



U.S. DEPARTMENT OF ENERGY

Virtual Tutorial Series

Open-Source Tools & Open-Access Solar Data

Webinar series part 2: Open-source analysis tools

Kevin Anderson, *Sandia National Laboratories*

Michael Deceglie, *National Renewable Energy Laboratory (NREL)*

Tassos Golnas, *Solar Energy Technologies Office (SETO), DOE*

Bennet Meyers, *SLAC National Accelerator Laboratory*

Kirsten Perry, *National Renewable Energy Laboratory (NREL)*



Data: a Means to an End

Better photovoltaic (PV) models and system performance through high-quality data.

PV models are important in:

- Project development and valuation
- Power plant operation and maintenance

Better system performance means lower cost of solar electricity

Prize goal:

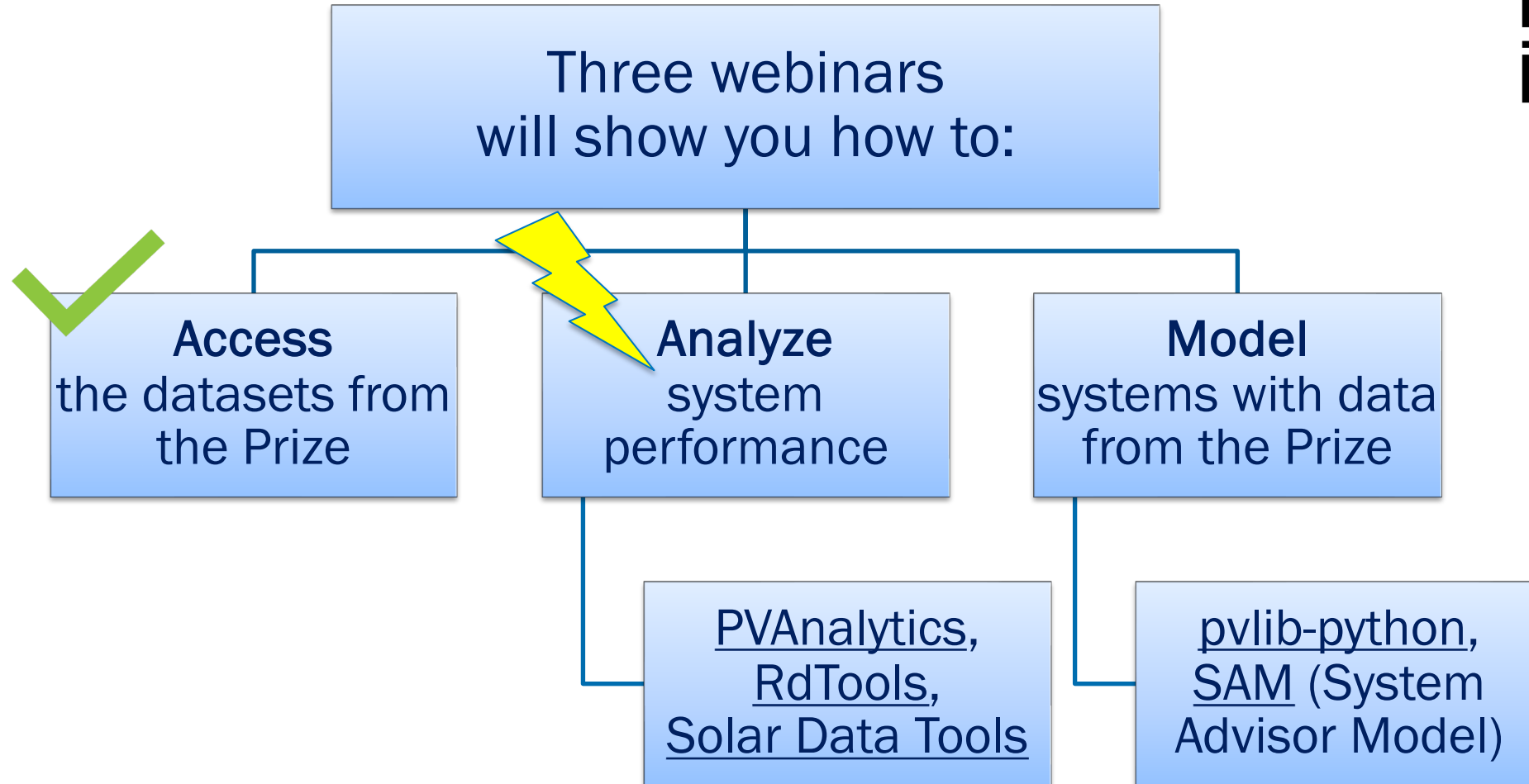
Support industry and academic research efforts to **develop, improve, evaluate, and validate** models of real-world PV system performance in diverse locations



Register!



Open-Access Data & Open-Source Tools



Analysis tutorial overview

We will look at three open-source tools for time series analytics



PVAnalytics

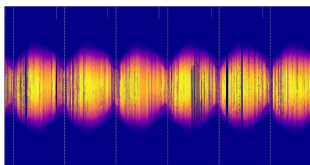
Library focused on PV time series preprocessing and QA



RdTools

Library built around loss factor analysis

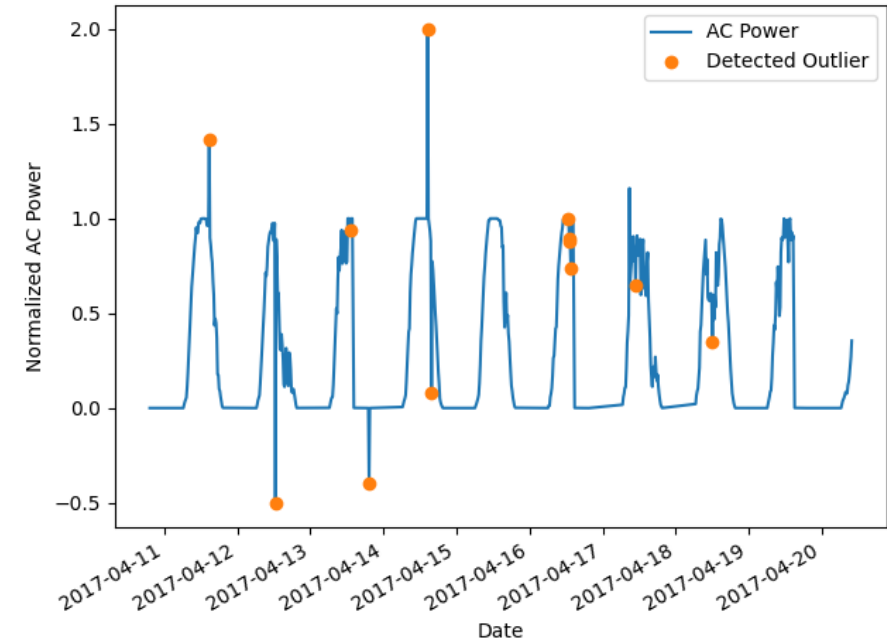
**Solar
Data
Tools**



Application-driven library for both data quality and loss factor analysis

PV Analytics

- A toolbox of functions for PV data labeling and quality assurance (QA)
- A foundational project for building out data processing pipelines, e.g. PV Fleets



- Install: <https://pypi.org/project/pvanalytics/>
 - `pip install pvanalytics`
- Documentation: <https://pvanalytics.readthedocs.io>
- Repository: <https://github.com/pvlib/pvanalytics/>

PV Analytics functionality

Data QA

Consistency
Time shifts
Capacity shifts
Outliers
Completeness

Data labeling

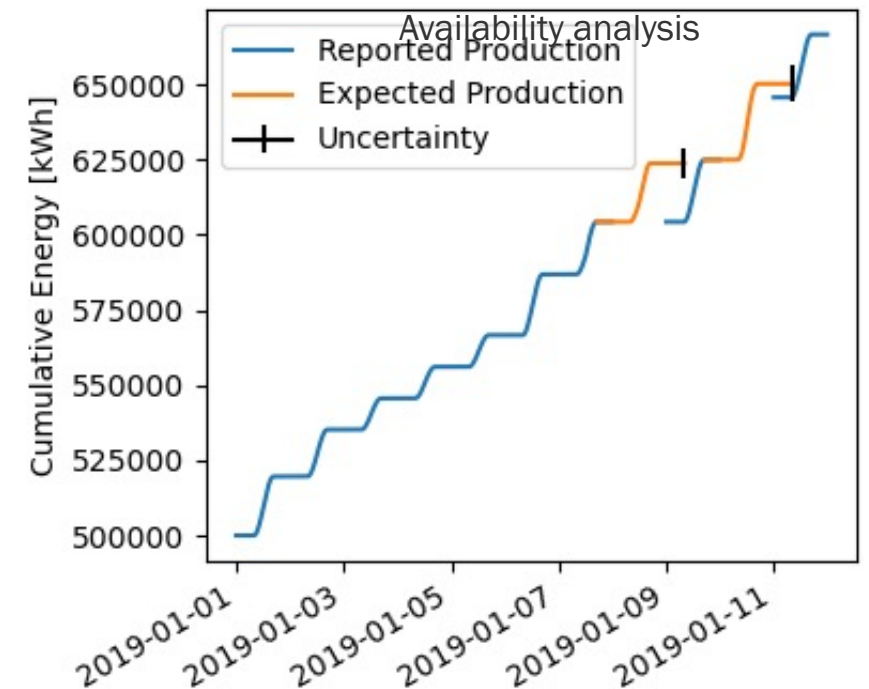
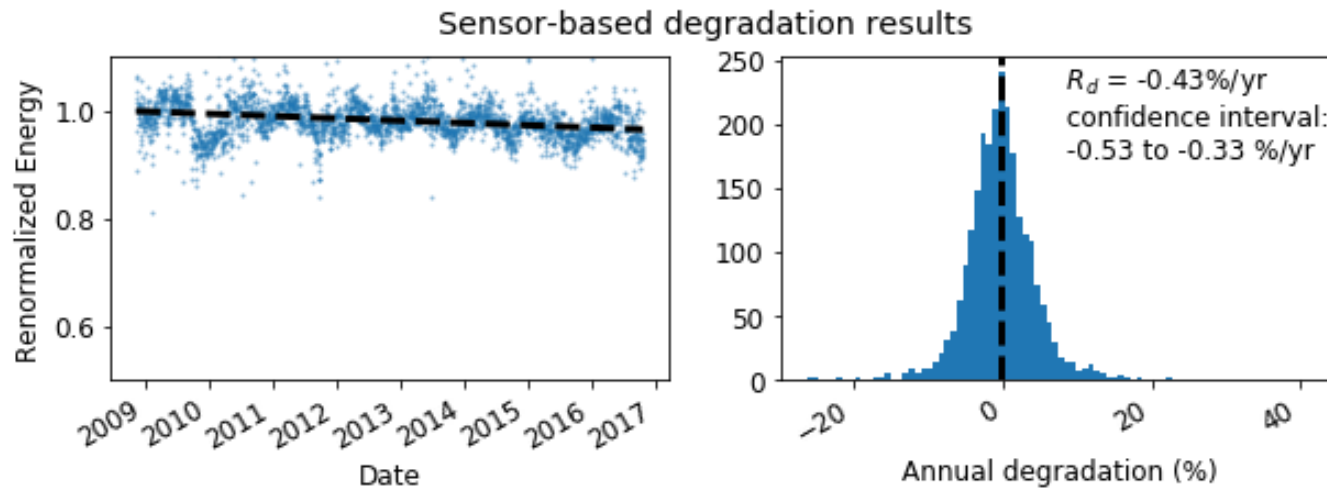
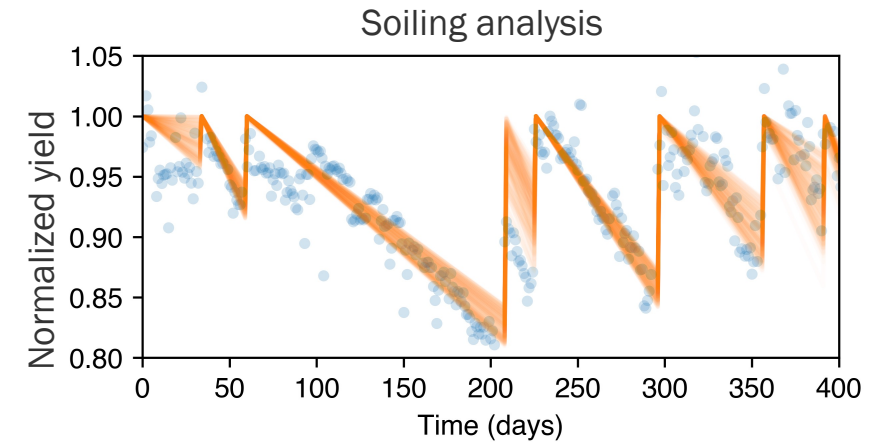
Day / night
Inverter clipping
Clear-sky

More!

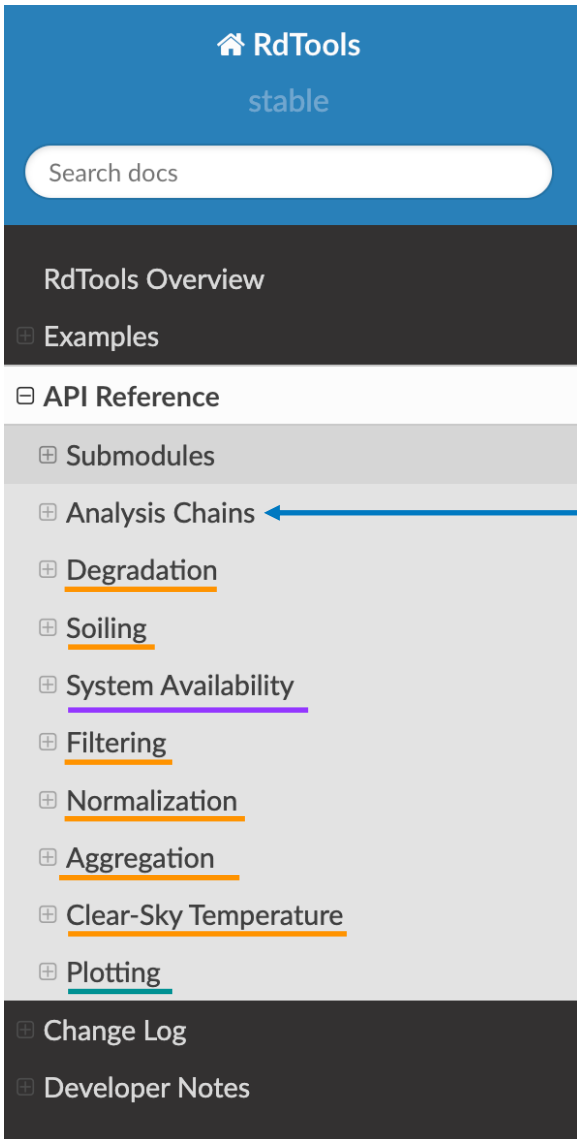
Variability index
PR
Orientation
Azimuth-tilt
etc...

RdTools

- Open-source python library for PV time series analytics
- Current workflows:
 - Degradation
 - Soiling
 - Availability
- We will focus on these workflows, but the building blocks are available in the library



RdTools structure and data requirements

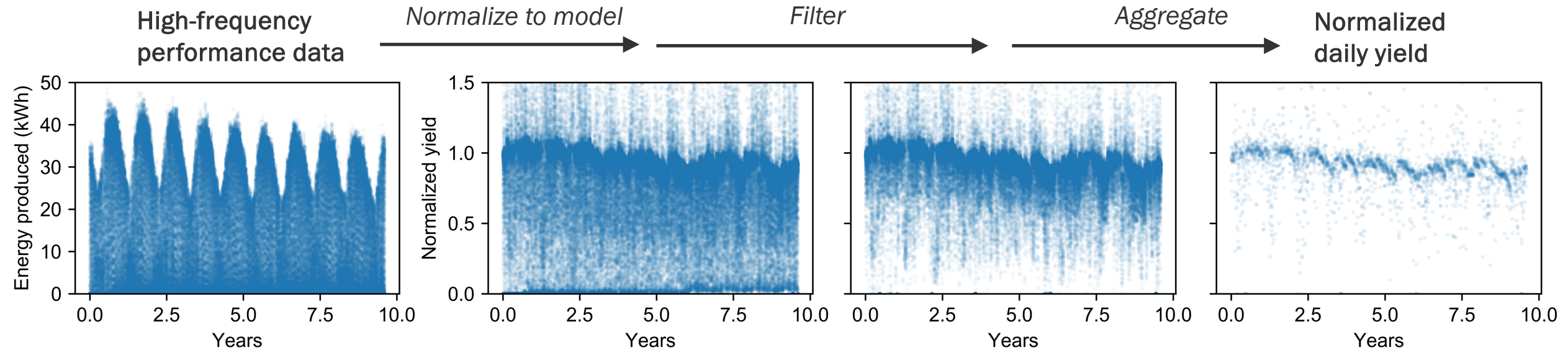


Canned analysis routines for **degradation** and **soiling**, object-oriented

Functional modules supporting **degradation** and **soiling**

- Input data
 - PV energy or power time series
 - Irradiance and temperature
 - Inverter and meter-level (both required for availability analysis)
 - Multiple years required for degradation
- Usually ~15-minute data, but other resolutions are technically supported
- Relies heavily on pandas

Degradation and soiling are based on daily yield

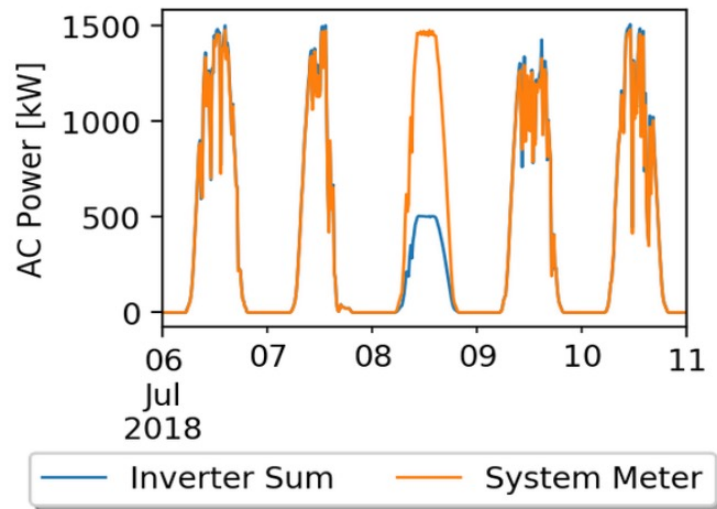


- The TrendAnalysis interface executes this workflow

Availability: really offline, or just missing data?

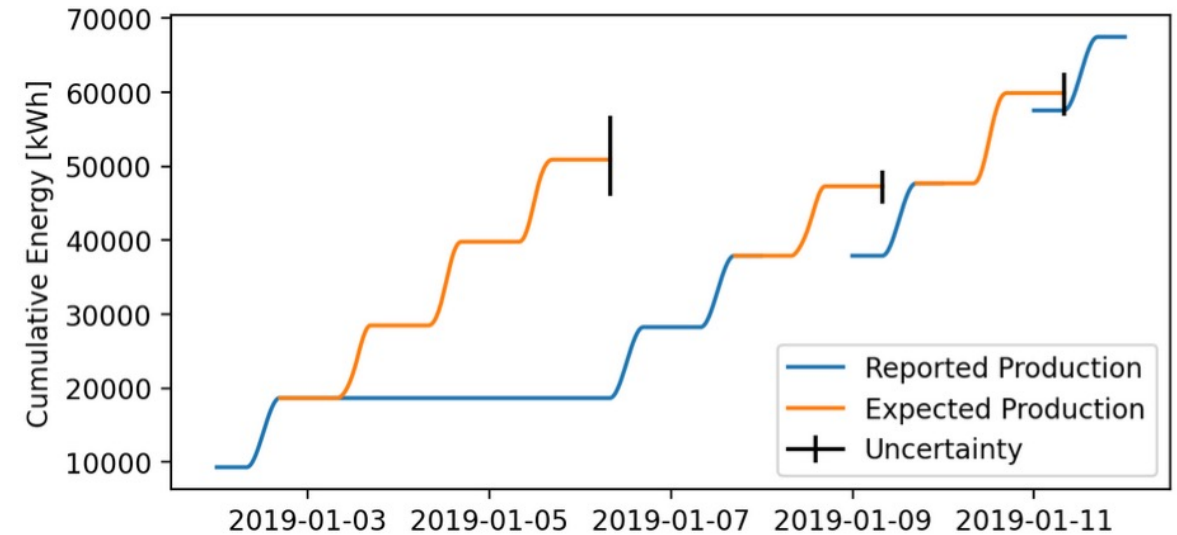
Partial outages

Peer-to-peer comparison using online inverters



Full outages

All inverters offline, use cumulative production data instead

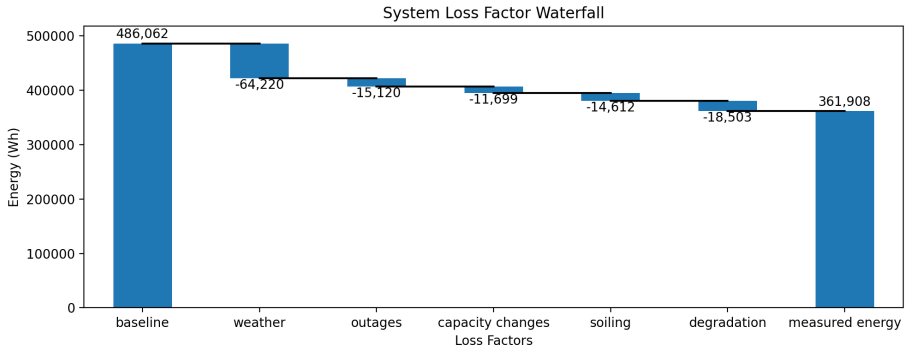
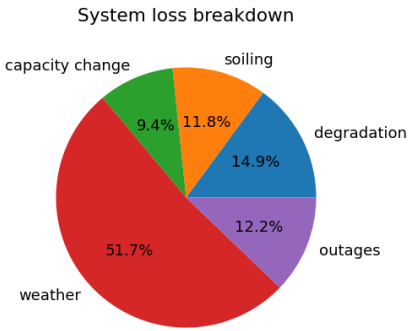
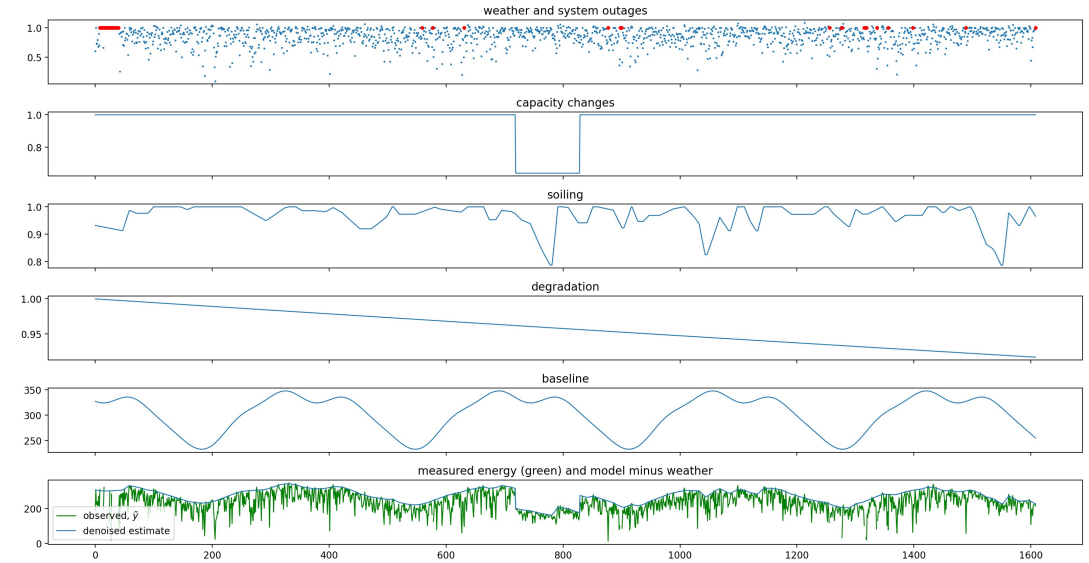
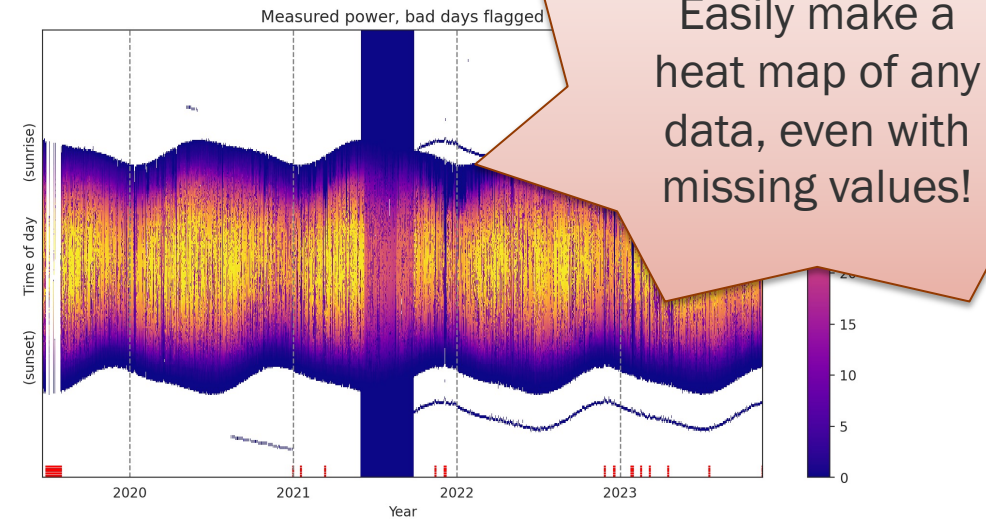


Where to find RdTools

- <https://pypi.org/project/rdtools/>
 - pip install rdtools
- <https://github.com/NREL/rdtools>
- <https://rdtools.readthedocs.io/>

Solar Data Tools (SDT)

- Open-source python library for analyzing PV power (and irradiance) data
- Methods for data cleaning, filtering, plotting, and analysis
- Goal to enable analysis of *unlabeled* PV data—no model, no meteorological data required
- Takes a statistical signal processing approach
- Data processing steps are largely **pre-defined** and **automatic** regardless of system type—from utility tracking system to multi-pitch rooftop systems



Where to find us

```
pip install solar-data-tools
```

<https://solar-data-tools.readthedocs.io>

<https://github.com/slacgismo/solar-data-tools>

The screenshot shows the documentation page for Solar Data Tools. At the top, there is a navigation bar with links for 'Solar Data Tools documentation', 'Getting Started', 'User Docs', 'Contribution Guidelines', and 'Project details'. A search bar and a 'Choose version' dropdown are also present. The main heading is 'Solar Data Tools', with sub-headers for 'pypi v1.0.5' and 'Anaconda.org 1.0.5'. A paragraph describes the tools' purpose: performing common tasks on solar PV data signals, such as finding clear days, data transforms, and fixing time stamp issues. Below this is an 'Important' notice stating that the documentation pages are under construction. Two main sections are highlighted: 'Getting Started' with a 'To the getting started guides' button, and 'User Guide' with a 'To the user guide' button.

The screenshot shows the GitHub repository page for solar-data-tools. The repository name is 'solar-data-tools' and the license is 'BSD-2-Clause license'. A table provides key statistics:

Latest Release	pypi v1.0.5 Anaconda.org 1.0.5 Last updated 13 Jan 2024
License	license BSD
Build Status	docs failing Tests no status build passing
Code Quality	lgtm grade no longer available lgtm alerts no longer available
Publications	DOI 10.5281/zenodo.10548791
PyPI Downloads	downloads 576/month
Conda Downloads	downloads 370 total
Test-Coverage	test-coverage 45%

Below the table, a paragraph describes the tools' purpose, similar to the documentation page. It also mentions close integration with the 'Statistical Clear Sky' repository and provides a link to the 'notebooks' folder for examples.

Currently working on a large pull request to significantly improve the documentation. Check back soon!

The Data Handler Object

- Most user interaction with SDT will be through the data handler

```
from solardatatools import DataHandler
df = ... # load data as pandas data frame
dh = DataHandler(df)
dh.run_pipeline()
```

- Main pipeline runs preprocessing, cleaning, and quality checks
 - Includes heatmap/matrix embedding, sunrise/sunset estimation, clear day labeling, and more
- After pipeline is run, user can invoke other analyses (loss factors, location estimation, *etc.*) or plotting functions

Demos



Library focused on PV time series preprocessing and QA

Structured similarly to pvlib

Largely model driven: some functions use met data

When to use

- Customizable and open-ended
- Looking for specific function
- Want to define own pipeline



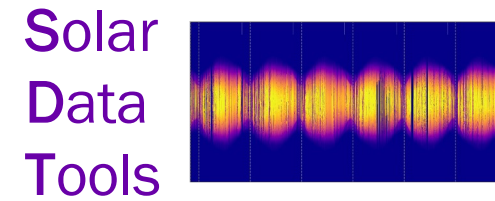
Library built around loss factor analysis

Provides canned analysis routines

Model driven: relies on met data

When to use

- Customize the analysis
- Desire a physical baseline approach
- Compare with Fleets Data Initiative
- Have met data available, quality irradiance



Application-driven library for both data quality and loss factor analysis

Runs automatically with little to no set up

PV-model free, does not require weather/irradiance data

When to use

- Want pre-defined pipeline
- Quick answers on minimal data
- Complex site/roofs
- Unavailable meteorological data, or another point of comparison

Thank you for joining!

All today's examples will be available at https://github.com/PV-Tutorials/2024_Analytics_Webinar