

Contents

1. Overview
2. Data Availability
3. Instructions for Replicators
4. List of Exhibits
5. Requirements
6. Code Description
7. Folder Structure

Overview

This package includes information needed to reproduce the Charts and Tables in *The World Bank's MFMod Framework in Python with Modelflow*. It is comprised of the data set used in the report and from which all the tables and charts are derived, the Jupyter Notebooks that perform the simulations and generate the graphics and tables in the report as well as a python build script that runs all of the Jupyter notebooks and outputs the Charts and Tables and binds them into a PDF which is the actual book. The Jupyter Notebooks can be run independently or in the python build program.

Data Availability

The data used for this report are drawn from the MFMod modelling system Burns et al. (2019), and are being released simultaneously with the report on the World Bank's github site.

Data Sources

Statement about Rights

We certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

Instructions for Replicators

To run the package, replicators will need to have anaconda Python version 3.13 installed, and the ModelFlow Package. To build the PDF file from the package using the build.py program they will also need to have installed the Jupyter Book package and a version of Latex.

The book and all files were tested using Miniconda and Anaconda versions of ModelFlow under windows. Full instructions on the installation of ModelFlow are included in Chapter 3 of the report, the technical details of which are reproduced in the Requirements section below.

List of Exhibits

All exhibits generated by the Jupyter Notebooks that form the book and these books have all of the code for all of the outputs. Figures and tables are not numbered per se. However, the code that generates them appears just before the output itself. Although the notebooks are meant to be executed consecutively to produce the book, each also can be executed independently. One exception to this is the notebook for Chapter 14, which takes as an input a file generated in Chapter 11. Therefore it will be necessary to run Chapter 11 before running Chapter 14.

Below is a list of notebooks in the package.

name	path	note
Foreword	introduction.ipynb	
1 Introduction	01_Introduction/Introduction.ipynb	
2 Macrostructural models	02_MacroStructuralModels/MacroStructuralModels.ipynb	
3 Installation	03_Installation/InstallingModelFlow.ipynb	
3.1 Installation of Python	03_Installation/InstallingPython.ipynb	
3.2 Installation of	03_Installation/InstallingModelFlowpackage.ipynb	
ModelFlow		
3.3 Updating ModelFlow	03_Installation/UpdateModelFlow.ipynb	
4 Introduction to Jupyter Notebook	04_PythonEssentials/Intro_Jupyter_notebook.ipynb	
5 Some Python basics	04_PythonEssentials/PythonBasics.ipynb	
6 Introduction to Pandas, Series and dataframes	04_PythonEssentials/PandasDataFrames.ipynb	
7 ModelFlow and Pandas DataFrames	04_PythonEssentials/Modelflow and Dataframes.ipynb	
7.1 The <code>.upd()</code> method returns a DataFrame with updated variables	04_PythonEssentials/UpdateCommand.ipynb	
7.2 The <code>.mfcalc()</code> method. Return a dataframe with transformed variables	04_PythonEssentials/mfcalc.ipynb	
8 Using ModelFlow with World Bank models	05_WBModels/AccessingWBModels.ipynb	
9 Working with a World Bank Model under ModelFlow	05_WBModels/WorkingwWBModels.ipynb	
10 Equations in MFMod and ModelFlow	05_WBModels/BehavioralEquations.ipynb	
11 Scenario analysis	05_WBModels/ScenarioAnalysis.ipynb	
12 More complex scenarios	05_WBModels/MoreComplexScenarios.ipynb	

name	path	note
13 A simulation that targets a specific outcome	05_WBModels/Targeting.ipynb	
14 Report writing and scenario results	05_WBModels/ReportWriting.ipynb	
15 Model structure and causal chains	06_ModelAnalytics/ModelStructure.ipynb	
16 Analyzing the impact of a shock	06_ModelAnalytics/AttributionSomeFeatures.ipynb	
17 Getting Help	07_MoreFeatures/GettingHelp.ipynb	
18 Modelflow methods reference	07_MoreFeatures/ModelFlowReference.ipynb	

Requirements

Computational Requirements

Although it is not strictly necessary, it is a good idea to create a specific python environment for `ModelFlow`, this will be a space where the `ModelFlow` package itself will be installed and in which the specific dependencies on which `ModelFlow` relies will be installed. Other environments may require packages that conflict with `ModelFlow`. Declaring a separate environment for `ModelFlow` will help prevent conflicts between different versions of packages from arising.

`ModelFlow` will install a range of dependencies, including Jupyter Notebook 6, pandas and matplotlib. `Modelflow` runs under Notebook 7 but the book and reproducibility package were built and tested under Jupyter Notebook 6.

Software Requirements

- **Python 3.13**
 - Modelflow 2.59
 - Jupyter Notebook 6
- Jupyter Book
- MikTex or TexLive

Installing Python MiniConda is a minimalist implementation of python. A guide for installing **MiniConda under windows** can be found here.

Installing Modelflow The commands below create a `ModelFlow` environment under `miniconda` or `anaconda` and install into it the `ModelFlow` package and several supporting packages that are needed when using `ModelFlow` in conjunction with `Jupyter Notebook`.

The commands should be cut and paste (either one by one, or as a block) into the Anaconda command prompt environment.

```
conda create -n ModelFlow -c ibh -c conda-forge ModelFlow_book jupyter-book -y
conda activate ModelFlow
```

```
pip install dash_interactive_graphviz
jupyter contrib nbextension install --user
jupyter nbextension enable hide_input_all/main
jupyter nbextension enable splitcell/splitcell
jupyter nbextension enable toc2/main
jupyter nbextension enable varInspector/main
```

Installing Jupyter Book The Jupyter book library is needed if the replicator wishes to generate the full publication, text as well as figures and tables.

Installing Tex To generate the book or the example report in PDF a latex package must also be installed on the computer, either the Miktex or the TexLive can be used and have been tested with the package. Installation instructions are provided on the linked sites.

Memory and Runtime and Storage Requirements

The build.py program should run and compile on a Windows 11 machine with an Intel 5 processor with 32G memory installed within 15 minutes.

Total storage required (after installation of python and required packages) is less than 500MB.

Replication Code Description

This project contains a set of Python scripts and Jupyter Notebooks that together replicate the analyses and figures presented in the book.

Unpack the mfbook.zip File

Once the Python environment has been set up, you are ready to replicate the book. All necessary files for replication are contained in the `mfbook.zip` archive.

Unzip this file to extract the full project structure. The core content is organized into a series of Jupyter Notebooks (`.ipynb` files), which are located in (subfolders)[#Folder Structure] under:

```
mfbook/content/
```

Creating the Jupyter Book

To build the Jupyter Book, open an **Anaconda Prompt** and activate the relevant Python environment:

```
conda activate ModelFlow
```

navigate to the directory where the replication package has been unzipped.

The main script used to compile the book is `build.py`. This script executes all the Jupyter Notebooks, generating all tables and charts used in the report. Note that interactive widgets are **not** included in the final output; instead, screenshots of the widgets are shown. You can run the corresponding notebook manually to interact with the widget and verify that the screenshot correspond to elements of the widget displayed.

- `build.py`: Executes all notebooks, compiles them to LaTeX, and generates an HTML version of the book located in:

```
mfbook/_build/html/
```

To build both the HTML and PDF versions of the book, run:

```
python build.py latex
```

If LaTeX is not installed on your system, the PDF will not be generated.

Running from Reproducibility README.ipynb

You may also run each notebook manually using Jupyter. To launch the Jupyter Notebook interface, enter the following in the Anaconda Prompt:

```
conda activate ModelFlow
jupyter notebook "Reproducibility README.ipynb"
```

Working from this Jupyter Notebook, each notebook listed above can then be clicked upon and executed directly.

Running Jupyter Notebooks Individually

You may also run each notebook manually using Jupyter. To launch the Jupyter Notebook interface, enter the following in the Anaconda Prompt:

```
conda activate ModelFlow
jupyter notebook
```

From there, navigate to `mfbook/content/` and select the notebook you wish to run.

Note: You should run
11 Scenario analysis
before
14 Report writing and scenario results,
as it produces a file required by the report writing step.

Note: During execution a few intermediate model files are created. These are store in “mfbook/content/05_WBModels/WorldBankModels/{country}/data” and can be safely be ignored. These files are not part of the zipped Reproducibility package, but are generated when running the Jupyter Notebooks. They are used among other things to save results generated by one Jupyter Notebook for later use in a second Jupyter Notebook. For example, the results from the simulations in Chapter 11 Scenario analysis are stored here and read in by the system in Chapter 14 Report writing and scenario results.

Folder Structure

```
Reproducibility README.md
Reproducibility README.ipynb
build.py
modelutil_cli.py
mfbook
  _config.yml
  _toc.yml
  genindex.md
  Reference.md
  reference.bib
  content
    01_introduction
    02_MacrostructuralModels
    03_Instalation
    04_PythonEssentials
    05_WBModels
    06_ModelAnalyticsintroduction
    07_MoreFeatures
  models
  introducion.ipynb
  Overview.ipynb
```