EUROMOD training course

Fiscal Policy Analysis Unit Joint Research Centre European Commission

Online course based on EUROMOD I3.0+, software v. 3.4.6 19-21 October 2021





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

The reuse policy of the European Commission is implemented by the Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents (OJ L 330, 14.12.2011, p. 39). Except otherwise noted, the reuse of this document is authorised under the Creative Commons Attribution 4.0 International (<u>CC BY 4.0</u>) licence. This means that reuse is allowed provided appropriate credit is given and any changes are indicated. For any use or reproduction of photos or other material that is not owned by the EU, permission must be sought directly from the copyright holders.



Acknowledgement

This EUROMOD training is indebted to colleagues from ISER - University of Essex who made it happen until 2020. These slides and exercises were originally developed by them and have been updated by the JRC for this course.



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



European Commission

Course outline

- **Session 1:** Introduction to microsimulation models and EUROMOD.
- **Session 2:** EUROMOD language: policies, functions and parameters.
- **Session 3:** Policy functions *Elig & ArithOp*. EUROMOD handling errors.
- **Session 4:** Policy functions *BenCalc* & *SchedCalc*. Defining constants and income lists.
- **Session 5:** Policy function *Allocate*. Defining tax units.
- **Session 6:** Variable types and system functions *DefVar* & *DefOutput*. Uprating indices.



Session 1 Introduction to microsimulation models and EUROMOD



In this session, you will learn about

- Microsimulation models: what they are, how they work and why they are useful
- What is EUROMOD, the European Commission (EC) microsimulation tool
- Some applications of EUROMOD
- How EUROMOD works: Input microdata, Model (taxbenefit policy code), Software
- Exercise 1: Running the model and using the Statistics Presenter for analysis



What are microsimulation models

Encompass modelling techniques that:

- Operate at the level of <u>individual units</u> (e.g. persons, households, vehicles, firms)
- Apply <u>rules</u> to simulate changes in state or behaviour of these units
- Estimate <u>distributional outcomes</u> after applying these rules at the micro level
- Focus on different areas, e.g.:
 - Traffic and transportation
 - Demand for health care
 - Spatial planning
 - Tax-benefit (TB) policies



Types of TB microsimulation models

• Static:

• Pure policy effect \rightarrow <u>"morning-after" effect</u>

• Behavioural:

- Policy effect accounts for behavioural responses
 - Change in preferences estimated using microeconomic models (e.g. labour supply models)

• Dynamic:

- People's characteristics are adjusted over time in response to natural processes and probabilities of relevant events
 - > e.g. fertility, death, marriage/divorce, labour market status



Static TB microsimulation models

- <u>Unit</u>: households and persons living in the households
- <u>Rules</u>: tax-benefit policies, mostly:
 - Cash transfers
 - Means-tested benefits
 - Non-means-tested benefits
 - Pensions
 - Social insurance contributions (SIC)
 - Direct taxes: Income tax, including tax allowances and tax credits, capital gains tax
- <u>Outcomes</u>: income distribution, measures of income inequality and poverty, (net) budgetary cost of policy changes, gainers and losers from policy changes, indicators of work incentives... and changes to them



Why TB microsimulation models

• Account for <u>interactions</u> between parts of the TB system

- E.g. old-age pension is taxable, so higher pensions mean more income tax revenue
- E.g. means-tested benefits depend on after-tax earnings, so a rise in income tax rate means higher spend on benefits
- Allow for "<u>ceteris paribus</u>" type of analysis, looking into the impact of changes in one policy holding everything else constant (useful for "understanding" as well as "monitoring")
- Bring <u>added value to survey/register microdata</u>, by providing information which is otherwise not (publicly) available
- Produce <u>more up-to-date results</u> for simulated variables, as data collection and release take time



Outcomes of TB microsimul. models

- Models enable analysis of the impact of changes in a policy:
 - On the mean of an outcome across several units, as with regression techniques
 - For one specific individual unit, as with OECD-style standard family type calculations
 - On the whole distribution of an outcome across units (distributional analysis)
 - On the government's budget (fiscal analysis)
 - On work incentives



Why EUROMOD

Models are built and maintained by researchers (for academic purposes) or ministries and civil service (for impact assessment of policies)

Models are often not accessible (neither is the model documentation) Some use nonpublicly available data (e.g. administrative data)

All models are single country models

EUROMOD



General features of EUROMOD

- The static TB microsimulation model for the EU-27
- Open-access, flexible, free and transparent
- No hard-wiring, everything is parameterised
- (relatively) Easy to simulate structural reforms
- Extensive and growing documentation and community of users
- Uses flexible and user friendly interface
- Provides consistent and meaningful cross-country comparisons



Key users of EUROMOD



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



International policy organisations

European Commission policy Directorates OECD, IMF, World Bank, UNICEF (hands-on & commissioned work)



National governments and public institutions

SK, MT, LT, LV, EE, EL SPICe, Welsh Government, NHS Scotland

Civil society: EUROMOD is the engine for simplified web-based models

EUROMOD-JRC Interface - European Commission https://euromod-web.jrc.ec.europa.eu/euromod-jrc-interface SORESI - Austrian Ministry for Social Affairs http://soresi.sozialministerium.at/soresi



A platform for developing non-EU models

e.g. Africa (South Africa, Namibia, Mozambique, +4), Latin America (Ecuador, Colombia, Mexico...), Russia, Serbia ...



European Commission

Key uses of EUROMOD

- EUROMOD can be used to analyse the impact of actual, proposed or hypothetical TB policy changes
- It can also be used as a tool in other type of analysis by adjusting the input data set:
 - How would a rise in earnings inequality affect the distribution of income?
 - How does demographic change affect government revenues?
- Or by doing further analysis using the output data in combination with other models:
 - How does labour supply respond to a change in tax credits?



Additional uses of EUROMOD

- Complex reforms (e.g. revenue-neutral packages, policy swaps)
- Counterfactual ("what if") scenarios (e.g. stress test for unemployment shocks)
- Design regional/national/EU-wide policy reforms
- Extending policy scope with additional microdata
 - Indirect taxes (FRS + HBS data)
 - Wealth and property taxes (HFCS data)
 - Tax evasion and non-take-up simulation/calibration (administrative data)
- Link to macro models, providing a micro perspective within a macro type of analysis (e.g. links with CGE and DSGE models)
- Hypothetical Household Tool (HHoT), tool for creating households and individuals with user-defined characteristics



Typical elements of TB analysis

- Compare two tax-benefit systems, a "baseline" and a "reform"
- For a particular population (dataset)
- Look at the average change in equivalised disposable income within income deciles:
 - Compute *total disposable income* for each household (private income + benefits taxes)
 - Divide by that household's equivalence scale to get total equivalised disposable income
 - Rank all individuals by their household's equivalised disposable income, and form 10 *income decile groups*
 - Calculate average (mean) change in equivalised disposable income between baseline and reform systems within each income decile group



Example 1: hypothetical reform

• Average change in tax by household and income decile, introducing a new zero rate band of £1,000 above personal allowance in Scotland



New 0 rate band on £1,000 over PA

Source: Berthier & Hudson (2017), *SPICe Briefing Notes:* using EUROMOD and FRS for analysis for Scotland



Example 2: actual policy changes

• Decomposing the change in real net income due to policy changes in 2011-2014



European Commission

Example 3: policy swaps

Impact on household incomes of own and other countries' child-related policies



Source: Levy et al. (2007), Journal of Social Policy



21

Example 4: EU-wide reforms

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



Source: Jara, Sutherland & Tumino (2017), *EUROMOD Working Paper 3/17 Notes:* ISC measures percentage of earnings loss that is retained through TB system



Example 5: work incentives

• Decomposing the mean Marginal Effective Tax Rate by type of policy in 2015



Source: Jara, Gasior & Makovec (2017), *EUROMOD Working Paper 3/17 Notes:* METRs measure the incentive to work longer hours



How EUROMOD works



- microdata of individuals and households
- representative of the population

- Tax-benefit policy rules (*Model*)
- current or historical systems
- hypothetical systems



- for every individual/household
- under multiple taxbenefit systems
- interaction between
 policies



- all variables of input data, plus
- individual /household tax liabilities and benefit entitlements
- disposable income

Access

- Input data → upon request
- Model → open source and downloadable for free
- Software → open source code and end-user licenced executable, downloadable for free

Analysis

- EUROMOD Statistics Presenter & In-depth Analysis plugin
- any other software (e.g. Excel, Stata)



EUROMOD workflow





EUROMOD project folder

Documentation

EM3Translation

| Input

l Log

Output

XMLParam

- \rightarrow Help and documentation in pdf
- \rightarrow [folder for internal use of EUROMOD]
- \rightarrow Default folder to store input data
- \rightarrow EUROMOD log with all versions
- \rightarrow Default folder to save output data
- \rightarrow Country files, where policies are coded!



EUROMOD in this course

- Input microdata
 - Training data
- Model (tax-benefit code):
 - Version I3.0+
 - Models for all EU member states: policy years (systems) for 2005/07-2020 for most countries and 2011-2020 for Croatia
- EUROMOD Software version: 3.4.6





EUROMOD INPUT MICRODATA



Sources of input data

- Default datasets: based on household survey microdata from EU-SILC
 - Only a set of variables from EU-SILC end up in the EUROMOD input data: those relevant to TB simulation or for distributional analysis
 - If the user has access to both the original EU-SILC and EUROMOD microdata, they can combine the two files
- Any other income source can be adapted to be used in EUROMOD (e.g. administrative data), following a specific structure and modelling conventions
- EUROMOD's Hypothetical Household Tool (HHoT) allows to generate input files with model households
 - Households and individuals with user-defined characteristics
 - Abstracts from complexities of real data focus on specific household types (Gasior & Recchia 2019)



Characteristics of input data

- Variables: demographic, labour, income, assets, expenditure
- Gross incomes, at the individual level
- Monetary variables recoded to (average) monthly basis
- No missing values



A typical input dataset

				self- employment							
	identifiers					age	wage	e income pensior			
	idhh	idperson	idpartner	idmother	idfather	dag	yem	yse	роа		
1	1	101	102	0	0	65	0	0	2429.72		
2	1	102	101	0	0	60	0	0	0		
3	1	103	0	102	101	30	0	0	0		
4	1	104	0	102	101	28	157.42	0	0		
5	2	201	202	0	0	29	1075.35	0	0		
6	2	202	201	0	0	25	0	951.36	0		
7	2	203	0	202	201	3	0	0	0		
8	2	284	0	0	0	2	0	0	0		
9	3	301	302	0	0	72	0	0	1959.98		
10	3	302	301	0	0	59	0	0	0		
15	5	501	0	0	0	86	0	0	1853.7		





EUROMOD VIA THE USER INTERFACE (UI)



Opening a project

			EUROMOD fidel (r:\b2\04 - euromod jrc\06 - working area\02 - developers\fidel\)	<u></u> 2		\times	
		Save Country (Ctrl+S)	on Tools Add-Ons Version Control Applications Help & Info				
	=	Save Country As					
	E1						
		New Project					
(iii 0	pen Project			_		×
	Proje	ct Folder r:\b2\01 - households\02 - te	chnical assistance\02 - fiscalta2 (lt-ro-sk, 2020-2023)\02 - ro\03 - trainings\01 - training 1\euromo	d i2.0+	λ	~	
			✓ OK X Cancel				
		Close All Countries/Add-Ons					
	+	Exit					
			Tav	teizer	2		
			102	wize.		14	



Configuring the output/input folders



Commission







RUNNING EUROMOD, PRODUCING OUTPUT DATA & ANALYSIS


Running the model

- Two ways of Running the model:
 - by clicking on button *Run EUROMOD*
 - by clicking on a country flag and then clicking on the button Run EUROMOD
- We can run the model (simultaneously) for:

i) one country and one or more systems

ii) several countries (and systems) iii) all countries and systems and for all data



Analysing the output data

- Running the model produces a txt file containing the output data. This will include all the input data variables plus the variables simulated by the model (with subscript _s)
- The file is stored in the Output folder and for the baseline model it is called cc_year_std (e.g. pt_2006_std)
- This file can be imported to any statistical software for analysis. Additionally, the EUROMOD tool Statistics Presenter can be used to produce several indicators (more on this in the next slides)



A typical output dataset

			identifier	S		age	wage	self- employmen income	it pensior	personal income tax
	idhh	idperson	idpartner	idmother	idfather	dag	yem	yse	poa	tin_s
1	1	101	102	0	0	65	0	0	2429.72	176.58
2	1	102	101	0	0	60	0	0	0	0
3	1	103	0	102	101	30	0	0	0	0
4	1	104	0	102	101	28	157.42	0	0	0
5	2	201	202	0	0	29	1075.35	0	0	4.32
6	2	202	201	0	0	25	0	951.36	0	36.94
7	2	203	0	202	201	3	0	0	0	0
8	2	204	0	0	0	2	0	0	0	0
9	3	301	302	0	0	72	0	0	1959.98	82.22
10	3	302	301	0	0	59	0	0	0	0
15	5	501	0	0	0	86	0	0	1853.7	101.49



The Statistics Presenter: uses

• The tool estimates:

- Fiscal costs: government revenues and expenditures
- Distributional and poverty measures: poverty rates, gini index, S80/20 ratio, income shares, decomposition of income sources, winners and losers
- The tool calculates statistics for:
 - A single or several countries and policy systems
 - Baseline and reform scenarios and differences
 - Indicators
 - A specific variable (e.g. income taxes)
- Results can be exported in Excel tables



The Statistics Presenter: concepts (1)

Disposable income

Market Income

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



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

Social transfers

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



The Statistics Presenter: concepts (2)

Equivalence scales

- Larger household size → higher income needs but
- Sharing expenses and assets \rightarrow needs do not double when size doubles

$$= 1$$

$$= 1 + 0.5 + 0.3 = 1.8$$

$$= 1 + 0.5 = 1.5$$

$$= 1 + 0.5 + 0.3 + 0.3 = 2.1$$

OECD-modified equivalence scale First adult = 1 Each additional member $\ge 14 = 0.5$ Each additional member < 14 = 0.3



The Statistics Presenter: concepts (3)

Equivalised disposable income



(equivalised disposable income is attributed to each member of the household)



The Statistics Presenter: concepts (4)

Income distribution



Commission



The Statistics Presenter: concepts (5)

At-risk-of-poverty rate

Equivalised disposable income





The Statistics Presenter: concepts (6)

At-risk-of-poverty gap









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!







Steps:

- 1. Run EUROMOD for Austria (system: 2019).
- 2. Run EUROMOD for Austria, Finland, Italy and Spain together (system: 2020).
- 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.



Step 1: Run AT 2019 (I)



Step 1: Run AT 2019 (II)

EUROMOD Run started 09/04/2021 16:47:03 and finished 09/04/2021 16:47:04



50





						s O Add-ons
Run	Country	System	Dataset			
\checkmark	AT	at_2007	AT_2008_a8 (Best Match)	\sim	1	1
\checkmark	AT	at_2008	AT_2008_a8 (Best Match)	\sim		
\checkmark	AT	at_2009	AT_2010_a4 (Best Match)	~		
\checkmark	AT	at_2010	AT_2010_a4 (Best Match)	~	_	
\checkmark	AT	at_2011	AT_2012_a8 (Best Match)	~	select all	unselect a
\checkmark	AT	at_2012	AT_2012_a8 (Best Match)	\sim	countries/	countries/
\checkmark	AT	at_2013	AT_2014_a6 (Best Match)	\sim	countries/	countries/
\checkmark	AT	at_2014	AT_2015_a3 (Best Match)	\sim	all systems	all systems
\checkmark	AT	at_2015	AT_2016_a3 (Best Match)	\sim		
\checkmark	AT	at_2016	AT_2017_a3 (Best Match)	\sim		
\checkmark	AT	at_2017	AT_2018_a1 (Best Match)	\sim		
\checkmark	AT	at_2018	AT_2018_a1 (Best Match)	\sim		
\checkmark	AT	at_2019	training_data	\sim		
\checkmark	AT	at_2020	AT_2018_a1 (Best Match)	\sim		
\checkmark	ES	ES_2005	ES_2006_a3 (Best Match)	\sim		
\checkmark	ES	ES_2006	ES_2007_a4 (Best Match)	\sim		
\checkmark	ES	ES_2007	ES_2008_a2 (Best Match)	\sim		
\checkmark	ES	ES 2008	ES 2008 a2 (Best Match)	\sim		

\\s-jrcsvqfs200p.net1.cec.eu.int\iptsmods\b2\04 - euromod jrc\08 - trainings\00 - general training\02 - exercises\exercise 01\euromodfiles_i3.0+_training_ex01\output\ Output path



Systems

Systems

EI

ET

DE

Step 2: Run AT, FI, IT & ES 2020 (II)





Step 3: Statistics Presenter Default (I)

	ountries	Display (Country Tools	Administ	ration Tools	Add-Ons	Version Control	Applications	Help & Info
Open Output File	Policy Effects	Hypothetical Household	In-depth Analysis	Statistics Presenter					
	Tools 🔺	EUF	ROMOD plugir	IS					





Step 3: Statistics Presenter Default (II)

Statistics Presenter - Select Files

 \times



Step 3: Statistics Presenter Default (III)

Statistics Presenter - Select Files

 \times

Commission



Step 3: Statistics Presenter Default (IV)

🥬 Summary Statistics - Default

- 0 X

Commission



Step 4: Statistics Presenter MultiSystem (I)







Step 4: Statistics Presenter MultiSystem (II)

Statistics Presenter - Select Files

 \times

Commissior

S	ummary Statis	stics - Mı	ultiSyste	em
Path:	\\s-jrcsvqfs200p.net1.cec.eu.int\iptsmo	ds\b2\04 - euromod jrc\08	- trainings\00 - general	training\02 - exercises\exercise 01\euromo
at_20	019_std.txt			
{0} at	_2020_std.txt : 2020_std.txt			
{2} fi_	_2020_std.txt			
{3} it_	_2020_std.txt			
				Use Ctrl or Shift to select multiple files The order of selection will be retained
		ОК	Cancel	Is your file not visible?
			1.1	
				Fure

Step 4: Statistics Presenter MultiSystem (III)

🌮 Summary Statistics - MultiSystem

Summary	Statisti	cs - I	Multi	ple Sy	rstems		[0
Fiscal Overview	Poverty	Inequ	uality	Mean ho	ousehold income	Mean inco	me (equ)	Meta	idata
Basic Pove	rty Indi	ces	₽ or	ne colu	mn per syste	em			
			Pover	rty Risk for t_2020	Poverty Risk for ES_2020	Poverty FI_2	Risk for 020	Poverty IT_2	Risk for 2020
Population				7.54 %	25.24 %	24.6	0 %	22.3	38 %
Children				2.72 %	24.90 %	21.4	0 %	17.5	51 %
Working Age			:	10.97 %	31.17 %	30.9	2 %	28.8	30 %
Working Age Ec	conomically A	ctive		6.43 %	20.04 %	17.5	8 %	16.8	32 %
Elderly				0.00 %	1.99 %	3.4	8 %	2.9	99 %
Poverty Line				980.02	937.48	82	8.22	80	2.42
Poverty Gap				6.39 %	31.31 %	16.2	4 %	34.0	05 %



 \times





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











Session 2 EUROMOD language: policies, functions and parameters



In this session, you will learn about

- The building blocks of EUROMOD tax-benefit microsimulation language:
 - I. Policies
 - II. Functions
 - III. Parameters





EUROMOD TAX-BENEFIT MICROSIMULATION LANGUAGE



Working environment

ж	٠	*											• •						naste	er_versi	on_i3.4	4 (\\ne	t1.cec.e	eu.int\	jrc-ser	vices\	svq-u	isers\ri	ccmat	\deskt	op\euro	mod_	master_\	/ersi
		Countri	es	Display	/ C	Country T	ools	Adm	ninistrat	ion To	ols	ροι	IC	y s	ys	ste	ems	5	lp & 1	Info														
(9											<u>=</u>	F	+		-8							+				۲		+-	•				*
EUR	Run .OMOD	Unite Kingd	ed om	AT	BE	BG	CY	CZ	DE	DK	EE	EL	S	FI	FR	H	R HU	IE	IT	LT	LU	LV	MT	r N	IL F	1	PT	RO	SE	SI	SK	SL	UK	₹
		loade	ed						1				ŀ	-																				
		Policy					Gr	rp/No	U	18	UK_2019	UK_2	020	Comme	ent																			
15		* • •	yem_	uk								swite	ch	DEF: I apply	1inimı 7)	um w	rage (sv	ritch or	n/off t	0														
16		۰.	covst	ocks_	uk	by		lic	kir	ng	lr y	on		DEF: 9 unem (i.e. e take-	Simula Iployn Iigibili IID	ntion nent ity to	of Covic change cORS)	l-19 sh s, furlo and SE	ocks: ughing ISS)														
17		، ۱	cjrs_	uk		or	n a	fla	ag		n/a	on		BEN: (Coror (CJRS	ovid- Tovid- Taviru 5) in 20	19 sı s Job 020	ubsidy f Reten	rom th tion Scl	e heme															
18		، ۱	seiss	_uk					n/a		n/a	on		BEN: (Self-e Schei	Covid- employ me (SI	19 gi ymei EISS)	rant fro nt Incor) in 2020	m the ne Sup)	port					•	<u> </u>	m	m	٥n	te	7				
19		+ •	neg_	uk					on		on	on		DEF: r empl	ecode oyed i	e to O ncon) negati ne	ve self							00	•••			13					
20	1	، (mif_u	ık					on		on	on		DEF: 0 self-e Minin of Un	Calcula employ num Ir iversa	ate g yed a ncom il Cre	ross ea as part o ne Floor edit	rnings of the (MIF)	for the	2														
21		۰ ۱	lha_u	k					on		on	on		DEF: L rates	ocal H	lousi	ing Allov	vance	(LHA)															
22	•	' ⊦ ●	bmac	t_uk					swite	h	switch	swite	ch	BEN: 9 simul	itatut ated i	ory I n the	Maternii e baselii	ty Pay ne)	(not															
23		•	bman	ic_uk					swite	h	switch	swite	ch	BEN: I simul	1ateri ated i	nity / n the	Allowan e baselir	ce (not 1e)																
24		•	bpact	t_uk					swite	h	switch	swite	ch	BEN: 9 simul	statut ated i	ory F n the	Paternit e baselii	y Pay (ne)	not															

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 the policy intends to simulate
- policies are run in the order they appear
 order of policies is called the 'spine'



Structure of a policy

- Policy column includes policy name, function and parameter names
 order of policies and functions reflects 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	5	Comments column
	System#1	System#2	System#3	
Policy#1 name	policy switch	policy switch	policy switch	Policy description
Function#1	function switch	function switch	function switch	Function description
Parameter#1	value	value	value	paremter description
Parameter#2	value	value	value	
Function#2	function switch	function switch	function switch	
Parameter#1	value	value	value	
Parameter#2	value	value	value	
Parameter#3	value	value	value	



Example of a policy

	poli	cy colun	าท			systen	n colui	nns	comments columns	•		
		Ļ]				1					
		Policy			Grp/No	UK_2016	UK_2017	UK_2018	Comment			
pol	icy	boaht	_uk			policy switch			BEN: pensioner's annual heating allowance (Winter Fuel Allowance)			
nar	ne		Calc			on	on	on	Benefit calculation			
30.1.	.1		comp Con	طر	1	functi	on swit	ch)				
30.1	.2 ft	unction# s param	1 and eters		1	\$WFA	\$WFA	\$WFA	amount if anybody in hh aged 60 or over			
30.1	30.1.3 #_AgeMin				1	\$SPAfem	\$SPAfem	\$SPAfem	par 169			
30.1	.4		comp_Con	d	2	(nPersInT u#2 >= 1)	(nPersInT u#2 >= 1)	(nPersInT u#2 >= 1)				
30.1	.5		comp_per]	τυ	2	\$WFAAdd 80	\$WFAAdd 80	\$WFAAdd 80	additional amount if anybody in hh aged or over par68	1 80		
30.1	.6		#_AgeMin		2	80	80	80				
30.1	.7		output_va	r		boaht_s	boaht_s	boaht_s				
30.1	.8		TAX_UNIT			tu_hou	tu_hou	tu_hous				
30.2		- fx Allo	cate			on	on	on				
30.2	.1		share			boaht_s	boaht_s	boaht_s				
30.2	i fur	nction#2	and	veen		(dag >= \$SPAfem)	(dag >= \$SPAfem)	(dag >= \$SPAfem)	par 169			
30.2	its	paramet	ers	fnoelig		yes	yes	yes				
30.2	.4		• -	}		boaht_s	boaht_s	boaht_s				
30.2	.5		TAX_UNIT			tu_hou	tu_hou	tu_hous				



Structure of a function





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

A	dd Pai	rameters					×
	Aritl	nOp (order: 4) in policy	y BTA_u	k		
	Add	Parameter	Replaces	Grp/No	Count	Description	*
		Output_Add_Var	output_var			Variable for storing the result of the f	
		Result_Var				Variable for storing the result of the f	
		Who_Must_Be_Elig				Function's calculations are carried o	
		Elig_Var				Variable indicating whether a person	
		Run Cond				Function is only carried out if the con	







Child benefits reforms

Implement a Child Benefit reform in Finland. Increase the monthly benefit amount for the first child from 94.88€ to 100€.

We will do the exercise together!







Steps:

- You will make a copy of the FI 2020 system and call it FI_2020reform.
- Then, you will modify the reform system FI_2020reform 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 (which is equal to €94.88 per month) to 100#m (€100 per month).
 - Save your changes.
- Run the model and use the Statistics Presenter Baseline/Reform option to analyse the distributional effect of the reform.


Hiding systems

	Coun	tries	Display	Country	y Tools	Adr	ninistrat	ion Tools	A	\dd-Ons	Vers	sion Cor	ntrol A	pplication	ns H	elp & Ir	ıfo								
Ð	ł		=•		3														+		¢)			*
Run EUROMOD	Finla	and	AT B	BG	CY	CZ	DE	DK	EE	EL	ES	FI	FR HR	HU	IE	IT		 		NI	PI F	т оф	RO	SE SI SK SI UK	Ŧ
	load	ded															ng	nt-	CIIC	κΟ	n sy	SI	en	n name	
	Polic	y			Grp/No	D	FI_201	.4	E	I_2015		FI_2	016	FI_2	2017		FI_2018		FI_20	19	FI_2	020)	Comment	
1	+ (Set	Default_fi		1		on		0	n		on		on			on		on		on			Copy/Paste System	
2	+ () up	rate_fi				on		0	n		on		on			on		on		on			Rename System	
3	+ (Co	nstDef_fi				on		0	n		on		on			on		on		on			Delete System	
4		ilso	def_fi				on		0	n		on		on			on		on		on			Restore System Order	TS
5	+ (ils	UDBdef_fi				on		0	n		on		on			on		on		on			Save System Order	
6	+ (Ini	tVars_fi				on		0	n		on		on			on		on		on	_		Save System order	
7		ID	ef fi				on		0	n		on		on			on		Selected S	System				Move To Hidden Systems Box 1	
		Da	ndom fi			_	n/a			/a		n/a		n/a			n/a		All System	ns But S	elected			Show Matrix View of Incomelists	
0		Kd				_	ηγα			/a		Π/α		170			Π/α	:	Select Sys	tems				Insert First Policy	
9	+ (Tra	ansLMA_fi				n/a		n	/a		n/a		n/a			n/a		All System	ns To Th	ne Left		→	Best Fit	;
																			All System	ns To Th	ne Right			Rest Fit (all columns)	
10) × (tu	def_fi				on		0	n		on		on			on		Unhida Al	II Custom				bescric (an columns)	
11	+ () ye	m_fi				switch	ı	s	witch		swit	tch	swit	tch		switch		onnide Al	ii systen	115			Best Fit (all system columns)	
12	+ (ne	g_fi				on		0	n		on		on			on		Show Hid	den Sys	tems Box			Set Width Column	0
																								Set Width All System Columns	
																								Louid-19 for the celt-employ	hou



Adding a reform system

	Cour	ntrie	s Disp	Display Country T			ry Tools Administration Tools			Add-On	s	Version	Control	Ap	
Run EUROMOD	Fin	land	AT	BE	BG	€ CY	cz	DE	DK	EE	EL	€ S	FI	FR	¥ HR
	loa	ded													
	Polic	y				Grp/No	o FI_2	2020	C	Copy/Pa	ste Svst	em			
37	•	↑	bed00_fi				on		R	ename	System				:5
38		ſ	tinmu_fi				on		D	elete Sy	/stem				ed
39	- I	•	bhope_fi				on		R	estore S	System (Order			nce
40	F (•	bho00_fi				on		S	ave Sys	tem Ord	ler			e
41	- × •	•	bch_fi				on		N	/love To	Hidder	n Syste	ems Box	· >	
42	F (•	bsa00_fi				on		S	how M	atrix Viev	w of In	ncomeli	sts	
43	(e 1	•	TCO_fi				swit	tch	Ir	nsert Fir	st Policy	/			
44	F (•	output_s	td_fi			on	ĥ	В	est Fit					
45	(e - 1	•	parben_o	output	_std_fi		swit	tch	В	est Fit (all colur	mns)			
46	F (•	output_s	td_hh	fi		off		В	est Fit (all syste	m col	umns)		
									S	et Widt	h Colun	nn			
÷									S	et Widt	h All Sys	tem C	Column	s	



Adding a reform system (2)





Implementing a reform (1)

	Right expar	click to nd policy	Grp/No	FI_2020	FI_2020reform	Comment
41	e bo	h_fi		on	on	BEN: Child benefit
41.1	→ fx	DefConst		on	on	Constants for Child Care Benefits
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



Implementing a reform (2)

	Polic	у			Grp/No	FI_2020)	FI_2020)reform	Comment
41	- (bo	h_f	ì		on		on		BEN: Child benefit
41.1	,	- fx	De	fConst		on		on		Constants for Child Care Benefits
41.1.1				\$bch1		94.88#	m	94.88#	m	Child benefit for the 1st child
41.1.2				\$bch2		104.84	#m	104.84;	#m	Child benefit for the 2nd child
41.1.3				\$bch3		133.79	#m	133.79	#m	Child benefit for the 3rd child
41.1.4				\$bch4		163.24	#m	163.24	#m	Child benefit for the 4th child
41.1.5				\$bch5		182.69	#m	182.69	#m	Child benefit for the 5th child & subsequent
41.1.6				\$bchsup		63.30#	m	63.30#I	m	Child benefit supplement for a lone parent
41.2		⊧ fx	Elig)		on		on		Child benefit: eligibility
41.3		, fx	Ber	nCalc		on		on		Child benefit: amount
41.3.1				who_must_be_elig		all		all		
41.3.2				comp_cond	1	IsNtoM	child#1=1	IsNtoMo	thild#1=1	
41.3.3				#_N	1	1		1		
41.3.4				#_M	1	1		1		
41.3.5				comp_cond	2	IsNtoM	child#2=1	IsNtoMo	child#2=1	
41.3.6				#_N	2	2		2		
41.3.7				#_M	2	2		2		
41.3.8				comp_cond	3	IsNtoM	child#3=1	IsNtoMo	child#3=1	
41.3.9				#_N	3	3		3		
41.3.10				#_M	3	3		3		
41.3.11				comp_cond	4	IsNtoM	child#4=1	IsNtoMo	thild#4=1	
41.3.12				#_N	4	4		4		
41.3.13				#_M	4	4		4		
41.3.14				comp_cond	5	IsNtoM	child#5=1	IsNtoMo	thild#5=1	
41.3.15				#_N	5	5		5		
41.3.16				#_M	5	99		99		
41.3.17				comp_perElig	1	\$bch1		\$bch1		Amount for the 1st child
41.3.18				comp_perElig	2	\$bch2		ébeb2		Amount for the 2nd child
41.3.19				comp_perElig	3	\$bch3	\$bch1 = 9	4.88#m		Amount for the 3rd child
41.3.20				comp_perElig	4	\$bch4		\$bch4		Amount for the 4th child
41.3.21				comp_perElig	5	\$bch5		\$bch5		Amount for the 5th & subsequent children
41.3.22				output_var		bch_s		bch_s		Child benefit (without lone parent supplement)
41.3.23				TAX_UNIT		tu_fami	ly17_fi	tu_fami	y17_fi	
41.4		⊧ fx	Elig)		on		on		Eligibility for Lone parent supplement
41.5		⊧ fx	Ari	thOp		on		on		Child benefit: single parent supplement



Implementing a reform (3)

Type the new amount in the reform system FI_2020reform Policy Grp/No FI_2020 Comment BEN: Child benefit 41 bch fi -ON ON Constants for Child Care Benefits 41.1 ON ON Child benefit for the 1st child \$bch1 94.88#m 41.1.1 100#m \$bch2 Child benefit for the 2nd child 104.84#m 104.84#m 41.1.2 \$bch3 Child benefit for the 3rd child 133.79#m 133.79#m 41.1.3 \$bch4 Child benefit for the 4th child 163.24#m 163.24#m 41.1.4 \$bch5 Child benefit for the 5th child & 41.1.5 182.69#m 182.69#m subsequent \$bchsup Child benefit supplement for a lone 63.30#m 63.30#m 41.1.6 parent



Producing output microdata (1)

∲ - ∓						Run	EURON	IOD							_		×
Main	View / Filter	/ Add-Ons	Advanced Settings														\sim
AT BE	BG CY CZ	DE DK EE	EL ES FI FR HR H	HU II Select	E IT LT	LU LV	* MT	NL PL	PT R	D SE	SI SK	(SL	UK =	Countries Substants Countries Substants Countries Substants Select all ,	O Coun O Syste O Add- / Unselect al	tries ems ons	Run
Run	Country	System	Dataset														
	FI	FI_2007	FI_2008_a3 (Best Match)	~													
	FI	FI_2008	FI_2008_a3 (Best Match)	~													
	FI	FI_2009	FI_2010_a2 (Best Match)	\sim													
	FI	FI_2010	FI_2010_a2 (Best Match)	\sim													
	FI	FI_2011	FI_2012_a3 (Best Match)	\sim													
	FI	FI_2012	FI_2012_a3 (Best Match)	\sim													
	FI	FI_2013	FI_2014_a3 (Best Match)	\sim													
	FI	FI_2014	FI_2015_a3 (Best Match)	\sim													
	FI	FI_2015	FI_2016_a3 (Best Match)	\sim													
	FI	FI_2016	FI_2017_a2 (Best Match)	\sim													
	FI	FI_2017	FI_2018_a1 (Best Match)	\sim													
	FI	FI_2018	FI_2018_a1 (Best Match)	\sim											_		
	FI	FI_2019	FI_2018_a1 (Best Match)	\sim			n	bo		וט <i>י</i>	itn	ut f	for	both t	he		
	FI	FI_2020	training_data	~	1			U u				мс і 					
	FI	FI_2020reform	training_data	~			Da	ase	line	e ar	าส เ	rete	orn	n			
Output p	aut _c:\eurom	od\training\exer	FI_2018_a1 (Best Match) FI_2017_a2 FI_2008_a3 FI_2010_a2 FI_2012_a3 FI_2014_a3 FI_2015_a3 FI_2015_a3 training_data FI_2020_hhot														



Producing output microdata (2)

our angen an or r	Status	Time	Show		Stop
raining_data (UK_2018);	finished	19:09:29 - 19:09:38 (00h:00m:08s)	Run Log	Error Log	Stop
raining_data (UK_2018reform);	finished	19:09:29 - 19:09:37 (00h:00m:07s)	Run Log	Error Log	Stop
produce output for both baseline and reform	the ^{:T_uk=off;V}	VLS_uk=off;			



Opening the Statistics Presenter

	Countri	es	Display	Country	/ Tools	Adminis	tration Tools	Add-Ons	Version Control	Applications	Help & Info
Open Output File	Po Eff	→● licy ects	Hypothetic Househol	al In-di d Anal	epth Sta lysis Pres	tistics senter				- · ·	
	Policy				Grp/No	FI_202	0 F	I_2020reform	Comment		
41	- •	bch_	fi	Г		on		חו	BEN: Child bene	fit	
41.1	-	fx D	efConst		Statistics	Presente	er - Select Ter	nplate			×
41.1.1			\$bch1								
41.1.2			\$bch2								
41.1.3			\$bch3		St	atis	tics P	resent	er		
41.1.4			\$bch4			acio					
41.1.5			\$bch5		Plea	ase sel	ect a Statist	ic Template			
41.1.6			\$bchsup								
41.2	F	fx E	lig		Summa	ary Stati	istics - Baseli	ne/Reform			~
41.3	-	fx B	enCalc		Summa	ry Stati	istics - Defau	lt			
41.3.1			who_must	_be_eli	Summa	iry Stati	istics - Baseli	ne/Reform			
41.3.2			comp_con	d	Summa	ry Stati	stics - MultiS	ystem			
41.3.3			#_N		Summa	ry Stati	istics - Variab	le			
41.3.4			#_M		1	1	1	L			



Statistics Presenter: Baseline/Reform (1)

Statistics Presenter - Select Files		
Summary Statis	tics - Baseline/Refo	rm
	ties - baseline/ Neio	
Select Files for Calculating Stat	istic	
Baseline Scenario	Altern	ative scenario(c)
r:\b2\04 - euromod jn\06 - working area\02 - devel	opers\klaus\other\em training\em	- euromod jrc\06 - working ar a\02 - developers\klaus\other\em training\em
fi_2020reform_std.txt	[0] h_	2020 cform_std.txt
n_2020_std.txt		U_sta.txt
		Han Ond an Shift to a start we do
		Use Ctri or Snift to select multipl The order of selection will be ret
	OK Car	ncel Is your file not vis



Statistics Presenter: Baseline/Reform (2)

	FI_2020 (base)	FI_2020reform	Difference to base
Decile 1	732.41	732.77	0.36
Decile 2	1,473.39	1,474.60	1.21
Decile 3	1,741.82	1,743.62	1.80
Decile 5	2,056.81	2,058.48	1.67
Decile 6	2,521.88	2,523.77	1.89
Decile 7	2,539.25	2,540.99	1.74
Decile 8	3,117.97	3,118.85	0.89
Decile 9	3,758.53	3,759.79	1.25
Decile 10	4,851.77	4,853.72	1.95
All	2,432.58	2,433.94	1.36
Poor	1,088.49	1,089.27	0.78

Table 2b: Mean household income by decile groups





• 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 option









	Policy		Grp/No	EE_2018	Comment
14	+ 0	bch00_ee		on	BEN: child allowance (lapsetoetus)
15	+ 0	bchlg_ee		n/a	BEN: allowance for families with 3+ children (kolme- ja enamalapselise pere toetus)
16	+ 0	bched_ee		n/a	BEN: school allowance (koolitoetus)
17	F 0	bchba_ee		on	BEN: childbirth allowance (sünnitoetus)
18	× 0	bcc00_ee		on	BEN: childcare allowance (lapsehooldustasu)
19	× •	bcclg_ee		on	BEN: parental allowance for families with 7+ children / many children (seitsme- ja enamalapselise pere vanema toetus / lasterikka pere toetus)
20	+ 0	bunct_ee		on	BEN: unemployment insurance benefit (töötuskindlustushüvitis) PART-SIMULATION
21	F 😑	bunnc_ee		on	BEN: unemployment assistance benefit (töötu abiraha)
22	+ 0	tscer_ee		on	SIC: employer social insurance contributions
23	F 🙆	tscct_ee		on	SIC: credited social insurance contributions
24	× 0	tscee_ee		on	SIC: employee social insurance contributions
25	+ 0	tintaag_ee		on	TAX: allowance for self-employment income from agriculture (applicable to income tax and self-employed SIC)
26	+ 0	tscse_ee		on	SIC: self-employed social insurance contributions
27	+ 0	tinwh_ee		on	TAX: withholding income tax (jooksvalt kinnipeetud tulumaks, current liability)
28	+ 0	tin_ee		on	TAX: income tax (deklaratsioonijärgne ehk lõplik maksukohustus, final liability)
29	+ 0	tinrf_ee		n/a	TAX: annual refund to low-paid employees (madala sissetulekuga töötava isiku iga-aastane tagasimakse)
30	+ 0	bsa00_ee		on	BEN: subsistence benefit (incl. housing component) (toimetulekutoetus)
31	× 0	bsach_ee		n/a	BEN: needs-based family benefit (vajaduspõhine lapsetoetus)
32	+ 0	bsape_ee		on	BEN: pensioner's living alone allowance (üksi elava pensionäri toetus)

THE POLICY SPINE



The policy spine:

- The policy spine:
 - The order of policies is called the 'spine'
 - We will take a quick tour through the policies in the Estonian model
 - > the models for all EU countries follow the same structure of the policy spine
 - See the EUROMOD Country Reports for more details



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 \rightarrow 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 individuallevel results



Definitions

	Policy		Grp/No	EE_2018	Comment
1	+ 😐	SetDefault_ee		on	DEF: DEFAULT VALUES FOR VARIABLES
2	• •	UAA_ee		switch	SWITCH: Uprating by Average Adjustment (ON=uprating by averages; OFF=uprating by statutory indexation)
3	۲ 🔘	Uprate_ee		on	DEF: UPRATING FACTORS
4	× 0	ConstDef_ee		on	DEF: CONSTANTS
5	۲ 🔘	ILSDef_ee		on	DEF: STANDARD INCOME CONCEPTS
6	+ •	ILSUDBDef_ee		on	DEF: STANDARD INCOME CONCEPTS (UDB related)
7	۲ 🔘	ILDef_ee		on	DEF: SPECIFIC INCOME CONCEPTS
8	۰ ۱	TUDef_ee		on	DEF: ASSESSMENT UNITS
9	+ •	allocate_ee		on	DEF: reallocate benefits within household
10	× •	BTA_ee		switch	SWITCH: Benefit Take-up Adjustments (ON=non take-up; OFF=full take up)
11	* •	FYA_ee		n/a	SWITCH: Full Year Adjustments, i.e. model annual policies instead of 30th June (switch on/off to apply)
12	۰ ۰	yem_ee		off	DEF: minimum wage
13	+ 0	neg_ee		on	DEF: recode negative incomes (e.g. self-employment income) to zero



Tax and benefit calculations

	Policy	Grp	/No EE_2018	Comment
14	۲ 🔘	bch00_ee	on	BEN: child allowance (lapsetoetus)
15	F 😖	bchlg_ee	n/a	BEN: allowance for families with 3+ children (kolme- ja enamalapselise pere toetus)
16	۰ (bched_ee	n/a	BEN: school allowance (koolitoetus)
17	F 🛛	bchba_ee	on	BEN: childbirth allowance (sünnitoetus)
18	۰ ه	bcc00_ee	on	BEN: childcare allowance (lapsehooldustasu)
19	× 0	bcclg_ee	on	BEN: parental allowance for families with 7+ children / many children (seitsme- ja enamalapselise pere vanema toetus / lasterikka pere toetus)
20	• •	bunct_ee	on	BEN: unemployment insurance benefit (töötuskindlustushüvitis) PART-SIMULATION
21	۰ ا	bunnc_ee	on	BEN: unemployment assistance benefit (töötu abiraha)
22	۰ (tscer_ee	on	SIC: employer social insurance contributions
23	F 😐	tscct_ee	on	SIC: credited social insurance contributions
24	۰ ه	tscee_ee	on	SIC: employee social insurance contributions
25	F 0	tintaag_ee	on	TAX: allowance for self-employment income from agriculture (applicable to income tax and self-employed SIC)
26	۰ ا	tscse_ee	on	SIC: self-employed social insurance contributions
27	• •	tinwh_ee	on	TAX: withholding income tax (jooksvalt kinnipeetud tulumaks, current liability)
28	• •	tin_ee	on	TAX: income tax (deklaratsioonijärgne ehk lõplik maksukohustus, final liability)
29	• •	tinrf_ee	n/a	TAX: annual refund to low-paid employees (madala sissetulekuga töötava isiku iga-aastane tagasimakse)
30	• •	bsa00_ee	on	BEN: subsistence benefit (incl. housing component) (toimetulekutoetus)
31) ()	bsach_ee	n/a	BEN: needs-based family benefit (vajaduspõhine lapsetoetus)
32	+ .	bsape_ee	on	BEN: pensioner's living alone allowance (üksi elava pensionäri toetus)



Output

	Policy	Grp/No	EE_2018	Comment	
33	ø output_std_ee		on	DEF: STANDARD OUTPUT INDIVIDUAL LEVEL	
34	> 💿 output_std_hh_	ee	off	DEF: STANDARD OUTPUT HOUSEHOLD LEVEL	



Session 3 Policy functions Elig & ArithOp. EM handling errors



In this session, you will learn about

- Tax units
- Implementing eligibility conditions with *Elig*.
- Benefit/tax amount calculations with *ArithOp*.
- EM parameters: amounts, formulas and queries.
- EM handling errors



Policy		System Name	Comment
Elig		on	Made-up example: condition to pay employee NI contributions
elig_cond		yem>0 & !IsCivilServant	with positive earnings (yem) and not a civil servant (!IsCivilServant)
	TAX_UNIT	tu_individual_uk	assessment unit used for the calculations
Ar	ithOp	on	Made-up example: pension contributions
	Who_Must_Be_Elig	one	calculations carried out if at least one member of assessment unit fulfills condition from last Elig function
	formula	yem*0.08	8% for Old-age Pension Fund
	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_uk	assessment unit used for the calculations

THE POLICY FUNCTIONS Elig & ArithOp



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 is the criteria, i.e. the **eligibility condition**
 - e.g. with earnings up to £800 per month and receives no pensions
- What is the **benefit/tax amount**:
 - > e.g. £20 per week 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.



Functions: ArithOp

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

Policy		System Name	Comment			
ArithOp		on	Made-up example: child benefit			
	formula	25#w*nDepChildrenInTu	£25 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_uk	assessment unit used for the calculations			

- Now let's have a look at some key parameter values we can use for formula
 - o amounts, formulas, queries



Parameters: Amounts

- (#m) Monetary (numbers; use . for decimal) followed by their period:
 - *#m* for monthly (no conversion)
 - #y for yearly
 - #q for quarterly
 - #w for weekly
 - #d for daily
 - #/ 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	25	£25 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_uk	assessment unit used for the calculations				



Parameters: Formula

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

Policy	System Name	Comment
ArithOp	on	Made-up example: child benefit
formula	(25#w - xed/2)*nDepChildrenInTu - bed s	£25 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_uk	assessment unit used for the calculations



Parameters: 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)



Parameters: Queries (2)

- No need to learn the query names by heart
- \rightarrow use "IntelliSense" feature to find and select the desired query

ArithOp	on	n Made-up example: child benefit			
formula	25#w <mark>`n</mark>	£25 per wee assessment	ek for each dependent child in the unit		
output_var Inddex output_var nDepParentsInTu TAX_UNIT nDepRelativesInTu nDepParentsAndRelat nChildrenOfCouple#x nDepChildrenOfCouple nPersInUnit#x nAdultsInTu#x nDepChildrenInTu#x		sInTu	in the variable bch_s ch: child, _s: simulated) unit used for the calculations		



Functions: *Elig* (1)

- 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



Functions: *Elig* (2)

Policy		System Name	Comment			
Elig		on	Made-up example: condition to pay employee NI contributions			
elig_	cond	!lsCivilServant	not a civil servant (!IsCivilServant)			
TAX UNIT		tu individual uk	assessment unit used for the calculations			
ArithOp	0	on	Made-up example: pension contributions			
form	nula	yem*0.08	8% of earnings (yem) for Old-age Pension Fund			
output_var TAX_UNIT		tsceepi_s	result saved in the variable tsceepi_s (t: tax, sc: social contributions, ee: employee, pi: pension insurance, _s: simulated)			
		tu_individual_uk	assessment unit used for the calculations			



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



Functions: *Elig* (3)

Policy	System Name	Comment
Elig	on	Made-up example: condition to pay employee NI contributions
elig_cond	!IsCivilServant	not a civil servant (!IsCivilServant)
TAX_UNIT	tu_individual_uk	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 fulfills condition from last Elig function
formula	yem*0.08	8% of earnings (yem) for Old-age Pension Fund
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 uk	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 fulfills condition from last Elig function
formula	yem*0.02	2% of earnings for Unemployment Fund
output_var	tsceeui_s	result saved in the variable tsceeui_s (t: tax, sc: social contributions, ee: employee, ui: unemployment insurance, _s: simulated)
TAX_UNIT	tu_individual_uk	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
 - o **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					Will	the	calcu	lation	s in
Elig	on	Made-up example: eligibility condition for social assistance					Arith who	Op b o_m i	e car u st_l	ried o be_e l	ut if: l ig=
elig_cond	dag>=80	individual should be of age (dag) of 80+ years		c				ult	_	lts I	-
TAX_UNIT	tu_individual_uk	assessment unit is the INDIVIDUAL	ے	erso	f	ν		ad		adu	ybod
ArithOp	on	Made-up example: social assistance amount	idh	idpi	dag	sel	one	one	all	all	nob
Who_Must_Be_Elig	?	who in the assessment unit must fulfil eligibility condition	1	11	80	1					
formula	100#m	benefit amount is £100	1	12	60	0	yes	yes	no	no	yes
		result saved in the variable	1	13	40	0					
output_var	bsa_s	assistance, _s: simulated)	2	21	80	1					
	tu bausabald uk	assessment unit is the	2	22	6	0	yes	yes	no	yes	yes
	tu_nousenoiu_uk	HOOSEHOLD	3	31	80	1	yes	yes	yes	yes	yes
			4	41	40	0					
			4	42	40	0	no	no	no	no	yes



Parameters: Eligibility

- In the current version of the model
 - conditions are surrounded by curly brackets { }
- In the next version of the model: <u>no more</u> { }
 the software no longer requires their use



NOW					
Policy System Name					
Elig	on				
elig_cond	dag>=80				
TAX_UNIT	tu_individual_uk				



Parameters: Output

(output_var; output_add_var)

• Either output_var or output_add_var must be indicated!

• Exception: func_Elig \rightarrow sel_s

Policy	System Name	Comment
ArithOp	on	Made-up example: child benefit
formula	25#w*nDepChildrenInTu	£25 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_uk	
Elig	on	Made-up example: eligibility condition
elig_cond	IsLoneParentOfDepChild	if a lone parent of a dependent child
TAX_UNIT	tu_bu_uk	
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_uk	


User Interface: Adding a function



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

Add Function Before	:) :	ArithOp
Add Function After	•	Elig
Delete Function(s)		BenCalc
Copy Function(s)		SchedCalc Min
Paste Function(s) Before		Max
Paste Function(s) After		Allocate
Copy Value(s)		on System Functions
Paste Value(s)		t Critica Special Functions
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 (1)

Policy	System Name	Cor
Elig	on	Ma
elig_cond	IsLoneParentOfDepChild	if a
TAX_UNIT	tu_bu_uk	
ArithOp	on	Ma sup
formula	15#w	£15
output_add_var	bch_s	ado fun
TAX_UNIT	tu_bu_uk	

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		- 1
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	n Ctrl+A	



User Interface: Adding a parameter (2)

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



onfiguration	Status	Time	Show		Stop
ining_data(UK_2018reform);	aborted	23:19:57 - 23:19:59 (00h:00m:02s)	Run Log	Error Log	Stop
ROR-LOG training_data (UK_2018r	eform); BTA_??=on;	SCT_uk=off;WLS_uk=off;			
ROR-LOG training_data (UK_2018n : 21.1.4 bch_uk/BenCalc/comp_perElig (3	eform): BTA_??=on; 31fac508-425a-4542-b94	:SCT_uk=off:WLS_uk=off; lc-58cdae9d69cf): Formula error: Expressio	n expected.		
ROR-LOG training_data (UK_2018r :21.1.4 bch_uk/BenCalc/comp_perElig (3	eform); BTA_??=on; 31fac508-425a-4542-b94	:SCT_uk=off;WLS_uk=off; lc-58cdae9d69cf): Formula error: Expressio	n expected.		

HANDLING ERRORS



Handling errors (1)

- 😐 bc	bch_fi		on	on	BEN: Child benefit		
- fx	DefConst		on		Constants for Child Care Benefits		
	\$bch1		94.88#m		typo in the syntax:		
	\$bch2		104.84#m 104.84#m		100# instead of 100#w		
	\$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		
⊢ fx	Elig		on	on	Child benefit: eligibility		
⊢ fx	BenCalc		on	on	Child benefit: amount		
⊢ fx	Elig		on	on	Eligibility for Lone parent supplement		
) fx	► fx ArithOp		on	on	Child benefit: single parent supplement		



Handling errors (2)

EUROMOD Run started 09/04/2021 19:51:39 and fi	inished 09/04/	/2021 19:51:41		_		
Configuration raining_data (FI_2020reform); BTA=on;MWA=off;	Status aborted	Time 19:51:39 - 19:51:41 (00h:00m:01s)	Run Log	Error Log	Stop	



Handling errors (3)

Configuration	Status	Time	Show		Stop	
training_data (FI_2020reform); BTA=on	MWA=off; aborted	19:51:39 - 19:51:41 (00h:00m:01s)	Run Log	Error Log	Stop	
ERROR-LOG training data (FI 20) error: 41.1.1 bch_fi/DefConst/\$bch1 3DD	20reform): BTA=on:MW/ 0683D-2806-4C60-BEC7-D5	A=off: 5F72C48C85) Formula error: Expression ex	pected.			^
on of the error: /, function, parameter	unique iden place of the	tifier for the exact error in the model				
						~



Handling errors (4)

= 	▶ Fi	inland	I - EUROMOD et	uromodfi	les_i3.0+_with_fi_c	original_xml (r:\b2\	04 - euromod jrc\08 - trainings\00 - ge	neral t — 🛛 🗆	×
	Countri	ies	Display Co	untry Too	ls dministratio	n Tools Add-On	s Version Control Applications	Help & Info	
	dd Sys Oelete S Clean Up	tem ystem o Syst	(s) ems	rt System rt System	(s) 4 (C) (s) 4 (C)	Admin Countr	y ♣ Add to, switch on ♣ Add to, switch off ■ Add to, switch off	nctions	
Cin	System Operations 10		or parameters by their						
38		y tin	mu_fi	01	on	ider	itifier		*
39	+ •	bho	ope_fi		on	on	BEN: Pensioner housing allowance		
40	، •	bho	000_fi		on	on	BEN: General housing allowance		
41	- •	bcł	n_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 I	BenCalc		on	on	Child benefit: amount		
41.4	×	fx I	Elig		on	on	Eligibility for Lone parent supplement		
41.5	×	fx /	ArithOp		on	on	Child benefit: single parent supplement		
42	+ •	bsa	a00_fi		on	on	BEN: Local authority income support		
43	+ •	out	tput_std_fi		on	on	DEF: STANDARD OUTPUT INDIVIDUAL LEVEL		
44	۰	out	tput_std		off	off	DEF: STANDARD OUTPUT HOUSEHOLD LEVEL		ļ



Handling errors (5)

copy unique identifier from error log and paste it here

	Countries I	Display C	ountry Tools	Administrat	ion Tools Add-Or	s Version Control Applications	Help & Info
() +	Add System	🔊 Imp	ort System(a 🔺 🔿	Admin Countr	.v 📥 Add to, switch on 💿	
🕸 🗙	Delete System(s))				Add to, switch off	
* 💌	Clean Up System	ns 📑 Exp	ort System(s) 💁 🗹	Set Swit	🗙 Remove from 🛛 🛔 🧧	
C	Sy	vstem Operatio	ons	To		Extensions	Compare Search, etc.
	Policy		Gr	FI_2020	FI_2020re	Comment	
33	🔸 🔍 tinto	ch_fi		n/a	n/a 🧡	TAX: Child tax credit	*
34	+ • tinda	fi 🗖		00		TAX: Income tax deficit	
51			Search by	Identifier		×	
35	🔸 🔍 tindo	csp_fi	Identifier	9DD0683D-280	6-4C60-BEC7-D55E72	ax special	
36	→ ● thr	fi			End	Canal ting tax	
50						ant (needed	
37	🔶 🛧 bed0)0_fi 🗖		on	on	for taxable income and	
						allowances calculation)	
20	•					TAX: Local income tax	
38	Tunn	u_11		on	on	(updated study grant is needed)	
20						BEN: Pensioner housing	
39	bhop	e_fi		on	on	allowance	
40	→ ● bho0)0 fi		on	on	BEN: General housing	
		<u> </u>				allowance	
41	• • bch_	_fi		on	on	BEN: Child Denefit	
41.1	⊤ fx De	efConst		on	on	Benefits	0
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	
41.1.6		\$bchsup		63.30#m	63.30#m	Child benefit supplement for a lone parent	
41.2	⊢ fx Elic	g		on	on	Child benefit: eligibility	0
41.3) fx Be	enCalc		on	on	Child benefit: amount	
41.4	6 E	-		00	00	Eligibility for Lone parent	-
+ 1	FI			011	bch fi - DefCons	t	Textsize:



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

20210409195141_9497b164c1e24b8da79525cfb8e5f2dc_EUROMOD_Log - Notepad				-		×	
File Edit Formet View Help							
===== EUROMOD LOG =====							~
Software-Version: 3.4.2							
<pre>Project: euromodfiles_i3.0+_with_fi_original_xml</pre>							
Start: 09 April 2021 19:51:39							
End: 09 April 2021 19:51:41							
Duration: 1.1765845s							
Output-Path: r:\b2\04 - euromod jrc\08 - trainings\00 - general training\02 - exercises\exer	cise 02\euromodfil	es_i3.0+	_with_fi_orig	inal_>	kml\ou	Jtp	
===== RUN-LOG =====				_	_		
Run-1d Status System Database Start End Duration Currency Exch	angerate BTA	MWA	Non-Default	Outpu	ut-Pat	:h i	
//989661-42+3-4db+-bb02-8a436e81960+ aborted F1_2020reform training_data 09 April 202	1 19:51:39 09 Apr	11 2021	19:51:41 1.1	/65845	5		
KUN-10: //969601-4213-4007-0002-604366619601							
ERROR: 41.1.1 DCn_t1/DetConst/\$DCn1 (9DD0665D-2606-4C00-DEC/-D55F/2C46C65): FOrmula error: E	xpression expected	•					L.
<						>	1
	Ln 1. Col 1	100%	Windows (CRLF)	UTF-8			
			(ener)	011 0		_	-







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 nonmeans-tested 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!







Steps:

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







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*.



Questions







- You learned how to use Elig and ArithOp to implement 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 a parameter



Session 4 Policy functions BenCalc & SchedCalc. Defining constants and income lists



In this session, you will learn about

- Combining the features of *Elig & ArithOp*, using policy function *BenCalc*
- Implementing a tax using policy function SchedCalc
- System functions to define
 - constants DefConst
 - income lists DefIL



Types of function

System functions

- functions used to define some general settings (e.g. to define income lists and tax units)
- we will learn about <u>DefIL</u>, <u>DefConst</u>, DefTU, DefVar, DefOutput, Uprate

Policy functions

- functions used to implement tax-benefit policies (e.g. define eligibility, calculate amount etc.)
- we already learned about *Elig* and *ArithOp*
- we will also learn about <u>BenCalc</u>, <u>SchedCalc</u>, 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



Function

olicy	Grp/No	System Name	Comment
enCalc		on	Made-up example: child benefit
Comp_Cond	1	nDepChildrenInTu>=1 & IsDepChild	if at least one 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_bu_uk	assessment unit used for the calculations

Policy function *BenCalc*



Function *BenCalc* (1)

- Very powerful function typically used to implement means-tested benefits
- Many benefits consist of several components, where a component is added if conditions are met by the assessment unit, and an income test, where certain types of income reduce benefit entitlement
- BenCalc allows these to be implemented/programmed easily
- It combines the functionalities of the functions Elig and ArithOp
- Compulsory parameters: Comp_Cond, Comp_PerTU/Comp_PerElig, output_var, TAX_UNIT



Function *BenCalc* (2)

- 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
- "groups" together Comp_Cond + Comp_perTU or Comp_perElig into one component
- can set upper and lower limits for each component: Comp_Lowlim, Comp_Uplim



Comp_perTU vs Comp_perElig

Policy		Grp/No	System Name	Comment	
BenCalc			on	Made-up example: child benefit	
	Comp_Cond 1 nDepChildrenInTu>=1 & IsDepCh		nDepChildrenInTu>=1 & IsDepChild	if at least one dependent child in the assessment unit	
	Comp_perTU	1	20#w	benefit amount is €20 per week	
	output_var		bch_s		
TAX_UNIT		tu_bu_uk		assessment unit used for the calculations	
Po	licy	Grp/No	System Name	Comment	
BenCalc			on	Made-up example: child benefit	
	Comp_Cond	1 nDepChildrenInTu>=1 & IsDepChild		if at least one dependent child in the assessment unit	
ſ	Comp perFlig	1	20 # w	benefit amount is €20 per week PER	





Function BenCalc (3)

 Ex. A lone parent family with two children, one with a disability, will receive €55#w

Po	licy	Grp/No	System Name	Comment Made-up example: child benefit	
Be	nCalc		on		
	Comp_Cond	1	nDepChildrenInTu>=1 & IsDepChild	if at least one 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	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_bu_uk	assessment unit used for the calculations	



Query IsNtoMchild#x

- IsNtoMchild#x is useful to implement child benefit which varies by n children
 - Returns 1 if a person belongs to the n to m oldest dependent children of the assessment unit, 0 otherwise.
 - n and m are defined by *footnote* parameters #_N and #_M.

	Policy BenCalc		Grp/No)	System Name	Comment
[]					on	Made-up example: child benefit
- footnote		Comp_Cond	1	I	sNtoMchild#1	if the dependent child is
parameters are		#_N	1	1	1	the first (oldest) child
referred to by #		#_M	1	1	1	
- they are	•	Comp_perElig	1		20#w	benefit amount is €20 per week for the first (oldest) child
grouped togothor with		Comp_Cond	2	I	sNtoMchild#2	if the dependent child is
main narameter		#_N	2		2	the second or any other child
they refer to		#_M	2	9	99	
		Comp_perElig	2	-	10#w	benefit amount is €10 per week for any other child
		output_var		k	bch_s	
		TAX_UNIT		t	tu_bu_uk	assessment unit used for the calculations



Function BenCalc (4)

- Withdraw parameters: subtract something from the calculated sum of components
 - Withdraw_Base: what is being subtracted (ex: variable, income list)
 - Withdraw_Rate: what percentage of the base is being subtracted
 - Withdraw_Start: level of the base where withdrawal starts
 - Withdraw_End: level of the base where sum of the componentsbase*rate is 0
- Negative result automatically set to 0
- Result=max(Sum of components-max(BASE-START, 0)*RATE, 0)
- _Rate and _End cannot be used simultaneously
 - If Withdraw_End is specified:
 - RATE=(sum of components)/(END-START)



Function BenCalc (5)

 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



Function BenCalc (6)

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	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_bu_uk	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 the UK

Perform a reform of the Child Benefit (CB) in the UK in 2018. Your task is to add a supplement to the CB for families with at least 2 dependent children below the age of 18, using the function BenCalc. The supplement is equal to £30 per week for the whole family but should be withdrawn with earnings: 10% should be withdrawn for each £1 that the family earns above £20,000 per year. Analyse the results after introducing the reform.









Steps:

- Create a new system in the UK based on the 2018 system and name it UK_2018ref2.
- 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.









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_bu_uk* (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. *bch_s*.
- The variable for earnings is called yem.



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

Level of Interpretation	used in condition parameters	used in other parameters
monetary variables and incomelists	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_uk	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_uk	assessment unit is the household



	Policy		Grp/No	UK_2020	Comment
27.22	- f x So	chedCalc		on	Tax schedule 2008-2009
27.22.1		Who_Must_Be_Elig		one	
27.22.2		base		il_Taxabley_NSND	Tax base
27.22.3		band_rate	1	0.2	First band rate
27.22.4		band_uplim	1	\$ITThresh2	Upper limit of first band
27.22.5		band_rate	2	0.4	Second band rate
27.22.6		band_uplim	2	\$ITThresh3	
27.22.7		band_rate	3	0.45	
27.22.8		Output_Var		i_tin_gross_rUK	(add)
27.22.9		TAX_UNIT		tu_individual_uk	

Policy function SchedCalc System functions DefConst, DefIL



Function SchedCalc (1)

- Used (mainly) in progressive taxes to define a tax schedule:
 - Tax bands: Band_UpLim / Band_LowLim
 - Tax rate: Band_Rate
 - Tax base: Base
- Group together Band_Rate and Band_UpLim/ Band_LowLim
- Instead of rates, for fixed amounts use Band_Amount

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		0.2	first band rate: 20%
Band_LowLim	1	12500#y	first band rate applies on income above £12.5k per year
Band_UpLim		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
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

Function SchedCalc (2)

- Quotient: split the base and apply the schedule separately
- Result = ((Base/Quotient)*Tax schedule)*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_uk	assessment unit used for the calculations



Function SchedCalc (3)

 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_ul	k assessment unit used for the calculations	


Function *DefConst* (1)

- Allows to define tax-benefit policy parameters as constants
 - e.g. monthly amounts, income thresholds, maximum amounts, tax rates
 - notated as x
- Advantages:
 - over-time changes to tax-benefit parameters are visible at one 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



Function DefConst (2)

	Policy	Gr	p/No	UK_2016	UK_2017	UK_2018	Comment	
3	🔻 🔵 Cons	tDef_uk		on	on	on	DEF: CONSTANTS	
3.1 - f x [efConst		on	on	on	Define constants	
3.1.1		\$SYSyear		2016	2017	2018		
3.1.2		\$MWRate		6.7	7.05	7.38	par60	
3.1.3		\$MWRate 1820		5.3	5.6	5.9	par61	
3.1.4		\$MWRate 1617		3.87	4.05	4.20	par62	
3.1.5		\$NLWRate		7.2	7.5	7.83	par	
3.1.6		\$NLWprem		0.5	0,45	0.45	par	
3.1.7		\$JSARate 1824		57.9#w	57.9#w	57.9#w	par63	
3.1.8		\$JSARate		73.1#w	73.1#w	73.1#w	par64	
3.1.9		\$JSAIncdisPen		50#w	50#w	50#w	par65	
3.1.10		\$JSAIncdisEarn		5#w	5#w	5#w	par66	
3.1.11		\$ITPerAll		11000#y	11500#y	11850#y		
		\$ITPerAll6574		n/a	n/a	n/a	20	
∣ user-d	efined	\$ITPerAll75		11000#y	n/a	n/a	changes to	
consta	ant	\$ITPerSa		1000#y	1000#y	1000#y	constant	
names	s \$ITx	\$ITPerSaHRP		500#y	500#y	500#y	values	
in "not		\$ITPerAllMCT		1100#y	1150#y	1190#y	kinklinktod	
	licy	\$ITyprAll		n/a	n/a	1000#y	nignlighted	
colum	n	\$ITLowAgeLim		65	65	65	par223	
3.1.19	j.	\$ITHigAgeLim		75	75	75	par224	
3.1.20		\$ITAllLim		100000#y	100000#y	100000#y	par82	
3.1.21		\$ITMca6574		n/a	n/a	n/a	par88	
3.1.22 3.1.23 3.1.24		\$ITMca75		8355#y	8445#y	8695#y	par83	
		\$ITAgeAllLim		n/a	n/a	n/a	par81 par87	
		\$ITMcaMin		3220#y	3260#y	3360#y		
3.1.25		\$ITThresh1		5000#y	5000#y	5000#y	par84	
3.1.26		\$ITThresh2		32000#y	33500#y	34500#y	par85	
3.1.27		\$ITThresh3		150000#y	150000#y	150000#y	par86	



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 DefIL allows to define income lists
 - standard, defined in the policy IIsDef_xx
 - > prefix ils_
 - used by the Statistics Presenter!
 - non-standard, defined in the policy IIDef_xx or in policies where they are used
 - > prefix il_



Function *Defll* (2)

Aggregates:

- variables
- pre-defined income lists
- fixed amounts
- constants
- Operations:
 - +
 - 0
 - shares (e.g. 0.8 means that the corresponding variable is multiplied by 0.8)

Use

- o once defined, available for all subsequent functions and policies
- behave like monetary variables
- continuously updated throughout the spine



Function *Defll* (3)





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
 - o ils_benmt: means-tested benefits
 - ils_ben: all benefits and public pensions
 - o ils_tax: taxes
 - ils_sicee: employee SICs
 - o ils_sicse: self-employed SICs
 - 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!



Standard income lists (2)





Income lists - matrix view

Summary of income list components

Countries	s Display	Country Tools A	dministration Tools	Add-Ons Version	Control App	lications Help & In	fo
Eull Spine	Condition	al Formatting	😌 Administrate	💡 Set Visible	•		Show Key Parameters
	📻 Automatic	Conditional Formatting	Add to	💿 Set Not Visible	e < Clear 📕 Set Bool		III Show Matrix View of Incomelists
Single Policy	Suspend System Formatting		🗙 Remove from	🚠 Expand	붬 Clear All		Show Hidden Systems Box
Policy View	Format		Gr	oups		Marking	Misc

	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						



Understanding constants and income lists

 We can now read and understand the code written in the SchedCalc function

	Policy	Policy		Grp/No	ES_2020	Comment
37.27	⊸ fx	Elig	9		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	Sd	hedCalc		on	National tax schedule on general income (cuota 1 general estatall)
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







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%.







Steps:

- Create a new system in Bulgaria based on the 2020 system and name it BG_2020reform.
- 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.







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.





You learned:

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



Questions





Session 5 Policy function Allocate. Defining tax units



In this session, you will learn about

- Allocating benefits and liabilities, using Allocate
- Parameters footnotes.
- System function to define tax units: *DefTU*



Types of functions

System functions

- functions used to define some general settings
- e.g. to define income lists and tax units
- we already learned about DefIL, DefConst
- we will learn today about DefTU

Policy functions

- functions used to implement tax-benefit policies
- e.g. define eligibility, calculate amount etc
- we already learned about Elig , ArithOp and BenCalc, SchedCalc
- we will learn today about 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

Function

Po	licy	System Name	Comment
Al	ocate	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_uk	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_uk	assessment unit used for the calculations



Function Allocate (3)

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

Po	licy	System Name	Comment
Al	ocate	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_lfzero	yes	if noone has earnings, then income tax is shared equally between the couple's partners
l	lgnore_neg_prop	yes	if someome'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_uk	assessment unit used for the calculations



Polic	cy	System Name	Comment
DefT	'u	on	Made-up example: individual
N	ame	tu_individual_uk	
Ту	уре	IND	
D	epChildCond	dag<16 (dag<19 & IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married
DefT	u	on	Made-up example: household
N	ame	tu_household_uk	
Ту	уре	НН	
D	epChildCond	dag<16 (dag<19 & IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married
DefT	'u	on	Made-up example: nuclear family
N	ame	tu_family_uk	
Ту	уре	SUBGROUP	
		Partner & OwnDepChild &	
N	1embers	LooseDepChild	head, head's partner, own and loose dependent children
D	epChildCond	dag<16 (dag<19 & IsInEducation & !IsMarried)	aged less than 16; or less than 19, in education and not married

ASSESSMENT (TAX) UNITS & THE SYSTEM FUNCTION DefTU



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_uk						
Туре	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_uk						
Туре	НН						
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_uk						
Туре	SUBGROUP						
Members	Partner & OwnDenChild	head head's partner own dependent children					
inclinets	dag<16 L (dag<19 &	aged less than 16: or less than 19, in education and not					
DepChildCond	IsInEducation & IsMarried)	married					



Assessment/tax units: examples

description	idhh	idperson	idpartner	idmother	idfather	dag	Household	Individual	Family
									(i.e. Subgroup)
typical family	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	2	201	202	0	0	56	A2	A2	A2
children	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	5	501	0	0	0	22	A5	A5	A5
living together	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, <u>optional</u> 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 !!sParent & 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

Ρο	licy	System Name	Comment					
De	fTu	on	Made-up example: couple with dependent parents					
	Name	tu_couple1_uk						
	Туре	SUBGROUP						
	Members	Partner & DepParent	head, head's partner and dependent parents					
		head:idmother=idperson	overwriting the default condition: only include parents of					
	DepParentCond	head:idfather=idperson	the head					
De	efTu	on	Made-up example: couple					
	Name	tu_couple2_uk						
	Туре	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"

Pc	olicy	System Name	Comment					
De	efTu	on	Made-up example: nuclear family					
	Name	tu_family1_uk						
	Туре	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	А
1	102	101	0	0	45	2100	yes	no	no	no	no	А
1	103	0	102	101	14	400	no	no	yes	yes	no	А
1	104	0	0	0	12	0	yes	no	yes	no	yes	В



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"

Po	licy	System Name	Comment					
DefTu		on	Made-up example: nuclear family					
	Name	tu_family2_uk						
	Туре	SUBGROUP						
		Partner & OwnDepChild &						
	Members	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	А
1	102	101	0	0	45	2100	yes	no	no	no	no	А
1	103	0	102	101	14	400	no	no	yes	yes	no	А
1	104	0	0	0	12	0	no	no	yes	no	yes	Α



Avoiding splitting up families

Query	Value type	Default	Description
AssignDepChOfDependents	yes/no	no	If set to yes dependent children of dependent unit members (i.e. persons who are not Head or Partner of the head) are assigned to the unit.
AssignPartnerOfDependents	yes/no	no	If set to yes partners of dependent unit members (i.e. persons who are not Head or Partner of the head) are assigned to the unit.



Using conditions which refer to income

- If the assessment unit is bigger than the individual, monetary variables or income lists are assessed at the level of the whole unit
- Use *footnote* parameters to specify an alternative assessment unit

Policy	Grp/No	System Name	Comment
DefTu		on	Made-up example: family
Name	(tu_family_uk	
Туре		SUBGROUP	
		Partner & OwnDepChild &	
Members		LooseDepChild	
			aged less than 25 and with (individual)
DepChildCond		dag<25 & yem <mark>#1</mark> < 1000#m	earnings of less than GBP 1,000 per month
		_	earnings are assessed at the individual
#_Level	1	tu_individual_uk	level

• Now let's have a look at footnote parameters



Parameters: Footnotes

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

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



The tax unit in the output file

Policy	Grp/ No	System Name
DefTu		on
Name		tu_family_uk
Туре		SUBGROUP
		Partner & OwnDepChild & LooseDepChild &
Members		DepParent
DepChildCond		dag<25 & yem#1<8000#y
DepParentCond		Default & dag>65 & yem#1<8000#y
#_Level	1	tu_individual_uk
AssignDepChOfDependents		yes
AssignPartnerChOfDependents		yes
LoneParentCond		Default & !IsMarried

idhh	idperson	idpartner	idmother	idfather	dag	ils_origy	yem	tu_family_uk_Headed	tu_family_uk_lsPartner	tu_family_uk_lsDepchild	tu_family_uk_lsDePparent	tu_family_uk_IsILoneParent
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
52	5,201	5,202	5,206	5,205	40	1,954	1,831	5,201	0	0	0	0
52	5,202	5,201	0	0	38	0	0	5,201	1	0	0	0
52	5,203	0	5,202	5,201	10	0	0	5,201	0	1	0	0
52	5,204	0	5,202	5,201	15	0	0	5,201	0	1	0	0
52	5,205	5,206	0	0	70	0	0	5,201	0	0	1	0
52	5,206	5,205	0	0	70	0	0	5,201	0	0	1	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






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.







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_2020refTU
 - 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.



Questions







- 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 6 System functions DefVar & DefOutput. Variable types. Uprating indices



In this session, you will learn about

- EM variables: types, naming conventions, variable library and output
- System functions to define
 - variables: Defvar
 - output: *DefOutput*
- Uprating indices and function Uprate
- EUROMOD documentation and help



xı 🔶 🥕 🚊									
Variables	Acronyms								
+ X	7	🚡 Select All Fil	ters	\checkmark	Monetary	\checkmark	Data	\checkmark	Inc
Add Variable Delete Variable	Apply Filters	🍾 Unselect All	Filters	\checkmark	Non-monetary	\checkmark	Simulated	\checkmark	HH
Edit									
Variables									
Name 🔺	Monetary	HH Level	Categorical		Automatic Label				
1118 tin_s	\checkmark				tax : income tax : s	imulated			
1119 tin00	\checkmark				tax : income tax : n	nain/basic			
1120 tin00_s	\checkmark				tax : income tax : n	nain/basic :	simulated		
1121 tin01_s	\checkmark				tax : income tax : 0	1:simulate	ed		
1122 tin02_s	\checkmark				tax : income tax : 0	2 : simulate	ed		
1123 tin03_s	\checkmark				tax : income tax : 0	3 : simulate	ed		
1124 tinag_s	\checkmark				tax : income tax : a	griculture :	simulated		
1125 tinav_s			tax : income tax : average rate : simulated						
1126 tinbd_s	\checkmark				tax : income tax : b	onds : simu	lated		

Variable types & system functions DefVar and DefOutput



Variable types

- Standard following EUROMOD naming conventions
 - Household and individual characteristics
 - Incomes: market incomes, simulated and nonsimulated benefits and taxes
 - Assets and expenditures

Intermediate

 Used to save the result of an intermediate calculation (e.g. an eligibility condition, number of twins in the tax unit, a subcomponent of a tax liability)



Variable naming conventions (1)

- Applicable to standard variables:
 - Included in the EUROMOD input microdata
 - 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-county comparisons



Variable naming conventions (2)

- Names are combination of acronyms: abb**[_s]
 - \circ **a** = type of information
 - » e.g. y=income, x=expenditure, d=demographic, l=labour
 - **bb** = specific for each type **a**, e.g.
 - y|em: employment income, y|se: self-employment income
 - ** = further bb's for additional information/detail, e.g.
 - y|em|xp: employment income, extra pay
 - _s for simulated variables
 - 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 file (VarConfig.xml)



Variables library (1)





Variables library (2)

← <i>≯</i> =				Adn	ninistration of Variables and Acronyms		- 0
■ Variables	Acronyms						
Add Variable	?	yem					
Delete Variable	Show variables	🔍 Sea	arch			acronyms use	d for names
Edit		Search	Va				
ariables						Acronyms	
Name	Monetary	HH Level	Categorical	Automatic Label	^	Description	Acronym
1518 yem				income : employment		▶ DEMOGRAPHIC	D
1519 yem_a				income : employment : add onn		LABOUR MARKET	L
1520 yem_s	\checkmark			income : employment : simulated		BENEFIT/PENSION	B/P
1521 yem00	\checkmark			income : employment : main/basic		- INCOME	Y
1522 yem01				income : employment : 01			DS
1523 yem01 s				income : employment : 01 : simulated		earnings	EA
1524 vem02				income ; employment ; 02		employment	EM
1525 vem02 s				income : employment : 02 : simulated			IY
1526 0013 6				income : employment : 03 : simulated	Ist of all variables def	rined in the	PP
				reme : employment : 04 : simulated	variables file in alpha	betical order	PR
name				income : employment : of : simulated	variables me maipha	Setted of def	PT
1E20 vomabet				income : employment : abread : patra		selfemployment	SE
1529 yemabitu				income : employment : abroad : hot taxable	variable is monetary or	non-monotary	
1530 yemabix				income : employment : abroad : taxable	variable is indiretary of	non-monetal y	IG
1551 yemaj				income : employment a difficult of		daily wage	DW
1532 yemajmy				income : employment : additonal jobs : me	per vear	▶ provider	
1533 yemcs				income : employment : company share	automatic label	▶ origin	
1534 yemdt				income : employment : date of interview		▶ employment	
1535 yemeq_s				income : employment : full-time equivalent	mployment : simulated	▶ invest income	
criptions						Categories	
Juntry Desci	iption				î	Value Description	
emplo	yment income			de	escription of variable		

for countries where it

is used



bg

су

cz

de

employment income

employment income

employment income

Income from employment (Prijmy ze zamestnani)

System function DefVar

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

	Policy	Grp/No BG_2020	Comment
27	🗸 🔵 bchbals_bg	on	BEN: birth grant (also for adoption) (еднократна помощ при раждане)
27.1	Fx DefConst	on	Parameters used in the policy - defined as constants
27.2	→ f _X DefVar	on	
27.2.1	i_nbaby	0	
27.3	▶ f _X BenCalc	on	number of babies (aged=0)
27.4	- fx Elig	OD	If the family has only one baby aged 0
27.4.1	elig_cond	i_nbaby=1	
27.4.2	TAX_UNIT	tu_bchmt00_bg	
27.5	Fx BenCalc	on	Benefit amount for families with only one baby aged 0
27.6	F _X Elig	on	If the family has twins, triplets etc.
27.7	► fx BenCalc	on	Benefit amount for families with twins, triplets etc.

 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
	yes	yes	yes	Use the variable directly
Standard variables	no	yes	yes	Use the variable directly
	no	no	yes	Create the variable in the variables list
	no	no	no	Create the acronyms and the variable in the variables list (rarely needed)
Intermediate	yes	n/a	n/a	Use the variable directly
variables	no	n/a	n/a	Create the variable with DefVar (not needed for sin??_s)



EUROMOD output microdata

- Content manipulated in policy output_std_cc
 - Controls level at which 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)
 - o (optional) non-standard income lists
 - o (optional) intermediate variables
 - o (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	-	out	put_std_ro		on	on		on			DEF: STANDARD OUTPUT INDIVIDUAL LEVEL
35.1	l r	fx 🛛	DefOutput		on	on		on			
35.1.1			file		RO_2018_std	RO_2019_std		RO_2020_std			
35.1.2			vargroup	Г	id*	id*		id*			ID variables
35.1.3			vargroup		d*	d*		d*			Demographic variables
35.1.4			vargroup		*	*		*			Labour market variables
35.1.5			vargroup		y*	names of	outpu	put files: _std			Market income variables
35.1.6	1		vargroup		p*						Public pensions variables
	1			<u> </u>	b*	XX_ye					Benefit variables
variable	es ir	ncl	uded in		tin*	tin*		tin*			Tax related variables
the out				tsc*	tsc*		tsc*			Social contribution related variables	
the out	pul	me	;		tpr*	tpr*		tpr*			
35.1.11			vargroup		a*	a*		a*			Asset variables
35.1.12			vargroup		x*	x*		x*			Expenditure variables
35.1.13	1		VarGroup		k*	k*		k*			
35.1.14	1		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_re	← -	-^ ^	
35.1.18			UnitInfo_TU	1	tu_bsa_ro	tu_bsa_ro		tu_bsa_ro		АЛ	
Tax assess	nen	t	UnitInfo_Id	1	HeadID	HeadID		HeadID	a	aaa	regation
			UnitInfo_TU	2	tu_bchmt_ro	tu_bchmt_ro		tu_bchmt_ro		-99	ganer
unit			UnitInfo_Id	2	HeadID	HeadID		HeadID			
identification	า	- I	UnitInfo_TU	3	tu_family_ro	tu_family_ro		tu_family_ro			
	•		UnitInfo_Id	3	HeadID	HeadID	HeadID		HeadID		
into			UnitInfo_TU	4	tu_bcc_ro	tu_bcc_ro		tu_bcc_ro			
35.1.25			UnitInfo_Id	4	HeadID	HeadID		HeadID			



	Policy		Grp/No	EE_2018	EE_2019	EE_2020	Comment	
1	+ •	Set	Default_ee		on	on	on	DEF: DEFAULT VALUES FOR VARIABLES
2	- •	Up	rate_ee		on	on	on	DEF: UPRATING 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	

Uprating indices & system function *Uprate*



Why uprate incomes?

- 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* the incomes in the input microdata to match the policy year
 - e.g. uprate 2017 input incomes to 2020 policy year
- We uprate incomes by source
 - define uprating indices in the Uprating Indices table
 - apply the uprating indices to the specific income variables from the input microdata, coded in the spine
 - they do not account for population 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*)



Raw indices (time-series)

Uprating Inc	dices								- 0	×
Raw Indice	Factors per Data and System									
	Index	Reference	2015	2016	2017	2018	2019	2020	Comment	^
1	Harmonised Index of Consumer Prices	\$HICP	100	100.8	104.48	1 .05	110.5	111.27	EUROSTAT; AMECO forecasts for 2020 values	
2	Consumer price index	\$f_cpi	1.6779	1.6795	1.7367	1)57	1.837	1.8774	Statistics Estonia (table IA001); 2019 MoF forecast	
3	CPI housing expenditure index	\$f_xhcot	2.5314	2.4884	2.5406	2 007	2.779	2.8402	Statistics Estonia (table IA001); 2019 MoF CPI forecast	
4	Nominal GDP, mln EUR	\$f_gdp	20782	21694	23-77		20 037	29467	Statistics Estonia (table RAA0012); 2019 MoF forecast	
5	Avg monthly salary, EUR	\$f_yem	1065	1146	p	olicy ye	ears 407	1484	Statistics Estonia (table PA5331); 2018 MoF forecast	
6	Avg more salary (lag 1), EUR	\$f_ye_g1	1005	1065	1146	1221	1310	1407	Statistics Estonia (table PA5331)	
7	Avg annu declared income from stocks, EUR	\$f_yiy	4257	3318	7325	2955	3174	3347	Tax reports (row 6.1); 2018-19 wage growth	
8	Avg annu declared income from other sources, EUR	\$f_yiy	1265	1359	1276	2711	2912	3071	Tax reports (row 7.1); 2018-19 wage growth	
46	Ave appled declared income from other assets,	\$f_yiy	16457	15541	17062	17465	18758	19784	Tax reports (row 6.3); 2018-19 wage growth	
	allinn, EUR per m2	\$f_xh	9.5	9.7	10.5	11	11.9	12	Statistics Estonia (table KV13, KV131); 2010 onwards real estate portal (kv.ee)	
de		\$f_xh	578	565	538	<mark>6</mark> 31	631	631	Tax reports (row 9.2); 2019-20 kept constant	
12	Total annual land tax revenues	efined i	ndex	58495	57708	57725	59081	59081	Statistics Estonia (table RR02); 2020 kept constant	
13	Avg monthly old age pension (e	¢f \	IIGOA	388.93	415.51	446.16	482.41	521	Statistics Estonia (table SK110); 2020 official indexation	
14		ק_וּ		222.81	241.09	268.62	298.52	322.4	Statistics Estonia (table SK110); 2020 official indexation	
15	Avg monthly survivors pension (end year), EUR	\$f_psu	179.88	190.36	200.06	215.87	234.08	252.81	Statistics Estonia able SK110); 2020 official indexatio	
16	Indexation of public pensions (cumulative)	\$f_ipens	2.5584	2.7042	2.8421	3.0581	3.315	3.5802	Statutory parame	
17	Birth allowance (main rate), EUR	\$f_bchba	320	320	320	320	320	320	Statutory parame	
18	Child allowance (1st child), EUR per month	\$f_bch00	45	50	50	55	60	60	Statutory parame	
19	Childcare allowance (child under 3), EUR per month	\$f_bcc00	38.4	38.4	38.4	38.4	38.4	38.4	Statutory parame	
20	Parental allowance for large families, EUR per month	\$f_bcclg	168.7	168.7	168.7	400	400	400	Statutory parame	
21	Lone parent benefit, EUR per month	\$f_bchlp	19.18	19.18	19.18	19.18	19.18	19.18		\sim
< ^^	I I I I I I I I I I I I I I I I I I I	A L	100.01	104 51	140.00	101.00	170 00	100.00	source of	
									information rear	~
Check Usa	ge								✓Save & Close XCancel	

European

Commission

Factors per data and system

Uprating Indices

Raw Indices Fa	ctors per Data and Sys	tem							
Dataset EE_20	18_c1 ~	Income Year 20	17 Update						
Index	EE_2013	EE_2014	EE_2015	EE_2016	EE_2017	EE_2018	EE_2019	EE_2020	^
\$HICP	0.952	0.9565	0.9571	0.9648	1	1.0342	1.0576	1.065	
\$f_cpi	0.972	0.971	0.9661	0.9671	1	1.034	1.0578	1.081	
\$f_xhcot	1.0259	1.0116	0.9964	0.9795	1	1.063	1.0938	1.1179	
\$f_gdp	0.8005	0.8488	0.8741	0.9124	1	1.0951	1.1792	1.2394	
\$f_yem	0.7772	0.8231	0.8722	0.9386	1	1.0729	1.1523	1.2154	
\$f_yemlag1	0.774	0.8281	0.877	0.9293	1	1.0654	1.1431	1.2277	
\$f_yiydv	0.3741	0.4887	0.5812	0.453	1	0.4034	0.4333	0.4569	
\$f_yiyit	0.8221	1.9404	0.9914	1.065	1	2.1246	2.2821	2.4067	
\$f_yiyot	0.9299	0.9869	0.9645	0.9109	1	1.0236	1.0994	1.1595	
\$f_xhcrt	0.8095	0.819	0.9048	0.9238	1	1.0476	1.1333	1.1429	
\$f_xhcmomi	1.1283	1.1357	1.0743	1.0502	1	1.1729	1.1729	1.1729	
\$f_tpr	0.9919	1.0216	1.0053	1.0136	1	1.0003	1.0238	1.0238	
\$f_poa00	0.7938	0.8381	0.8884	0.936	1	1.0738	1.161	1.2539	
\$f_pdi	0.7839	0.8284	0.8766	0.9242	1	1.1142	1.2382	1.3373	
\$f_psu	0.7987	0.8473	0.8991	0.9515	1	1.079	1.17	1.2637	
\$f_ipens	0.8004	0.8468	0.9002	0.9515	1	1.076	1.1664	1.2597	\checkmark

- For the selected dataset, the implicit uprating factors for each system are shown, e.g.:
 - Dataset EE_2018_c1 → income referring to 2017
 - Prices (\$f_cpi) increased by 8.1% between 2017 and 2020

! The policy year of each system is the one declared when creating it, not necessarily what you see in the name!



System function Uprate (1)

 Defines which indices to apply on which income variables from the input microdata

	Policy				Grp/No	EE_2018	EE_2019		EE_2020	Comment	
	2	-	Uprat	te_ee		on	on		on	DEF: UPRATING FACTORS	
	2.1	-	fx Up	rate		on	on		on	apply uprating factors	
	2.1.1			Dataset		EE_20??_??	EE_20??_	<u>??</u>	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		input data sets to which		
	2.1.6	_		afcsh		\$f_cpi	\$f_cpi		pat aata a		
variahl	20			bedet		\$f_cpi	\$f_cpi	up	prating ap	opnes	
Variasi	that are		bedot bsa00			\$f cpi	\$f_cpi		\$f_cpi		
that ar						\$f_cpi		\$f_cpi			
uprate	d			bsach		0	0		0		
	2,1,11		bsals		\$fi		\$f_cpi		\$f_cpi		
	2.1.12			bsape		\$f_sape	\$f_bsape	:	\$f_bsape		
	2.1.13			kivho	\$fi		\$f_cpi	\$f_cpi			
	2.1.14			yds	namo	of uprati	ina		1	used for validation only	
	2.1.15			ydses_o		or uprati	ing		1	used for validation only	
	2.1.16			yprro	factor				\$f_cpi		
	2.1.17			yprho		\$t_cpi	\$f_cpi		\$f_cpi		
	2.1.18			xed		\$f_cpi	\$f_cpi		\$f_cpi		
	2.1.19			xcd		\$f_cpi	\$f_cpi		\$f_cpi		
	2.1.20			ypp02		\$f_cpi	\$f_cpi		\$f_cpi		
	2.1.21	ypp03		\$f_cpi	\$f_cpi		\$f_cpi				
	2.1.22			bchba		\$f_bchba	\$f_bchba		\$f_bchba		
	2.1.23			bch00		\$f_bch00	\$f_bch00)	\$f_bch00		



System function Uprate (2)

- Uprating the components of an aggregate variable
- Using different uprating indices for different groups

🛛 🌒 Upra	te_bg		on	
⊸ fx Up	orate		on	
	dataset		BG_20??_??	
	yemtx		\$f_yem	
	yemnr		\$f_yem	
	AggVar_Name	1	yem	
	AggVar_Part	1	yemtx	
	AggVar_Part	1	yemnr	
	AggVar_Tolerance	1	1	
	ysetx		\$f_yem	
	ysenr		\$f_yem	
	AggVar_Name	2	yse	
	AggVar_Part	2	ysetx	
	AggVar_Part	2	ysenr	
	AggVar_Tolerance	2	1	

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







Modifying uprating factors in Romania to account for differential wage evolution

The average monthly salary in Romania in 2017 was 3,223 RON. Imagine that you have the following (hypothetical) information about the evolution of salaries between 2017 and 2020:

	Increase in relation to previous year						
Monthly salary in 2017	2018	2019	2020				
Up to average	10%	30%	13%				
Above average	20%	38%	18%				

- Do the necessary changes in EUROMOD so that it reflects this different evolution of salaries, when running the 2020 system.
- Use the Statistics Presenter to compare the new system with the original 2020 system [only with SILC-based input data]







Steps:

- Create two new time-series in the uprating indices table (tab Raw indices) to account for the different changes in salaries between 2017 and 2020.
- Check that they correctly reflect the information in the table (tab Factors per data and system)
- Create a copy of the RO 2020 system (e.g. RO_2020_uprating).
- Modify the uprate_ro policy to account for the new information.
- Run RO_2020 and RO_2020_uprating in the Statics Presenter and analyse the fiscal and distributional impact [only with SILC-based input data]







Hints:

- Before modifying any parameter in EUROMOD, compute in Excel index numbers, considering 2017 as the base (=100)
- In Uprating indices → Raw indices add two rows for two new indices, \$f_yemlow and \$f_yemhigh, and paste the results from Excel. Bear in mind that for this exercise we don't care about what happened before 2017.
- In the new system RO_2020_uprating go to the uprate_ro policy, search for the Uprate function corresponding to 2018 data and tell EUROMOD how to uprate yem:
 - Using the new indices: you have to add two placeholders and two Factor_condition parameters.
 - Setting to n/a the original uprating factor used for yem.



Questions







• You learned how to:

- include new uprating factors
- use them in the EUROMOD spine



2017-2020 Tax-benefit systems

2017-2020 Tax-benefit systems

These reports accompany the forthcoming release of EUROMOD I3.0+. There may be minor differences between the results presented here and those obtained with I3.0+ due to further improvements since the report was prepared.

These reports have been prepared by the National Teams in collaboration with the University of Essex: <u>https://www.microsimulation.ac.uk/euromod/resources/documentation/country-reports/</u>

Belgium	Download 난
Bulgaria	Download 난
Czechia	Download 난

Documentation



Documentation (1)

- Built-in help
 - EUROMOD terminology
 - Running EUROMOD and basic concepts
 - EUROMOD functions
- Data Requirement Document (DRD) 1 per dataset
- Documentation folder:
 - EUROMOD built-in help saved in pdf
 - HHoT manual
 - Add-ons' documentation
 - What's new document
- Log folder:
 - EUROMOD version log



Documentation (2)

Country report

- Main document accompanying the model
- Provides information about:
 - tax-benefit system
 - > model
 - underlying data
 - accuracy of simulations
- Each CR covers the policy systems
 - from the income year of the latest available input data (2017 for I3.0+)
 - > to the most recent policy year (2020 for I3.0+)



Documentation (3)

- Online documentation:
 - EUROMOD website: <u>https://euromod-web.jrc.ec.europa.eu/</u>
 - Country Reports (CR): <u>https://euromod-</u> web.jrc.ec.europa.eu/using-euromod/country-reports
 - EUROMOD Modelling Conventions (EMC): <u>https://www.microsimulation.ac.uk/publications/euromod-modelling-conventions/</u>
 - EUROMOD Working Paper series: <u>https://www.microsimulation.ac.uk/research-and-policy-analysis/publications/euromod-working-paper-series/</u>



Keep in touch



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



https://euromod-web.jrc.ec.europa.eu/newsand-events/newsletters



JRC-EUROMOD@ec.europa.eu



Thank you



© European Union 2021

Unless otherwise noted the reuse of this presentation is authorised under the <u>CC BY 4.0</u> 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.

