Handbook for the e3 Prototype Model in Mongolia

Summary

January 2023

Macroeconomic modeling enables policymakers to be aware of how climate change directly affects key economic sectors and how it impacts other sectors along the value chains. Reducing these economic risks needs an integrated and foresightedly adaptation planning. The E3 modeling tool in combination with scenario analysis allows to discover possible future impacts of climate change and adaptation planning and thus supports the transition to a climate-resilient economy.

This E3 model handbook describes the Excel-based DIOM-X model building framework and the simplified E3 prototype model for Mongolia based on an E3 model template and international datasets which can be applied to simulate the macroeconomic effects of country specific climate risks and adaptation measures.

Published by:

giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Supported by:

Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection

based on a decision of the German Bundestag

IKI INTERNATIONAL CLIMATE INITIATIVE

In cooperation with:

GWS SPECIALISTS IN EMPIRICAL ECONOMIC RESEARCH

Table of contents

List of figures

List of tables

List of abbreviations

BAU	Business-as-usual
BN	Billion
CBA	Cost-benefit analysis
CPA	Statistical classification of products by activity
CSV	Comma separated value
DIOM-X	Dynamic Input-Output Models in Excel
E3	Economy, Energy, Emissions
ERC	Energy Regulatory Commission
GDP	Gross Domestic Product
GHG	Greenhouse gas
GIZ	Deutsche Gesellschaft für Internationale Zusammenarbeit
GWS	Institute of Economic Structures Research (Gesellschaft für wirtschaftliche Strukturforschung)
IEA	International Energy Agency
IMF	International Monetary Fund
ILO	International Labor Organization
IO	Input-output
IOT	Input-output table
LHS	Lefthand side
LM	Linear model
MN	Million
MNT	Mongolian Tugrik
MS Excel	Microsoft Excel
NACE	Classification of industries (Nomenclature statistique des activités économiques dans la Communauté européenne)
NPISH	Non-profit institutions serving households
NSO	National statistical office
OLS	Ordinary least squares

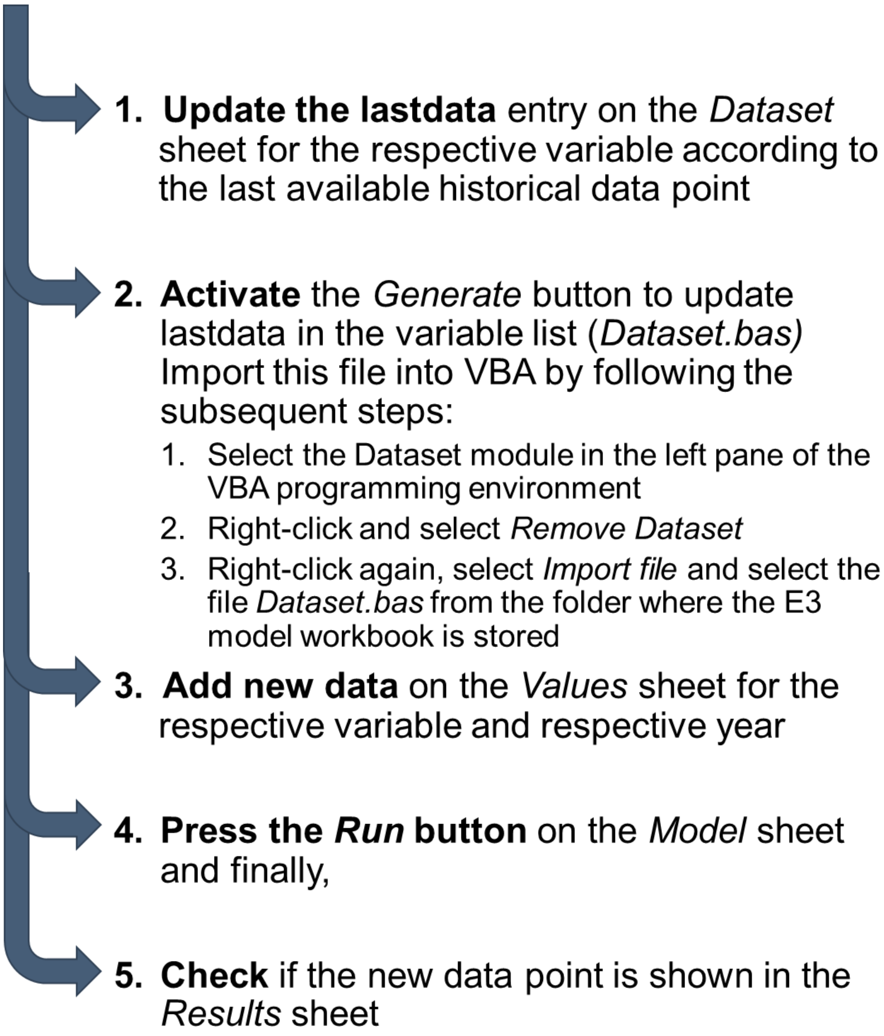
RHS	Righthand side
SNA	System of National Accounts
TFEC	Total final energy consumption
TJ	Terajoule
TPES	Total primary energy supply
UN	United Nations
UNFCCC	United Nations Framework Convention on Climate Change
USD	US Dollar
VBA	Visual Basic for Applications

HOW TO…?

This section provides a brief description of the most important steps that may arise when updating and upgrading the e3.mn model. These steps are explained in greater detail in subsequent sections of the model handbook. The blue arrows show mandatory steps, the turquoise arrows indicate steps that are optional (depending on the context).

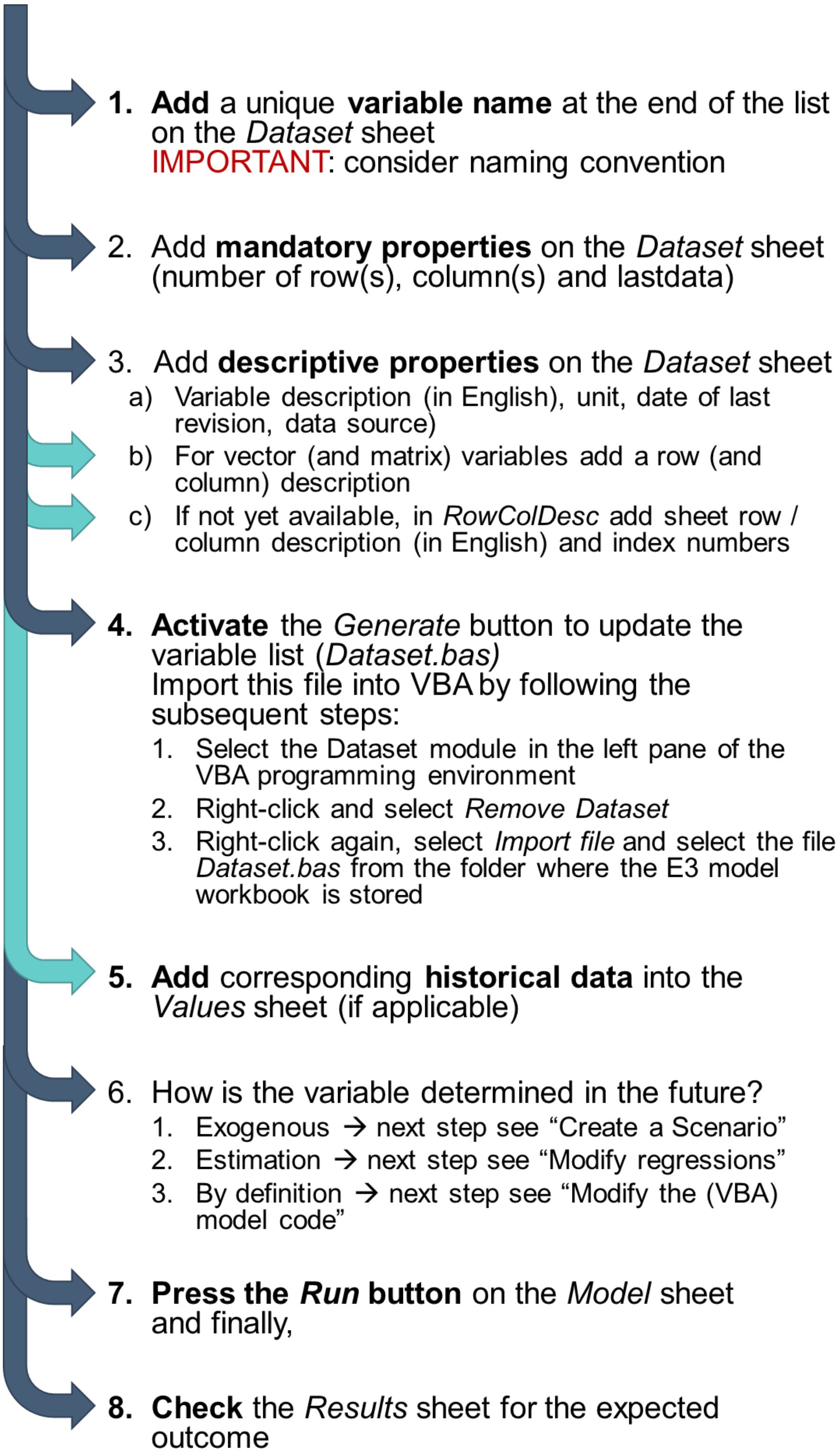
REMEMBER ALWAYS: SAVE YOUR CHANGES!

[How to] Update data for an existing model variable



1. **Update the lastdata** entry on the *Dataset* sheet for the respective variable according to the last available historical data point
2. **Activate** the *Generate* button to update lastdata in the variable list (*Dataset.bas)* Import this file into VBA by following the subsequent steps:
   1. Select the Dataset module in the left pane of the VBA programming environment
   2. Right-click and select *Remove Dataset*
   3. Right-click again, select *Import file* and select the file *Dataset.bas* from the folder where the E3 model workbook is stored
3. **Add new data** on the *Values* sheet for the respective variable and respective year
4. **Press the *Run* button** on the *Model* sheet and finally,
5. **Check** if the new data point is shown in the *Results* sheet

[How to] Add a new model variable



1. **Add** a unique **variable name** at the end of the list on the *Dataset* sheet
   IMPORTANT: consider naming convention

2. Add **mandatory properties** on the *Dataset* sheet (number of row(s), column(s) and lastdata)

3. Add **descriptive properties** on the *Dataset* sheet
   a) Variable description (in English), unit, date of last revision, data source)
   b) For vector (and matrix) variables add a row (and column) description
   c) If not yet available, in *RowColDesc* add sheet row / column description (in English) and index numbers

4. **Activate** the *Generate* button to update the variable list (*Dataset.bas)*
   Import this file into VBA by following the subsequent steps:
   1. Select the Dataset module in the left pane of the VBA programming environment
   2. Right-click and select *Remove Dataset*
   3. Right-click again, select *Import file* and select the file *Dataset.bas* from the folder where the E3 model workbook is stored

5. **Add** corresponding **historical data** into the *Values* sheet (if applicable)

6. How is the variable determined in the future?
   1. Exogenous → next step see “Create a Scenario”
   2. Estimation → next step see “Modify regressions”
   3. By definition → next step see “Modify the (VBA) model code”

7. **Press the *Run* button** on the *Model* sheet and finally,

8. **Check** the *Results* sheet for the expected outcome

[How to] Modify regressions

1. **Press the *Export data for regressions button*** on the *Values* sheet to generate the *series.csv* file

2. **Open** "**R**" **project file** by double-click on *regs.Rproj* in the *regs* directory of the model

3. ***Execute the code lines before startProcessing() in the file allRegs.R*** (Ctrl+Enter) to *load the data set*

4. **Select** appropriate **regression file**, e. g. *employment.R*

   Modify regressions by

   a) Adjusting the limits (*startyear, endyear*) due to new data, if necessary

   b) Adjusting the specification of regression equation

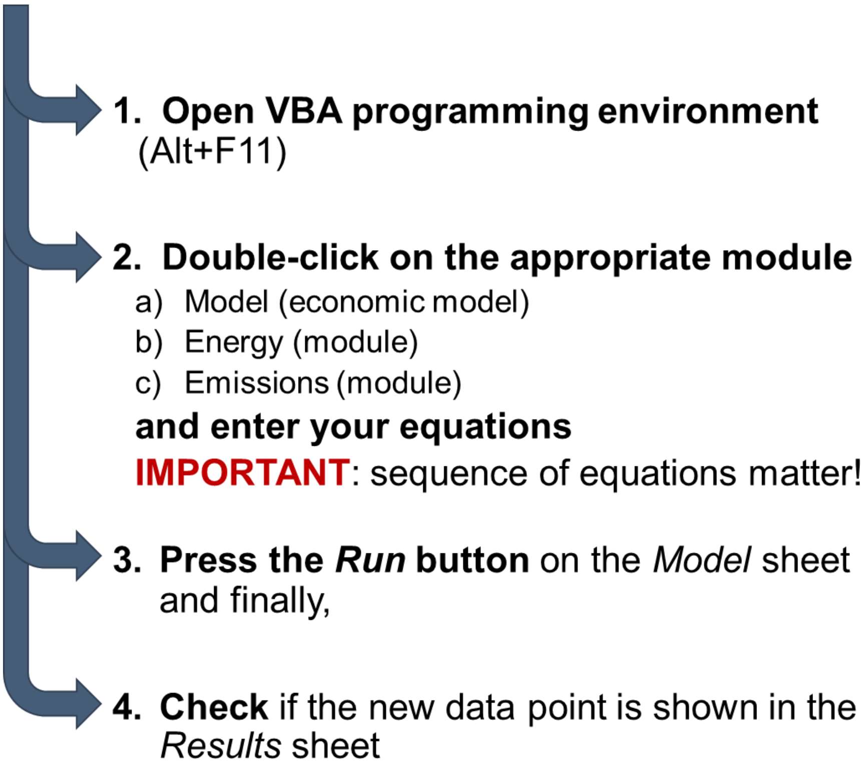
5. **Execute the file allRegs.R** (Ctrl+Shift+Enter) to update the *Regression.bas* file including *estimated parameters and regression equations* in (VBA) model code

6. **Import the *Regression.bas* file into VBA** by following the subsequent steps:
   1. Select the *Regressions* module in the left pane of the VBA programming environment
   2. Right-click and select *Remove Regressions*
   3. Right-click again, select *Import file* and select the file *Regressions.bas* from the folder *regs* where the E3 model workbook is stored

7. **Press the *Run* button** on the *Model* sheet and finally,

8. **Check** the *Results* sheet for the expected outcome

[How to] Modify the (VBA) model code



1. **Open VBA programming environment** (Alt+F11)

2. **Double-click on the appropriate module**
   a) Model (economic model)
   b) Energy (module)
   c) Emissions (module)

   and enter your equations

   IMPORTANT: sequence of equations matter!

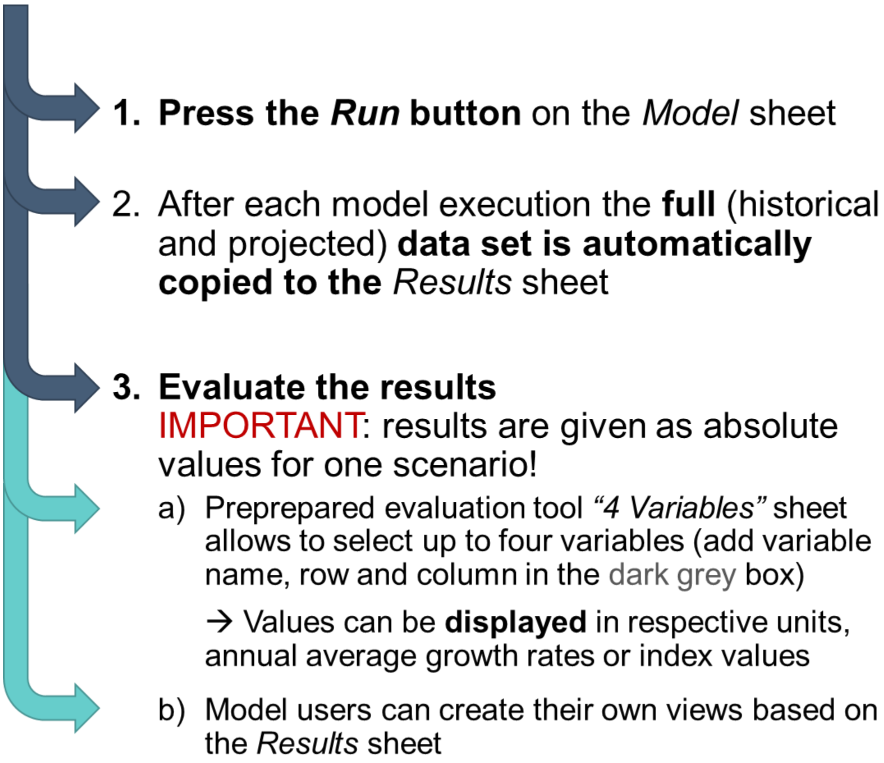
3. **Press the *Run* button** on the *Model* sheet and finally,

4. **Check** if the new data point is shown in the *Results* sheet

[How to] Create a scenario (or [How to] tweak model variables

1. **Add (or select) the model variable to be tweaked** in the *Scenario* sheet
   IMPORTANT: not each variable / each element is tweakable
2. Add **mandatory information** on the *Scenario* sheet (row number, column number, tweak type and activate tweak)
3. If appropriate, activate **interpolation**
4. Enter the **values** for the respective years
   IMPORTANT: tweaking is not possible before lastdata
5. If appropriate, (de-)activate **other tweaks**
6. **Press the *Run* button** on the *Model* sheet and finally,
7. **Check** the *Results* sheet for the expected outcome

[How to] Evaluate results of one scenario



1. **Press the *Run* button** on the *Model* sheet
2. After each model execution the **full** (historical and projected) **data set is automatically copied to the** *Results* sheet
3. **Evaluate the results**
   IMPORTANT: results are given as absolute values for one scenario!
   a) Preprepared evaluation tool *"4 Variables"* sheet allows to select up to four variables (add variable name, row and column in the dark grey box)

   → Values can be **displayed** in respective units, annual average growth rates or index values

   b) Model users can create their own views based on the *Results* sheet

[How to] Compare and evaluate results of two scenarios

1. **Open *_Compare2Scenarios_.xlsb*** in the folder *scenarios* (IMPORTANT: allow execution of macros)
2. ***Press the "1. Step: Update List of Scenarios" button*** on the *SelectScenarios* sheet (new update necessary once new scenarios are calculated)
3. **Click in the left box** on the scenario which should serve as **reference** scenario
4. **Click in the right box** on the scenario which should serve as an **alternative** scenario
5. If appropriate, **extend the list with variables** in the *SelectScenarios* sheet which should be evaluated
6. **Press the "2. Step: Read and COMPARE Results" button** (update necessary once new scenarios are selected)
7. **Analyze results**
   IMPORTANT: results are given as differences between the two scenarios for selected model variables!
   a) Preconfigured graphs in grey colored sheets show differences for a single year.
      The respective year must be selected from the red box on the *SelectScenarios* sheet.
   b) Preconfigured graphs in red colored sheets show differences for the years 2022–2050.

1 Introduction

1.1 E3.cc – a tool for modeling the economic impacts of climate change adaptation scenarios

Environmentally extended macroeconomic models become more and more important to evaluate climate change policies. In combination with scenario analysis, these models are applied to analyze the economy-wide impacts of climate change and adaptation measures (GIZ 2021, see our knowledge products at GIZ). The results provide valuable information for adaptation planning processes. However, they are not limited to these analyses and can be applied to analyze climate protection measures and other economic shocks for example the economic impacts of COVID19 or of Russia's attack on Ukraine.

Building on experiences[1] within the global program "Policy Advice for Climate Resilient Economic Development" (CRED) implemented by Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) on behalf of the Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (BMU), the simplified E3 (Economy-Energy-Emission) model e3.mn for Mongolia was developed based on the model prototype described in GIZ (2022b).

Such an E3 model enhances the single sector analysis of climate change impacts and adaptation measures by incorporating indirect and induced impacts. Supply chain and income-induced impacts are covered and shed light on the sectoral interdependencies and income and demand relationships.

The ability to consider long-term, economic developments (until 2050) in Mongolia supports policy makers to detect possible climate change impacts on key economic sectors and to plan adaptation measures foresightedly.

The e3.mn model is fully developed in Microsoft (MS) Excel using the model building framework DIOM-X (Dynamic Input-Output Modelling in Excel). The framework is built upon the Excel built-in programming language Visual Basic for Applications (VBA) and was developed for creating Dynamic Input-Output Models in Excel (Großmann and Hohmann 2019). The DIOM-X model framework code and its documentation are pre-existing work of GWS. It is distributed with royalty-free, non-exclusive rights to project partners and training participants. It may be used to operate, maintain, and extend the E3 model but not to develop similar and / or commercial products ("fair use").

1.2 Structure of the E3 model handbook

This manual provides a description of the e3.mn model. In particular, it serves as a guide for model builders and model users and is aimed to support the active work with the e3.mn model.

Chapter 2 gives a detailed, technical description of the model which is shown at a glance in Figure 1 (respective manual sections are indicated in parenthesis). In section 2.2 the main model building steps are introduced on which the structure and design of the DIOM-X model building framework are built upon (section 2.3). The Microsoft (MS) Excel file that contains the e3.mn model is described with all its different worksheets and functionalities (section 2.4), the VBA programming environment (section 2.5) and modules (section 2.6). Besides Microsoft Excel, "R" is used to perform regression analysis. Section 2.7 provides information on the linkages between MS Excel and "R" and how the exchange of data and regression equations works.

Chapter 3 provides a description of the three interlinked model parts (economy, energy, emission). First, a general overview of the model building approach of the e3.mn model for Mongolia is presented (section 3.1).

[1] See for example, GIZ, 2022a

In the subsequent sections 3.2 – 3.4, all three model parts are explained by giving a general overview on the modelling approach, the data sets used and the implementation of each part into DIOM-X.

In the last chapter 4, an introduction to scenario analysis with e3.mn is given. The purpose of "what-if"-analyses, data requirements and the procedure on how to implement climate impacts and climate change adaptation scenarios into e3.mn are described.

The manual is accompanied by the Excel file *e3mn.xlsb* (only for project partners and training participants) including the model framework code, the full data set, all model equations including regressions and model results.

If you have any questions and/or comments, please contact us via email:

grossmann@gws-os.com (E3 modelling, scenario analysis)
hohmann@gws-os.com (Excel-based model building framework DIOM-X, E3 modelling)

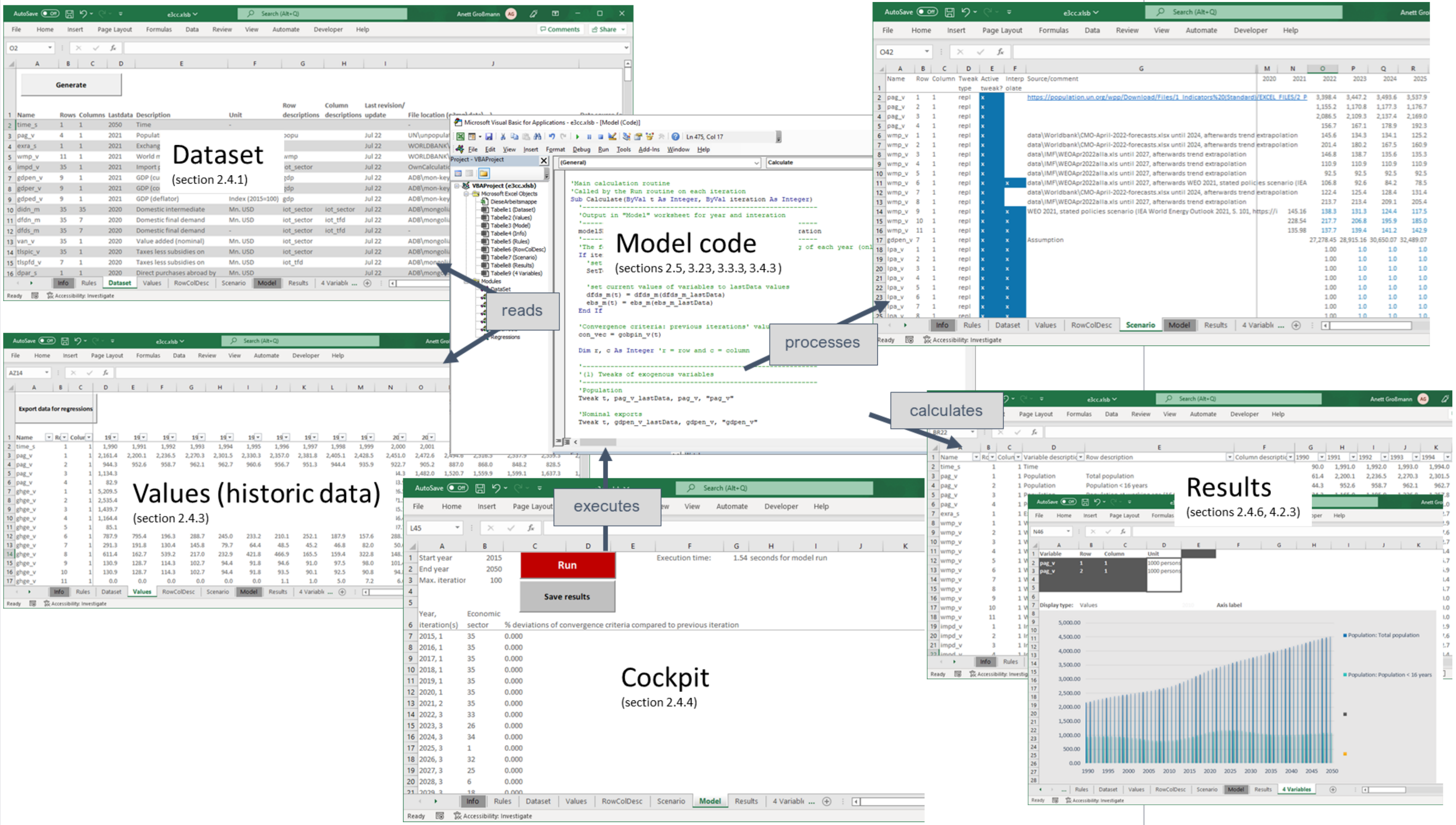


Figure 1: Overview of the Excel-based e3.cc model

Source: own illustration

2 DIOM-X model building framework

2.1 Overview

Today, hundreds of programming languages exist and each of them has been developed to overcome some problems or to improve certain features of other languages. In principle, almost any of these programming languages may be used to build an environmentally enhanced economic model (E3 model), although each of them has its strengths and weaknesses.

MS Excel as a typical spreadsheet program is usually applied for static analytical problems by depicting relations between variables via links between spreadsheet cells. For an elaborated E3 model, this approach is not suitable because the various interdependencies cannot be efficiently expressed via cell linkages. Thus, the model building framework DIOM-X is based on the VBA programming language instead. The core of an E3 model is a set of equations which describe the various model inter-dependencies. The number of variables and equations does not allow for an explicit solution of the equations system but requires an iterative solution algorithm.

Most users of MS Excel neither know nor require the programming language VBA which is part of the MS Office suite of applications. The use of MS Excel as a foundation for Input-Output (IO) modelling has the following advantages:

- MS Excel is already installed on most professionally used computers
- Most statistics are available in MS Excel format
- Existing knowledge about data processing in MS Excel can be reused
- Data (historical, projected) and model code can be stored and distributed as a single unit (= workbook)
- The VBA programming is sufficiently fast to solve equation systems of moderately sized E3 models (from 10 sec up to 1 min per simulation)

A model by definition is always a simplification of a (real world) counterpart. Thus, an E3 model depicts the relevant relations within an economy and between the other E3 model parts but omits detail that is not necessary for the main topic the model tries to address. In order to apply and interpret such a model properly, model builders and model users need to be aware of the set of assumptions and model simplifications. Some models are facing the allegation being a “black-box” due to the fact that underlying data, model assumptions and equations are not fully visible to other modelers. The approach applied solves that issue. It stores the full data set, framework and model code in a single workbook ensures that all aspects of the model can be examined, verified and – if necessary – extended (“white-box” approach).

The framework supports an iterative model building process that encompasses four main steps (section 2.2). The core framework architecture is briefly described in section 2.3. A set of worksheets (section 2.4) represents the visual components of the model. They establish the interaction between users and the model (data set maintenance, scenario definition, model execution, etc.). Section 2.5 discusses the main features of the VBA programming environment. The model code containing both the framework and the equation system is organized in separate programming units called modules (section 2.6) which are embedded in the aforementioned VBA programming environment.

MS Excel as a spreadsheet application offers limited functionality with regard to regression analysis. Section 2.7 shows how to perform regression analysis with “R” and the procedure to include the resulting regression equations and coefficients into VBA.

2.2 Model building steps

Although E3 models greatly differ with respect to their research focus and detail, they share the same basic development cycle:

1. Database management
2. Regression analysis
3. Model implementation
4. Scenario analysis

(1) Database management: The foundation of every quantitative model is data collect-ed from different sources which needs to be harmonized before it can be processed by the model. This involves i.e. assigning variable names and meta information (e.g., unit) as well as harmonizing the layout of the values (e.g., years in columns). DIOM-X pro-vides the necessary template sheets (section 2.4) and data processing routines.

(2) Regression analysis: The availability of historical data is the most important prerequisite for regression analysis which is carried out by econometric models to estimate model parameters for behavioral equations. Instead of using elasticities from the literature, relationships of variables known from e.g., economic theory are econometrically tested against historical data. MS Excel’s capabilities are rather limited compared to more specialized software such as EViews or “R”. In order to easily share historical data, DIOM-X provides a data export routine for the most popular econometric and statistical software packages (see sections 2.4.3 and 2.7).

(3) Model implementation: The main task of this model building step is to create the model structure by assembling the set of equations. Most equations – both regressions and definitions – are not independent but interrelated. Left-hand-side (LHS) variables of one statement occur as right-hand-side (RHS) variables in another equation. One of the biggest problems in this stage is that at the beginning many variables which occur as RHS variables in some equations have not yet been specified. Thus, a lot of mathematical problems occur, e.g., division by zero. To minimize such problems, the DIOM-X framework keeps the values of variables constant in the future unless a specification is given. The framework also helps protect historical data from being overwritten accidently by evaluation of the *lastdata* property that indicates the last historical year for which data is available. The integrated debugging tools of the VBA programming environment further help to track down problems through runtime-inspection and step-by-step execution.

(4) Scenario analysis: DIOM-X provides an easy-to-use mechanism for performing scenario analysis. Model users can specify sets of quantified assumptions which will be automatically injected into the model at runtime. The assumptions can vary with respect to time frames, values and type (e.g., growth rate or value replacement). Once the scenario calculation is finished, the framework automatically copies the full data set (both historic and projected data) to the Results worksheet for further inspection (section 2.4.6).

The more complex the model, the more unlikely it is that these steps can be performed in strict sequence. With most models, these steps are carried out iteratively for various reasons – mostly, because in a certain step missing information or errors are detected which induces another loop.

DIOM-X actively supports this iterative approach by logically separating the different kinds of information.

2.3 Model framework

Although quantitative models greatly differ in detail and their underlying data sets, they share – from a technical point of view – many properties: Data needs to be read in, as-signed to variables, assumptions need to be fed into the model, an equation system must be repeatedly solved until a certain convergence criterion is fulfilled, results need to be stored, etc. The DIOM-X modeling framework captures these similarities in an easy to use, model-independent framework built upon a set of visual and non-visual components.

The visual components are dedicated to the interaction with both model builders and model users, i. e. database maintenance, model execution, scenario definition and visualization of model results. The related information is provided via a set of worksheets which are described in the next section 2.4. The visual components do not require any programming skills, thus model users with knowledge in the handling of the MS Excel application are able to compute and evaluate scenarios as well as maintain the dataset. The interpretation of the results requires model users to be familiar with the core building blocks of the model.

The non-visual components are related to the programming part of the model. The framework provides automated, model-independent procedures to perform scenario analysis as well as a set of helper functions not found in MS Excel to simplify the formulation of the equation system (i.e. matrix algebra). The model framework code is separated from the model code. These separated program units are called modules (see section 2.6).

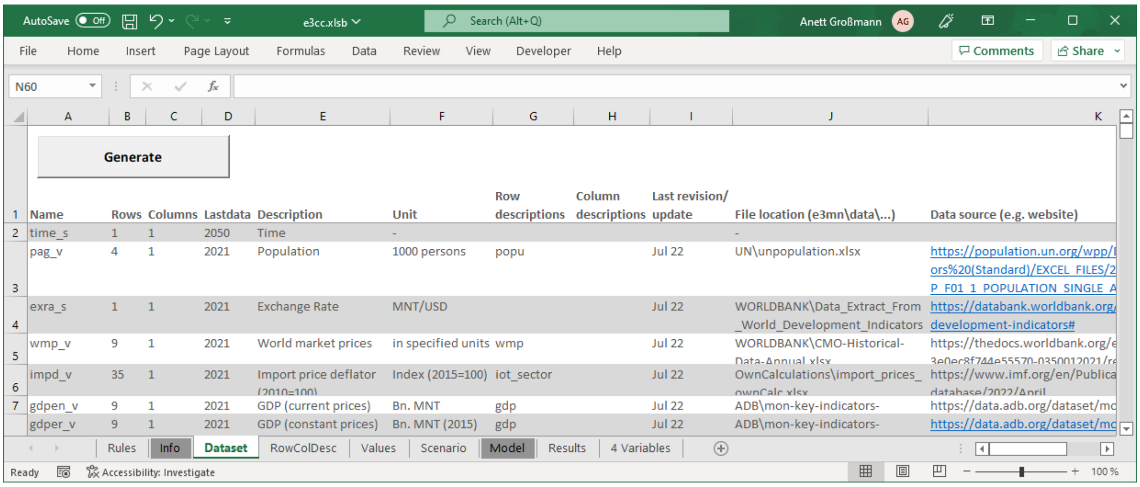
The MS Excel workbook serves as a container for both the visual and non-visual com-ponents which ensures that all relevant information is available any time in its most re-cent version.

2.4 MS Excel sheets

2.4.1 DATASET

The core of any quantitative model is the data set which is compiled from a number of model variables. The DIOM-X framework provides the Dataset worksheet as a template to collect and maintain the set of variables (Figure 2).

Each variable is described by a set of properties. Some of these properties are essential for model operation (core properties, e.g., variable name). Additional properties contain meta-information which is not used by the underlying model code but help understanding and maintaining the data set.



Name	Rows	Columns	Lastdata	Description	Unit	Row descriptions	Column descriptions	Last revision/ update	File location (e3mn\data\...)	Data source (e.g. website)
time_s	1	1	2050	Time	-				-	
pag_v	4	1	2021	Population	1000 persons	popu		Jul 22	UN\unpopulation.xlsx	https://population.un.org/wpp/I ors%20(Standard)/EXCEL_FILES/2 P_F01_1_POPULATION_SINGLE_A
exra_s	1	1	2021	Exchange Rate	MNT/USD			Jul 22	WORLDBANK\Data_Extract_From _World_Development_Indicators	https://databank.worldbank.org/ development-indicators#
wmp_v	9	1	2021	World market prices	in specified units	wmp		Jul 22	WORLDBANK\CMO-Historical- Data-Annual.xlsx	https://thedocs.worldbank.org/e 3e0ec8f744e55570-0350012021/re
impd_v	35	1	2021	Import price deflator (2010=100)	Index (2015=100)	iot_sector		Jul 22	OwnCalculations\import_prices_ ownCalc.xlsx	https://www.imf.org/en/Publica database/2022/April
gdpen_v	9	1	2021	GDP (current prices)	Bn. MNT	gdp		Jul 22	ADB\mon-key-indicators-	https://data.adb.org/dataset/mo
gdper_v	9	1	2021	GDP (constant prices)	Bn. MNT (2015)	gdp		Jul 22	ADB\mon-key-indicators-	https://data.adb.org/dataset/mo

Figure 2: Dataset worksheet

Source: own illustration

The four core properties of a model variable are the following:

Name: Each model variable must be uniquely named. The name is used to address the variable in the model code (i. e. assigning data, performing calculations). With hundreds, sometimes thousands of variable names, identifying each of them becomes difficult without a proper naming convention based on a set of rules. The following rules present a guideline for a naming convention:

- Use lower case characters for variable names only
- Take the first character of every (English) noun / important word describing the variable, e.g., total final demand: *tfd*. If there are only one or two words, try to use four characters for the name, e.g., Employment: *empl*
- Append one character indicating the price concept or shares (if applicable):

n	nominal values, e.g. *tfdn*
d	deflator
r	real
s	shares, e.g., *tfds*

- add a suffix indicating the variable type (see the next properties *Rows* and *Columns* for details):

_v	vector, e.g., *empl_v*
_m	matrix, e.g., *eb_m*
_s	scalar, e.g., *lf_s*

Rows / Columns: The framework distinguishes three types of time series. A simple time series has one value per year (e.g., labor force). For information available for economic activities, the framework allows to combine such information into one variable which stores more than one value per year (e.g., employment by economic activities).

A simple time series with one value per year has only one row and one column. If a variable contains sector information, these values are usually stored in multiple *rows* (*vector*), thus the value of the property Rows is greater than one and the value of *Columns* is equal to one (e.g., employment by economic activities).

For variables which contain information describing certain relations (e.g., energy balance, Input-output table (IOT)) both the number of *rows* and *columns* are greater than one (*matrix*).

In the model code, elements of multidimensional variables are addressed via indices given as integer values.

This approach not only greatly reduces the number of variables but also simplifies the programming of the equation system. The framework contains a set of routines to per-form vector / matrix algebra that help to streamline the model code.

Lastdata: The data of a quantitative model contains both historical and projected data. The *lastdata* property describes the last year for which historical data is available. This information is used by the framework to protect historical data from being (accidently) overwritten.

The other columns in the Dataset worksheet contain meta information which help to further describe a variable. Model builders might be tempted not to properly fill in this information as it is not used by the calculation engine. Considering that often more than one person is engaged into the model building process and that a model usually needs to be maintained for several years, model builders are strongly advised to provide the following additional information for each variable:

Description: This property contains a short description of a variable.

Unit: This property describes the unit of the data. Such information is especially useful for defining / verifying unit conversions in the model code.

Row / Column Description: These columns contain the names of row and column description lists for variables that store more than one value per year (*vectors* and *matrices*). Details are stored in the *RowColDesc* worksheet (see next section).

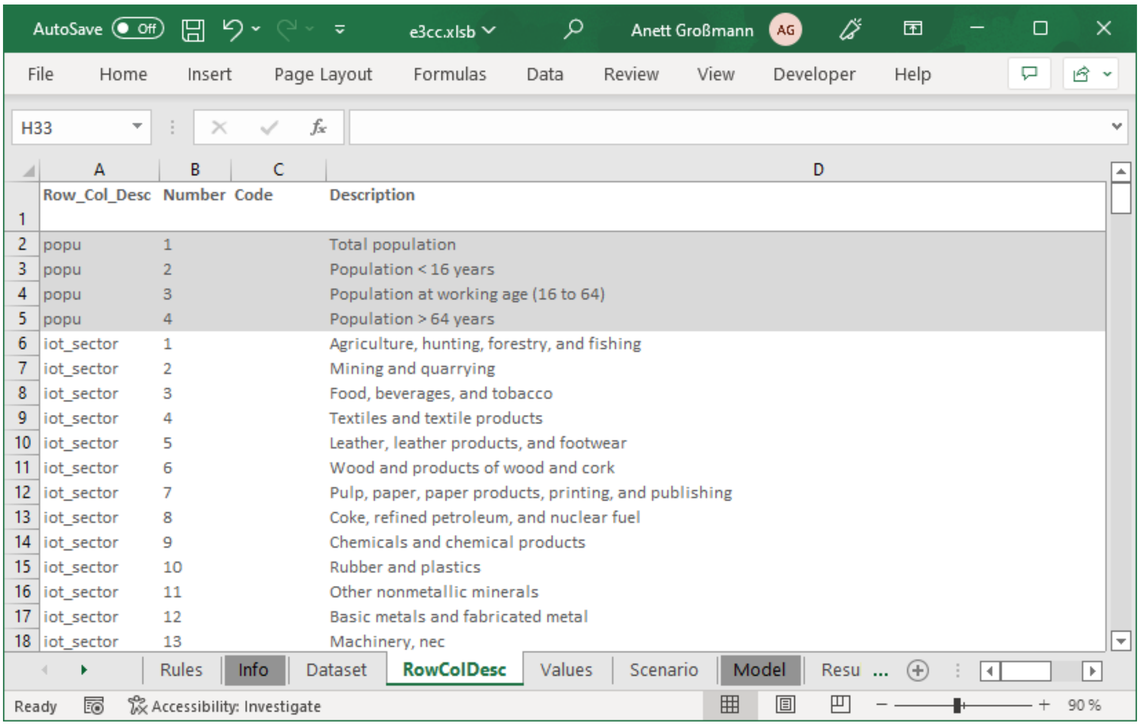
Last revision: This property should be used to indicate when a variable the last time has been checked for updates. This information is essential for model maintenance.

Data Source: This property indicates the provider of the data for a certain variable which could either be some external data source or the model itself if the values are calculated internally or are exogenously given.

The *Generate* button connects the data set to the model code (for details refer to section 2.6).

2.4.2 ROWCOLDESC

This worksheet (Figure 3) contains descriptions for multidimensional model variables (time series of *vectors* and *matrices*).



	A	B	C	D
1	Row_Col_Desc	Number	Code	Description
2	popu	1		Total population
3	popu	2		Population < 16 years
4	popu	3		Population at working age (16 to 64)
5	popu	4		Population > 64 years
6	iot_sector	1		Agriculture, hunting, forestry, and fishing
7	iot_sector	2		Mining and quarrying
8	iot_sector	3		Food, beverages, and tobacco
9	iot_sector	4		Textiles and textile products
10	iot_sector	5		Leather, leather products, and footwear
11	iot_sector	6		Wood and products of wood and cork
12	iot_sector	7		Pulp, paper, paper products, printing, and publishing
13	iot_sector	8		Coke, refined petroleum, and nuclear fuel
14	iot_sector	9		Chemicals and chemical products
15	iot_sector	10		Rubber and plastics
16	iot_sector	11		Other nonmetallic minerals
17	iot_sector	12		Basic metals and fabricated metal
18	iot_sector	13		Machinery, nec

Figure 3: *RowColDesc* worksheet

Source: Own illustration.

Row_Col_Desc: This column contains the name of a list/group of components. As mentioned before, this name is used as a reference in the *Dataset* worksheet (see previous section).

Number: This value serves as an index for the different rows (and columns) of a multi-dimensional variable.

Code: If a statistical item code is available, it may be entered here, e.g., NACE item code.

Description: This property contains an (English) description of the element.

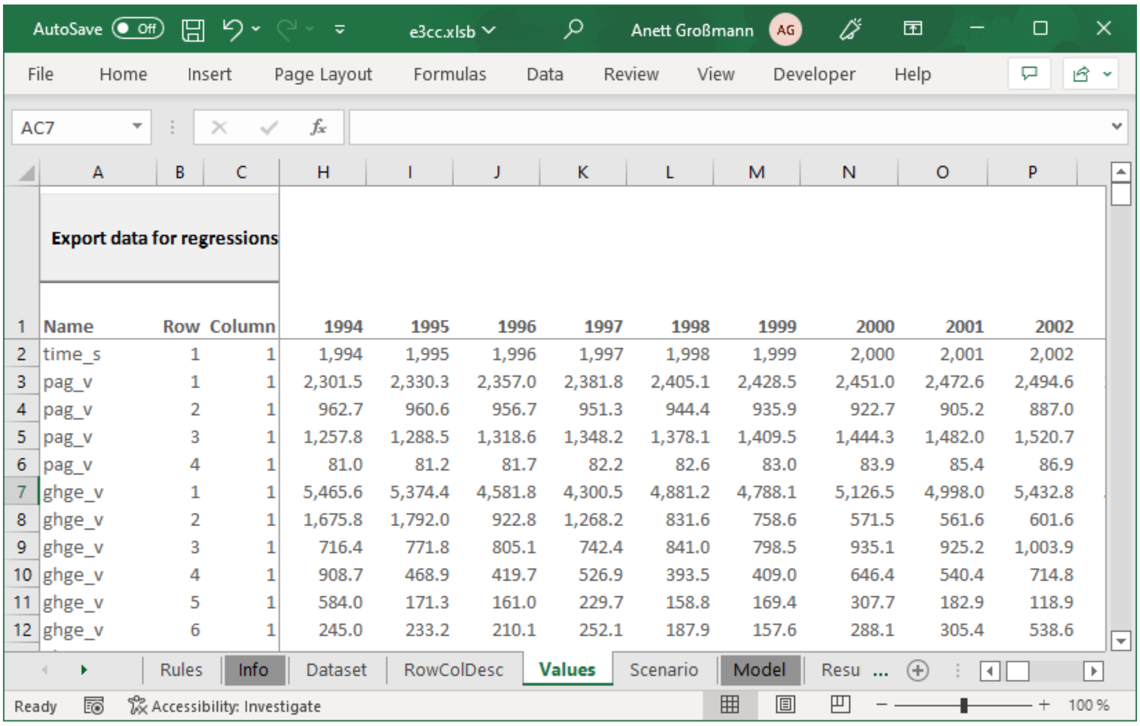
The framework uses this information to annotate the model results in the *Results* worksheet (see section 2.4.6). Thus, model builders and users do not have to remember the variable names and elements of multidimensional variables. Furthermore, creating customized subsets of the data set (dashboards) becomes much easier with all necessary information combined within a single worksheet.

2.4.3 VALUES

The *Values* worksheet contains the historical data that is associated to the variables of the data set (Figure 4). All data that needs to be processed by the model must be part of this worksheet. The data can be incorporated in different ways, depending on the structure of the original data sources as well as the data processing skills of the model builder:

- Copy & paste
- Linking of cells to the original data sources (stored in the *data* folder, cf. section 3.1)

Import via automated data pre-processing tasks in VBA or other programming languages



1	Name	Row	Column	1994	1995	1996	1997	1998	1999	2000	2001	2002
2	time_s	1	1	1,994	1,995	1,996	1,997	1,998	1,999	2,000	2,001	2,002
3	pag_v	1	1	2,301.5	2,330.3	2,357.0	2,381.8	2,405.1	2,428.5	2,451.0	2,472.6	2,494.6
4	pag_v	2	1	962.7	960.6	956.7	951.3	944.4	935.9	922.7	905.2	887.0
5	pag_v	3	1	1,257.8	1,288.5	1,318.6	1,348.2	1,378.1	1,409.5	1,444.3	1,482.0	1,520.7
6	pag_v	4	1	81.0	81.2	81.7	82.2	82.6	83.0	83.9	85.4	86.9
7	ghge_v	1	1	5,465.6	5,374.4	4,581.8	4,300.5	4,881.2	4,788.1	5,126.5	4,998.0	5,432.8
8	ghge_v	2	1	1,675.8	1,792.0	922.8	1,268.2	831.6	758.6	571.5	561.6	601.6
9	ghge_v	3	1	716.4	771.8	805.1	742.4	841.0	798.5	935.1	925.2	1,003.9
10	ghge_v	4	1	908.7	468.9	419.7	526.9	393.5	409.0	646.4	540.4	714.8
11	ghge_v	5	1	584.0	171.3	161.0	229.7	158.8	169.4	307.7	182.9	118.9
12	ghge_v	6	1	245.0	233.2	210.1	252.1	187.9	157.6	288.1	305.4	538.6

Figure 4: *Values* worksheet

Source: Own illustration

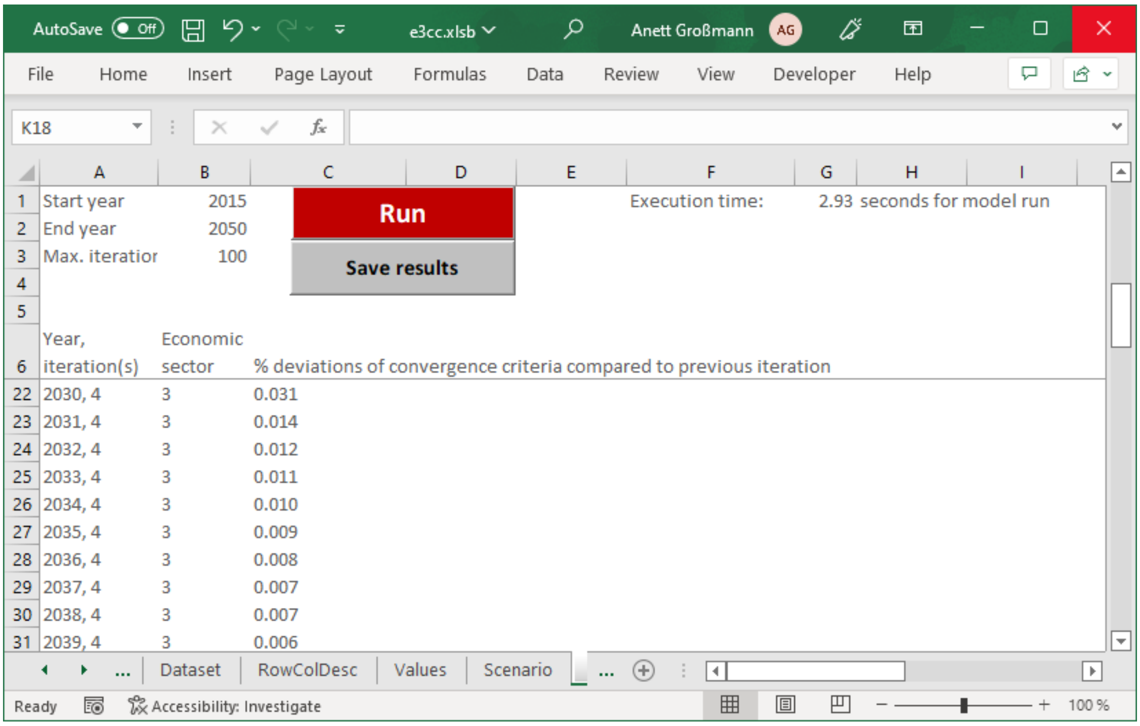
The structure of the data is the same for simple time series as well as for time series of vectors and matrices. In column one *Name*, the variable name is required. In the following two columns, the *Row* and *Column* indices must be specified. In the remaining columns, the values for the respective years must be given. The model framework automatically reads the full data set at the beginning of a simulation and assigns the data to the respective variables.

The historical data is the main prerequisite for step 2: *regression analysis* of the model building process. MS Excel's capabilities to perform regression analysis are rather limited. Thus, the framework on request exports the historical data by pressing the *Export data for regressions* button in CSV format with variable names in columns and years in rows. This special format can not only be imported by “R” but by most other statistical software packages. For details on regression analysis with “R”, please refer to section 2.7.

Variables that are endogenously calculated by the model do not have to appear in the *Values* worksheet.

2.4.4 MODEL

This worksheet is dedicated to the operation of the model (Figure 5). Execution of the model is initiated by pressing the *Run* button which then reads in the historical data, processes the scenario parameters as given in worksheet *Scenario* (see section 2.4.5 and chapter 4), and solves the underlying equation system for the given time span. Details about the algorithm which controls the model execution are presented in section 2.6 which is dedicated to model programming.



	A	B	C	D	E	F	G	H	I
1	Start year	2015				Execution time:	2.93	seconds for model run	
2	End year	2050							
3	Max. iteratior	100							
4									
5									
6	Year, iteration(s)	Economic sector	% deviations of convergence criteria compared to previous iteration						
22	2030, 4	3	0.031						
23	2031, 4	3	0.014						
24	2032, 4	3	0.012						
25	2033, 4	3	0.011						
26	2034, 4	3	0.010						
27	2035, 4	3	0.009						
28	2036, 4	3	0.008						
29	2037, 4	3	0.007						
30	2038, 4	3	0.007						
31	2039, 4	3	0.006						

Figure 5: *Model* worksheet

Source: Own illustration

Execution of the model can be fine-tuned by adjusting the start and end year as well as the maximum number of iterations per year. Changing these values should only be done by experienced model builders after testing model stability for different time spans.

In cases when the model does not successfully converge, raising the number of maxi-mum iterations per year might help to solve the convergence issue. A correctly specified model in most cases should converge with less than twenty iterations per year, though. Higher numbers usually indicate that the model implementation should be checked for unresolved issues.

At model runtime, starting in cell B7 the economic sector with the greatest difference is shown (or the last element of the vector *gobpin_v* used as convergence criteria) compared to the previous iteration for the respective year (cf. Figure 5). In case of convergence issues, it may help to check first that sector to find the reasons for instability.

Model users should be aware that the results are overwritten each time the model is executed. The results of the most recent model run can be stored separately by pressing the *Save results* button (Figure 5).

2.4.5 SCENARIO

In the *Scenario* worksheet, model users define assumptions the model has to consider at model runtime. An assumption is defined as one or more numerical value(s) which depict a certain policy, measure or event (e.g., a natural disaster). It may cover a certain time range, may occur just once or repeatedly. A scenario is a combination of one or more assumptions. Defining, applying and comparing scenarios is the main application of an E3 model based on DIOM-X and does not require any programming skills.

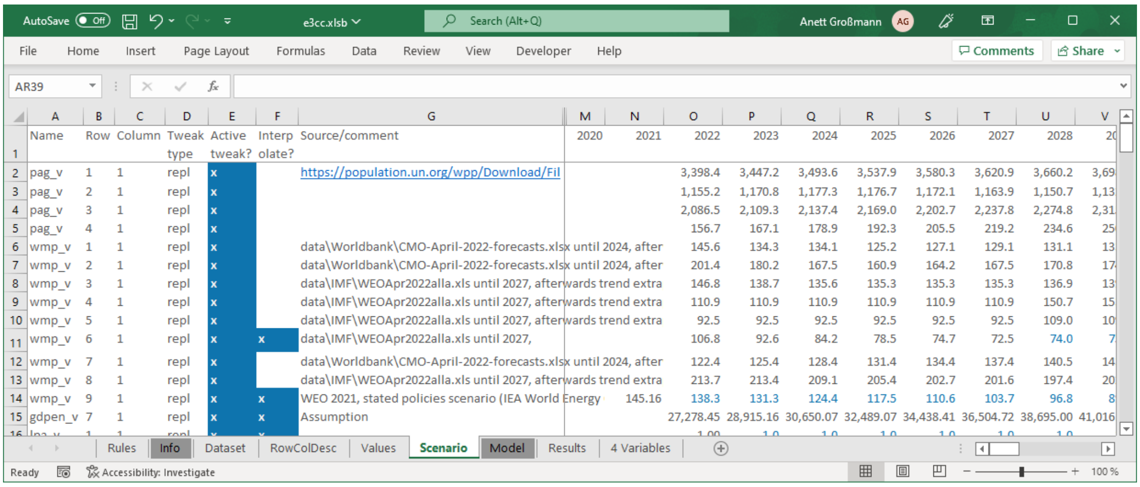
The DIOM-X model building framework provides four different types of adjustments (*tweaks*) which can be used to build simple as well as complex scenarios:

Table 1: Tweak types

Tweak type	Functioning of the tweak	Example
gr	▪ The last value is projected with the given annual average growth rate in %.	▪ Last model value in t: 100, ▪ gr-tweak in t+1: 3 ▪ → result in t+1: 103
add	▪ The given value is added to the value calculated by the model.	▪ Model value: 100, ▪ add-tweak: 30 ▪ → result: 130
mult	▪ The calculated value is multiplied by the given factor.	▪ Model value: 100, ▪ mult-tweak: 1.5 ▪ → result: 150
repl	▪ The calculated value is replaced by the given value.	▪ Model value: 100, ▪ repl-tweak: 110 ▪ → result: 110

Source: Own illustration

The tweaks may be applied to simple time series as well as elements of time series of vectors and matrices (Figure 6).



AutoSave Off
e3cc.xlsb
Search (Alt+Q)
Anett Großmann AG
File Home Insert Page Layout Formulas Data Review View Developer Help
Comments Share
AR39
Rules Info Dataset RowColDesc Values Scenario Model Results 4 Variables
Ready Accessibility: Investigate
100 %

	A	B	C	D	E	F	G	M	N	O	P	Q	R	S	T	U	V
1	Name	Row	Column	Tweak type	Active tweak?	Interpolate?	Source/comment	2020	2021	2022	2023	2024	2025	2026	2027	2028	20
2	pag_v	1	1	repl	x		https://population.un.org/wpp/Download/Fil			3,398.4	3,447.2	3,493.6	3,537.9	3,580.3	3,620.9	3,660.2	3,69
3	pag_v	2	1	repl	x					1,155.2	1,170.8	1,177.3	1,176.7	1,172.1	1,163.9	1,150.7	1,13
4	pag_v	3	1	repl	x					2,086.5	2,109.3	2,137.4	2,169.0	2,202.7	2,237.8	2,274.8	2,31
5	pag_v	4	1	repl	x					156.7	167.1	178.9	192.3	205.5	219.2	234.6	25
6	wmp_v	1	1	repl	x		data\Worldbank\CMO-April-2022-forecasts.xlsx until 2024, after			145.6	134.3	134.1	125.2	127.1	129.1	131.1	13
7	wmp_v	2	1	repl	x		data\Worldbank\CMO-April-2022-forecasts.xlsx until 2024, after			201.4	180.2	167.5	160.9	164.2	167.5	170.8	17
8	wmp_v	3	1	repl	x		data\IMF\WEOApr2022alla.xls until 2027, afterwards trend extra			146.8	138.7	135.6	135.3	135.3	135.3	136.9	13
9	wmp_v	4	1	repl	x		data\IMF\WEOApr2022alla.xls until 2027, afterwards trend extra			110.9	110.9	110.9	110.9	110.9	110.9	150.7	15
10	wmp_v	5	1	repl	x		data\IMF\WEOApr2022alla.xls until 2027, afterwards trend extra			92.5	92.5	92.5	92.5	92.5	92.5	109.0	10
11	wmp_v	6	1	repl	x	x	data\IMF\WEOApr2022alla.xls until 2027,			106.8	92.6	84.2	78.5	74.7	72.5	74.0	7
12	wmp_v	7	1	repl	x		data\Worldbank\CMO-April-2022-forecasts.xlsx until 2024, after			122.4	125.4	128.4	131.4	134.4	137.4	140.5	14
13	wmp_v	8	1	repl	x		data\IMF\WEOApr2022alla.xls until 2027, afterwards trend extra			213.7	213.4	209.1	205.4	202.7	201.6	197.4	20
14	wmp_v	9	1	repl	x	x	WEO 2021, stated policies scenario (IEA World Energy		145.16	138.3	131.3	124.4	117.5	110.6	103.7	96.8	8
15	gdpen_v	7	1	repl	x	x	Assumption			27,278.45	28,915.16	30,650.07	32,489.07	34,438.41	36,504.72	38,695.00	41,016

Figure 6: *Scenario* worksheet

Source: Own illustration.

The necessary information to apply a tweak to a certain variable must be given as follows:

Name: The name of the variable to be tweaked. It must already be part of the data set.

Row and **Column**: The row and column indices of the element which is tweaked. If a simple time series is tweaked, both indices must be 1.

Tweak type: The name of one of the four tweak types as listed in Table 1.

Active tweak: Indicates whether the tweak is active (x) or not (blank). This setting helps to formulate and experiment with complex scenarios by trying different combinations of tweaks.

Interpolate: If this setting is activated (x), the framework automatically interpolates the values between two non-consecutive years. This greatly reduces the amount of numbers that need to be typed in.

The remaining columns contain the years with the respective tweak values.

If necessary, a combination of different tweaks may be applied to a single variable. For example, a growth rate could be applied for a certain time span which is then followed by a *replace* tweak for following years.

The content of the *Scenario* worksheet is automatically processed by the framework at model runtime: If a tweak is defined for a certain variable, the given value(s) adjust(s) or replace(s) the values which were calculated by the model.

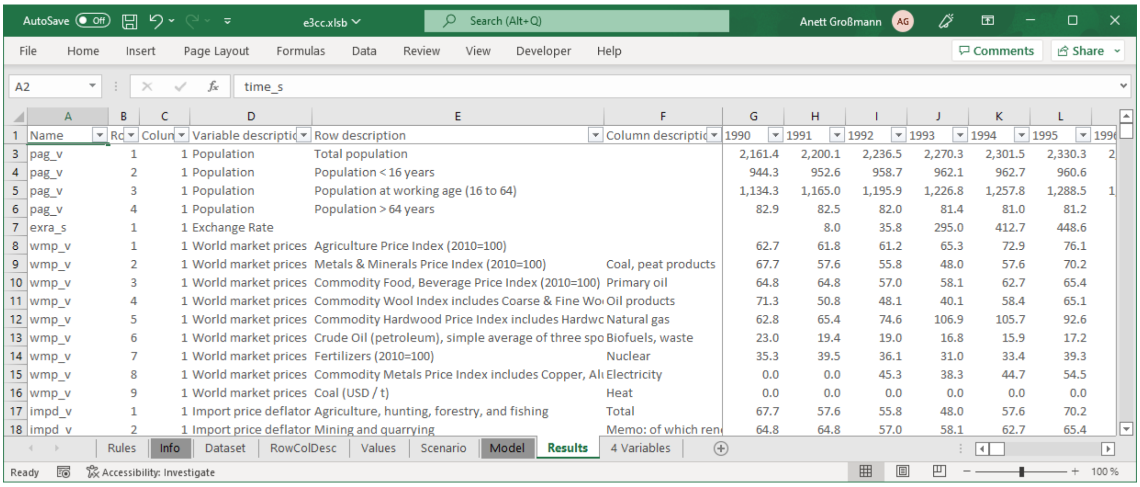
Model builders and model users should be aware of the fact that – due to the high degree of interdependency in the equation system – the full set of effects of a certain tweak is not always obvious. Therefore, it is strongly advised to test and analyze assumptions separately and build more complex combinations of assumptions later. Furthermore, it must be pointed out that unplausible assumptions usually yield unplausible results as the model is not able to verify the plausibility of any assumption.

Moreover, model users should be aware that not all variables of the data set can be tweaked. Column N in the worksheet *Dataset* (section 2.4.1) indicates if a model variable is tweakable, party tweakable or not tweakable. The model code must contain anchors for tweakable variables which need to be defined by the model builder (see section 2.6.3). Tweaks that do not have these anchors in the model code are simply ignored and thus have no effect or impact. In principle, only exogenous variables or variables which are determined via behavioral equations should be tweakable.

The practical use of the tweaking mechanism for scenario building is discussed in detail in chapter 4.

2.4.6 RESULTS

At the end of each model execution, the framework automatically copies the full data set (both historical and projected data) to the *Results* worksheet (Figure 7).



	A	B	C	D	E	F	G	H	I	J	K	L	
1	Name	Ro	Colun	Variable descriptio	Row description	Column descriptio	1990	1991	1992	1993	1994	1995	199
3	pag_v	1	1	Population	Total population		2,161.4	2,200.1	2,236.5	2,270.3	2,301.5	2,330.3	2
4	pag_v	2	1	Population	Population < 16 years		944.3	952.6	958.7	962.1	962.7	960.6	
5	pag_v	3	1	Population	Population at working age (16 to 64)		1,134.3	1,165.0	1,195.9	1,226.8	1,257.8	1,288.5	1
6	pag_v	4	1	Population	Population > 64 years		82.9	82.5	82.0	81.4	81.0	81.2	
7	exra_s	1	1	Exchange Rate				8.0	35.8	295.0	412.7	448.6	
8	wmp_v	1	1	World market prices	Agriculture Price Index (2010=100)		62.7	61.8	61.2	65.3	72.9	76.1	
9	wmp_v	2	1	World market prices	Metals & Minerals Price Index (2010=100)	Coal, peat products	67.7	57.6	55.8	48.0	57.6	70.2	
10	wmp_v	3	1	World market prices	Commodity Food, Beverage Price Index (2010=100)	Primary oil	64.8	64.8	57.0	58.1	62.7	65.4	
11	wmp_v	4	1	World market prices	Commodity Wool Index includes Coarse & Fine Wo	Oil products	71.3	50.8	48.1	40.1	58.4	65.1	
12	wmp_v	5	1	World market prices	Commodity Hardwood Price Index includes Hardwc	Natural gas	62.8	65.4	74.6	106.9	105.7	92.6	
13	wmp_v	6	1	World market prices	Crude Oil (petroleum), simple average of three spo	Biofuels, waste	23.0	19.4	19.0	16.8	15.9	17.2	
14	wmp_v	7	1	World market prices	Fertilizers (2010=100)	Nuclear	35.3	39.5	36.1	31.0	33.4	39.3	
15	wmp_v	8	1	World market prices	Commodity Metals Price Index includes Copper, Alı	Electricity	0.0	0.0	45.3	38.3	44.7	54.5	
16	wmp_v	9	1	World market prices	Coal (USD / t)	Heat	0.0	0.0	0.0	0.0	0.0	0.0	
17	impd_v	1	1	Import price deflator	Agriculture, hunting, forestry, and fishing	Total	67.7	57.6	55.8	48.0	57.6	70.2	
18	impd_v	2	1	Import price deflator	Mining and quarrying	Memo: of which ren	64.8	64.8	57.0	58.1	62.7	65.4	

Figure 7: *Results* worksheet

Source: Own illustration.

The layout of this worksheet basically matches the layout of the *Values* worksheet. In addition, for each variable and its components the row and column descriptions are included.

It is important to keep in mind that this worksheet is automatically overwritten each time the model is executed. The main reason is that with higher numbers of calculated scenarios the size of the workbook as well as processing time for storing/loading would increase dramatically. It is up to the model user to save relevant model results separately by pressing the *Save results* button on the *Model* worksheet (cf. section 2.4.4).

Model users and model builders use the *Results* worksheet to evaluate scenarios, either by looking at certain variables of one scenario or by comparing variables from different scenarios (see section 4.2.3). By having all data for one scenario in a single worksheet, the application of MS Excel's functionality to create tables and graphs is straight forward and does not require any programming skills or in-depth knowledge of MS Excel. Advanced users may create their own tools for scenario evaluation.

2.5 VBA Programming Environment

The VBA programming environment of the MS Office suite of programs is not used by most regular users. Thus, MS decided not to activate it in the main menu ribbon. Activation can be accomplished by either pressing the key combination **Alt+F11** or by customizing the main menu ribbon via the menu entry **File / Options / Customize** ribbon:

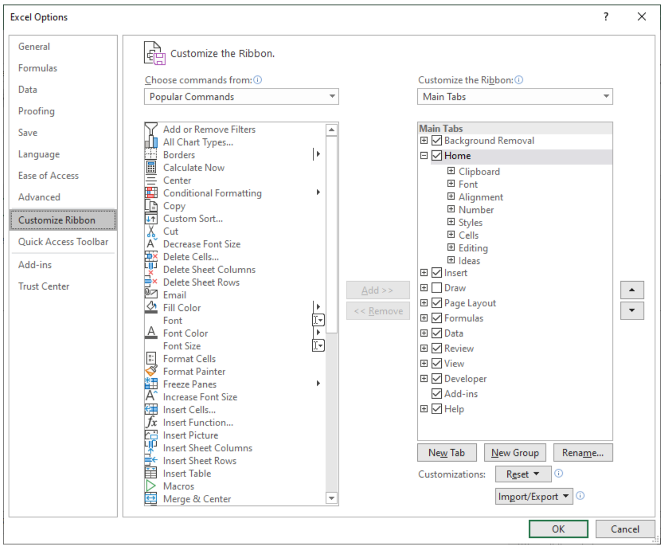
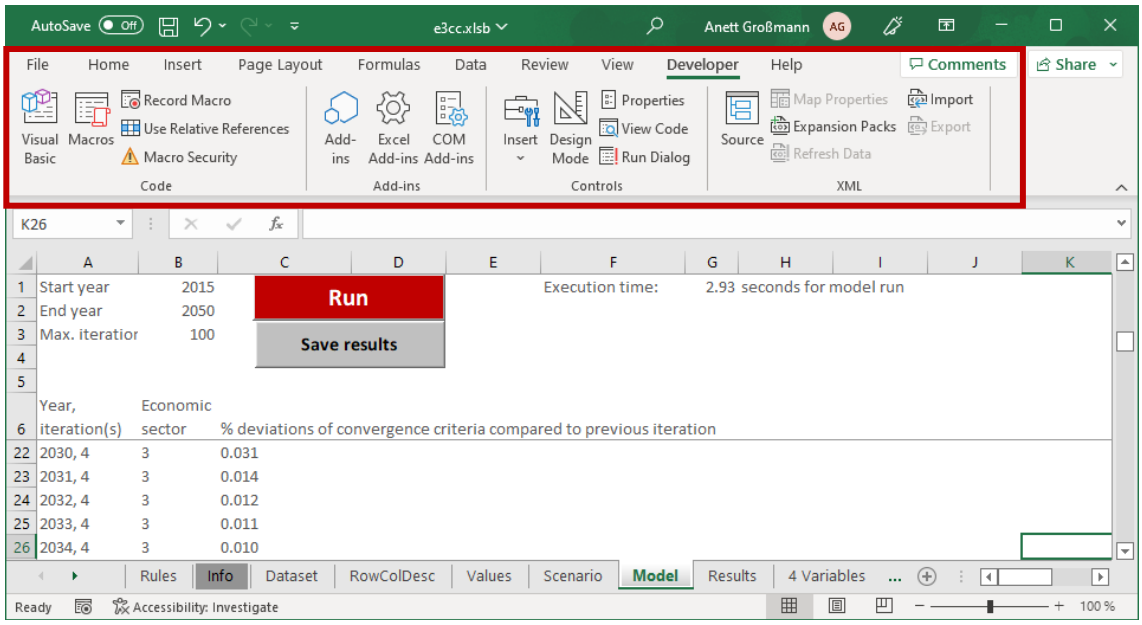


Figure 8: Excel option for developers

Source: Own illustration.

To activate the developer options, the *Developer* entry in the right pane of the dialog needs to be selected (Figure 8). This reveals the hidden developer entries in the main menu ribbon (Figure 9):



	A	B	C	D	E	F	G	H
1	Start year	2015				Execution time:	2.93	seconds for model run
2	End year	2050						
3	Max. iteratior	100						
4								
5								
6	Year, iteration(s)	Economic sector	% deviations of convergence criteria compared to previous iteration					
22	2030, 4	3	0.031					
23	2031, 4	3	0.014					
24	2032, 4	3	0.012					
25	2033, 4	3	0.011					
26	2034, 4	3	0.010					

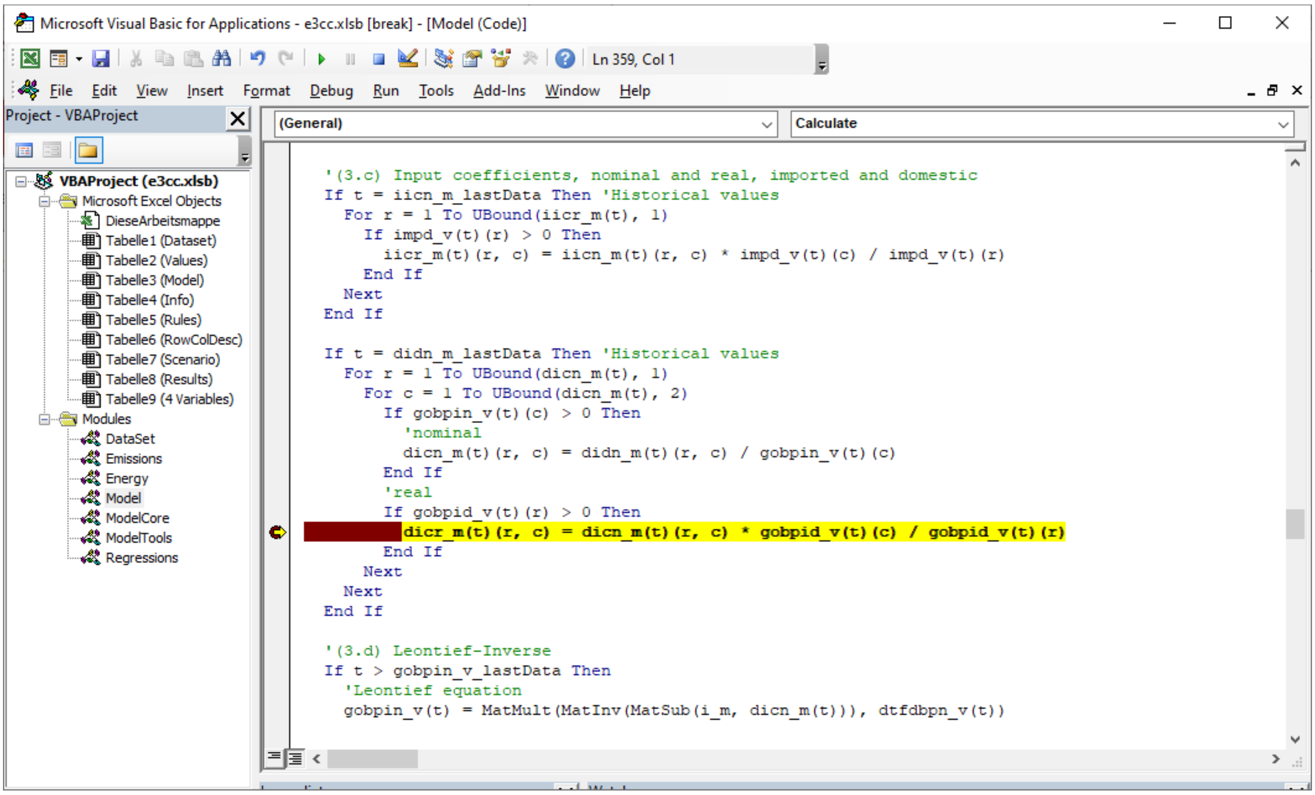
Figure 9: Excel ribbon with activated developer options

Source: Own illustration.

By clicking the *Visual Basic* button, the programming environment is opened. The aforementioned key combination **Alt+F11** has the same effect (Figure 10).

It is out of the scope of this document to explain the programming environment in detail. Model builders not being familiar with VBA should refer to the huge number of internet resources and books that are available in different levels of detail and languages.

One of the most important features is the integrated debugger which allows for interrupting model execution at any program statement. Once execution is paused, the full data set and other internally used variables (e.g., loop variables) can be examined which greatly helps with tracking down data or programming issues. Model builders should familiarize themselves with the different entries in the *Debug* menu.



```
'(3.c) Input coefficients, nominal and real, imported and domestic
If t = iicn_m_lastData Then 'Historical values
  For r = 1 To UBound(iicr_m(t), 1)
    If impd_v(t)(r) > 0 Then
      iicr_m(t)(r, c) = iicn_m(t)(r, c) * impd_v(t)(c) / impd_v(t)(r)
    End If
  Next
End If

If t = didn_m_lastData Then 'Historical values
  For r = 1 To UBound(dicn_m(t), 1)
    For c = 1 To UBound(dicn_m(t), 2)
      If gobpin_v(t)(c) > 0 Then
        'nominal
        dicn_m(t)(r, c) = didn_m(t)(r, c) / gobpin_v(t)(c)
      End If
      'real
      If gobpid_v(t)(r) > 0 Then
        dicr_m(t)(r, c) = dicn_m(t)(r, c) * gobpid_v(t)(c) / gobpid_v(t)(r)
      End If
    Next
  Next
End If

'(3.d) Leontief-Inverse
If t > gobpin_v_lastData Then
  'Leontief equation
  gobpin_v(t) = MatMult(MatInv(MatSub(i_m, dicn_m(t))), dtfdbpn_v(t))
```

Figure 10: *Model debugging*

Source: Own illustration.

Another important feature is the concept of separate program units called modules (cf. Figure 9). Program modules allow for

- Separating model-specific from model-independent code (the framework)
- Dividing the model structure into smaller logical units

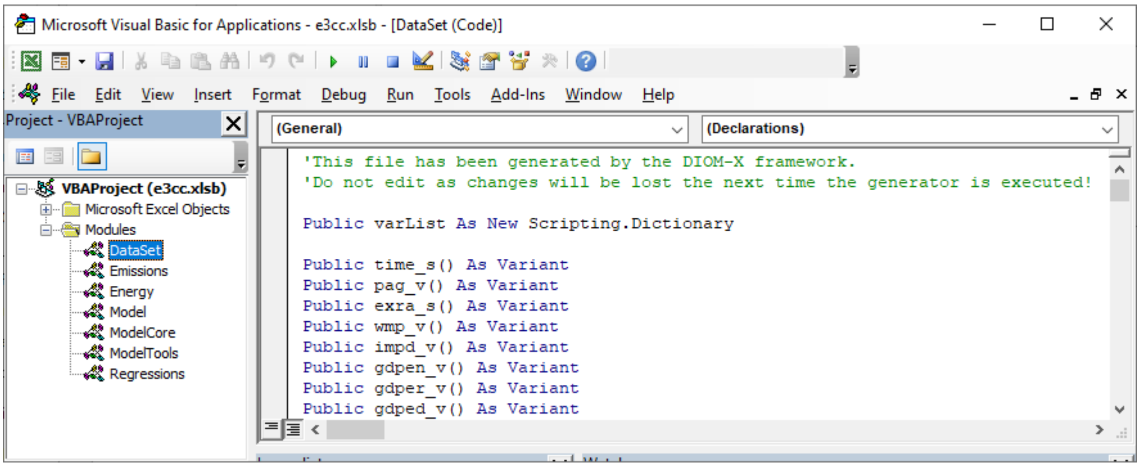
and thus, greatly improve the structure of a model. Model builders are encouraged to use this feature where appropriate.

The following section discusses the different modules of the DIOM-X model building framework.

2.6 Modules

2.6.1 DATASET

In section 2.4.1, the data set and the different properties of model variables were introduced. The definition of the data set is not sufficient to be able to use the variables in the model code (i. e. the equation system), though. For each of these variables, a variable declaration in VBA syntax is required (Figure 11). To prevent model builders from having to manually perform variable declarations, the *Dataset* worksheet contains the *Generate* button which instructs the framework to generate the full set of variable declarations automatically. The list is saved to a file called *Dataset.bas* in the same directory where the model workbook is stored. For security reasons, Excel does not allow to import this file automatically. This measure ensures that a MS Excel workbook cannot easily create and activate malicious code.



```
'This file has been generated by the DIOM-X framework.
'Do not edit as changes will be lost the next time the generator is executed!

Public varList As New Scripting.Dictionary

Public time_s() As Variant
Public pag_v() As Variant
Public exra_s() As Variant
Public wmp_v() As Variant
Public impd_v() As Variant
Public gdpen_v() As Variant
Public gdper_v() As Variant
Public gdped_v() As Variant
```

Figure 11: *Dataset* module

Source: Own illustration.

The necessary steps to import the module containing the variable declarations are as follows:

1. Select the *Dataset* module in the left pane of the VBE programming environment:
2. Right-click and select *Remove Dataset*
3. Right-click again, select *Import file* and import the file *Dataset.bas* from the folder where the workbook is stored.

Unfortunately, Excel does not allow to overwrite an existing module. Therefore, the existing *Dataset* module needs to be removed first.

The framework not only generates the proper variable declarations but also constants for the l*astdata* property and specific code to automatically assign the available historical data to the respective variables.

This procedure must be repeated if the data set has changed (e.g., changes to the *lastdata* property, added/removed variables, changes in the number of rows/columns of a variable) to ensure proper model operation.

2.6.2 MODEL CORE

The *ModelCore* module contains the model-independent DIOM-X framework code which is used by the model builder “as is” and therefore not discussed in detail.

The most important part of the framework is the processing algorithm which gets executed if the *Run* button is pressed (see section 2.4.4 by the model user. The schematic representation of the algorithm is shown in the following Nassi-Shneiderman diagram (Figure 12):

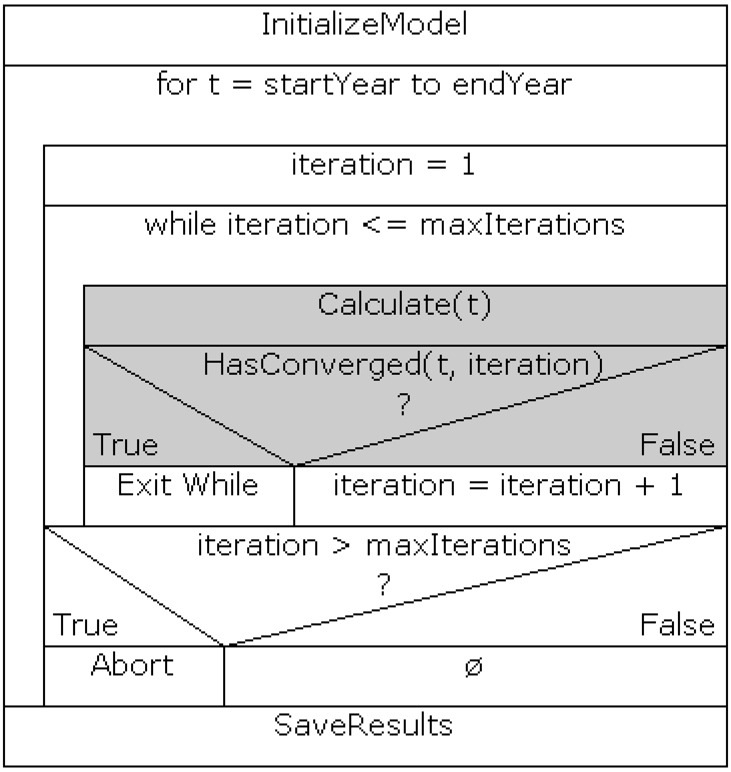

InitializeModel
for t = startYear to endYear
iteration = 1
while iteration <= maxIterations
Calculate(t)
HasConverged(t, iteration)
?
True
False
Exit While
iteration = iteration + 1
iteration > maxIterations
?
True
False
Abort
ø
SaveResults

Figure 12: Model processing algorithm (Nassi-Shneiderman diagram)

Source: Own illustration.

As shown in the figure, the algorithm is responsible for the full set of actions that are related to model processing ranging from initializing the model (i. e. reading in and as signing the historical data) to saving model results to the *Results* worksheet (see section 2.4.6).

An elaborated E3 model consists of a large set of variables and equations which usually needs to be solved in iterations. A typical and widely used approach is to create a huge matrix containing coefficients and apply a solving algorithm such as Gauss-Seidel. The main disadvantage of this approach is that the equations are not explicitly accessible as code statements. This complicates troubleshooting especially in cases where the algorithm fails to successfully solve the equation system.

The DIOM-X framework promotes a different approach: For each year, the algorithm iteratively calls two routines (gray boxes in Figure 12) which need to be provided by the model builder in the *Model* module (see section 2.3.6). The *Calculate* subroutine gets called first and has to provide the full set of model equations. After each iteration, the framework calls the *HasConverged* function. Its return value indicates whether the attempt to solve the equation system was successful (return value *true*) or not (return value *false*). If the return value is *false*, the framework initiates another iteration. In some cases, a model cannot converge (i. e. due to missing specifications, data problems). By evaluating the value of the maximum number of iterations as given on the *Model* worksheet (see section 2.4.4), the framework ensures that the model is not caught in an endless loop. If the number of iterations reaches the given upper limit, the model prematurely stops execution and informs the model user that the calculation of the scenario was not successful.

As already pointed out, the equation system usually cannot be solved explicitly. Thus, the model builder has to define a proper convergence criterion in the *HasConverged* function that indicates whether the model successfully converged or not. A common approach is to monitor the value of a certain key variable. If the percentage difference between two consecutive iterations is lower than a given threshold, the *HasConverged* function returns true which instructs the framework to proceed with the calculation for the next year.

Some models need to initialize variables which are not part of the data set (e.g. a unit matrix. In these cases, the model builder can use a subroutine *InitializeModel* which is once called by the framework at model start.

Scenario analysis is performed by tweaking the values of certain variables at model runtime. The framework replaces the calculated values of these variables if

- The *Scenario* worksheet contains active tweak entries and
- A corresponding tweak statement is found in the model code as part of the aforementioned *Calculate* routine in the *Model* module (see section 2.6.3).

The tweak statement is defined as

```
Sub Tweak(ByVal t As Integer, ByVal lastData As Integer,
ByRef var As Variant, ByVal varName As String)
```

t: Year to check for the presence of a tweak.

lastData: Last year of available data of the variable.

Var: Variable reference

varName: Variable name to apply the tweak for

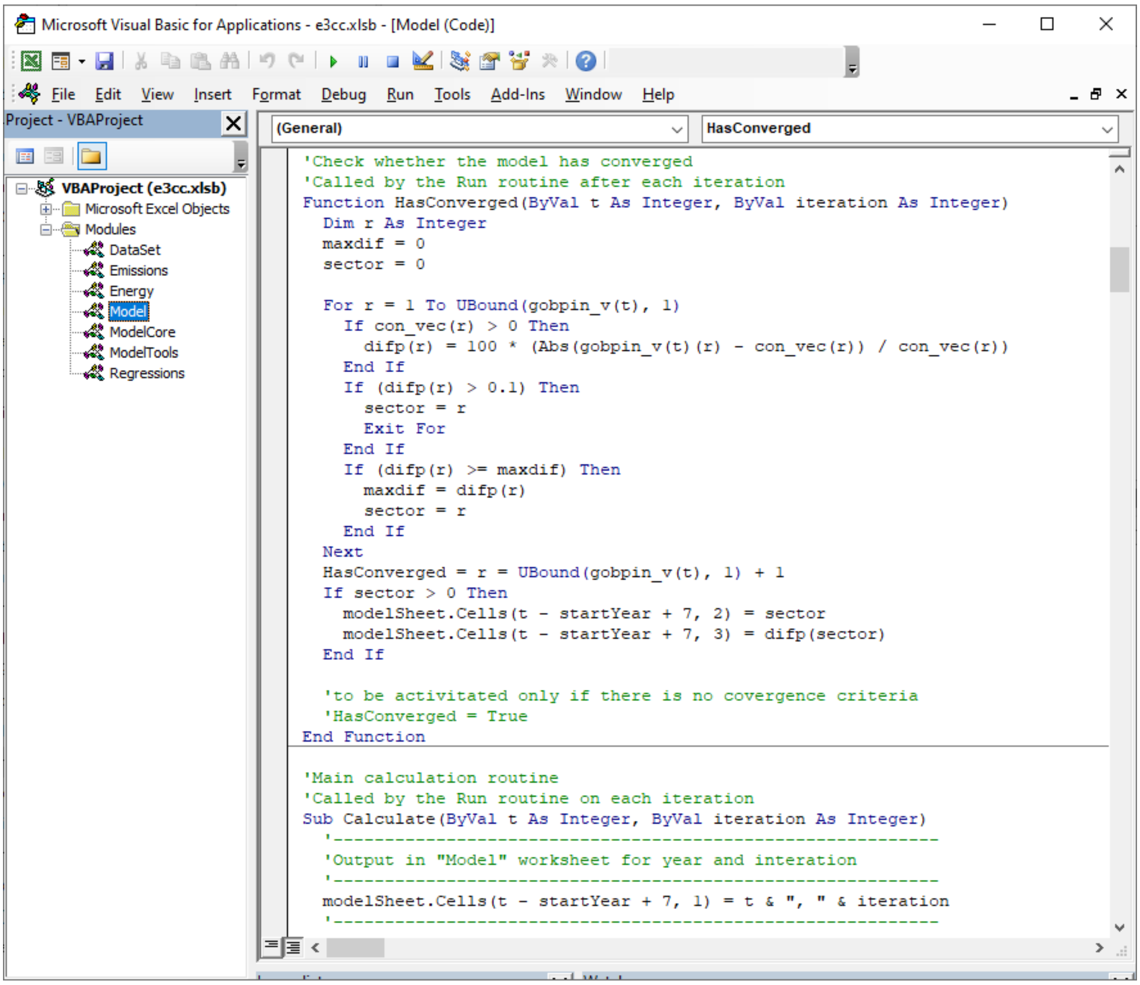
Example: `Tweak t, lf_s_lastData, lf_s, “lf_s”`

For convenience, the model framework automatically generates a constant for the “last data” value of each variable by appending *_lastData* to the respective variable name.

In addition to routines that are dedicated to proper model operation and scenario specification, the framework provides a set of routines related especially to matrix algebra which are not present in the MS Excel libraries (e.g., matrix multiplication).

2.6.3 MODEL

The Model module contains the model specifications defined by the model builder. As already discussed in the previous section 2.6.2, model builders have to provide two subroutines *Calculate* and *HasConverged* to ensure proper model operation (Figure 13).



```
'Check whether the model has converged
'Called by the Run routine after each iteration
Function HasConverged(ByVal t As Integer, ByVal iteration As Integer)
  Dim r As Integer
  maxdif = 0
  sector = 0

  For r = 1 To UBound(gobpin_v(t), 1)
    If con_vec(r) > 0 Then
      difp(r) = 100 * (Abs(gobpin_v(t)(r) - con_vec(r)) / con_vec(r))
    End If
    If (difp(r) > 0.1) Then
      sector = r
      Exit For
    End If
    If (difp(r) >= maxdif) Then
      maxdif = difp(r)
      sector = r
    End If
  Next
  HasConverged = r = UBound(gobpin_v(t), 1) + 1
  If sector > 0 Then
    modelSheet.Cells(t - startYear + 7, 2) = sector
    modelSheet.Cells(t - startYear + 7, 3) = difp(sector)
  End If

  'to be activitated only if there is no covergence criteria
  'HasConverged = True
End Function

'Main calculation routine
'Called by the Run routine on each iteration
Sub Calculate(ByVal t As Integer, ByVal iteration As Integer)
  '--------------------------------------------------------------
  'Output in "Model" worksheet for year and interation
  '--------------------------------------------------------------
  modelSheet.Cells(t - startYear + 7, 1) = t & ", " & iteration
  '--------------------------------------------------------------
```

Figure 13: *Calculate* and *HasConverged* subroutines

Source: Own illustration.

The equation system consists of (mostly algebraic) statements which are processing the variables of the data set. Each of these variables is a time series, thus the first index of such a variable indicates the year *t* for which the data is read or written, e.g.

Labor force for year *t*: `lf_s(t)`

Elements of time series of vectors and matrices are accessed by providing the appropriate row (and column) indices. Examples:

- World market prices by commodities, row 6: world market price for crude oil, in 2015:
  `wmp_v(2015)(6)`
- Domestic input coefficients nominal for year t at row r and column c:
  `dicn_m(t)(r, c)`

In order to protect historical data from being overwritten, model builders should use the last data constants as shown in the following example:

```
If t > empl_v_lastData Then
      'some calculation
End If
```

By using these constants, model builders ensure proper model operation even after data updates as long as the data set properties have been carefully updated as well.

By providing the full set of equations in readable form, model issues can be tracked down much easier than in systems using so-called solvers: The model builder can set a breakpoint in the source and let the integrated debugger stop execution at any time. Once execution is paused, the current values of all variables may be inspected which greatly simplifies the process of model debugging.

Although it is technically possible to embed the full model code in the *Model* module, model builders are advised to logically separate the three E's of an E3 model by using modules. To keep the *Model* module clean and lean, it might be useful to also separate model-specific generic functions and store them in a separate module, e.g.,
ModelTools. One prominent example are aggregate routines for vectors / matrices.

The content of the different modules is discussed in chapter 3.

2.7 Regression analysis with “R”

All behavioral parameters of the model are estimated econometrically, and different specifications of the regression equations are tested against each other, which gives the model an empirical validation.

Although Excel provides basic routines for estimation, usage of a dedicated external application is preferrable for the following reasons:

- Regression equations cannot be edited easily (cell ranges need to be marked and routed to the appropriate Excel functions)
- Regression graphs need to be prepared separately
- Different combinations of equations referring to the same variable cannot be easily stored
- Large dataset cannot be conveniently handled

Using an external application for regression analysis entails two problems: First, the historical data stored in the *Values* sheet in the model workbook must be accessible by the regression application (e.g., by transforming data into another file format). Second, regression results, i. e. coefficients, need to be transferred back into the model. For an elaborated model such as e3.mn, this cannot be efficiently handled manually. The DIOM-X modelling framework provides the necessary tools to tackle both problems automatically.

The software “R”[2] is widely used for all kinds of data-related research. Besides the rich set of available libraries, one of its main advantages is that it is an open-source application which may be used free of charge even in non-academic environments.

This section explains how “R” can be integrated into the model building process. It is assumed that the reader has basic knowledge in operating “R” and performing Ordinary Least Square (OLS) regressions with it. Furthermore, the “R” application must be already installed on the computer. It is available free of charge from

[2] “For an introduction in “R”, please refer to various online documentations (e.g. https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf).

https://www.r-project.org/

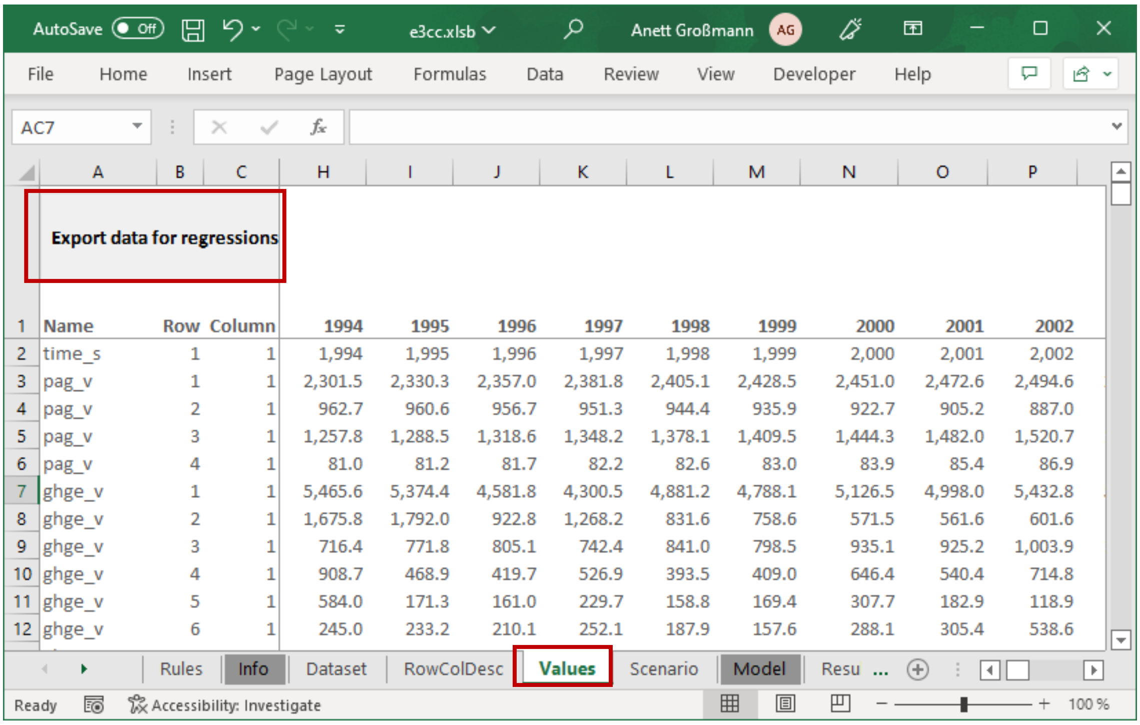
“R” does not come with a developer-friendly user interface; it is therefore assumed that the addon software *R Studio* available free of charge from

https://www.rstudio.com/

is installed as well.

A very common estimation approach is the OLS method which yields a linear regression model by minimizing the sum of squared residuals, i. e. the difference between realized and predicted data points for the past. Statistical test measures were included in the evaluation of single regression equations. For individual variables the corresponding t-statistics are of major importance. These and their associated p-values allow for an evaluation of the statistical significance of the variable in question with respect to its impact on the dependent variable. The R^2, the so-called coefficient of determination, can be used to judge the quality of the regression model. It shows the proportion of variance of the dependent variable that is explained by the explanatory variables. In general, it holds that the higher the R^2, the higher the explanatory value of the model is. Different regression equations can thus be compared using the R^2 (or in the case of more than one explaining variable the adjusted R^2, which controls for the positive effect of additional explaining variables by imposing a penalty on the respective equation).

Before the OLS method can be applied, the e3.mn model data needs to be transferred to “R” first. Although “R” is capable of reading Excel files, it cannot easily interpret the e3.mn data set. “R” usually expects variables stored in columns which is not possible with larger datasets in Excel due to a strict limit in the maximum number of columns. To overcome this restriction, the DIOM-X framework provides an automatic conversion routine which exports the entire historical data set from the *Values* sheet into a format that can be conveniently read by “R” (and other statistical program packages such as EViews). The conversion process is triggered by clicking the *Export data for regressions* button (see Figure 14 and also section 2.4.3).

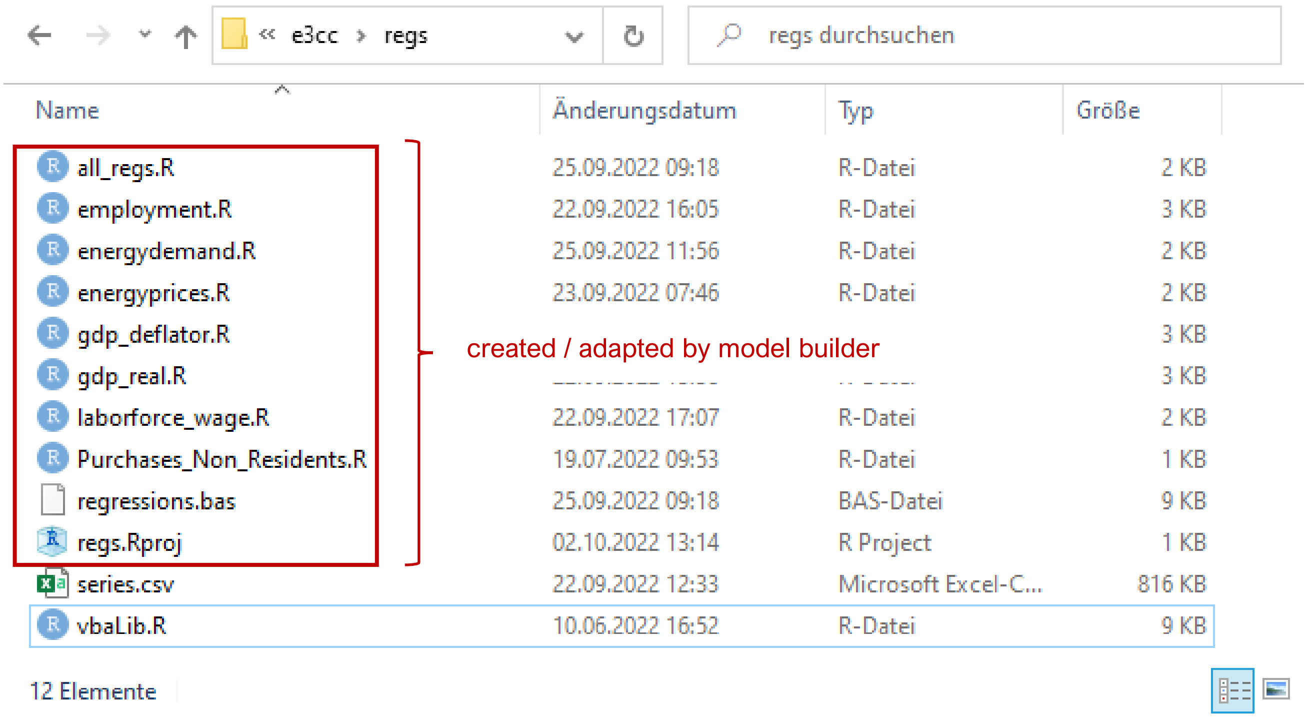


	A	B	C	H	I	J	K	L	M	N	O	P
1	Name	Row	Column	1994	1995	1996	1997	1998	1999	2000	2001	2002
2	time_s	1	1	1,994	1,995	1,996	1,997	1,998	1,999	2,000	2,001	2,002
3	pag_v	1	1	2,301.5	2,330.3	2,357.0	2,381.8	2,405.1	2,428.5	2,451.0	2,472.6	2,494.6
4	pag_v	2	1	962.7	960.6	956.7	951.3	944.4	935.9	922.7	905.2	887.0
5	pag_v	3	1	1,257.8	1,288.5	1,318.6	1,348.2	1,378.1	1,409.5	1,444.3	1,482.0	1,520.7
6	pag_v	4	1	81.0	81.2	81.7	82.2	82.6	83.0	83.9	85.4	86.9
7	ghge_v	1	1	5,465.6	5,374.4	4,581.8	4,300.5	4,881.2	4,788.1	5,126.5	4,998.0	5,432.8
8	ghge_v	2	1	1,675.8	1,792.0	922.8	1,268.2	831.6	758.6	571.5	561.6	601.6
9	ghge_v	3	1	716.4	771.8	805.1	742.4	841.0	798.5	935.1	925.2	1,003.9
10	ghge_v	4	1	908.7	468.9	419.7	526.9	393.5	409.0	646.4	540.4	714.8
11	ghge_v	5	1	584.0	171.3	161.0	229.7	158.8	169.4	307.7	182.9	118.9
12	ghge_v	6	1	245.0	233.2	210.1	252.1	187.9	157.6	288.1	305.4	538.6

Figure 14: Values worksheet

Source: Own illustration

The generated file *series.csv* includes the entire historical database and is stored in the folder *regs*. The folder is also used to store the different estimation equation files (e.g., *gdp_real.R*) as well as the library *vbaLib.R* which provides helper functions for OLS estimation and translation of regression results back into the e3.cc model code (Figure 15).



Name	Änderungsdatum	Typ	Größe
all_regs.R	25.09.2022 09:18	R-Datei	2 KB
employment.R	22.09.2022 16:05	R-Datei	3 KB
energydemand.R	25.09.2022 11:56	R-Datei	2 KB
energyprices.R	23.09.2022 07:46	R-Datei	2 KB
gdp_deflator.R			3 KB
gdp_real.R			3 KB
laborforce_wage.R	22.09.2022 17:07	R-Datei	2 KB
Purchases_Non_Residents.R	19.07.2022 09:53	R-Datei	1 KB
regressions.bas	25.09.2022 09:18	BAS-Datei	9 KB
regs.Rproj	02.10.2022 13:14	R Project	1 KB
series.csv	22.09.2022 12:33	Microsoft Excel-C...	816 KB
vbaLib.R	10.06.2022 16:52	R-Datei	9 KB

12 Elemente

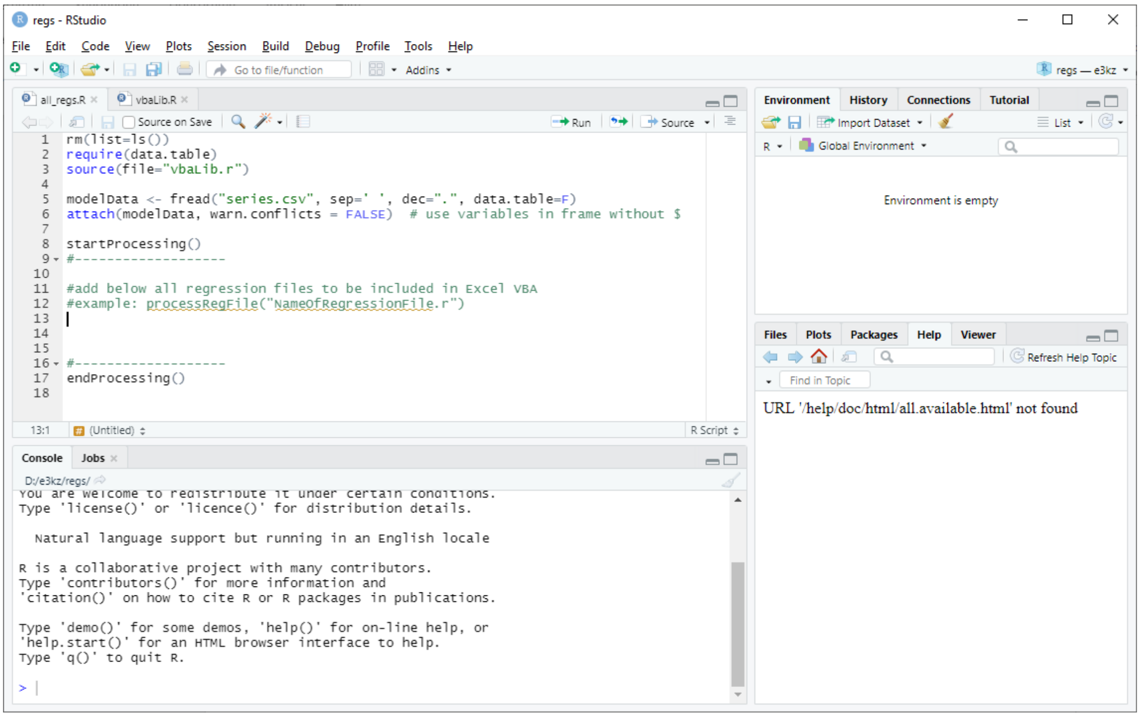
Figure 15: *regs* folder of e3.cc

Source: Own illustration

Performing regression analysis as part of an E3 model is an iterative process. Throughout model runtime, the model iteratively solves the underlying equation system by checking a certain convergence criterion which must be fulfilled every year. Such a system usually behaves very sensitive to changes in the regression parameters. Thus, it is common practice that several regression specifications need to be tested in the context of the equation system to ensure model stability.

For most estimations, historical data is limited to a time period of around 15 to 30 years. In view of the long-term modelling horizon up to 2050, particular attention must be paid to the sign and magnitude of the parameters for the explanatory variables, and, of course, to the overall stability of the model. Estimated parameters can be further verified by a comparison with other studies and / or model exercises. The model should also reflect expectations regarding, e.g., economic development and energy demand, identified in other studies.

To perform regressions in “R” based on the exported data set in *series.csv*, the project file needs to be opened by double-clicking on *regs.Rproj* in the *regs* directory of the model. If this file type is not associated with “R” Studio, the file needs to opened manually (see Figure 16).



```
rm(list=ls())
require(data.table)
source(file="vbaLib.r")

modelData <- fread("series.csv", sep=' ', dec=".", data.table=F)
attach(modelData, warn.conflicts = FALSE)  # use variables in frame without $

startProcessing()
#-------------------

#add below all regression files to be included in Excel VBA
#example: processRegFile("NameOfRegressionFile.r")



#-------------------
endProcessing()
```

```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

URL '/help/doc/html/all.available.html' not found

Figure 16: “R” graphical user interface

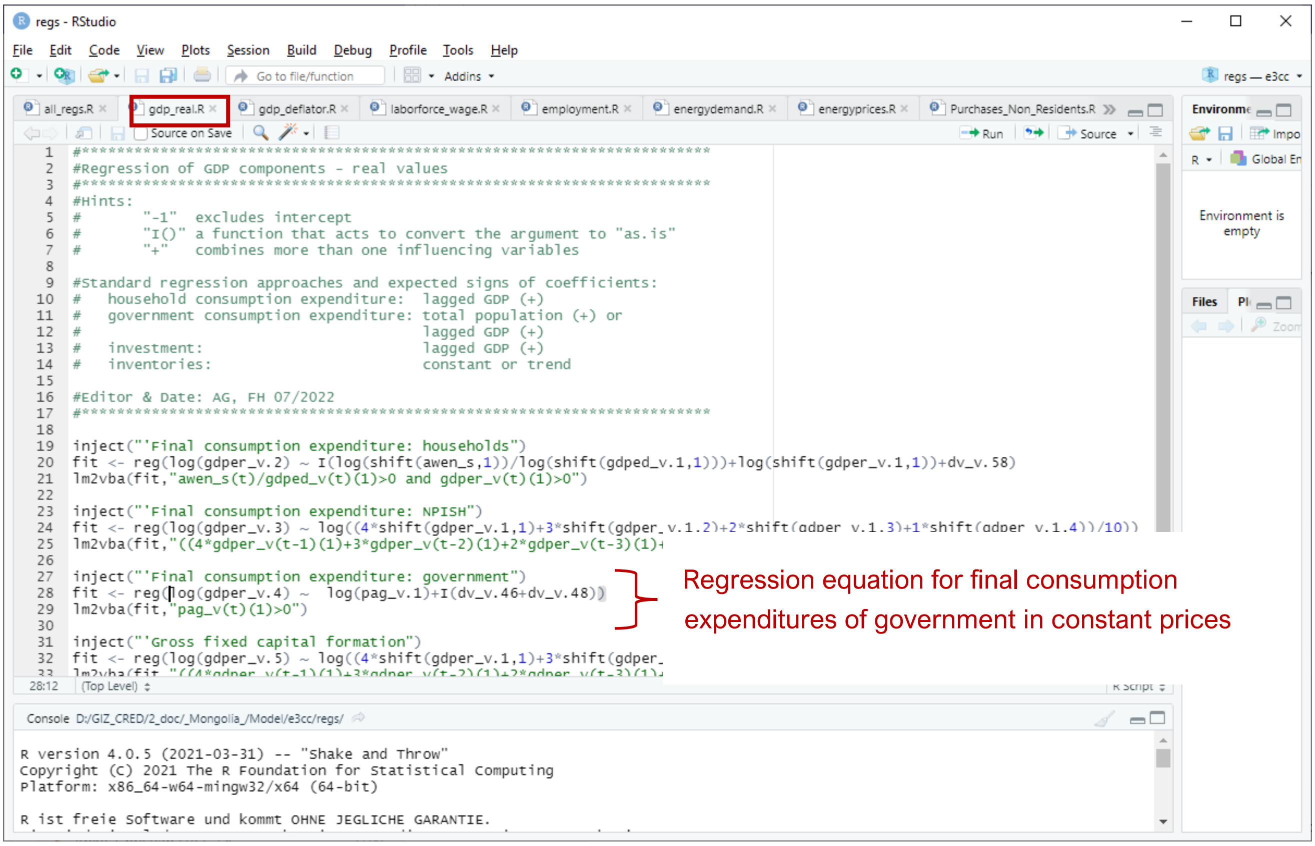
Source: Own illustration

The file *allRegs.R* is the master file which triggers all subsequent actions, i. e. loading of the data set and automatic translation of the estimation parameters into model code.

All regression files that need to be processed must be placed between the *startProcessing()* and *endProcessing()* function calls which ensure proper translation from “R” to VBA code.

This file needs to be executed if the dataset or any of the regression equations have changed to ensure that the latest changes will be reflected in the model.

The e3.mn model incorporates several variables (e.g., GDP components in constant prices and respective deflators, employment etc.) that are estimated. For each of them a “R” script is created including the specifications of the regression equations (Figure 17). Each regression file is named accordingly to the content. For example, in case of GDP components in constant prices, the regression file is named *gdp_real.R* and stored in the folder *regs*. This name will show up in the VBA code as well.



```
#***************************************************************************
#Regression of GDP components - real values
#***************************************************************************
#Hints:
#       "-1"  excludes intercept
#       "I()" a function that acts to convert the argument to "as.is"
#       "+"   combines more than one influencing variables

#Standard regression approaches and expected signs of coefficients:
#   household consumption expenditure:  lagged GDP (+)
#   government consumption expenditure: total population (+) or
#                                       lagged GDP (+)
#   investment:                         lagged GDP (+)
#   inventories:                        constant or trend

#Editor & Date: AG, FH 07/2022
#***************************************************************************

inject("'Final consumption expenditure: households")
fit <- reg(log(gdper_v.2) ~ I(log(shift(awen_s,1))/log(shift(gdped_v.1,1)))+log(shift(gdper_v.1,1))+dv_v.58)
lm2vba(fit,"awen_s(t)/gdped_v(t)(1)>0 and gdper_v(t)(1)>0")

inject("'Final consumption expenditure: NPISH")
fit <- reg(log(gdper_v.3) ~ log((4*shift(gdper_v.1,1)+3*shift(gdper_v.1,2)+2*shift(gdper_v.1,3)+1*shift(gdper_v.1,4))/10))
lm2vba(fit,"((4*gdper_v(t-1)(1)+3*gdper_v(t-2)(1)+2*gdper_v(t-3)(1)+

inject("'Final consumption expenditure: government")
fit <- reg(log(gdper_v.4) ~  log(pag_v.1)+I(dv_v.46+dv_v.48))
lm2vba(fit,"pag_v(t)(1)>0")

inject("'Gross fixed capital formation")
fit <- reg(log(gdper_v.5) ~ log((4*shift(gdper_v.1,1)+3*shift(gdper_
lm2vba(fit,"((4*gdper_v(t-1)(1)+3*gdper_v(t-2)(1)+2*gdper_v(t-3)(1)+
```

Regression equation for final consumption expenditures of government in constant prices

```
Console D:/GIZ_CRED/2_doc/_Mongolia_/Model/e3cc/regs/

R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
```

Figure 17: *Regression examples*

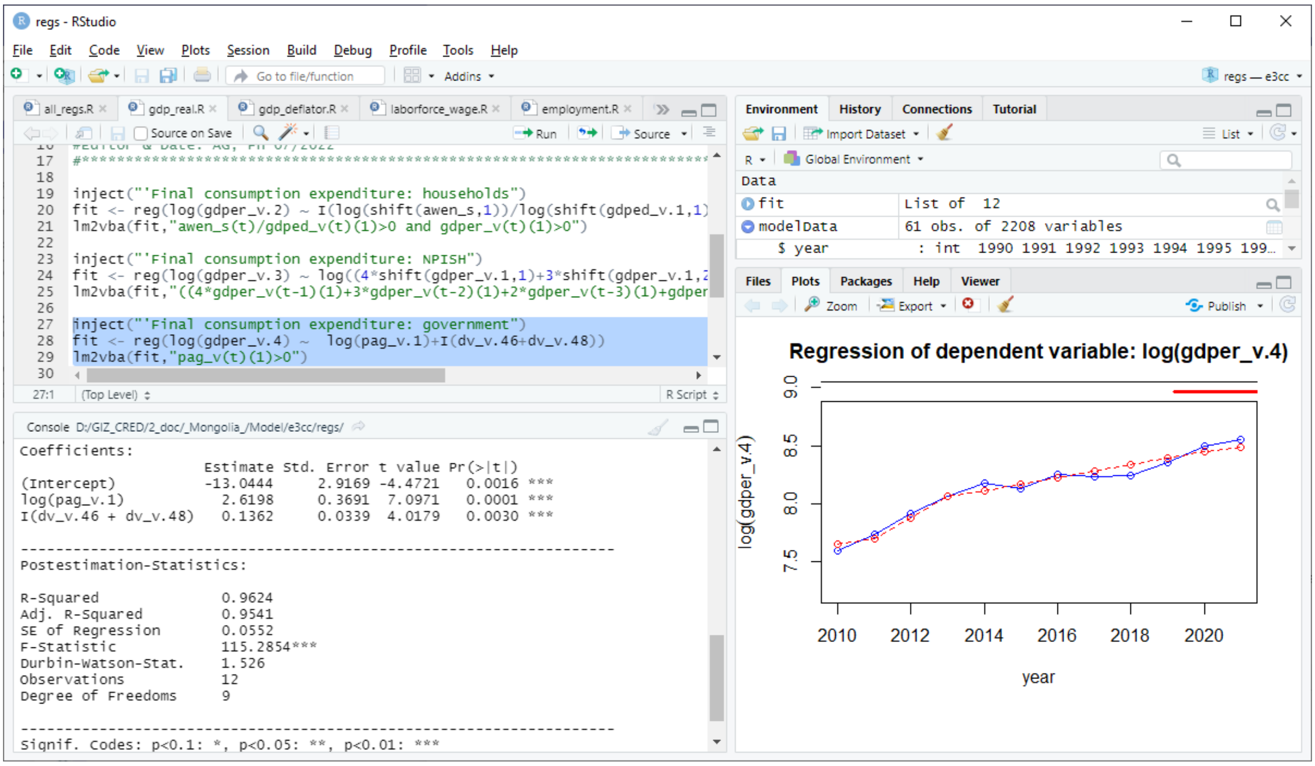
Source: Own illustration

Each regression equation needs the following three statements

- *inject()*: This statement injects a comment into the VBA model code and should be used with each regression to explain the purpose/content of the regression
- *fit <- reg()*: specification of the regression equation (Which explanatory variable(s) should be included? Linear regressions for linear relationships or linear regressions with logarithmic transformations for non-linear relationships?). The underlying time span for the regression can be set with function parameters “startyear” and “endyear”.
- *lm2vba(fit,)*: This function automatically translates the regression results into VBA code. All results are stored in a single file named *regressions.bas* which later will be inserted into the model.

The regression function *reg()* is a custom “R” function that computes regressions based on the entered formula and prints the results to the console output pane. It plots the actual values of the dependent variable against the fitted values in the ”Plots” pane of “R” Studio. The function has three inputs: *rformula*, *startyear*, and *endyear*. *Rformula* is the input for the formula of the regression equation, *startyear* and *endyear* are user-defined values for the underlying time span for the regression estimation. While *rformula* must be specified in any case, the user is free to omit *startyear* or *endyear*. If omitted, the function automatically regresses the given variables for their largest common time span.

In the console output pane (see Figure 18), various regression specifications can be tested. As already mentioned, a regression summary containing various statistical tests and key figures will be printed and plotted automatically. Various “R” packages provide further options to analyze and visualize the results which are not part of that manual but can be explored in “R” blogs and books about “R”.



```
#*********************************************************************

inject("'Final consumption expenditure: households")
fit <- reg(log(gdper_v.2) ~ I(log(shift(awen_s,1))/log(shift(gdped_v.1,1)
lm2vba(fit,"awen_s(t)/gdped_v(t)(1)>0 and gdper_v(t)(1)>0")

inject("'Final consumption expenditure: NPISH")
fit <- reg(log(gdper_v.3) ~ log((4*shift(gdper_v.1,1)+3*shift(gdper_v.1,2
lm2vba(fit,"((4*gdper_v(t-1)(1)+3*gdper_v(t-2)(1)+2*gdper_v(t-3)(1)+gdper

inject("'Final consumption expenditure: government")
fit <- reg(log(gdper_v.4) ~  log(pag_v.1)+I(dv_v.46+dv_v.48))
lm2vba(fit,"pag_v(t)(1)>0")
```

```
Data
fit                List of  12
modelData          61 obs. of 2208 variables
  $ year           : int  1990 1991 1992 1993 1994 1995 199…
```

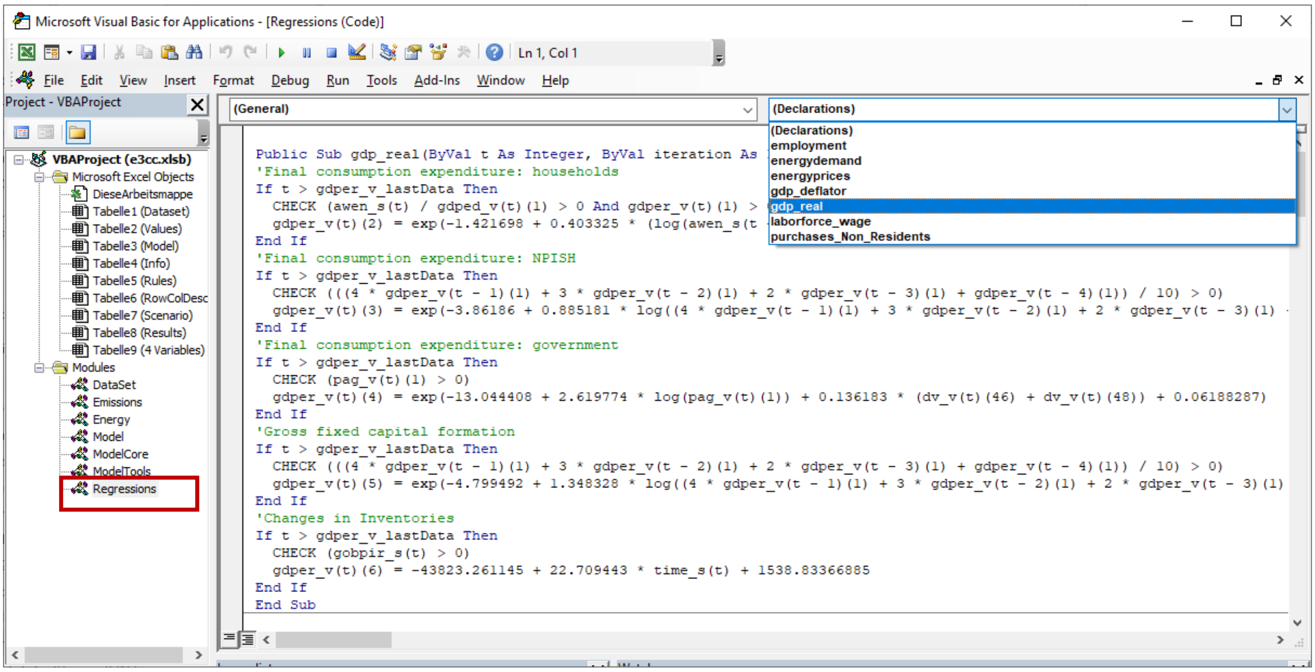
Console D:/GIZ_CRED/2_doc/_Mongolia_/Model/e3cc/regs/

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -13.0444     2.9169 -4.4721   0.0016 ***
log(pag_v.1)           2.6198     0.3691  7.0971   0.0001 ***
I(dv_v.46 + dv_v.48)   0.1362     0.0339  4.0179   0.0030 ***

--------------------------------------------------------------------
Postestimation-Statistics:

R-Squared              0.9624
Adj. R-Squared         0.9541
SE of Regression       0.0552
F-Statistic            115.2854***
Durbin-Watson-Stat.    1.526
Observations           12
Degree of Freedoms     9

--------------------------------------------------------------------
Signif. Codes: p<0.1: *, p<0.05: **, p<0.01: ***
```

Figure 18: Overview of statistical test results and plot of actual and fitted dependent variable using reg()

Source: Own illustration

After all model variables are estimated and final regression equations are formulated in the respective “R” files, these file names must be included in *allRegs.R* for automatic translation into VBA code. Once this file is executed (Ctrl+Shift+Enter), the file *regressions.bas* is created including all regression equations with their parameters. This file must be integrated into VBA (Figure 19).



```
Public Sub gdp_real(ByVal t As Integer, ByVal iteration As
'Final consumption expenditure: households
If t > gdper_v_lastData Then
  CHECK (awen_s(t) / gdped_v(t)(1) > 0 And gdper_v(t)(1) >
  gdper_v(t)(2) = exp(-1.421698 + 0.403325 * (log(awen_s(t
End If
'Final consumption expenditure: NPISH
If t > gdper_v_lastData Then
  CHECK (((4 * gdper_v(t - 1)(1) + 3 * gdper_v(t - 2)(1) + 2 * gdper_v(t - 3)(1) + gdper_v(t - 4)(1)) / 10) > 0)
  gdper_v(t)(3) = exp(-3.86186 + 0.885181 * log((4 * gdper_v(t - 1)(1) + 3 * gdper_v(t - 2)(1) + 2 * gdper_v(t - 3)(1)
End If
'Final consumption expenditure: government
If t > gdper_v_lastData Then
  CHECK (pag_v(t)(1) > 0)
  gdper_v(t)(4) = exp(-13.044408 + 2.619774 * log(pag_v(t)(1)) + 0.136183 * (dv_v(t)(46) + dv_v(t)(48)) + 0.06188287)
End If
'Gross fixed capital formation
If t > gdper_v_lastData Then
  CHECK (((4 * gdper_v(t - 1)(1) + 3 * gdper_v(t - 2)(1) + 2 * gdper_v(t - 3)(1) + gdper_v(t - 4)(1)) / 10) > 0)
  gdper_v(t)(5) = exp(-4.799492 + 1.348328 * log((4 * gdper_v(t - 1)(1) + 3 * gdper_v(t - 2)(1) + 2 * gdper_v(t - 3)(1)
End If
'Changes in Inventories
If t > gdper_v_lastData Then
  CHECK (gobpir_s(t) > 0)
  gdper_v(t)(6) = -43823.261145 + 22.709443 * time_s(t) + 1538.83366885
End If
End Sub
```

Figure 19: *Regression* module

Source: Own illustration

The necessary steps to import the module containing the regression equations and parameters are as follows:

1. Select the *Regressions* module in the left pane of the VBA programming environment:
2. Right-click and select *Remove Regressions.* Removing the old version is unfortunately necessary because Excel will not update the existing version but create another module with a different name.

Right-click again, select *Import file* and import the file *Regressions.bas* from the folder *regs* where the workbook is stored.

3 E3.cc – a national economy – energy– emission model: The case of Mongolia (e3.mn)

The E3 model is a macro-econometric Input-Output (IO) model covering the structure of the Mongolian economy and its main connections to the environment, i.e. the use of energy resources and the release of CO_2 emissions into the environment. This integrated modeling approach of the three Es in one model assures a consistent view of possible transition pathways in Mongolia. It enables the user to calculate impacts for the whole economy (e.g., GDP) and single economic sectors (e.g., employment in agriculture) as well as to draw conclusions on social balance and environmental benefits.

Figure 20 briefly shows the three model parts and their interrelations. Number 1 indicates the economic core of the E3 model, number 2 shows the energy module and number 3 the emission module.

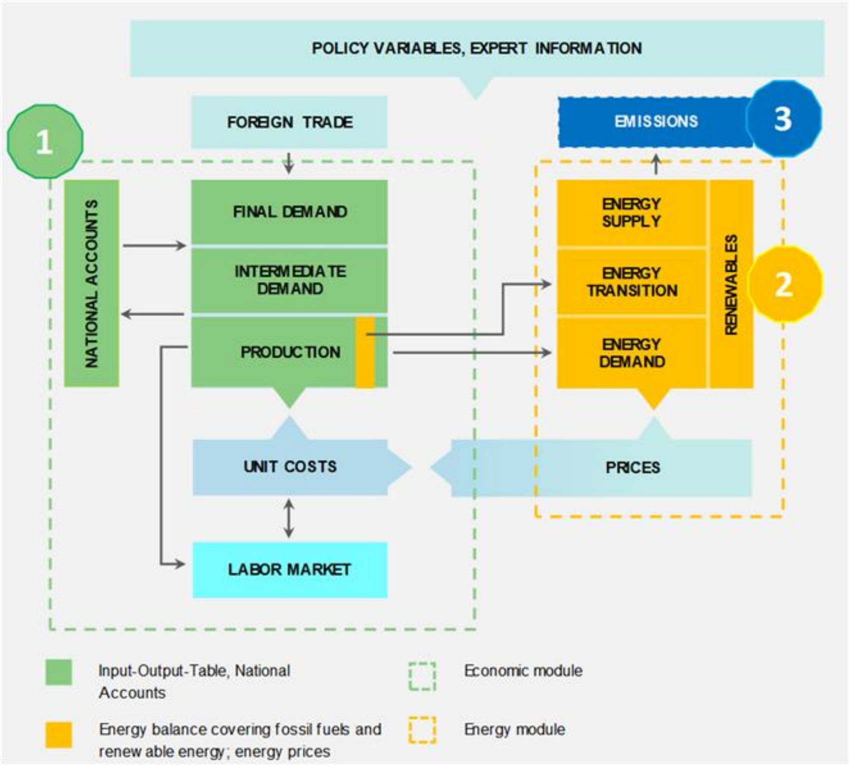


Figure 20: E3 model at a glance

Source: own illustration by GWS

Each of the three parts are based on a rich and up-to-date, country-specific dataset given as time series based on annual data. The economic part of the model is built from an IOT[3] and the system of national accounts (macroeconomic data) plus population and labor market data such as employment. Energy balances, which include energy supply, transformation and (final) energy demand, are at the center of the energy part. Based on these data in physical units, combustion-related CO_2 emissions can be directly calculated using fixed emission factors.

The E3 model for Mongolia draws on the observed and theoretical relationships represented as a set of mathematical equations which describe the workings of the Mongolian economy in a simplified way. Apart from calculations given by definition such as the GDP, which is the sum of consumption, investment and foreign trade, econometric methods and expert knowledge are used to derive future developments from historical data.

The model results – which depict a possible future pathway of the country's economy – depend on some exogenous inputs (e.g., population, world market prices) as well as the modelled relationships within the three model parts and between them.

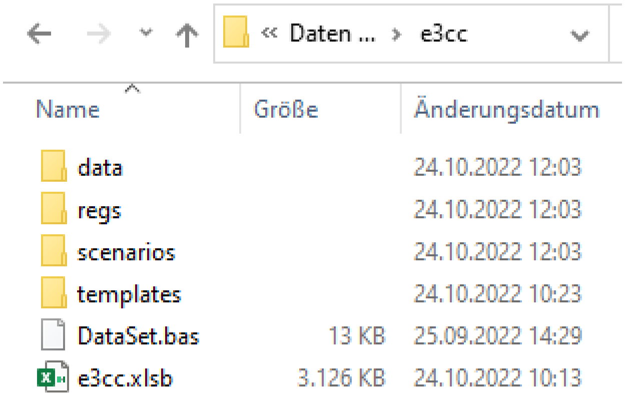
Once the E3 model is fully established, it can be applied for scenario analyses which is the main purpose of the model. Then, for example, exogenous variables can be adjusted to discover how it impacts the macroeconomy and its sectors, employment, and environmental indicators.

[3] For more details on input-output analysis and the system of national accounts, please refer to EC, 2019 and UN, 2018.

The details of each of the three model parts are depicted in the subsequent sections. In the sections “Implementation into DIOM-X”, code snippets are explained using specific examples. The full model code is part of the *e3cc.xlsb* workbook.

3.1 Folder structure of e3.cc

It is meaningful for model builders to create a model folder structure that reflects the main model building steps[4] (Figure 21) although model users finally only need the workbook *e3cc.xlsb* to run simulations.



Name	Größe	Änderungsdatum
data		24.10.2022 12:03
regs		24.10.2022 12:03
scenarios		24.10.2022 12:03
templates		24.10.2022 10:23
DataSet.bas	13 KB	25.09.2022 14:29
e3cc.xlsb	3.126 KB	24.10.2022 10:13

Figure 21: Folder structure of e3.cc

Source: Own illustration

The folder *data* contains the historical data collected from official statistics[5] and the folder “OwnCalculations” which includes necessary own calculations and data transformation based on official statistics (for more details please refer to the sections 3.2.2 and 3.3.2).

With large data sets, it is useful to create subfolders to group the data, e.g., by institutional data providers (e.g., ADB, ILO, World Bank) and – if necessary – also by content (e.g. GDP, LABOUR). The data sets specified in each E3 subsection (see for example section 3.2.2) can be easily found according to the subfolder structure organized by data providers.

All e3.cc data are included in the *Values* worksheet. It is only necessary to store original data files and transformed data (see folder “OwnCalculations”) in the *data* folder if these data are linked to the *Values* sheet.

In the folder *regs,* all files needed for the regression analysis are stored. Regressions are done in an external regression program such as “R”, but need of course the historical data set stored in the file *series.csv*. Also, the tested regression approaches and resulting final regression equations including the coefficients (*regressions.bas*) are stored in that folder so that the model builder can easily insert these into the VBA code.

The folder *scenarios* contains all scenarios (including tweaks and results) and an evaluation file for comparing two scenarios (*_Compare2Scenarios_.xlsb*) – in particular relevant for scenario resp. impact analysis – and to evaluate a single scenario (*_Eval1Scenario_.xlsb*).

The folder *templates* includes an MS Excel template with tables and graphs which are used by default for scenario evaluation.

[4] The model code is stored in the *e3cc.xlsb* workbook and therefore no dedicated folder exists.

[5] Using data from official statistics supports transparency, ensures regular data updates, assures high degree of quality.

3.2 Economic model

3.2.1 GENERAL OVERVIEW

The core of the E3 model is a macro-econometric (or dynamic) IO model. The underlying modelling approach is based on so-called Post Keynesian theory: Economic growth is determined from a demand-side perspective. Thus, economic activity is driven by expenditure decisions for consumption purposes of households, investments by companies or export demand from abroad. These are a key determinant of demand, production, and employment. The model considers the quantity and price relationships at macro level and (scarce) input factors such as labor or energy. While assumed export and import prices rely on exogenous third-party projections (e.g., world market price projections from the World Bank, IMF and IEA), household and government consumption as well as investments are (regression) functions of other model variables and thus determined endogenously within the model. These aggregated macro values are then distributed to the level of economic sectors by using their shares of the respective variable given from the final demand in the IOT (top-down approach).

The IOT is the main component of this macro-econometric IO model (Figure 22). It shows the key industries, sales, and cost structure (purchases by economic sectors) of the country’s economy. The cost structure is represented by the need for input factors (material and energy input) and labor (primary inputs such as compensation for employees which is part of the value added) of each economic sector. Prices are derived by using a unit cost approach considering the cost components. The prices in turn are influencing factors in the regression functions for the GDP components (consumption, investment, etc.) mentioned previously.

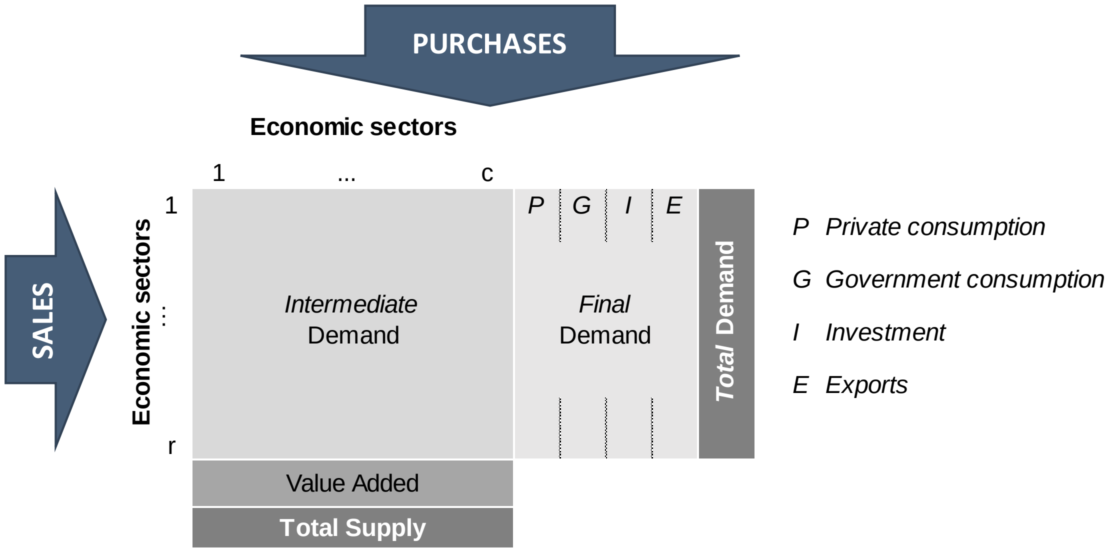


Figure 22: Schematic illustration of an IOT

Source: own representation

As prices reflect the cost situation, they do not necessarily and smoothly balance supply and demand. In that sense, the model does not adhere to the equilibrium economics school (unlike the family of CGE models; for conceptual differences see for example Pollitt et al. 2019). Imbalances between supply and demand are more likely to be offset by (price-dependent) demand-driven effects rather than by price effects.

The input coefficients, representing how much and what kind of inputs are needed to produce one unit of a certain good or service, are derived from the IOT and reflect the production technologies. Technological change alters the technology and thus the production structure.

The IO model shows the economy-wide (direct and indirect) effects on production caused by demand changes (Leontief multiplier) by applying the so-called Leontief-equation, which answers the questions “How does production react when final demand changes? Which industries are affected by these changes?”. Impacts of additional investments (which are part of the final demand, c.f. Figure 22) in irrigation systems and any other investments are reflected here. Depending on the import-dependencies, production in the respective industries increases (investment accelerator).

The income-induced effects can be seen in combination with the modeling of employment (employment and income multiplier). Labor demand is influenced by the economic activity in the various sectors. While a higher production level creates additional jobs, a raising wage rate tends to lower the employment. Wage rates might be influenced by a shortage of labor, inflation and overall labor productivity or are set by the government.

Figure 23 shows the economic model and the modelled interrelations at a glance. The economic model contexts are captured via identities (e.g., in the IO context; solid lines), behavioral equations that are empirically validated (dashed line) and a few exogenously given variables (yellow marked) such as population. However, the specification of the model is not finished with the estimation of single equations. The complete, non-linear, interdependent model equation system is solved iteratively for each year. The iteration process finishes once a given criterion is fulfilled. This criterion has to be an endogenously calculated model variable which for e3.cc is the output by economic sectors. As long as the model has not converged, all model equations are recalculated for the current year (cf. 0 and 0). Afterwards, all equations are solved year by year until the end of simulation period which was set at 2050.

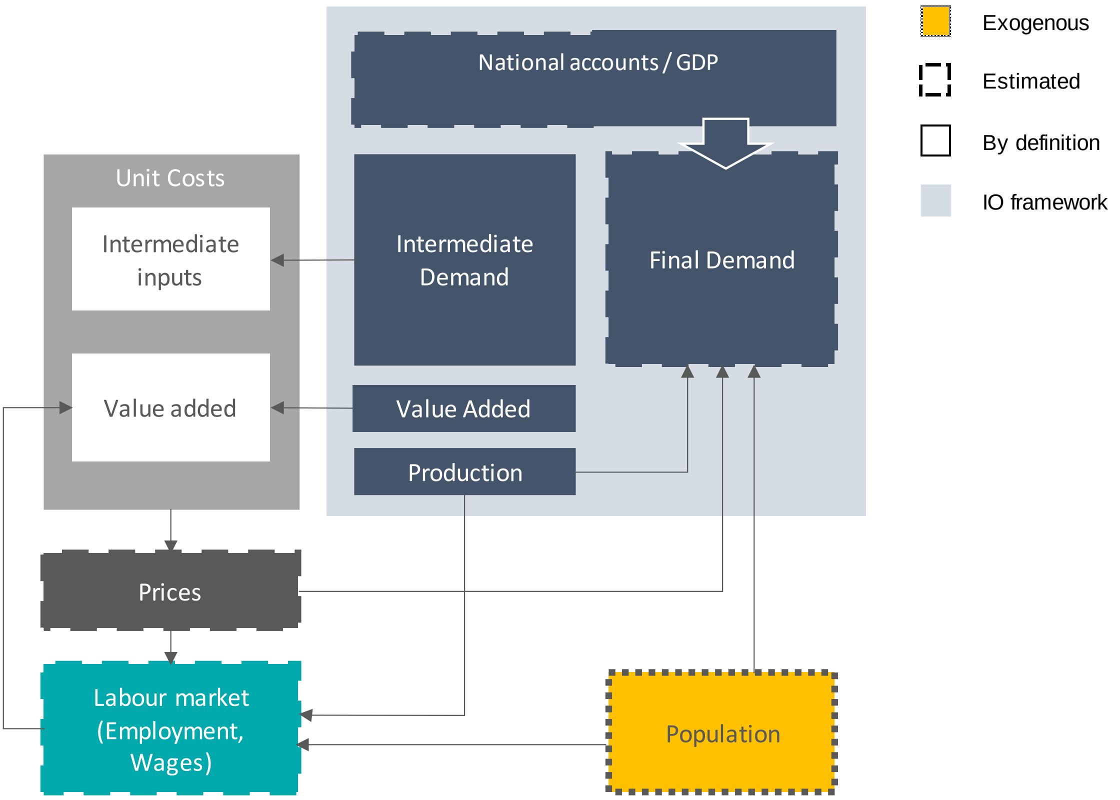


Figure 23: Simplified illustration of the macro-econometric IO model

Source: Own illustration

3.2.2 DATASET

The World Bank, ADB, ILO and UN and other international data providers offer rich and up-to-date data on an annual basis for most of the countries worldwide which is needed for developing country models according to the model prototype. If no data is available for essential indicators from these sources, data from the national statistical office is used, e.g., producer price indices by economic sectors from NSO in Mongolia.

The economic model for Mongolia is based on the data set shown in the following table.

Table 2: Data set for the economic model

Description	Time frame	Sources
MACROECONOMIC AND SOCIOECONOMIC DATA		
Population ▪ Total ▪ By age groups	▪ 1990-2050	▪ UN DESA
Labor market ▪ Employment ▪ Unemployment ▪ Labor force ▪ Average wage per employee	▪ 2000-2021	▪ ADB
GDP components (expenditure approach ▪ Current prices ▪ Constant prices ▪ Deflator	▪ 2000-2021	▪ ADB
Exchange rate	▪ 1991-2021	▪ World Bank
SECTORAL DATA		
Symmetric IOT ▪ Intermediate flow matrix ▪ Final demand matrix	▪ 2000-2020	▪ ADB
Gross production ▪ Current prices ▪ Price indices	▪ 2000-2020 ▪ 2010-2021	▪ ADB ▪ NSO
Intermediate inputs	▪ 2000-2020	▪ ADB
Value added and components, e.g. ▪ Compensation of employees	▪ 2000-2020	▪ ADB
Import price indices	▪ 1990 (2011)-2021	▪ IMF
Labor market ▪ Employment	▪ 2000-2021	▪ ADB

Source: Own illustration

Certain data needs and data properties must be fulfilled to build an E3 model:

- Annual data, provided in MS Excel format,
- Data as (sufficient) time series are important for estimations,
- Data must be consistent,
- Data needs must be fulfilled (e.g., an IOT is a prerequisite for an IO model) and
- Sufficient sectoral detail.

To fulfil these data needs and properties, data transformations and additional calculations might be necessary to derive missing data. For simplicity and traceability, these calculations are performed in MS Excel and results are stored in the folder *OwnCalculations*.

Data transformation in the economic part of the model comprises:

(1) Combination of time series of constant GDP components from ADB and the NSO to calculate GDP deflators by components,

(2) Determination and reindexing of producer price indices for 35 economic sectors (2015=100).

(3) Calculation of a time series of gross output by economic sectors in real terms as well as gross output (nominal, real, price indices) for 5 and 12 economic sectors,

(4) Determination of import deflators for 35 economics sectors,

(5) Calculation of imported input coefficients and

(6) Aggregation of population by age groups.

3.2.3 IMPLEMENTATION INTO DIOM-X

Once the historical data set is established, relationships between model variables must be defined. The detailed illustration of the macro-econometric IO model as implemented into the model building framework DIOM-X shows Figure 24. The description of the model variables is given in the worksheet *Dataset* in the workbook *e3cc.xlsb*.

The projection of the values of each model variable is either given exogenously (following other projections (yellow color), is specified endogenously in regression equations (dashed lines) or is given by definition (solid lines).

All model equations which are not estimated and related to this subsection are part of the *Model* module in VBA. Regression equations are part of the *Regressions* module.

1 Economic growth is determined first at macro level by modelling quantity and price relationships and then broken down to the economic sectors (top-down approach). The final demand components – either price-adjusted values or nominal values as well as deflators – are determined at macro level.

The **price-adjusted GDP components** such as the final consumption expenditures of private households *gdper_v(t)(2)*[6], expenditures of non-profit institutions serving households (NPISH) *gdper_v(t)(3)* as well as gross fixed capital formation *gdper_v(t)(5)* are estimated[7] as a weighted moving average (WMA) of price-

[6] Values in "()" indicate the row (and column) elements of a vector (labelled with r) or a matrix (labelled with r and c) which could be the number of given GDP components, or the number of economic activities given for an economic variable such as employment. For the description of rows and / or columns, please refer to the worksheet *RowColDesc* in the workbook e3cc.xlsb.

Grouping of single variables to a vector (having more than one row or column element) or a matrix (having more than one row and column) result in a clearer structure and less single variables. The elements are numbered from 1 to n according to their structure. For example, all GDP components in current prices are part of one vector named *gdpen_v(t)* with seven elements. Employment by economic activity is also grouped as a vector named *empl_v(t)* with as many elements as economic activities.

[7] All estimations are done in "R" and results are then inserted into the VBA code. In section 2.7, the steps to be taken to implement the regression results into VBA are described.

adjusted GDP *WMA(gdper_v(t)(1))*. The “R” code snippet shows how Formula (1) is translated to “R” code and how it appears in the VBA model code.

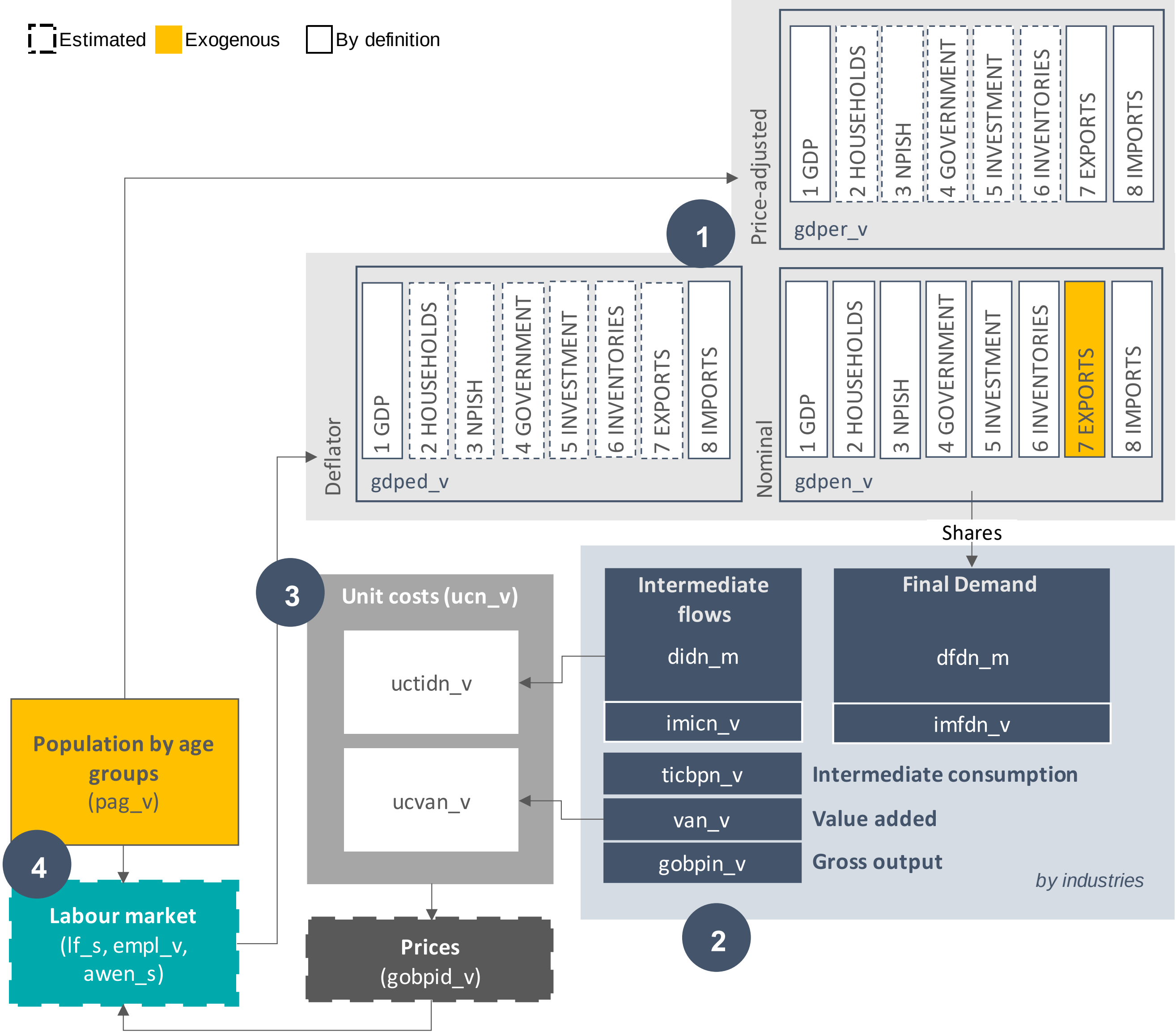


Figure 24: Simplified structure of the economic model

Source: Own illustration

Price adjusted government expenditures *gdper_v(t)(4)* depends on the development of total population *pag_v(t)(1)* assuming that the government expenditures increase with population. Changes in inventories *gdper_v(t)(6)* follow a time trend.

$$(1) \qquad gdper\_v(t)(r) = f\big(WMA(gdper\_v(t)(1))\big) \qquad r \in (2 \ldots 3, 5)$$

$$gdper\_v(t)(4) = f(pag\_v(t)(1))$$

“R” code

```
inject("'Final consumption expenditure: households")
fit <- reg(log(gdper_v.2) ~ log((4*shift(gdper_v.1,1)+3*shift(gdper_v.1,2)+
                                2*shift(gdper_v.1,3)+1*shift(gdper_v.1,4))/10))
lm2vba(fit,"((4*gdper_v(t-1)(1)+3*gdper_v(t-2)(1)+2*gdper_v(t-3)(1)+
              gdper_v(t-4)(1))/10)>0")
...
```

VBA regression function

```
Public Sub gdp_real(ByVal t As Integer, ByVal iteration As Integer)
  'Final consumption expenditure: households
  If t > gdper_v_lastData Then
      CHECK (((4 * gdper_v(t - 1)(1) + 3 * gdper_v(t - 2)(1) +
               2 * gdper_v(t - 3)(1) + gdper_v(t - 4)(1)) / 10) > 0)
      gdper_v(t)(2) = exp(0.707046 + 0.89078 * log((4 * gdper_v(t - 1)(1) +
                      3 * gdper_v(t - 2)(1) + 2 * gdper_v(t - 3)(1) +
                      1 * gdper_v(t - 4)(1)) / 10) + -0.05725106)
  End If
  ...
End Sub
```

The regression equation(s) are included as VBA functions in the model code. The name of the function *gdp_real* is identical to the name of the file including the estimation equation(s) (see *regs* folder).

Call of regression function in the model code

```
gdp_real t, iteration
```

The **deflators for the GDP** components gdped_v(t)(r) – except for imports – are estimated as a dependent variable of the import deflator *gdped_v(t)(8)* and / or the average wage per employed person *awen_s(t)*. Both variables are important factors for the domestic price development.

$$(2) \qquad gdped\_v(t)(r) = f(awepn\_s(t), gdped\_v(t)(8)) \qquad\qquad r \in (2 \ldots 5, 7)$$

Import deflators by economic sectors *impd_v(t)* follow the exogenously given world market prices *wmp_v(t)* by selected commodities (see *Scenario* worksheet) or the overall import deflator.

Nominal GDP components *gdpen_v(t)* are basically derived from corresponding deflators *gdped_v*(t) and price-adjusted values *gdper_v(t)*. By using the *for…next* loop, the following statement is executed as long as the condition *(r=2 to 6)* is met. Thus, the calculation is done for consumption of private households (r=2) and NPISH (r=3), government consumption (r=4), gross fixed capital formation (r=5) and changes in inventories (r=6).

```
For r = 2 To 6
  gdpen_v(t)(r) = gdper_v(t)(r) * gdped_v(t)(r) / 100
Next
```

An exception exists for nominal exports *gdpen_v(t)(7)* which are given exogenously by using the tweak mechanism (see *Scenario* worksheet and cf. section 2.4.5) and imports which are calculated bottom-up within the IO framework.

```
Tweak t, gdpen_v_lastData, gdpen_v, „gdpen_v“
```

2 The transition from the domestic concept (GDP expenditure approach) to the national concept (total final demand *tfdbpn_m(r,c)*) occurs under consideration of direct purchases abroad by residents *dpar_s(t)* and non-residents *pdtnr_s(t*).

The total values of each final demand component are distributed to the single industries by using constant shares *dfds_m(r,c)* calculated until the last historical year 2020. The respective *lastData* value is retrieved by using the *lastData* function (see VBA code below). Domestic final demand is given as a matrix and thus has rows r and columns c. For the columns 1 to 6 of the matrix *dfdn_m(t)*, the shares for the row elements 1 to 35 are calculated as share of their overall values given in basic prices *tfdbpn_v(t)*. By using the VBA *Ubound* function, the upper bound of rows and / or columns in an array (here for the variable *dfdn_m(t)* which is specified in the worksheet Dataset) is returned.

```
If t <= dfdn_m_lastData Then
  For c = 1 To 6
        For r = 1 To Ubound(dfdn_m(t), 1)
           dfds_m(t)(r, c) = dfdn_m(t)(r, c) / tfdbpn_v(t)(c)
        Next
   Next
End If
```

The domestic final demand by industries and its components are then calculated by multiplying the overall values for each final demand component with their respective shares. The difference between the domestic final demand overall economic sectors and the total final demand are the final demand imports.

In a next step, **gross output by industries** *gobpin_v(t)* can be calculated by multiplying domestic total final demand in basic prices *dtfdbpn_v(t)* and the Leontief-Inverse $(i\_m - ticn\_m(t))^{-1}$.

The identity matrix *i_m* is set with ones on the main diagonal. The total input coefficient matrix *dicn_m(t)* is calculated for the last historical year 2020 by dividing the total intermediate demand matrix *didn_m(t)* by gross output *gobpin_v(t)*.

The following functions are used for the calculation of gross output by industries:

- MatMult(matrix1, matrix2) which multiplies two matrices
- MatInv(matrix1) which inverts a matrix
- MatSub(matrix1, matrix2) which subtracts two matrices

```
If t = didn_m_lastData Then
   For r = 1 To Ubound(dicn_m(t), 1)
      For c = 1 To Ubound(dicn_m(t), 2)
```

```
            dicn_m(t)(r, c) = didn_m(t)(r, c) / gobpin_v(t)
        Next
    Next
End If

'Leontief-Inverse
gobpin_v(t) = MatMult(MatInv(MatSub(i_m, dicn_m(t))), dtfdbpn_v(t))
```

Total intermediate consumption at basic prices *ticbpn_v(t)* is calculated by the sum of products of total input coefficients *dicn_m(t)(r,c),* gross output at basic prices *gobpin_v(t)* and the production-induced imports. **Value added** by industries *van_v(t)* is the difference between output and intermediate consumption at basic' prices.

Intermediate consumption and value added as proxy for the compensation of employees feed back into the calculation of unit costs.

3

Unit costs are defined as costs per unit of price-adjusted output by industry *gobpir_v(t)* while *r* represents all industries *n* as given from the underlying data set. Main cost components are labor costs which are part of value added *ucvan_v(t)* and the costs for intermediate demand *uctidn_v(t).* The value added costs by each industry are calculated as the ratio of value added and the price-adjusted output. The intermediate costs are calculated from the price-adjusted input coefficients for imported *iicr_m(t)(r,c)* and domestically produced products d*icr_m(t)(r,c)* purchased by the respective industry at a given price *gobpid_v(t)* resp. *impd_v(t)*.

$$\begin{aligned} uctidn\_v(t)(c) &= \sum_{r=1}^{n}\left(iicr\_m(t)(r,c)\cdot impd\_v(t)(r)\right) && c \in (1 \dots n) \\ &\quad + \sum_{r=1}^{n}\left(dicr\_m(t)(r,c)\cdot gobpid\_v(t)(r)\right) && r \in (1 \dots n) \\ ucvan\_v(t)(c) &= \frac{empl35\_v(t)(c)\cdot awen\_s(t)}{gobpir\_v(t)(c)} \\ ucn\_v(t)(c) &= uctidn\_v(t)(c) + ucvan\_v(t)(c) \end{aligned} \tag{3}$$

Output deflators *gobpid_v(t)* are projected as a growth rate of total costs *ucn_v(t)* for each industry assuming that an increase in production costs increases directly producer prices. If sufficient time series data for costs and output deflators are available, output deflators could be estimated with sector-specific unit costs.

$$gobpid\_v(t)(r) = gobpid\_v(t-1)(r)\cdot\frac{ucn\_v(t)(r)}{ucn\_v(t-1)(r)} \qquad r \in (1 \dots n) \tag{4}$$

4

The **labor market** is modelled at an aggregate level. Demographic development, i. e. population at working age *pag_v(t)(3)* determines **labor force** *lf_s(t)*. Population by age groups is an exogenous variable and taken from the World population prospects of the UN (2022) until 2050.

$$\begin{aligned} &pag\_v(t)(r) = exogenous && r \in (1 \dots 4) \\ &lf\_s(t) = f\left(pag\_v(t)(3)\right) \end{aligned} \tag{5}$$

Employment for *r* economic activities *empl_v(t)* is estimated as a function of price-adjusted, aggregated gross output *agobpir12_v(t)*[8]. Overall employment *empl_s(t)* is calculated as the sum overall economic activities.

[8] For the description of each element please refer to the *RowColDesc* worksheet in the e3cc model workbook.

$$(6) \qquad empl\_v(t)(r) = f(agobpir21\_v(t)(r)) \qquad\qquad r \in (1 \dots n)$$

An **average wage rate** per employee over all industries a*wen_s(t)* is estimated as a function of macroeconomic labor productivity *gdper_v(t-1)(1) / empl_s(t-1)*, the consumer price deflator *gdped_v(t-1)(2)* and a labor scarcity factor *lf_s(t-1) / empl_s(t-1). (t-1)* indicates that the influencing factors are considered with a time lag of one year.

$$(7) \qquad awen\_s(t) = f\left(\frac{gdper_{v(t-1)(1)}}{empl_{s(t-1)}}, gdped\_v(t-1)(2), \frac{lf\_s(t-1)}{empl\_s(t-1)}\right) \qquad r \in (1 \dots n)$$

3.3 Energy module

3.3.1 GENERAL OVERVIEW

The energy module describes the relations within the energy sector in greater detail than the economic model. It depicts the energy demand, supply, and transformation in physical terms (TJ) by different fossil fuels and renewables (see Table 3).

Table 3: Represented energy sources

Primary energy source	Secondary energy sources
Coal and peat	Coal and peat products
Oil	Oil products
Natural gas	Electricity
Biofuels and waste	Heat
Nuclear	
Renewable energy	

Source: Own illustration

The energy demand is mapped for various consumers such as private households, the industry sector (including manufacturing, construction, and mining) and the transport sector (Figure 25). Sector-specific energy demand is explained by key drivers such as sector production and population.

The energy supply is determined by the energy demand of all sectors. Energy is either produced domestically or imported. Primary energy inputs for heat and power generation are captured in the energy transformation sector.

The prices of fossil fuels, e.g., crude oil, are exogenous world market prices taken from the IEA. Energy prices to be paid for energy consumption by households and industrial consumers follow the world market prices.

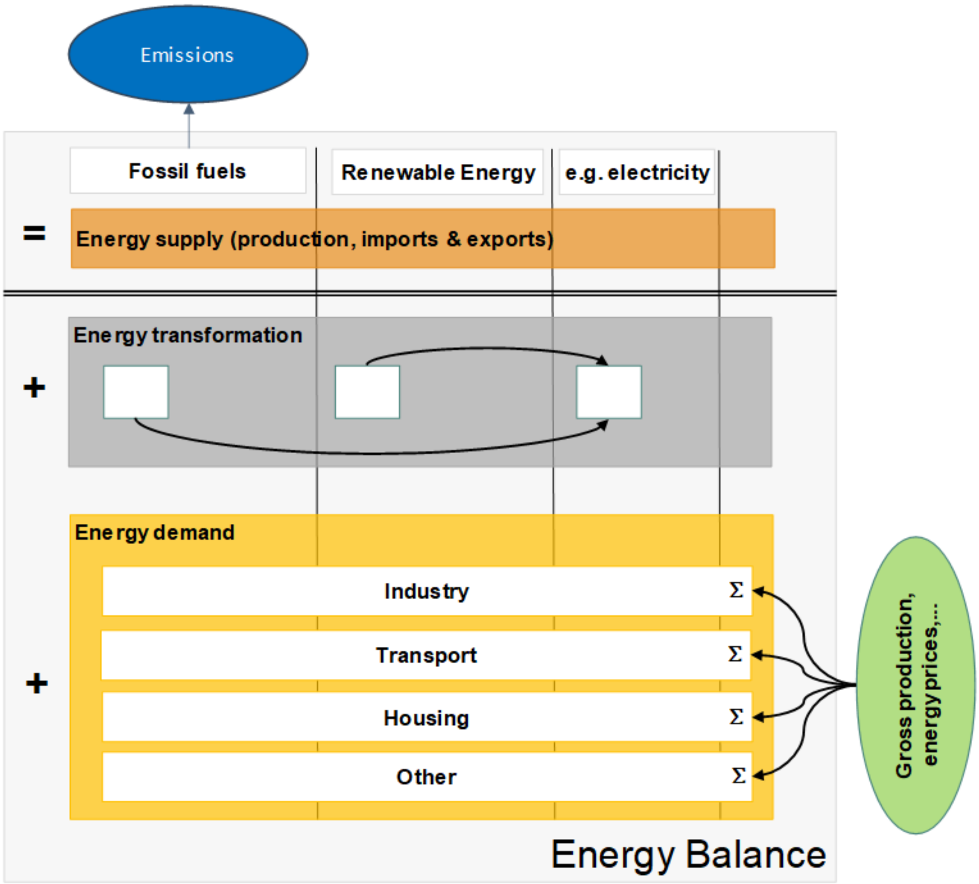


Figure 25: Energy module at a glance

Source: Own illustration by GWS

3.3.2 DATASET

The energy module is based on the data from e.g., the UN and the World Bank (Table 4). If no data is available for essential indicators from international data providers, data from the Energy Regulatory Commission (ERC) is used, e.g., for electricity price.

Table 4: Data set for the energy module

Description	Time frame	Sources
ENERGY		
Energy balance ▪ Energy supply ▪ Energy transformation ▪ Energy consumption by sectors ▪ Various energy sources e.g., fossil fuels, renewable energy, electricity	▪ 1991-2019	▪ UN
Energy prices for various energy sources ▪ World market / import prices for fossil fuels ▪ End-user prices e.g., for diesel	▪ 1991 (2000)-2021	▪ World Bank ▪ ADB ▪ IEA ▪ ERC

Source: Own illustration

The following data transformations in the energy module are required to fulfil the data needs and properties. These calculations are performed in MS Excel and results are stored in the folder *OwnCalculations*.

(1) Calculation of energy balance shares, e.g., share of coal in total final energy consumption of households and

(2) Creating sufficient time series for energy prices by fuel.

3.3.3 IMPLEMENTATION INTO DIOM-X

All model equations which are not estimated and related to this subsection are part of the *Energy* Module. Regression equations are part of the *Regressions* module.

In the energy module, both the energy prices and energy quantities (demand and supply) are projected given the relationships described in the subsequent paragraphs.

The **energy prices** *ep_v(t)* to be paid by households and industrial consumers are estimated by the crude oil world market prices *wmp_v(t)(6)* in the case of e.g., oil products. For heat and electricity prices the world market price for coal (wmp_v(t)(9) is relevant as mainly coal is used in the heat and electricity generation process.

$$\begin{aligned} &wmp\_v(t)(r) = exogenous \\ &ep\_v(t)(r) = f(wmp\_v(t)(r)) \end{aligned} \qquad r \in (1 \dots ec) \tag{8}$$

```
“R” code

inject(“’Coal”)
fit <- reg(log(ep_v.1) ~ log(wmp_v.9))
lm2vba(fit,”wmp_v(t)(9)>0”)

inject(“’Oil products”)
fit <- reg(log(ep_v.2) ~ log(wmp_v.6))
lm2vba(fit,”wmp_v(t)(6)>0”)
...

VBA regression function

Public Sub energyprices(ByVal t As Integer, ByVal iteration As Integer)
'Coal
ep_v(t)(1) = exp(6.262294 + 1.204018 * log(wmp_v(t)(9)) +
                         0.627991 * dv_v(t)(52) + 0.04896763)

‘Oil products
ep_v(t)(2) = exp(1.434413 + 1.295718 * log(wmp_v(t)(6)) 1)

...
End Sub
```

World market price for crude oil is taken from IEA projections, is set exogenously in the *Scenario* worksheet and enters the model via the tweak mechanism:

```
Tweak t, wmp_v_lastData, wmp_v, “wmp_v”
```

The modelling of **energy quantities** is determined by the structure of the country’s energy balance which shows the relationships between energy supply, energy transformation and energy demand for fossil fuels and renewable energy sources in Terajoule (TJ). Table 5 provides an overview of the approaches used to project the elements of the energy balance.

Table 5: Approaches for projecting the elements of the energy balance

columns	1	2	3	4	5	6	7	8	9	10	11
rows	Primary coal, peat	Coal, peat products	Primary oil	Oil products	Natural gas	Biofuels, waste	Nuclear	Electricity	Heat	Total energy supply	Memo: of which renewables
1 Primary production	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	▒	🟪
2 Imports	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	🟪
3 Exports	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	▒	🟪
4 International marine bunkers	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	▒	x
5 International aviation bunkers	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	▒	x
6 Stock changes	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	▒	x
7 Total energy supply	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	🟪
8 Statistical differences	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	▒	🟪
9 Transformation	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	🟪
10 ...Electricity, Heat and CHP plants	▒	▒	▒	▒	▒	▒	▒	🟦	▒	▒	x
11 ......Electricity plants - main activity producers	x	x	x	🟩	x	x	x	▒	x	▒	x
12 ......CHP plants - main activity producers	🟩	x	x	🟩	x	x	x	🟡	🟦	▒	x
21 ...Other transformation	x	x	x	x	x	🟪	x	x	x	x	🟪
22 Energy industries own use	x	x	x	x	x	x	x	🔷	🔷	▒	x
23 Losses	🟢	x	x	x	x	x	x	🟢	🟢	▒	x
24 Final Consumption	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	🟪
25 Final Energy Consumption	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	🟪
26 Manufacturing, construction and non-fuel mining industries	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟧	x
40 Transport	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟧	x
47 Other Consumption	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	🟪
48 ...Agriculture, forestry and fishing	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟧	x
49 ...Commerce and public services	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟧	x
50 ...Households	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟧	🟪
51 ...Other consumption not elsewhere specified	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟧	x

🟧	Estimated
🟨	Shares
🟪	Constant
⬛	Exogenous
▒	(Sub-)Totals / by definition
🟦	Output-demand ratio (respective column)
🟩	Input-output ratio (respective row)
🟡	Heat-power ratio (respective row)
🟢	Loss-demand ratio (respective column)
🔷	Own use-output ratio (respective column)
	x - not available

Source: Own illustration based on the Mongolian energy balance

Final energy demand is mapped for the industry sector, for private households and the transport sector. Key drivers of the sector-specific energy demand *eb_m(t)(r, tot)* over all energy sources *tot* are the economic activity of the respective sectors a*gobpir(t)(r)*. A positive sectoral economic development increases the energy demand. Past improvements in the energy efficiency are captured in the regression coefficients. The direction and magnitude of the relationship between the relevant variables is derived by applying econometric methods. The resulting regression equations for all sectors are included in the VBA code.

The energy demand of private households *eb_m(t)(hh, tot)* is estimated with population *pag_v(t)(1)*.

$$
\begin{aligned}
&eb\_m(t)(r,tot) = f\big(agobpir5(t)(r)\big) && r\ \in\ (1 \ldots s) \\
&eb\_m(r)(hh,tot) = f(pag\_v(t)(1))
\end{aligned}
\tag{9}
$$

Total final energy demand then determines the final energy demand for each energy source by sectors according to their historical shares *ebs_m(t)(r,c)* in final energy consumption.

```
“R” code

inject(“’Commerce and public”)
fit <- reg(log(eb_m.49.10) ~ -1+log(agobpir5_v.4)+dv_v.54+dv_v.44)
lm2vba(fit,"agobpir5_v(t)(4)>0 ")
…

VBA regression function

Public Sub energydemand(ByVal t As Integer, ByVal iteration As Integer)
'Commerce and public
eb_m(t)(49, 10) = exp(0.990078 * log(agobpir5_v(t)(4))

End If…
End Sub
```

Total final energy consumption (TFEC) for each energy source *eb_m(t) (24, c)* respectively is the sum of the final energy demand over all energy consumers

Since primary energy (e.g., crude oil) is not provided directly to the consumer, the energy module contains the energy transformation sector, which converts primary energy to final energy (e.g., electricity).

TFEC determines the **energy transformation** output. In the case of heat, the ratio of heat output to heat demand is calculated for the last historical year of the energy balance and applied to derive transformation output of the heat producers assuming that with increasing heat demand heat generation is increasing.

For generating heat, coal and oil products are used (**energy transformation** input). The use of the fossil fuels is determined as an input-output ratio for the heat producers.

Transfers and statistical differences are assumed to be constant. The energy industries own use as well as the energy losses depends on the activity in the transformation sector.

Total primary energy supply (TPES) *eb_m(t)(7, c)* is calculated from the demand perspective as shown in the following table.

Table 6: Determination of total primary energy supply

	Demand perspective
	Total Final Energy Consumption
-	Energy Industry Own Use (-)
-	Losses (-)
+	Statistical Difference
-	Transformation
+	Non-energy use
=	**Total Primary Energy Supply**

Source: Own illustration

Next, remaining components on the supply side are determined. **Domestic production** for fossil fuels and biofuels is exogenously given. Also, **exports** of fossil fuels *eb_m(t)(3, …)* must be given exogenously.

International marine and aviation bunkers as well as stock changes are assumed to be constant. Finally, **imports** *eb_m(2, c)* can be calculated as residuals.

```
For c = 1 To Ubound(eb_m(t), 2) – 1
  eb_m(t)(2, c) = eb_m(t)(7, c) – MatColSumRange(eb_m(t), c, 3, 6) -
                        eb_m(t)(1, c)
Next
```

3.4 Emission module

3.4.1 GENERAL OVERVIEW

The emission module projects the combustion-related CO_2 emissions[9] in different sectors *s* such as in the energy industry, manufacturing, and construction as well as transport and other sectors such as commercial and residential sector. Reductions in the use of fossil fuels can be seen in CO_2 savings. The combustion-related CO_2 emissions are connected to the primary energy use of fossil fuels via emission factors.

3.4.2 DATASET

The data on CO_2 emissions (in Gg CO_2 equivalents) and emission factors for various activities (in kg per TJ) are published by UNFCCC for the period 1990 to 2014.

Table 7: Data set for the emission module

Description	Time frame	Sources
	EMISSIONS	
CO_2 emissions by sectors	▪ 1990-2014	▪ UNFCCC
Implied (or default) emission factors		▪ UNFCCC

Source: Own illustration

3.4.3 IMPLEMENTATION INTO DIOM-X

All model equations related to these subsections are part of the *Emissions* Module in VBA.

Basically, CO_2 emissions by sectors *ghge_v(t)(r)* are calculated for fossil fuels *f* by multiplying the respective emission factors *ef_v(t)(r, c)* given in kg/TJ and corresponding sectoral energy consumption of fossil fuels as stated in the energy balance *eb_m(t)(r,* c) in TJ. Energy use of the energy transformation sector and of final consumers are considered. Resulting GHG emissions are then converted from kg to Gg.

$$(10) \qquad ghge\_v(t)(r) = \sum_{r=1}^{ff} eb\_m(t)(r,c) \cdot ef\_v(t)(r,c) \qquad \begin{array}{l} r \in (1 \dots s) \\ c \in (1 \dots f) \end{array}$$

A snippet of the corresponding VBA code is shown below.

[9] Other GHG emissions (e.g., CH_4, N_2O) and other GHG source and sink categories (e.g., industrial processes and use, agriculture) are not projected in the e3.cc model.

```
Const KgToGg = 1000000

'Energy industries
 ghge_v(t)(1) = Abs(eb_m(t)(10, 1)) * ef_v(t)(7)
                  + Abs(eb_m(t)(10, 4)) * ef_v(t)(4)

'Manufacturing industries, construction
ghge_v(t)(2) = eb_m(t)(26, 1) * ef_v(t)(10)

'Conversion kg into Gg
For r = 1 To UBound(ghge_m(t), 1)
  ghge_m(t)(r, c) = ghge_m(t)(r, c) / KgToGg
Next
```

4 Scenario or “what-if” analysis

4.1 General remarks

The E3 model is used to simulate the economic effects of climate change impacts and climate change adaptation measures. This so-called scenario (or “what-if”) analysis is used to answer questions such as

- “What are the economic impacts of climate change?” or
- “What are the economic impacts of sector-specific adaptation measures?”

Scenario analysis is the methodology which scientists use to deal with the uncertainties of the future. A scenario comprises a consistent set of quantified assumptions describing the future development. These assumptions are assigned to suitable E3 model variables and then cause chain reactions in the E3 model. Model results show the economy-wide impacts of the scenarios under consideration.

A scenario helps to better understand what could happen, who / what is affected and how. Thus, a scenario should not be considered a precise forecast; instead, it shows one of various possible development paths that are reactions to the assumptions made (“what-if” analysis).

The starting point of such analysis usually is the so-called business as usual (BAU) scenario which reflects the continuation of the economy under the assumption that past behavior continues, and some exogenous factors. The BAU scenario does not include the policy measure(s) under consideration or unexpected events, here i. e. climate hazards or adaptation policies and actions. It therefore provides the baseline against which climate change and adaptation scenarios are quantified.

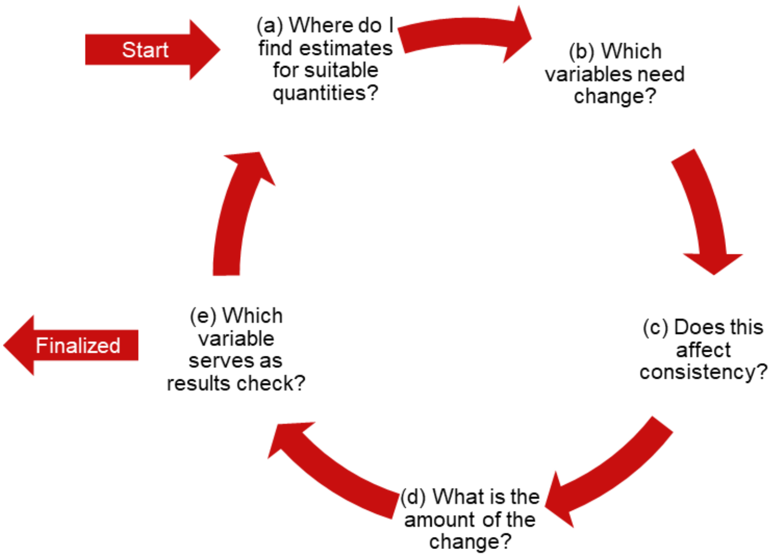

Start
(a) Where do I find estimates for suitable quantities?
(b) Which variables need change?
(c) Does this affect consistency?
(d) What is the amount of the change?
(e) Which variable serves as results check?
Finalized

Figure 26: Scenario building

Source: Own illustration

Building a scenario is based on the steps shown in Figure 26. According to the underlying questions that should be answered, sources (literature, experts) for suitable quantities need to be identified (a). Then, the climate hazard or adaptation measure must be translated into changes of appropriate model variable(s) (b). For this, knowledge about the model structure is required. Not every model variable can be adjusted or "tweaked", i. e. no left-hand side variables of identities – otherwise the definition does not hold anymore. For example, GDP by definition equals consumption, investments and net exports. Thus, it should never be "tweaked". Instead, one of the GDP components (e.g., consumption) could be tweaked (c).

Basically, all model variables that are estimated, exogenously[10] given or not projected (assumed to be constant in the future) can be "tweaked". In the worksheet *Dataset column N* of the workbook e3cc.xlsb, it is indicated if a certain model variable is tweakable ("Yes"), partially tweakable ("Partially") or not tweakable ("No"). *Partially tweakable* indicates that only some elements of a vector or a matrix variable can be tweaked.

In a next step (d), the amount of change is given for the selected variable(s). Each assumption in a scenario needs to be quantified, carefully checked, and evaluated with expert knowledge. The model cannot check the plausibility of an assumption. Implausible assumption yields implausible results and might even stop model execution prematurely if the model fails to converge.

Before **running the scenario**, the model user should have an idea which model variable(s) he / she expects to be impacted by the initial change / shock. This step helps to identify whether the results are plausible or not. As described above, the chain reactions within the E3 modelling system are manifold (e.g., second round effects, income-induced effects). After having identified the respective variables for the results check, the scenario is either finished and the model user prepares and **evaluates the scenario results,** or the model

[10] At least all exogenous variables e.g., population *pag_v* must be" tweaked". Otherwise, they will have the same value for the future as observed for the last year where historical data is available.

user revises the scenario and searches for further data and information and starts the corresponding cycle from the beginning.

Scenario building can be simple in terms of tweaking only one model variable (e.g., exports) but also quite complex if a scenario is based on a set of assumptions (e.g., climate change scenario, see example in section 4.2.2). It is recommended to build comprehensive scenarios step-by-step to first understand the effects of single assumptions. Otherwise, interpretation of scenario results becomes quite difficult. Additionally, knowledge about the model structure is required to get a good understanding of the scenario results.

To see the effects of an alternative scenario, this scenario is compared to the reference scenario (e.g., the BAU or reference scenario) which does not contain the assumptions of the alternative measure. The resulting deviations for various model variables, e.g., GDP, sectoral employment and production can be interpreted as impacts of the examined "alternative". Deviations for each model variable are given in % and / or absolute deviations in the underlying units e.g., monetary values or physical units (Figure 27).

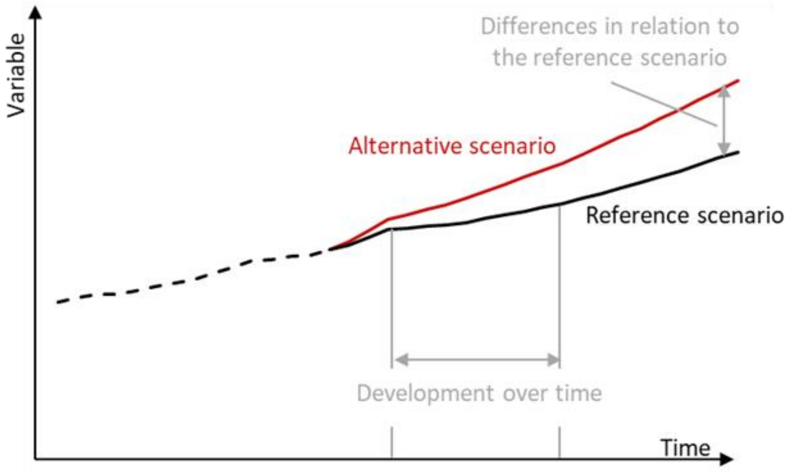


Figure 27: Scenario comparison

Source: Own illustration.

4.2 Climate change and adaptation scenarios

The starting point for the macroeconomic analysis of climate change adaptation is the consideration of climate change scenarios including e.g. extreme weather events and their economic impacts in the E3 model. Afterwards, adaptation measures aiming at minimizing or preventing the adverse impacts of climate change are considered in the E3 model as well.

Contrasting the macroeconomic results of various climate change scenarios with the hypothetical no-climate change scenario (BAU) provides insights into how vulnerable the overall economy is to certain climate events. The comparison of a specific climate change scenarios with various climate change adaptation scenarios reveals which adaptation measures are effective to reduce the negative impacts of climate change.

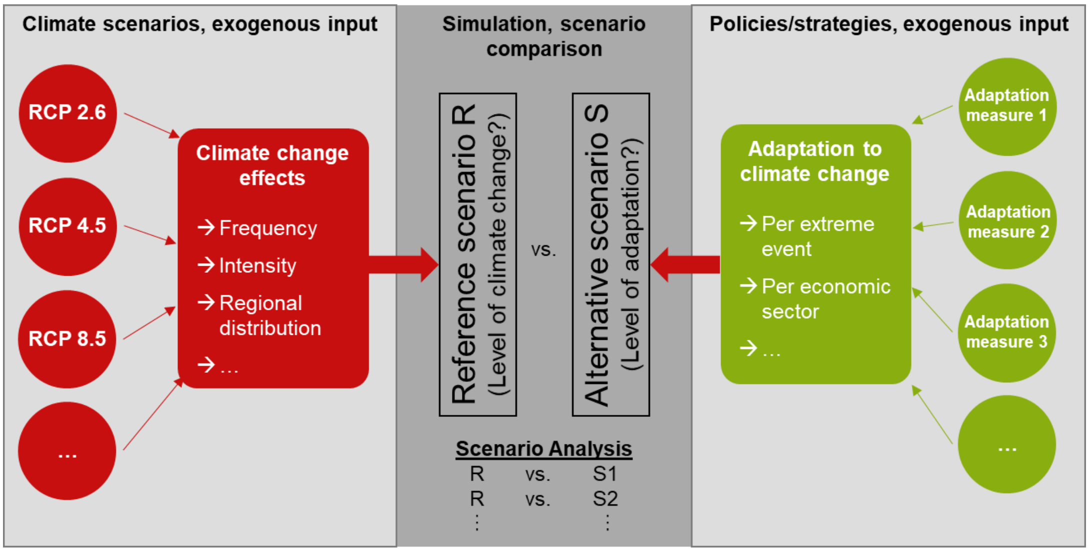


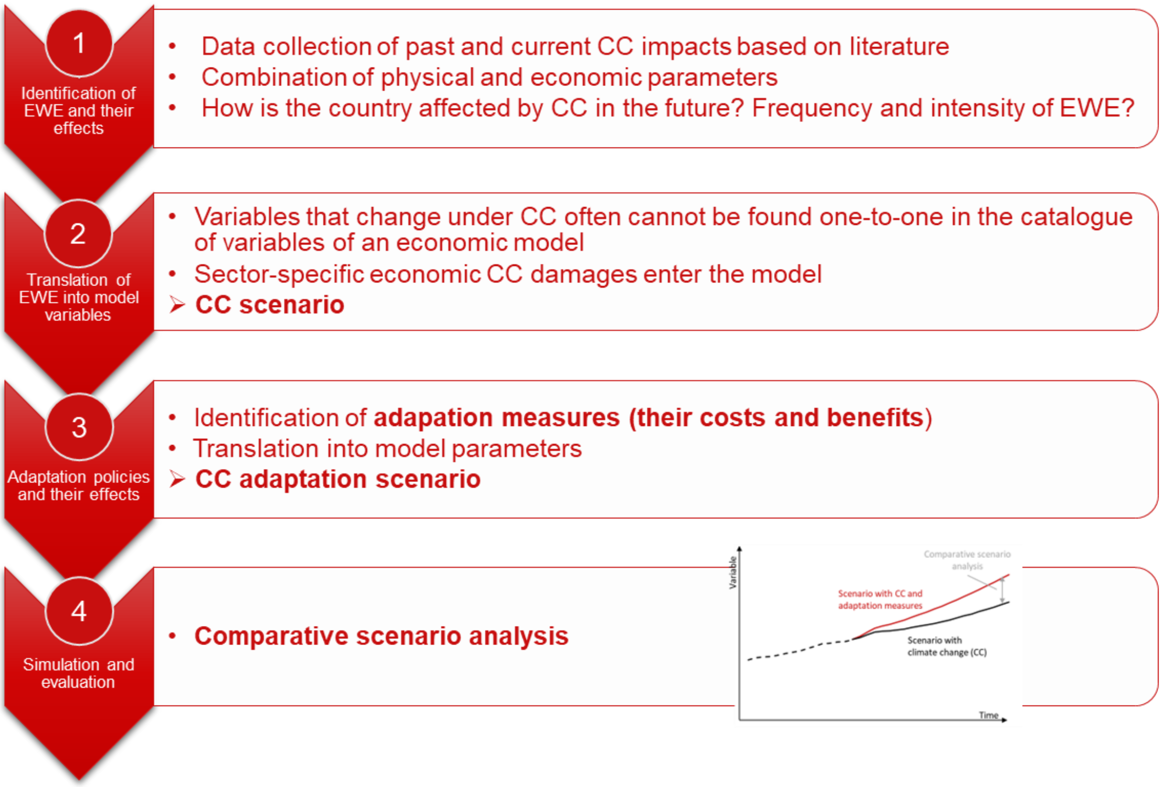
Figure 28: Comparison of climate change and adaptation scenarios

Source: GIZ, 2022a

The main prerequisites to conduct such scenario analyses are (GIZ, 2022a):

- Past and current economic damages due to climate change,
- Identification of the most vulnerable sectors and affect chains of climate hazards,
- Future evolution of climate hazards
- Cost and benefit analyses (CBA) of sector-specific adaptation measures.

This information is implemented into the E3 model following the 4-step approach shown in Figure 29.



1 Identification of EWE and their effects

- Data collection of past and current CC impacts based on literature
- Combination of physical and economic parameters
- How is the country affected by CC in the future? Frequency and intensity of EWE?

2 Translation of EWE into model variables

- Variables that change under CC often cannot be found one-to-one in the catalogue of variables of an economic model
- Sector-specific economic CC damages enter the model
- ➢ **CC scenario**

3 Adaptation policies and their effects

- Identification of **adapation measures (their costs and benefits)**
- Translation into model parameters
- ➢ **CC adaptation scenario**

4 Simulation and evaluation

- **Comparative scenario analysis**

Variable
Comparative scenario analysis
Scenario with CC and adaptation measures
Scenario with climate change (CC)
Time

Figure 29: Four-step approach to implement climate change and adaptation in an economic model

Source: GIZ, 2022a.

1) In a first step, relevant EWEs and their (bio-)physical effects are identified. Relevant interfaces and effect chains must be discovered to detect the main impacts and affected economic sectors (e.g., a drought may impact agricultural production and power generation due to water scarcity).
   Sector-specific damage data from past climate events serve as benchmarks to estimate future climate change impacts. Adjustments will be made to the benchmarks in scenarios to reflect the expected intensity of climate hazards (e.g., drought) by assuming that, for example, the doubling of hazards per year will also double the benchmark damages.
   Climate scientists may provide future evolution and intensity of country-specific climate hazards.
2) In a second step, the identified climate change effects need to be assigned to E3 model variables. The structure of the E3 model may require translations. For example, some variables in the model cannot be directly tweaked because they are given by definition. For example, changes in production are implemented by adjusting either demand or imports.
   Basically, the initial impacts of climate events are implemented as effects on human behavior, production factors and / or infrastructure (indicated by ● in Figure 30), e. g.

   - Household consumption expenditures by various products,
   - Exports by various products,
   - Investments goods,
   - Imports by various products,
   - Employment in various economic sectors,
   - Prices for various products,
   - Intermediate demand and
   - Lower output from power generation.

which impact other economic sectors not directly impacted by climate change as well as the macroeconomy and environmental indicators. Section 4.2.2 shows how the impacts of an EWE are implemented in the E3 model, using the example of a drought. Section 4.2.3 gives examples how to evaluate the impacts.

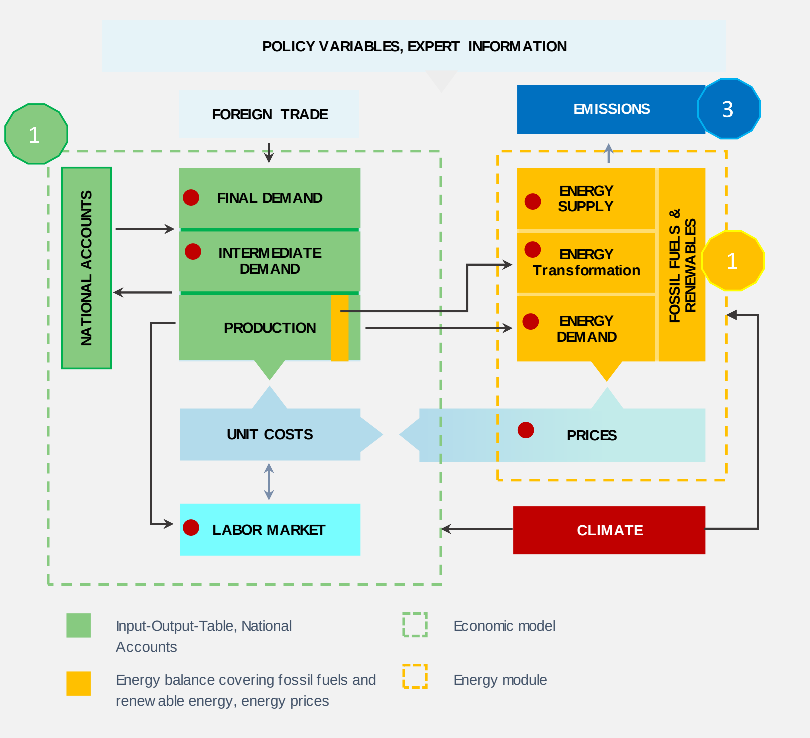


Figure 30: Implementing climate change impacts into an E3 model

Source: GIZ, 2022a

3) In a third step, appropriate adaptation measures are identified. CBAs are an important prerequisite to consider the sector-specific investments of the adaptation measure and its benefits – meaning the reverse impacts of the respective climate hazard – in the E3 model.
   Again, the costs and benefits must be assigned to appropriate model variables.
4) The final step comprises the evaluation of the macroeconomic impacts of climate change adaptation scenarios. If there is more than one sector-specific adaptation option for a certain climate event, the model helps to rank the options. Selection criteria might be the biggest avoided damages, employment effects or synergies with other strategies such as mitigation which needs to be prioritized by policymakers.

4.2.1 DATA SETS

…FOR IMPLEMENTING CLIMATE CHANGE IMPACTS

The data needed to implement the impacts of climate change hazards into an economic model comprises both data on the evolution of a climate hazard indicator in Mongolia and sector-specific economic damages from past climate hazards. The combination of both data sets enables the creation of a time series on future climate damages. Another option is to rely on country-specific forecasts from bottom-up or sector models.

Country-specific climate hazard indicators can be derived from global climate models under different climate scenarios such as the RCP8.5 (see for example Navarro and Jordà 2021 for the case of Georgia and Kazakhstan). The evolution of hazards can be estimated as the difference between the average over the period (2011-2040, 2041-2070) and the historical average (1976-2005) for various climate events such as droughts and extreme precipitation.

The average annual growth rates of the number of events per year can then be combined with the benchmark damages collected from past events.

Sector-specific economic damages from past events were collected in a brief study based on interviews, publicly available statistics, and reports with the help of local experts (Ailtgui 2022), e.g.:

- National Statistic Office, 2022
  - Livestock mortality https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_029V1&13999001_select_all=1&13999001SingleSelect=&BAG_select_all=0&BAGSingleSelect=_0&YearY_select_all=1&YearYSingleSelect=&viewtype=table)
  - Disaster occurrences and damages https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_2400_012V1&CL_24_12_select_all=1&CL_24_12SingleSelect=_10&YearM_select_all=0&YearMSingleSelect=_201412_201512_201612_201712_201812_201912_202012_202112_202209&YearY_select_all=1&YearYSingleSelect=&viewtype=table
- Mongolia's National Emergency Management Agency (https://www.thethirdpole.net/en/climate/in-cold-dry-mongolia-floods-are-now-common-with-devastating-urban-impacts/)
- Number of floods and damages https://ikon.mn/n/1bae (Disaster Research Institute)
- World Bank 2021. Climate Risk Country Profile Mongolia
- CFE-DM 2022. Mongolia Disaster Management Reference Handbook.
- ADPC, UNDRR 2019. Disaster Risk Reduction in Mongolia. Status Report 2019
- MET 2018. Third National Communication of Mongolia.
- MEGD, UNEP, GEF 2013. Technology needs assessment. Volume 1 – Climate change adaptation in Mongolia

Table 8 and Table 9 show the reported non-monetary and monetary damages by EWE reported in Mongolia at a glance. Directly impacted from climate change are mainly the agriculture, construction, transport, and energy sector. The largest known monetary damages were caused dzuds which is a national phenomenon characterized by a summer drought followed by extreme harsh conditions in winter. Extreme precipitation and floods continue to increase and cause major damages.

While Table 8 depicts sector-specific – non-monetary – damages, the monetary damages in Table 9 show – with the exception of dzuds – the totals. For this EWE, number of livestock losses by animals are valued with the average market value of livestock to derive the monetary damages for the agriculture sector.

For analyzing the inter-industry effects and to derive sound impacts for the macro-economy, sector-specific impacts are very important. Otherwise, sector-specific labor-intensities and import dependencies cannot be considered. Thus, further evidence and expert knowledge should be consulted.

Table 8: Reported non-monetary damages (examples)

Economic sectors affected	Extreme precipitation, floods	Extreme Wind	Drought, dzuds
Agriculture	Damaged crops		Damaged crops
			Reduced pasture productivity

	Destroyed agricultural land, soil degradation	Soil degradation	
	Killed livestock	Killed livestock	Killed livestock
Construction	Damaged buildings	Damaged buildings	
	Damaged bridges		
	Damages roads		
Energy	Damaged electric substations and transmission lines	Damaged electric substations, lighting poles	Impaired hydropower generation
Transport	Blocked roads and passes	Blocked road and train traffic	

Source: Own illustration based on data compilation by Ailtgui 2022 and GWS.

Table 9: Reported monetary damages (in Mn. MNT)

Year	Floods[1]	Dzuds[2]	Total damage[3,*]
1996			5,200
1997			14,800
1998			5,200
1999			7,000
2000			88,000
2001			162,000
2002			116,000
2003			38,000
2004			3,000
2005			6,000
2006			12,000
2007	228		201,000
2008	17		35,100
2009	4,201		9,800
2010	745	600,5	534,800

2011	56	66,400	9,600
2012	98	63,500	94,480
2013	3108	202,800	22,000
2014	755	110,100	25,200
2015	324	154,800	79,950
2016	4,030	279,700	43,100
2017	634.3	241,600	96,700
2018	16,318	848,900	81,900
2019	107	291,300	35,800
2020	3,657	689,600	31,320
2021	4,302	926,400	25,800
2022			26,700
Total	**13,562**	**4,475,700**	**1,810,450**

Sources: Own illustration based on data compilation by Ailtgui 2022.

1) https://ikon.mn/n/1bae

2)https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_040V2&SOUM_select_all=0&SOUMSingleSelect=_0_511&LivestockPrice_select_all=0&LivestockPriceSingleSelect=_1_111_121_12_11_13_131_14_141_15_151&YearM_select_all=0&YearMSingleSelect=_200001_200101_200201_200301_200901_201001_201101_201201_201301_201401_201501_201601_200601_200501_200701_200401_200801&viewtype=table

and

https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_029V1&13999001_select_all=1&13999001SingleSelect=&BAG_select_all=0&BAGSingleSelect=_0&YearY_select_all=1&YearYSingleSelect=&viewtype=table

https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_042V2&SOUM_select_all=0&SOUMSingleSelect=_0_341_34101&mal_select_all=0&malSingleSelect=_0_1_11_2_21_3_31_4_41_5_51&YearM_select_all=0&YearMSingleSelect=_202209_202109_202009_201909_201809_201709&viewtype=table

3)https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_2400_012V1&CL_24_12_select_all=1&CL_24_12SingleSelect=_10&YearM_select_all=0&YearMSingleSelect=_201412_201512_201612_201712_201812_201912_202012_202112_202209&YearY_select_all=1&YearYSingleSelect=&viewtype=table

* Does not correspond to the sum of the damages of the individual events.

…FOR IMPLEMENTING ADAPTATION MEASURES

The implementation of adaptation measures requires costs and benefits (in terms of damage reduction) of each measure. As with data on climate change impacts, ideally the information is country specific. Otherwise, knowledge from international experiences may be adopted for a first estimate but must be verified by local sector-experts.

For Mongolia, a technology needs assessment was conducted which provides quantified information on the costs for adaptation measures. However, the quantified benefits of such measures in terms of damage reduction are very rare (MEGD, UNEP, GEF 2013). Thus, in a first step the benefits are more assumption driven and must be revised later.

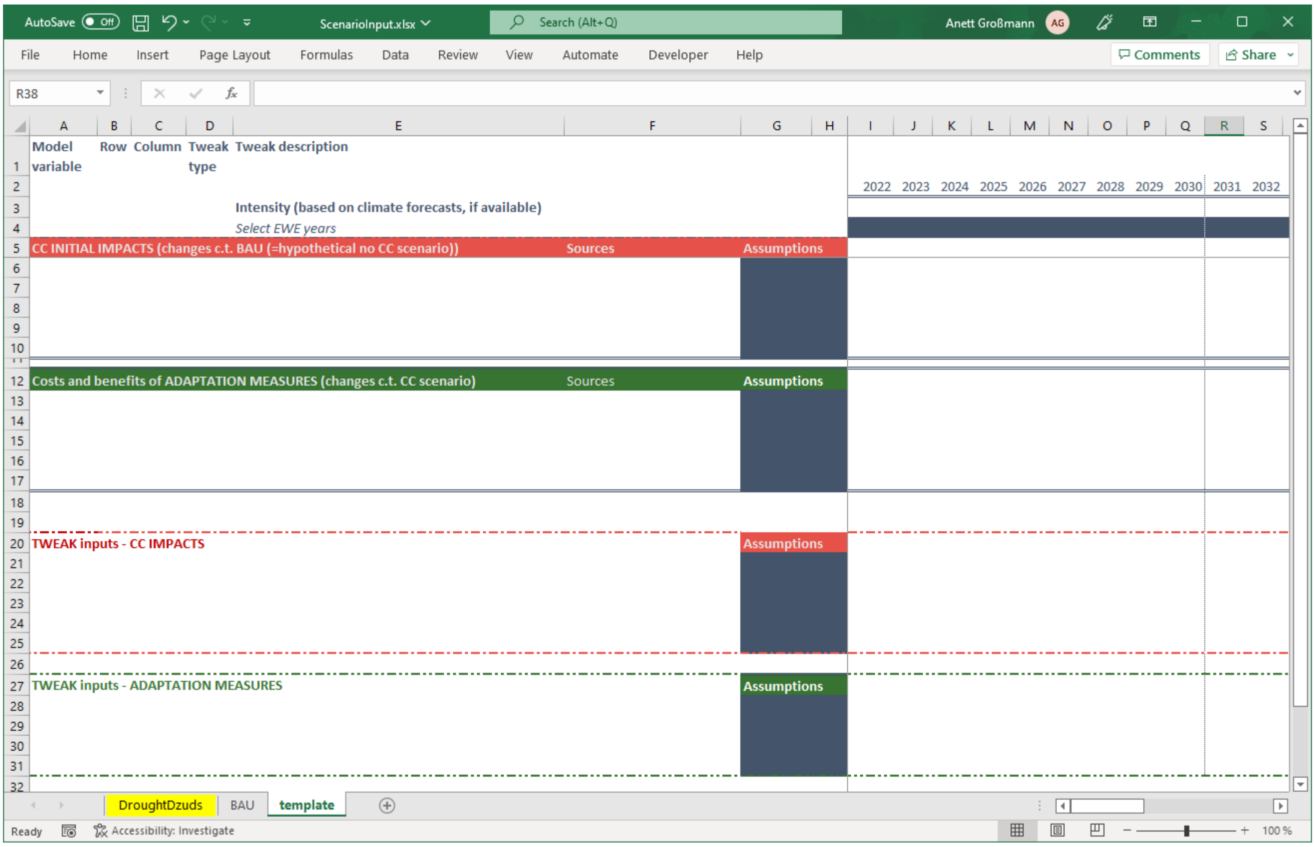
4.2.2 IMPLEMENTATION OF SCENARIOS

Scenario assumptions are usually defined in the *Scenario* worksheet of the workbook *e3cc.xlsb*. Some assumptions require auxiliary calculations which cannot be easily implemented there. Such calculations should be done in separate files / worksheets (workbook *"ScenarioInput.xlsx")* to keep the model clean and lean. The final results of these calculations are then integrated into the *Scenario* worksheet.

Most scenarios can be implemented directly in the *Scenario* worksheet. In that case, the description of how to create the tweak inputs in the workbook *ScenarioInput.xlsx* can be skipped. Climate change scenarios are more complex; thus tweak inputs are created in *ScenarioInput.xlsx.*

How to create the tweak inputs in the workbook *ScenarioInput.xlsx?*

Figure 31 shows the structure of the *ScenarioInput.xlsx* workbook. The workbook may contain several worksheets each one for a specific climate hazard. These sheets have a color code (e. g. yellow for droughts/dzuds), which is also used in the *Scenario* worksheet to identify the set of assumptions more easily. Each worksheet includes the set of assumptions for the climate change and adaptation scenarios to be defined by the scenario builder. The *template* worksheet serves as a blueprint.


AutoSave Off
ScenarioInput.xlsx
Search (Alt+Q)
Anett Großmann AG
File Home Insert Page Layout Formulas Data Review View Automate Developer Help
Comments Share
R38
DroughtDzuds BAU template
Ready Accessibility: Investigate
100 %

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Model variable	Row	Column	Tweak type	Tweak description														
2									2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
3					Intensity (based on climate forecasts, if available)														
4					*Select EWE years*														
5	CC INITIAL IMPACTS (changes c.t. BAU (=hypothetical no CC scenario))					Sources	Assumptions												
12	Costs and benefits of ADAPTATION MEASURES (changes c.t. CC scenario)					Sources	Assumptions												
20	TWEAK inputs - CC IMPACTS						Assumptions												
27	TWEAK inputs - ADAPTATION MEASURES						Assumptions												

Figure 31: Workbook "ScenarioInput.xlsx"

Source: Own illustration.

Each of the worksheets considering climate hazards has the same structure (Figure 31). The first four *columns* (A to D) look exactly as the ones in the *Scenario* worksheet where the model variable, its row and column indices and the applied tweak type need to be specified. The next column E is a placeholder for the tweak descriptions and column F contains the source of the assumptions. Column G and H include the assumptions to be applied in the simulation period until 2050.

In row three, the model user can define, if available, the average annual growth rates of the respective climate hazard. In row four, the scenario builder selects the years where the next respective climate events are expected to occur.

Starting with row six, the climate damages are quantified. The number of rows depends on the specified climate impacts and can be extended by inserting additional rows.

In this example, from row 13 onwards the assumptions on costs and benefits of adaptation measures are set. Again, additional rows can be integrated for more assumptions on adaptation measures.

Taken the example of a dzud, from past observations the losses of livestock due to dzuds in agriculture are known. The valuation with the average market price per kind of livestock results in the monetary damage, which ranges between 64 Bn. MNT and 926 Bn. MNT (Figure 32cf. section 4.2.1). In the last 12 years, the average monetary loss was 373 Bn. MNT per EWE.

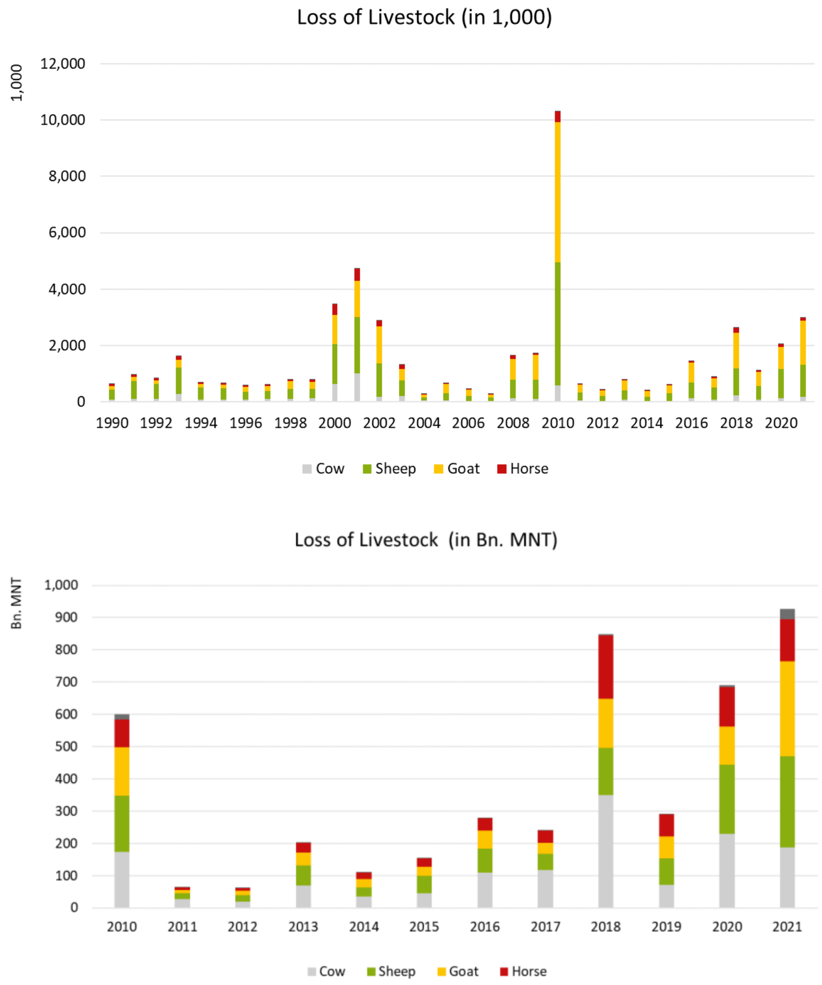


Figure 32: Loss of livestock in 1,000 (top figure) and Bn. MNT (bottom figure)

Source: Own illustration based on data compilation by Ailtgui 2022.11

Some assumptions can be easily linked to e3.mn model variables, e. g. higher investments for construction of water canals which is the model variable *dfdn_m(18,4)*. Other assumptions require additional calculations, e.g., livestock losses in agriculture. Production values cannot be tweaked because it is a result of the demand driven E3 model. Thus, the scenario builder has to tweak the demand and / or imports of “agriculture, hunting products and related services” instead of production of the said economic sector.

The final tweak inputs (framed with a dot-dash line) are given starting in row 21 (cf. Figure 31). These tweaks then are linked (or copied and pasted) to the *Scenario* worksheet in the *e3cc.xlsb* workbook.

At the example of a drought event, the steps to implement the scenario are described subsequently and follow the general scenario building procedure (cf. section 4.1 and 4.2)

First, it is assumed that every five years a drought / dzud will occur starting in 2023 (see the “x” in row four in Figure 33). Second, the benchmark damage for livestock is multiplied by the expected frequency and intensity of the droughts / dzuds. This step assures that an increasing number or intensity of drought / dzud events is reflected in the future damages.

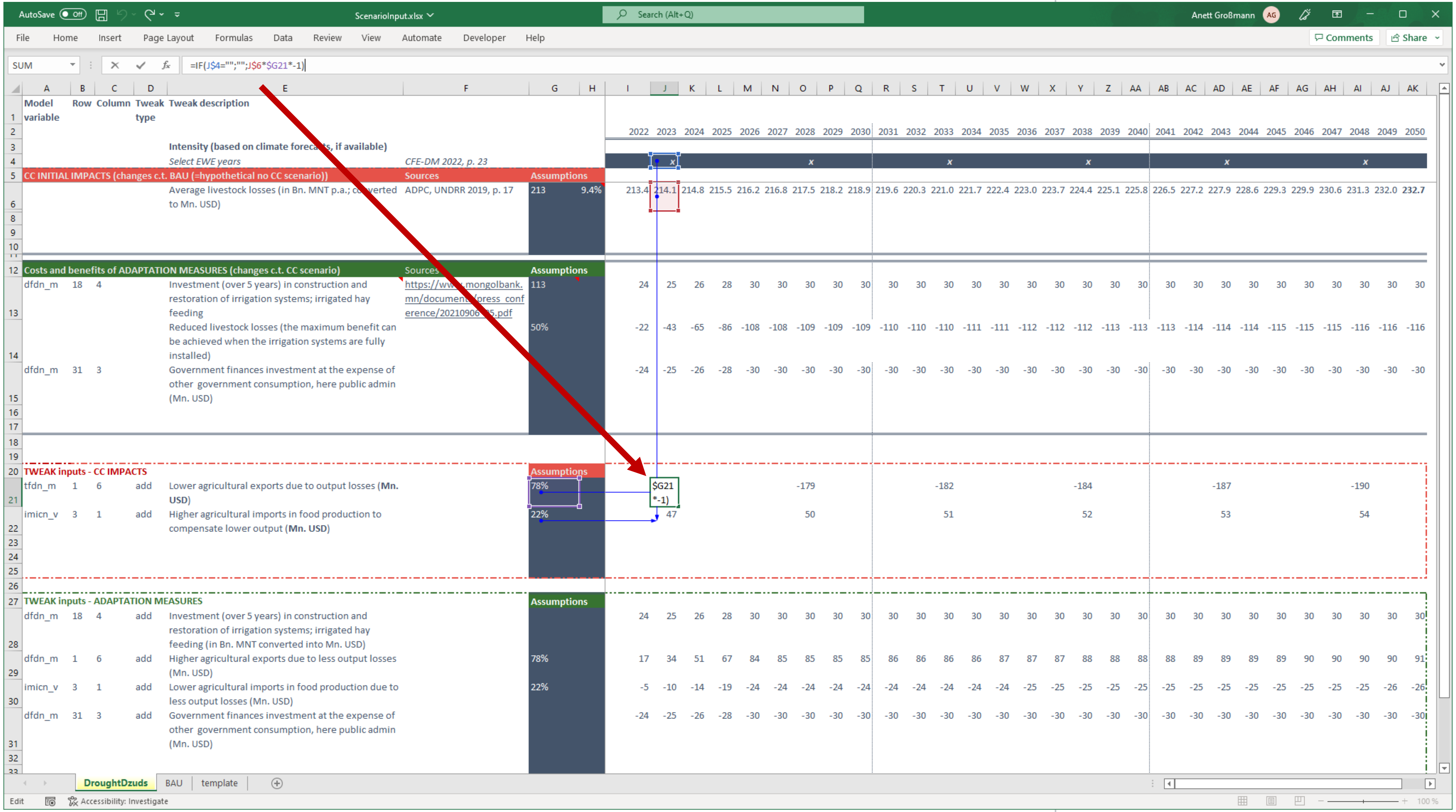
Afterwards, the livestock loss must be translated into tweakable model variable(s) (Figure 33 row 21 to 22). Two options are available. On the one hand, agricultural exports d*fdn_m(1, 6)* can be tweaked assuming that the droughts / dzuds results in foregone export chances. On the other hand, the demand for agricultural products on the domestic market needs to be compensated by higher imports of the food manufacturing sector *imicn_v(3,1)*. In this example, it is assumed that 78%[12] of the loss in agriculture result in lower exports and the remaining 22% in higher intermediate imports in the food processing sector to compensate for losses in agriculture.

For each of the model variables, a tweak type must be given. For the agricultural exports and imports an *add* tweak is selected because in each year, the values must be added resp. subtracted from the values calculated by the model.

[11] https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_040V2&SOUM_select_all=0&SOUMSingleSelect=_0_511&LivestockPrice_select_all=0&LivestockPriceSingleSelect=_1_111_121_12_11_13_131_14_141_15_151&YearM_select_all=0&YearMSingleSelect=_200001_200101_200201_200301_200901_201001_201101_201201_201301_201401_201501_201601_200601_200501_200701_200401_200801&viewtype=table

https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_029V1&13999001_select_all=1&13999001SingleSelect=&BAG_select_all=0&BAGSingleSelect=_0&YearY_select_all=1&YearYSingleSelect=&viewtype=table
https://www.1212.mn/tables.aspx?tbl_id=DT_NSO_1001_042V2&SOUM_select_all=0&SOUMSingleSelect=_0_341_34101&mal_select_all=0&malSingleSelect=_0_1_11_2_21_3_31_4_41_5_51&YearM_select_all=0&YearMSingleSelect=_202209_202109_202009_201909_201809_201709&viewtype=table

[12] According to statistic ~78% of agriculture products are exported (https://data.adb.org/dataset/mongolia-input-output-economic-indicators).



SUM =IF(J$4="";"";J$6*$G21*-1)

	A	B	C	D	E	F	G	H
1	Model variable	Row	Column	Tweak type	Tweak description			
2								
3					Intensity (based on climate forecasts, if available)			
4					*Select EWE years*	*CFE-DM 2022, p. 23*		
5	CC INITIAL IMPACTS (changes c.t. BAU (=hypothetical no CC scenario))					Sources	Assumptions	
6					Average livestock losses (in Bn. MNT p.a.; converted to Mn. USD)	ADPC, UNDRR 2019, p. 17	213	9.4%
8								
9								
10								
12	Costs and benefits of ADAPTATION MEASURES (changes c.t. CC scenario)					Sources	Assumptions	
13	dfdn_m	18	4		Investment (over 5 years) in construction and restoration of irrigation systems; irrigated hay feeding	https://www.mongolbank.mn/documents/press_conference/20210906_05.pdf	113	
14					Reduced livestock losses (the maximum benefit can be achieved when the irrigation systems are fully installed)		50%	
15	dfdn_m	31	3		Government finances investment at the expense of other government consumption, here public admin (Mn. USD)			
16								
17								
18								
19								
20	TWEAK inputs - CC IMPACTS						Assumptions	
21	tfdn_m	1	6	add	Lower agricultural exports due to output losses (**Mn. USD**)		78%	
22	imicn_v	3	1	add	Higher agricultural imports in food production to compensate lower output (**Mn. USD**)		22%	
23								
24								
25								
26								
27	TWEAK inputs - ADAPTATION MEASURES						Assumptions	
28	dfdn_m	18	4	add	Investment (over 5 years) in construction and restoration of irrigation systems; irrigated hay feeding (in Bn. MNT converted into Mn. USD)			
29	dfdn_m	1	6	add	Higher agricultural exports due to less output losses (Mn. USD)		78%	
30	imicn_v	3	1	add	Lower agricultural imports in food production due to less output losses (Mn. USD)		22%	
31	dfdn_m	31	3	add	Government finances investment at the expense of other government consumption, here public admin (Mn. USD)			
32								

	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
2	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050
4		*x*					*x*					*x*					*x*					*x*					*x*		
6	213.4	214.1	214.8	215.5	216.2	216.8	217.5	218.2	218.9	219.6	220.3	221.0	221.7	222.4	223.0	223.7	224.4	225.1	225.8	226.5	227.2	227.9	228.6	229.3	229.9	230.6	231.3	232.0	**232.7**
13	24	25	26	28	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
14	-22	-43	-65	-86	-108	-108	-109	-109	-109	-110	-110	-110	-111	-111	-112	-112	-112	-113	-113	-113	-114	-114	-114	-115	-115	-115	-116	-116	-116
15	-24	-25	-26	-28	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30
21		$G21*-1)					-179					-182					-184					-187					-190		
22		47					50					51					52					53					54		
28	24	25	26	28	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
29	17	34	51	67	84	85	85	85	85	86	86	86	86	87	87	87	88	88	88	88	89	89	89	89	90	90	90	90	91
30	-5	-10	-14	-19	-24	-24	-24	-24	-24	-24	-24	-24	-24	-24	-25	-25	-25	-25	-25	-25	-25	-25	-25	-25	-25	-25	-25	-26	-26
31	-24	-25	-26	-28	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30	-30

Figure 33: „DroughtDzuds“ worksheet

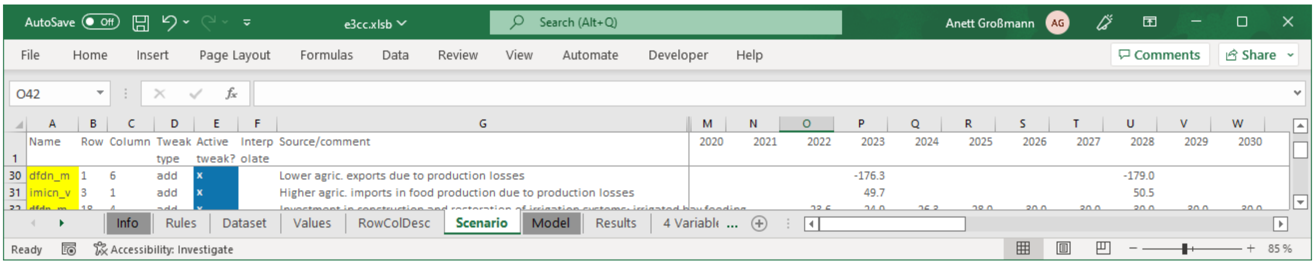
Source: Own illustration.

Implementation of the tweak inputs in the *Scenario* worksheet

The results of the preparatory work in the *ScenarioInput.xlsx* worksheet are then simply linked (or copied and pasted) into the *Scenario* worksheet in the *e3mn.xlsb* workbook (yellow cells in Figure 34). Linking cells has the advantage that if the underlying data is updated, the tweaks in the *Scenario* worksheet are automatically updated.

Then, column E "Active tweak?" must be activated for all variables where the tweak should be active. Otherwise, the tweak is inactive.

If necessary, also column F "Interpolate?" must be activated. The framework would then interpolate the given values for non-consecutive years. Since the EWE is assumed to occur only in the specified years, interpolation is deactivated.

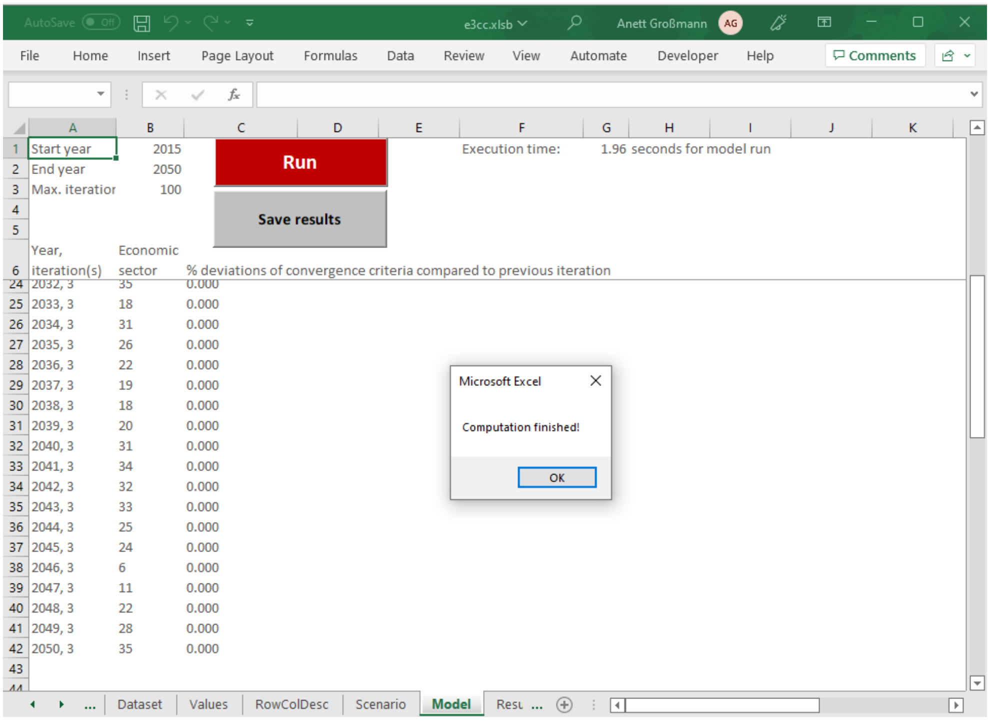


	A	B	C	D	E	F	G	M	N	O	P	Q	R	S	T	U	V	W
1	Name	Row	Column	Tweak type	Active tweak?	Interpolate	Source/comment	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030
30	dfdn_m	1	6	add	x		Lower agric. exports due to production losses				-176.3					-179.0		
31	imicn_v	3	1	add	x		Higher agric. imports in food production due to production losses				49.7					50.5		
32	dfdn_m	18	4	add	x		Investment in construction and restoration of irrigation systems; irrigated hay feeding			23.6	24.0	26.3	28.0	30.0	30.0	30.0	30.0	30.0

Figure 34: "Scenario" worksheet

Source: Own illustration.

Finally, the model must be executed by pressing the *Run* button on the *Model* worksheet. If the model run is successfully finished, results for all model variables are automatically stored in the *Results* worksheet (Figure 35).



	A	B	C	F	G
1	Start year	2015		Execution time:	1.96 seconds for model run
2	End year	2050			
3	Max. iteratior	100			
4					
5					
6	Year, iteration(s)	Economic sector	% deviations of convergence criteria compared to previous iteration		
24	2032, 3	35	0.000		
25	2033, 3	18	0.000		
26	2034, 3	31	0.000		
27	2035, 3	26	0.000		
28	2036, 3	22	0.000		
29	2037, 3	19	0.000		
30	2038, 3	18	0.000		
31	2039, 3	20	0.000		
32	2040, 3	31	0.000		
33	2041, 3	34	0.000		
34	2042, 3	32	0.000		
35	2043, 3	33	0.000		
36	2044, 3	25	0.000		
37	2045, 3	24	0.000		
38	2046, 3	6	0.000		
39	2047, 3	11	0.000		
40	2048, 3	22	0.000		
41	2049, 3	28	0.000		
42	2050, 3	35	0.000		
43					

Microsoft Excel

Computation finished!

OK

Figure 35: Model run

Source: Own illustration.

IMPORTANT: Each time the model is executed, new results are stored in the *Results* worksheet and thus overwrite previous results. Once the button *Save results* is executed, the current scenario results (*Results*, *4 Variables*) and assumptions (*Scenario*) are stored in a MS Excel file in the folder *scenarios* with an appropriate filename, e. g. “DroughtDzud_agric” (cf. section 3.1). Thus, when carrying out another simulation, the results of the “DroughtDzud_agric” will not be overwritten.

4.2.3 EVALUATION OF RESULTS

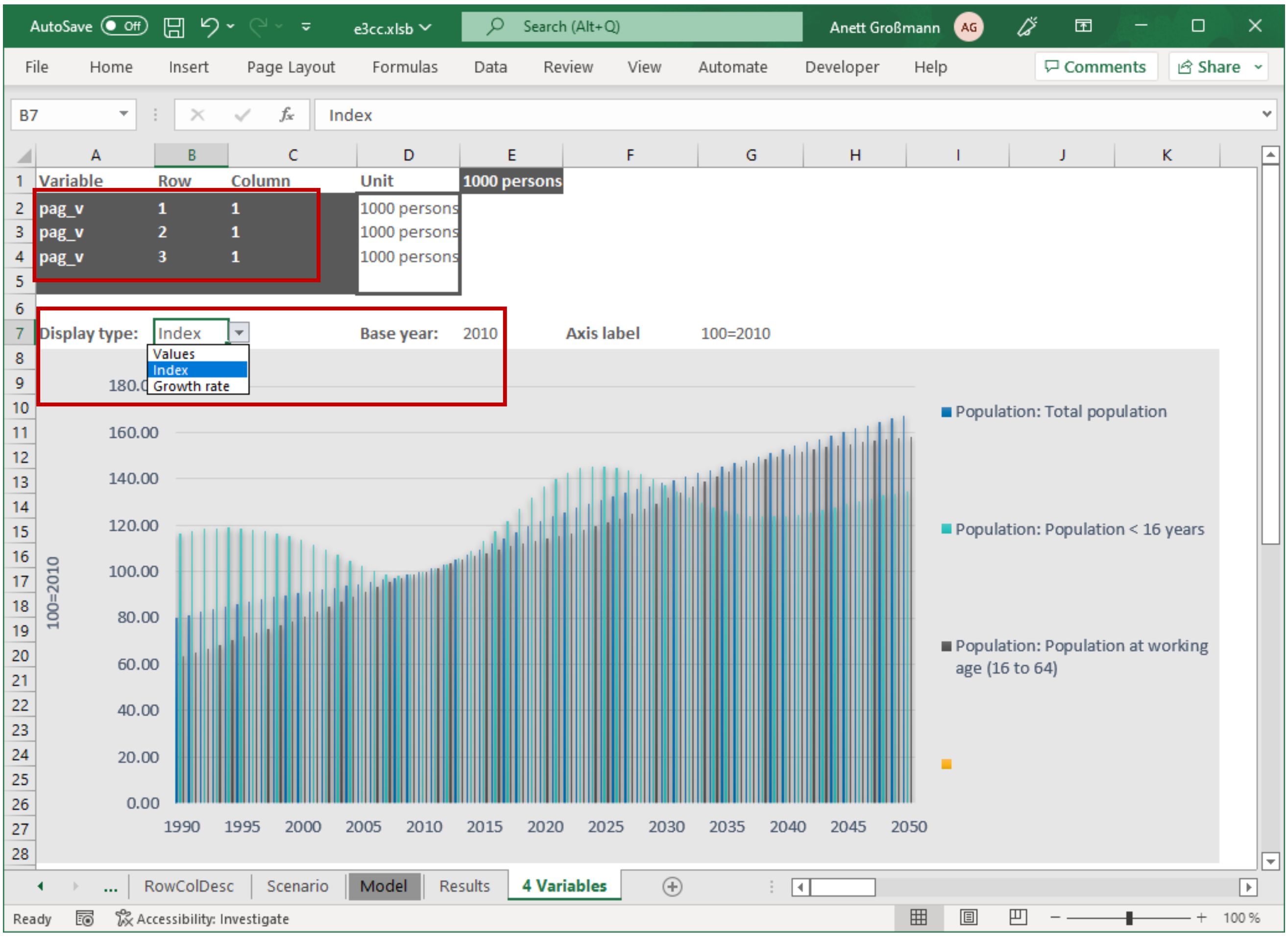
The *Results* worksheet contains the full set of historical and projected data. For policy analysis, a condensed view of the data is more practical. Model users can easily create customized views (dashboards) based on the *Results* worksheet.

A pre-prepared evaluation tool is shown in the worksheet *4 Variables*. The model user can select up to four variables (see *Dataset* worksheet) for ONE scenario which are displayed in graphical and table format. For each of the variables, the name, row, and column must be given by the user (see red rectangle in Figure 36). Units (column D) are updated automatically.

Three display types are available to be selected from the list in cell B7:

- Absolute values in the respective units,
- Annual average growth rates and
- Index values for a user specified base year (cell E7).

The figure updates automatically after selecting the display type.



Variable	Row	Column	Unit	1000 persons
pag_v	1	1	1000 persons	
pag_v	2	1	1000 persons	
pag_v	3	1	1000 persons	

Display type: Index   Base year: 2010   Axis label 100=2010

Figure 36: Evaluation tool “4 Variables”

Source: Own illustration.

Comparison and evaluation of TWO scenarios

To see the effects of a certain (alternative) scenario, it must be compared to another scenario that does not consider these assumptions. The differences between the scenarios can then be attributed to the different assumptions in the scenarios and the triggered reactions in the model.

The evaluation tool *_Compare2Scenarios_.xlsb* in the folder *scenarios* (cf. section 3.1) allows for a comparative analysis. In a first step, the reference and the alternative scenario have to be selected. By pressing the button "*1. Step: UPDATE List of Scenarios*", all scenarios stored in the folder *scenarios* are displayed in the two list boxes. In the left (right) list box the scenario which should serve as reference (alternative) must be selected by clicking on it.

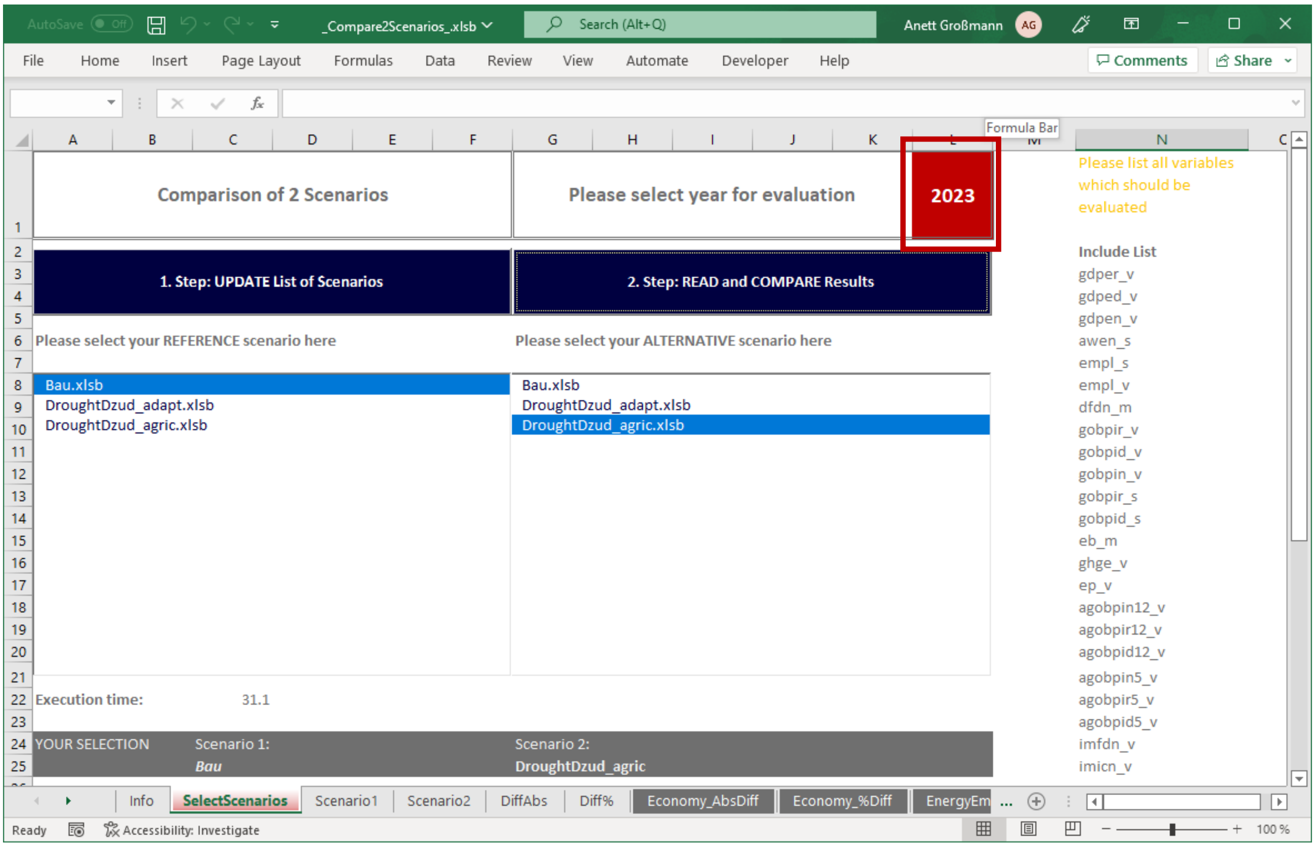
In the second step, the results of both scenarios are compared in relative and absolute values by activating the button "*2. Step: READ and COMPARE Results*". Basically, the deviations in absolute values and in percent can be calculated for all model variables and stored in this workbook. To speed up the comparison, the user should list only those model variables which need to be evaluated (column N in Figure 37).

Preconfigured graphs / dashboards – either for one selected year or as times series – for the most important e3.cc model variables are shown in different worksheets (Figure 38). The results are always given **as differences** for the selected model variables **between two scenarios**!

- Worksheet *Economy_AbsDiff*: employment and real production by economic sectors for a single year in absolute values
- Worksheet *Economy_%Diff*: employment and real production by economic sectors for a single year in relative values
- Worksheet *EnergyEmissions*: absolute and relative differences for emissions and the energy balance
- Worksheet *4Variables_TimeSeries*: displays up to four selected model variables
- Worksheet *ScCompareFig_TimeSeries*: Employment by economic activities and GDP components in absolute and relative differences as time series (2022 to 2050)

The selection of a specific year is done in the worksheet *SelectScenarios* (see red rectangle in Figure 37) and is valid for all graphs that are referring to a single year. Once a new year is selected, all graphs are refreshed automatically.

Further evaluation sheets / dashboards with figures and tables based on the results in the worksheets *DiffAbs* and *Diff%* can be created by the user. For example, a set of indicators (e. g. output, employment) can be considered in a dashboard for only one economic sector (e. g. agriculture, construction) which is in focus.



Comparison of 2 Scenarios

Please select year for evaluation

2023

1. Step: UPDATE List of Scenarios

2. Step: READ and COMPARE Results

Please select your REFERENCE scenario here

- Bau.xlsb
- DroughtDzud_adapt.xlsb
- DroughtDzud_agric.xlsb

Please select your ALTERNATIVE scenario here

- Bau.xlsb
- DroughtDzud_adapt.xlsb
- DroughtDzud_agric.xlsb

Execution time: 31.1

YOUR SELECTION

Scenario 1: *Bau*

Scenario 2: DroughtDzud_agric

Please list all variables which should be evaluated

Include List

gdper_v
gdped_v
gdpen_v
awen_s
empl_s
empl_v
dfdn_m
gobpir_v
gobpid_v
gobpin_v
gobpir_s
gobpid_s
eb_m
ghge_v
ep_v
agobpin12_v
agobpir12_v
agobpid12_v
agobpin5_v
agobpir5_v
agobpid5_v
imfdn_v
imicn_v

Figure 37: Evaluation tool _Compare2Scenarios_.xlsb

Source: Own illustration.

Additionally, for both selected scenarios the absolute values (NO differences!) are stored in the “*Scenario1”* resp. “*Scenario2”* sheets. Thus, the model user can create a graph for ONE model variable including the development over time in each scenario and the difference (resp. deviations) between the two scenarios.

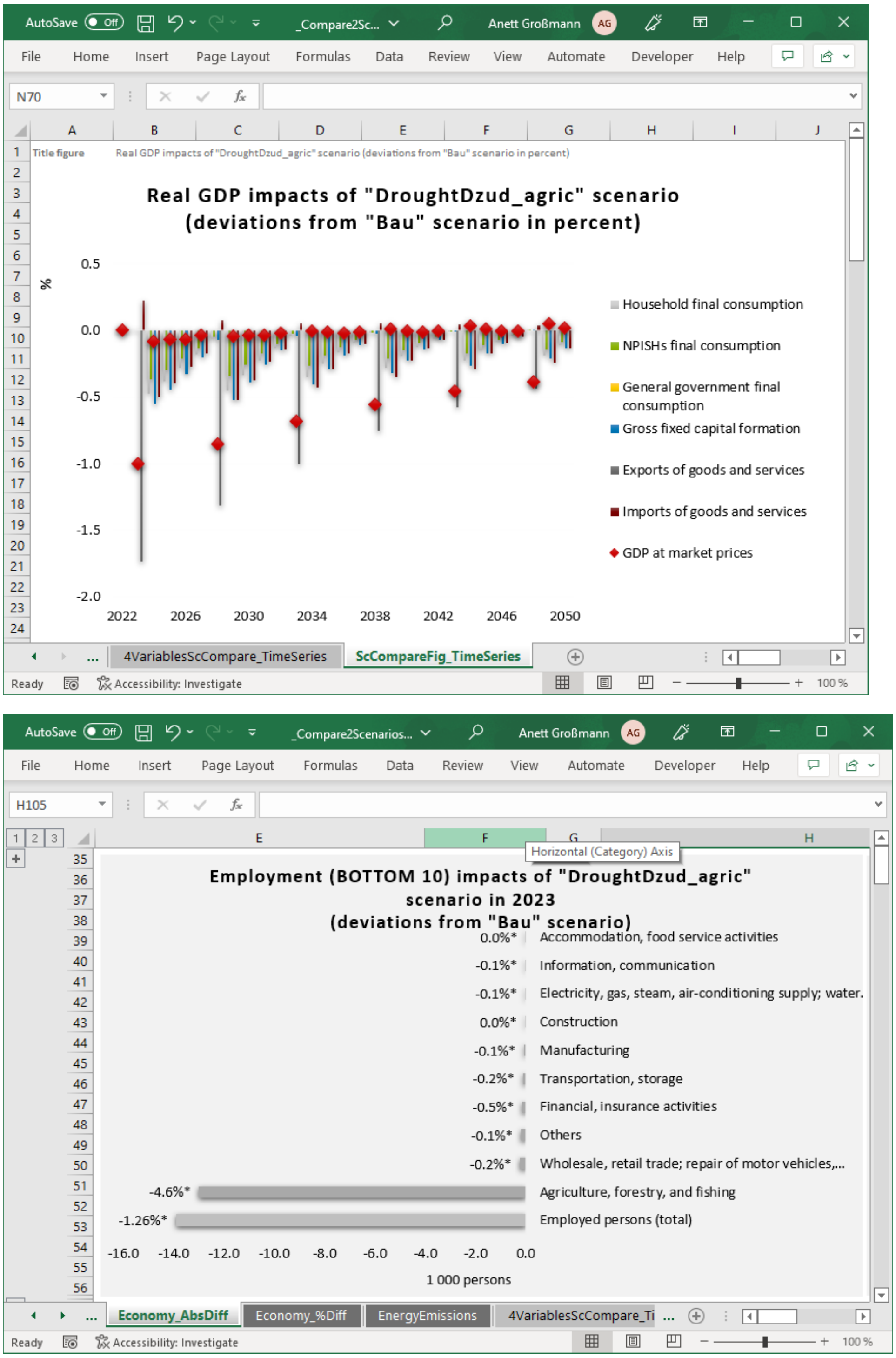

AutoSave Off
_Compare2Sc...
Anett Großmann AG
File Home Insert Page Layout Formulas Data Review View Automate Developer Help
N70
Title figure
Real GDP impacts of "DroughtDzud_agric" scenario (deviations from "Bau" scenario in percent)
Real GDP impacts of "DroughtDzud_agric" scenario
(deviations from "Bau" scenario in percent)
%
0.5
0.0
-0.5
-1.0
-1.5
-2.0
2022
2026
2030
2034
2038
2042
2046
2050
Household final consumption
NPISHs final consumption
General government final consumption
Gross fixed capital formation
Exports of goods and services
Imports of goods and services
GDP at market prices
4VariablesScCompare_TimeSeries
ScCompareFig_TimeSeries
Ready
Accessibility: Investigate
100 %
AutoSave Off
_Compare2Scenarios...
Anett Großmann AG
File Home Insert Page Layout Formulas Data Review View Automate Developer Help
H105
Horizontal (Category) Axis
Employment (BOTTOM 10) impacts of "DroughtDzud_agric"
scenario in 2023
(deviations from "Bau" scenario)
0.0%*
Accommodation, food service activities
-0.1%*
Information, communication
-0.1%*
Electricity, gas, steam, air-conditioning supply; water.
0.0%*
Construction
-0.1%*
Manufacturing
-0.2%*
Transportation, storage
-0.5%*
Financial, insurance activities
-0.1%*
Others
-0.2%*
Wholesale, retail trade; repair of motor vehicles,...
-4.6%*
Agriculture, forestry, and fishing
-1.26%*
Employed persons (total)
-16.0
-14.0
-12.0
-10.0
-8.0
-6.0
-4.0
-2.0
0.0
1 000 persons
Economy_AbsDiff
Economy_%Diff
EnergyEmissions
4VariablesScCompare_Ti ...
Ready
Accessibility: Investigate
100 %

Figure 38: *Selected examples of comparative evaluation of two scenarios*

Source: Own illustration.

For the evaluation of the “DroughtDzud_agric” scenario, the reference scenario (hypothetical no climate change scenario) is selected as the reference and the “DroughtDzud_agric” scenario as the alternative scenario. The preconfigured evaluation tool is then used for doing the macroeconomic analysis of the EWE to detect the inter-sectoral and overall effects.

For the evaluation of an adaptation measure, the climate change scenario is selected as the reference and the adaptation scenario as the alternative scenario. Again, the preconfigured evaluation tool is then used analyzing the macroeconomic and inter-sectoral effects of the hazard- and sector-specific adaptation measure.

The exemplary results of the selected EWE are presented during the on-site training sessions in December 2022 and outlined in a short report.

References

Ailtgui (2022): Available studies on climate risks and vulnerability assessments and implemented / planned adaptation measures for priority sectors of economy in Mongolia. Unpublished internal report.

European Communities, International Monetary Fund, Organisation for Economic Co-operation and Development, United Nations and World Bank (2009). System of National Accounts 2008. New York.

Großmann, A. & Hohmann, F. (2019): Static and Dynamic Input-Output Modelling with Microsoft Excel. SHAIO Conference Paper 2019, Osnabrück. https://downloads.gws-os.com/GrossmannHohmann2019_DIOM-X.pdf (last accessed at November 29th)

GIZ, 2022a (Großmann, A., Hohmann, F., Lutz, C. & Reuschel, S.): Supporting Climate Resilient Economic Development in Kazakhstan. Application of the e3.kz Model to Analyze the Economy-wide Impacts of Climate Change Adaptation, Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH, Berlin.

GIZ, 2022b (Großmann, A. & Hohmann, F.): Macroeconomic Modelling for Climate Policy Planning. Impact Analysis with an Excel-based E3 (Economy-Energy-Emission) Model Building Framework, Bonn. https://downloads.gws-os.com/giz-2022-EN-CRED-macroeconomic-modelling-for-climate-policy-planning.pdf (last accessed at November 29th)

Navarro, J. S., Jordà, G. (2021). Climatic hazards evolution in the 21stcentury in the frame of the CRED project. Mediterrane-an Institute for Advanced Studies (IMEDEA, UIB-CSIC), University of the Balearic Islands (UIB)

Pollitt, H., Lewney, R., Mercure, J.-F., 2019. Conceptual differences between macro-econometric and CGE models. Paper presented at the 27th IIOA conference. https://www.iioa.org/conferences/27th/papers/files/3597_20190430081_IIOApaperE3MEsession.pdf (last accessed at November 29th)

United Nations (UN) Department of Economic and Social Affairs, Population Division (2022). World Population Prospects 2022, Online Edition.

United Nations (UN) (2018). Handbook on Supply and Use Tables and Input-Output Tables with Extensions and Applications. Department of Economic and Social Affairs Statistics Division. Studies in Methods. Handbook of National Accounting, Series F No. 74, Rev. 1. https://unstats.un.org/unsd/nationalaccount/docs/SUT_IOT_HB_Final_Cover.pdf (last accessed at November 29th)

As a federally owned enterprise, GIZ supports the German Government in achieving its objectives in the field of international cooperation for sustainable development.

Published by:
Deutsche Gesellschaft für
Internationale Zusammenarbeit (GIZ) GmbH

Registered offices:
Bonn and Eschborn, Germany

Address:
Deutsche Gesellschaft für
Internationale Zusammenarbeit (GIZ) GmbH
Köthener Str. 2
10963, Berlin, Germany
T +49 61 96 79-0
F +49 61 96 79-11 15
E info@giz.de
I www.giz.de/en

Programme description:
IKI Global Programme on
Policy Advice for Climate Resilient Economic Development (CRED)

Project Director:
Stefanie Springorum
stefanie.springorum@giz.de

This programme is part of the International Climate Initiative (IKI). The Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV) supports this initiative on the basis of a decision adopted by the German Bundestag.

Authors:
Dr. Anett Großmann, GWS (www.gws-os.com)
Frank Hohmann, GWS (www.gws-os.com)

Concept & Design:
Atelier Löwentor GmbH, Darmstadt

Layout:
Anne Weltin, GIZ
Tom Stadler, GIZ

URL links:
Responsibility for the content of external websites linked in this publication always lies with their respective publishers. GIZ expressly dissociates itself from such content.

The contents of this handbook are the sole responsibility of the authors and can in no way reflect the official opinion of the GIZ global program.

Germany, 2023