The Python package **pyam** for analysis, validation & visualization of integrated-assessment and energy-systems scenarios

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 821471 (ENGAGE).
Part 1

Introduction & motivation
Introduction: From model results to scenario analysis

There are many solutions and tools for scenario analysis & data visualization, but most are tools either “hard-wired” to a modelling framework or general purpose.

Model
Integrated assessment models, energy systems, power sector, land use (change), other sectors

Data processing and analysis tools “hard-wired” to a specific framework
- e.g., TIMES-VEDA, OSeMOSYS, MESSAGEix, REMIND, GCAM, mimi.jl, TEMOA, pypsa, PLEXOS, ...

Processing of model outputs

Validation of scenario results

Evaluation, analysis and visualization

Results, insights and scientific publications

General-purpose packages for data visualization
- e.g., matplotlib, seaborn, ggplot & shiny

General-purpose packages for data manipulation
- e.g., numpy, pandas & tidyverse

Reference data
Better-practices for scripts for scenario analysis and data visualization

Many modelling frameworks adopt “best-practice of collaborative development”, but scripts for scenario analysis are often written in an ad-hoc fashion

• A common approach to scenario analysis & data visualization
  ⇒ Write a few lines of code for a simple feature – a few features – and a little bit more …

• Caveats of this incremental approach (not always, but way too often)
  ⇒ copy-paste of large snippets of code from one project to the next
  ⇒ No version management for the analysis scripts
  ⇒ Insufficient documentation of code
  ⇒ No testing, no continuous-integration-strategy

• Why is this a problem for open & reproducible science?
  ⇒ Limited reproducibility or transparency of the results
  ⇒ Risk of errors or bugs in existing features during further development
  ⇒ Risk of errors or bugs due to dependency updates
Our vision: a community Python toolbox for energy & climate research

The pyam package offers a suite of model-independent methods to streamline the processing, analysis & visualization of scenario results

• Design principles:
  ⇒ Harmonized data structure and formats
  ⇒ Model-independent standardized methods for scenario analysis & visualization
  ⇒ Modular package architecture and simple integration in other packages & workflows

• Advantages for modellers and analysts
  ⇒ Standardized interface following the pandas & matplotlib packages
  ⇒ Comprehensive documentation, tutorials, email list, Slack workspace, ...
  ⇒ High-performance implementation as pandas.Series statt pandas.DataFrame
  ⇒ Increased transparency & better intelligibility through shorter analysis scripts
  ⇒ Higher reliability thanks to a well-testing package with a continuous-integration-strategy
Part 2

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Model</td>
<td>Scenario</td>
<td>Region</td>
<td>Variable</td>
<td>Unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MESSAGE</td>
<td>CD-LINKS 400</td>
<td>World</td>
<td>Primary Energy</td>
<td>EJ/y</td>
<td>462.5</td>
<td>500.7</td>
</tr>
</tbody>
</table>

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 \textit{pandas} data analysis package
- Tabular data (\textit{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, ...


License | Apache 2.0  
--- | ---
Python | 3.7 | 3.8 | 3.9
Chat | Slack  
Mail | groups.io

Code style | black  
--- | ---
Pytest | passing  
Docs | passing  
Codecov | 95%

DOI | 10.5281/zenodo.1470400  
--- | ---
ORE | 10.12688/openreseurope.13633.2

Repository hosted on  
Community supported by  
Documentation hosted by

[GitHub](https://github.com) | [Groups.io](https://groups.io) | [Slack](https://slack.com) | [Read the Docs](https://readthedocs.org)

#pyam_iamc  
pyam-iamc.readthedocs.io
Developing a community for a community package

*We made an effort to make the pyam package usable for modellers & analysts with a wide range of experience levels and scientific backgrounds*

- Simple installation
  - Available via the common Python managers *pypi* and *conda*
- Open-access manuscript & comprehensive documentation
  - Several tutorials and full-fledged API documentation
- For novice users or moderate-interest users:
  - An email list for announcements of new releases and questions
- For users interested in frequent updates, tips-and-tricks and more interaction
  - A Slack workspace with a *#helpdesk* channel
- For expert users and anyone interested in contributing
  - The GitHub repo for collaborative scientific software development like issues and pull requests, continuous-integration workflows, release management, etc.
Part 3

A live demo & interactive discussion

See the open-source notebook at
https://github.com/danielhuppmann/ENGAGE-pyam-tutorial
Thank you very much for your attention!