



Tech Clinic: Programming in *Julia*

The Data Lab for Climate and Nature of the Hoffmann Centre for Global Sustainability invites researchers and practitioners working on sustainability to participate in a one-day Tech Clinic on programming in Julia, a computationally fast and energy-saving software.

The Hoffmann Centre for Global Sustainability (HCGS) is a leading research and policy hub in Europe dedicated to promoting integrated sustainability that is both nature positive and socially equitable. The HCGS works across seven thematic tracks and a cross-cutting Data Lab for Climate and Nature. The Lab fosters different initiatives that can improve the collection, curation, analysis and dissemination of data for nature.

What is a Tech Clinic?

Tech Clinics consists of a series of lectures, organized as an intensive, in-person course, presenting econometric methods, tools and techniques that are relevant for research and decision making on sustainability issues.

The Clinics are technical in nature and taught by leading econometricians, statisticians and/or data scientists. The target audience involves doctoral researchers enrolled in programs in Switzerland, selected advanced master students and pre-doctoral researchers with the relevant background, postdoctoral researchers, young faculty, practitioners, applied researchers and technical staff in Swiss-based international organizations.

Tech Clinics focus on the development of applicable skills and hands-on practice. If suitable, lectures are complemented by in-class coding demonstrations, the provision of



take-home coding or applied exercises, and/or the exposition of some "how to" modules within the lectures.

There is no exam or assessment of participants' knowledge. The HCGS issues a certificate of attendance to all participants present in the entirety of the clinic.

Why a Tech Clinic on programming in *Julia*?

Working with data for nature—including high-resolution satellite imagery, global climate model outputs, biodiversity occurrence records, ecosystem monitoring networks, and environmental sensor arrays—presents unique computational challenges that traditional software and workflows often cannot efficiently address. These datasets frequently involve terabytes of multidimensional arrays, require processing at fine spatial and temporal resolutions (often sub-kilometer grids across continental scales), and demand intensive statistical modeling, machine learning, and simulation techniques.

Julia combines the ease of high-level languages with the performance of low-level compiled code. As such, its computational efficiency translates into energy-efficiency: see [Pareira et al. \(2017\)](#) and [Stewart et al. \(2025\)](#) for efficiency comparisons, some of which indicate *Julia* outperforms alternatives like Python in terms of efficiency by a factor of 40!

As *Julia* is an open-source language, the computational tools taught in the Clinic remain accessible to researchers and practitioners, regardless of institutional resources or geographic location, fostering global collaboration in addressing urgent social and environmental challenges.

What topics will be covered?

- Introduction
- Basics and scalars (object creation, basic scalar operations, object types, packages)
- Functions
- Vectors



- Matrices, arrays and other data-structures (incl. data frames)
- Conditionals, loops and variable-scopes
- Path-/Folder-management
- Storing and loading data (incl. csv- and excel-files)
- Random Variables & Densities
- Econometrics
- Plotting
- Workspace management
- Parallelization
- Time-permitting, further topics (error handling, timing, interpolation, optimization, numerical integration, symbolic calculus, markdown-integration, Python-integration, ...)

Who is the instructor?

This Tech Clinic will be taught by [Prof. Marko Mlikota](#), Assistant Professor in Economics at the Geneva Graduate Institute, PhD UPenn.

When and where will the Tech Clinic take place?

The Clinic will take place on Friday 10 October 2025, in **room S2** the Maison de la Paix of the Geneva Graduate Institute ([how to get to IHEID](#)). The Clinic will start at 8.15 AM and participants are asked to arrive 10 minutes before the start.

The teaching plan for the day is as follows:

08.05: Arrival of participants and check-in

08.15 – 10.00: Teaching block 1

10.00 – 12.00: Break / Individual study time in public access areas

12.15 – 14.00: Teaching block 2



14.00 – 16.00: Break / Individual study time in public access areas

16.15 – 18.00: Teaching block 3

What is the cost of the Tech Clinic?

This Tech Clinic is fully subsidized by the Hoffmann Centre for Global Sustainability.

Refreshments, meals, transport and accommodation are not included and are thus at the expense of participants.

How do participants enroll in this Tech Clinic?

Interested participants should fill in this form <ADD HYPERLINK> before 10 September 2025.

Places are limited, and enrollment will be confirmed or declined by the HCGS by email. This confirmation email needs to be presented when checking in on arrival at the Tech Clinic.

What pre-requisites are participants meant to fulfil?

There are no formal pre-requisites. However, undergraduate level knowledge of mathematics, statistics and probability will be assumed.

Participants will need to attend the Tech Clinic with a functional laptop. They will need to install the software Julia and the editor Visual Studio Code before the workshop. See instructions below.

You can find plenty further materials helpful for working with Julia on the software's official website: <https://docs.julialang.org/en/v1/>.

Instructions for setting up *Julia*



1. Go to <https://julialang.org/downloads/> and download Julia and install Julia. Downloading Julia manually from this page works for both Mac- and Windows-users.¹
2. Go to <https://visualstudio.microsoft.com/downloads/> and download Visual Studio Code (not Visual Studio!). VS Code is an editor, with which one can write, store and execute Julia code (akin to the editor RStudio commonly used to work with the software R).
3. In VS Code, go to Code>Settings>Extensions, and install the extension “julialang”. Once installed, the extension should automatically locate Julia on your computer. To verify whether it worked for you, open a new file, store it as “myFirstJuliaCode.jl” (the extension .jl will tell VS Code that the file is code written for Julia), type the following few lines into it --

```
a = 3  
b = 4  
a/b  
a*b
```

-- and execute it either by highlighting the whole text and pressing shift+enter or by clicking on the play button on the top right. In case VS Code did not find Julia automatically, go to Code>Settings>Extensions and read the instructions on how to locate Julia manually by specifying the path to it.

¹ For Mac users, it would be recommended to install Julia in a way that will make updates much easier to carry out, without the need to repeat all the steps mentioned above to install a new version: see <https://julialang.org/install/>. You're welcome to try to do so, but then the instructions below may need additional steps before Julia can be used on VS Code.