GTAPinR Course

2025-09-22

## Frequently Asked Questions (FAQ)

### About the Course & Why R

###### What is the GTAP Model and why is it relevant for this course?

|  |  |
| --- | --- |
|  | **Answer**  The GTAP Model is a computable general equilibrium framework widely used to analyze trade policy, energy and environmental questions, poverty, and migration. Although the model specification is open, the traditional workflow historically relied on proprietary solvers. This course shows how to conduct the same class of analyses in R, so that you can integrate data processing, modeling, and visualization within a single, open-source environment. For step-by-step examples of the workflow, please consult the course website. See more at the <https://tsimonato.github.io/GTAPinR_Course/#setting-up> |

##### Why should I run GTAP in R instead of using proprietary software?

|  |  |
| --- | --- |
|  | **Answer**  Running GTAP in R allows you to avoid licensing constraints, run on local machines or in the cloud without license servers, and maintain reproducible pipelines that keep data handling, modeling, and plotting in one place. You also gain access to a large open-source ecosystem and community. For practical demonstrations of this approach, including code and screenshots, refer to the course pages. See also the <https://tsimonato.github.io/GTAPinR_Course/#setting-up> |

##### Which packages form the core of the GTAPinR approach?

|  |  |
| --- | --- |
|  | **Answer**  The workflow hinges on two packages. HARr enables reading and writing GTAP header-array and solution files directly within R, whereas tabloToR translates TABLO models, connects the model code to the GTAP Data Base, defines the closure, applies shocks, solves, and exposes results for inspection and plotting. Additional helper packages are used in the course materials; see the website for installation commands and usage examples. See package details in the <https://tsimonato.github.io/GTAPinR_Course/#setting-up> |

### Requirements & Setup

##### What software setup do you recommend before starting?

|  |  |
| --- | --- |
|  | **Answer**  You should use a recent version of R and RStudio on Windows, macOS, or Linux, and you should be able to install packages from GitHub. The course’s “Setting Up” section guides you through installing dependencies, configuring your working directory, and verifying that the model files and database are discoverable by R. Please follow that section for exact commands and example scripts. Detailed instructions are in the <https://tsimonato.github.io/GTAPinR_Course/#setting-up> |

##### How do I install the key packages?

|  |  |
| --- | --- |
|  | **Answer**  You can install the packages using devtools::install\_github(). For example:  install.packages("devtools") devtools::install\_github("USDA-ERS/MTED-HARr") devtools::install\_github("tsimonato/MTED-TabloToR")  Check your working directory and ensure the model subfolder has the necessary files. The course website includes detailed steps and troubleshooting advice. See installation steps in the <https://tsimonato.github.io/GTAPinR_Course/#setting-up> |

### R Data Structures

##### Which data structures in R will I actually use for GTAP work?

|  |  |
| --- | --- |
|  | **Answer**  You will primarily use data frames for tabular manipulation, arrays for multi-dimensional numeric information that mirrors GTAP headers and solution blocks, and lists that act as containers for these objects. Lists are the natural way to store the collection of headers loaded from a header-array file, whereas arrays preserve dimensional structure for efficient slicing and aggregation. The “Data Structures in R” portion of the course shows concrete indexing patterns and transformations. Examples are provided in the <https://tsimonato.github.io/GTAPinR_Course/#exploring-the-data> |

##### How do I read GTAP .har and .sl4 files into R?

|  |  |
| --- | --- |
|  | **Answer**  You can load header-array files using HARr::read\_har("path/to/file.har"), which returns a list whose elements correspond to the headers in the file, with numeric headers stored as arrays. Solution files can be read similarly. After loading, you can call summary() on the returned object to list available headers, access a specific header with the dollar operator, and subset numeric arrays with standard R indexing. The course includes runnable examples. See details in the <https://tsimonato.github.io/GTAPinR_Course/#loading-the-gtap-data-files> |

### HARr

##### How do I explore and convert arrays for analysis?

|  |  |
| --- | --- |
|  | **Answer**  You can use summary() to inspect headers, extract arrays with $, and subset with standard R indexing. To convert arrays for plotting, use as.data.frame.table(). This produces a tidy format suitable for filtering and graphing. The course shows practical examples and how to integrate with ggplot2 or ggalluvial for visualization. Examples are shown in the <https://tsimonato.github.io/GTAPinR_Course/#exploring-the-data> |

##### How do I write new .har files from R?

|  |  |
| --- | --- |
|  | **Answer**  You can use the function HARr::write\_har() to create new header-array files. Sets should be stored as character vectors, numeric data as arrays with properly defined dimensions, and each header should include a short code and a long description. The course materials demonstrate how to prepare and validate these objects before writing. See instructions in the <https://tsimonato.github.io/GTAPinR_Course/#writing-data> |

### tabloToR

##### What is a GEModel object and why do I need it?

|  |  |
| --- | --- |
|  | **Answer**  The GEModel object is the container for a single experiment. It holds the parsed TABLO model, the database built with HARr::read\_har, the closure, shocks, the solver, and the results. The course demonstrates how to create and use GEModel step by step. See model initialization in the <https://tsimonato.github.io/GTAPinR_Course/#initializing-the-gtap-model> |

##### What is a closure and how do I specify it?

|  |  |
| --- | --- |
|  | **Answer**  The closure defines which variables are exogenous and which are endogenous. In tabloToR, you set exogenous variables and initialize their shocks, typically starting from zero. The course provides examples of closures and explains how to switch them depending on your research scenario. See closure examples in the <https://tsimonato.github.io/GTAPinR_Course/#setting-up-closure-and-shocks> |

##### How do I apply a policy shock and solve the model?

|  |  |
| --- | --- |
|  | **Answer**  After loading the TABLO file and database into a GEModel and defining a closure, you can assign a value to an exogenous variable as a shock and run the solver. For example, you could model a productivity increase by setting the relevant variable exogenous, applying a positive change, and then calling the solve function. The course demonstrates this with hands-on exercises. Detailed workflow is in the <https://tsimonato.github.io/GTAPinR_Course/#setting-up-closure-and-shocks> |

##### Where are results stored and how do I save them?

|  |  |
| --- | --- |
|  | **Answer**  After solving, results are stored in the outputData list inside the GEModel. You can inspect updated arrays directly or convert them to data frames for plotting. To persist results, you can use saveRDS() and reload them later with readRDS(). The course shows practical examples of saving, reloading, and comparing results across runs. See outputs in the <https://tsimonato.github.io/GTAPinR_Course/#solving-and-interpreting-the-results> |

### Troubleshooting

##### Why am I getting an error when installing packages from GitHub?

|  |  |
| --- | --- |
|  | **Answer**  Installation errors often stem from missing system build tools (Rtools on Windows, Command Line Tools on macOS), firewall or proxy restrictions, or not restarting R after installing toolchains. The course’s setup section provides guidance for resolving these issues. |

##### Why can’t I open a .har file that I am certain exists?

|  |  |
| --- | --- |
|  | **Answer**  The most common cause is a mismatch in working directory or file path. You should check the current working directory with getwd() and confirm the path. Encoding issues in header names may also cause failures. The course offers methods to test file paths and confirm encoding. Check file handling in the <https://tsimonato.github.io/GTAPinR_Course/#loading-the-gtap-data-files> |

##### Why does tabloToR fail to solve or complain about variables not found?

|  |  |
| --- | --- |
|  | **Answer**  This typically means that the closure is inconsistent with the TABLO file, or a shock was applied to a variable that was not defined as exogenous. Re-check variable names in the TABLO file and ensure that your closure aligns with your shocks. The course includes troubleshooting checklists for these cases. See guidance in the <https://tsimonato.github.io/GTAPinR_Course/#setting-up-closure-and-shocks> |

### Advanced

##### Can I compare tabloToR results with GEMPACK?

|  |  |
| --- | --- |
|  | **Answer**  Yes. You can read GEMPACK .sl4 files using HARr, convert them to data frames, and compare directly with tabloToR output using the same conversion. When the same closure and solution method are applied, the results are highly comparable. The course illustrates this process and highlights small differences you may encounter. Comparison examples are provided in the <https://tsimonato.github.io/GTAPinR_Course/#solving-and-interpreting-the-results> |

##### Can I extend the workflow to baselines or regional models?

|  |  |
| --- | --- |
|  | **Answer**  The package has been developed and tested specifically with the standard GTAP version 7 model. While it is technically possible that the workflow could be adapted for other models or for extensions such as baseline construction and regional disaggregation, these uses have not been the focus of development and have not been systematically tested. Users attempting such applications should be aware of these limitations. |