5	TAX_UNIT		tu_individual_sl	tu_individual_sl	

The next step is to add another new policy, before the output policy (output_std_sl), to perform additional tasks at the end of each iteration. Name this function **loopend_sl**.

In this policy, we should first use function *Totals* to calculate total disposable income (**agg=ils_dispy**), pre- and post-reform. This requires you to add two new *Totals* functions (Add function>Special Functions>Totals). By default these new functions only include the parameter *TAX_UNIT*. We need to add the following four additional parameters*:*

- *Run_Cond*: to define whether the function is carried out for the pre-reform scenario
  - **(Run_Cond=loopcount_revneutral=1)** for the pre-reform scenario
  - **(Run_Cond=loopcount_revneutral>1)** for the post-reform scenario
- *Agg*: to define the income list
- *Use_Weights*: to calculate totals using household weights
- *Varname_Sum*; storing results in **$pre_ils_dispy** and **$post_ils_dispy**

Figure 12.3: Calculating disposable income pre and post reform

12	**loopend_sl**		n/a	**on**	**DEF: End the loop**
12.1	Totals		n/a	**on**	Calculate pre-reform aggregate disposable income
12.1.1	Run_Cond		n/a	loopcount_revneutral=1	
12.1.2	Agg		n/a	ils_dispy	
12.1.3	Varname_Sum		n/a	pre	
12.1.4	Use_Weights		n/a	yes	
12.1.5	TAX_UNIT		n/a	tu_individual_sl	
12.2	Totals		n/a	**on**	Calculate post-reform aggregate disposable income
12.2.1	Run_Cond		n/a	loopcount_revneutral>1	
12.2.2	Agg		n/a	ils_dispy	
12.2.3	Varname_Sum		n/a	post	
12.2.4	Use_Weights		n/a	yes	
12.2.5	TAX_UNIT		n/a	tu_individual_sl	

Then add a function *Loop* (Add function>Special Functions>Loop) within the **loopend_sl** policy to specify which policies are included in the loop - *First_pol*, *Last_pol* - and what is the exit condition – *BreakCond* (Figure 12.4). However, when we add this function, we will only get *Loop_Id* as the default parameter and we will need to add the other parameters: *First_Pol*, *Last_Pol*, *BreakCond.*

The break condition needs to ensure that the loop does not stop on the first run when the $pre and $post disposable income variables are equal:

loopcount_revneutral>1

and that the calculation is carried out until the difference between the pre-reform and post-reform disposable income is less than 0.01%:

abs(((post_ils_dispy-pre_ils_dispy) / pre_ils_dispy) < 0.01%))

At the same time, we restrict the number of loops to 100 to avoid never ending loops:

loopcount_revneutral = 100

The full break condition is as follows:

(loopcount_revneutral=100) | ((loopcount_revneutral>1) & abs(((post_ils_dispy-pre_ils_dispy)/pre_ils_dispy)<0.01%))

Figure 12.4: Setting he loop parameters

12	**loopend_sl**		n/a	on	**DEF: End the loop**
12.1	Totals		n/a	on	Calculate pre-reform aggregate disposable income
12.2	Totals		n/a	on	Calculate post-reform aggregate disposable income
12.3	Loop		n/a	on	Define loop
12.3.1	Loop_Id		n/a	revneutral	ID
12.3.2	First_Pol		n/a	bch_sl	First policy
12.3.3	Last_Pol		n/a	loopend_sl	Last policy
12.3.4	BreakCond		n/a	(loopcount_revneutral=100) | ((loopcount_revneutral>1) & abs(((post_ils_dispy-pre_ils_dispy)/pre_ils_dispy)<0.01%))	Break condition

Our next step is to adjust model parameters for the child benefit amount and the income tax rate (Figure 12.5). To do that, we first add two *ArithOp* functions within the **loopend_sl** policy. In the first one we set the value of the child benefit (**amt_bch**) to the desired amount. In the second function we adjust the income tax rate **rate_pit** depending on how much the total disposable income differs from the baseline disposable income. For example according to this formula:

new rate = current rate * (current disposable income / baseline disposable income)

Figure 12.5: Adjusting the tax rate and child benefit amount

12	**loopend_sl**		n/a	on	**DEF: End the loop**
12.1	Totals		n/a	on	Calculate pre-reform aggregate disposable income
12.2	Totals		n/a	on	Calculate post-reform aggregate disposable income
12.3	Loop		n/a	on	Define loop
12.4	ArithOp		n/a	on	Adjust tax rate
12.4.1	Run_Cond		n/a	loopcount_revneutral>1	
12.4.2	Formula		n/a	rate_pit*(post_ils_dispy/pre_ils_dispy)	
12.4.3	Output_Var		n/a	rate_pit	
12.4.4	TAX_UNIT		n/a	tu_individual_sl	
12.5	ArithOp		n/a	on	Adjust child benefit amount
12.5.1	Run_Cond		n/a	loopcount_revneutral=1	
12.5.2	Formula		n/a	400#m	
12.5.3	Output_Var		n/a	amt_bch	
12.5.4	TAX_UNIT		n/a	tu_individual_sl	

Finally, we should include the variables in the **output_std_sl** policy to show the income tax rate after the convergence and the number of iterations performed (Figure 12.6).

Figure 12.6: Adding the new variables to the output

13	output_std_sl		on	on
13.1	DefOutput		on	on
13.1.1	file		sl_demo_std	SL_demo_E12_std
13.1.2	Var		n/a	loopcount_revneutral
13.1.3	Var		n/a	amt_bch
13.1.4	Var		n/a	rate_pit
13.1.5	Var		n/a	pre_ils_dispy
13.1.6	Var		n/a	post_ils_dispy

Task 3: Producing the micro-output

Run EUROMOD in order to produce micro-output for the initial and new system - SL_demo and SL_demo_E12 - using dataset '**sl_demo_v4'**. Also see what the new income tax rate is and how many iterations were needed.

Task 4: Analyse the results

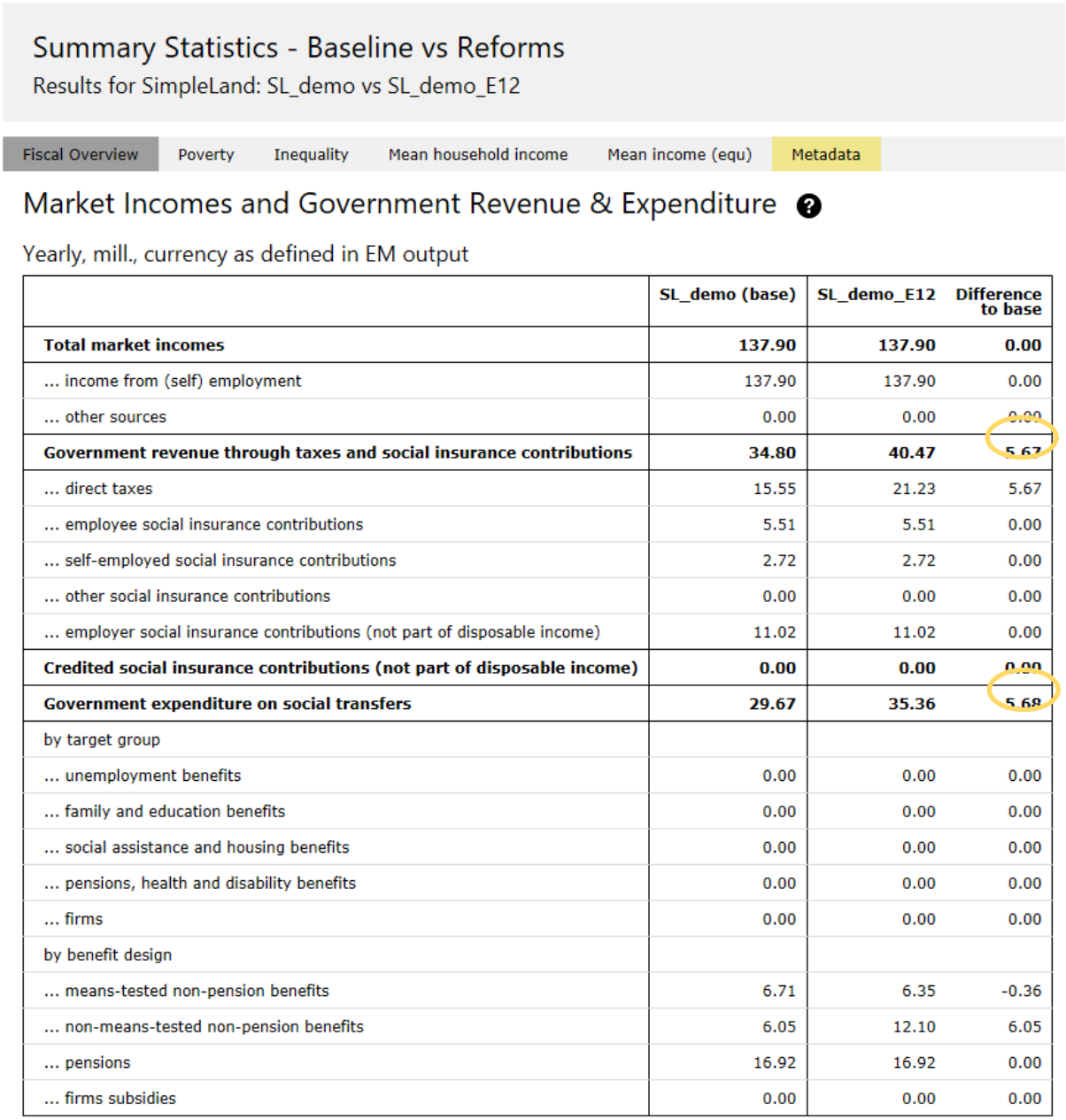
Use the *Statistics Presenter tool* to analyse differences in benefits and disposable income due to the changes.

In the micro data output file (Figure 12.7) we can see that the new tax rate is 25% and 41 iterations were needed to get to the result. As requested, the difference between pre-reform and post-reform disposable income is less than 0.01%. In Figure 12.8 we can see the summary statistics showing the revenue-neutrality of the reform.

Figure 12.7: Output file

loopcount_revneutral	amt_bch	rate_pit	pre_ils_dispy	post_ils_dispy
41	400	.25	11982559	11983596

Figure 12.7: Summary statistics



Summary Statistics - Baseline vs Reforms

Results for SimpleLand: SL_demo vs SL_demo_E12

Fiscal Overview | Poverty | Inequality | Mean household income | Mean income (equ) | Metadata

Market Incomes and Government Revenue & Expenditure

Yearly, mill., currency as defined in EM output

	SL_demo (base)	SL_demo_E12	Difference to base
Total market incomes	**137.90**	**137.90**	**0.00**
... income from (self) employment	137.90	137.90	0.00
... other sources	0.00	0.00	0.00
Government revenue through taxes and social insurance contributions	**34.80**	**40.47**	**5.67**
... direct taxes	15.55	21.23	5.67
... employee social insurance contributions	5.51	5.51	0.00
... self-employed social insurance contributions	2.72	2.72	0.00
... other social insurance contributions	0.00	0.00	0.00
... employer social insurance contributions (not part of disposable income)	11.02	11.02	0.00
Credited social insurance contributions (not part of disposable income)	**0.00**	**0.00**	**0.00**
Government expenditure on social transfers	**29.67**	**35.36**	**5.68**
by target group			
... unemployment benefits	0.00	0.00	0.00
... family and education benefits	0.00	0.00	0.00
... social assistance and housing benefits	0.00	0.00	0.00
... pensions, health and disability benefits	0.00	0.00	0.00
... firms	0.00	0.00	0.00
by benefit design			
... means-tested non-pension benefits	6.71	6.35	-0.36
... non-means-tested non-pension benefits	6.05	12.10	6.05
... pensions	16.92	16.92	0.00
... firms subsidies	0.00	0.00	0.00

Exercise 13: Implementing an EU child benefit using an AddOn

The aim of this exercise is to learn how to build, export and modify an AddOn which will implement a common child benefit scheme across EU countries.

Steps:

1. Create a new system in Simpleland based on SL_demo system and name it bcheu_AddOn;

2. Implement the new child benefit. Call the new policy bcheu and the new benefit bchot_s. Place the new policy before the output policies;

3. Technical step: create a new output policy and switch off the existing output policy using a change parameter function. Make sure that the new output file terminates with _std;

4. Analyse the distributional impact of the benefit in Simpleland;

5. Export the reform system as an AddOn;

6. Analyse the AddOn;

7. Modify the AddOn such that it can be used with the chosen countries;

8. Analyse the results

Description:

Italy and Netherlands have been chosen as pilot countries for the introduction of an EU child benefit aimed at alleviating child poverty in low income households. The scheme will have common rules across countries and, if successful, will then be extended to the remaining EU countries. The benefit rules specify that beneficiaries of the EU benefit will be children aged 14 or younger living in households in which the sum of market incomes (ils_origy) and pensions (ils_pen) amounts to less than 24000 Euros per year. Each child in eligible households is entitled to a benefit of 6000 Euros per year. Analyse the impact of the benefit on child poverty.

Although it would be possible to implement the reform independently in each country, the reform is suited to be implemented through an AddOn given the common features of the scheme across member states. AddOns cannot be built from scratch and users have two possibilities. The first consists in modifying an existing add-on. The second is to implement the changes in a country of choice, and then to export these changes as an add-on. As a further step, the newly created AddOn will have to be modified to be able to run it on other countries. In this exercise we follow the second possibility, and create an AddOn starting from Simpleland.

Solution and further information

Here are the steps at a glance:

Task 1: Create a new system

Task 2: Implement the new policy

We should include the newly created benefit among the components of the standard income lists ils_benmt. This step allows EUROMOD to take into account the EU child benefit when computing disposable income.

Task 3: Create an output policy

Task 4: Run the model and analyse the effect of the reform on Simpleland

Task 5: Export the policy reform as an add-on

Task 6: Analyse the AddOn

Task 7: Modify the AddOn such that it can be used with the chosen countries

Task 8: Analyse the results

Task 1: Creating a new system

Our first task is to open Simpleland and create a new system using the SL_demo system as a base. Give the name bcheu_AddOn to the new system.

Task 2: Implementing a new policy

Our next step is to add a new policy containing the implementation of the EU child benefit. We will place the new policy before **Output_std_sl**. We'll be asked to change the name of the policy bcheu in bcheu_sl, because we are working on Simpleland. In this case we should press "No" because our aim is to create an AddOn to be used in other countries as well.

Figure 13.1

	Policy	Grp/No	SL_demo	bcheu_AddOn	Comment
1	Uprate_sl		on	on	DEF: UPRATING FACTORS
2	ILsDef_sl		on	on	DEF: STANDARD INCOME CONCEPTS
3	ILDef_sl		on	on	DEF: SPECIFIC INCOME CONCEPTS
4	TUDef_sl		on	on	DEF: ASSESSMENT UNITS
5	yem_sl		on	on	DEF: Minimum Wage
6	neg_sl		on	on	DEF: recode negative self-employment income to zero
7	sic_sl		on	on	SIC: Social Insurance Contributions
8	bch_sl		on	on	BEN: Child Benefit
9	tin_sl		on	on	TAX: Income Tax
10	bsa_sl		on	on	BEN: Social Assistance
11	bcheu		n/a	on	BEN:

The policy can be implemented by defining an income list which contains the relevant incomes for the means tested benefit and calculating benefit entitlement through a function *BenCalc*. Assign the name *bchot_s* to the EU child benefit. Remember to calculate the benefit for each eligible child and not for the Tax Unit in total (Comp_perElig).

Figure 13.2

bcheu		n/a	on	**BEN:**
DefIl		n/a	on	Thr sum of market incomes and pensions for this benefit unit
Name		n/a	il_bcheu	
ils_origy		n/a	+	market incomes
ils_pen		n/a	+	pensions
BenCalc		n/a	on	how this benefit is calculated
Comp_Cond	1	n/a	dag <= 14 & il_bcheu < 24000	beneficiaries of the EU benefit will be children aged 14 or younger living in households in which the sum of market incomes (ils_origy) and pensions (ils_pen) amounts to less than 24000 Euros per year.
Comp_perElig	1	n/a	6000#Y	Each child in eligible households is entitled to a benefit of 6000 Euros per year.
Output_Var		n/a	bchot_s	saving calculation to a variable
TAX_UNIT		n/a	tu_household_sl	tax unit is the household

We should also remember to include the newly created benefit among the components of the standard income lists *ils_benmt*. This step allows EUROMOD to take into account the EU child benefit when computing disposable income.

Figure 13.3

ILsDef_sl		on	on	**DEF: STANDARD INCOME CONCEPTS**
DefIl		on	on	earnings
DefIl		on	on	original income
DefIl		on	on	simulated benefits
DefIl		on	on	Pensions
DefIl		on	on	Means-tested benefits
name		ils_benmt	ils_benmt	
bsa_s		+	+	Social Assistance
bchot_s		n/a	+	EU child benefit

Task 3: Technical step – create a new output policy

Copy and paste the existing output policy **output_std_sl.** Paste it before **Output_std_sl**. Again if you are asked, select 'No' as we would like this policy to be applicable across different countries. Call the new policy **output_bcheu**, access it and modify the function *Def_Output* by changing the value of the parameter file in **sl_bcheu_std**. As a second step, introduce a *ChangeParam* function to switch the original output policy off. The reason behind the technical step will be explained in the remainder of the solutions.

Figure 13.4

Task 4: Run the model and analyse the effect of the reform on Simpleland

Using the Statistics Presenter, create statistics using the base/reform option to analyse the distributional effect of the reform using 'sl_demo_v4' as the dataset. Your outputs should look like below.

Market Incomes and Government Revenue & Expenditure

Yearly, mill., currency as defined in EM output

	SL_demo (base)	bcheu_AddOn
Total market incomes	**137.90**	**137.90**
... income from (self) employment	137.90	137.90
... other sources	0.00	0.00
Government revenue through taxes and social insurance contributions	**34.80**	**34.80**
... direct taxes	15.55	15.55
... employee social insurance contributions	5.51	5.51
... self-employed social insurance contributions	2.72	2.72
... other social insurance contributions	0.00	0.00
... employer social insurance contributions (not part of disposable income)	11.02	11.02
Credited social insurance contributions (not part of disposable income)	**0.00**	**0.00**
Government expenditure on social transfers	**29.67**	**31.95**
by target group		
... unemployment benefits	0.00	0.00
... family and education benefits	0.00	0.00
... social assistance and housing benefits	0.00	0.00
... pensions, health and disability benefits	0.00	0.00
... firms	0.00	0.00
by benefit design		
... means-tested non-pension benefits	6.71	8.99
... non-means-tested non-pension benefits	6.05	6.05
... pensions	16.92	16.92
... firms subsidies	0.00	0.00

Basic Poverty Indices

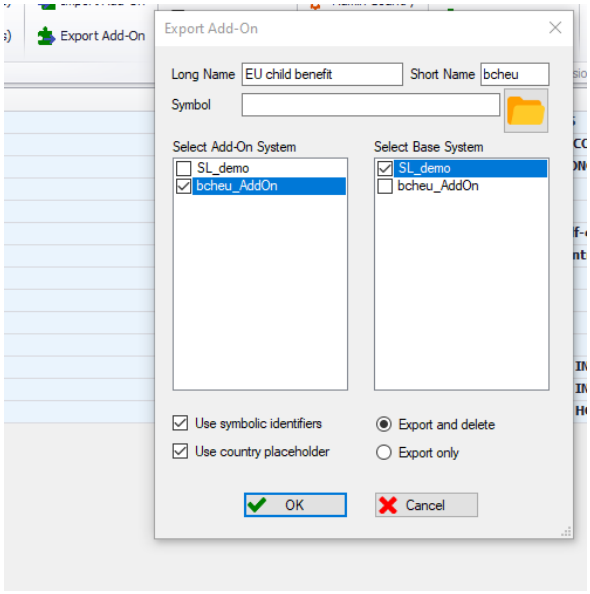
	Poverty Risk for SL_demo (base)	Poverty Risk for bcheu_AddOn	Difference to base
Population	18.33 %	14.84 %	-3.49pp
Children	8.95 %	1.17 %	-7.78pp
Working Age	19.70 %	16.83 %	-2.87pp
Working Age Economically Active	7.75 %	6.05 %	-1.70pp
Elderly	24.88 %	24.38 %	-0.50pp
Fixed Poverty Line	693.12		

Task 5: Export the policy reform as an add-on

We can do that using the Export Add-On button which is placed in the tab Country tools tab. From here you need to:

- fill a long and a short name for our Add-On
- select the Add-On and base systems from the list.
- By ticking 'Use symbolic identifiers', references to changes of specific functions and parameters will be done using the symbolic identifier rather than the cell identifier.
- By ticking 'Use country placeholder', references to Simpleland will be replaced by the placeholder =cc=, making the process of adaption of the AddOn to other countries easier.

Figure 13.5



Click on the Ribbon menu item 'Add-Ons', the *bcheu* icon should be there (Figure 13.6). Click on it to open it.

Figure 13.6

Task 6: Analyse the AddOn

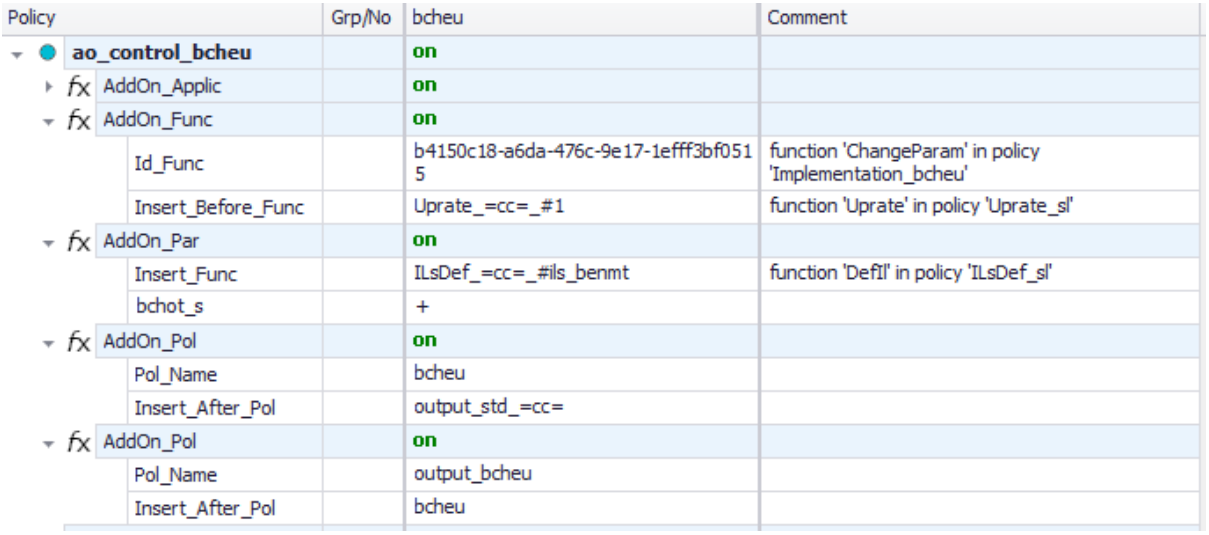
The exported AddOn is composed of four policies, namely **ao_control_bcheu, Implementation_bcheu; bcheu** and **Output_bcheu**. We can find it under the Add-On tab, selecting **bcheu**.

The policy **ao_control_bcheu** contains the following functions:

- *AddOn_Applic*: specifies that the AddOn runs for the system SL_demo
- *AddOn_Func*: contains a unique identifier referring to a function that is created by the AddOn and which will be added before the first function (#1) of the policy Uprate of a given country (note that =cc= is used instead of the country acronym). This function is not needed and can be dropped, the reasons are discussed below.

- The function *AddOn_Par* modifies the income list called **ils_benmt** in each country by including the newly simulated benefit
- The two functions *AddOn_Pol* specify the names of the policies that will be added and their position in the spine.

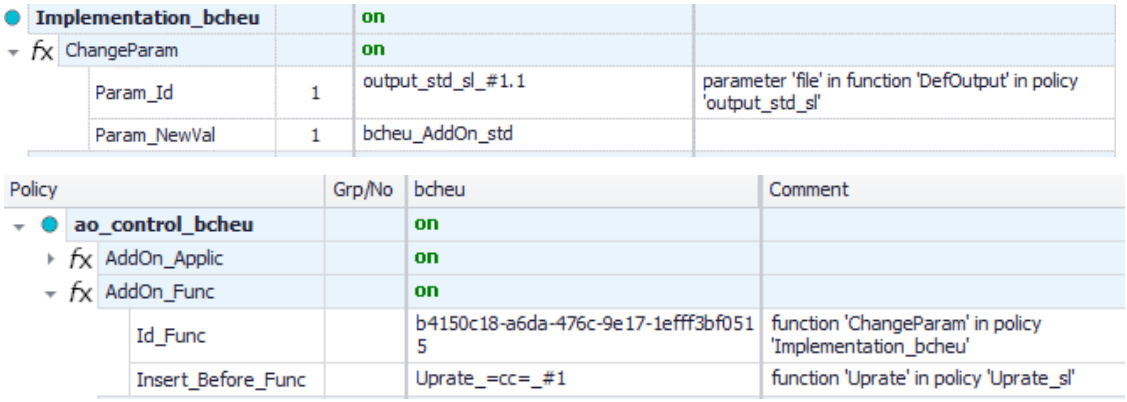
Figure 13.7



Policy	Grp/No	bcheu	Comment
ao_control_bcheu		**on**	
AddOn_Applic		**on**	
AddOn_Func		**on**	
Id_Func		b4150c18-a6da-476c-9e17-1efff3bf0515	function 'ChangeParam' in policy 'Implementation_bcheu'
Insert_Before_Func		Uprate_=cc=_#1	function 'Uprate' in policy 'Uprate_sl'
AddOn_Par		**on**	
Insert_Func		ILsDef_=cc=_#ils_benmt	function 'DefIl' in policy 'ILsDef_sl'
bchot_s		+	
AddOn_Pol		**on**	
Pol_Name		bcheu	
Insert_After_Pol		output_std_=cc=	
AddOn_Pol		**on**	
Pol_Name		output_bcheu	
Insert_After_Pol		bcheu	

The policy called **Implementation_bcheu** contains a function change parameter. This parameter changes the name of the output file. However, we have already specified a new output file name in the **output_bcheu** policy as well as a *ChangeParam* function to switch off the standard output. Thus, this policy can be dropped from the AddOn. As mentioned above the related function *AddOn_Func* of **ao_control_bcheu** in 1.2 can be dropped from the add-on.

Figure 13.8



Implementation_bcheu		on	
ChangeParam		on	
Param_Id	1	output_std_sl_#1.1	parameter 'file' in function 'DefOutput' in policy 'output_std_sl'
Param_NewVal	1	bcheu_AddOn_std	

Policy	Grp/No	bcheu	Comment
ao_control_bcheu		on	
AddOn_Applic		on	
AddOn_Func		on	
Id_Func		b4150c18-a6da-476c-9e17-1efff3bf0515	function 'ChangeParam' in policy 'Implementation_bcheu'
Insert_Before_Func		Uprate_=cc=_#1	function 'Uprate' in policy 'Uprate_sl'

The policies **bcheu** and **output_bcheu** contain the implementation of the benefit and of the output policy.

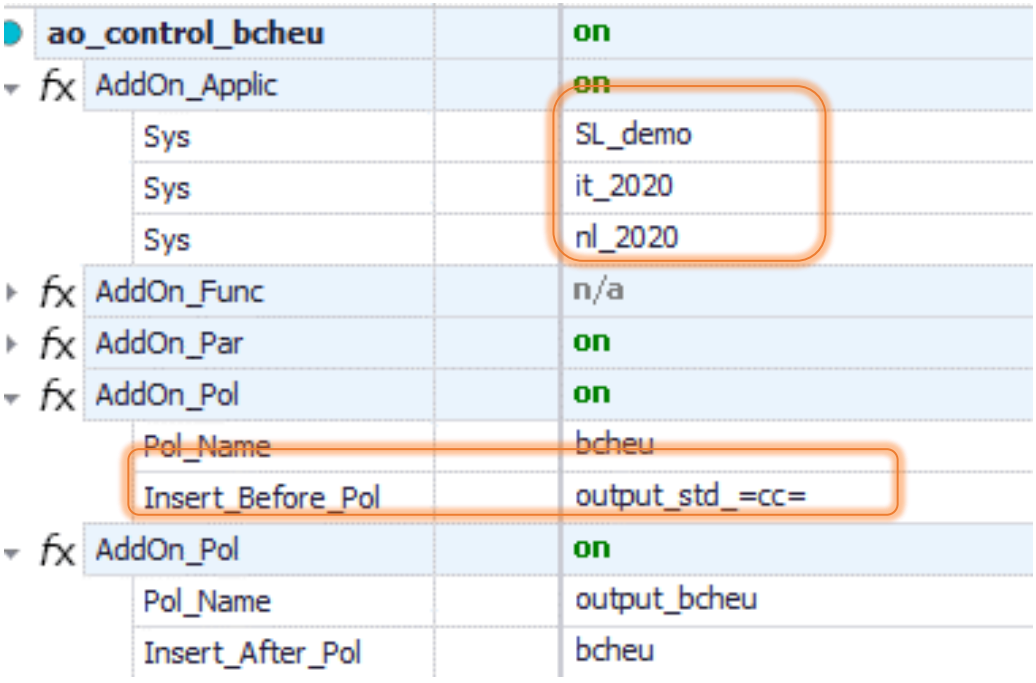
Task 7: Modify the AddOn such that it can be used with the chosen countries

The **AddOn** now needs to be modified such that it will be possible to run it on the 2020 systems of the Netherlands and Italy. The following points summarise the changes needed.

The following changes are required in **the ao_control_bcheu** policy:

1. Add system/year combinations for all three countries
2. Change "*Insert_After_Pol*" to "*Insert_Before_Pol"* in the first **AddOn_Pol**. This can be done by simply clicking on it and EUROMOD will ask you whether you want to change it. This change is necessary as not all countries might have a bsa policy and not always at the end of the spine. Thus, it is better to add the new policy right before the standard output policy available in all countries (**output_std_=cc=**).

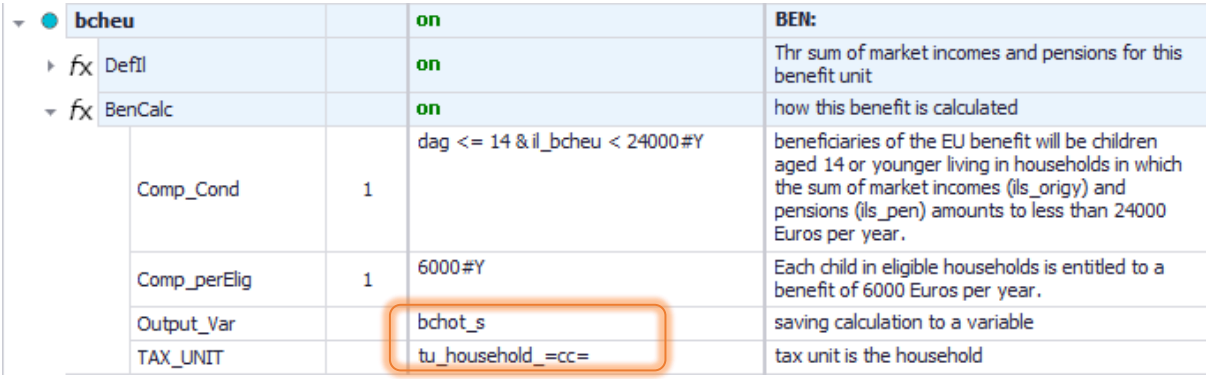
Figure 13.9



ao_control_bcheu		on
AddOn_Applic		on
Sys		SL_demo
Sys		it_2020
Sys		nl_2020
AddOn_Func		n/a
AddOn_Par		on
AddOn_Pol		on
Pol_Name		bcheu
Insert_Before_Pol		output_std_=cc=
AddOn_Pol		on
Pol_Name		output_bcheu
Insert_After_Pol		bcheu

The **bcheu** policy has already been adjusted automatically by EUROMOD when the AddOn was created.

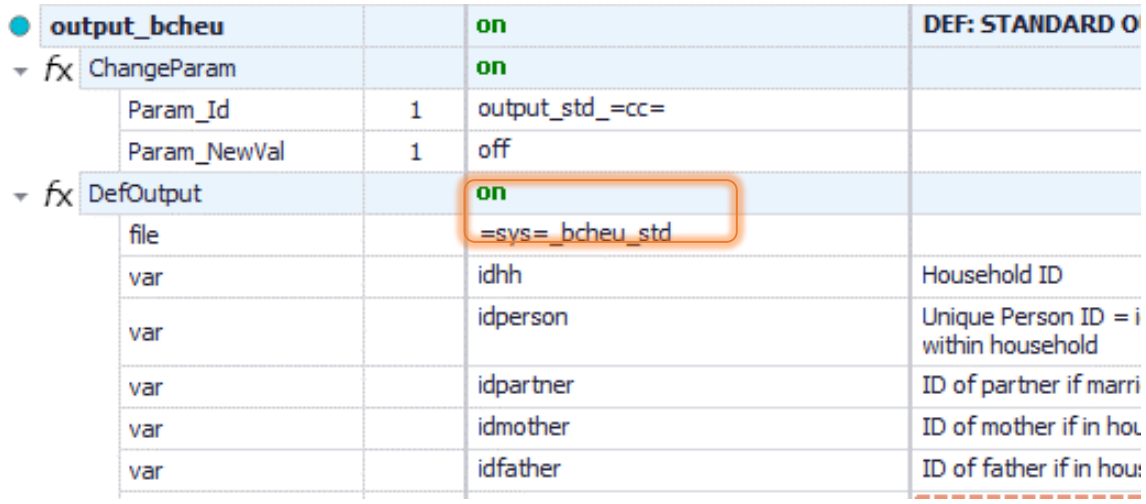
Figure 13.10



bcheu		**on**	**BEN:**
DefIl		**on**	Thr sum of market incomes and pensions for this benefit unit
BenCalc		**on**	how this benefit is calculated
Comp_Cond	1	dag <= 14 &il_bcheu < 24000#Y	beneficiaries of the EU benefit will be children aged 14 or younger living in households in which the sum of market incomes (ils_origy) and pensions (ils_pen) amounts to less than 24000 Euros per year.
Comp_perElig	1	6000#Y	Each child in eligible households is entitled to a benefit of 6000 Euros per year.
Output_Var		bchot_s	saving calculation to a variable
TAX_UNIT		tu_household_=cc=	tax unit is the household

The same is true for **output_bcheu**, except for the name of the output file which needs to be changed to *=sys=_bcheu_std.*

Figure 13.11



output_bcheu		**on**	**DEF: STANDARD O**
ChangeParam		**on**	
Param_Id	1	output_std_=cc=	
Param_NewVal	1	off	
DefOutput		**on**	
file		=sys=_bcheu_std	
var		idhh	Household ID
var		idperson	Unique Person ID = i within household
var		idpartner	ID of partner if marri
var		idmother	ID of mother if in hou
var		idfather	ID of father if in hou

Task 8: Analyse the results

We now have to run the model and each model with the AddOn and analyse the results. On the Run menu, click on View/Filter/Add-Ons > Add-Ons> bcheu (Figure 13.12). Then select the systems for Italy (IT) and Netherlands (NL), filter them for year 2020 and run with the 'training_data' as the dataset. Make sure you have ticked the box for the *bcheu* add on. (Figure 13.13). To analyse the results, we can again use the statistics presenter. In this case we would select multisystem option and then select the four output of the baselines and the reform scenarios. Please note that the name of the reform scenarios appear the same as the baselines, so is it important to remember the order with which you select the files. Then, we can export the statistics in excel and there we can change the name of the reforms scenarios (Figure 13.14).

Figure 13.12

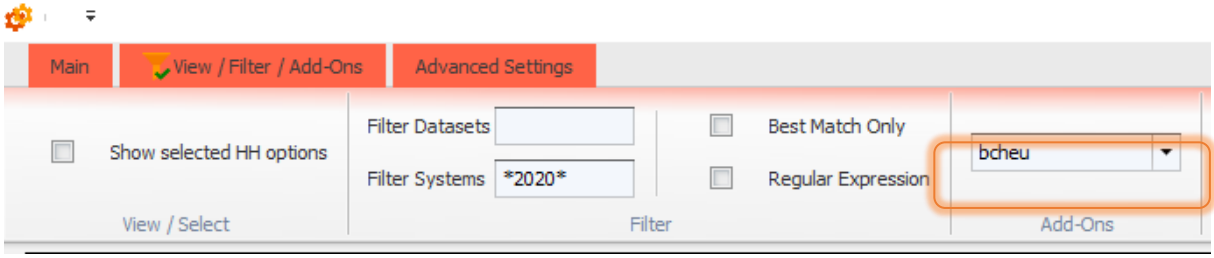


Figure 13.13

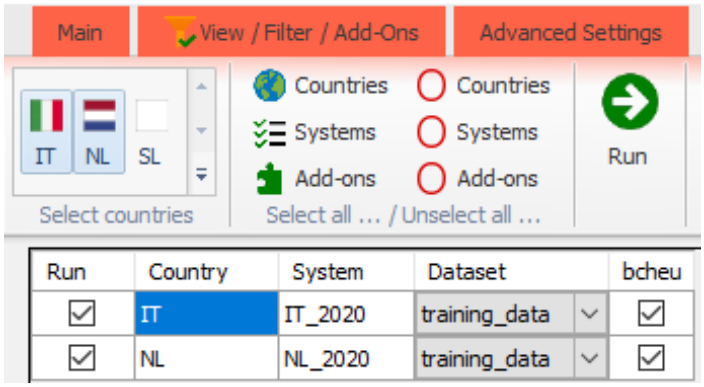


Figure 13.14

Market Incomes and Government Revenue && Expenditure				
Yearly, mill., currency as defined in EM output				
	IT_2020	***IT_2020_bcheu***	***NL_2020***	***NL_2020_bcheu***
Total market incomes	**137.90**	**137.90**	**137.90**	**137.90**
... income from (self) employment	137.90	137.90	137.90	137.90
... other sources	0.00	0.00	0.00	0.00
Government revenue through taxes and social insurance contributions	**87.87**	**87.87**	**61.02**	**61.02**
... direct taxes	33.68	33.68	11.64	11.64
... employee social insurance contributions	10.91	10.91	13.85	13.85
... self-employed social insurance contributions	6.42	6.42	0.00	0.00
... other social insurance contributions	0.00	0.00	17.09	17.09
... employer social insurance contributions (not part of disposable income)	36.86	36.86	18.44	18.44
Credited social insurance contributions (not part of disposable income)	**0.00**	**0.00**	**1.09**	**0.00**
Government expenditure on social transfers	**54.49**	**56.59**	**54.01**	**56.29**
by target group				
... unemployment benefits	1.15	0.00	0.00	0.00
... family and education benefits	1.24	0.00	4.65	0.00
... social assistance and housing benefits	4.35	0.00	16.70	0.00
... pensions, health and disability benefits	47.74	0.00	32.65	0.00
... firms	0.00	0.00	0.00	0.00
by benefit design				
... means-tested non-pension benefits	5.38	7.48	26.58	28.86
... non-means-tested non-pension benefits	1.37	1.37	2.82	2.82
... pensions	47.74	47.74	24.60	24.60
... firms subsidies	0.00	0.00	0.00	0.00

Basic Poverty Indices				
	AROP for IT_2020	***AROP IT_2020_bcheu***	***AROP NL_2020***	***AROP NL_2020_bcheu***
Population	22.38%	19.92%	4.21%	3.73%
Children	17.51%	11.28%	3.50%	1.95%
Working Age	28.80%	27.06%	5.49%	5.24%
Working Age Economically Active	16.82%	16.45%	3.40%	3.40%
Elderly	2.99%	2.49%	0.0%	0.0%
Poverty Line	802.86	811.28	784.24	805.72
Poverty Gap	34.09%	18.50%	38.77%	40.38%

Exercise 14: Modifying uprating factors in Estonia to account for differential wage evolution

The aim of this exercise is to understand the concept of uprating factors and how to feed them with external information on the evolution of a variable.

Description

The average monthly salary in Estonia in 2018 was 1,310 EUR/month. Imagine that you have the following additional (hypothetical) information about salaries between 2018 and 2021:

	Average salary			
Monthly salary in 2018	2018	2019	2020	2021
"Low" salaries (up to average)	1,000	1,030	1,060.9	1,092.727
"High" salaries (above average)	2,000	2,080	2,184	2,315.04

- Do the necessary changes in EUROMOD so that it reflects this different evolution of "low" and "high" salaries, when running the 2021 system.
- Use the In-depth Analysis plugin to compare the new system with the original 2021 system *[only with SILC-based input data]*

Steps

Task 1. Create two new time-series in the uprating indices table (tab Raw indices) to account for the differential evolution of "low" and "high" salaries between 2018 and 2021.

Task 2. Check the differential evolution of the two time series (table Factors per data and system).

Task 3. Create a copy of the EE 2021 system (e.g. EE_2021_uprating).

Task 4. Modify the uprate_ee policy to account for the new information.

Task 5. Run EE_2021 and EE_2021_uprating in the In-depth Analysis plugin and analyse the fiscal and distributional impact *[only with SILC-based input data]*

Hints

- In "Uprating indices → Raw indices" add two rows for two new indices, $f_yemlow and $f_yemhigh, and paste the two time series. Bear in mind that for this exercise we don't care about what happened before 2018, so you can leave those values blank (EUROMOD will fill them with missing values).
- In the new system EE_2021_uprating go to the Uprate function in the uprate_ee policy and tell EUROMOD how to uprate yem00:

- Using the new indices: you have to add two placeholders and two Factor_condition parameters.
- Setting to n/a the original uprating factor used for yem.

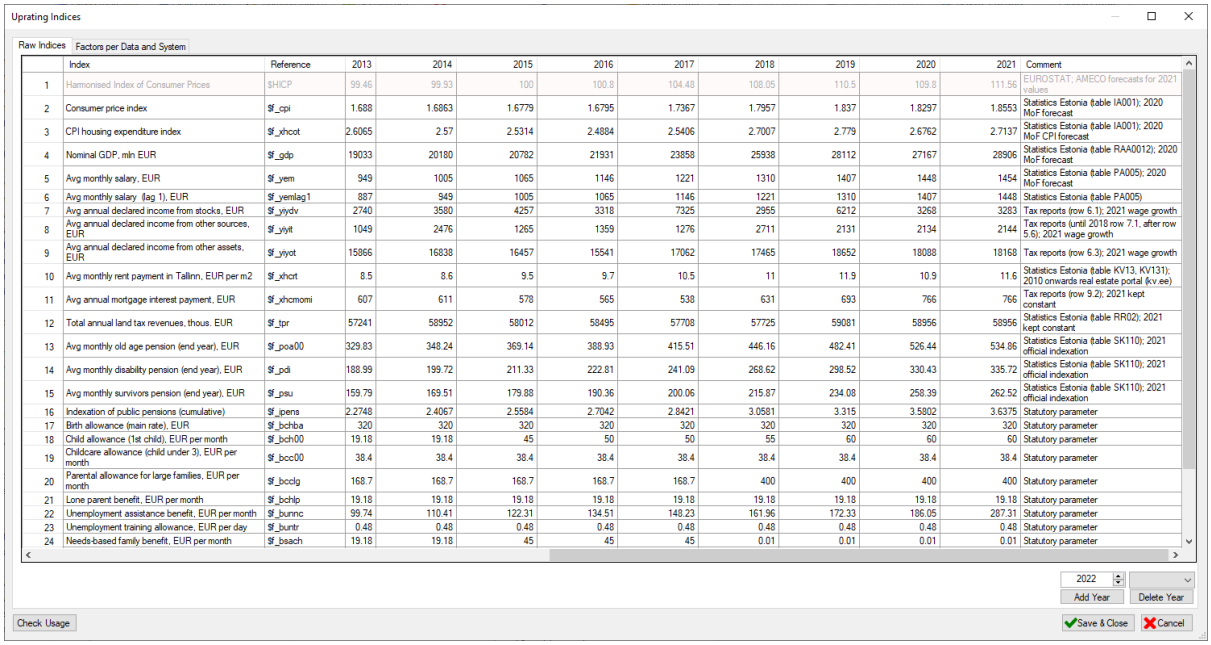
Note: although in most cases the solution proposed for this exercise works correctly, double uprating might occur for some observations whose income is very close to the threshold, due to the sequential application of the two factors. There are ways to circumvent this issue, but they lie beyond the scope of this training. In any case, the exercise fits the double purpose of (1) showing how to use factor conditions and (2) understand the distributional impact of using differential factors by income levels.

Solution and further information

Task 1. Create two new time-series in the uprating indices table (tab Raw indices) to account for the different changes in salaries between 2018 and 2021.

Our first task is to understand how the uprating indices table is constructed in EUROMOD. We open Estonia, we go to Country tools - Uprating indices and we see what is shown in Figure 14.1.

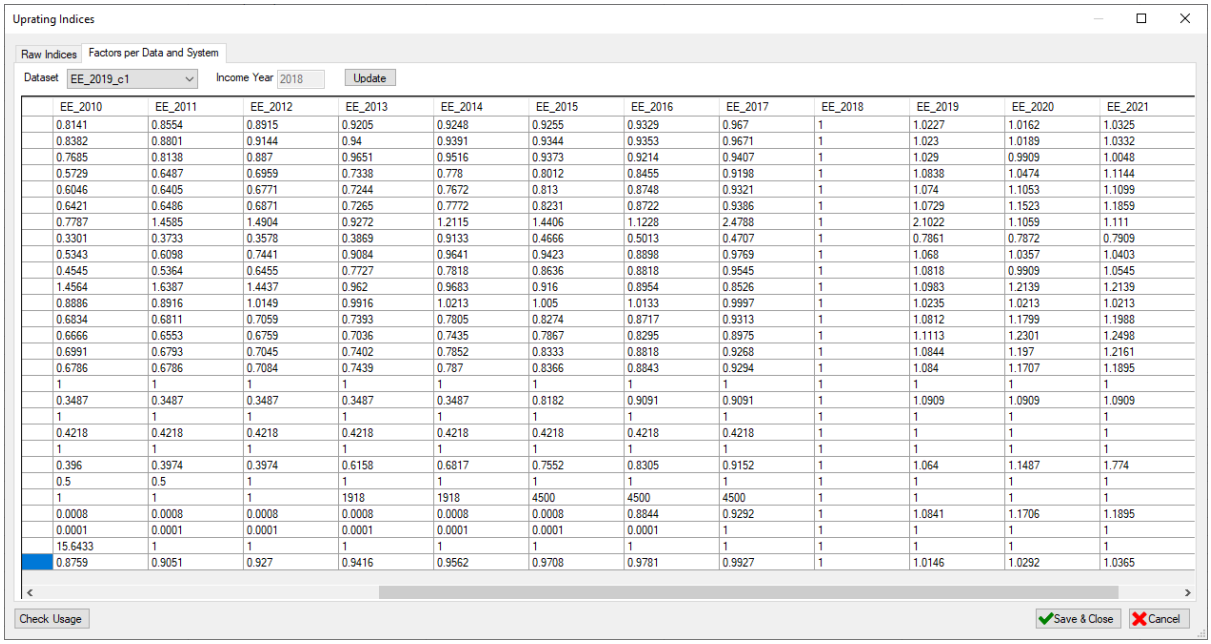
Figure 14.1: Uprating indices - Raw indices



	Index	Reference	2013	2014	2015	2016	2017	2018	2019	2020	2021	Comment
1	Harmonised Index of Consumer Prices	$HICP	99.46	99.93	100	100.8	104.48	108.05	110.5	109.8	111.56	EUROSTAT; AMECO forecasts for 2021 values
2	Consumer price index	$f_cpi	1.688	1.6863	1.6779	1.6795	1.7367	1.7957	1.837	1.8297	1.8553	Statistics Estonia (table IA001); 2020 MoF forecast
3	CPI housing expenditure index	$f_xhcot	2.6065	2.57	2.5314	2.4884	2.5406	2.7007	2.779	2.6762	2.7137	Statistics Estonia (table IA001); 2020 MoF CPI forecast
4	Nominal GDP, mln EUR	$f_gdp	19033	20180	20782	21931	23858	25938	28112	27167	28906	Statistics Estonia (table RAA0012); 2020 MoF forecast
5	Avg monthly salary, EUR	$f_yem	949	1005	1065	1146	1221	1310	1407	1448	1454	Statistics Estonia (table PA005); 2020 MoF forecast
6	Avg monthly salary (lag 1), EUR	$f_yemlag1	887	949	1005	1065	1146	1221	1310	1407	1448	Statistics Estonia (table PA005)
7	Avg annual declared income from stocks, EUR	$f_yydv	2740	3580	4257	3318	7325	2955	6212	3268	3283	Tax reports (row 6.1); 2021 wage growth
8	Avg annual declared income from other sources, EUR	$f_yyit	1049	2476	1265	1359	1276	2711	2131	2134	2144	Tax reports (until 2018 row 7.1, after row 5.6); 2021 wage growth
9	Avg annual declared income from other assets, EUR	$f_yiyot	15866	16838	16457	15541	17062	17465	18652	18088	18168	Tax reports (row 6.3); 2021 wage growth
10	Avg monthly rent payment in Tallinn, EUR per m2	$f_yhcrt	8.5	8.6	9.5	9.7	10.5	11	11.9	10.9	11.6	Statistics Estonia (table KV13, KV131); 2010 onwards real estate portal (kv.ee)
11	Avg annual mortgage interest payment, EUR	$f_xhcmomi	607	611	578	565	538	631	693	766	766	Tax reports (row 9.2); 2021 kept constant
12	Total annual land tax revenues, thous. EUR	$f_tpr	57241	58952	58012	58495	57708	57725	59081	58956	58956	Statistics Estonia (table RR02); 2021 kept constant
13	Avg monthly old age pension (end year), EUR	$f_poa00	329.83	348.24	369.14	388.93	415.51	446.16	482.41	526.44	534.86	Statistics Estonia (table SK110); 2021 official indexation
14	Avg monthly disability pension (end year), EUR	$f_pdi	188.99	199.72	211.33	222.81	241.09	268.62	298.52	330.43	335.72	Statistics Estonia (table SK110); 2021 official indexation
15	Avg monthly survivors pension (end year), EUR	$f_psu	159.79	169.51	179.88	190.36	200.06	215.87	234.08	258.39	262.52	Statistics Estonia (table SK110); 2021 official indexation
16	Indexation of public pensions (cumulative)	$f_ipens	2.2748	2.4067	2.5584	2.7042	2.8421	3.0581	3.315	3.5802	3.6375	Statutory parameter
17	Birth allowance (main rate), EUR	$f_bchba	320	320	320	320	320	320	320	320	320	Statutory parameter
18	Child allowance (1st child), EUR per month	$f_bch00	19.18	19.18	45	50	50	55	60	60	60	Statutory parameter
19	Childcare allowance (child under 3), EUR per month	$f_bcc00	38.4	38.4	38.4	38.4	38.4	38.4	38.4	38.4	38.4	Statutory parameter
20	Parental allowance for large families, EUR per month	$f_bcclg	168.7	168.7	168.7	168.7	168.7	400	400	400	400	Statutory parameter
21	Lone parent benefit, EUR per month	$f_bchlp	19.18	19.18	19.18	19.18	19.18	19.18	19.18	19.18	19.18	Statutory parameter
22	Unemployment assistance benefit, EUR per month	$f_bunnc	99.74	110.41	122.31	134.51	148.23	161.96	172.33	186.05	287.31	Statutory parameter
23	Unemployment training allowance, EUR per day	$f_buntr	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	Statutory parameter
24	Needs-based family benefit, EUR per month	$f_bsach	19.18	19.18	45	45	45	0.01	0.01	0.01	0.01	Statutory parameter

In row 5 we see the index for salaries ($f_yem), expressed in monthly EUR. This means that, if $f_yem is used in the model to uprate yem, this variable will be uprated using the corresponding time series, when simulating 2021 policies. In particular, the average salary increased from 1,310 to 1,454, i.e. 10.99%.This can be seen in the second tab, Factors per data and system, when selecting the 2019 data in the dropdown list (Figure 14.2). The fifth row of the last column shows the factor applied (1.1099, i.e. 1+0.1099).

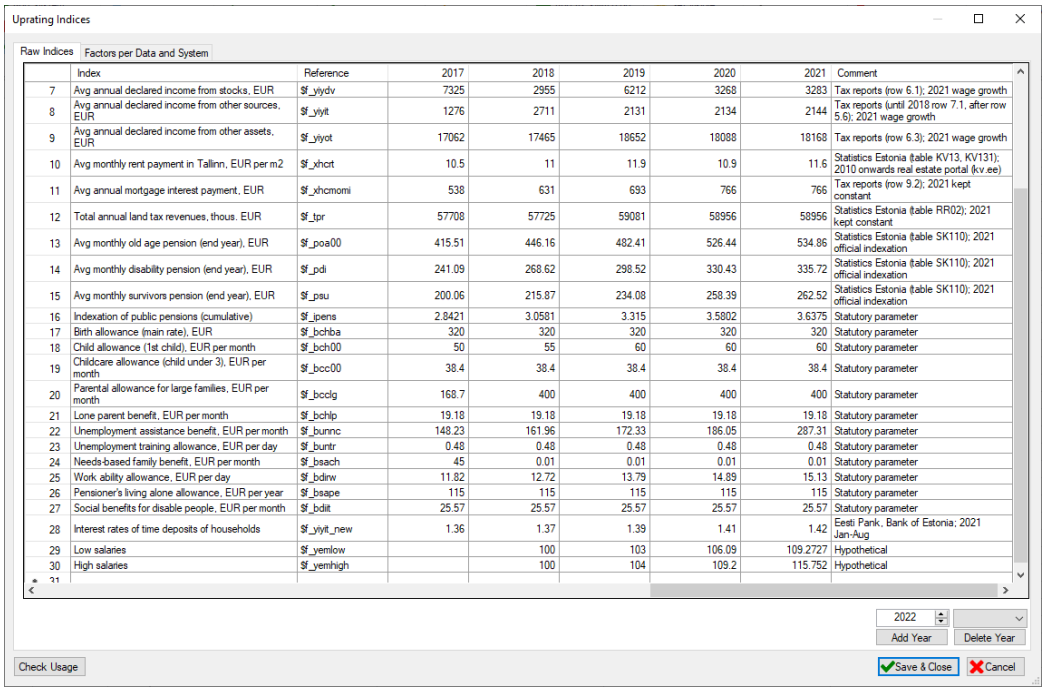
Figure 14.2: Uprating indices - Factors per data and system



EE_2010	EE_2011	EE_2012	EE_2013	EE_2014	EE_2015	EE_2016	EE_2017	EE_2018	EE_2019	EE_2020	EE_2021
0.8141	0.8554	0.8915	0.9205	0.9248	0.9255	0.9329	0.967	1	1.0227	1.0162	1.0325
0.8382	0.8801	0.9144	0.94	0.9391	0.9344	0.9353	0.9671	1	1.023	1.0189	1.0332
0.7685	0.8138	0.887	0.9651	0.9516	0.9373	0.9214	0.9407	1	1.029	0.9909	1.0048
0.5729	0.6487	0.6959	0.7338	0.778	0.8012	0.8455	0.9198	1	1.0838	1.0474	1.1144
0.6046	0.6405	0.6771	0.7244	0.7672	0.813	0.8748	0.9321	1	1.074	1.1053	1.1099
0.6421	0.6486	0.6871	0.7265	0.7772	0.8231	0.8722	0.9386	1	1.0729	1.1523	1.1859
0.7787	1.4585	1.4904	0.9272	1.2115	1.4406	1.1228	2.4788	1	2.1022	1.1059	1.111
0.3301	0.3733	0.3578	0.3869	0.9133	0.4666	0.5013	0.4707	1	0.7861	0.7872	0.7909
0.5343	0.6098	0.7441	0.9084	0.9641	0.9423	0.8898	0.9769	1	1.068	1.0357	1.0403
0.4545	0.5364	0.6455	0.7727	0.7818	0.8636	0.8818	0.9545	1	1.0818	0.9909	1.0545
1.4564	1.6387	1.4437	0.962	0.9683	0.916	0.8954	0.8526	1	1.0983	1.2139	1.2139
0.8886	0.8916	1.0149	0.9916	1.0213	1.005	1.0133	0.9997	1	1.0235	1.0213	1.0213
0.6834	0.6811	0.7059	0.7393	0.7805	0.8274	0.8717	0.9313	1	1.0812	1.1799	1.1988
0.6666	0.6553	0.6759	0.7036	0.7435	0.7867	0.8295	0.8975	1	1.1113	1.2301	1.2498
0.6991	0.6793	0.7045	0.7402	0.7852	0.8333	0.8818	0.9268	1	1.0844	1.197	1.2161
0.6786	0.6786	0.7084	0.7439	0.787	0.8366	0.8843	0.9294	1	1.084	1.1707	1.1895
1	1	1	1	1	1	1	1	1	1	1	1
0.3487	0.3487	0.3487	0.3487	0.3487	0.8182	0.9091	0.9091	1	1.0909	1.0909	1.0909
1	1	1	1	1	1	1	1	1	1	1	1
0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0.396	0.3974	0.3974	0.6158	0.6817	0.7552	0.8305	0.9152	1	1.064	1.1487	1.774
0.5	0.5	1	1	1	1	1	1	1	1	1	1
1	1	1	1918	1918	4500	4500	4500	1	1	1	1
0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.8844	0.9292	1	1.0841	1.1706	1.1895
0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	1	1	1	1	1
15.6433	1	1	1	1	1	1	1	1	1	1	1
0.8759	0.9051	0.927	0.9416	0.9562	0.9708	0.9781	0.9927	1	1.0146	1.0292	1.0365

Now we go back to the Raw indices table and add two new rows at the end, that we can name $f_yemlow and $f_yemhigh. We can easily do this by just pasting the time series below the corresponding years (new rows will be automatically created); the result is shown in Figure 14.3. Then we Save and close.

Figure 14.3: Adding new uprating factors

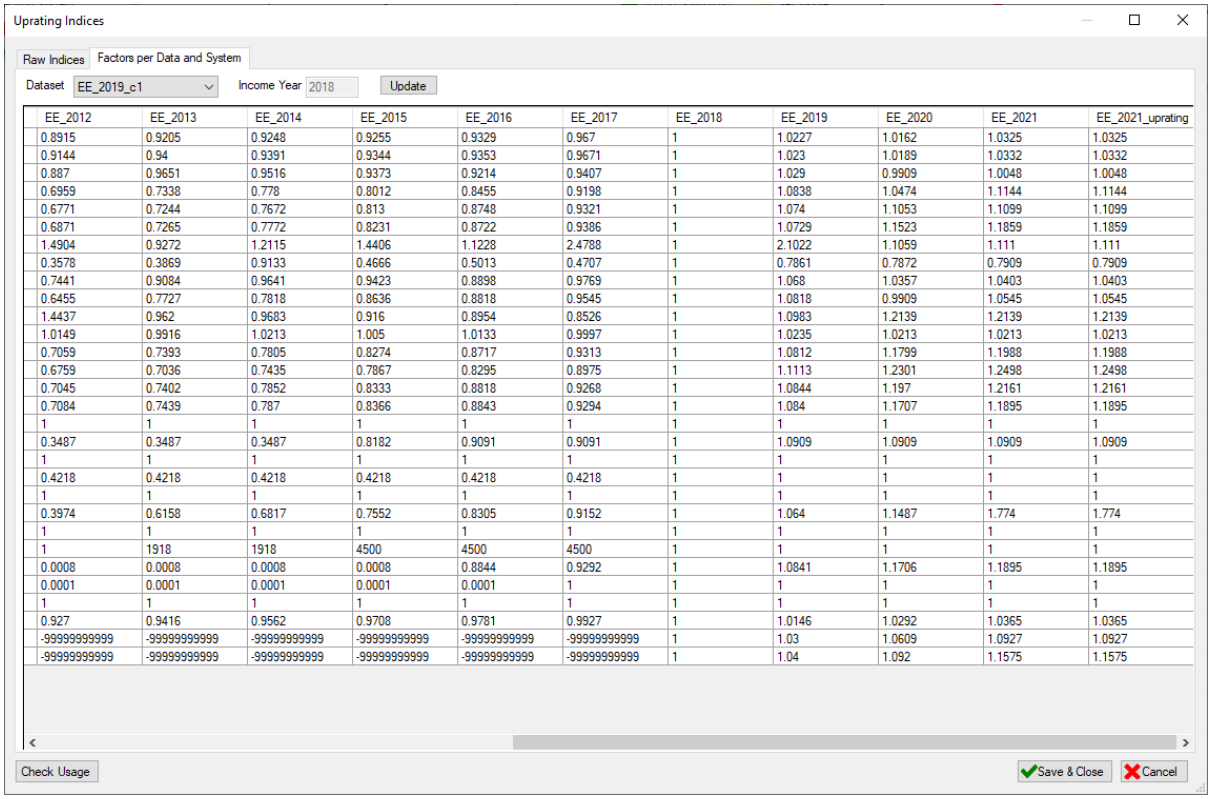


	Index	Reference	2017	2018	2019	2020	2021	Comment
7	Avg annual declared income from stocks, EUR	$f_yiydv	7325	2955	6212	3268	3283	Tax reports (row 6.1); 2021 wage growth
8	Avg annual declared income from other sources, EUR	$f_yiyit	1276	2711	2131	2134	2144	Tax reports (until 2018 row 7.1, after row 5.6); 2021 wage growth
9	Avg annual declared income from other assets, EUR	$f_yiyot	17062	17465	18652	18088	18168	Tax reports (row 6.3); 2021 wage growth
10	Avg monthly rent payment in Tallinn, EUR per m2	$f_yhcrt	10.5	11	11.9	10.9	11.6	Statistics Estonia (table KV13, KV131); 2010 onwards real estate portal (kv.ee)
11	Avg annual mortgage interest payment, EUR	$f_yhcmomi	538	631	693	766	766	Tax reports (row 9.2); 2021 kept constant
12	Total annual land tax revenues, thous. EUR	$f_tpr	57708	57725	59081	58956	58956	Statistics Estonia (table RR02); 2021 kept constant
13	Avg monthly old age pension (end year), EUR	$f_poa00	415.51	446.16	482.41	526.44	534.86	Statistics Estonia (table SK110); 2021 official indexation
14	Avg monthly disability pension (end year), EUR	$f_pdi	241.09	268.62	298.52	330.43	335.72	Statistics Estonia (table SK110); 2021 official indexation
15	Avg monthly survivors pension (end year), EUR	$f_psu	200.06	215.87	234.08	258.39	262.52	Statistics Estonia (table SK110); 2021 official indexation
16	Indexation of public pensions (cumulative)	$f_ipens	2.8421	3.0581	3.315	3.5802	3.6375	Statutory parameter
17	Birth allowance (main rate), EUR	$f_bchba	320	320	320	320	320	Statutory parameter
18	Child allowance (1st child), EUR per month	$f_bch00	50	55	60	60	60	Statutory parameter
19	Childcare allowance (child under 3), EUR per month	$f_bcc00	38.4	38.4	38.4	38.4	38.4	Statutory parameter
20	Parental allowance for large families, EUR per month	$f_bcclg	168.7	400	400	400	400	Statutory parameter
21	Lone parent benefit, EUR per month	$f_bchlp	19.18	19.18	19.18	19.18	19.18	Statutory parameter
22	Unemployment assistance benefit, EUR per month	$f_bunnc	148.23	161.96	172.33	186.05	287.31	Statutory parameter
23	Unemployment training allowance, EUR per day	$f_buntr	0.48	0.48	0.48	0.48	0.48	Statutory parameter
24	Needs-based family benefit, EUR per month	$f_bsach	45	0.01	0.01	0.01	0.01	Statutory parameter
25	Work ability allowance, EUR per day	$f_bdirw	11.82	12.72	13.79	14.89	15.13	Statutory parameter
26	Pensioner's living alone allowance, EUR per year	$f_bsape	115	115	115	115	115	Statutory parameter
27	Social benefits for disable people, EUR per month	$f_bdit	25.57	25.57	25.57	25.57	25.57	Statutory parameter
28	Interest rates of time deposits of households	$f_yiyit_new	1.36	1.37	1.39	1.41	1.42	Eesti Pank, Bank of Estonia; 2021 Jan-Aug
29	Low salaries	$f_yemlow		100	103	106.09	109.2727	Hypothetical
30	High salaries	$f_yemhigh		100	104	109.2	115.752	Hypothetical

Task 2. Check the differential evolution of the two time series (table Factors per data and system).

Now we go again to the second tab, select the 2019 dataset and see the impact that the use of the new time series would have. We can compare them with $f_yem, to see the differences between using this index or the two that we have just created: 1.1099 vs. 1.0927 and 1.1575 (Figure 14.4).

Figure 14.4: Impact of the new indices



EE_2012	EE_2013	EE_2014	EE_2015	EE_2016	EE_2017	EE_2018	EE_2019	EE_2020	EE_2021	EE_2021_uprating
0.8915	0.9205	0.9248	0.9255	0.9329	0.967	1	1.0227	1.0162	1.0325	1.0325
0.9144	0.94	0.9391	0.9344	0.9353	0.9671	1	1.023	1.0189	1.0332	1.0332
0.887	0.9651	0.9516	0.9373	0.9214	0.9407	1	1.029	0.9909	1.0048	1.0048
0.6959	0.7338	0.778	0.8012	0.8455	0.9198	1	1.0838	1.0474	1.1144	1.1144
0.6771	0.7244	0.7672	0.813	0.8748	0.9321	1	1.074	1.1053	1.1099	1.1099
0.6871	0.7265	0.7772	0.8231	0.8722	0.9386	1	1.0729	1.1523	1.1859	1.1859
1.4904	0.9272	1.2115	1.4406	1.1228	2.4788	1	2.1022	1.1059	1.111	1.111
0.3578	0.3869	0.9133	0.4666	0.5013	0.4707	1	0.7861	0.7872	0.7909	0.7909
0.7441	0.9084	0.9641	0.9423	0.8898	0.9769	1	1.068	1.0357	1.0403	1.0403
0.6455	0.7727	0.7818	0.8636	0.8818	0.9545	1	1.0818	0.9909	1.0545	1.0545
1.4437	0.962	0.9683	0.916	0.8954	0.8526	1	1.0983	1.2139	1.2139	1.2139
1.0149	0.9916	1.0213	1.005	1.0133	0.9997	1	1.0235	1.0213	1.0213	1.0213
0.7059	0.7393	0.7805	0.8274	0.8717	0.9313	1	1.0812	1.1799	1.1988	1.1988
0.6759	0.7036	0.7435	0.7867	0.8295	0.8975	1	1.1113	1.2301	1.2498	1.2498
0.7045	0.7402	0.7852	0.8333	0.8818	0.9268	1	1.0844	1.197	1.2161	1.2161
0.7084	0.7439	0.787	0.8366	0.8843	0.9294	1	1.084	1.1707	1.1895	1.1895
1	1	1	1	1	1	1	1	1	1	1
0.3487	0.3487	0.3487	0.8182	0.9091	0.9091	1	1.0909	1.0909	1.0909	1.0909
1	1	1	1	1	1	1	1	1	1	1
0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
0.3974	0.6158	0.6817	0.7552	0.8305	0.9152	1	1.064	1.1487	1.774	1.774
1	1	1	1	1	1	1	1	1	1	1
1	1918	1918	4500	4500	4500	1	1	1	1	1
0.0008	0.0008	0.0008	0.0008	0.8844	0.9292	1	1.0841	1.1706	1.1895	1.1895
0.0001	0.0001	0.0001	0.0001	0.0001	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
0.927	0.9416	0.9562	0.9708	0.9781	0.9927	1	1.0146	1.0292	1.0365	1.0365
-99999999999	-99999999999	-99999999999	-99999999999	-99999999999	-99999999999	1	1.03	1.0609	1.0927	1.0927
-99999999999	-99999999999	-99999999999	-99999999999	-99999999999	-99999999999	1	1.04	1.092	1.1575	1.1575

Task 3. Create a copy of the EE 2021 system (e.g. EE_2021_uprating).

We right click on EE_2021, select Copy/paste system, and name the new as EL_2021_uprating.

Task 4. Modify the uprate_ee policy to account for the new information.

Until now we have just created two new series of uprating indices, but this won't have any impact on the results of the model unless we use them in the EUROMOD spine (as $f_yemlow and $f_yemhigh). We can do this in the policy uprate_ee (row 2, function 2.1). We see that currently the main variable for salaries (yem00) is uprated using the index $f_yem (row 2.1.49).

Now we have more detailed information in indices $f_yemlow $f_yemhigh, so we want to add new rows to account for them. We right click on the uprate function 2.1, we click on "Show add parameter form" and we add the parameters (two of each!) shown in Figure 14.5. Then we click on "Add" and close.

The factor conditions allow us to differentiate the treatment of salaries below and above the average wage, which we do in the spine as shown in Figure 14.6. We use the value of the 2018 average salary because the income reference period for the 2019 dataset is 2018. Don't forget to set the old index for yem00 ($f_yem) to n/a, since we don't want to use it anymore (we would be applying two indices to the same variable, yem00).

Figure 14.5: Adding new parameters

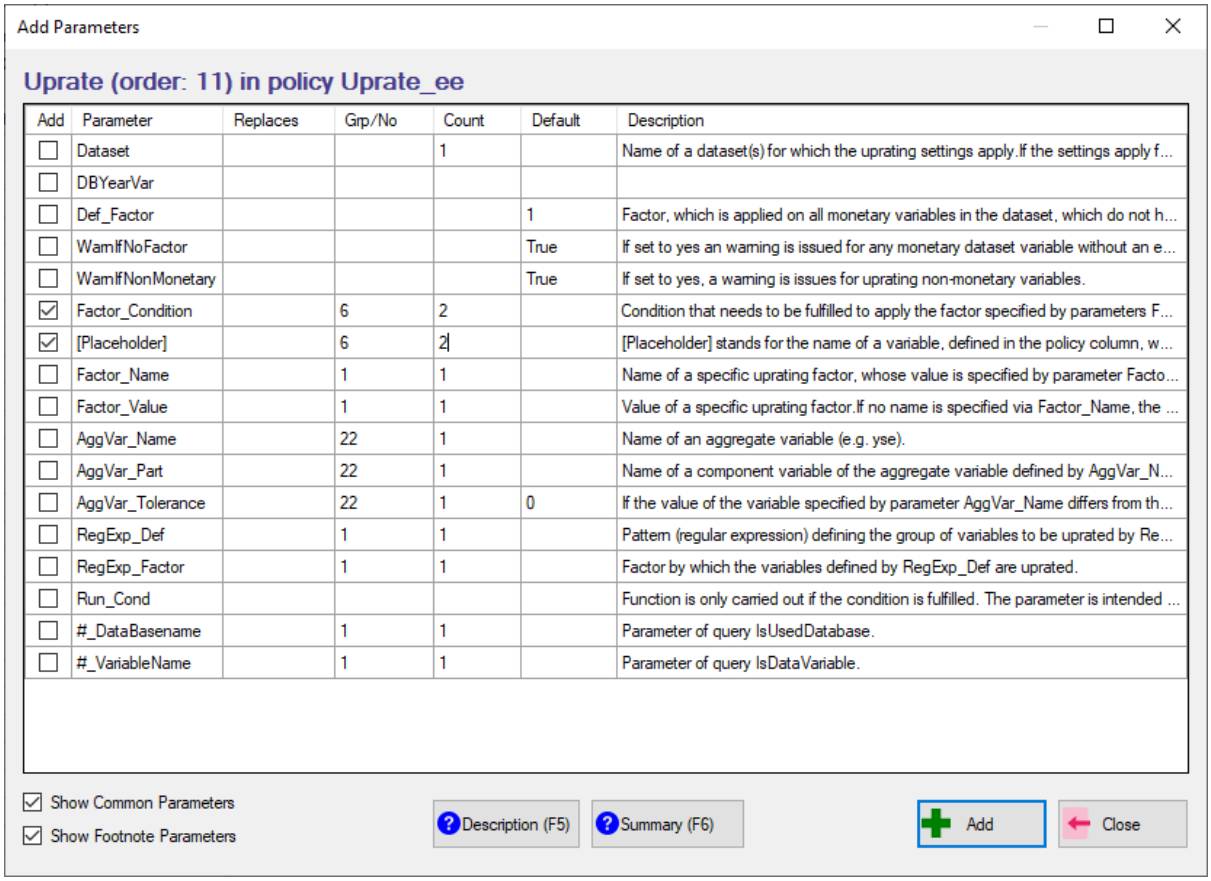


Figure 14.6: Applying new uprating indices

2.1.47	yem19_a	4	$f_yem	$f_yem	
2.1.48	yem20_a	5	$f_yem	$f_yem	
2.1.49	yem00		$f_yem	n/a	
2.1.50	Factor_Cond...	6	n/a	yem00 <= 1310#m	For salaries equal or below the 2018 average salary...
2.1.51	yem00	6	n/a	$f_yemlow	... apply this factor
2.1.52	Factor_Cond...	7	n/a	yem00 > 1310#m	For salaries above...
2.1.53	yem00	7	n/a	$f_yemhigh	... apply this factor
2.1.54	yemabnt		$f_yem	$f_yem	
2.1.55	yemabtx		$f_yem	$f_yem	

Task 5. Run EE_2021 and EE_2021_uprating and analyse the fiscal impact in the In-depth Analysis plugin [only with SILC-based input data]

Note that this can only be done with EU-SILC based data, since training data is not uprated. You just have to run the 2021 and the 2021_uprating system and compare them using the In-depth plugin. A couple of tables are shown in Figure 14.7. Be aware that there are no differences between the tax-benefit policies applied, so all the differences come from the different evolution of salaries assumed from 2018 to 2021. Hence, the results should not be interpreted as the result of a reform, but rather as the impact of the use of more accurate uprating factors.

Figure 14.7: Comparing results

In-depth Analysis

1. Fiscal | 2. Distributional | 3. Inequality and Poverty | Metadata | Parameter settings

1. Fiscal

1.1. Aggregate earnings, government revenue and expenditure (annual)

	Total Estonia 2021 (Baseline)	Total Estonia 2021 uprating	Estonia 2021 uprating Diff. w.r.t. Baseline	Estonia 2021 uprating Diff. % of Baseline
+ income from registered self-employment activities (ysera)	124,885,335	124,885,335	0	0.0 %
+ income from unregistered self-employment activities (ysena)	42,275,030	42,275,030	0	0.0 %
+ self-employment income from agriculture (yseag)	0	0	0	0.0 %
+ self-employment income from other than agriculture (ysebs)	0	0	0	0.0 %
+ self-employment income (abroad) (yseab)	0	0	0	0.0 %
+ income from creative activities (yseil)	0	0	0	0.0 %
+ taxable employment income (abroad) (yemabtx)	0	0	0	0.0 %
+ non-taxable employment income (abroad) (yemabnt)	0	0	0	0.0 %
+ employment income (domestic) (yem00)	10,511,089,829	10,808,787,808	297,697,979	2.8 %
+ covid-19 compensation paid by the firm (yemmc_s)	0	0	0	0.0 %
Total earnings (ils_earns)	**10,678,250,153**	**10,975,948,156**	**297,698,003**	**2.8 %**
Other original income (ils_origy - ils_earns)	**446,726,025**	**446,725,784**	**-240**	**0.0 %**
+ income tax (tulumaks) (tin_s)	1,774,437,837	1,852,027,256	77,589,418	4.4 %
+ land tax (maamaks) (tpr)	15,140,924	15,140,924	0	0.0 %
Total taxes (ils_tax)	**1,789,578,609**	**1,867,168,076**	**77,589,467**	**4.3 %**
+ self-employed pension SIC transfer (ttfsepi_s)	1,900,662	1,903,927	3,265	0.2 %
+ self-employed SIC: funded pension contribution (tpcsepi_s)	950,423	951,992	1,569	0.2 %
+ self-employed SIC: pension (tscsepi_s)	16,124,781	16,145,441	20,661	0.1 %
+ self-employed SIC: health (tscsehl_s)	11,716,623	11,732,141	15,518	0.1 %
+ employee SIC: unemployment (tsceeui_s)	155,640,583	160,201,340	4,560,757	2.9 %
+ employee SIC: funded pension contribution (tpceepi_s)	170,768,601	175,858,054	5,089,452	3.0 %

In-depth Analysis

1. Fiscal | 2. Distributional | Metadata | Parameter settings

2. Distributional

by deciles of equivalised disposable income

2.6. Mean eq. disp. income (annual)

	Total Estonia 2021 (Baseline)	Total Estonia 2021 uprating	Estonia 2021 uprating Diff. w.r.t. Baseline	Estonia 2021 uprating Diff. % of Baseline
Decile 1	4,805	4,786	-19	-0.40 %
Decile 2	6,869	6,854	-15	-0.21 %
Decile 3	8,361	8,343	-18	-0.22 %
Decile 4	9,966	9,975	9	0.09 %
Decile 5	11,680	11,741	61	0.52 %
Decile 6	13,413	13,598	185	1.38 %
Decile 7	15,378	15,694	316	2.05 %
Decile 8	17,714	18,137	423	2.39 %
Decile 9	21,295	21,935	640	3.00 %
Decile 10	28,199	29,047	848	3.01 %
All	**13,765**	**14,007**	**243**	**1.76 %**

Exercise 15: Introducing a benefit for secondary education in Simpleland

The aim of this exercise is to implement a new benefit from scratch in Simpleland and analyse it with the in-depth analysis tool.

Description

Unlike real EU countries, Simpleland has a very simple tax-benefit system with a reduced number of taxes and benefits. A typical benefit that is missing is an education benefit, so we propose to include one. For learning purposes, we choose to implement a benefit only for students enrolled in secondary education, with the following features:

Entitlement	Households meeting the means test and with at least one person enrolled in secondary education
Means-test	Below 30,000 EUR/year of taxable income at household level
Benefit amount	100 EUR/year per person enrolled in secondary education
Interactions with the tax-benefit system	Non-taxable and not subject to SIC Part of the means test for social assistance

Implement this benefit in Simpleland and use the In-depth Analysis tool to analyse its impact.

Steps

Task 1. Check if the variable exists and add it if needed.

Task 2. Create a new system by copy-pasting the baseline.

Task 3. Create a new EUROMOD policy. Make sure that you place it correctly in the spine.

Task 4. Add the variable the relevant income lists

Task 5. Output additional variables and run baseline and reform

Task 6. Analyse the output with the In-depth Analysis tool

Hints

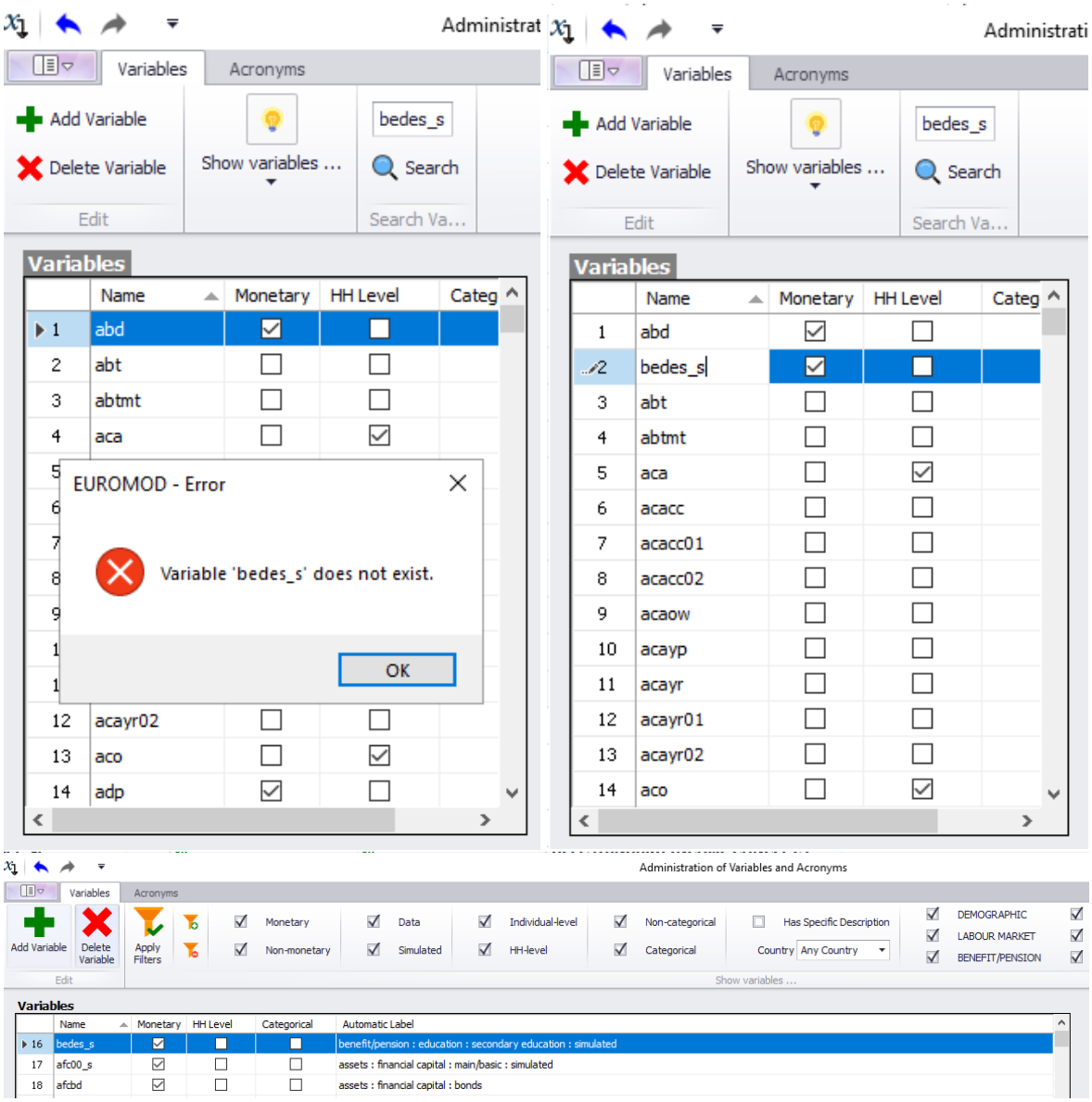
- The output variable should be named bedes_s (benefit | education | secondary | simulated). It doesn't exist in the variable list, so you have to create it there.
- The new policy should be placed after PIT and SIC and before social assistance.
- The variable has to be added to simulated benefits, means-tested benefits and the income list used for the social assistance means test. In real countries, it should be also added to the income lists for education benefits, so that they are correctly classified in the Statistics Presenter (these income lists don't exist in Simpleland, but they can be created).
- There are several ways of coding the benefit with the functions learnt so far.

Solution and further information

Task 1. Check if the variable exists and add it if needed.

The most appropriate name for the output variable is bedes_s (benefit | education | secondary | simulated). Go to Administration Tools - Variables and search it using the Search box. Once you confirm it doesn't exist, create it using the Add Variable button. You just have to type it and EUROMOD will fill the full name, since the acronyms already exist (Figure 15.1). Don't worry about the order, EUROMOD will reorder everything alphabetically once you close the Variable list.

Figure 15.1: Creating a new variable



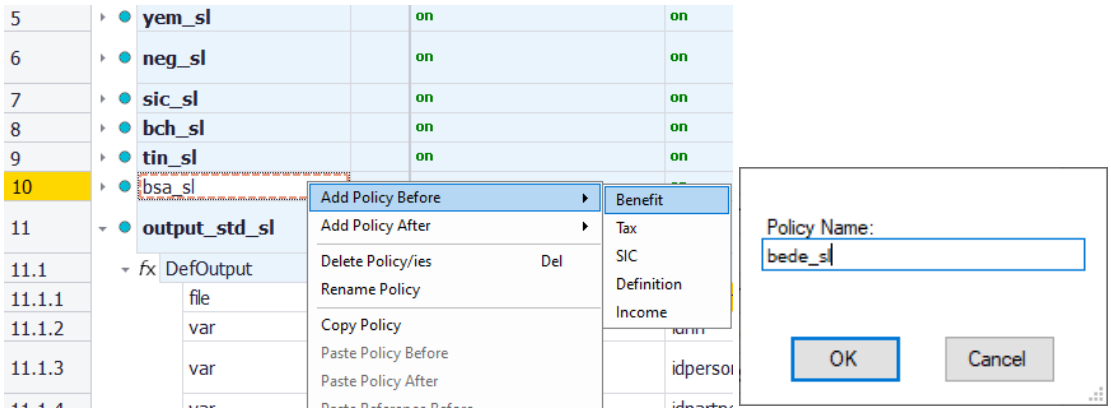
Task 2. Create a new system

Like in previous exercises, create first a new system named. You can do this by right clicking on the heading of SL_demo → Copy/paste system and name it SL_demo_ref.

Task 3. Create a new EUROMOD policy

The only restriction in terms of position of the policy is that it has to be placed before social assistance (bsa_sl), since the new benefit is part of the means test. Since it's not taxable by PIT of SIC, the position in relation to those policies is irrelevant, but for clarity we may place it immediately before bsa_sl (there is no doubt that it doesn't affect any other policy). In order to do this, right-click on bsa_sl → Add Policy Before → Benefit and name it bedes_sl.

Figure 15.2: Adding a new policy



The benefit can be easily simulated with the combination Elig + BenCalc, as shown in Figure 15.3.

Figure 15.3: Adding functions and parameters

10	**bedes_sl**		n/a	on	**BEN: Secondary education benefit**
10.1	fx Elig		n/a	on	Means test
10.1.1	Elig_Cond		n/a	il_TaxableY < 30000#y	Households with taxable income below 30,000 EUR/year
10.1.2	Result_Var		n/a	sin01_s	Saved as temporary variable for checks
10.1.3	TAX_UNIT		n/a	tu_household_sl	
10.2	fx BenCalc		n/a	on	Computation of benefit
10.2.1	Who_Must_Be_Elig		n/a	one	For households meeting the means test...
10.2.2	Comp_Cond	1	n/a	dec = 3 | dec = 4	.. for each person secondary education...
10.2.3	Comp_perElig	1	n/a	100#m	... 100 EUR/month
10.2.4	Output_Var		n/a	bedes_s	
10.2.5	TAX_UNIT		n/a	tu_household_sl	

Some comments:

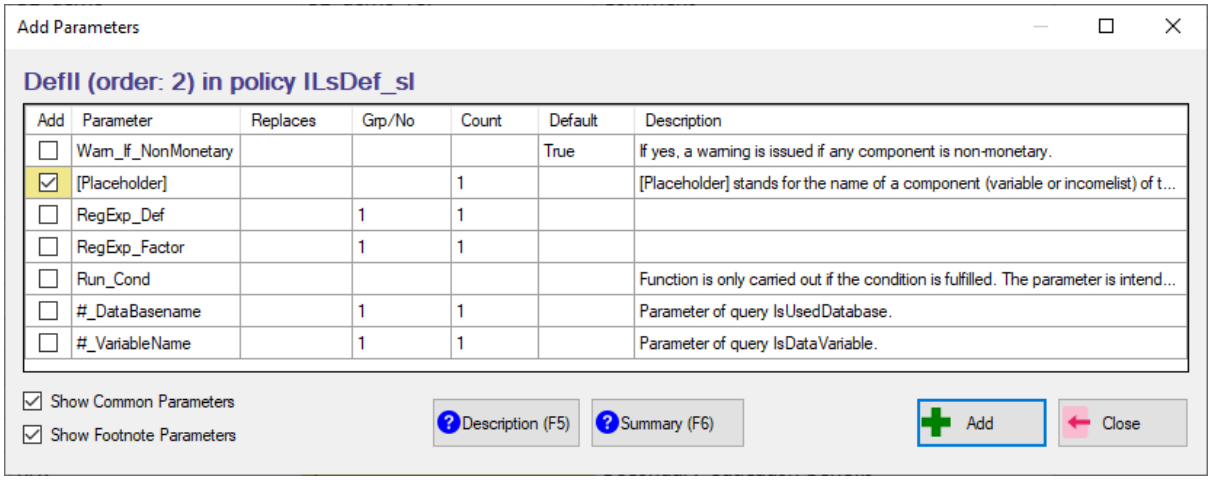
- The income test relies on taxable income, which is already defined in an income list in the spine of Simpleland (il_TaxableY).
- ResultVar in 10.1.2 allow us to save the dummy of Elig in a temporary variable named sin01_s. Remember that for this specific type of variables you don't need to use the DefVar function before. In this case we only create the variable for testing purposes, in case we want to check the microdata.
- Note that the dummy will be 1 for all the individuals of the eligible households. This is why in 10.2 it's enough if one person is eligible (selecting All would have the same effect).
- The variable dec states the current education of status of the individual; 3 and 4 are the values that correspond to secondary school (you can check this information in the DRD codebook in your input folder).
- Note that we give 100 per eligible individual (Comp_perElig), not per household (for this we should use Com_perTU).

Task 4. Add the variable to the relevant income lists

The variable bedes_s has to be added to the following income lists: simulated benefits (ils_bensim), means-tested benefits (ils_benmt) and the income list used for the social assistance means test (il_bsa).

The standard lists (ils_*) can be found in the policy ILsDef_sl, and the non-standard (il_bsa) in ILDef_sl. You can add the variable by adding a placeholder with the Add parameter form option in the corresponding function, either right-clicking or pressing CTRL+A (see Figure 15.4).

Figure 15.4: Adding a parameter to a DefIl function



In real countries, it should be also added to the income lists for education benefits, so that they are correctly classified in the Statistics Presenter (these income lists don't exist in Simpleland, but they can be created).

Task 5. Output additional variables and run baseline and reform

For the analysis we are interested in having three additional variables in the output that are not outputted by default in Simpleland:

- Income list il_TaxableY, in order to have it available for the In-depth Analysis tool.
- sin01_s, for testing purposes.
- tscer_s, since its absence issues a warning when running the In-depth Analysis tool.

For this we use three different parameters (using again the Add parameter form):

- ILGroup to add il_*: it will output all non-standard income lists, including il_TaxableY.
- VarGroup to add sin*: it will output all variables starting by "sin", including sin01_s.
- Var to add tscer_s, to add this single variable.

The result can be seen in Figure 15.5. In the case of tscer_s, we add it also in the baseline, to avoid the abovementioned warning.

Figure 15.5: Adding parameters to a DefOutput function

12.1.30	ILGroup		n/a	il_*	Non-standard income lists
12.1.31	VarGroup		n/a	sin*	Temporary variables
12.1.32	Var		tscer_s	tscer_s	employers' SIC (not part of ils_dispy)
12.1.33	nDecimals		2	2	
12.1.34	TAX_UNIT		tu_individual_sl	tu_individual_sl	

When done, run both baseline and reform scenarios.

Task 6. Analyse the output with the In-depth Analysis tool

In order to use this tool, we have to go to Applications → EUROMOD Statistics → In-depth Analysis. Since this tool allows for full customisation, we take advantage of it by going to the Advanced menu and select the indicators you want to customise. An example of customisation of 2. Distributional settings is shown in Figure 15.6. Once customised, you click on ok and select also the tables you want to be computed. If you want to save this customisation, you can do it by clicking on File → Save (Figure 15.7).

Then you just have to Select baselines (sl_demo_std.txt) and reforms (sl_demo_ref_std.txt) by clicking on the corresponding button, and click on Run. Figure 15.8 shows (partially) one of the tabs of the output. You can save the output by clicking on the top-right button.

Figure 15.6: Customising the distributional tables

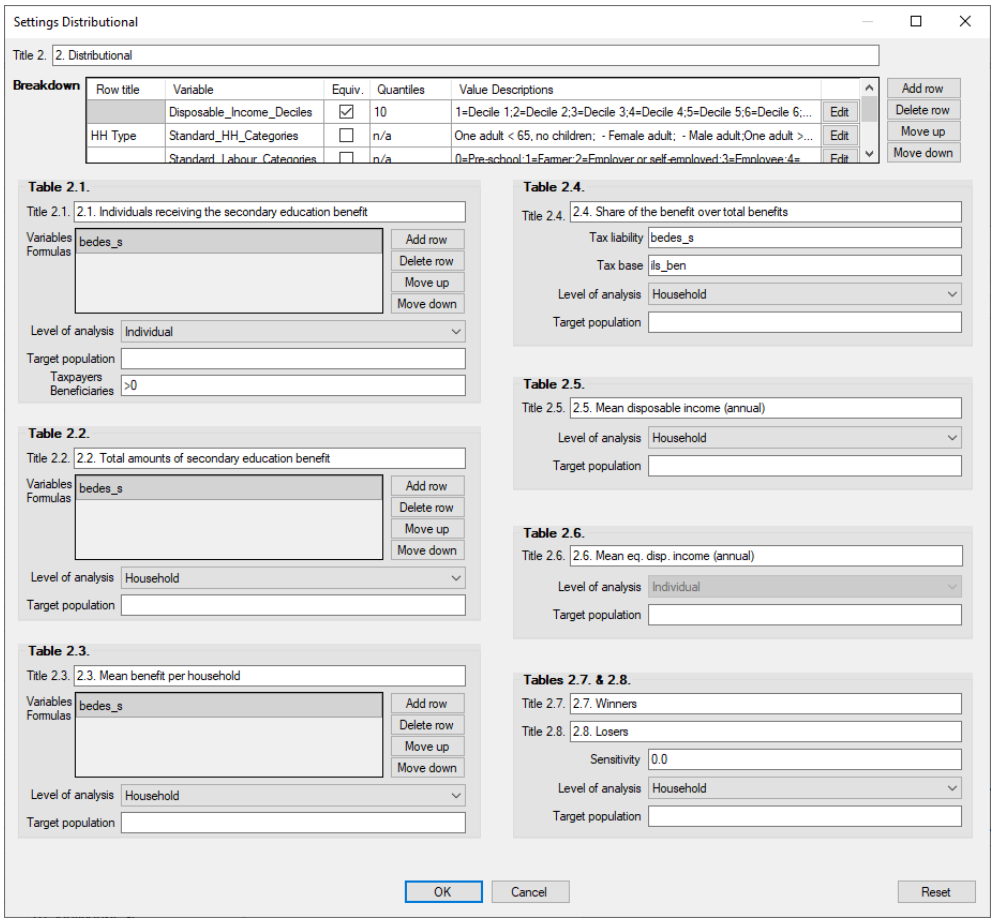


Figure 15.7: Saving the customisation

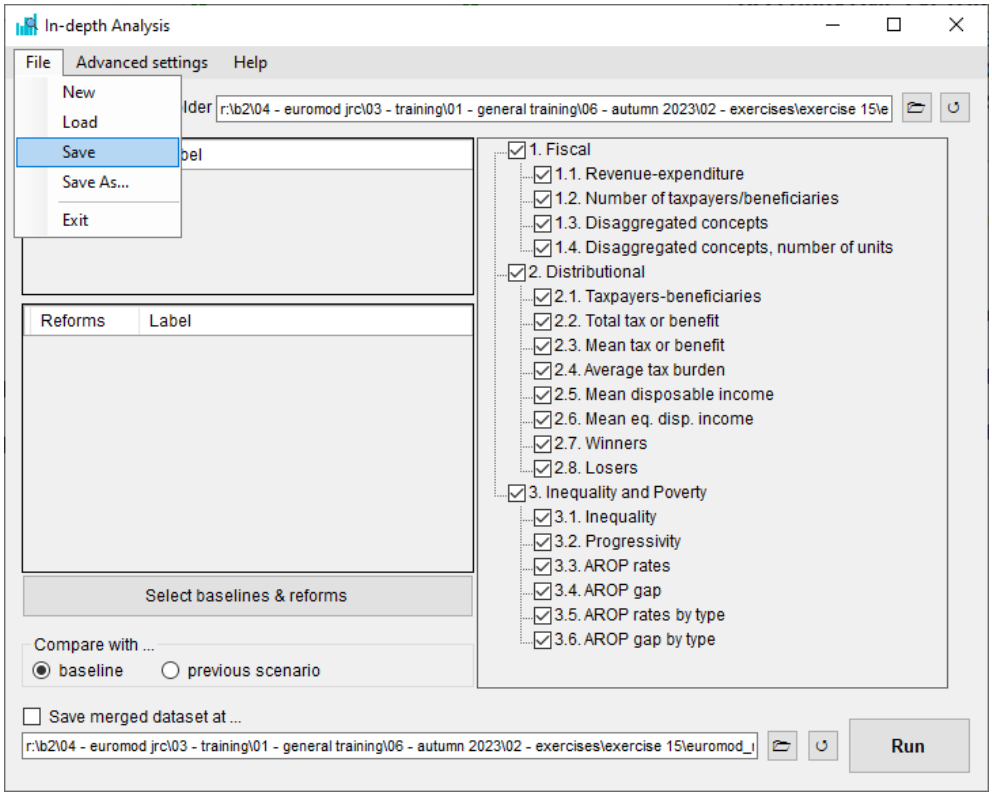
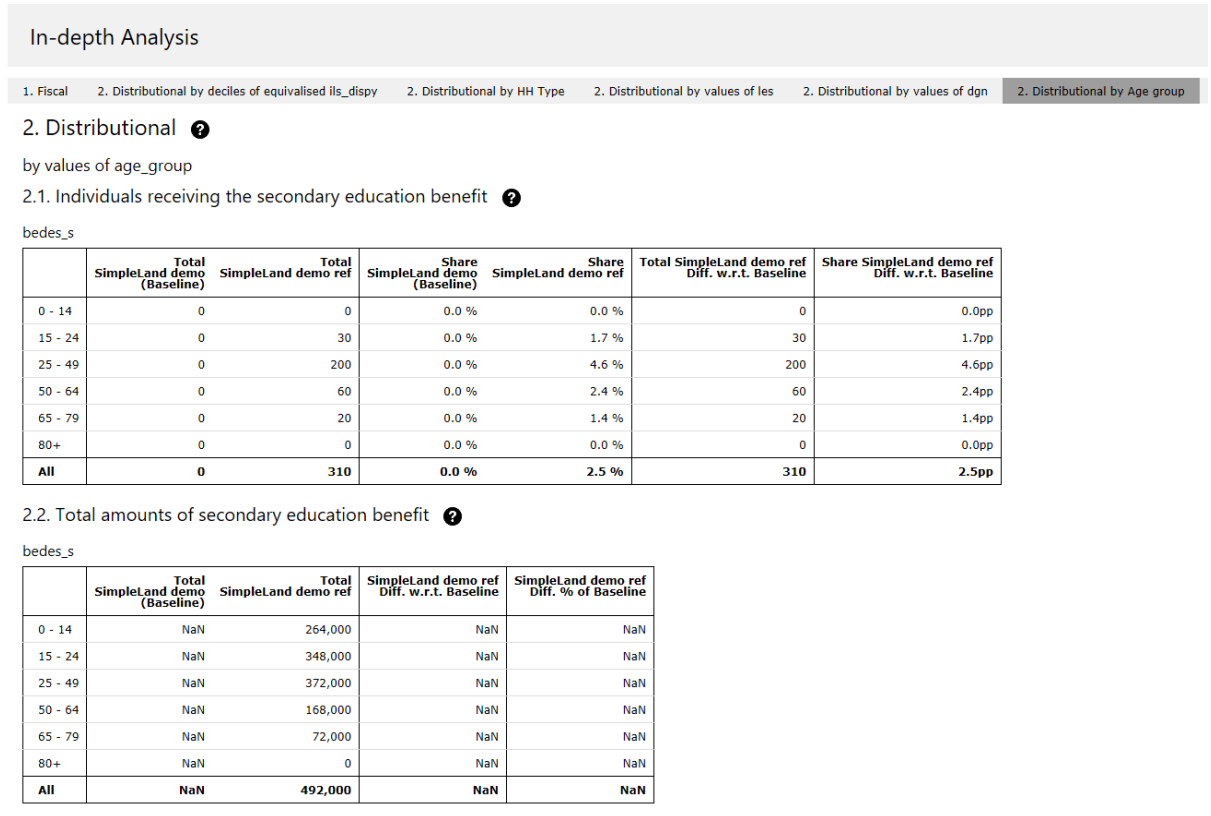


Figure 15.8: Checking the results



	Total SimpleLand demo (Baseline)	Total SimpleLand demo ref	Share SimpleLand demo (Baseline)	Share SimpleLand demo ref	Total SimpleLand demo ref Diff. w.r.t. Baseline	Share SimpleLand demo ref Diff. w.r.t. Baseline
0 - 14	0	0	0.0 %	0.0 %	0	0.0pp
15 - 24	0	30	0.0 %	1.7 %	30	1.7pp
25 - 49	0	200	0.0 %	4.6 %	200	4.6pp
50 - 64	0	60	0.0 %	2.4 %	60	2.4pp
65 - 79	0	20	0.0 %	1.4 %	20	1.4pp
80+	0	0	0.0 %	0.0 %	0	0.0pp
All	**0**	**310**	**0.0 %**	**2.5 %**	**310**	**2.5pp**

2.2. Total amounts of secondary education benefit

bedes_s

	Total SimpleLand demo (Baseline)	Total SimpleLand demo ref	SimpleLand demo ref Diff. w.r.t. Baseline	SimpleLand demo ref Diff. % of Baseline
0 - 14	NaN	264,000	NaN	NaN
15 - 24	NaN	348,000	NaN	NaN
25 - 49	NaN	372,000	NaN	NaN
50 - 64	NaN	168,000	NaN	NaN
65 - 79	NaN	72,000	NaN	NaN
80+	NaN	0	NaN	NaN
All	**NaN**	**492,000**	**NaN**	**NaN**