



EUROMOD training course

Course based on
EUROMOD I6.0++, software v. 3.6.19
15-17 May 2024

Fiscal Policy Analysis Unit
Joint Research Centre
European Commission



TRAINING
EUROMOD

Joint
Research
Centre

Course structure



Mix of lecture and hands-on exercises



Extra exercises offered to do after the course



Please ask questions



Tell us what you would like to do with the model

Course outline

- Session 1** Introduction to microsimulation; EUROMOD and its language; handling errors
- Session 2** Variable types; system function *DefVar*; policy functions *Elig* & *ArithOp*
- Session 3** Policy functions *BenCalc* & *SchedCalc*; defining constants and income lists
- Session 4** Policy function *Allocate*; defining tax units
- Session 5** System function *DefOutput*; uprating indices, EUROMOD resources
- Session 6** Open Q&A session (optional)

EUROMOD (advanced) tools not covered in this course

ITT

Enhances EUROMOD by incorporating the simulation of indirect taxes

HHoT

Enables the creation of hypothetical households & the generation of corresponding data based on selected household characteristics

PET

Estimates the first-order effects of tax-benefit policies on household incomes

MTR

Enables the calculation of marginal tax rates

NRR

Enables the calculation of net replacement rates

LMA

Allows users to modify the labour market situation of individual observations in the data

TCA

Accounts for varying degrees of income misreporting and income tax compliance scenarios



TRAINING
EUROMOD

Session 1

Introduction to microsimulation
EUROMOD and its language
Handling errors

In this session you will learn about

- Microsimulation models: what they are, how they work and why they are useful
- The basics of EUROMOD and some applications
- The building blocks of the EUROMOD language
- Handling errors when running EUROMOD
- Exercises:
 - 1. Run EUROMOD and use the Statistics Presenter to analyse the results
 - 2. Child benefit reform in Finland

Microsimulation models (MSM)

Modelling techniques that

- Operate at the level of **individual units**
 - in EUROMOD: persons in households
- Apply **rules** to simulate (changes in) the state or behaviour of these units
 - in EUROMOD: taxes and benefits
- Estimate **distributional outcomes** after applying these rules at the micro level
 - in EUROMOD: distribution of disposable income

Microsimulation models (MSM)

Types:

- **Dynamic vs. static** → individual characteristics can be adjusted (dynamic) or not (static) over time
 - EUROMOD is static, although it can incorporate dynamic features through add-ons
- **Behavioural vs. non-behavioural** → models may allow individuals to react to changes (behavioural) or not (non-behavioural)
 - EUROMOD is non-behavioural, although it can be linked to behavioural models

Example: static non-behavioural tax-benefit MSM

- **Tax-benefit rules:** 20% tax, 100 EUR benefit if market income $\leq 1,000$

original data		simulations		
id	market income	tax	benefit	disposable income
1	0	0	100	100
2	1,000	200	100	900
3	1,500	300	0	1,200
4	2,000	400	0	1,600
5	5,000	1,000	0	4,000

Budgetary impact: revenue = 1,900; expenditure = 200

Distributional impact: Gini = 0.3555; S80/S20 = 40; AROP rate = 20%

Specific features of EUROMOD

- Scope
 - Static non-behavioural microsimulation model for the EU-27
 - Households and individuals living in the households
 - Cash benefits, personal taxes and social insurance contributions, accounting for interactions
 - Budgetary, distributional and poverty indicators
- Access
 - Open source and free
 - Extensive documentation and community of users
- Use
 - Microsimulation-oriented language and platform
 - Flexible, transparent and user-friendly interface

Uses of EUROMOD

- Standard EUROMOD
 - Assess the budgetary, distributional and poverty impact of actual, proposed or hypothetical tax-benefit policy changes
 - Evaluate the impact on specific hypothetical households created with the Hypothetical Household Tool (HHoT)
 - Estimate the impact of changes in the population by adjusting the input dataset
- Beyond standard EUROMOD
 - Carry out further analysis by linking EUROMOD with other models (e.g. labour supply, CGE, DSGE)
 - Extend policy scope with additional microdata (e.g. SILC+HBS, HFCS, administrative data)

Key users of EUROMOD



Academic research in (e.g.) public economics & quantitative social policy



International policy organisations

European Commission policy Directorates, ECB, OECD, IMF, World Bank, UNICEF



National governments and public institutions

Ministries, e.g. EE, EL, LT, MT, RO, SK
Central Banks, e.g. EL, ES, IT, LV, PT



Broader audience: EUROMOD is the engine for simplified web-based models

EUROMOD Online - European Commission
<https://euromod-web.jrc.ec.europa.eu/euromod-online>
SORESI - Austrian Ministry for Social Affairs
<http://soresi.sozialministerium.at/soresi>

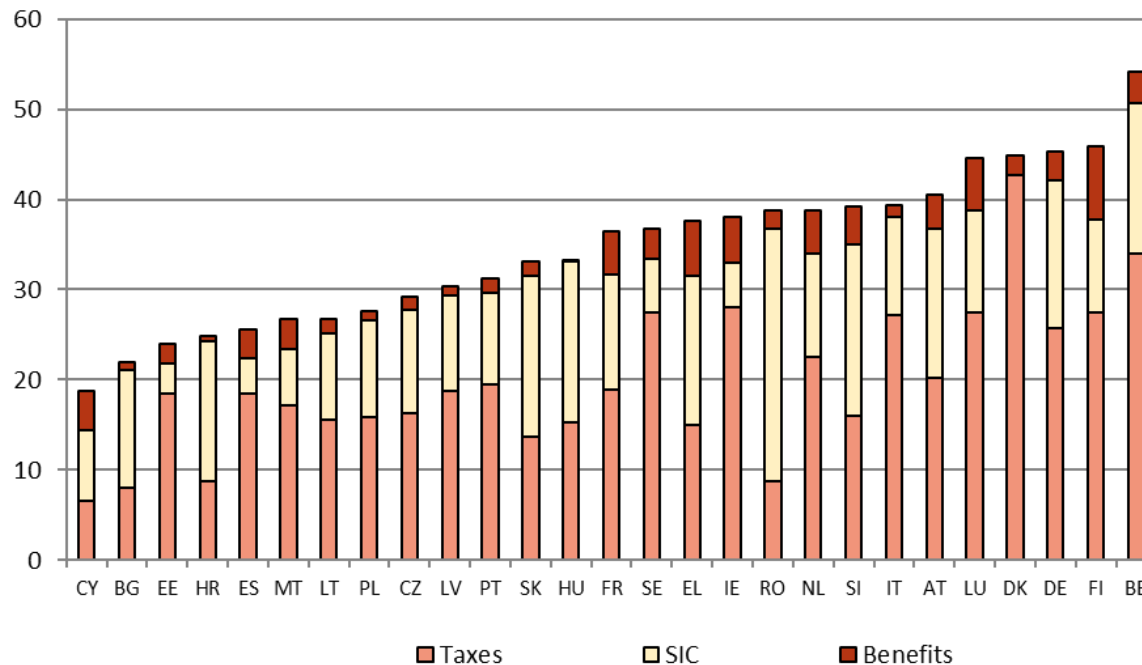


A platform for developing non-EU models

e.g. UKMOD, SOUTHMOD for the Global South
<https://www.microsimulation.ac.uk/euromod/models/>

Example 1: baseline analysis

- Decomposition of mean Marginal Effective Tax Rates by policy in 2018

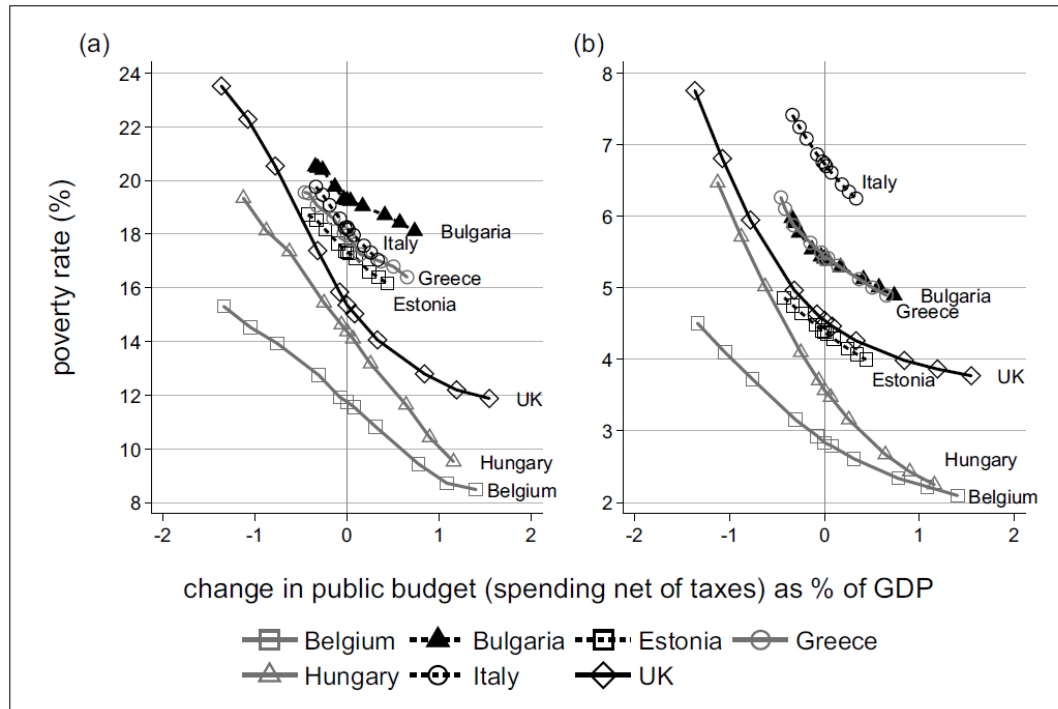


Source: Maier, S., Ricci, M. (2022), EUROMOD baseline report, *JRC Working Papers on Taxation and Structural Reforms*, 1/2022 [\[link\]](#)

Note: METRs measure the potential (dis)incentive to work longer hours

Example 2: policy changes

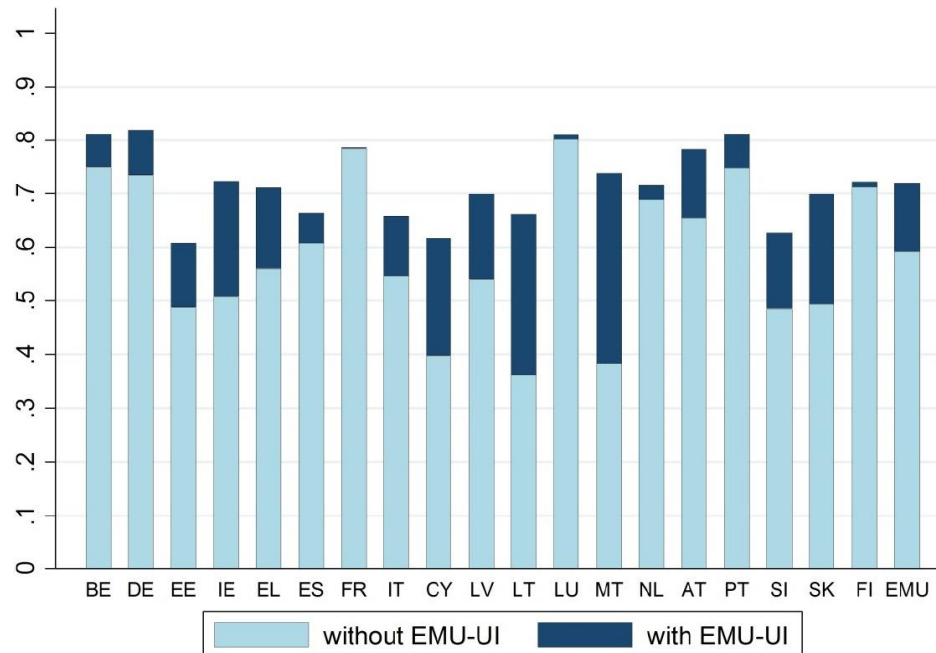
- Poverty-reducing impact of different levels of child benefits



Source: Leventi, C., Sutherland, H., Tasseva, I.V. (2019), "Improving poverty reduction in Europe: What works best where?", *Journal of European Social Policy*, vol. 29(1) 29-43 [\[link\]](#)

Example 3: EU-wide reforms

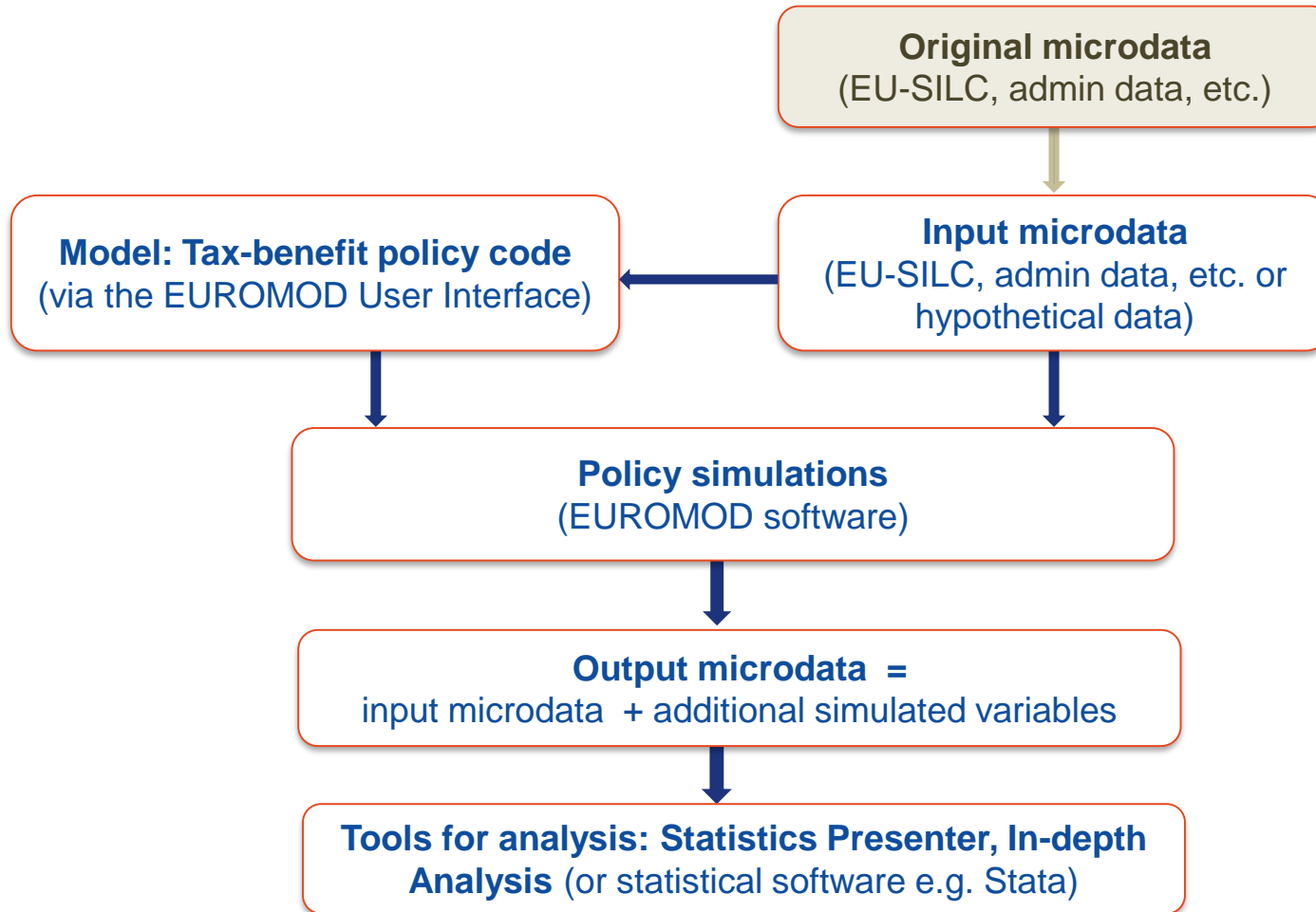
- Income Stabilisation Coefficient (ISC) with and without European Monetary Union-Unemployment Insurance (EMU-UI) benefit



Source: Jara, H.X., Tumino, A. (2021), "The Income Protection Role of an EMU-wide Unemployment Insurance System: the Case of Atypical Workers", *EUROMOD Working Papers*, 6/21 [\[link\]](#)

Note: ISC measures percentage of earnings loss that is retained through tax-benefit system






EUROMOD workflow



EUROMOD project folder

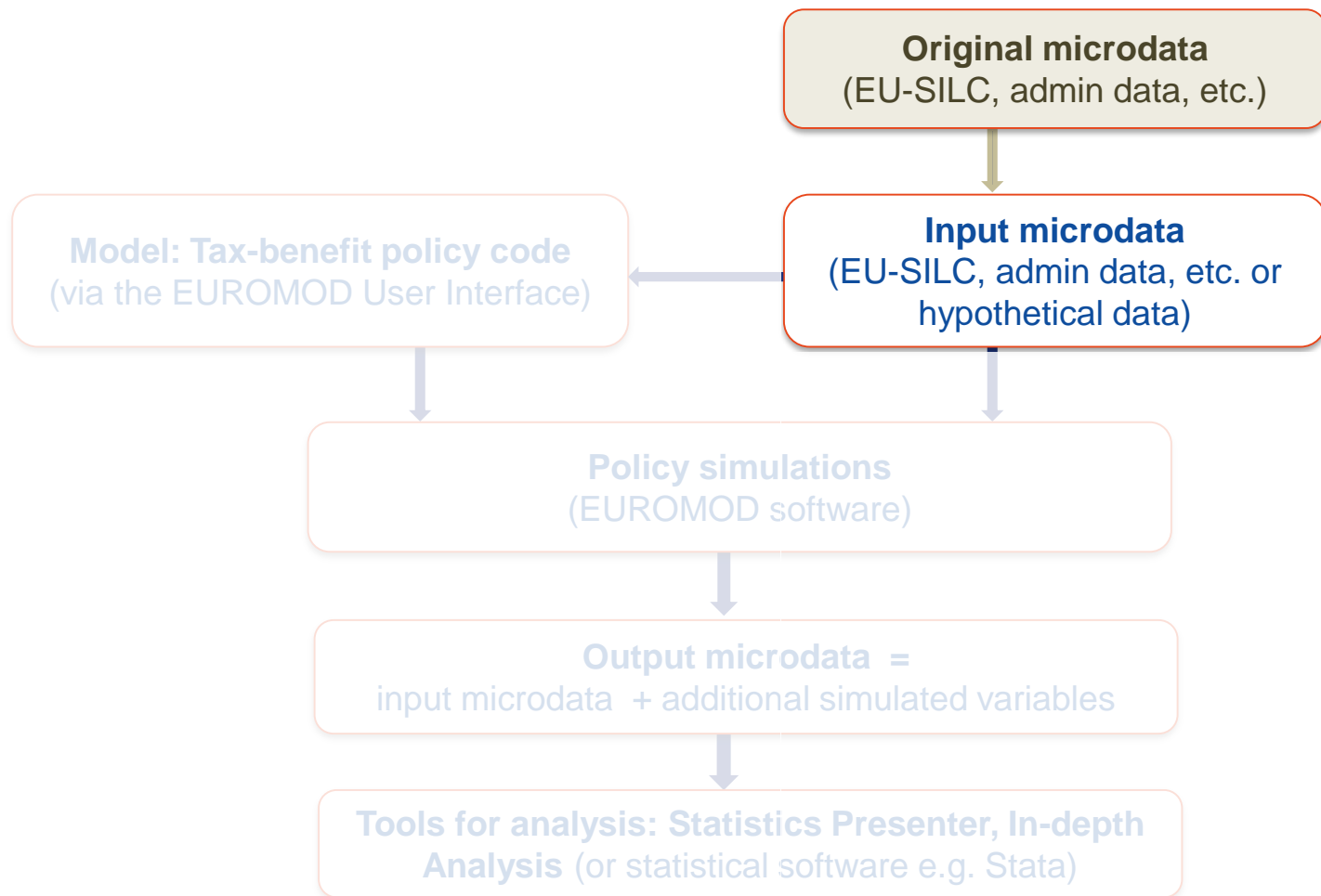
› EUROMOD_I4.0+ › EuromodFiles_I4.0+

Name

-  **Input** → Default folder to store input **data**
-  **Log** → EUROMOD log with all versions
-  **Output** → Default folder to save output data
-  **XMLParam** → **Model**, where policies are coded!
-  **Licence.txt** → Open source license

EUROMOD in this course

- Input microdata
 - Training data
- Model (tax-benefit code):
 - Version I6.0++
 - Models for all EU member states: policy years (systems) for 2005/07-2023 for most countries and 2011-2023 for Croatia
- EUROMOD Software version:
 - Version 3.6.19



INPUT MICRODATA

Sources of input data

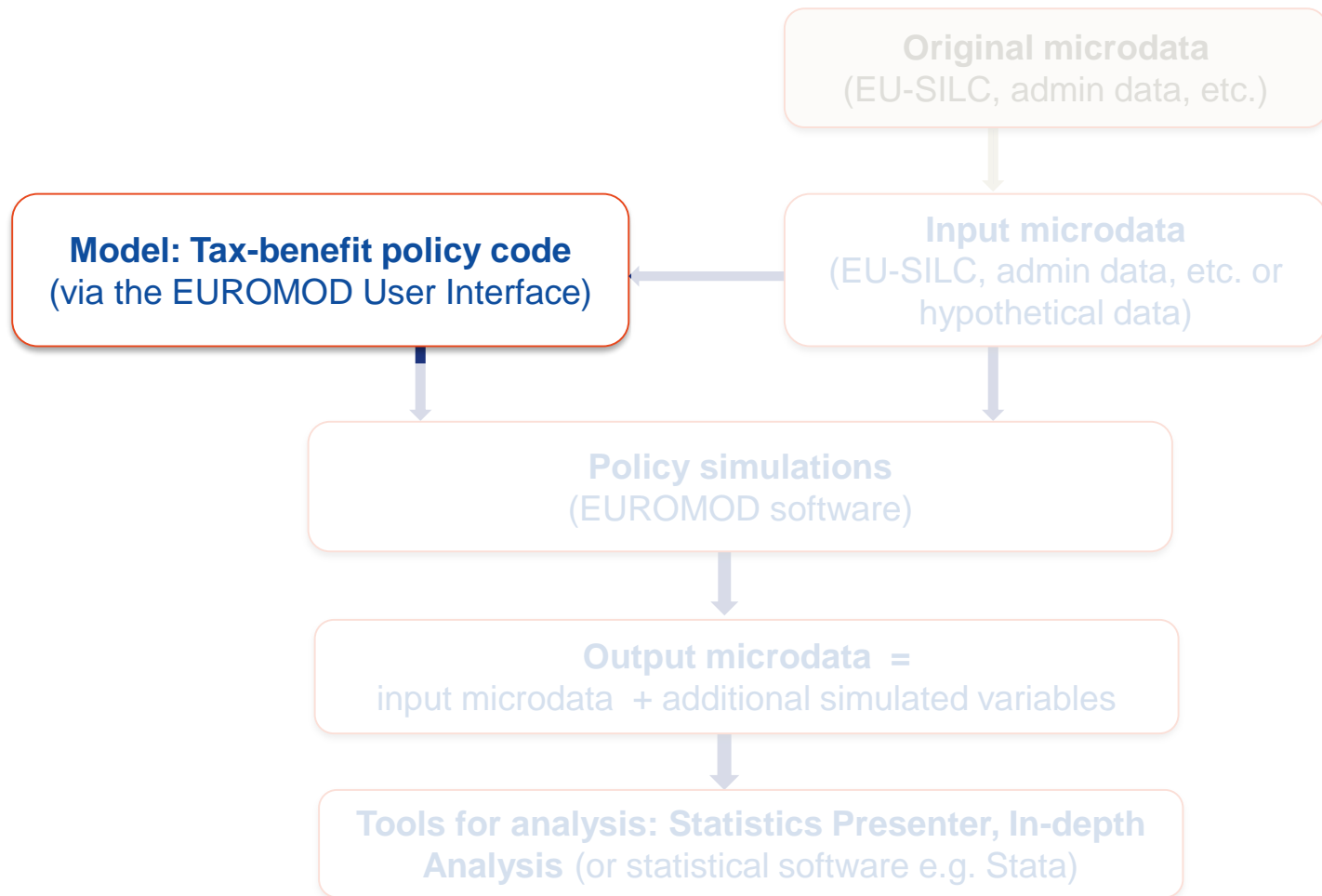
- Default datasets: based on survey microdata from European Union Statistics on Income and Living Conditions ([EU-SILC](#)) [[Eurostat authorisation](#)]
- Hypothetical data mimicking EU-SILC-based data:
 - EUROMOD's [Hypothetical Household Tool \(HHoT\)](#) allows to generate input files with model households
 - Synthetic training dataset included with the model
- Any other data source can be adapted to be used in EUROMOD (e.g. [HFCS](#), [administrative microdata](#)), following a specific structure and modelling conventions

Characteristics of default input data

- Variables: demographic, labour, income, assets, expenditure
- Gross incomes, at the individual level
- Monetary variables recoded to (average) monthly basis
- Identification close family relationships within the household (mothers, fathers and partners)
- No missing values

A typical input dataset

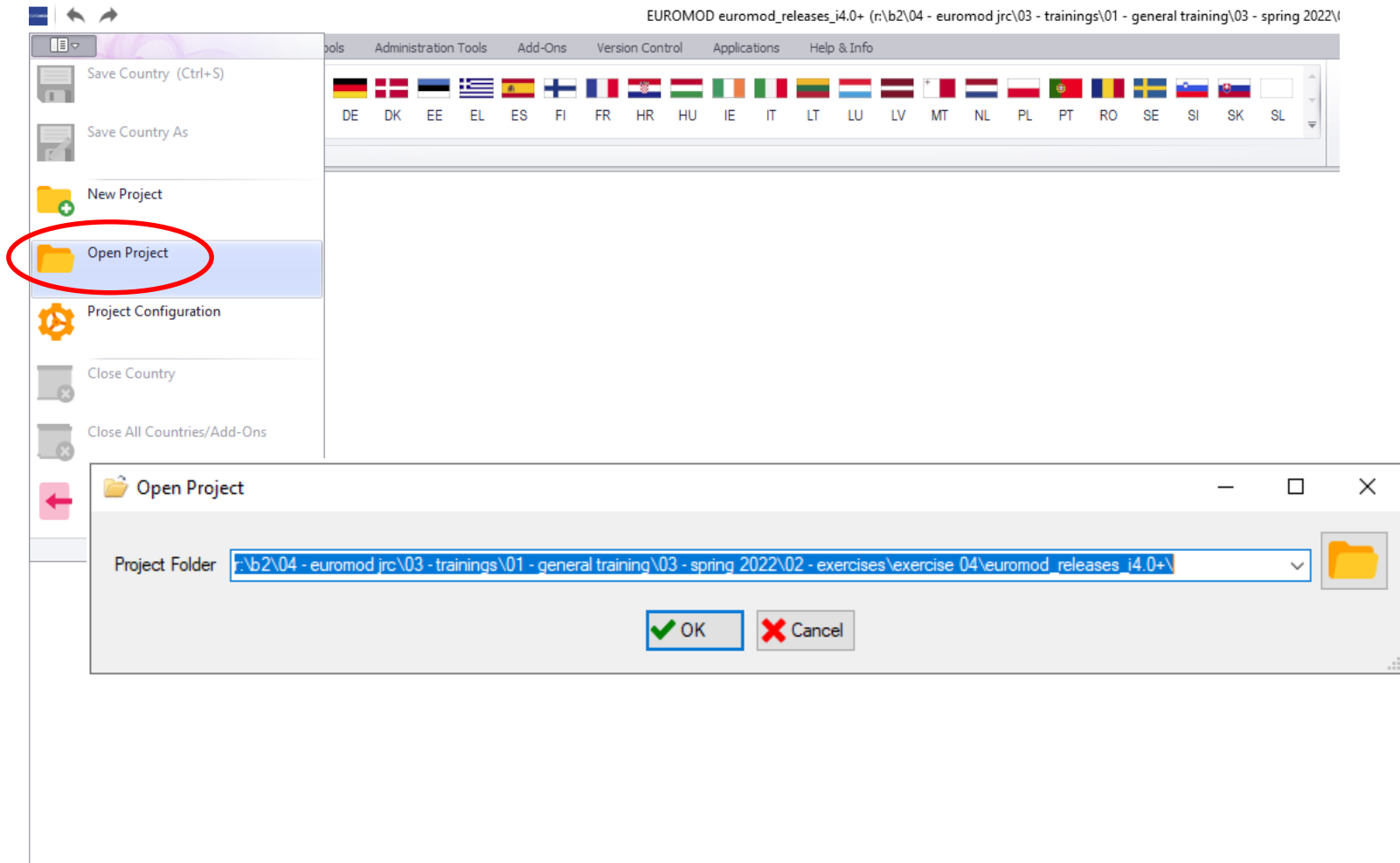
----- identifiers -----					age	wages	pensions
idhh	idperson	idpartner	idmother	idfather	dag	yem	poa
1	101	102	0	0	65	0	2017.546
1	102	101	0	0	60	0	0
1	103	0	102	101	30	0	0
1	104	0	102	101	28	157.4188	0
2	201	202	0	0	29	1075.353	0
2	202	201	0	0	25	951.3614	0
2	203	0	202	201	3	0	0
2	204	0	202	201	2	0	0
3	301	302	0	0	72	0	1627.492
3	302	301	0	0	59	0	0
5	501	0	0	0	86	0	1539.242



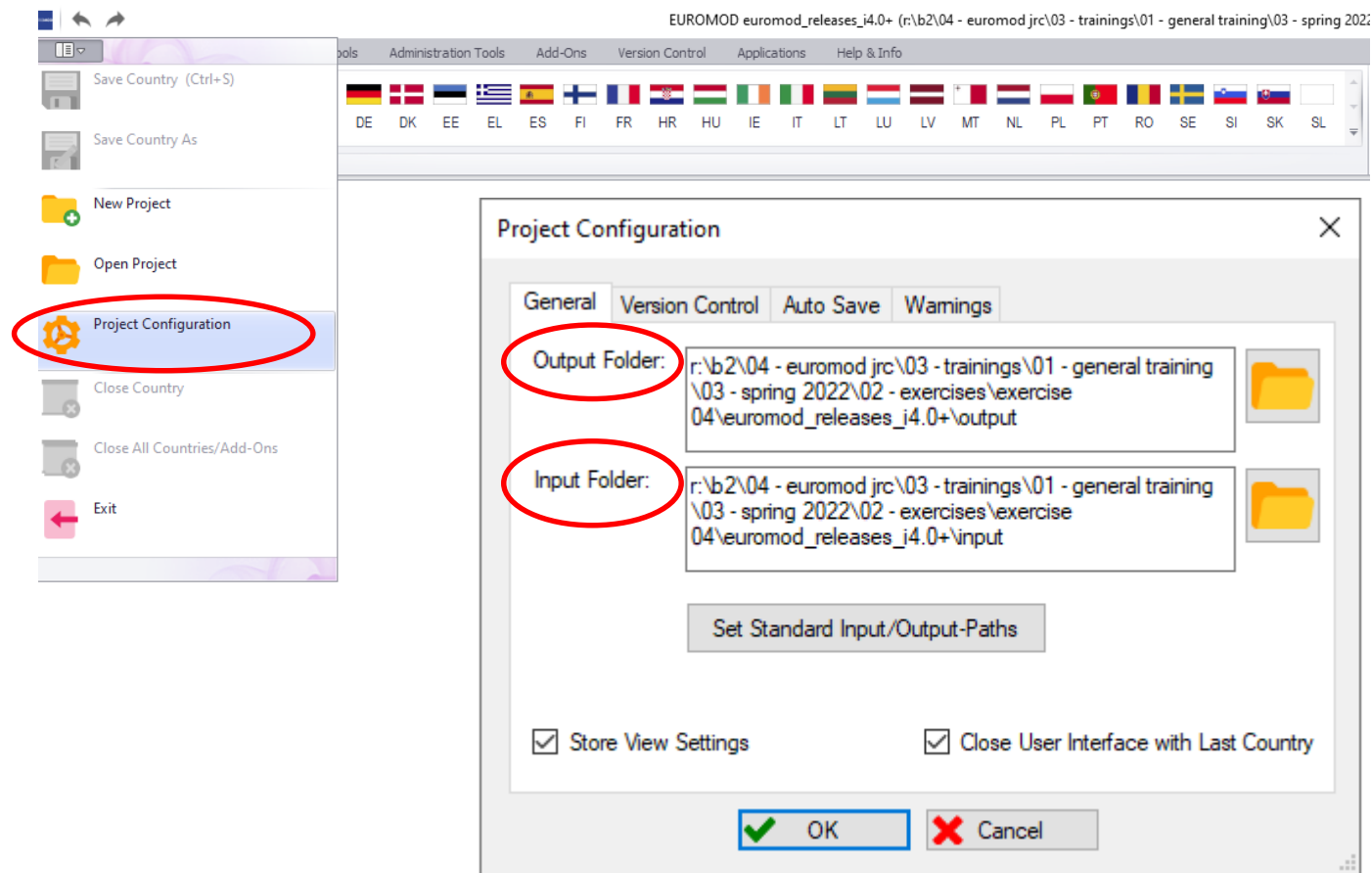
MODEL: TAX-BENEFIT POLICY CODE

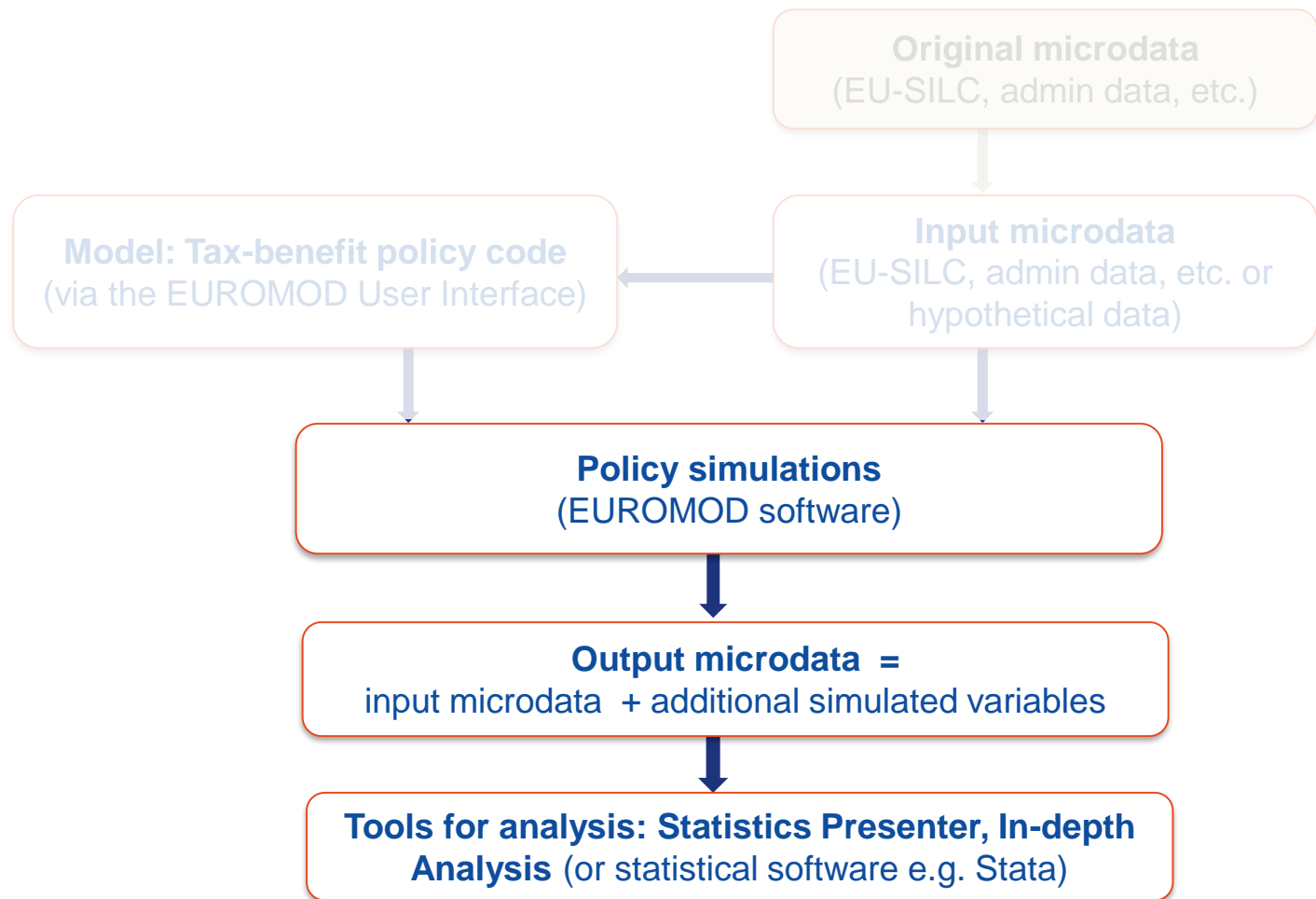
VIA THE EUROMOD USER INTERFACE (UI)

Opening a project



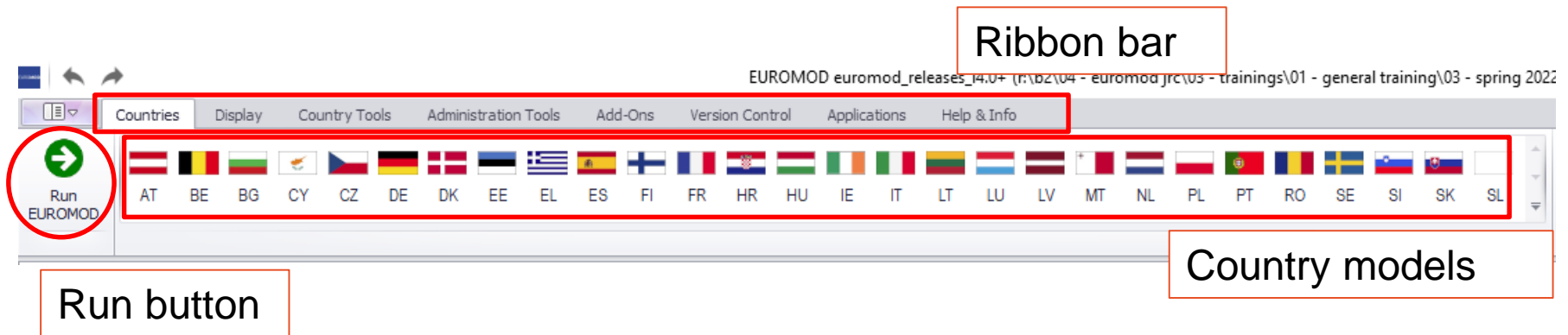
Configuring the output/input folders





POLICY SIMULATIONS, OUTPUT MICRODATA & TOOLS FOR ANALYSIS

Selecting and running a country



- When running the model you may get:
 - Warnings → the model runs but informs you about an issue
 - Errors → the model aborts
- In both cases you get precise information [more information in the backup slides]

A typical output dataset

----- identifiers -----					age	wages	pensions	PIT
idhh	idperson	idpartner	idmother	idfather	dag	yem	poa	tin_s
1	101	102	0	0	65	0	2017.546	403.5092
1	102	101	0	0	60	0	0	0
1	103	0	102	101	30	0	0	0
1	104	0	102	101	28	157.4188	0	31.48376
2	201	202	0	0	29	1075.353	0	215.0706
2	202	201	0	0	25	951.3614	0	190.2723
2	203	0	202	201	3	0	0	0
2	204	0	202	201	2	0	0	0
3	301	302	0	0	72	0	1627.492	325.4984
3	302	301	0	0	59	0	0	0
5	501	0	0	0	86	0	1539.242	307.8484

Analysing the output

- Text file containing the output data: all input data variables plus the variables simulated by the model (*_s)
- File stored in the Output folder and called (for the baseline model) [cc]_[year]_std (e.g. pt_2006_std)
- EUROMOD-embedded tools Statistics Presenter and In-depth Analysis can be used to produce several indicators [more on indicators in backup slides]
- The file can be also imported from any statistical software for further analysis

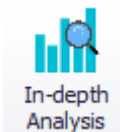
Tools to analyse output

Two tools to obtain summary statistics from EUROMOD output microdata:



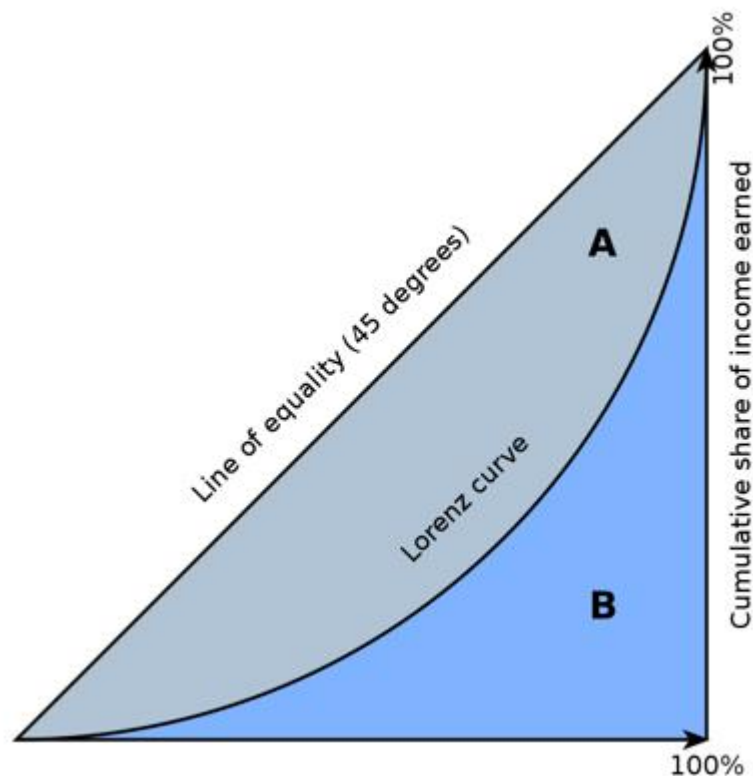
Statistics Presenter

- Descriptive statistics of income and population
- Predefined tables for budgetary and distributional impact



In-depth Analysis tool

- Customised tables for budgetary and distributional impact of reforms



STATISTICAL AND INCOME INDICATORS

Statistical and income concepts

Disposable income



Market Income

- + Salaries
- + Self-employment income
- + Investment income
- + Property income
- + etc.

Taxes and SIC

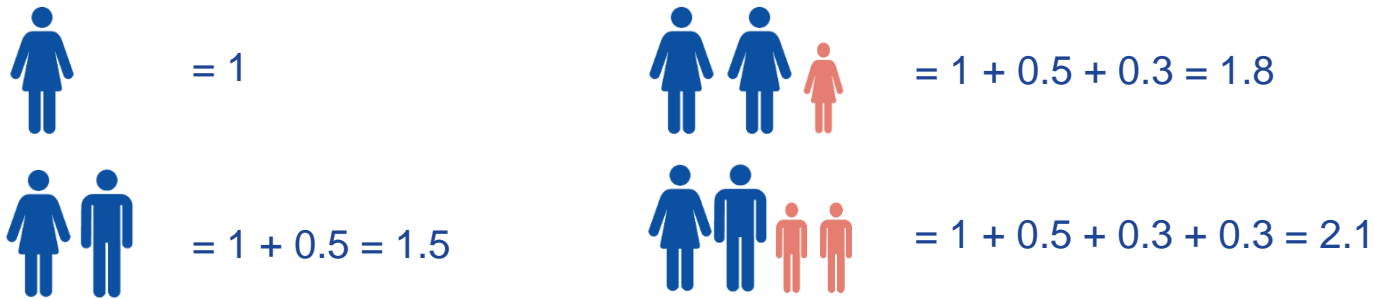
- Personal Income Tax
- Employee Social Insurance Contributions
- etc.

Social transfers

- + Pensions
- + Family benefits
- + Minimum Income Schemes
- + etc.

Statistical and income concepts

- Larger household size → higher income needs, but...
- ... sharing expenses and assets → needs do not double when size doubles



OECD-modified equivalence scale

First adult = 1

Each additional member ≥ 14 = 0.5

Each additional member < 14 = 0.3

Statistical and income concepts

Equivalised disposable income



= 1

DI = 10,000

EDI = 10,000/1 = 10,000



= 1 + 0.5 + 0.3 = 1.8

DI = 18,000

EDI = 18,000/1.8 = 10,000



= 1 + 0.5 = 1.5

DI = 30,000

EDI = 30,000/1.5 = 20,000



= 1 + 0.5 + 0.3 + 0.3 = 2.1

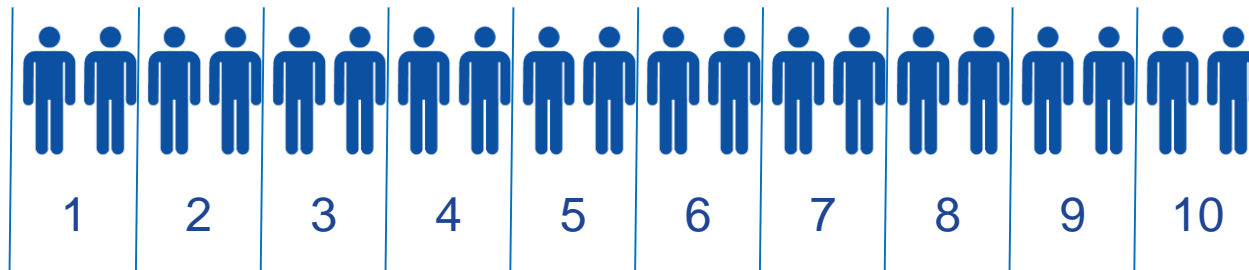
DI = 31,500

EDI = 31,500/2.1 = 15,000

Statistical and income concepts

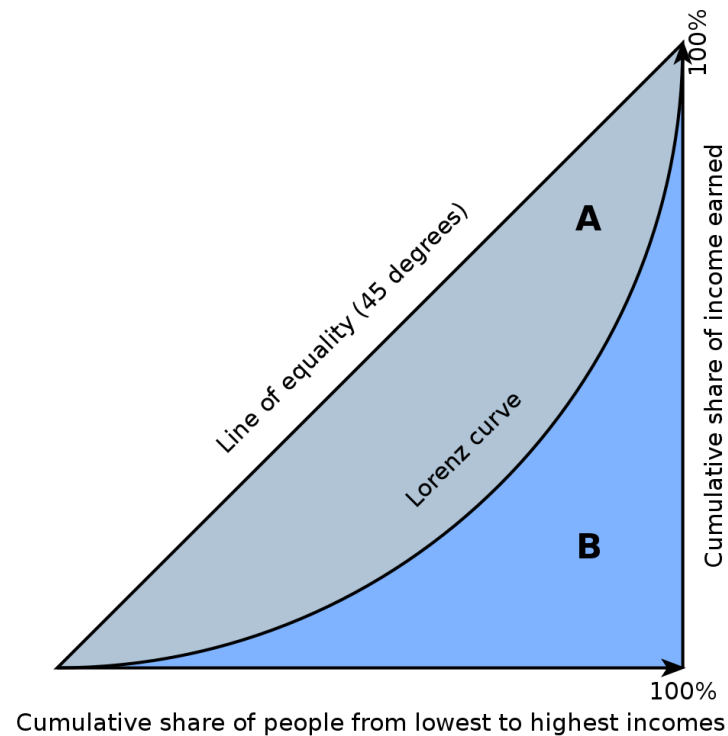
Income deciles

Equivalised disposable income (in ascending order)



Statistical and income concepts

Inequality

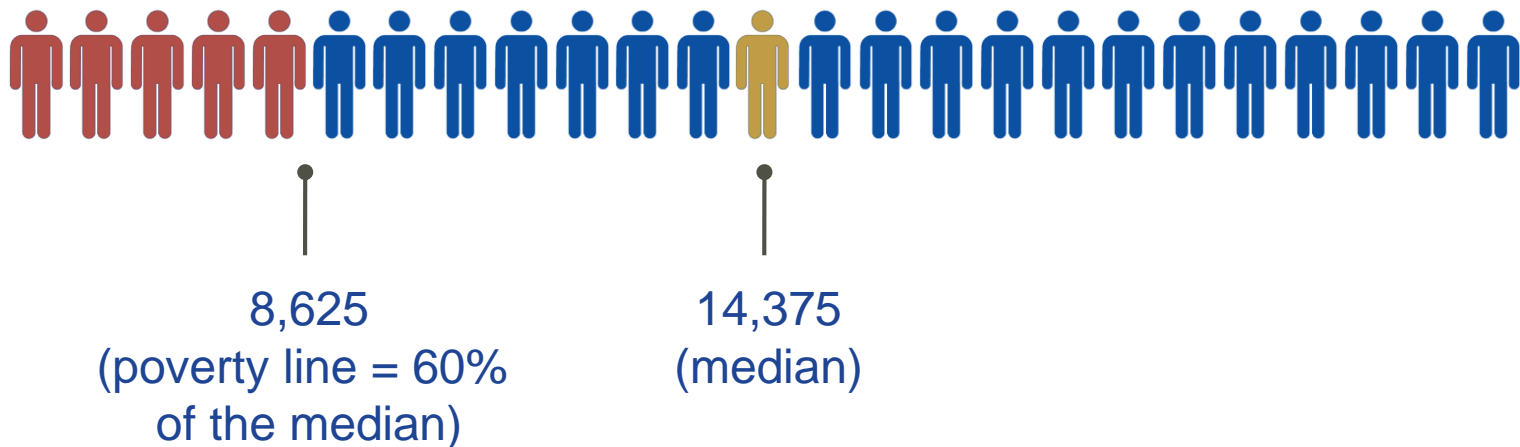


$$\text{Gini coefficient} = \frac{A}{A+B} = 2A$$

Statistical and income concepts

At-risk-of-poverty rate

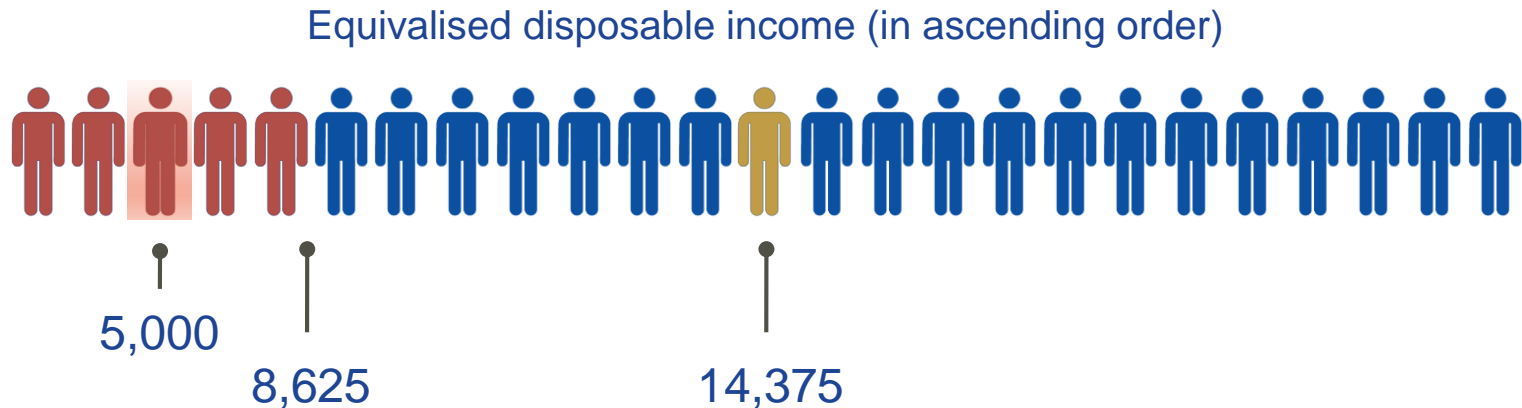
Equivalised disposable income (in ascending order)



$$\text{AROP rate} = 5/25 = 20\%$$

Statistical and income concepts

Median at-risk-of-poverty gap



Difference between the poverty line and the median equivalised disposable income of people at risk of poverty, divided by the poverty line. In our example:

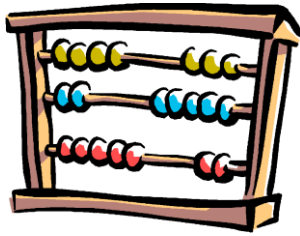
$$\text{Median AROP gap} = \frac{8,625 - 5,000}{8,625} = 42\%$$



Q&A

Questions



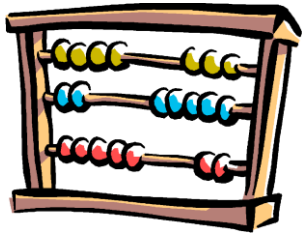


Exercise 1

Run EUROMOD and use the Statistics Presenter to analyse the results

Run the policy systems for Austria (systems: 2019 and 2020) as well as for Finland, Italy and Spain (system: 2020 only). Then use the Statistics Presenter tool to analyse the results and make cross-country comparisons on inequality and poverty.

We will do the exercise together!



Exercise 1

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.



Summary of Exercise 1

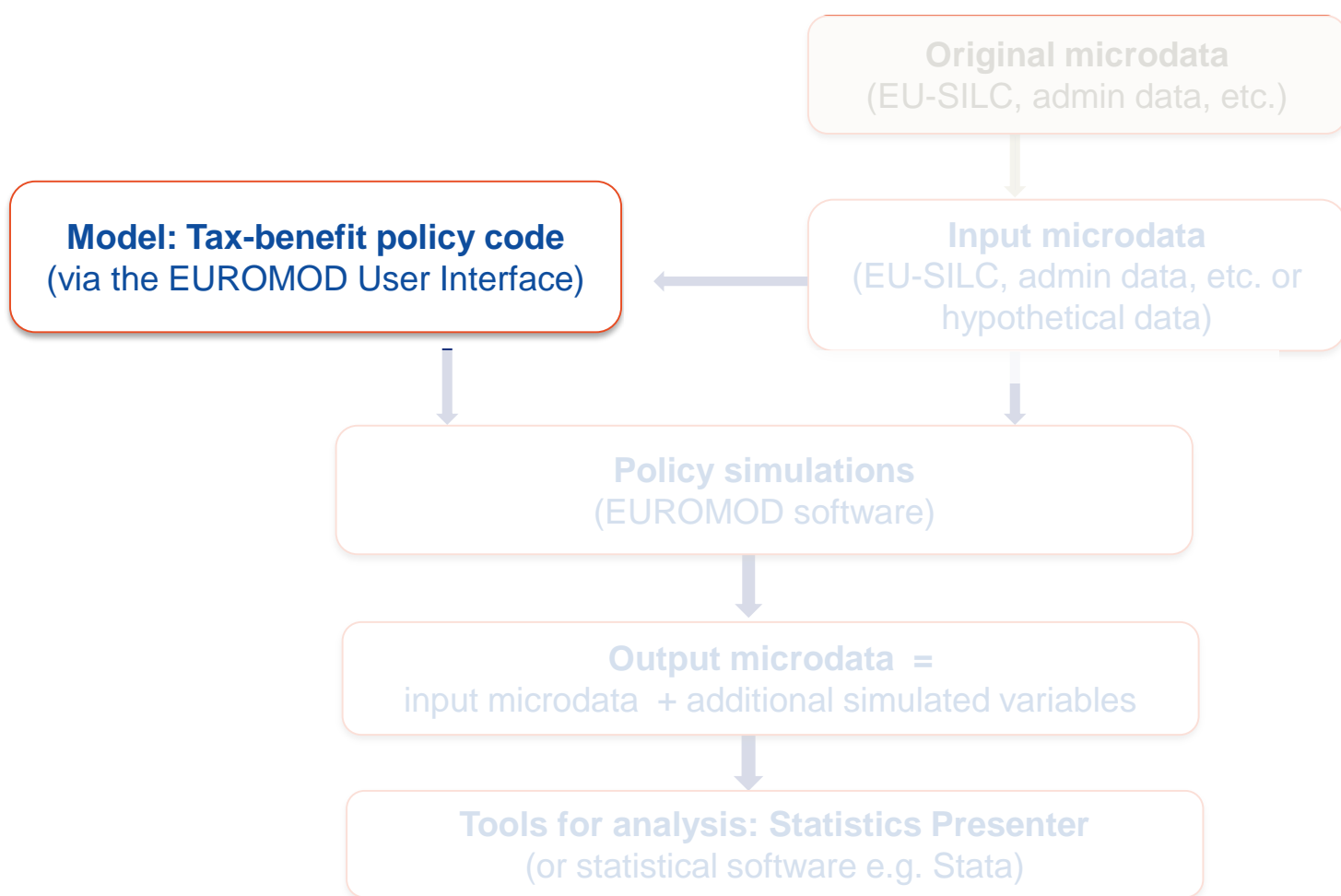
- You learned how to:
 - Open the model
 - Run a single system or several systems at the same time
 - Analyse the output microdata with the Statistics Presenter, using the:
 - Default option
 - MultiSystem option



Q&A

Questions

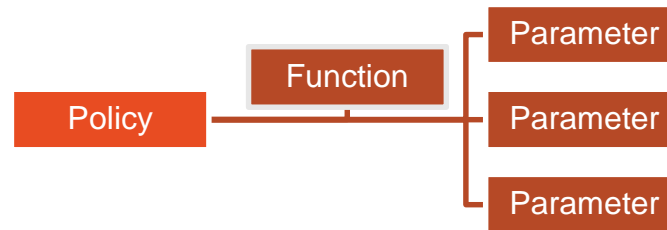




EUROMOD TAX-BENEFIT MICROSIMULATION LANGUAGE

Building blocks of EUROMOD

- Building blocks
 - Policies
 - Functions
 - Parameters

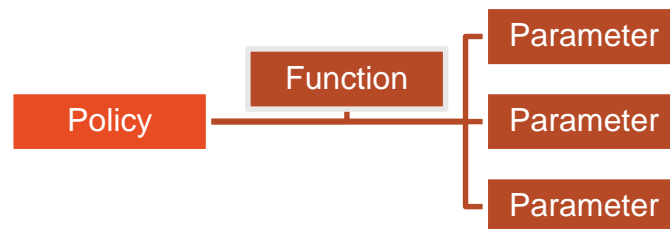


Building blocks (1)

- **Policy column** includes **policy** name, **function** and **parameter** names → order of policies and functions = order of calculations
- **System columns** include the values of the tax-benefit parameters by policy year
- **Comments column** includes policy description

Policy column	System columns			Comments column
	System#1	System#2	System#3	
Policy#1 name	<i>policy switch</i>	<i>policy switch</i>	<i>policy switch</i>	Policy description
Function#1	<i>function switch</i>	<i>function switch</i>	<i>function switch</i>	Function description
Parameter#1	value	value	value	parameter description
Parameter#2	value	value	value	
Function#2	<i>function switch</i>	<i>function switch</i>	<i>function switch</i>	
Parameter#1	value	value	value	
Parameter#2	value	value	value	
Parameter#3	value	value	value	

Policies



- a **block of functions** completes a policy simulation
- **policy names** end (usually) with the country acronym
 - policies can have any name – but good to follow naming conventions
- **comments** (in the comments column) explain what policies intend to simulate
- policies are run **in the order they appear**

Policies: example

policy systems

Cyprus - EUROMOD euromodfiles_i4.0+ (c:\users\chrysa\desktop\euromod_i4.0+\euromodfiles_i4.0+\)

Run EUROMOD

Cyprus loaded

AT BE BG CY CZ DE DK EE EL ES FI FR HR HU IE IT LT LU LV MT NL PL PT RO SE SI SK SL

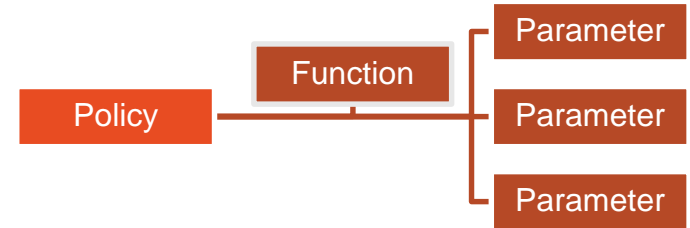
open country by clicking on a flag

Policy	Grp	cy_2019	cy_2020	cy_2021	Comment
11 yem_cy		switch	switch	switch	DEF: Minimum Wage
12 neg				on	DEF: recode to 0 negative values of yse
13 yem				on	BEN: Wage compensation scheme for employees called "Special unemployment benefit scheme for employees".
14 ysecomp_cy				on	BEN: Wage compensation scheme for self-employed called "Special unemployment benefit scheme for self-employed".
15 paycut_cy		on	on	on	INC: pay cut for public employees and pensioners
16 tscee_cy		on	on	on	SIC: Employee social insurance contribution (Ασφαλιστικές εισφορές εργαζομένων)
17 tscer_cy		on	on	on	SIC: Employer SIC (Ασφαλιστικές εισφορές εργοδοτών)
18 tscse_cy		on	on	on	SIC: Self employed social Insurance contribution (Ασφαλιστικές εισφορές αυτοεργαζόμενων)
19 tscpe_cy		on	on	on	SIC: Contributions by pensioners
20 tscot_cy		on	on	on	SIC: Contributions by other income earners
21 tscgv_cy		on	on	on	SIC: Government social insurance contribution (Κυβερνητικές ασφαλιστικές εισφορές)

comments

policies

Functions



Function name

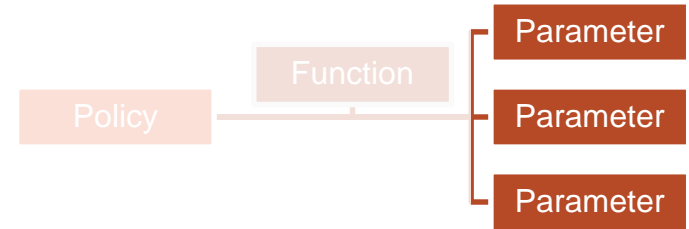
switch: on / off / n/a

Policy	Grp/No	EL_2020	EL_2021	Comment
<div> <div>fx</div> <div>ArithOp</div> <div> <div>formula</div> <div>output_var</div> <div>TAX_UNIT</div> </div> </div>		<div> <div>on</div> <div>399.25#m</div> <div>bunct_s</div> <div>tu_individual_el</div> </div>	<div> <div>on</div> <div>399.25#m</div> <div>bunct_s</div> <div>tu_individual_el</div> </div>	benefit amount: basic

parameter names

parameter values for each system

Parameters



- Can be:
 - common to several functions or specific to one function
 - compulsory (i.e. error generated if not used) or optional
 - which parameters are compulsory/optional depends on the function
- Order of parameters in a function is not important
 - but order of functions in a policy is!
- Manipulated via context menu (right-click)
 - only relevant parameters for the given function are shown
- Drag & drop can be used

Parameters: example

policy column		system columns			comments columns
Policy	Grp/No	cy_2019	cy_2020	cy_2021	Comment
policy name bchlp_cy		on	on		BEN: Single parent benefit
fx DefConst		on	on		
\$bchlp_it1	1	39000#y	39000#y	39000#y	family income threshold
\$bchlp_it2	2	49000#y	49000#y	49000#y	Family income threshold
\$bchlp_it3	3	n/a	n/a	n/a	
\$bchlp_it4	4	n/a	n/a	n/a	
\$bchlp_it5	5	n/a	n/a	n/a	
\$bchlp_it6	6	n/a	n/a	n/a	
\$bchlp_amt1	8	180#y	182.72#y	182.72#y	Benefit level
\$bchlp_amt2	9	160#y	162.42#y	162.42#y	Benefit level
\$bchlp_amt3	10	n/a	n/a	n/a	
\$bchlp_amt4	11	n/a	n/a	n/a	
\$bchlp_amt5	12	n/a	n/a	n/a	
\$bchlp_amt6	13	n/a	n/a	n/a	
fx Elig		on	on	on	Eligibility for the benefit
elig_cond		IsLoneParent & bch_s>0	IsLoneParent & bch_s>0	IsLoneParent & bch_s>0	Identify monoparental families receiving child benefit
TAX_UNIT		tu_bch_cy	tu_bch_cy	tu_bch_cy	

Policy spine

- The order of policies is called the **spine**
- The models for all EU countries follow the same structure of the policy spine
- See the EUROMOD Country Reports for more details

Order of policies in the spine

DEFINITIONS

- define certain parameters or concepts; or do some pre-policy calculations
- e.g. define assessment units; index (uprate) earnings if input microdata year lags behind policy year

TAX AND BENEFIT CALCULATIONS

- (almost) every tax and benefit implemented in a separate policy; policy name according to our naming convention
- e.g. bch00_ee (Child Allowance → b: benefit, ch: child, 00: main)

OUTPUT

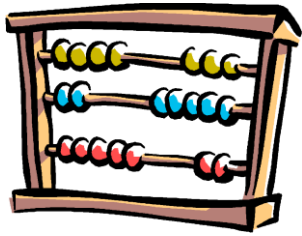
- defines what variables to be included in the output microdata and assessment unit for the results
- e.g. include both simulated and non-simulated variables; output individual-level results



Q&A

Questions



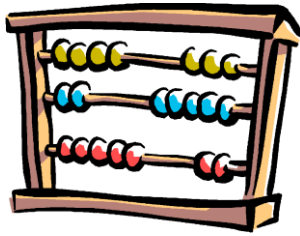


Exercise 2

Child benefit reform

Implement a Child benefit reform in Finland in 2021. Increase the benefit amount for the first child from €94.88 to €100 per month.

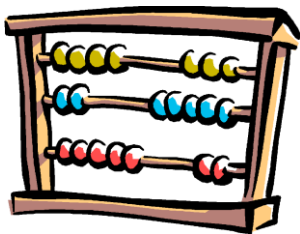
We will do the exercise together!



Exercise 2

Steps

- You will make a copy of the FI 2021 system and call it FI_2021_E2
- Then, you will modify the reform system FI_2021_E2 as follows:
 - Go to the Child Benefit policy (bch_fi) and open the functions DefConst and BenCalc
 - Change the parameter value of \$bch1 from 94.88#m to 100#m
 - Save your changes
- Run the model and use the *Statistics Presenter* – *Baseline/Reform* option to analyse the distributional effect of the reform



Exercise 2 – solution

2.1.a Mean household income (monthly) by decile groups ?

	FI_2021 (base)	FI_2021_E2	Difference to base	% Difference to base
Decile 1	1,248.48	1,248.82	0.33	0.03 %
Decile 2	1,716.76	1,717.48	0.72	0.04 %
Decile 3	2,016.92	2,017.61	0.69	0.03 %
Decile 4	2,529.37	2,530.41	1.03	0.04 %
Decile 5	3,032.09	3,033.39	1.31	0.04 %
Decile 6	3,378.66	3,379.97	1.30	0.04 %
Decile 7	3,722.91	3,724.11	1.20	0.03 %
Decile 8	4,285.10	4,286.31	1.20	0.03 %
Decile 9	5,015.46	5,016.59	1.13	0.02 %
Decile 10	7,914.23	7,915.16	0.93	0.01 %
All	3,283.44	3,284.38	0.94	0.03 %
Poor	1,296.92	1,296.46	-0.46	-0.04 %

STATISTICS PRESENTER OUTPUT



Summary: Exercise 2

- You learned:
 - how to add a reform system
 - about basic options such as (un)hiding systems, expanding a policy and conditional formatting
 - how to analyse the baseline and reform output microdata with the Statistics Presenter, using:
 - Baseline/Reform template



Q&A

Questions



Configuration	Status	Time	... Show		Stop
training_data (FI_2021reform); BTA=on;MWA=off;	aborted	14:53:00 - 14:53:00 (00h:00m:00s)	Run Log	Error Log	Stop

ERROR-LOG training_data (FI_2021reform); BTA=on;MWA=off:
error: 41.1.1 bch_fi/DefConst/\$bch1 (9DD0683D-2806-4C60-BEC7-D55F72C48C85): Formula error: Expression expected.

HANDLING ERRORS

Handling errors (1)

bch_fi		on	on	BEN: Child benefit
fx	DefConst	on	on	Constants for Child Care Benefits
	\$bch1	94.88#m	100#	
	\$bch2	104.84#m	104.84#m	
	\$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

typo in the syntax:
100# instead of 100#m

Handling errors (2)

⚙️ EUROMOD Run started 18/04/2022 14:56:00 and finished 18/04/2022 14:56:01

Configuration	Status	Time	... Show	Stop
training_data (FI_2021reform); BTA=on;MWA=off;	aborted	14:56:00 - 14:56:01 (00h:00m:00s)	Run Log	Error Log

Handling errors (3)

⚙️ EUROMOD Run started 18/04/2022 14:56:00 and finished 18/04/2022 14:56:01

— □ ×

Configuration	Status	Time	... Show		Stop
training_data (FI_2021reform); BTA=on;MWA=off;	aborted	14:56:00 - 14:56:01 (00h:00m:00s)	Run Log	Error Log	Stop

ERROR-LOG training_data (FI_2021reform): BTA=on;MWA=off:
error: 41.1.1 bch_fi/DefConst/\$bch1 9DD0683D-2806-4C60-BEC7-D55F72C48C85) Formula error: Expression expected.

↑ location of the error:
policy, function, parameter

↑ unique identifier for the exact
place of the error in the model

Handling errors (4)

Finland - EUROMOD euromodfiles_i4.0+ (c:\users\chrysa\desktop\euromod_i4.0+\euromodfiles_i4.0+\)

Countries Display **Country Tools** Administration Tools Add-Ons Version Control Applications Help & Info

Add System Delete System(s) Clean Up Systems Import System(s) Export System(s) Admin Country Set Switches Add to, switch on Add to, switch off

search policies, functions or parameters by their identifier

	Policy	Grp/No	FI_2021	FI_2021reform	
41	● bch_fi		on	on	BEN: Child benefit
41.1	fx DefConst		on	on	Constants for Child Care Benefits
41.1.1	\$bch1		94.88#m	100#	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	fx Elig		on	on	Child benefit: eligibility
41.3	fx BenCalc		on	on	Child benefit: amount
41.4	fx Elig		on	on	Eligibility for Lone parent supplement
41.5	fx ArithOp		on	on	Child benefit: single parent supplement
42	● bsa00_fi		on	on	BEN: Local authority income support

Handling errors (5)

The screenshot displays the EUROMOD software interface. The top menu bar includes 'Countries', 'Display', 'Country Tools', 'Administration Tools', 'Add-Ons', 'Version Control', 'Applications', and 'Help & Info'. Below this, there are several toolbars with icons for various functions like 'Add System', 'Delete System(s)', 'Clean Up Systems', 'Import System(s)', 'Export System(s)', 'Admin Country', 'Set Switches', 'Add to, switch on', 'Add to, switch off', 'Remove from', 'Set Visible', 'Set Not Visible', and 'Expand'.

The main table shows a list of policy parameters. The first column is 'Policy', followed by 'Grp/No', 'FI_2021', 'FI_2021reform', and 'Comment'. The row for 'bch_fi' is highlighted in yellow. Below it, the 'DefConst' section is expanded, showing a list of parameters: '\$bch1', '\$bch2', '\$bch3', '\$bch4', '\$bch5', and '\$bchsup'. The 'Comment' column for these parameters includes 'BEN: Child benefit', 'Constants for Child Care Benefits', and 'Child benefit for the 2nd child'.

A blue box with the text 'copy unique identifier from error log and paste it here' is overlaid on the table. A red arrow points from this box to the 'Identifier' field in the 'Search by Identifier' dialog box. The dialog box has a title bar 'Search by Identifier' and a close button. The 'Identifier' field contains the text '9DD0683D-2806-4C60-BEC7-D55F72C48C85'. Below the field are 'Find' and 'Cancel' buttons.

Policy	Grp/No	FI_2021	FI_2021reform	Comment
41				
41.1				
41.1.1				
41.1.2				
41.1.3				
41.1.4				
41.1.5				
41.1.6				
41.2				
41.3				
41.4				
41.5				

Handling errors (6)

- Output folder: error log file (text format)
- Same info as in the running dialog box
- Error logs contain time stamp of their creation
- Info about EUROMOD version, policy system where error occurred and dataset used

```
20220419102812_089fc7ee324d4740b15b10859c4ff22d_EUROMOD_Log - Notepad
File Edit Format View Help
===== EUROMOD LOG =====
Software-Version: 3.4.7
Project: euromodfiles_i4.0+
Start: 19 April 2022 10:28:11
End: 19 April 2022 10:28:12
Duration: 0.5839648s
Output-Path: c:\users\chrysa\desktop\euromod_i4.0+\euromodfiles_i4.0+\output\
-----
===== RUN-LOG =====
Run-Id Status System Database Start End Duration Currency Exchangerate BTA MWA Non-Default Ou
7028fa66-ab00-4cbf-95e8-bf83c3f1c0f2 aborted FI_2021reform training_data 19 April 2022 10:28:11 19 April 2022 10:28:12 0.5839
-----
===== ERROR-LOG =====
Run-Id: 7028fa66-ab00-4cbf-95e8-bf83c3f1c0f2
ERROR: 41.1.1 bch_fi/DefConst/$bch1 (9DD0683D-2806-4C60-BEC7-D55F72C48C85) Formula error: Expression expected.
```



Q&A

Questions





Session 2

Variable types

System function *DefVar*

Policy functions *Elig* & *ArithOp*

In this session you will learn about

- Variable types and system function *DefVar*
- Benefit/tax amount calculations with function *ArithOp*
- Implementing eligibility conditions with function *Elig*
- Function parameters: amounts, formulas and queries
- Exercises:
 - 3a. Child allowance reform in Estonia
 - 3b. Child benefit reform in Greece

Variables: types

- **Standard** - following naming conventions
 - Household and individual characteristics
 - Incomes: market incomes, simulated and non-simulated benefits and taxes
 - Assets and expenditures
- **Intermediate**
 - Used to save the result of an intermediate calculation (e.g. an eligibility condition, a subcomponent of a tax liability etc.)

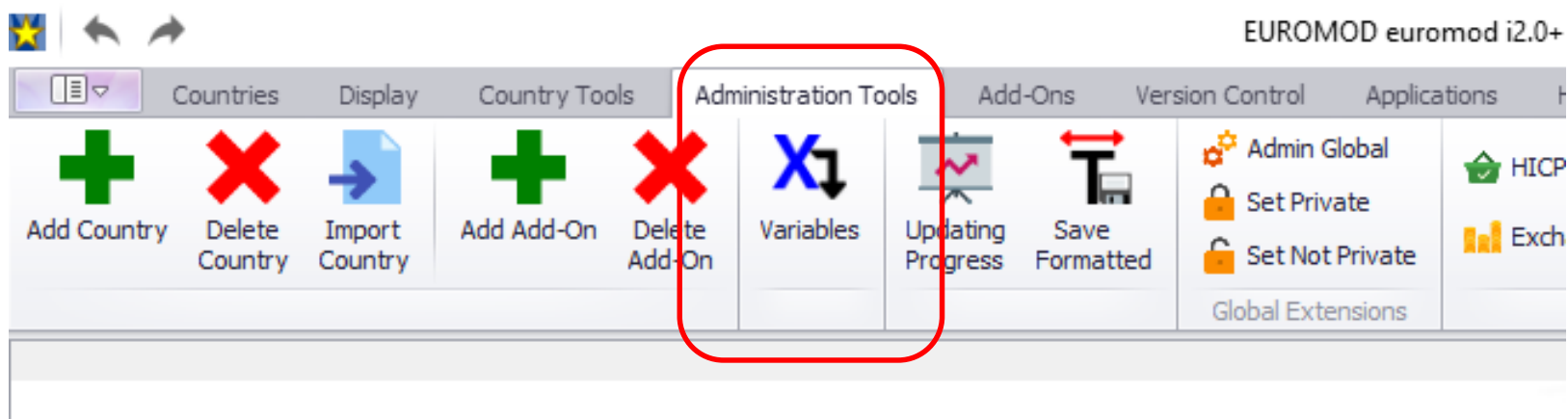
Variables: naming conventions (1)

- Applicable to **standard** variables:
 - Included in the EUROMOD input microdata, or
 - Created in the EUROMOD spine for policy simulations and saved in the output microdata (*_s)
- The goal is to achieve:
 - Intuitive variable names
 - Harmonised variable names to allow for consistent cross-country comparisons

Variables: naming conventions (2)

- Names are combination of acronyms: **abb**[_s]**
 - **a** = type of information
 - e.g. y=income, x=expenditure, d=demographic, l=labour
 - **bb** = specific for each type **a**, e.g.
 - e.g. y|em: employment income, y|se: self-employment income
 - ****** = further **bb's** for additional information/detail, e.g.
 - e.g. y|em|xp: employment income, extra pay
 - **_s** for simulated variables
 - e.g. b|sa|rg_s: benefit, social assistance, regional, simulated
 - exception id*, e.g. idperson, idmother
- Acronyms and list of variables stored in a common variables library

Variables: library (1)



Variables: library (2)

Administration of Variables and Acronyms

Variables Acronyms

+ Add Variable
- Delete Variable
Show variables ...
Search
Search Va...

Variables

Name	Monetary	HH Level	Categorical	Automatic Label
1518 yem	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment
1519 yem_a	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : add on
1520 yem_s	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : simulated
1521 yem00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : main/basic
1522 yem01	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : 01
1523 yem01_s	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : 01 : simulated
1524 yem02	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : 02
1525 yem02_s	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : 02 : simulated
1526 yem03_s	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : 03 : simulated
1527 yem04_s	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : 04 : simulated
1528 yemabnt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : abroad
1529 yemabnt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : abroad : not taxable
1530 yemabtx	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : abroad : taxable
1531 yemaj	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : additional jobs
1532 yemajmy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : additional jobs : my per year
1533 yemcs	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : company share
1534 yemdt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : date of interview
1535 yemeq_s	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	income : employment : full-time equivalent employment : simulated

Acronyms

Description	Acronym
DEMOGRAPHIC	D
LABOUR MARKET	L
BENEFIT/PENSION	B/P
INCOME	Y
main	
disposable	DS
earnings	EA
employment	EM
	IY
	PP
	PR
	PT
self employment	SE
	UN
	IV
	IG
daily wage	DW
provider	
origin	
employment	
invest income	

Descriptions

Country	Description
at	employment income
be	employment income
bg	employment income
cy	employment income
cz	Income from employment (Prijmy ze zamestnani)
de	employment income

Categories

Value	Description
-------	-------------

Annotations:

- acronyms used for names
- list of all variables defined in the variables file in alphabetical order
- variable is monetary or non-monetary
- automatic label
- description of variable for countries where it is used
- name

Variables: system function *DefVar*

- To define **intermediate** (temporary) variables not included in the Variables library

	Policy	Gr...	BG_2020	Comment
31	▼ ● bchbals_bg		on	BEN: Birth grant (also for adoption) (еднократна помощ при раждане)
31.1	▸ fx DefConst		on	Parameters used in the policy - defined as constants
31.2	▼ fx DefVar		on	
31.2.1	i_nbaby		0	
31.3	▼ fx BenCalc		on	number of babies (aged=0)
31.3.1	Comp_Cond	1	(dag=0)	
31.3.2	Comp_perElig	1	1	
31.3.3	Output_Var		i_nbaby	
31.3.4	TAX_UNIT		tu_bchmt00_bg	
31.4	▼ fx Elig		on	If the family has only one baby aged 0
31.4.1	elig_cond		i_nbaby=1	
31.4.2	TAX_UNIT		tu_bchmt00_bg	

- Not needed for pre-defined system variables sin00_s - sin50_s

Variables: summary

	Used in the spine	Variable exists in the variables list	Acronyms exist in the variables list	Action
Standard variables	yes	yes	yes	Use the variable directly
	no	yes	yes	Use the variable directly
	no	no	yes	Create the variable in the variables list
	no	no	no	Create the acronyms <i>and</i> the variable in the variables list (rarely needed)
Intermediate variables	yes	n/a	n/a	Use the variable directly
	no	n/a	n/a	Create the variable with DefVar (not needed for sin??_s)



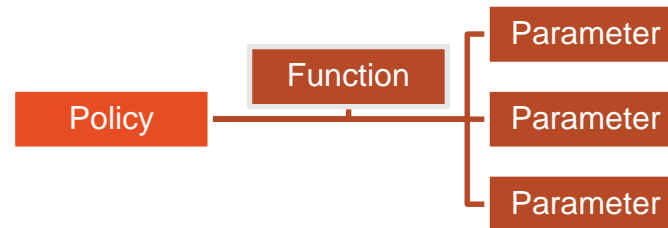
Q&A

Questions

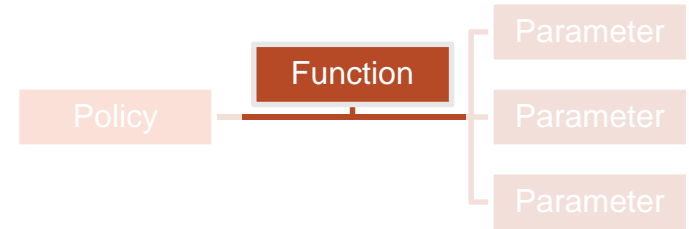


Recall building block of the EUROMOD language

- Building blocks
 - Policies
 - Functions
 - Parameters



Types of functions



System functions

- functions used to define some general settings (e.g. to define tax units)
- In Session 2 we have learned about *DefVar*
- In Session 3 we will learn *DefIL* and *DefConst*
- In Session 4 we will learn *DefTU*
- In Session 5 we will learn *DefOutput* and *Uprate*

Policy functions

- functions used to implement tax-benefit policies (e.g. define eligibility, calculate amount etc.)
- Now we are about to learn *Elig* and *ArithOp*
- In Session 3 we will learn *BenCalc* and *SchedCalc*
- In Session 4 we will learn *Allocate*

Special functions

- more advanced functions that perform more complicated tasks (e.g. loops, change parameters at run-time etc.)
- **not covered in this course**

Calculating a benefit/tax

- What are the policy rules:
 - Who is entitled to the benefit/liable to the tax, i.e. the **assessment unit**:
 - e.g. the individual, family or household
 - What are the criteria, i.e. **eligibility conditions**
 - e.g. being a lone parent
 - What is the **benefit/tax amount**:
 - e.g. €50 per month or 20% of taxable income
- Write down the policy rule
 - Using EUROMOD **tax-benefit language**

Assessment units

Main assessment units (TAX_UNIT) defined in EUROMOD:

- **HOUSEHOLD** (e.g. `tu_household_xx`): all individuals of the household are in the same unit
- **INDIVIDUAL** (e.g. `tu_individual_xx`): each individual of the household forms its own unit
- **FAMILY** (e.g. `tu_family_xx`, `tu_bu_xx`): the nuclear family – e.g. the couple (cohabiting or married) or single adult plus any dependent children
 - The household may be split into several units of different size



All (policy) functions in EUROMOD are computed in relation to an assessment unit → indicated in the TAX_UNIT parameter

▼ fx	Elig		on	Eligibility for Lone parent supplement
	elig_cond		nDepChildrenInTU > 0 & IsLoneParent	
	TAX_UNIT		tu_family17_fi	
▼ fx	ArithOp		on	Child benefit: single parent supplement
	who_must_be_elig		one	
	formula		nDepChildrenInTU * \$bchsup	
	output_add_var		bch_s	Child benefit (with lone parent supplement)
	TAX_UNIT		tu_family17_fi	

THE POLICY FUNCTIONS

ArithOp & Elig

Function ArithOp

- Arithmetical **calculator**
- Compulsory parameters: **formula**, **output_var**, **TAX_UNIT**
 - The parameter **formula** contains calculations
 - The result is stored as output variable via parameter **output_var**
 - To add something to the result of a previous function, use **output_add_var**

Policy	System Name	Comment
ArithOp	on	Made-up example: child benefit
formula	20#w*nDepChildrenInTu	€20 per week for each dependent child in the assessment unit
output_var	bch_s	result saved in the variable bch_s (b: benefit, ch: child, _s: simulated)
TAX_UNIT	tu_bu_ee	assessment unit used for the calculations

- Now let's have a look at what we can use in parameter **formula**
 - **amounts, operations, queries**

Parameter formula: amounts

- Monetary (numbers; use . for decimal) followed by their periodicity:
 - #m for monthly (no conversion)
 - #y for yearly
 - #q for quarterly
 - #w for weekly
 - #d for daily
 - #l for labour day
 - #s for six day labour week
 - #c for capital (no conversion)
- Default is #m (monthly)

Policy	System Name	Comment
ArithOp	on	Made-up example: child benefit
formula	20#w* ⁿ DepChildrenInTu	€20 per week for each dependent child in the assessment unit
output_var	bch_s	result saved in the variable bch_s (b: benefit, ch: child, _s: simulated)
TAX_UNIT	tu_bu_ee	assessment unit used for the calculations

Parameter formula: operations

- Operations: *, /, +, -, ^, min(), max(), abs(), (), !(), %

Policy	System Name	Comment
ArithOp	on	Made-up example: child benefit
formula	(20#w - xed/2)*nDepChildrenInTu - bed_s	€20 per week minus half of educational expenses (xed), for each dependent child in the assessment unit; and deduct any education benefits (bed_s)
output_var	bch_s	result saved in the variable bch_s (b: benefit, ch: child, _s: simulated)
TAX_UNIT	tu_bu_ee	assessment unit used for the calculations

Parameter formula: queries (1)

- Frequently used ready-made calculations
- The result of a query is:
 - yes (=1) or no (=0) (e.g. IsDepChild)
 - some (monetary or non monetary) value (e.g. nDepChildrenInTu)
- Well-documented in the Help file of the User Interface

Query	Description	Parameters
IsDepChild	Returns 1 if a person is a 'dependent child', i.e. fulfils the DepChildCond of the assessment unit specification, 0 otherwise	
nDepChildrenInTu#x	Returns the number of dependent children in the assessment unit who fulfil dag >= parameter #_AgeMin and dag <= parameter #_AgeMax	#_AgeMin, #_AgeMax (optional)

Parameter formula: queries (2)

- No need to learn the query names by heart
→ use “IntelliSense” feature to find and select the desired query

ArithOp	on	Made-up example: child benefit
formula	20#w*n	€20 per week for each dependent child in the assessment unit
output_var	bcl	the variable bch_s child, _s: simulated)
TAX_UNIT	tu	t used for the calculations

IntelliSense dropdown menu showing a list of queries:

- nLooseDepChildrenInTu
- nDepParentsInTu
- nDepRelativesInTu
- nDepParentsAndRelativesInTu
- nChildrenOfCouple#x
- nDepChildrenOfCouple#x
- nPersInUnit#x
- nAdultsInTu#x
- nDepChildrenInTu#x** (highlighted with a red box)

Function Elig

- Used to implement **conditions**
- Compulsory parameters: **elig_cond**, **TAX_UNIT**
 - Condition is defined in parameter **elig_cond**
 - Creates a variable equal to 0 or 1 (by default **sel_s**)
 - Subsequent functions use this information via parameter **Who_Must_Be_Elig**

Function Elig

Policy	System Name	Comment
Elig	on	Made-up example: condition to pay employee social insurance contributions (SIC)
elig_cond	!IsCivilServant	not a civil servant (!IsCivilServant)
TAX_UNIT	tu_individual_ee	assessment unit used for the calculations
ArithOp	on	Made-up example: pension contributions
formula	yem*0.08	8% of earnings (yem) for old-age pension SIC
output_var	tsceepi_s	result saved in the variable tsceepi_s (t: tax, sc: social contributions, ee: employee, pi: pension insurance, _s: simulated)
TAX_UNIT	tu_individual_ee	assessment unit used for the calculations



calculations in ArithOp are carried out for everyone with earnings, including civil servants

Function Elig

Policy	System Name	Comment
Elig	on	Made-up example: condition to pay employee social insurance contributions (SIC)
elig_cond	!IsCivilServant	not a civil servant (!IsCivilServant)
TAX_UNIT	tu_individual_ee	assessment unit used for the calculations
ArithOp	On	Made-up example: pension contributions
Who Must Be Elig	One	calculations carried out if at least one member of assessment unit fulfils condition from last Elig function
formula	yem*0.08	8% of earnings (yem) for old-age pension SIC
output_var	tsceepi_s	result saved in the variable tsceepi_s (t: tax, sc: social contributions, ee: employee, pi: pension insurance, _s: simulated)
TAX_UNIT	tu_individual_ee	assessment unit used for the calculations
ArithOp	on	Made-up example: unemployment contributions
Who Must Be Elig	one	calculations carried out if at least one member of assessment unit fulfils condition from last Elig function
formula	yem*0.02	2% of earnings for unemployment SIC
output_var	tsceeu_i_s	result saved in the variable tsceeu_i_s (t: tax, sc: social contributions, ee: employee, ui: unemployment insurance, _s: simulated)
TAX_UNIT	tu_individual_ee	assessment unit used for the calculations

Parameters: eligibility

(elig_cond; Who_Must_Be_Elig)

- **Who_Must_Be_Elig**: calculations of the function are carried out if...
 - **one**: one member of the assessment unit is eligible
 - **one_adult**: one adult member of the assessment unit is eligible
 - **all**: all members of the assessment unit are eligible
 - **all_adults**: all adult members of the assessment unit are eligible
 - **nobody**: calculations are carried out for each assessment unit (default)
- By default eligibility result is saved in the variable **sel_s** (can use other variable in parameter elig_var)
 - 0: person is not eligible
 - 1: person is eligible

Parameters: eligibility

(elig_cond; Who_Must_Be_Elig)

Policy	System Name	Comment
Elig	on	Made-up example: eligibility condition for social assistance
elig_cond	dag>=80	individual should be of age (dag) of 80+ years
TAX_UNIT	tu_individual_ee	assessment unit is the INDIVIDUAL
ArithOp	On	Made-up example: social assistance amount
Who_Must_Be_Elig	?	who in the assessment unit must fulfil eligibility condition
formula	100#m	benefit amount is €100
output_var	bsa_s	result saved in the variable bsa_s (b: benefit, sa: social assistance, _s: simulated)
TAX_UNIT	tu_household_ee	assessment unit is the HOUSEHOLD

				Will the calculations in ArithOp be carried out if: who_must_be_elig=				
idhh	idperson	dag	sel_s	one	one_adult	all	all_adults	nobody
1	11	80	1					
1	12	60	0	yes	yes	no	no	yes
1	13	40	0					
2	21	80	1					
2	22	6	0	yes	yes	no	yes	yes
3	31	80	1	yes	yes	yes	yes	yes
4	41	40	0					
4	42	40	0	no	no	no	no	yes

Parameters: output

(output_var; output_add_var)

- Either output_var or output_add_var must be indicated!
- Exception: func_Elig → sel_s

Policy	System Name	Comment
ArithOp	on	Made-up example: child benefit
formula	20#w*nDepChildrenInTu	€20 per week for each dependent child in the assessment unit
output_var	bch_s	result saved in the variable bch_s (b: benefit, ch: child, _s: simulated)
TAX_UNIT	tu_bu_ee	
Elig	on	Made-up example: eligibility condition
elig_cond	IsLoneParentOfDepChild	if a lone parent of a dependent child
TAX_UNIT	tu_bu_ee	
ArithOp	on	Made-up example: child benefit supplement for lone parents
Who_Must_Be_Elig	one	
formula	15#w	€15 per week for the assessment unit
output_add_var	bch_s	add answer to the result of the previous function, saved in the variable bch_s
TAX_UNIT	tu_bu_ee	

Parameter values and assessment unit: conditions vs. other parameters (1)

Level of Interpretation	... used in condition parameters	... used in other parameters
monetary variables and income lists ...	assessment unit	assessment unit
non-monetary variables and individual level queries ...	individual	head of assessment unit
non individual level queries ...	consult description in section EUROMOD Functions - Queries	consult description in section EUROMOD Functions - Queries

Examples:

- monetary variables and income lists: yem, ils_origy
- non-monetary variables and individual-level queries: dag, IsParent
- non-individual level queries: nDepChildren, nPersInUnit

Parameter values and assessment unit: conditions vs. other parameters (2)

Policy	System Name	Comment
Elig	on	Made-up example: condition to receive a housing benefit
elig_cond	yem<30000#y & nPersonsInTU<=2 & dag>=60	household's earnings are less than €30,000 per year, there are max 2 persons in the household, and person's age is 60+
TAX_UNIT	tu_household_ee	assessment unit is the household
ArithOp	on	Made-up example: housing benefit
Who_Must_Be_Elig	all	calculations carried out if each member of assessment unit fulfils condition from last Elig function
formula	IsDisabled * (xhc – yem)	benefit funds expenditure for housing (xhc) if household head is disabled; benefit is reduced with household's earnings
lowlim	0	min benefit amount is 0
output_var	bho_s	
TAX_UNIT	tu_household_ee	assessment unit is the household

User Interface: Adding a function

Policy	System Name
Elig	on
elig_cond	IsLoneParentOfDepChild
TAX_UNIT	tu_bu_ee

find existing function, after/before which to place new function, and right-click on function or parameter name to open “menu”

The screenshot shows the 'Add Function' context menu in the EUROMOD software. The menu is open, displaying various options for adding, deleting, copying, and pasting functions. The 'Add Function Before' and 'Add Function After' options are highlighted with a red box. The 'Elig' function name is highlighted in the table above.

Policy	System Name
Elig	on
elig_cond	IsLoneParentOfDepChild
TAX_UNIT	tu_bu_ee

find existing function, after/before which to place new function, and right-click on function or parameter name to open “menu”

- Add Function Before
- Add Function After
- Delete Function(s)
- Copy Function(s)
- Paste Function(s) Before
- Paste Function(s) After
- Copy Value(s)
- Paste Value(s)
- Move Function(s) Up Ctrl+Up
- Move Function(s) Down Ctrl+Down
- Copy Identifier
- Copy Symbolic Identifier
- Set/Unset Private
- Groups
- Extensions
- Expand All Functions
- Collapse All Functions
- Delete Parameter(s) Del
- Show Add Parameter Form Ctrl+A

ArithOp
Elig
BenCalc
SchedCalc
Min
Max
Allocate
System Functions
Special Functions

User Interface: Adding a parameter (1)

Policy	System Name	Con
Elig	on	Mac
elig_cond	IsLoneParentOfDepChild	if a
TAX_UNIT	tu_bu_ee	
ArithOp	On	Mac
formula	15#w	€15
output_add_var	bch_s	add
TAX_UNIT	tu_bu_ee	func

find relevant function and right-click on function or parameter name to open "menu"

- Add Function Before
- Add Function After
- Delete Function(s)
- Copy Function(s)
- Paste Function(s) Before
- Paste Function(s) After
- Copy Value(s)
- Paste Value(s)
- Move Function(s) Up Ctrl+Up
- Move Function(s) Down Ctrl+Down
- Copy Identifier
- Copy Symbolic Identifier
- Set/Unset Private
- Groups
- Extensions
- Expand All Functions
- Collapse All Functions
- Delete Parameter(s) Del
- Show Add Parameter Form Ctrl+A

User Interface: Adding a parameter (2)

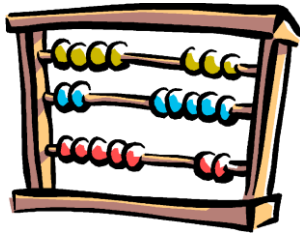
Add Parameters

ArithOp (order: 1)

function parameters refer to

Add	Parameter	Replaces	Grp/No	Count	Default	Description
<input type="checkbox"/>	Output_Add_Var	Output_Var				Variable for storing the result of the function. Result of functio...
<input type="checkbox"/>	Result_Var					Variable for storing the result of the function. Result of functio...
<input type="checkbox"/>	Who_Must_Be_Elig					Function's calculations are carried out if ...- one (one_member...
<input type="checkbox"/>	Elig_Var				sel_s	Variable indicating whether a person is 'eligible' (see paramete...
<input type="checkbox"/>	Run_Cond					Function is only carried out if the condition is fulfilled. The par...
<input type="checkbox"/>	LowLim				-1.79769313486232E+308	Replaces result of function if result is smaller.
<input type="checkbox"/>	UpLim				1.79769313486232E+308	Replaces result of function if result is higher.
<input type="checkbox"/>	Threshold				-1.79769313486232E+308	Replaces result of function if result is smaller: if lower limit is no...
<input type="checkbox"/>	Limpriority				n/a	Parameter for the further specification of an operand:Possible ...
<input type="checkbox"/>	Round_Down					Result is rounded down to nearest whole number if set to 1, to...
<input type="checkbox"/>	Round_Up					Result is rounded up to nearest whole number if set to 1, to n...
<input type="checkbox"/>	Round_to					Result is rounded to nearest whole number if set to 1, to near...
<input type="checkbox"/>	#_LowLim		1	1	-1.79769313486232E+308	Footnote parameter for the further specification of an operand...
<input type="checkbox"/>	#_UpLim		1	1	1.79769313486232E+308	Footnote parameter for the further specification of an operand...
<input type="checkbox"/>	#_LimPriority		1	1	n/a	1-Footnote parameter for the further specification of an opera...
<input type="checkbox"/>	#_Level		1	1		Footnote parameter for the further specification of an operand...
<input type="checkbox"/>	#_Amount		1	1		Footnote parameter for the further specification of an operand...
<input type="checkbox"/>	#_AgeMin		1	1	0	Parameter of several queries (e.g. nDepChildrenInTu).
<input type="checkbox"/>	#_AgeMax		1	1	1.79769313486232E+308	Parameter of several queries (e.g. nDepChildrenInTu).
<input type="checkbox"/>	#_n		1	1		Parameter of query IsNtoMchild.

☒ Show Common Parameters
☒ Show Footnote Parameters



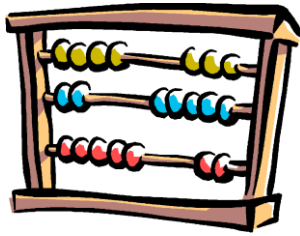
Exercise 3a

Child allowance reform

Consider the effect of a Child Allowance (CA) reform in Estonia, in 2012. In that year, the CA was a monthly universal benefit. The amount per child was two times the Child Allowance Rate (CAR equal to 9.59 EUR), and 6 times the CAR for the third and any consequent child.

The reform adds a means-tested supplement to the universal CA for families with two children and with earnings below 200 EUR per month. The supplement is equal to 1800 EUR.

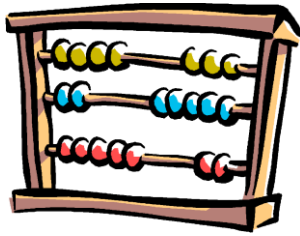
You will do this exercise on your own!



Exercise 3a

Steps:

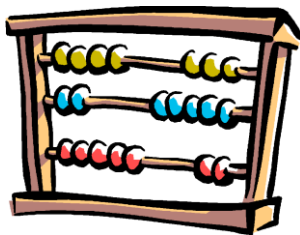
- Create a new system in Estonia based on the 2012 system and name it EE_2012_E3;
- Modify Child Allowance (CA) in way that access to the supplement should be restricted to families with at least two children and with earnings below 200 EUR per month. The supplement is equal to 1800 EUR per month;
- Run the model and use the *Statistics Presenter* – Baseline/Reform option to analyse the distributional effect of the reform.



Exercise 3a

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*.

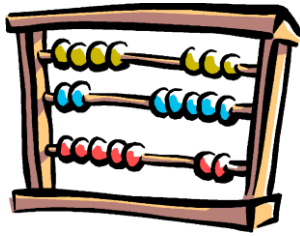


Exercise 3a: solution

Market Incomes and Government Revenue & Expenditure ?

Yearly, mill., currency as defined in EM output

	EE_2012 (base) (annual)	EE_2012_E3 (annual)	Difference to base (annual)	% Difference to base
Total market incomes	8,728.13	8,728.13	0.00	0.00 %
... income from (self) employment	8,728.13	8,728.13	0.00	0.00 %
... other sources	0.00	0.00	0.00	0.00 %
Government revenue through taxes and social insurance contributions	5,133.70	5,133.70	0.00	0.00 %
... direct taxes	1,694.55	1,694.55	0.00	0.00 %
... employee social insurance contributions	403.67	403.67	0.00	0.00 %
... self-employed social insurance contributions	109.55	109.55	0.00	0.00 %
... other social insurance contributions	0.00	0.00	0.00	0.00 %
... employer social insurance contributions (not part of disposable income)	2,925.93	2,925.93	0.00	0.00 %
Credited social insurance contributions (not part of disposable income)	64.23	64.23	0.00	0.00 %
Government expenditure on social transfers	2,376.26	2,511.65	135.39	5.70 %
by target group				
... unemployment benefits	254.74	254.74	0.00	0.00 %
... family and education benefits	87.84	230.16	142.32	162.02 %
... social assistance and housing benefits	97.94	91.01	-6.93	-7.08 %
... pensions, health and disability benefits	1,935.74	1,935.74	0.00	0.00 %
... firms	0.00	0.00	0.00	0.00 %
by benefit design				
... means-tested non-pension benefits	97.94	91.01	-6.93	-7.08 %
... non-means-tested non-pension benefits	342.58	484.90	142.32	41.54 %
... pensions	1,935.74	1,935.74	0.00	0.00 %
... firms subsidies	0.00	0.00	0.00	0.00 %



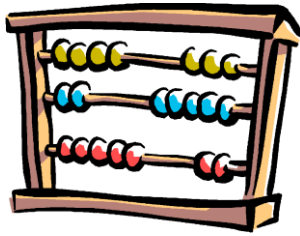
Exercise 3b

Child benefit reform in Greece

The aim of this exercise is to perform a reform of the child benefit 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 1st and 2nd child was 70 EUR/month, and for the 3rd 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 brings the child benefit amount to 300 EUR/month. It does not reduce it, if it is higher than this amount.

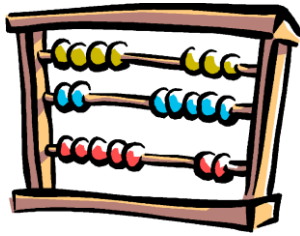
You will do this exercise on your own!



Exercise 3b

Steps:

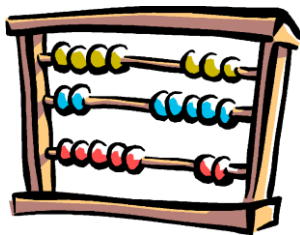
- Create a new system in Greece based on the 2021 system and name it EL_2021_E3;
- Add a supplement (top up) to the existing child benefit (bch_s). The supplement is given to recipient families with at least one child aged up to 6 and increases the child benefit amount to 300 EUR per month. It does not reduce it, if it is higher than this amount;
- Run EUROMOD to produce microdata-outputs for these two systems (EL_2021 and EL_2021_E3);
- Analyse the results with the statistics presenter.



Exercise 3b

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`.



Exercise 3b: solution

Market Incomes and Government Revenue & Expenditure ?

Yearly, mill., currency as defined in EM output

	EL_2021 (base) (annual)	EL_2021_E3 (annual)	Difference to base (annual)	% Difference to base
Total market incomes	38,367.24	38,367.24	0.00	0.00 %
... income from (self) employment	38,367.24	38,367.24	0.00	0.00 %
... other sources	0.00	0.00	0.00	0.00 %
Government revenue through taxes and social insurance contributions	22,365.79	22,365.79	0.00	0.00 %
... direct taxes	3,928.10	3,928.10	0.00	0.00 %
... employee social insurance contributions	5,417.45	5,417.45	0.00	0.00 %
... self-employed social insurance contributions	2,938.17	2,938.17	0.00	0.00 %
... other social insurance contributions	1,434.14	1,434.14	0.00	0.00 %
... employer social insurance contributions (not part of disposable income)	8,647.93	8,647.93	0.00	0.00 %
Credited social insurance contributions (not part of disposable income)	0.00	0.00	0.00	0.00 %
Government expenditure on social transfers	33,907.88	34,651.56	743.68	2.19 %
by target group				
... unemployment benefits	5,267.97	5,267.97	0.00	0.00 %
... family and education benefits	1,062.07	1,822.43	760.36	71.59 %
... social assistance and housing benefits	3,409.29	3,392.61	-16.68	-0.49 %
... pensions, health and disability benefits	24,168.55	24,168.55	0.00	0.00 %
... firms	0.00	0.00	0.00	0.00 %
by benefit design				
... means-tested non-pension benefits	4,315.21	5,058.88	743.68	17.23 %
... non-means-tested non-pension benefits	5,277.90	5,277.90	0.00	0.00 %
... pensions	24,314.77	24,314.77	0.00	0.00 %
... firms subsidies	0.00	0.00	0.00	0.00 %

Questions





Summary: Exercises 3a, 3b

- You learned how to use DefVar, Elig and ArithOp to simulate/reform a benefit, using
 - Who_Must_Be_Elig to link Elig and ArithOp
 - Queries & amount parameters
 - Parameter output_add_var
- You learned how to add a new function and parameters



Session 3

Policy functions *BenCalc* & *SchedCalc*
Defining constants and income lists

In this session you will learn about

- Combining the features of *Elig* & *ArithOp*: the policy function *BenCalc*
- Implementing tax schedules using the policy function *SchedCalc*
- Defining constants (system function *DefConst*) and income lists (system function *DefIL*)
- Exercises:
 - 4. Introducing a supplement, withdrawn with earnings, to the Child Benefit in Estonia
 - 5a. Introducing a zero tax band to the flat income tax in Bulgaria

Policy	Grp/No	EE_2021
▼ fx BenCalc		on
comp_cond	1	IsNtoMchild#1
#_N	1	1
#_M	1	1
comp_perElig	1	\$CB_Ch1
comp_cond	2	IsNtoMchild#2
#_N	2	2
#_M	2	2
comp_perElig	2	\$CB_Ch2
Comp_Cond	3	IsNtoMchild#3
#_N	3	3
#_M	3	99
Comp_perElig	3	\$CB_Ch3plus
output_var		bch00_s
TAX_UNIT		tu_CBfamily_ee

Policy function *BenCalc*

BenCalc: overview

- Function that combines the functionalities of the functions **Elig** and **ArithOp**
- Typically used to implement means-tested benefits, because they usually consist of:
 - Several components, where a component is added if conditions are met by the assessment unit
 - An income test, where certain types of income reduce benefit entitlement
- Compulsory parameters
 - **Comp_Cond**
 - **Comp_PerTU/Comp_PerElig**
 - **Output_Var**
 - **TAX_UNIT**

BenCalc: main parameters

- **Comp_Cond** used to define a condition
 - as *Elig_Cond* in function *Elig*
- **Comp_perTU** or **Comp_perElig** used to calculate a formula, as *Formula* in function *ArithOp*
 - **Comp_perTU**: amount is added once to the assessment unit
 - **Comp_perElig**: amount is added once for each individual fulfilling the condition in the assessment unit
- Column **Grp/No** used to group together **Comp_Cond** and **Comp_perTU/Comp_perElig** into one component
- Upper and lower limits for each component can be set with **Comp_Lowlim** and **Comp_Uplim** respectively

BenCalc: Comp_perTU vs Comp_perElig

Policy	Grp/No	System Name	Comment
BenCalc		on	Made-up example: child benefit
Comp_Cond	1	IsDepChild	if a dependent child in the assessment unit...
Comp_perTU	1	20#w	...benefit amount is €20 per week
output_var		bch_s	
TAX_UNIT		tu_CBfamily_ee	assessment unit used for the calculations



Policy	Grp/No	System Name	Comment
BenCalc		on	Made-up example: child benefit
Comp_Cond	1	IsDepChild	if a dependent child in the assessment unit...
Comp_perElig	1	20#w	...benefit amount is €20 per week PER CHILD
output_var		bch_s	
TAX_UNIT		tu_CBfamily_ee	assessment unit used for the calculations

BenCalc: example

- A lone parent family with two children, one with a disability, will receive ...?

€55#w

Policy	Grp/No	System Name	Comment
BenCalc		on	Made-up example: child benefit
Comp_Cond	1	IsDepChild	if a dependent child in the assessment unit...
Comp_perElig	1	20#w	...benefit amount is €20 per week per child
Comp_Cond	2	IsDepChild & IsDisabled	if a child with a disability in the assessment unit...
Comp_perElig	2	10#w	...€10 per week for each child with a disability
Comp_Cond	3	IsLoneParentOfDepChild	if a lone parent in the assessment unit...
Comp_perTU	3	5#w	...€5 per week for the assessment unit
output_var		bch_s	
TAX_UNIT		tu_CBfamily_ee	assessment unit used for the calculations

BenCalc: query IsNtoMchild#x

- IsNtoMchild#x is useful to implement child benefit which varies by children:
 - Condition fulfilled if a person belongs to the n to m oldest dependent children of the assessment unit, not fulfilled otherwise.
 - n and m are defined by *footnote* parameters #_N and #_M.

- footnote parameters are referred to by #

- they are grouped together with main parameter they refer to

Policy	Grp/No	System Name	Comment
BenCalc		on	Made-up example: child benefit
Comp_Cond	1	IsNtoMchild#1	if the dependent child is...
#_N	1	1	...the first (oldest) child
#_M	1	1	
Comp_perElig	1	20#w	benefit amount is €20 per week for the first (oldest) child
Comp_Cond	2	IsNtoMchild#2	if the dependent child is...
#_N	2	2	...the second or any other child
#_M	2	99	
Comp_perElig	2	10#w	benefit amount is €10 per week for any other child
output_var		bch_s	
TAX_UNIT		tu_CBfamily_ee	assessment unit used for the calculations

Footnote parameters

- They serve the further specification of other parameters.
- Identified by #i (i=number from 1 to....)
 - Limits
 - Amounts
 - Assessment units
 - Specification of queries

Policy	Grp/No	System Name	Comment
ArithOp		on	Made-up example: tax allowance for pensions
Formula		il_pensions#1 + (ils_earns#2 + GetPartnerIncome#3) * 0.3	the person's pensions - up to 1,000 per month - plus 30% of the sum of the couple's earned income and the pension of the partner
# UpLim	1	1000#m	upper limit on the person's pensions
# Level	2	tu couple lv	alternative assessment unit
# Income	3	poa	get the pension of the partner
output_var		tintape_s	
TAX_UNIT		tu_individual_lv	

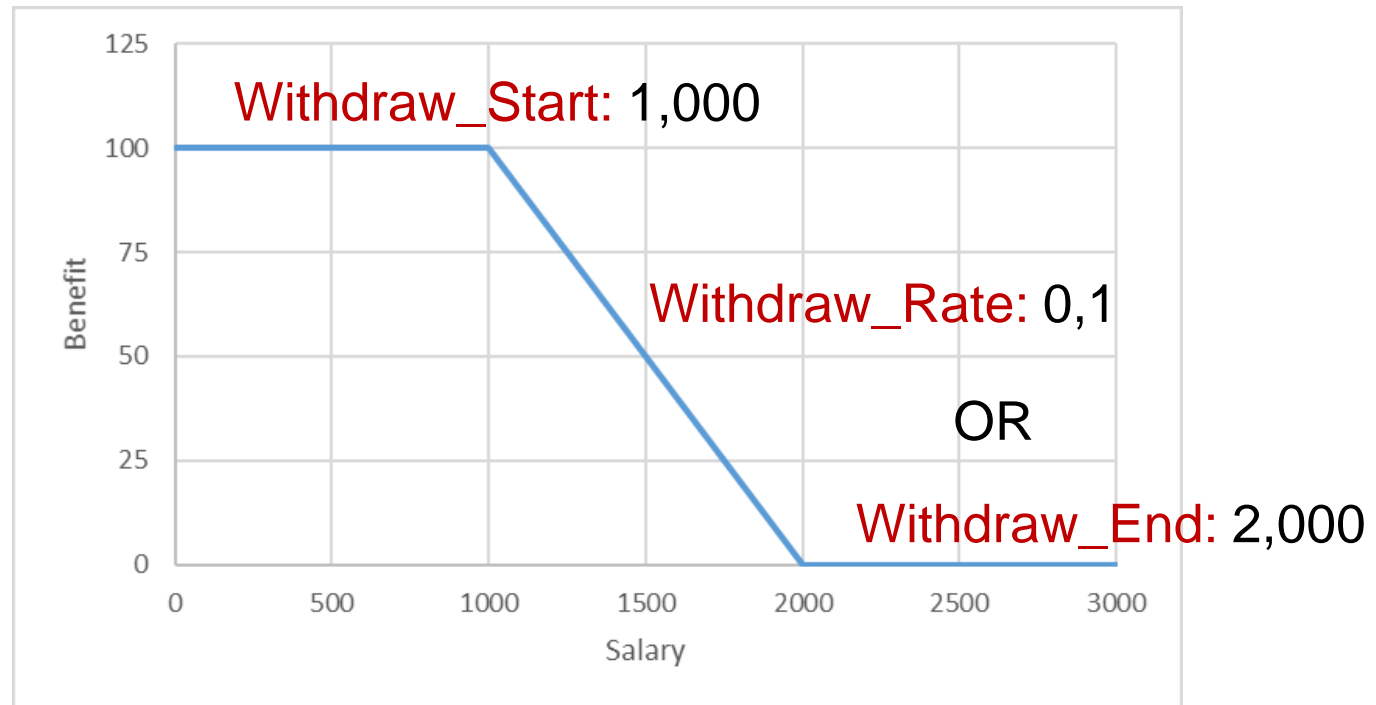
BenCalc: withdraw parameters

- Withdraw parameters: subtract something from the calculated sum of components
 - **Withdraw_Base**: variable or income list to compute the withdrawal
 - **Withdraw_Start**: level of the base where withdrawal starts
 - **Withdraw_Rate**: what percentage of the base is being withdrawn
 - **Withdraw_End**: level of the base where sum of components is 0
- Negative result automatically set to 0
- $\text{Result} = \max(\text{Sum of components} - \max(\text{Withdraw_Base} - \text{Withdraw_Start}, 0) * \text{Withdraw_rate}, 0)$
- **_Rate** and **_End** cannot be used simultaneously
 - If Withdraw_End is specified:
$$\text{RATE} = (\text{Sum of components}) / (\text{Withdraw_End} - \text{Withdraw_Start})$$

BenCalc: withdraw parameters (example)

- Ex. Benefit for lone parents: 100 EUR/month benefit up to 1,000 EUR/month salary; for each additional EUR of salary, 0.1 EUR of the benefit are withdrawn

Withdraw_Base: salary

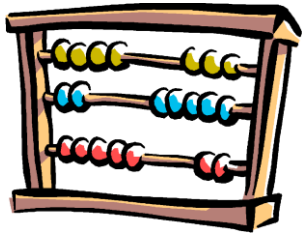


BenCalc: withdraw parameters (example)

Policy	Grp/No	System Name	Comment
BenCalc		on	Made-up example: child benefit
Comp_Cond	1	IsLoneParentOfDepChild	if there is a lone parent in the assessment unit...
Comp_perTu	1	100#m	...benefit amount is €100 per month as a general rule
Withdraw_Base		yem	If salary...
Withdraw_Start		1000#m	... is above €100 per month...
Withdraw_Rate		0.1	... €0.1 are withdrawn for each additional €1 earned above €1,000 per month
output_var		bch_s	
TAX_UNIT		tu_CBfamily_ee	assessment unit used for the calculations



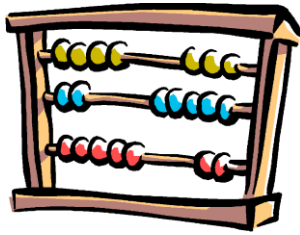
- the calculations done with the withdraw parameters apply to the sum of all components
- withdraw parameters do not need to be grouped



Exercise 4

Introducing a supplement, withdrawn with earnings, to the Child Benefit in Estonia

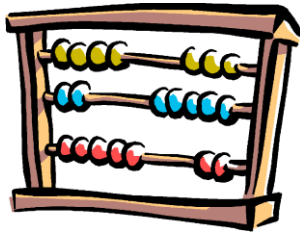
Perform a reform of the Child Allowance in Estonia in 2021. Your task is to **add** a supplement to the Child Allowance for families with at least 2 dependent children below the age of 18, using the function *BenCalc*. The supplement is equal to €30 per week but should be withdrawn with earnings: 10%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



Exercise 4

Steps

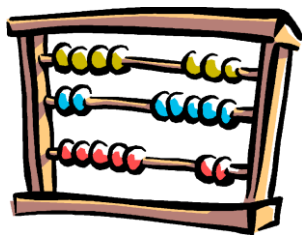
- Create a new system in the EE based on the 2021 system and name it EE_2021_E4.
- Introduce a supplement for large families to the Child Benefit, which is withdrawn with earnings.
- Run EUROMOD to produce micro-outputs for the baseline and reform systems.
- Analyse the distributive impact of this reform.



Exercise 4

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*.



Exercise 4 – solution

2.1.a Mean household income (monthly) by decile groups ?

	EE_2021 (base)	EE_2021_E4	Difference to base	% Difference to base
Decile 1	374.99	376.76	1.77	0.47 %
Decile 2	725.70	746.15	20.45	2.82 %
Decile 3	708.90	716.07	7.17	1.01 %
Decile 4	1,160.77	1,183.42	22.65	1.95 %
Decile 5	1,383.11	1,395.93	12.82	0.93 %
Decile 6	1,707.50	1,717.09	9.59	0.56 %
Decile 7	1,926.25	1,930.32	4.07	0.21 %
Decile 8	2,192.65	2,193.44	0.80	0.04 %
Decile 9	2,457.93	2,457.93	0.00	0.00 %
Decile 10	2,757.48	2,757.48	0.00	0.00 %
All	1,384.74	1,392.62	7.88	0.57 %
Poor	511.29	507.42	-3.87	-0.76 %

STATISTICS PRESENTER OUTPUT

▼ fx	SchedCalc		on	National Income tax schedule
	Base		il_taxableY_nattax	Taxbase is income as defined in incomelist il_taxableY_nattax
	Band_UpLim	1	15000#y	Income up to this annual amount is taxed ...
	Band_Rate	1	0.23	... with this rate ...
	Band_UpLim	2	28000#y	... income above the previous annual amount and up to this annual amount is taxed...
	Band_Rate	2	0.27	... with this rate ...
	Band_UpLim	3	55000#y	... income above the previous annual amount and up to this annual amount is taxed...
	Band_Rate	3	0.38	... with this rate ...
	Band_UpLim	4	75000#y	... income above the previous annual amount and up to this annual amount is taxed...
	Band_Rate	4	0.41	... with this rate ...
	Band_Rate	5	0.43	... income above the previous annual amount is taxed with this rate
	output_var		tintsna_s	National Income tax schedule (i.e. National Income tax before tax credits)
	lowlim		0	"This is to avoid negative results (i.e. if Taxbase is negative, the tax is set to 0)"
	TAX_UNIT		tu_individual_it	

Policy function *SchedCalc*

SchedCalc: overview

- Used (mainly) in progressive taxes to define a tax schedule:
 - Tax base: **Base**
 - Tax bands: **Band_UpLim** / **Band_LowLim**
 - Tax rate: **Band_Rate**
- Groups together **Band_Rate** and **Band_UpLim/ Band_LowLim**
- For fixed amounts use **Band_Amount** instead of **Band_Rate**

Policy	Grp/No	System Name	Comment
SchedCalc		on	Made-up example: income tax
Base		tintb_s	income tax calculated based on taxable income (t: tax, in: income, tb: tax base, _s: simulated)
Band_Rate	1	0.2	first band rate: 20%
Band_LowLim	1	12500#y	first band rate applies on income above €12.5k per year...
Band_UpLim	1	50000#y	...and up to €50 per year
Band_Rate	2	0.4	second band rate: 40%
Band_UpLim	2	150000#y	second band rate applies on income above €50k and up to €150k per year
Band_Rate	3	0.45	third band rate: 45%; applies on income above €150k per year
output_var		tin_s	result saved in variable tin_s (t: tax, in: income, _s: simulated)
TAX_UNIT		tu_individual_it	assessment unit used for the calculations

SchedCalc: parameter Quotient

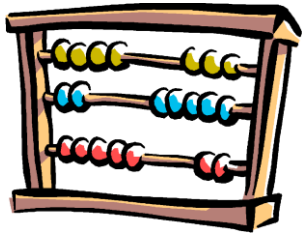
- **Quotient**: split the base and apply the schedule separately
- $\text{Result} = ((\text{Base}/\text{Quotient}) * \text{Tax schedule}) * \text{Quotient}$

Policy	Grp/No	System Name	Comment
SchedCalc		on	Made-up example: income tax based on joint taxation
Base		tintb_s	income tax calculated based on taxable income (t: tax, in: income, tb: tax base, _s: simulated)
Band_Rate	1	0.2	first band rate: 20%
Band_LowLim	1	12500#y	first band rate applies on income above €12.5k per year...
Band_UpLim	1	50000#y	...and up to €50k per year
Band_Rate	2	0.4	second band rate: 40%
Band_UpLim	2	150000#y	second band rate applies on income above €50k and up to €150k per year
Band_Rate	3	0.45	third band rate: 45%; applies on income above €150k per year
Quotient	2		Base is divided by the quotient before the schedule is applied. Afterwards the result is multiplied by the quotient.
output_var		tin_s	result saved in variable tin_s (t: tax, in: income, _s: simulated)
TAX_UNIT		tu_couple_it	assessment unit used for the calculations

SchedCalc: parameter Simple_Prog

- **Simple_Prog**: apply on the whole base the highest marginal tax rate reached by it

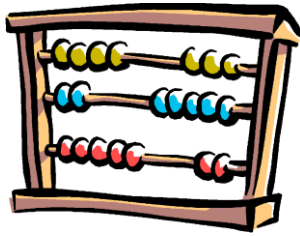
Policy	Grp/No	System Name	Comment
SchedCalc		on	Made-up example: income tax
Base		tintb_s	income tax calculated based on taxable income (t: tax, in: income, tb: tax base, _s: simulated)
Band_Rate	1	0.2	first band rate: 20%
Band_LowLim	1	12500#y	first band rate applies on the whole income if income is between €12.5k per year...
Band_UpLim	1	50000#y	...and €50k per year
Band_Rate	2	0.4	second band rate: 40%
Band_UpLim	2	150000#y	second band rate applies on the whole income if income is between €50k and up to €150k per year
Band_Rate	3	0.45	third band rate: 45%; applies on the whole income if income is above €150k per year
Simple_Prog		yes	
output_var		tin_s	result saved in variable tin_s (t: tax, in: income, _s: simulated)
TAX_UNIT		tu_individual_uk	assessment unit used for the calculations



Exercise 5a

Introducing a zero tax band to the flat income tax in Bulgaria

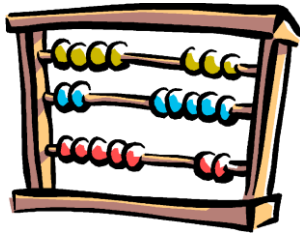
Currently, taxable income in Bulgaria is taxed at 10% from the first BGN earned. You are asked to introduce a zero tax band of BGN 3,200 per year, i.e. individuals should pay tax only on taxable income above BGN 3,200 per year. Furthermore, you should raise the marginal rate of income tax from 10% to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



Exercise 5a

Steps

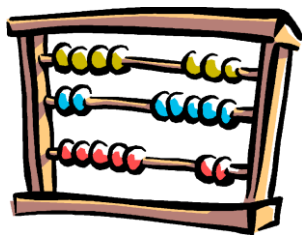
- Create a new system in Bulgaria based on the 2021 system and name it BG_2021_E5.
- Introduce a zero tax band to the flat income tax and modify the tax rate.
- Run EUROMOD to produce micro-outputs for the baseline and reform systems.
- Analyse the distributive impact of this reform.



Exercise 5a

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.



Exercise 5a – solution

2.1.a Mean household income (monthly) by decile groups ?

	BG_2021 (base)	BG_2021_E5	Difference to base	% Difference to base
Decile 1	145.24	146.15	0.91	0.63 %
Decile 2	156.09	156.23	0.14	0.09 %
Decile 3	351.96	361.27	9.31	2.65 %
Decile 4	534.29	540.16	5.87	1.10 %
Decile 5	650.81	661.23	10.42	1.60 %
Decile 6	932.34	947.64	15.30	1.64 %
Decile 7	1,237.86	1,246.24	8.38	0.68 %
Decile 8	1,504.20	1,504.07	-0.13	-0.01 %
Decile 9	1,688.45	1,680.73	-7.72	-0.46 %
Decile 10	2,176.47	2,140.95	-35.52	-1.63 %
All	862.37	861.50	-0.87	-0.10 %
Poor	186.18	186.74	0.56	0.30 %

STATISTICS PRESENTER OUTPUT



Summary of exercises

You learned:

- How to use *BenCalc* to implement a benefit, using parameters $\text{Comp_Cond} + \text{Comp_perTU}$ and $\text{Withdraw_Base} + \text{Withdraw_Start} + \text{Withdraw_Rate}$
- How to use *SchedCalc* to reform the income tax policy, using parameter Band_LowLim and a *DefConst* function

Questions



Constants: function *DefConst* (1)

- Policy parameters can be declared as constants with function *DefConst*:
 - e.g. monthly amounts, income thresholds, maximum amounts, tax rates
 - notated as \$x
- Advantages:
 - over-time changes to tax-benefit parameters are visible at a glance
 - updating of tax-benefit parameters is easier
 - the same constant can be used in several policies (e.g. minimum wage)
- Constants are usually defined in the policy *ConstDef_cc* (if they are used in more than one policy) or at the beginning of a policy which calculates a tax/benefit

Constants: function *DefConst* (2)

Policy	Grp/No	RO_2021	RO_2020	RO_2019	Comment
▼ ConstDef_ro		on	on	on	DEF: CONSTANTS
▼ fx DefConst		on	on	on	
\$FlatIncTax	1	0.10	0.10	0.10	
const_mone...	1	no	no	no	
\$lhw	2	40	40	40	Standard number of working hours per week
efConst		on	on	on	Average gross wage
\$AvGrossWage	1	5380 #m	5429 #m	5163 #m	"The provisional average gross wage used issued by the National Institute of Statistics";

user-defined constant names \$x in "policy" column

changes to constant values highlighted

Income lists: function *Defll* (1)

- Income lists are aggregates of several components
 - **standard**, used for distributional analysis (e.g. disposable income) or fiscal overview (e.g. taxes), compulsory for all countries
 - **non-standard**, used for policy implementation (e.g. taxable income), optional and country-specific
- System function *Defll* allows to define income lists
 - **standard**, defined in the policy *IlsDef_xx*
 - prefix **ils_**
 - used by the Statistics Presenter!
 - **non-standard**, defined in the policy *llDef_xx* or in policies where they are used
 - prefix **il_**

Income lists: function *Defll* (2)

- Aggregates:
 - variables
 - pre-defined income lists
 - fixed amounts
 - constants
- Operations:
 - +
 - -
 - shares (e.g. 0.8 means that the corresponding variable is multiplied by 0.8)
- Use
 - once defined, available for all subsequent functions and policies
 - behave like monetary variables
 - continuously updated throughout the spine

Income lists: function *Defll* (3)

user-defined income list name in “system” column

- components of the income list
- some are being added, other subtracted

income list used to define another income list

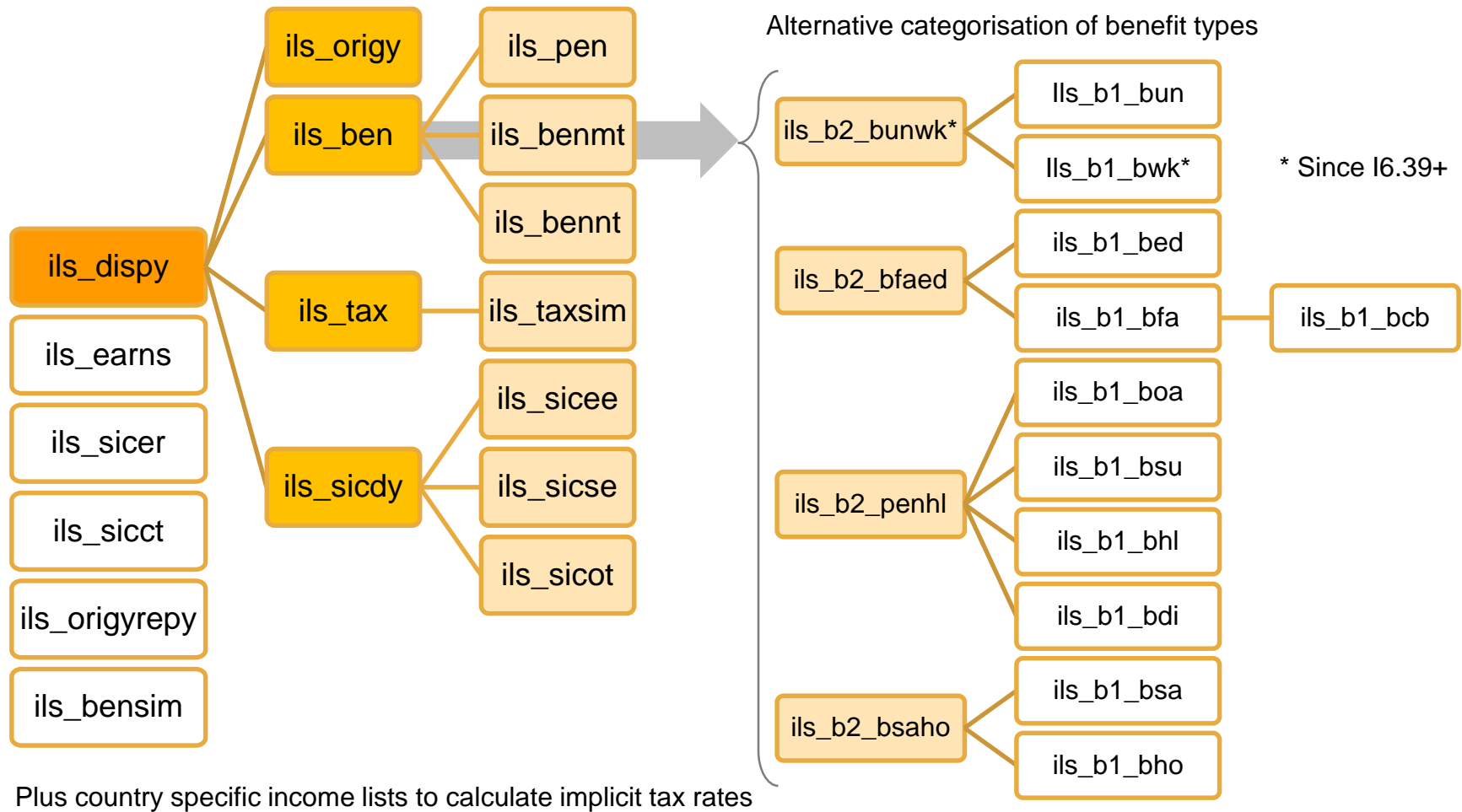
Policy	Grp/No	BG_2020	Comment
▼ ● ILDef_bg		on	DEF: NON-STANDARD INCOME CONCEPTS
▼ fx Defll		on	Taxable income (before any tax exemptions/deductions) (il_taxableY)
name		il_taxableY	
yemtx		+	Taxable employment income
yemmc_s		+	Covid-19 in 2020: 60/40 wage compensation contribution paid by the employer
bwkmcee_s		+	Covid-19 in 2020: 60/40 wage compensation contribution paid by the state
ysetx		+	Taxable self-employment income
ypr		+	Income from property
yot		+	Income received by children
ils_sicee		-	employee social insurance contributions
ils_sicse		-	self-employed social insurance contributions

Income lists: standard income lists (1)

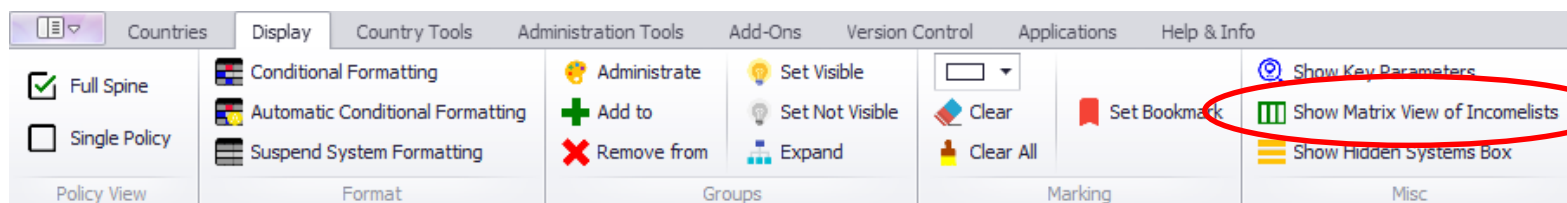
- Defined in every country
- Built in a comparable way to facilitate cross-national analysis
 - **ils_earns**: earnings
 - **ils_origy**: market incomes
 - **ils_pen**: public pensions
 - **ils_bennt**: non-means-tested benefits
 - **ils_benmt**: means-tested benefits
 - **ils_ben**: all benefits and public pensions
 - **ils_tax**: taxes
 - **ils_sicee**: employee SICs
 - **ils_sicse**: self-employed SICs
 - **ils_sicot**: other SICs paid by the individual
 - **ils_sicdy**: total SICs paid by individual
 - **ils_sicer**: employer SICs
 - **ils_sicct**: credited SICs
 - **ils_dispy**: disposable income
 - **ils_bensim**: simulated benefits
 - **ils_taxsim**: simulated taxes

Remember:
if you add a new policy that
simulates a new
benefit/tax, then you need
to add the benefit/tax to the
right income list!

Income lists: standard income lists (2)



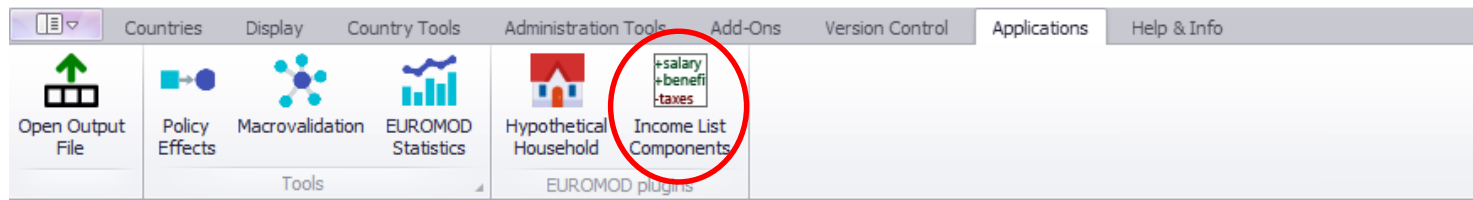
Income lists: matrix view



Matrix View of Incomelists for Bulgaria BG_2020

	il_bsaY	_extdispy	il_bchY	_taxable\	il_bsaht	il_pen	ils_eams	ils_origy	s_origrepy	ils_sicer	ils_sicct	ils_sicse	ils_sicee	ls_taxsim	ils_tax
yemtx	1	1	1	1	1		1	1	1						
bmaprct_s	1	1	1		1				1						
bmaycct_s	1	1	1		1				1						
bmapmc_s	1	1			1										
pdinc	1	1	1		0.7225...	1									
pdiuc	1	1	1		0.7225...	1									
pdi00	1	1	1		0.7225...	1			1						
psu	1	1	1		0.7225...	1			1						
poa00	1	1	1		0.7225...	1			1						
poadi	1	1	1		0.7225...	1									
bhl	1	1	1		1				1						
bunct_s	1	1	1		1				1						
yot	1	1	1	1	1			1	1						
ypr	1	1	1	1	1			1	1						
ysetx	1	1	1	1	1		1	1	1						

Income lists: components



Country & Year Selection

Please select the Country / Year combinations you wish to analyse

	1996	2005	2006	2007	2008	2009	2010	2011
AT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
BE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CZ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Export for CR View Components Close

Results

Income List Components

country	year	name	description	components	components d...
AT	2007	ils_origy	Original income	+ yem + yse + ...	+ employment in...
AT	2007	ils_pen	Public pensions	+ poa00 + psu ...	+ Old-age pensi...
AT	2007	ils_origrepy	Original replace...	+ yem + yse + ...	+ employment in...
AT	2007	ils_earn	Earnings	+ yem + yse	+ employment in...
AT	2007	ils_bensim	Simulated benefits	+ bch00_s + bc...	+ Main child ben...
AT	2007	ils_benmt	Means-tested be...	+ bho + bed + ...	+ Housing bene...
AT	2007	ils_benrt	Non-means test...	+ bhl00 + bac0...	+ Health benefit...
AT	2007	ils_ben	Total benefits	+ poa00 + psu ...	+ Old-age pensi...
AT	2007	ils_taxsim	Simulated taxes	+ tin_s + tiniv_s	+ Income tax (Ei...
AT	2007	ils_tax	Taxes	+ tin_s + tiniv_s	+ Income tax (Ei...
AT	2007	ils_sicee	Employee social i...	+ tsceehl01_s ...	+ employee SIC ...
AT	2007	ils_sicse	Self employed so...	+ tsceac_s + t...	+ self-employed...
AT	2007	ils_sicer	Employer social i...	+ tsceac_s + t...	+ employer SIC ...

Close

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	ils_origy																			
2	Variable	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	Description	
3	yem	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	employment income	
4	yse	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	self-employment inc	
5	yi	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	investment income	
6	yot01	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	income from military	
7	ypp01	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	additional occupatio	
8	ypp02	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	private pension (Priv	
9	ypt	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	Private transfers rec	
10	ypr	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	property income (Ein	

Using constants and income lists

- We can now read and understand the code written in the *SchedCalc* function using constants and income lists

	Policy	Grp/No	ES_2020	Comment
37.27	▼ fx Elig		on	General tax base higher than national tax allowance
37.27.1	Elig_Cond		tintbit_s-il_tintcit > 0	
37.27.2	TAX_UNIT		tu_individual_es	
37.28	▼ fx SchedCalc		on	National tax schedule on general income (cuota 1 general estattall)
37.28.1	Who_Must_Be_Elig		all	
37.28.2	base		tintbit_s	tax base
37.28.3	band_uplim	1	\$tin_ts_lt1	
37.28.4	band_uplim	2	\$tin_ts_lt2	
37.28.5	band_uplim	3	\$tin_ts_lt3	
37.28.6	band_uplim	4	\$tin_ts_lt4	
37.28.7	band_uplim	5	\$tin_ts_lt5	
37.28.8	band_uplim	6	\$tin_ts_lt6	
37.28.9	band_rate	1	\$tin_ts_rt1	
37.28.10	band_rate	2	\$tin_ts_rt2	
37.28.11	band_rate	3	\$tin_ts_rt3	
37.28.12	band_rate	4	\$tin_ts_rt4	
37.28.13	band_rate	5	\$tin_ts_rt5	
37.28.14	band_rate	6	\$tin_ts_rt6	
37.28.15	band_rate	7	\$tin_ts_rt7	
37.28.16	output_var		i_tiningt	gross income tax individual taxation
37.28.17	TAX_UNIT		tu_individual_es	tu: individual

Questions





Session 4

Policy function Allocate

Defining tax units

In this session you will learn about

- Allocating benefits and liabilities, using Allocate
- System function to define tax units: *DefTU*
- Exercise:
 - 7. Reforming PT Child Benefit by narrowing its age condition

Policy	System Name	Comment
Allocate	on	Made-up example: Allocate Housing Benefit to the person responsible for housing costs and council tax
Share	bho_s	Housing Benefit to be reallocated between the members of the assessment unit (b: benefit, ho: housing, _s: simulated)
Share_Between	dhr=1	share the benefit between those responsible for the housing (dhr=1)
Share_All_IfNoElig	yes	if no one is responsible for the housing, share among all members
output_var	bho_s	save result in bho_s: result of Allocate function overwrites the current value of the variable
TAX_UNIT	tu_bu_lv	assessment unit used for the calculations

POLICY FUNCTION *Allocate*

Result of a function

- It is always assigned to the **head of the assessment unit**
- For all other members of the unit and for those in not eligible units (defined by `who_must_be_elig`) :
 - `output_var` is set to zero.
 - `output_add_var` not changed or set to 0 if undefined before
 - `result_var` is set to zero.

Function *Allocate* (1)

- Default: result is assigned to the head of the assessment unit
- Function *Allocate* is used to share/allocate the amount of a variable among specific members of the assessment unit
 - E.g. share the income tax from a joint income taxation among the partners in a couple → important for the simulation of benefits which depend on individual after-tax income
- Parameters of function *Allocate*
 - **Share**: which variable to split
 - amount to split first summed up across assessment unit members
 - **Share_Between**: condition parameter; who are the members 'participating' in the split
 - default is all members of the assessment unit
 - **Share_Prop**: in what proportion to split between the various qualifying members (i.e. those satisfying the **Share_Between** condition)
 - default is sharing in equal proportions

Function *Allocate* (2)

- **Allocate** reallocates amounts between members of assessment units (subject to conditions)

Policy	System Name	Comment
Allocate	on	Made-up example: Allocate Housing Benefit to the person responsible for housing costs and council tax
Share	bho_s	Housing Benefit to be reallocated between the members of the assessment unit (b: benefit, ho: housing, _s: simulated)
Share_Between	dhr=1	share the benefit between those responsible for the housing (dhr=1)
Share_All_IfNoElig	yes	if noone is responsible for the housing, share among all members
output_var	bho_s	save result in bho_s: result of Allocate function overwrites the current value of the variable
TAX_UNIT	tu_bu_lv	assessment unit used for the calculations

Function *Allocate* (3)

- **Allocate** reallocates amounts between members of assessment units (subject to conditions)

Policy	System Name	Comment
Allocate	on	Made-up example: Allocate the income tax - based on joint taxation - between the couple's partners
Share	tin_s	Income Tax to be reallocated between the partners of the couple (t: tax, in: income, _s: simulated)
Share_Prop	yem	share the income tax in proportion to the persons' earnings
Share_equ_Ifzero	yes	if noone has earnings, then income tax is shared equally between the couple's partners
Ignore_neg_prop	yes	if someone's earnings are negative, they are ignored (i.e. considered to be zero)
output_var	tin_s	save result in tin_s: result of Allocate function overwrites the current value of the variable
TAX_UNIT	tu_couple_lv	assessment unit used for the calculations

ASSESSMENT (TAX) UNITS & THE SYSTEM FUNCTION *DefTU*

Policy	System Name	Comment
DefTu	on	Made-up example: individual
Name	tu_individual_lv	
Type	IND	
DepChildCond	dag<16 (dag<19 & !IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married
DefTu	on	Made-up example: household
Name	tu_household_lv	
Type	HH	
DepChildCond	dag<16 (dag<19 & !IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married
DefTu	on	Made-up example: nuclear family
Name	tu_family_lv	
Type	SUBGROUP	
Members	Partner & OwnDepChild & LooseDepChild	head, head's partner, own and loose dependent children
DepChildCond	dag<16 (dag<19 & !IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married

Assessment/Tax Unit

- Unit: group of household members to be considered together
- Defined with System function *DefTU* usually in policy *TUDef_xx*
- Defined the first time used by the model
 - (although can be updated – see *UpdateTU* in extra slides)
- Naming convention: *tu_yyyy_xx*
- Parameter *Type* in *DefTU* defines the composition of the tax unit
 - **HH**: all individuals of the household are in the same unit.
 - **IND**: each individual of the household forms its own unit.
 - **SUBGROUP**: individuals determined by parameter *Members* form a unit. The household may be split into several units of different size.

Type of assessment/tax unit

Policy	System Name	Comment
DefTu	on	Made-up example: individual
Name	tu_individual_lv	
Type	IND	
DepChildCond	dag<16 (dag<19 & !IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married
DefTu	on	Made-up example: household
Name	tu_household_lv	
Type	HH	
DepChildCond	dag<16 (dag<19 & !IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married
DefTu	on	Made-up example: nuclear family
Name	tu_family_lv	
Type	SUBGROUP	
Members	Partner & OwnDepChild	head, head's partner, own dependent children
DepChildCond	dag<16 (dag<19 & !IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married

Assessment/tax units: examples

description	idhh	idperson	idpartner	idmother	idfather	dag	Household	Individual	Family (i.e. Subgroup)
Couple with two children	1	101	102	0	0	30	A1	A1	A1
	1	102	101	0	0	28	A1	B1	A1
	1	103	0	102	101	3	A1	C1	A1
	1	104	0	102	101	1	A1	D1	A1
couple without children	2	201	202	0	0	56	A2	A2	A2
	2	202	201	0	0	55	A2	B2	A2
lone parent	3	301	0	0	0	35	A3	A3	A3
	3	302	0	301	0	6	A3	B3	A3
single	4	401	0	0	0	25	A4	A4	A4
two singles living together	5	501	0	0	0	22	A5	A5	A5
	5	502	0	0	0	23	A5	B5	B5
large family	6	601	602	606	0	48	A6	A6	A6
	6	602	601	0	0	45	A6	B6	A6
	6	603	0	602	601	20	A6	C6	B6
	6	604	0	602	601	15	A6	D6	A6
	6	605	0	602	601	10	A6	E6	A6
	6	606	0	0	0	70	A6	F6	C6

Head of the tax unit

- The **head of a tax unit** is by default
 - the richest member (System Configuration: Income for Head Definition)
 - the oldest
 - with the smallest personal id (i.e. *idperson*)
- Additional, optional specifications for the **head**:

Query	Value type	Default	Description
HeadDefInc	variable or incomelist	ils_origy	Incomelist used for determining who is the richest person in the assessment unit, see parameter ExtHeadCond.
ExtHeadCond	condition	!IsDepChild	Condition further defining the head of the assessment unit.
StopIfNoHeadFound	yes/no	no	If set to yes: an error is issued if ExtHeadCond rules out all household members. If set to no: no error issued, instead ExtHeadCond is dropped for affected households.
NoChildIfHead	yes/no	no	If set to yes (possible) child status is removed if person is the Head of the assessment unit.
NoChildIfPartner	yes/no	no	If set to yes (possible) child status is removed if person is Partner as defined by parameter PartnerCond.

Members of the tax unit (1)

- If tax unit type is SUBGROUP, parameter **Members** defines which hh members form a unit
- Relations are often defined with respect to the *head* of the unit
 - e.g. Partner, OwnChild, DepParent
- The status of each member is defined by a **xCond** parameter
 - e.g. PartnerCond, OwnChildCond, DepParentCond

Members of the tax unit (2)

Query	Default	Description
Members	n/a	Specifies which members of the household form a unit, if parameter Type is set to SUBGROUP, e.g. Partner & OwnChild.
PartnerCond	head:idperson=idpartner	Condition defining who is a partner, i.e. the partner of the head.
DepChildCond	0, i.e. nobody is a child	Condition defining who is a dependent child. The parameter equals 0, i.e. nobody is child, if not defined or set to n/a. However, setting the parameter to Default is the same as typing !IsParent & idpartner=0 .
OwnChildCond	head:idperson=idmother head:idperson=idfather partner:idperson=idmother partner:idperson=idfather	Condition defining who is an own child: either the head or the head's partner is the mother/father of the child.
OwnDepChildCond	IsOwnChild & IsDepChild	Condition defining who is an own dependent child.
LooseDepChildCond	idmother=0 & idfather=0 & IsDepChild	Condition defining who is a loose dependent child.
DepParentCond	head:idmother=idperson head:idfather=idperson partner:idmother=idperson partner:idfather=idperson	Condition defining who is a dependent parent, i.e. the parent of the head or the head's partner.
DepRelativeCond	0, i.e. nobody is a dependent relative	Condition defining who is a dependent relative.
LoneParentCond	IsParentOfDepChild & idpartner=0	Condition defining who is a lone parent.

Examples: Members of the tax unit (1)

- **head**: subsequent variable refers to the head
- **partner**: subsequent variable refers to the partner of the head
- **Default** setting can be overwritten or combined with further specifications

Policy	System Name	Comment
DefTu	on	Made-up example: couple with dependent parents
Name	tu_couple1_lv	
Type	SUBGROUP	
Members	Partner & DepParent	head, head's partner and dependent parents
DepParentCond	head:idmother=idperson head:idfather=idperson	overwriting the default condition: only include parents of the head
DefTu	on	Made-up example: couple
Name	tu_couple2_lv	
Type	SUBGROUP	
Members	Partner	head and head's partner
PartnerCond	Default & IsMarried	default condition (i.e. head's partner) + partner is married

Examples: Members of the tax unit (2)

- **DepChildCond**: determines who is the dependent child
- **OwnDepChild**: “a son or daughter” (see **OwnChildCond**), who is a dependent child
- **LooseDepChild**: “someone, who is a dependent child, but doesn’t cohabit with parent/s”

Policy	System Name	Comment
DefTu	on	Made-up example: nuclear family
Name	tu_family1_lv	
Type	SUBGROUP	
Members	Partner & OwnDepChild	head, head's partner and own dependent children
DepChildCond	dag<=15	definition of the dependent child: aged 15 or less

idhh	idperson	idpartner	idmother	idfather	dag	ils_origy	IsHead	IsPartner	IsDepChild	IsOwnDepChild	IsLooseDepChild	tax unit
1	101	102	0	0	44	2000	no	yes	no	no	no	A
1	102	101	0	0	45	2100	yes	no	no	no	no	A
1	103	0	102	101	14	400	no	no	yes	yes	no	A
1	104	0	0	0	12	0	yes	no	yes	no	yes	B

Examples: Members of the tax unit (3)

- **DepChildCond**: determines who is the dependent child
- **OwnDepChild**: “a son or daughter” (see **OwnChildCond**), who is a dependent child
- **LooseDepChild**: “someone, who is a dependent child, but doesn’t cohabit with parent/s”

Policy	System Name	Comment
DefTu	on	Made-up example: nuclear family
Name	tu_family2_lv	
Type	SUBGROUP	
Members	Partner & OwnDepChild & LooseDepChild	head, head's partner, own and loose dependent children
DepChildCond	dag<=15	definition of the dependent child: aged 15 or less

idhh	idperson	idpartner	idmother	idfather	dag	ils_origy	IsHead	IsPartner	IsDepChild	IsOwnDepChild	IsLooseDepChild	tax unit
1	101	102	0	0	44	2000	no	yes	no	no	no	A
1	102	101	0	0	45	2100	yes	no	no	no	no	A
1	103	0	102	101	14	400	no	no	yes	yes	no	A
1	104	0	0	0	12	0	no	no	yes	no	yes	A

Using conditions which refer to income

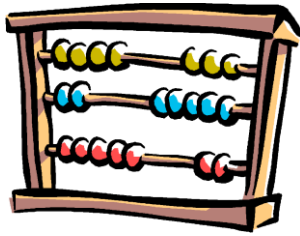
- If the assessment unit is bigger than the individual, monetary variables or income lists are **usually** assessed at the level of the whole unit.
- Not in the DefTu function!
- Assessment level is always the individual

Policy	Grp/No	System Name	Comment
DefTu		on	Made-up example: family
Name		tu_family_lv	
Type		SUBGROUP	
Members		Partner & OwnDepChild & LooseDepChild	
DepChildCond		dag<25 & yem < 1000#m	aged less than 25 and with (individual) earnings of less than GBP 1,000 per month

The tax unit in the output file

Policy	Grp/ No	System Name
DefTu		on
Name		tu_family_lv
Type		SUBGROUP
Members		Partner & OwnDepChild & LooseDepChild & DepParent
DepChildCond		dag<25 & yem<8000#y
DepParentCond		Default & dag>65 & yem<8000#y
LoneParentCond		Default & !IsMarried

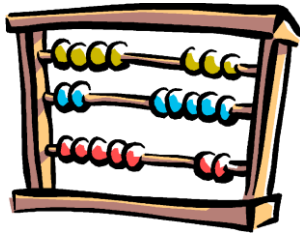
idhh	idperson	idpartner	idmother	idfather	dag	ils_orig	yem	tu_family_lv_Headed	tu_family_lv_IsPartner	tu_family_lv_IsDepchild	tu_family_lv_IsDePparent	tu_family_lv_IsLoneParent
1	101	102	0	0	65	0	0	101	0	0	0	0
1	102	101	0	0	60	0	0	101	1	0	0	0
1	103	0	102	101	30	0	0	103	0	0	0	0
1	104	0	102	101	28	166	147	104	0	0	0	0
2	201	202	0	0	29	1,085	1,007	201	0	0	0	0
2	202	201	0	0	25	953	891	201	1	0	0	0
2	203	0	202	201	3	0	0	201	0	1	0	0
2	204	0	202	201	2	0	0	201	0	1	0	0
92	9,201	0	0	0	80	0	0	9,202	0	0	1	0
92	9,202	0	0	9,201	38	3,740	3,502	9,202	0	0	0	1
92	9,203	0	0	9,201	34	2,483	2,324	9,203	0	0	0	0
92	9,204	0	0	9,202	11	0	0	9,202	0	1	0	0



Exercise 7

Reforming PT Child Benefit by narrowing its age condition

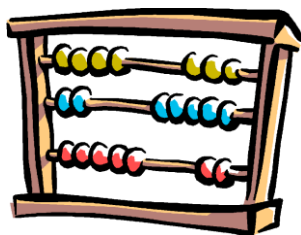
Limit the provision of the Child benefit in the Portugal in 2020, to families with children aged up to 13, i.e. families with children aged 14+ will lose the benefit. Currently the benefit is paid up to the age of 16 or until 24 if a child is in full-time education.



Exercise 7

Steps:

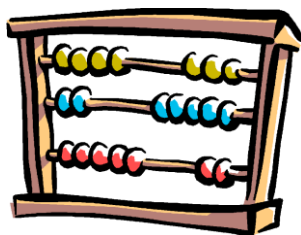
- Open the PT 2020 system
- Use the function DefTU to create a new assessment unit *tu_buref_pt* which includes the reformed age condition for a dependent child
 - Make a copy of PT 2020 system and call it e.g. PT_2020_E7
 - You can copy/paste an existing tax unit (*tu_bch_fa_pt*) and modify it
 - Reform the Child Benefit so that families with children aged 14+ no longer receive this benefit
- Run the new system and use the Statistics Presenter to analyse the impact of the reform on inequality and poverty.



Exercise 7: solution

2.2.a Mean equivalised income (monthly) by decile groups

	PT_2020 (base)	PT_2020_E7	Difference to base	% Difference to base
Decile 1	399.62	384.21	-15.40	-3.85 %
Decile 2	608.76	599.11	-9.65	-1.59 %
Decile 3	759.22	755.76	-3.47	-0.46 %
Decile 4	880.24	878.56	-1.68	-0.19 %
Decile 5	1,004.59	1,004.79	0.21	0.02 %
Decile 6	1,130.88	1,130.60	-0.29	-0.03 %
Decile 7	1,268.71	1,269.27	0.56	0.04 %
Decile 8	1,431.13	1,431.53	0.39	0.03 %
Decile 9	1,631.78	1,631.85	0.06	0.00 %
Decile 10	1,988.83	1,988.83	0.00	0.00 %
All	1,107.84	1,104.90	-2.95	-0.27 %
Poor	464.13	453.40	-10.72	-2.31 %



Exercise 7: solution

Market Incomes and Government Revenue & Expenditure ?

Yearly, mill., currency as defined in EM output

	PT_2020 (base) (annual)	PT_2020_E7 (annual)	Difference to base (annual)	% Difference to base
Total market incomes	70,807.13	70,807.13	0.00	0.00 %
... income from (self) employment	70,807.13	70,807.13	0.00	0.00 %
... other sources	0.00	0.00	0.00	0.00 %
Government revenue through taxes and social insurance contributions	33,860.71	33,860.71	0.00	0.00 %
... direct taxes	10,203.86	10,203.86	0.00	0.00 %
... employee social insurance contributions	7,255.79	7,255.79	0.00	0.00 %
... self-employed social insurance contributions	735.45	735.45	0.00	0.00 %
... other social insurance contributions	0.00	0.00	0.00	0.00 %
... employer social insurance contributions (not part of disposable income)	15,665.62	15,665.62	0.00	0.00 %
Credited social insurance contributions (not part of disposable income)	0.00	0.00	0.00	0.00 %
Government expenditure on social transfers	34,887.06	34,692.07	-194.99	-0.56 %
by target group				
... unemployment benefits	6,312.04	6,312.04	0.00	0.00 %
... family and education benefits	759.30	564.31	-194.99	-25.68 %
... social assistance and housing benefits	438.91	438.91	0.00	0.00 %
... pensions, health and disability benefits	27,376.80	27,376.80	0.00	0.00 %
... firms	0.00	0.00	0.00	0.00 %
by benefit design				
... means-tested non-pension benefits	1,198.21	1,003.22	-194.99	-16.27 %
... non-means-tested non-pension benefits	6,312.04	6,312.04	0.00	0.00 %
... pensions	27,376.80	27,376.80	0.00	0.00 %
... firms subsidies	0.00	0.00	0.00	0.00 %

Questions





Summary: Exercise 7

- You learned how to
 - Create a new assessment unit
 - Copy/paste a function
 - Modify those assessment units that are already defined into the model



Session 5

System function *DefOutput*

Upgrading indices

EUROMOD resources

In this session you will learn about

- System function to define output: *DefOutput*
- Updating indices and function *Uprate*
- EUROMOD resources
- Exercise:
 - 15. Introducing a benefit for secondary education in Simpleland

output_std_fr		on	on	on	on	on
fx	DefOutput	on	on	on	on	on
	File	FR_2018_std	FR_2019_std	FR_2020_std	FR_2021_std	FR_2023_std
	vargroup	id*	id*	id*	id*	id*
	vargroup	d*	d*	d*	d*	d*
	vargroup	l*	l*	l*	l*	l*
	vargroup	y*	y*	y*	y*	y*
	vargroup	p*	p*	p*	p*	p*
	vargroup	b*	b*	b*	b*	b*
	vargroup	t*	t*	t*	t*	t*
	vargroup	x*	x*	x*	x*	x*
	VarGroup	k*	k*	k*	k*	k*
	vargroup	a*	a*	a*	a*	a*
	VarGroup	n/a	n/a	n/a	n/a	n/a
	VarGroup	n/a	n/a	n/a	n/a	n/a
	VarGroup	n/a	n/a	n/a	n/a	n/a
	VarGroup	i_*	i_*	i_*	i_*	i_*
	VarGroup	n/a	n/a	n/a	n/a	n/a
	ilgroup	ils_*	ils_*	ils_*	ils_*	ils_*

SYSTEM FUNCTION *DefOutput*

EUROMOD output microdata

- Content manipulated in policy **output_std_cc**
 - Controls at which level info is outputted (e.g. individual, family or household)
 - Which variables to be included in the output
- Variables usually included:
 - All **variables** present **in the input microdata** file
 - **Simulated variables** (i.e. simulated taxes and benefits)
 - Standardised **income lists** (e.g. all benefits, all taxes)
 - (optional) non-standardised income lists
 - (optional) intermediate variables
 - (optional) tax/assessment unit identification info

System function *DefOutput*

- Determines the content of the output file

	Policy	Grp/No	RO_2018	RO_2019	RO_2020	Comment
35	output_std_ro		on	on	on	DEF: STANDARD OUTPUT INDIVIDUAL LEVEL
35.1	fx DefOutput		on	on	on	
35.1.1	file		RO_2018_std	RO_2019_std	RO_2020_std	
35.1.2	vargroup		id*	id*	id*	ID variables
35.1.3	vargroup		d*	d*	d*	Demographic variables
35.1.4	vargroup		l*	l*	l*	Labour market variables
35.1.5	vargroup		y*			Market income variables
35.1.6	vargroup		p*			Public pensions variables
			b*			Benefit variables
			tin*	tin*	tin*	Tax related variables
			tsc*	tsc*	tsc*	Social contribution related variables
			tpr*	tpr*	tpr*	
35.1.11	vargroup		a*	a*	a*	Asset variables
35.1.12	vargroup		x*	x*	x*	Expenditure variables
35.1.13	VarGroup		k*	k*	k*	
35.1.14	ilgroup		il_*	il_*	il_*	Non-standardized income lists
35.1.15	ilgroup		ils_*	ils_*	ils_*	Standardized income lists
35.1.16	nDecimals		5	5	5	
35.1.17	TAX_UNIT		tu_individual_ro	tu_individual_ro	tu_individual_ro	
35.1.18	UnitInfo_TU	1	tu_bsa_ro	tu_bsa_ro	tu_bsa_ro	
	UnitInfo_Id	1	HeadID	HeadID	HeadID	
	UnitInfo_TU	2	tu_bchmt_ro	tu_bchmt_ro	tu_bchmt_ro	
	UnitInfo_Id	2	HeadID	HeadID	HeadID	
	UnitInfo_TU	3	tu_family_ro	tu_family_ro	tu_family_ro	
	UnitInfo_Id	3	HeadID	HeadID	HeadID	
	UnitInfo_TU	4	tu_bcc_ro	tu_bcc_ro	tu_bcc_ro	
	UnitInfo_Id	4	HeadID	HeadID	HeadID	

names of output files:
cc_year_std

variables included in
the output file

TAX_UNIT: level of
aggregation

Tax assessment
unit
identification
info

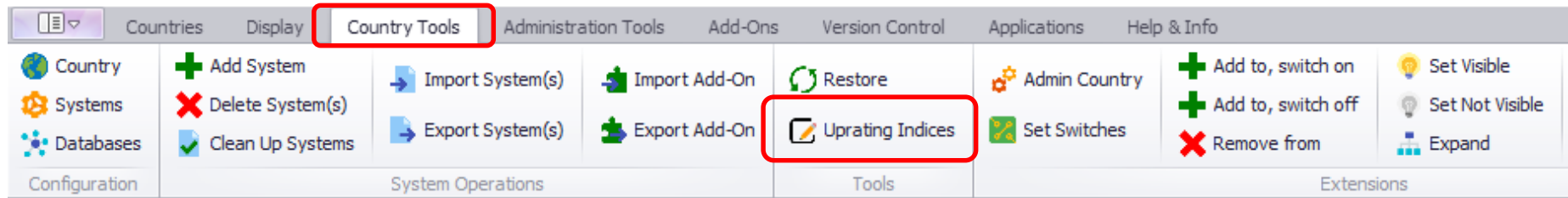
	Policy	Grp/No	EE_2018	EE_2019	EE_2020	Comment
1	▶ ● SetDefault_ee		on	on	on	DEF: DEFAULT VALUES FOR VARIABLES
2	▼ ● Uprate_ee		on	on	on	DEF: UPDATING FACTORS
2.1	▼ fx Uprate		on	on	on	apply uprating factors
2.1.1	Dataset		EE_20??_??	EE_20??_??	EE_20??_??	all EE datasets (except HHoT datasets)
2.1.2	Dataset		EE_20??_???	EE_20??_???	EE_20??_???	
2.1.3	afc		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.4	afcbd		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.5	afcsa		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.6	afcsh		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.7	bedet		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.8	bedot		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.9	bsa00		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.10	bsach		0	0	0	
2.1.11	bsals		\$f_cpi	\$f_cpi	\$f_cpi	

UPDATING INDICES SYSTEM FUNCTION Uprate

Why uprating monetary values?

- Income reference period of input microdata and policy system may not match...
 - e.g. data availability issue: survey data become available usually with several years lag
- ... so we need to adjust – i.e. *uprate* – monetary values in the input microdata to match the policy year
 - e.g. uprate 2018 input incomes to 2021 policy year
- We uprate monetary values by source
 - define uprating indices in the *Uprating Indices* table
 - apply the uprating indices to the specific monetary variables from the input microdata, coded in the spine
 - uprating indices do not account for population or behavioural changes between the data and policy year e.g. changes to the labour market

Uprating indices table



- Time-series information for the uprating indices (tab *Raw Indices*)
- EUROMOD calculates implicit uprating factors (tab *Factors per Data and System*)

Raw indices (time-series)

Updating Indices									
Raw Indices Factors per Data and System									
Index	Reference	2016	2017	2018	2019	2020	2021	Comment	
1 Hamonised Index of Consumer Prices	\$HICP	100.8	104.48	108.05	110.5	109.8	111.56	EUROSTAT; AMECO forecasts for 2021 values	
2 Consumer price index	\$f_cpi	1.6795	1.7367	1.7957	1.837	1.8297	1.8553	Statistics Estonia (table IA001); 2020 MoF forecast	
3 CPI housing expenditure index	\$f_xhcot	2.4884	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	21931	23858			27167	28906	Statistics Estonia (table RAA0012); 2020 MoF forecast	
5 Avg monthly salary, EUR	\$f_yem	1146	1221			1448	1454	Statistics Estonia (table PA005); 2020 MoF forecast	
6 Avg monthly salary (lag 1), EUR	\$f_yemlag1	1065	1146	1221	1310	1407	1448	Statistics Estonia (table PA005)	
7 Avg annual declared income from stocks, EUR	\$f_yiydv	3318	7325	2955	6212	3268	3283	Tax reports (row 6.1); 2021 wage growth	
8 Avg annual declared income from other sources, EUR	\$f_yiyit	1359	1276	2711	2131	2134	2144	Tax reports (until 2018 row 7.1, after row 5.6); 2021 wage growth	
9 Avg monthly salary, EUR	\$f_yiyot	15541	17062	17465	18652	18088	18168	Tax reports (row 6.3); 2021 wage growth	
10 Avg monthly salary, EUR	\$f_xhort	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 monthly salary, EUR	\$f_xhcm	565	538	631	693	766	766	Tax reports (row 9.2); 2021 kept constant	
12 Total annual land tax revenues, thous. EUR	\$f_tpr	58495	57708	57725	59081	58956	58956	Statistics Estonia (table RR02); 2021 kept constant	
13 Avg monthly old age pension (end year), EUR	\$f_poa0	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			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			200.06	215.87	234.08	258.39	262.52	Statistics Estonia (table SK110); 2021 official indexation	
16 Indexation of public pensions (current year), EUR			2.8421	3.0581	3.315	3.5802	3.6375	Statutory parameter	
17 Birth allowance (main rate), EUR	\$f_bchda	320	320	320	320	320	320	Statutory parameter	
18 Child allowance (1st child), EUR per month	\$f_bch00	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	Statutory parameter	
20 Parental allowance for large families, EUR per month	\$f_bcc0g	168.7	168.7	400	400	400	400	Statutory parameter	

the index description

user-defined index name(\$f_)

policy years

source of information

Factors per data and system

Uprating Indices

Raw Indices		Factors per Data and System						
Dataset	EE_2019_c1	Income Year		2018	Update			
Index	EE_2014	EE_2015	EE_2016	EE_2017	EE_2018	EE_2019	EE_2020	EE_2021
\$HICP	0.9248	0.9255	0.9329	0.967	1	1.0227	1.0162	1.0325
\$f_cpi	0.9391	0.9344	0.9353	0.9671	1	1.023	1.0189	1.0332
\$f_xhcot	0.9516	0.9373	0.9214	0.9407	1	1.029	0.9909	1.0048
\$f_gdp	0.778	0.8012	0.8455	0.9198	1	1.0838	1.0474	1.1144
\$f_yem	0.7672	0.813	0.8748	0.9321	1	1.074	1.1053	1.1099
\$f_yemlag1	0.7772	0.8231	0.8722	0.9386	1	1.0729	1.1523	1.1859
\$f_yiydv	1.2115	1.4406	1.1228	2.4788	1	2.1022	1.1059	1.111
\$f_yiyit	0.9133	0.4666	0.5013	0.4707	1	0.7861	0.7872	0.7909
\$f_yiyot	0.9641	0.9423	0.8898	0.9769	1	1.068	1.0357	1.0403
\$f_xhcr	0.7818	0.8636	0.8818	0.9545	1	1.0818	0.9909	1.0545
\$f_xhcmomi	0.9683	0.916	0.8954	0.8526	1	1.0983	1.2139	1.2139
\$f_tpr	1.0213	1.005	1.0133	0.9997	1	1.0235	1.0213	1.0213
\$f_poa00	0.7805	0.8274	0.8717	0.9313	1	1.0812	1.1799	1.1988
\$f_pdi	0.7435	0.7867	0.8295	0.8975	1	1.1113	1.2301	1.2498
\$f_psu	0.7852	0.8333	0.8818	0.9268	1	1.0844	1.197	1.2161
\$f_ipens	0.787	0.8366	0.8843	0.9294	1	1.084	1.1707	1.1895

- For the selected dataset, the implicit uprating factors for each system are shown, e.g.:
 - Dataset EE_2019_c1 → income referring to 2018
 - Prices (\$f_cpi) increased by 3.32% between 2018 and 2021

System function *Uprate* (1)

- Defines which indices to apply on each monetary variables in the input microdata

	Policy	Grp/No	EE_2019	EE_2020	EE_2021	Comment
2	Uprate_ee		on	on	on	DEF: UPDATING FACTORS
2.1	Uprate		on	on	on	apply updating factors
2.1.1	Dataset		EE_20??_??	EE_20??_??	EE_20??_??	all EE datasets (except HHoT datasets)
2.1.2	Dataset		EE_20??_???	EE_20??_???	EE_20??_???	
2.1.3	afc		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.4	afcbd		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.5	afcsa		\$f_cpi	\$f_cpi	\$f_cpi	
2.1.6	afcsh		\$f_cpi	\$f_cpi	\$f_cpi	
	bedet		\$f_cpi	\$f_cpi	\$f_cpi	
	bedot		\$f_cpi	\$f_cpi	\$f_cpi	
	bsa00		\$f_cpi	\$f_cpi	\$f_cpi	
	bsach		0	0	0	
	bsals		\$f_cpi	\$f_cpi	\$f_cpi	
	bsape		\$f_bsape	\$f_bsape	\$f_bsape	
	kivho		\$f_cpi	\$f_cpi	\$f_cpi	
	yds				1	used for validation only
	ydses_o				1	used for validation only
	yprro				\$f_cpi	
	yprho		\$f_cpi	\$f_cpi	\$f_cpi	
	xed		\$f_cpi	\$f_cpi	\$f_cpi	
	xcd		\$f_cpi	\$f_cpi	\$f_cpi	
	ypp02		\$f_cpi	\$f_cpi	\$f_cpi	
	ypp03		\$f_cpi	\$f_cpi	\$f_cpi	
	bchba		\$f_bchba	\$f_bchba	\$f_bchba	
	bch00		\$f_bch00	\$f_bch00	\$f_bch00	

variables
that are
updated

input datasets to which
uprating applies

name of uprating
factor

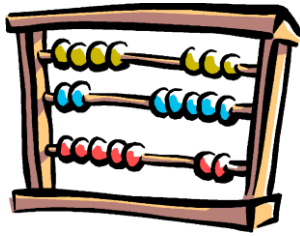
System function *Uprate* (2)

- Uprating the components of an aggregate variable

Uprate_bg		on
fx Uprate		on
dataset		BG_20??_??
yemtx		\$f_yem
yemn		\$f_yem
AggVar_Name	1	yem
AggVar_Part	1	yemtx
AggVar_Part	1	yemn
AggVar_Tolerance	1	1
ysetx		\$f_yem
yse		\$f_yem
AggVar_Name	2	yse
AggVar_Part	2	ysetx
AggVar_Part	2	yse
AggVar_Tolerance	2	1

- Using different uprating indices for different groups

uprate_el		on	DEF: UPRATING FACTORS
fx Uprate		on	apply uprating factors
dataset		EL_20??_??	all EL datasets
Factor_Con...	1	(pmfc = 4)	.. workers in public enterprises
yem	1	\$f_yem4	
Factor_Con...	2	(pmfc = 5)	.. banking employees
yem	2	\$f_yem5	
Factor_Con...	3	(pmfc = 7)	.. civil servants
yem	3	\$f_yem7	
Factor_Con...	4	(pmfc = 1) (pmfc = 8) (pmfc = 9)	".. other private sector employees (IKA, liberal professions, seamen)"
yem	4	\$f_yem189	
Factor_Con...	5	(pmfc = 0) (pmfc = 2) (pmfc = 3) (pmfc = 6) (pmfc = -1)	.. other (e.g. self-employed)
yem	5	\$f_yem	



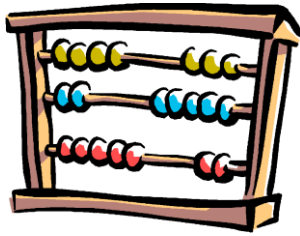
Exercise 16

Introducing a benefit for secondary education in Simpleland

Unlike real EU countries, Simpleland (SL) 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/month 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

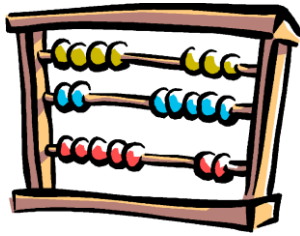
Implement this benefit in Simpleland and use the In-depth Analysis tool to analyse its impact.



Exercise 16

Steps:

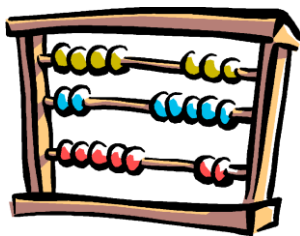
- Task 1. Check if the variable exists and add it if needed.
- Task 2. Create a new system by copy/pasting the baseline.
- Task 3. Create a new EUROMOD policy. Make sure that you place it correctly in the spine.
- Task 4. Add the variable to the relevant income lists.
- Task 5. Output additional variables and run baseline and reform.
- Task 6. Analyse the output with the In-depth Analysis tool.



Exercise 16

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 In-depth Analysis tool (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.



Exercise 16: solution

In-depth Analysis

1. Fiscal 2. Distributional by deciles of equivalised ils_dispy **2. Distributional by HH Type** 2. Distributional by values of les 2. Distributional by values of dgn 2. Distributional by Age group 3. In

2. Distributional ?

by values of hh_type

2.1. Individuals receiving the secondary education benefit ?

bedes_s

	Total SimpleLand 1996 (Baseline)	Total SimpleLand 1996 e16	Share SimpleLand 1996 (Baseline)	Share SimpleLand 1996 e16	Total SimpleLand 1996 e16 Diff. w.r.t. Baseline	Share SimpleLand 1996 e16 Diff. w.r.t. Baseline
One adult < 65, no children	0	0	0.0 %	0.0 %	0	0.0pp
- Female adult	0	0	0.0 %	0.0 %	0	0.0pp
- Male adult	0	0	0.0 %	0.0 %	0	0.0pp
One adult >= 65, no children	0	0	0.0 %	0.0 %	0	0.0pp
- Female adult	0	0	0.0 %	0.0 %	0	0.0pp
- Male adult	0	0	0.0 %	0.0 %	0	0.0pp
One adult with children	0	5,058	0.0 %	26.5 %	5,058	26.5pp
- Female adult	0	4,465	0.0 %	26.4 %	4,465	26.4pp
- Male adult	0	593	0.0 %	27.6 %	593	27.6pp
Two adults < 65, no children	0	0	0.0 %	0.0 %	0	0.0pp
Two adults, at least one >= 65, no children	0	0	0.0 %	0.0 %	0	0.0pp
Two adults with one child	0	8,432	0.0 %	10.3 %	8,432	10.3pp
Two adults with two children	0	8,996	0.0 %	7.2 %	8,996	7.2pp
Two adults with three or more children	0	2,791	0.0 %	5.5 %	2,791	5.5pp
Three or more adults, no children	0	0	0.0 %	0.0 %	0	0.0pp
Three or more adults with children	0	18,894	0.0 %	14.3 %	18,894	14.3pp
All	0	44,170	0.0 %	5.0 %	44,170	5.0pp

Questions





Summary: Exercise 16

- You learned how to:
 - Implement a new benefit from scratch
 - Use Simpleland (SL)
 - Analyse results using the In-depth Analysis tool



Country Reports

Reports for each EU member state



Model documentation

Documentation of the latest release



Statistics

Statistics on poverty, income and inequality

Resources

Resources (1)

- Built-in help (Help & Info tab or F1)
 - EUROMOD terminology
 - Running EUROMOD and basic concepts
 - EUROMOD functions (F5 = current function; F6 = parameters of current function)
 - Specific help for plug-ins, e.g. Statistics Presenter
- Documentation folder:
 - EUROMOD built-in help saved in pdf
 - HHoT manual
 - Add-ons' documentation
 - What's new document
- Data Requirement Document (DRD) - 1 per dataset
- Log folder:
 - EUROMOD version log

Resources (2)

- Country report
 - Main document accompanying each country model
 - Provides information about:
 - tax-benefit system
 - modelling decisions and limitations
 - underlying data
 - accuracy of simulations
 - Each report covers the policy systems
 - from the income year of the latest available input data (2020 for I6.0+)
 - to the most recent policy year (2023 for I6.0+)

Resources (3)

- Online documentation:
 - EUROMOD homepage: <https://euromod-web.jrc.ec.europa.eu/>
 - EUROMOD training material: </resources/training>
 - Country Reports: </resources/country-reports>
 - Model documentation: </resources/model-documentation>
 - Projects and publications using EUROMOD: </research>

Keep in touch



<https://euromod-web.jrc.ec.europa.eu/>



<https://euromod-web.jrc.ec.europa.eu/news-and-events/newsletters>



JRC-EUROMOD@ec.europa.eu

Questions, comments, suggestions?



Thank you



© European Union, Institute for Social and Economic Research, University of Essex, 2021-2024

Unless otherwise noted the reuse of this presentation is authorised under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license. For any use or reproduction of elements that are not owned by the EU, permission may need to be sought directly from the respective right holders.