Teaching best-practice of open-source collaborative scientific software development

Panel 3: “Educating Future Macro-Energy Systems Researchers“
Macro-Energy Systems Workshop hosted by Stanford Energy

Daniel Huppmann

September 18, 2020

Background

Starting a new course from scratch…

Two years ago, I was asked to create a new course at TU Wien on "open-source energy system modelling"…

TECHNISCHE UNIVERSITÄT WIEN

Overview and teaching goals:

⇒ Introduction to scientific programming: open-source software & FAIR data

What is it, why do we it, how do we do it?

⇒ Assessment of climate change & sustainable development

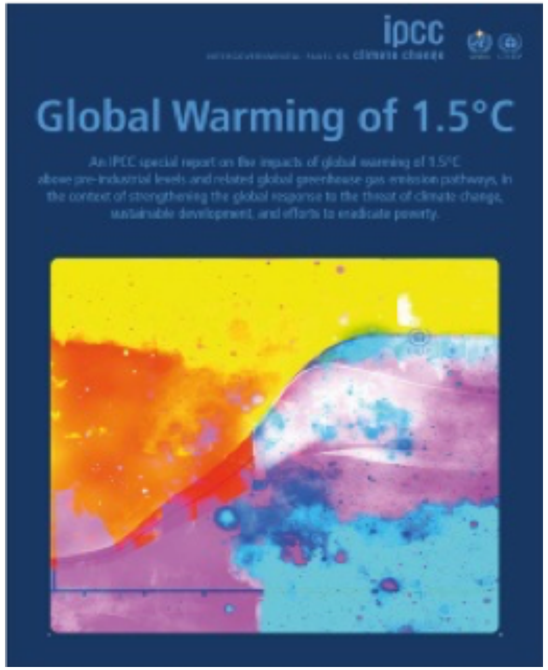
How can scenarios be used for analysis in line with open science?

⇒ ~~Development of a stylized national energy system model~~

~~How can we develop our own scenarios to analyse policy measures?~~

Not the topic of this talk…

The second part of the lecture is based on my work for the *IPCC Special Report on Global Warming of 1.5°C*
www.ipcc.ch/sr15



Best practice in scientific software development

Concepts that should be familiar to any macro-energy modeller

Overview of concepts discussed in the course (part 1)

- ⇒ Open-source licenses
- ⇒ FAIR principles: Findable, Accessible, Interoperable, Reusable
- ⇒ Git version control & code review (using Github or similar services)
- ⇒ Style guides and documentation
- ⇒ Semantic versioning
- ⇒ Testing and "continuous integration"

Homework assignment (for part 1):

- ⇒ Student uses some code snippet from her/his prior work
- ⇒ Add continuous-integration features (unit testing and style guide validation)

A one-slide guide to open & FAIR research

Even accomplished researchers aren't always up to speed…

DOI:
10.22022/ene/04-2020.16404

Please cite as: Daniel Huppmann et al., 2020
Five best-practice steps to make your research open & FAIR v1.0
doi: 10.22022/ene/04-2020.16404 | url: openENTRANCE.eu

Five best-practice steps to make your research open & FAIR v1.0

open ENTRANCE

You may think that putting your work* on a website already makes it free & open.
But that's not quite true – follow these steps to implement best practice of ***#openscience***!

* data sets, text, tables, figures & illustrations, source code, scientific software, … even #Horizon2020 deliverables

1. Open — If you want your ***work to be read, used & shared by others***, be explicit about it: For text, data, figures, … – use the CC-BY license | For code, visit choosealicense.com

2. Findable — To make it easy for others to find and cite your work, get a digital object identifier (DOI) and add a ***recommended citation***

3. Accessible — Depositing your work in an institutional repository or a service like zenodo ensures that your work is still ***available even after the end of the project***

4. Interoperable — Using established community standards, data formats and software packages lets others ***quickly understand and use your work***

5. Reusable — To make it easy for others to ***build on your work***, make sure to assign a version number and relevant (machine-readable) metadata

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 835896

Rationale for open-source research

One can only stand on the shoulders of giants if they let you…

- Science has to be built on transparency and reproducibility
- Describing methods/algorithms with words is a sub-optimal way to share ideas
- Evolution of sharing & openness in the energy modelling community:

🙄 "Available upon reasonable request"

😐 Dumping the final version of a model or scientific software on an institutional server or a service like [GitHub]() or [Zenodo]()

🤩 Separating the model or scientific software into several modules and making parts open-source as early as possible, writing comprehensive documentation & tutorials, etc.

Rationale for best-practice scientific programming

Following best-practice principles in your work
*will give you **more time** to do **better research***

Modelling and scientific analysis is usually a “constant prototyping” exercise

- ⇒ ”Just adding one more feature” often breaks existing functionality
- ⇒ Dependencies (open-source packages) change over time
- ⇒ Models and tools are too complex to immediately notice changed behaviour

Who has not yet experienced the panic & stress
from a model not solving shortly before a deadline…?

Following best-practice principles…

- ⇒ Guards against models and tools failing to work (as expected)
- ⇒ Helps you to understand *your own thinking* a few months later

An example of open & FAIR science

The IPCC SR15 as a case study of open & FAIR scenario analysis

Interactive online scenario explorer at data.ene.iiasa.ac.at/iamc-1.5c-explorer

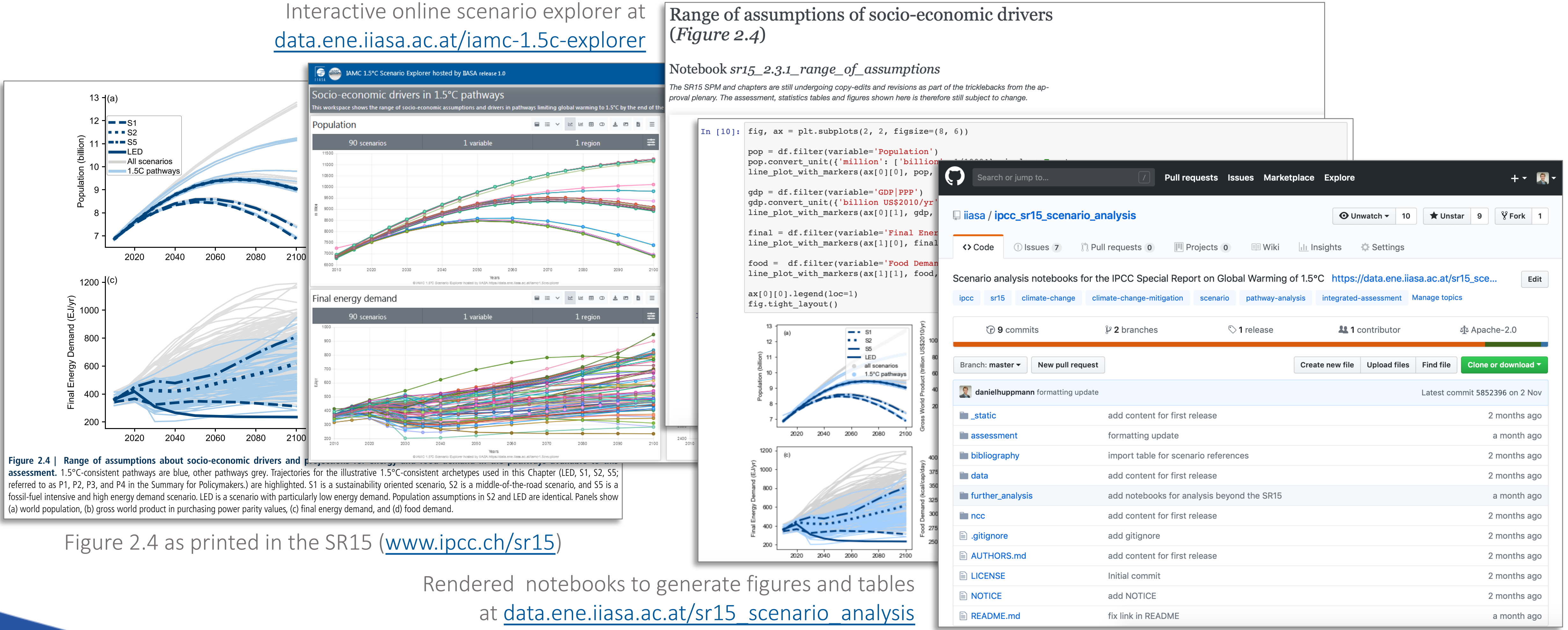


Figure 2.4 as printed in the SR15 (www.ipcc.ch/sr15)

Rendered notebooks to generate figures and tables at data.ene.iiasa.ac.at/sr15_scenario_analysis

```
$ git clone git@github.com:iiasa/ipcc_sr15_scenario_analysis.git
```

The road ahead for macro-energy research

Open science has to go beyond open source…

In an IIASA nexus blog post published today,
I discuss how open-source scientific software and
FAIR data can bring us one step closer to a community of open science.

⇒ https://blog.iiasa.ac.at/

More information on my course at TU Wien (VU 370.062)

⇒ https://tiss.tuwien.ac.at

⇒ https://data.ene.iiasa.ac.at/teaching (including recording of some lectures)

My plea to the audience of this workshop

⇒ Make teaching resources openly available!

#freethelectures

Thank you very much for your attention!

Dr. Daniel Huppmann

Research Scholar – Energy Program

International Institute for Applied Systems Analysis (IIASA)
Laxenburg, Austria

huppmann@iiasa.ac.at

@daniel_huppmann

http://www.iiasa.ac.at/staff/huppmann

Backup slides (for IPCC SR15 analysis)

The IAMC template for timeseries data

A community standard for compiling scenario results

IAMC
Integrated Assessment Modeling Consortium
Founded 2007

The integrated-assessment community (IAMC)
developed a tabular scenario data format for energy systems, SDG dimensions, …

⇒ Used in IPCC Reports (AR5, SR15), Horizon 2020 projects, EMF, …

⇒ Adopted by ~50 teams globally

	A	B	C	D	E	F	G	H
1	**Model**	**Scenario**	**Region**	**Variable**	**Unit**	**2005**	**2010**	**2015**
2	MESSAGE	CD-LINKS 400	World	Primary Energy	EJ/y	462.5	500.7	…

Current and recent projects:

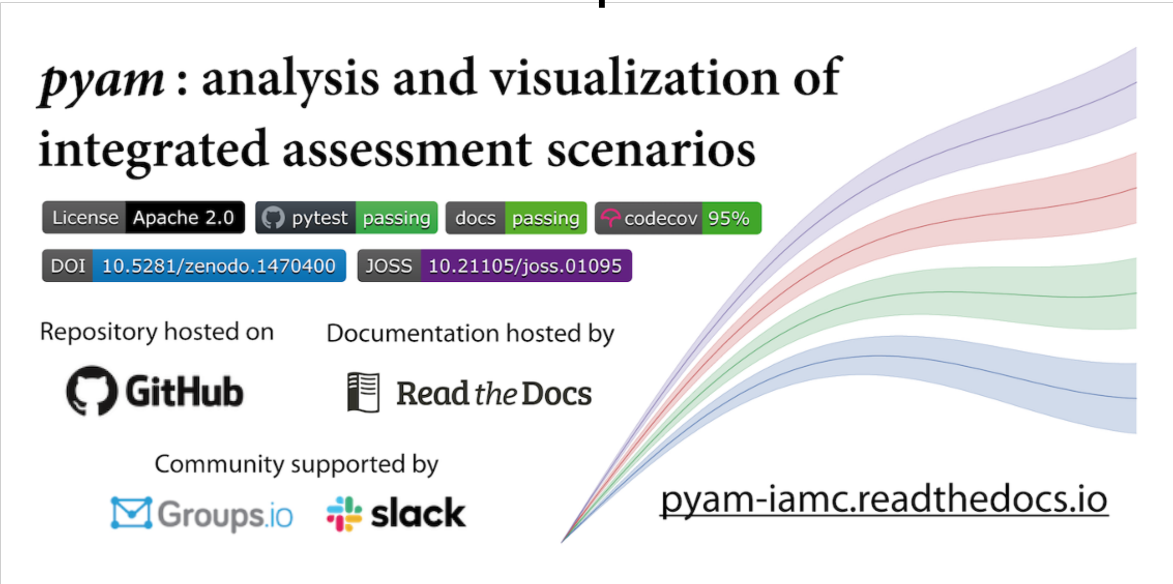
EMF · CD-LINKS · ENGAGE FEASIBILITY OF CLIMATE PATHWAYS · NAVIGATE · open ENTRANCE

pyam: a Python package for scenario analysis

An open-source package to facilitate streamlined processing, validation and assessment of scenario results

python™

- Aim: develop a package of useful functions for scenario analysis & visualization following best practice of collaborative scientific software development
- Features:
  - ⇒ Scenario processing workflow (I/O, units, …)
  - ⇒ Analysis & validation
  - ⇒ Categorization & quantitative indicators
  - ⇒ Visualization features & plotting library



- More information:
  - ⇒ Documentation: pyam-iamc.readthedocs.io
  - ⇒ Scientific reference: M. Gidden and D. Huppmann (2019). Journal of Open Source Software 4(33):1095. doi: 10.21105/joss.01095

#pyam_iamc