Das Python-Paket *pyam* zur Analyse, Validierung & Visualisierung von Energiesystem- und Klimaszenarien

Repository hosted on GitHub

Community supported by Groups.io slack

Documentation hosted by Read the Docs

Daniel Huppmann

Online-Strommarkttreffen – 14. Juli 2021

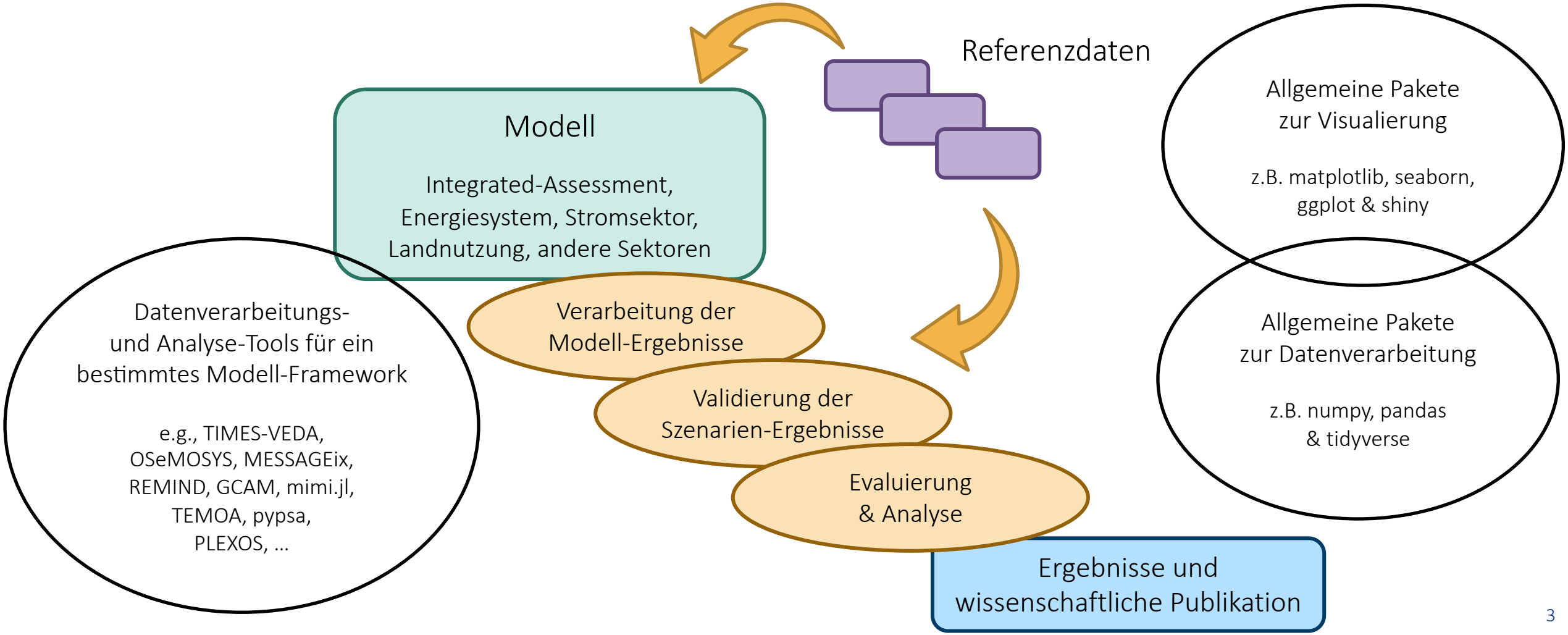
This work has received funding via several grants under the European Union's Horizon 2020 research and innovation programme

Teil 1

Einleitung & Motivation

Einleitung: von Modellergebnissen zu Analyse

Es gibt viele Lösungen & Tools zur Szenario-Datenverarbeitung & Visualisierung, aber die meisten sind in ein Modell-Framework integriert oder komplett allgemein



Skripte zur Verarbeitung und Analyse von Modell-Ergebnissen

Im Gegensatz zu (open-source) Modell-Frameworks folgen interne Skripte zur Szenario-Analyse selten „best-practice of scientific software development“

- Das übliche Schema der Erstellung von Skripten zur Analyse von Modellergebnissen:
  ⇒ Ein paar Zeilen Code – dann noch ein paar Features – dann noch ein paar Features …
- Folgen der der inkrementellen Herangehensweise… (nicht immer, aber oft)
  ⇒ *copy-paste* von Code-Blöcken von Projekt zu Projekt
  ⇒ Kein Versions-Management der Analyse-Skripte
  ⇒ Keine (ausreichende) Dokumentation des Codes
  ⇒ Keine automatisierten Tests (im Sinne einer *continuous-integration*-Strategie)
- Warum ist das ein Problem für gute Wissenschaft?
  ⇒ Kaum Nachvollziehbarkeit, Reproduzierbarkeit und Transparenz der Ergebnisse
  ⇒ Risiko von Fehlern/Bugs in existierenden Features bei der Weiterentwicklung
  ⇒ Risiko von Fehlern/Bugs durch Updates von verwendeten Software-Paketen

Vision: eine Python-Toolbox für Energie- und Klima-Modellierung

Das pyam-Paket bietet eine Reihe an modell-unabhängigen Funktionen, um die Analyse & Visualisierung von Szenarien zu vereinfachen

- Design-Prinzip:
  - ⇒ Harmonisiertes Daten-Modell (=Struktur)
  - ⇒ Modell-unabhängige Standard-Funktionen zur Analyse & Visualisierung
  - ⇒ Modulare Paket-Architektur und einfache Integration von/in andere Python-Pakete

- Vorteile für Modelierer·innen:
  - ⇒ Standardisiertes Interface orientiert an *pandas* & *matplotlib* für effiziente Analyse
  - ⇒ Umfangreiche Dokumentation, Tutorials, Mail-Verteiler, Slack-Workspace, …
  - ⇒ Performante interne Implementierung als `pandas.Series` statt `pandas.DataFrame`
  - ⇒ Verbesserte Transparenz & Nachvollziehbarkeit durch kürzere Analyse-Skripte
  - ⇒ Erhöhte Zuverlässigkeit durch umfangreiche Tests & *continuous-integration*-Strategie

Teil 2

Das pyam-Paket und das zugrunde liegende Daten-Modell

Supported data models and file formats

The package supports various formats & types of timeseries data and is currently used by more than a dozen modelling teams

Supported timeseries data formats:

The *pyam* package was initially developed to work with the *IAMC template*, a tabular format for yearly timeseries data

	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	...

But the package also supports sub-annual time resolution

⇒ Continuous-time formats (e.g., hourly timeseries data)

⇒ Representative sub-annual timeslices (e.g., "winter-night")

Compatible i/o and file formats:

⇒ Full integration with the *pandas* data analysis package

⇒ Tabular data (`xlsx`, `csv`) & "frictionless" datapackage format

The *pyam* package for integrated assessment & macro-energy modelling

A community package for scenario processing, analysis & visualization following best practice of collaborative scientific software development

Use cases and features

- ⇒ Data processing — Data i/o & file format conversion, aggregation, downscaling, unit conversion, …
- ⇒ Validation — Checks for completeness of data, internal/external consistency, numerical plausibility …
- ⇒ Analysis & visualization — Categorization and statistics of scenario ensembles, plotting library, …

D. Huppmann, M. Gidden, et al. (2021). *Open Research Europe* **1**:74. doi: 10.12688/openreseurope.13633.1

Repository hosted on GitHub

Community supported by Groups.io slack

Documentation hosted by Read the Docs

#pyam_iamc

pyam-iamc.readthedocs.io

Teil 3

Rule number 1 of live demos – never do a live demo…

Thank you very much for your attention!

Read the docs on pyam-iamc.readthedocs.io

Join the mailing list on groups.io
or the Slack workspace

Create an issue or start a pull request
on github.com/IAMconsortium/pyam/

Read the Docs Groups.io slack GitHub

Dr. Daniel Huppmann
Research Scholar – Energy, Climate, and Environment Program (ECE)
International Institute for Applied Systems Analysis (IIASA)
Schlossplatz 1, A-2361 Laxenburg, Austria
huppmann@iiasa.ac.at
@daniel_huppmann
www.iiasa.ac.at/staff/huppmann

Backup Slides

Feature support & ongoing development

We are committed to support more use cases & applications

Time domain

- ⇒ Standard IAMC format: yearly data
- ⇒ Subannual timeslices
  e.g., representative days
- ⇒ Continuous-time format
  e.g., ISO 8601, Python `datetime`

Directional data

use “>” in the region column
(e.g., `Germany>France`)

File types

- ⇒ tabular data (`xlsx`, `csv`)
- ⇒ “frictionless” data format
- ⇒ database format (`netcdf`)

Additional “extra” columns

import file with non-standard index or columns

Metadata and license info

Provenance tracking

Fully tested & documented | Experimental support | Not (yet) implemented

The “variable” column

The IAMC data format uses the “variable” column to implement a semi-hierarchical structure

The “variable” column can be used to implement a hierarchical tree

- ⇒ Aggregate: `Primary Energy`
- ⇒ Subcategory: `Primary Energy|Coal`
- ⇒ Further detail: `Primary Energy|Coal|w/CCS`

⇒ The package offers many tools to work with such hierarchical trees

```
df.filter(variable='Primary Energy*', level=1)
df.aggregate(variable='Primary Energy')
```

Read the docs for more information:

⇒ https://pyam-iamc.readthedocs.io/en/stable/data.html

Good practice for scenarios ensemble analysis

As part of the effort supporting the IPCC SR15 assessment, we wrote a list of "do's and don'ts" for model/scenario comparison

A user's guide to the analysis and interpretation of (unstructured) scenario ensembles

Don't interpret the scenario ensemble as a statistical sample or as likelihood/agreement.

Don't focus only on the medians, but consider the full range over the scenario set.

Don't cherry-pick individual scenarios to make general conclusions.

Don't over-interpret scenario results & don't venture too far from the original question.

Don't conclude that the absence of a particular scenario (necessarily) means that this scenario is not feasible or possible.

Based on Box 1, Huppmann et al., Nature Climate Change 8:1027-1030 (2018).
doi: 10.1038/s41558-018-0317-4 | open-access version: https://rdcu.be/9i8a