Contents

Exercise 1: Add a new system ................................................................. 3

Solution ................................................................................................. 3

Exercise 2: Increase the VAT rate ........................................................ 4

Solution ................................................................................................. 4

Exercise 3: Change the Turnover Tax policy in 2016 ............................... 6

Solution ................................................................................................. 6

Exercise 4: Change the VAT rules by making certain additional items VAT-exempt ......................................................... 8

Solution ................................................................................................. 8

Exercise 5: Change the Personal Income Tax policy by increasing the tax rates for all but the lowest band. 10

Solution ................................................................................................. 10

Exercise 6: Introduce a child benefit .......................................................... 12

Solution ................................................................................................. 12

Exercise 7: Introduce an old aged benefit for older people 65+ .................. 18

Solution ................................................................................................. 18

Exercise 8: Introduce a benefit for unemployed young people ................... 22

Solution ................................................................................................. 22

Exercise 9: Introduce three benefit reforms and change the income tax policy to cover the costs ......................... 27

Solution ................................................................................................. 27

Exercise 10: Amending SCT for 2017 ........................................................ 28

Solution ................................................................................................. 28
Note: The exercises and solutions presented below are applicable to MicroZAMOD model v2.0.

Part A: Preparing to implement a policy reform in MICROZAMOD

Exercise 1: Add a new system
Difficulty level: Beginner

Activity:
Add a new system to MicroZAMOD.

Solution

Step 1: Before amending existing or adding new policies you need to add a new system; then do the amendments and additions in the new system.

• How to add a new system?

Right click on the system ‘ZM_2017’ → copy/paste system → name it ‘ZM_2017_reform’
Part B: Changing Existing Policies in MICROZAMOD

Exercise 2: Increase the VAT rate
Difficulty level: Beginner

Activity:
Increase the VAT rate from 16% to 18%.

Question:
How much extra VAT would this raise?

Solution

Step 1: Before amending existing or adding new policies you need to add a new system (See Part 1); then do the amendments in the new system.

Step 2: Amend the value of the parameter $VAT_Rate in the Constants

- How to update the value of the parameter $VAT_RATE?

As the VAT rate is included in the policy tva_ZM as a constant, the value of the constant needs to be amended in the constdef_ZM policy. Go to the policy constdef_ZM and change the value of the parameter $VAT_RATE from 0.16 to 0.18 (only for the ZM_2017_reform system). You do not need to make any amendments to the policy tva_zm.
Step 4: Run the ZM_2017 and ZM_2017_Reform systems in Statistics Presenter to identify how much extra VAT the reform would raise.

Answer: 228.53 million Kwacha
Exercise 3: Change the Turnover Tax policy in 2016  
Difficulty level: Intermediate

Activity:
Amend the Turnover Tax policy in 2016. Increase the upper limit for Turnover tax from 800,000 Kwacha to 1,000,000 Kwacha and increase the turnover tax rate from 3% to 5%.

Questions:
How much will the government gain or lose in direct taxes? What would the impact on poverty and inequality be?

Solution

Step 1: Before amending existing or adding new policies you need to add a new system (See Part 1); then do the amendments in the new system.

Step 2: Amend the value of the parameters $tt\_upper\_limit$ & $tt\_rate$ for the 2016 reform system only

- How to change the value of the parameter $tt\_upper\_limit$ and $tt\_rate$?

As the turnover tax upper limit amount is included in the policy $tt\_ZM$ as a constant, the value of the constant needs to be amended in the $constdef\_ZM$ policy. Go to the policy $constdef\_ZM$ and amend the values for the parameter called $tt\_upper\_limit$. Change the value of the parameter $tt\_upper\_limit$ from 800,000 per year (which is shown as #y) to 1,000,000 per year (only for the ZM_2016_reform system). Change the value of the parameter $tt\_rate$ from 0.03 to 0.05 (only for the ZM_2016_reform system).
Step 3: Run the ZM_2016 and ZM_2016_Reform systems in Statistics Presenter to identify how much revenue has been lost/gained in direct taxes as a result of this reform.

Note: The turnover tax threshold is also taken into account in the rules for personal income tax – see the second BenCalc function in the policy tin_ZM. As the reform involves raising the threshold for turnover tax, this means that some people will move from paying personal income tax to paying turnover tax.

Answer: Gained 563.57 million Kwacha in direct tax revenue

Answer: In terms of consumption-based poverty and inequality, this reform leads to an increase in the poverty rate of 0.4 percentage points and an increase in the Gini coefficient of 0.0010.
Exercise 4: Change the VAT rules by making certain additional items VAT-exempt

Difficulty level: Beginner

Activity:

A number of products which had been included in the normal rate of VAT are to become exempted. These products are all bread and cereals (coicop code: x011197), vegetables (coicop code: x011797) and meat (coicop code: x011297).

Question:

How much will the government lose in indirect taxes?

Solution

Step 1: Before amending existing or adding new policies you need to add a new system (See Part 1); then do the amendments in the new system. Call this new system: ZM_2017_reform_3.

Step 2: The standard rated items for VAT are in the policy ildef_ZM.

- How to change standard rated items to VAT-exempt status?

Go to the policy ildef_ZM → Standard rated items for VAT purposes (used in VAT policy) → replace ‘+’ with ‘n/a’ for the products which are now exempted (only ZM_2017_reform_3 system).
Step 3: Run the ZM_2017 and ZM_2017_reform_3 systems in Statistics Presenter to identify how much revenue has been lost in indirect taxes as a result of this reform.

Answer: A reduction in indirect tax revenue of 109.36 million Kwacha
Exercise 5: Change the Personal Income Tax policy by increasing the tax rates for all but the lowest band

Difficulty level: Beginner

Activity:

Amend the Personal Income Tax policy. The current tax band rates are 0%, 25%, 30%, and 37.5%. Increase the personal income tax rate for every tax band (excluding the first band) by one percentage point.

Questions:

How much will the government gain in tax revenue? What would the impact on poverty and inequality be?

Solution

Step 1: Before amending existing or adding new policies you need to add a new system (See Part 1); then do the amendments in the new system. Call the new system: ZM_2017_Reform_4.

Step 2: Amend the tax rates in the policy tin_ZM, within the function ‘SchedCalc’

- How to change the tax rates?

In this example, the lowest tax band (band 1) is unaffected. For all other tax bands, increase the tax band by one percentage point by amending the parameter Band_Rate.
Step 3: Run the ZM_2017 and ZM_2017_4 reform systems in Statistics Presenter to identify how much revenue has been gained in direct taxes as a result of this reform.

Answer: Additional direct tax revenue of 193.13 million Kwacha.

This would have no impact on poverty rates. It would lead to a very small reduction (0.0005) in the Gini coefficient.

Useful Tip: If one is making a number of policy amendments or policy reforms, one can elect to ‘hide’ one or more systems for simplicity. To do this, simply right click on the system you want to ‘hide’, go to ‘Move to Hidden Systems Box’ and select the systems you wish to hide. You can also ‘unhide’ systems by following the same steps and simply go to ‘Unhide All Systems’
Part C: Introducing New Policies in MicroZAMOD

Exercise 6: Introduce a child benefit
Difficulty level: Intermediate

Activity:
Introduce a universal child benefit of 180 Kwacha per month for children who are aged 0-2 inclusive.

Question:
How much will this policy cost the government? What would the impact on poverty and inequality be?

Solution

Step 1: Before adding a new policy you need to add a new system (See Part 1); then make the changes in the new system. Call this new system: ZM_2017_Reform_5.

Step 2: The new policy will be called bch_zm

- How to add this new policy?

For this example, the new benefit can be added to the end of the list of benefits and – importantly – it does not affect any of the other existing benefits. So, simply right click on the last ‘benefits’ policy \( \rightarrow \) add policy after \( \rightarrow \) benefit \( \rightarrow \) name the policy ‘bch_zm’ \( \rightarrow \) ok \( \rightarrow \) switch on the policy only for the system ZM_2017_reform_5.
Step 3: Add a BenCalc function to the policy bch_zm

- How to add the BenCalc function?

Right click on the policy bch_zm → add function → BenCalc → turn on the function only for the system ZM_2017_reform_5. This process adds all of the compulsory parameters for a BenCalc function.

Step 4: Add the eligibility conditions to the ‘Comp_Cond’ parameter: children aged 0-2 inclusive are eligible.
Step 5: specify the benefit amount

i. go to the policy constdef_ZM and create a new parameter $child_benefit_amount. Add the amount of the benefit (180 Kwacha) to this parameter.

right click on the last parameter → show add parameter form → placeholder → add
→ replace ‘placeholder’ with ‘$child_benefit_amount’ → add 180#m (only ZM_2017_reform_5 system).
ii. go back to \texttt{bch\_zm} and modify \textquote{Comp\_perTU} by adding the constant - \texttt{$child\_benefit\_amount$}.

iii. Modify the Output\_Var and TAX\_UNIT as follows for the ZM\_2017\_reform\_5.

iv. Whenever you add a new benefit policy on the spine, you also need to add it in the policy ildef\_ZM in two income lists [Simulated benefits & the relevant income list needed in order for the statistics presenter to work, i.e. Child benefits (\textquote{Child benefits} in Statistics Presenter)] and put ‘+’ only for ZM\_2017\_reform\_5.
<table>
<thead>
<tr>
<th>Policy</th>
<th>ZM_2017</th>
<th>ZM_2017_Reform_1</th>
<th>ZM_2017_Reform_2</th>
<th>ZM_2017_Reform_3</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Taxable income (used in income tax policy where there is turnover tax)</td>
</tr>
<tr>
<td>2.3</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Original income</td>
</tr>
<tr>
<td>2.4</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Simulated Benefits</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Name</td>
<td>lls_bensim</td>
<td>lls_bensim</td>
<td>lls_bensim</td>
<td>-</td>
</tr>
<tr>
<td>2.4.2</td>
<td>lls_bensim</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>Social cash transfer</td>
</tr>
<tr>
<td>2.4.3</td>
<td>bedot_s</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>Home grown</td>
</tr>
<tr>
<td>2.4.4</td>
<td>bot_s</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>Farmer Input support pack</td>
</tr>
<tr>
<td>2.5</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Benefits</td>
</tr>
<tr>
<td>2.6</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Employer social insurance contributions (needed for Summary Statistics Plug In)</td>
</tr>
<tr>
<td>2.7</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Employer social insurance contributions (needed for Summary Statistics Plug In)</td>
</tr>
<tr>
<td>2.8</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Self-employed social insurance contributions (not relevant but needed for Summary Stats Plug In)</td>
</tr>
<tr>
<td>2.9</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Define income list - Simulated Taxes</td>
</tr>
<tr>
<td>2.10</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Taxes (&quot;Direct taxes&quot; in Statistics Presenter)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Policy</th>
<th>ZM_2017</th>
<th>ZM_2017_Reform_4</th>
<th>ZM_2017_Reform_5</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>uprate_zm</td>
<td>-</td>
<td>-</td>
<td>DEF: UPGRATING FACTORS</td>
</tr>
<tr>
<td>2</td>
<td>ildef_cm</td>
<td>-</td>
<td>-</td>
<td>DEF: INCOME CONCEPTS</td>
</tr>
<tr>
<td>2.1</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>Define income list - Taxable income (used in income tax policy where there is turnover tax)</td>
</tr>
<tr>
<td>2.2</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>Define income list - Taxable income (used in income tax policy where there is turnover tax)</td>
</tr>
<tr>
<td>2.3</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>Define income list - Original income</td>
</tr>
<tr>
<td>2.4</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>Define income list - Simulated Benefits</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Name</td>
<td>lls_bensim</td>
<td>lls_bensim</td>
<td>lls_bensim</td>
</tr>
<tr>
<td>2.4.2</td>
<td>lls_bensim</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>2.4.3</td>
<td>bedot_s</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>2.4.4</td>
<td>bot_s</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>2.4.5</td>
<td>bch_s</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>2.5</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.6</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.7</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.8</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.9</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.10</td>
<td>DefII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
**Answers:** This reform would cost the government 2,423.72 million Kwacha in 2017. In terms of consumption-based poverty, it would lead to a reduction in the poverty rate of 3.9 percentage points. In terms of consumption-based inequality, it would lead to a reduction in the Gini coefficient of -0.0194.
Exercise 7: Introduce an old aged benefit for older people 65+

Difficulty level: Intermediate

Activity:

Introduce an old age benefit for people who are aged 65 and over. Decide on the monetary amount and the periodicity. For this example, eligible individuals are awarded 150 Kwacha per month.

Question:

How much will this policy cost the government? What impact would this have on poverty and inequality?

Solution

Step 1: Before adding a new policy, you need to add a new system (See Part 1); then make the changes in the new system. Call this new system: ZM_2017_Reform_6.

Step 2: The new policy will be called boa_zm

- How to add this new policy?

For this example, the new benefit can be added to the end of the list of benefits. So, right click on the last ‘benefits’ policy → add policy after → benefit → name the policy ‘boa_zm’ → ok → switch on the policy only for the system ZM_2017_reform_6

Step 3: Add a BenCalc function to the policy boa_zm

- How to add the BenCalc function?
Right click on the policy boa_zm ➔ add function ➔ BenCalc ➔ turn on the function only for the system 2M_2017_reform_6. This process adds all of the compulsory parameters for a BenCalc function.

Step 4: Add the eligibility conditions to the parameter ‘Comp_Cond’

i. Go to the policy constdef_ZM and create a new parameter $old_age_amount.

  right click on the last parameter ➔ show add parameter form ➔ placeholder ➔ add
replace ‘placeholder’ with ‘$old_age_amount’ \rightarrow add 150#m (only ZM_2017_reform_6 system).

ii. Go back to bch_zm and modify ‘Comp_perTU’.

iii. Modify the Output_Var and TAX_UNIT as follows in the ZM_2017_reform_6 system.

iv. Whenever you add a new benefit policy on the spine, you also need to add it in the policy ildef_ZM in two income lists [Simulated benefits & the relevant income list needed in order for the statistics presenter to work, i.e Pension benefits (‘Pension benefits’ in Statistics Presenter) and put ‘+’ only for ZM_2017_reform_6.
Step 4: Run the ZM_2017 and ZM_2017_6 reform systems in Statistics Presenter to identify how much government expenditure will be on this reform. What is the resulting impact on poverty and inequality of this reform?

Answer: This reform would cost the government 799.52 million Kwacha in 2017. It would lead to a 1.1 percentage point reduction in the overall poverty rate (consumption-based), but a 9.0 percentage point reduction in the poverty rate for households containing older persons. It would also result in a reduction of the Gini coefficient of -0.0055.
Exercise 8: Introduce a benefit for unemployed young people

Difficulty level: Intermediate

Activity:

Introduce a benefit for unemployed youth. Decide on the age band, monetary amount and periodicity. For this example, the age band 18 to 35 inclusive was used. All eligible individuals receive 100 Kwacha per month.

Question:

How much will this policy cost the government? What is the impact on poverty and inequality?

Solution

Step 1: Before adding a new policy, you need to add a new system (See Part 1); then make the changes in the new system. Call the new system: ZM_2017_Reform_7.

Step 2: The new policy will be called bun_zm. (Note: B: Benefit; UN: Unemployment) As per EUROMOD naming conventions:

- How to add this new policy?

For this example, the new benefit can be added to the end of the list of benefits. So, right click on the last ‘benefits’ policy → add policy after → benefit → name the policy ‘bun_zm’ → ok → switch on the policy only for the system ZM_2017_reform_7.
Step 3: Add a BenCalc function to the policy bun_zm

- How to add the BenCalc function?

  Right click on the policy bun_zm → add function → BenCalc → turn on the function only for the system ZM_2017_reform_7. This process adds all of the compulsory parameters for a BenCalc function.
Step 4: Add the eligibility conditions to the parameter ‘Comp_Cond’

i. go to the policy constdef_ZM and create a new parameter $unemployed_benefit_amount
   
   right click on the last parameter → show add parameter form → placeholder → add replace ‘placeholder’ with $unemployed_benefit_amount’ → add 100#m (only ZM_2017_reform_7 system).
ii. go back to bun_zm and modify ‘Comp_perTU’ with the new constant

iii. Modify the Output_Var and TAX_UNIT as follows for the ZM_2017_reform_7 system

iv. Whenever you add a new benefit policy to the model, you also need to add it in the policy ildef_ZM in two income lists [Simulated benefits & the relevant income list needed in order for the statistics presenter to work, i.e. (‘Unemployment benefits’ in Statistics Presenter) and put ‘+’ only for ZM_2017_reform_7.
Step 5: Run the ZM_2017 and ZM_2017_reform_7 systems in Statistics Presenter to identify how much this reform will cost the government. What is the impact on poverty and inequality?

Answers: This reform would cost 656.02 million Kwacha in 2017. It would lead to a reduction in the (consumption-based) poverty rate of 0.6 percentage points and it would lead to a decrease in the (consumption-based) Gini coefficient of -0.0026.
Exercise 9: Introduce three benefit reforms and change the income tax policy to cover the costs
Difficulty level: Advanced

Activity:
Introduce all three benefit reforms (child benefit, older persons benefit, and unemployed youth benefit) at the same time, then change the income tax policy to cover the costs.

Question:
What is the impact on poverty and inequality?

Solution

Step 1: Before adding a new policy, you need to add a new system (See Part 1); then make the changes in the new system. Call the new system: ZM_2017_Reform_8.

Step 2: Repeat the process of introducing bch_zm, boa_zm and bun_zm policies, but this time introducing them all into the same reform scenario: ZM_2017_Reform_8.

Step 3: Modify the income tax policy (tin_zm) in the ZM_2017_Reform_8 system by changing the tax band thresholds in the SchedCalc function as follows:

<table>
<thead>
<tr>
<th>SchedCalc</th>
<th>ZM_2017</th>
<th>ZM_2017_Reform_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>Band_LowLim</td>
<td>1</td>
<td>0#y</td>
</tr>
<tr>
<td>Band_Rate</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Band_LowLim</td>
<td>2</td>
<td>39600#y</td>
</tr>
<tr>
<td>Band_Rate</td>
<td>2</td>
<td>0.25</td>
</tr>
<tr>
<td>Band_LowLim</td>
<td>3</td>
<td>49200#y</td>
</tr>
<tr>
<td>Band_Rate</td>
<td>3</td>
<td>0.3</td>
</tr>
<tr>
<td>Band_LowLim</td>
<td>4</td>
<td>74400#y</td>
</tr>
<tr>
<td>Band_Rate</td>
<td>4</td>
<td>0.375</td>
</tr>
<tr>
<td>output_var</td>
<td>tin_s</td>
<td>tin_s</td>
</tr>
<tr>
<td>TAX_UNIT</td>
<td>tu_individual_zm</td>
<td>tu_individual_zm</td>
</tr>
</tbody>
</table>

Step 4: Run the ZM_2017 and ZM_2017_Reform_8 systems in Statistics Presenter to identify how much government expenditure will be on this combination of benefit reforms, and how much extra revenue the government will raise through the tax reform. What are the resulting impacts on poverty and inequality of these combined tax and benefit reforms? Are the reforms budget neutral?

Answers: The three benefit reforms will cost an extra 3,879.26 million kwachas in 2017, but this would be more than offset by the additional income tax revenue of 4,020.46 million kwacha. The combined tax and benefit reforms would lead to a 4.7 percentage point reduction in the poverty rate and a reduction in the Gini coefficient of -0.0323.
**Part D: Changing Eligibility Rules in MicroZAMOD**

**Exercise 10: Amending SCT for 2017**

**Difficulty level: Advanced**

**Activity:**

In 2017 the eligibility rules for SCT changed. Currently, in MicroZAMOD v2.0, the policy rules for SCT are identical to 2016 and are therefore not correct, so we need to update them to reflect the new rules that applied in 2017.

**Questions:**

How much more or less will the government expenditure on SCT be in 2017 under the new SCT rules compared to what would have been required under the old SCT rules? What are the effects on poverty and inequality in 2017 of changing the SCT eligibility rules?

**Solution**

**Step 1:** Firstly we will hide all systems in MicroZMAOD v2.0 except the 2017 system – (this will help us to focus on the changes that we make to the SCT policy).

*Right-click on the column header ‘ZM_2017’ → ‘Move To Hidden Systems Box’ → ‘All Systems But Selected’.*

**Step 2:** Next we need to make a copy of the ‘ZM_2017’ system and call it ‘ZM_2017_SCT’ to show that we are going to model the changes to SCT.
Step 3: Check to make sure that the ‘base’ system for our new ‘ZM_2017_SCT’ system is the ‘ZM_2017’ system so that the ‘conditional formatting’ will show any changes we make in our new system compared to the base system.

Go to the ‘Display’ tab at the top of the screen → ‘Conditional Formatting’ → new ‘ZM_2017_SCT’ system → check the base system is ‘ZM_2017’.
Step 4: Open up the policy ‘bsa_rural_zm’. The eligibility rules for this policy in MicroZAMOD v2.0 currently consist of the following: (i) a district residency test; (ii) a fit-for-work ratio test; and (iii) a living conditions index test. If we open up the function DefVar we see that we have specified four temporary variables to enable us to operationalise these three eligibility criteria. All four of these temporary variables are set to a value of zero for all in the dataset in this initial DefVar function, but they are then re-coded as appropriate in the relevant functions below.

The new SCT eligibility rules in operation in 2017 consist of: (i) the same district residency test as before; (ii) a new ‘elderly person’ eligibility test; (iii) a new ‘female-headed household’ eligibility test; (iv) a new ‘child-headed household’ eligibility test; (v) a new ‘disabled person in the household’ test; and (vi) the same living conditions index test as before. We therefore need to add new temporary variables for the eligibility categories that were not present in the previous SCT rules, and we need to set the i_rural_fit_for_work parameter to n/a in ‘ZM_2017_SCT’ as this eligibility criterion is no longer applicable.

Note: The underpinning dataset already contains a variable called ddi01 which is a binary flag to indicate whether anyone in the household is disabled, so we do not need to add a temporary variable to calculate this. As such, we need new temporary variables for ‘elderly person’, ‘female-headed’ and ‘child-headed’.
Right-clicking on the parameter 'i_rural_fit_for_work' → add three blank [Placeholder] parameters using the 'Add Parameters' tool.
Make sure that the new parameters are all given a value of zero in the ‘ZM_2017_SCT’ system and are set to ‘n/a’ in the base ‘ZM_2017’ system.
Next, we need to remove the ‘fit-for-work’ eligibility test from the new SCT rules in the ‘ZM_2017_SCT’ system. We do this by setting the relevant function to ‘n/a’ (and making sure all the parameters within that function are also set to ‘n/a’). This ensures that the ‘fit-for-work’ test will not form part of the eligibility assessment for SCT in the ‘ZM_2017_SCT’ system.

We can then begin to add the new eligibility criteria into the ‘ZM_2017_SCT’ system.

**Step 5:** We will start with the ‘elderly person’ criterion. The eligibility rule is that any household with an elderly person (aged 65 or over) is now eligible for SCT. Because SCT is paid to the household head in MicroZAMOD (it is possible to change who the benefit is paid to, but this would add unnecessary complexity to the model) we need to construct a function to tell MicroZAMOD that the household head is eligible for SCT if *anyone* in the household is aged 65 or over. There are a few different ways to implement this on the model, but the cleanest way is to use a BenCalc function.

We will place our new BenCalc after the ‘fit-for-work’ Elig function which we have just set to ‘n/a’:

- **Right-click on the ‘fit-for-work’ Elig function** → **‘Add Function After’** → **‘BenCalc’** → **‘Turn this function ‘on’** → **‘put a suitable label in the ‘Comment’ column.**

We now need to specify the eligibility rules:

The first thing to consider is whether this function should perform its assessment at the level of the household or the individual. The policy documentation states that a household is eligible if any member is aged 65 or over. So although we need to assess each individual to see whether he/she is aged 65 or over, we actually want to give the result of the eligibility test to the household head, as it is the head who will potentially be eligible to receive SCT on behalf of the household. We can do this by operationalising this particular function at household level, by specifying the TAX_UNIT to be ‘tu_household_zm’. We can then begin to think through how we need to set up the other parameters in the function.
Starting with the Comp_Cond parameter, we need to specify that each individual should be assessed in terms of their age to see whether they are aged 65 or over. We do this by typing \( \{ \text{dag} \geq 65 \} \) into this parameter. In addition, because we are currently looking at the rural element of SCT, we need to tell the function that we are only interested in people living in rural households, so we add a further condition to this parameter, typing \( \{ \text{dru} = 0 \} \). These two conditions are combined with an ‘&’ sign to specify that only people aged 65 or over AND living in rural areas are able to pass this part of the eligibility test for the rural component of SCT.

**NOTE:** Remember that because we have operationalised this function using the household tax unit, the head of household will be flagged as being eligible on behalf of the household if anyone within the household meets the criteria set out in the Comp_Cond parameter.

In the Comp_per_TU parameter we need to tell MicroZAMOD what value to allocate to household heads where their household meets the ‘elderly’ eligibility criteria. We want to simply flag up a household’s eligibility, so we can add a value of 1 to the Comp_per_TU parameter. This means that for any household that does contain an elderly person, the household head will be allocated a value of 1. All other people in the dataset will receive a value of zero. Finally, we tell MicroZMAOD that we wish for this eligibility flag to be outputted into the temporary variable ‘i_rural_elderly’ which we introduced into the model in the earlier step.
Step 6: Adding the ‘female-headed household’ criterion. To qualify for SCT under this eligibility rule, the person needs to satisfy the following criteria: (i) be the head of household; (ii) be female; (iii) be aged 19-64 inclusive; (iv) not be married; (v) be living in a rural area; and (vi) have three of more dependent children aged 0-18 inclusive.

Unlike the ‘elderly persons’ eligibility criterion above, in which the head of household became eligible even if he or she was not elderly (as long as there was one or more elderly people in the household), the ‘female-headed’ households criterion is slightly more straightforward because, by definition, only the head of household can be eligible. As such, this part of the SCT eligibility rules can be implemented using an Elig function rather than a BenCalc.

Right-click on the BenCalc we added above → ‘Add Function After’ → ‘Elig’ → Turn this function ‘on’ → put a suitable label in the ‘Comment’ column.
As we have discussed, the outcome of an Elig function is a binary 0/1 variable, where a value of 0 means the person does not meet the specified eligibility conditions, and a value of 1 means the person does meet the specified eligibility conditions. Earlier in this practical exercise we added a temporary variable called ‘i_rural_female_headed’ which we intended to populate in this way. In order to enable us to do this, we first need to use the ‘Add parameter’ option to add the ‘Output Var’ parameter to our new Elig function. In the Output_Var parameter we can then type ‘i_rural_female_headed’.

We then need to add the necessary eligibility rules to the Elig_Cond parameter of the function. As noted at the beginning of Step 6 of this exercise, there are six different elements to this particular eligibility rule:

1. The condition for being head of household is \( dhh = 1 \);
2. The condition for being female is \( dgn = 0 \);
3. The conditions for being aged between 19 and 64 inclusive are \( dag \geq 19 \) and \( dag \leq 64 \);
4. The condition for not being married is \( !dms = 2 \) and note that the exclamation mark before the bracket means ‘not’; and
5. The condition for living in a rural area is \( dru = 0 \).

- These are all straightforward to apply as they are all individual level characteristics relating to the person themselves.
6. However, the final eligibility component specifies that, in addition to the above eligibility factors, the person will only be eligible for SCT if she has three or more dependent children aged 0-18 inclusive. As we do not have a variable in the underpinning dataset showing how many children aged 0-18 a person has, we will need to ask MicroZAMOD to calculate this for us ‘on model’. We do this by using one of MicroZAMOD’s in-built ‘queries’: \( nDepChInTu \geq 3 \) which can be translated as ‘number of dependent children in the tax unit’ and, in this case, is expressed as \( \geq 3 \) which means the number of dependent children in the tax unit must be greater or equal to 3. This is one of the many queries that MicroZAMOD automatically calculates and allows users to draw upon.
7. The important thing here is to ensure that we specify the correct tax unit. If we go back up the ‘tudef_zm’ policy towards the top of the model we can expand it out to see that it contains three functions, each of which is a DefTu function. The DefTu function specifies what we mean by ‘household’ or ‘family’ etc, and we can see that currently two of the three functions are turned on. We see that ‘family’ is not currently turned on because it is not use elsewhere in the model. For our ‘female-headed households’ Elig function in the SCT policy below we know we need a type of household tax unit because we need to assess how many dependent children aged 0-18 inclusive belong to the female head of household. Within the ‘tudef_zm’ policy, we can see that the tax unit called ‘tu_household_zm’ is not suitable for our ‘female-headed households’ policy criterion because the ‘tu_household_zm’ tax unit only regards children as being dependent children if they are aged 0-17 inclusive. So, for the purpose of modelling this part of the SCT policy, we need to create another tax unit called ‘tu_household2_zm’ by copying the existing ‘tu_household_zm’ function and pasting it directly below the existing ‘tu_household_zm’ function, then renaming the newly pasted function ‘tu_household2_zm’. Make sure that this new function is turned on. We then need to modify the dependent children age range in this new ‘tu_household2_zm’ tax unit to be between 0-18 inclusive, which is what is needed for this part of the SCT policy. Do to this we simply change the upper age range in the parameter ‘DepChildCond’ to be: \{dag<=18\} & \{dag>=0\}. This means that our definition of dependent children in tax unit ‘tu_household2_zm’ matches the definition of children needed for the ‘female-headed households’ eligibility criterion of SCT.

Turning back to the SCT policy, we need to set the TAX_UNIT parameter to be ‘tu_household2_zm’ in our ‘ZM_2017_SCT’ system. We also need to tell MicroZAMOD to output the resulting 0/1 eligibility flag into the relevant temporary variable that we created earlier: ‘i_rural_female_headed’.
Step 7: Adding in the ‘child-headed household’ criterion. The final new eligibility criterion for SCT in our ‘ZM_2017_SCT’ system is households that are ‘child-headed’. This is the most straightforward of the new eligibility components, as any household headed by a person aged 18 or under who is not married will qualify for SCT under this rule. As this rule is dependent only on the personal information relating to the household head, it can be easily implemented using an Elig function and specifying the tax unit as ‘tu_individual_zm’. To be eligible the person needs to be the head of household \( \{dhh=1\} \) and aged 18 or under \( \{dag<=18\} \) and not be married \( \{dms=2\} \) and be living in a rural area \( \{dru=0\} \). We again need to add the ‘Output Var’ parameter and this time tell MicroZAMOD to place the resulting 0/1 eligibility flag into the relevant temporary variable we created earlier, called ‘i_rural_child_headed’.
Once we have implemented the new ‘child-headed household’ eligibility criterion to the model as explained above, we see that the next functions within this policy are used to implement the ‘living conditions index’ test. It requires eleven separate functions to implement the ‘living conditions index’ test for people living in rural areas, starting with the BenCalc function that assesses the highest level of education in the household and ending with the ArithOp function that is used to transform the composite ‘living conditions index score’ onto a scale ranging from 0 to 1000. We have been informed by our government contacts that the ‘living conditions index’ test has not changed since 2016, and so we do not need to amend these eleven functions.

Step 8: Bringing the constituent eligibility criteria together to apply the overall eligibility rules for SCT.
Following the calculation of the ‘living conditions index’, there are two functions remaining in the bsa_rural_zm policy. The first of these two remaining functions performs the task of assessing each person in the dataset to determine whether they qualify for SCT based on the combination of eligibility criteria. As noted above, some of the eligibility criteria remained the same from 2016 to 2017 (specifically, that to be eligible to claim SCT on behalf of the household the person needs to be: head of household; resident in the same rural district for the last 12 months; and have a living conditions index score lower than the specified threshold), while one eligibility criterion in 2016 was no longer applicable in 2017 (the ‘fit-for-work ratio test’) and a series of new eligibility criteria were introduced in 2017. The new eligibility criteria are the three we modelled above: ‘elderly persons’, ‘female-headed household’ and ‘child-headed household’, plus an additional criterion of having a disabled person within the household. The ‘disabled person’ criterion can be easily incorporated here because we already have a variable in the input dataset called ‘ddi01’ which relates to the presence of a disabled person within the household: if the household contains one or more disabled people, then all members of the household are allocated a value of 1 on the ‘ddi01’ variable, whereas if the household does not contain a disabled person then all members of the household are allocated a value of zero on the ‘ddi01’ variable.
To implement the new SCT rules for 2017 in our ‘ZM_2017_SCT’ system, we can use the contents of the BenCalc function in the ‘ZM_2017’ system as our starting point and modify it accordingly. We need to remove the condition relating to the ‘fit-for-work ratio test’ and add the conditions relating to the new eligibility criteria. It is important here to ensure that the correct usage of the AND (\&) and OR (\|) operators are used to combine the respective components of the overall eligibility test, and remember to use standard brackets (...) to combine conditions when appropriate.

The final BenCalc in this policy allocates an additional ‘disability premium’ to households which quality for SCT and which contain a disabled person. We have been informed during this training course that this component of the SCT policy is still in operation and so we do not need to amend this final function within the policy.

This completes the updating of the 2017 bsa_rural_zm policy in our ‘ZM_2017_SCT’ system.

**Step 9:** repeat for the urban component of the SCT policy. Your next task is to repeat the steps above to update the bsa_urban_zm policy within our ‘ZM_2017_SCT’ system. Note that the same changes made for bsa_rural_zm need to be made for bsa_urban_zm.

**Step 10:** check the results: Once you have completed the changes to both the rural and urban SCT policies, you can run the model and assess the output using the Statistics Presenter. If you use the ‘SOUTHMOD COMPARISON’ option within the Statistics Presenter, specifying the ‘ZM_2017’ system as the ‘base scenario’ and the ‘ZM_2017_SCT’ system as the ‘alternative scenario’, and look at consumption-based poverty measures, then you should get the following results:
### Tax-benefit policy

**Yearly, mill. national currency**

<table>
<thead>
<tr>
<th>Description</th>
<th>Base scenario</th>
<th>Alternative scenario</th>
<th>Difference base vs alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Government revenue through taxes, SSC and indirect taxes</td>
<td>13,168.05</td>
<td>13,168.05</td>
<td>0.00</td>
</tr>
<tr>
<td>... direct taxes</td>
<td>6,548.98</td>
<td>6,548.98</td>
<td>0.00</td>
</tr>
<tr>
<td>... indirect taxes</td>
<td>2,171.23</td>
<td>2,171.23</td>
<td>0.00</td>
</tr>
<tr>
<td>... social security contributions (employee and employer)</td>
<td>4,447.84</td>
<td>4,447.84</td>
<td>0.00</td>
</tr>
<tr>
<td>Government expenditure on social transfers</td>
<td>594.98</td>
<td>938.24</td>
<td>343.25</td>
</tr>
<tr>
<td>... child benefits</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>... social assistance</td>
<td>594.98</td>
<td>938.24</td>
<td>343.25</td>
</tr>
<tr>
<td>... orphan/widow benefits</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>... disabled benefits</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>... unemployment benefits</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>... pension benefits</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Poverty - Consumption based

**after taxes and transfers**

<table>
<thead>
<tr>
<th>Share of poor population, in %:</th>
<th>Base scenario</th>
<th>Alternative scenario</th>
<th>Difference base vs alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>41.43</td>
<td>40.90</td>
<td>-0.53</td>
</tr>
<tr>
<td>Poor households out of ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... male headed households</td>
<td>41.15</td>
<td>40.86</td>
<td>-0.29</td>
</tr>
<tr>
<td>... female headed households</td>
<td>42.58</td>
<td>41.07</td>
<td>-1.51</td>
</tr>
<tr>
<td>... households with children</td>
<td>42.80</td>
<td>42.33</td>
<td>-0.47</td>
</tr>
<tr>
<td>... households with older persons</td>
<td>48.24</td>
<td>45.82</td>
<td>-2.42</td>
</tr>
<tr>
<td>Poverty gap (average normalised poverty gap, FGT(1)):</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>17.56</td>
<td>17.17</td>
<td>-0.40</td>
</tr>
<tr>
<td>Poor households out of ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... male headed households</td>
<td>17.30</td>
<td>17.12</td>
<td>-0.18</td>
</tr>
<tr>
<td>... female headed households</td>
<td>18.64</td>
<td>17.36</td>
<td>-1.28</td>
</tr>
<tr>
<td>... households with children</td>
<td>18.24</td>
<td>17.90</td>
<td>-0.34</td>
</tr>
<tr>
<td>... households with older persons</td>
<td>20.10</td>
<td>17.90</td>
<td>-2.20</td>
</tr>
<tr>
<td>Absolute national poverty line, in national currency, yearly:</td>
<td>2,358.00</td>
<td>2,358.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
## Inequality and the household income distribution - Consumption based

### Consumption based after taxes and transfers, yearly

<table>
<thead>
<tr>
<th></th>
<th>Base scenario</th>
<th>Alternative scenario</th>
<th>Difference base vs alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gini (household income)</td>
<td>0.54</td>
<td>0.54</td>
<td>0.00</td>
</tr>
<tr>
<td>P80/P20</td>
<td>5.31</td>
<td>5.24</td>
<td>-0.07</td>
</tr>
</tbody>
</table>

### Quantiles of distribution and median

<table>
<thead>
<tr>
<th>Quantile</th>
<th>Base scenario</th>
<th>Alternative scenario</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>20th</td>
<td>1,326.31</td>
<td>1,350.02</td>
<td>23.70</td>
</tr>
<tr>
<td>40th</td>
<td>2,271.48</td>
<td>2,307.88</td>
<td>36.40</td>
</tr>
<tr>
<td>50th</td>
<td>2,938.83</td>
<td>2,986.51</td>
<td>47.68</td>
</tr>
<tr>
<td>60th</td>
<td>3,860.18</td>
<td>3,881.63</td>
<td>21.45</td>
</tr>
<tr>
<td>80th</td>
<td>7,042.75</td>
<td>7,072.28</td>
<td>29.53</td>
</tr>
</tbody>
</table>

### Absolute national poverty line, in national currency, yearly

<table>
<thead>
<tr>
<th></th>
<th>Base scenario</th>
<th>Alternative scenario</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>20th</td>
<td>2,358.00</td>
<td>2,358.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>