Cheatography

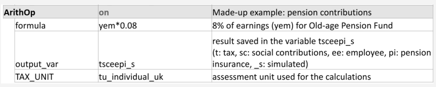
Personal Cheat Sheet

by bobbey via cheatography.com/197967/cs/41831/

Policy Functions

Elig	Function to determine eligibility for receiving benefits
ArithOp	Simple calculator for common arithmetic operations
BenCalc	Used for modelling policy instruments with various components
DefConst	Define constants
SchedCalc	Allows for the implementation of the most common (tax) schedules

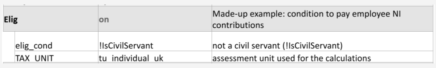
ArithOp and Interaction



ArithOp	on	Made-up example: pension contributions
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 allows for simple calculations. It requires three compulsory parameters: **formula** contains the calculations to perform; **output_var** indicates the variable where results are stored; and **TAX_UNIT** indicates the assessment unit for the calculations performed.

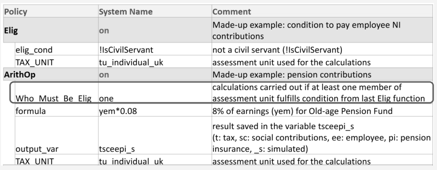
Elig



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

Elig determines the eligibility for of a unit according to a specific condition. It requires two parameters: **elig_cond** that determines the condition, and **TAX_UNIT** that determines the assessment unit.

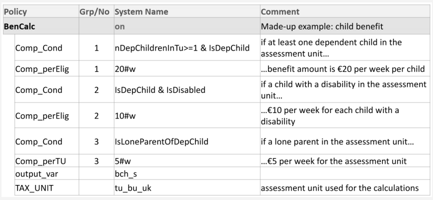
Elig + ArithOp



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

Elig and **ArithOp** can be combined via the parameter **Who_Must_Be_Elig**. Depending on the selection, this parameter ensures that calculations of the **ArithOp** are carried out if, one, one_adult, all, all_adults or nobody in the assessment unit fill the conditions defined defined in **Elig**.

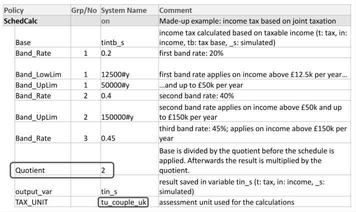
BenCalc



Policy	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…
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

BenCalc is a powerful function typically used to implement means-tested benefits that consist of several components. It combines the functionalities of **Elig** and **ArithOp** and has 4 compulsory parameters: **Comp_cond** defines a condtion; **Comp_perXX** calculates a formula an applies it the assessment unit; **Grp/No** groups together condition and formula; **outputvar** and **TAX_UNIT**.

SchedCalc



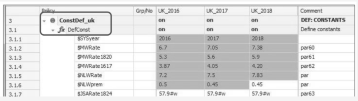
Policy	Grp/No	System Name	Comment
SchedCalc		on	Made-up example: income tax based on joint taxation
Base		tintb_s	income tax calculated based on taxable income (t: tax, in: income, tb: tax base, _s: simulated)
Band_Rate	1	0.2	first band rate: 20%
Band_LowLim	1	12500#y	first band rate applies on income 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

SchedCalc is used in progressive taxes to define a tax schedule. It has various parameters that can be used, such as: tax bands (**Band_UpLim / Band_LowLim**), tax rates (**Band_Rate**) and tax base (**Base**). These parameters are grouped together via **Grp/No**. To split the base and apply the schedule separately the parameter **Quotient** can be used.

System functions

Uprate	Uprating of monetary dataset variables
SetDefault	Definition of income lists
DefTU	Definition of assessment units
DefOutput	Definition of model output
DefVar & DefConst	Definition of intermediate variables and constants

DefConst



	Policy	Grp/No	UK_2016	UK_2017	UK_2018	Comment
3	ConstDef_uk		on	on	on	DEF: CONSTANTS
3.1	DefConst		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	$MWRate1820		5.3	5.6	5.9	par61
3.1.4	$MWRate1617		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	$JSARate1824		57.9#w	57.9#w	57.9#w	par63

DefConst allows to define tax-benefit policy parameters as constants (e.g. monthly amounts, income thresholds, maximum amounts, etc.). For notation purposes, constants are notated with a $ prefix.

Special Functions	
Loop & UnitLoop	Repeat part (or all) tax-benefit calculations
Store & Restore	Store and restore variables to their previous values
Change-Param	Change value parameters
Totals	Calculate aggregates of variables or incomelists over the whole population or a subset
DropUnit & KeepUnit	Drop (or keep) individuals, families or households with special characteristics
ILVarOp	Operations on the variables of an incomelist
RandSeed	Generate a series of pseudo-random numbers
CallPr-ogramme	Call external application
DefInput	Read values for one or more variables from a text file