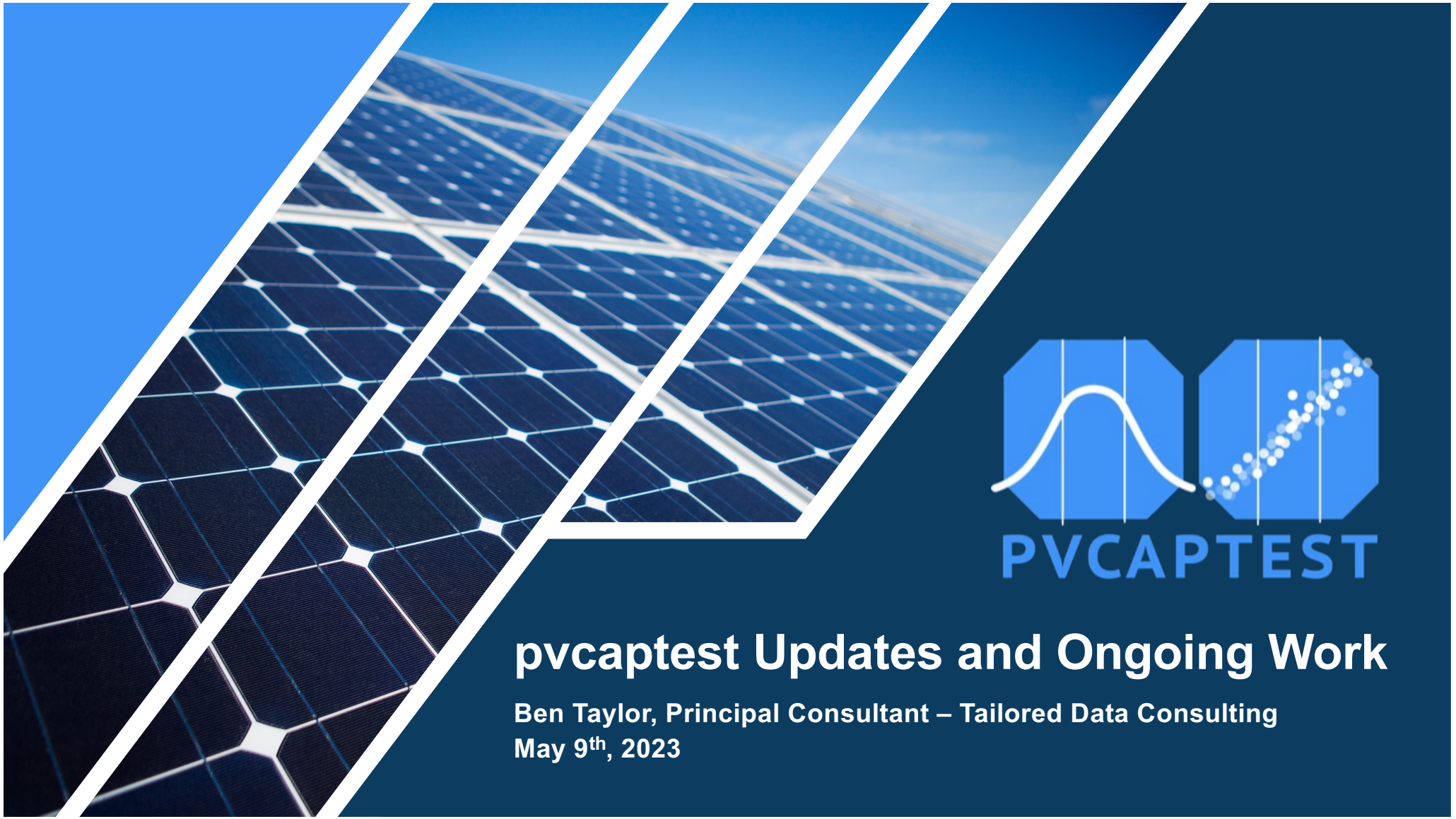


PVCAPTEST

pvcaptest Updates and Ongoing Work

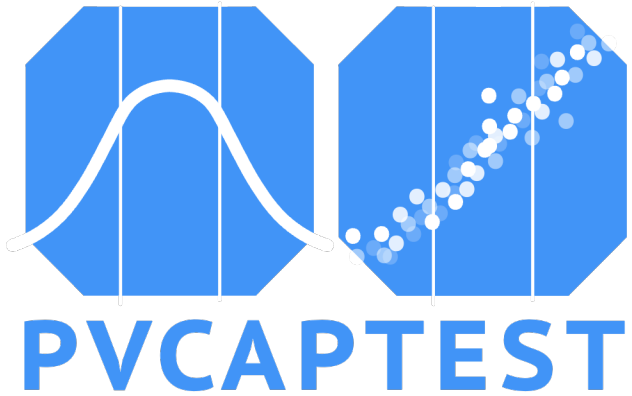
Ben Taylor, Principal Consultant – Tailored Data Consulting

May 9th, 2023

What is pvcaptest?

- Python package for running a capacity test following ASTM E2848
- MIT licensed
- Introduced at PVPMC in 2019:

  captest - Open Source package for Reproducible Performance Testing
- Documented on Read the Docs
- Available as a conda package (recommended) or from pypi
  - conda install –c conda-forge pvcaptest



Goal is to provide a full featured, flexible, open tool to allow all parties performing a test to replicate results with pvcaptest.

Background - What is an ASTM E2848 Capacity Test

1. Build and commission PV plant
2. Collect data through DAS / SCADA system
3. Quality check validate data

$$P = E_{POA}(a_1 + a_2 * E_{POA} + a_3 * T_a + a_4 * v)$$

4. Load data (csv, excel, parquet) into a data processing tool (Excel, R, Pandas, pvcaptest) — load_data or load_pvsyt
5. Organize and transform data – column labels, units
6. Visualize data
7. Filter data - low irradiance, clear / cloudy periods, outliers, clipping, shading
8. Calculate reporting conditions
9. Fit regression: AC power ~ POA irradiance, ambient temperature, wind speed
10. Predict AC power with regression coefficients, regression equation, and reporting conditions
11. Compare predicted power from measured data vs predicted power from modeled data
12. Calculate uncertainty of result ?
13. Present / document results

CapData methods

- Measured
- Modeled (PVsyst)

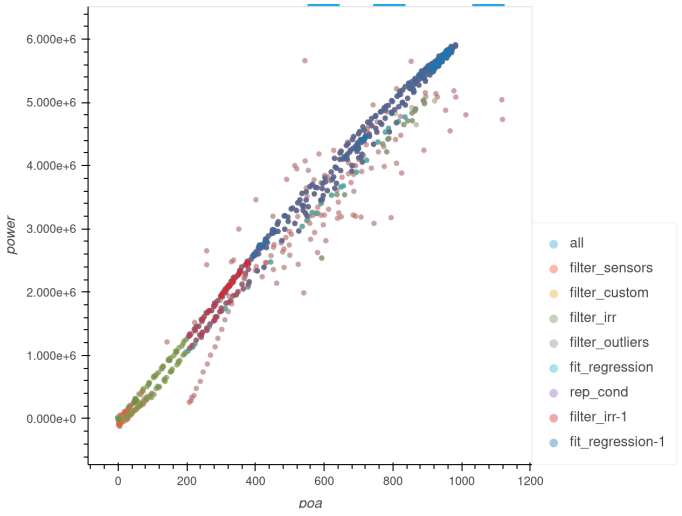
PVCAPTEST

Capacity Test with pvcaptest

das = load_data(path, group_columns, file_reader)

```
das.data.head(3)
```

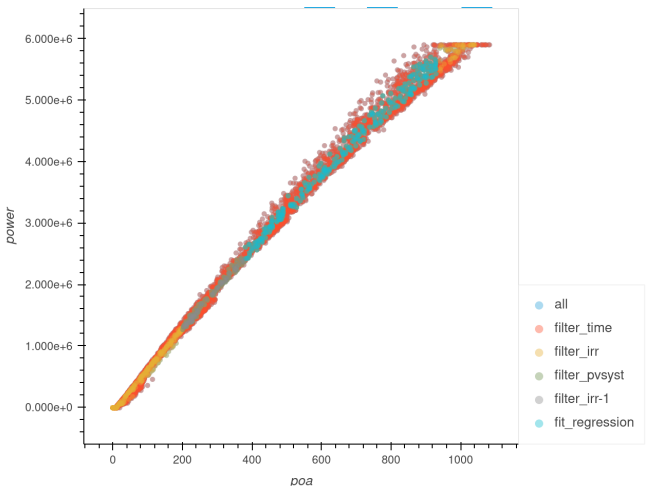
	met1_poa_refcell	met2_poa_refcell	met1_poa_pyranometer	met2_poa_pyranometer	met1_ghi_pyran
1990-10-09 00:00:00	0.0	0.0	0.0	0.0	
1990-10-09 00:05:00	0.0	0.0	0.0	0.0	
1990-10-09 00:10:00	0.0	0.0	0.0	0.0	



sim = load_pvsyt(path)

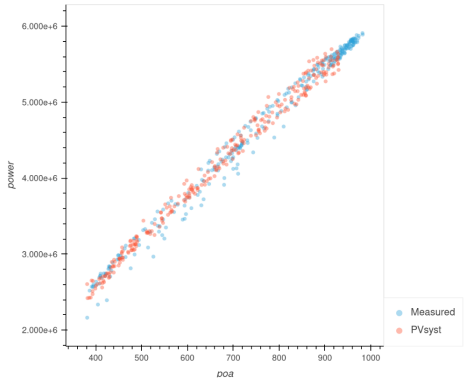
```
sim.data
```

Timestamp	GlobInc	GlobHor	T_Amb	TArray
1990-01-01 00:00:00	0.0	0.0	11.7	0.0
1990-01-01 01:00:00	0.0	0.0	12.2	0.0
1990-01-01 02:00:00	0.0	0.0	12.2	0.0
1990-01-01 03:00:00	0.0	0.0	11.1	0.0
1990-01-01 04:00:00	0.0	0.0	10.0	0.0




```
Using reporting conditions from das.

Capacity Test Result:    FAIL
Modeled test output:          4213300.189
Actual test output:           4915552.297
Tested output ratio:          1.167
Tested Capacity:              7000.050
Bounds:                       5580.0, 6420.0


Using reporting conditions from das.

Capacity Test Result:    FAIL
Modeled test output:          4215133.791
Actual test output:           4915552.297
Tested output ratio:          1.166
Tested Capacity:              6997.005
Bounds:                       5580.0, 6420.0


116.670% - Cap Ratio
116.620% - Cap Ratio after pval check
```

	das_pvals	sim_pvals	das_params	sim_params
poa	0.00000	0.00000	7,691.89402	7,662.79447
I(poa * poa)	0.00000	0.00000	1.59764	-0.83351
I(poa * t_amb)	0.00000	0.00000	-51.99357	-31.28454
I(poa * w_vel)	0.00163	0.28878	14.21468	-1.20866



PVCAPTEST

What's New

Overview / Agenda

v0.5.3 – May 12, 2019

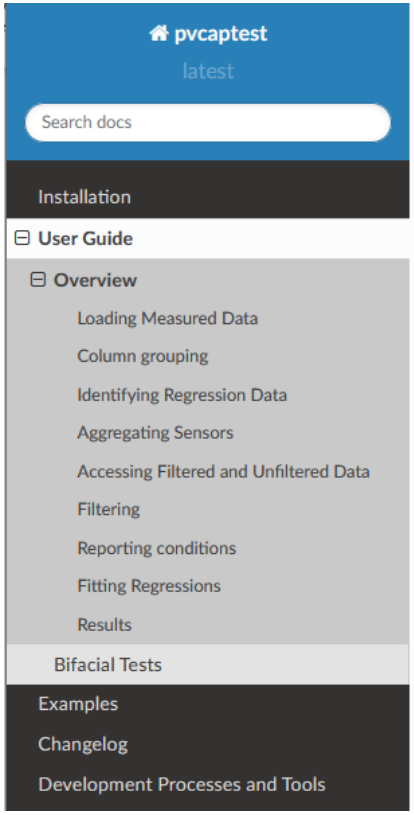
- Project Name - captest
- captest package
  - capdata
- Loading Data – CapData.load_data()
- Grouping Columns – CapData._CapData__set_trans
- Filtering methods – 8 methods

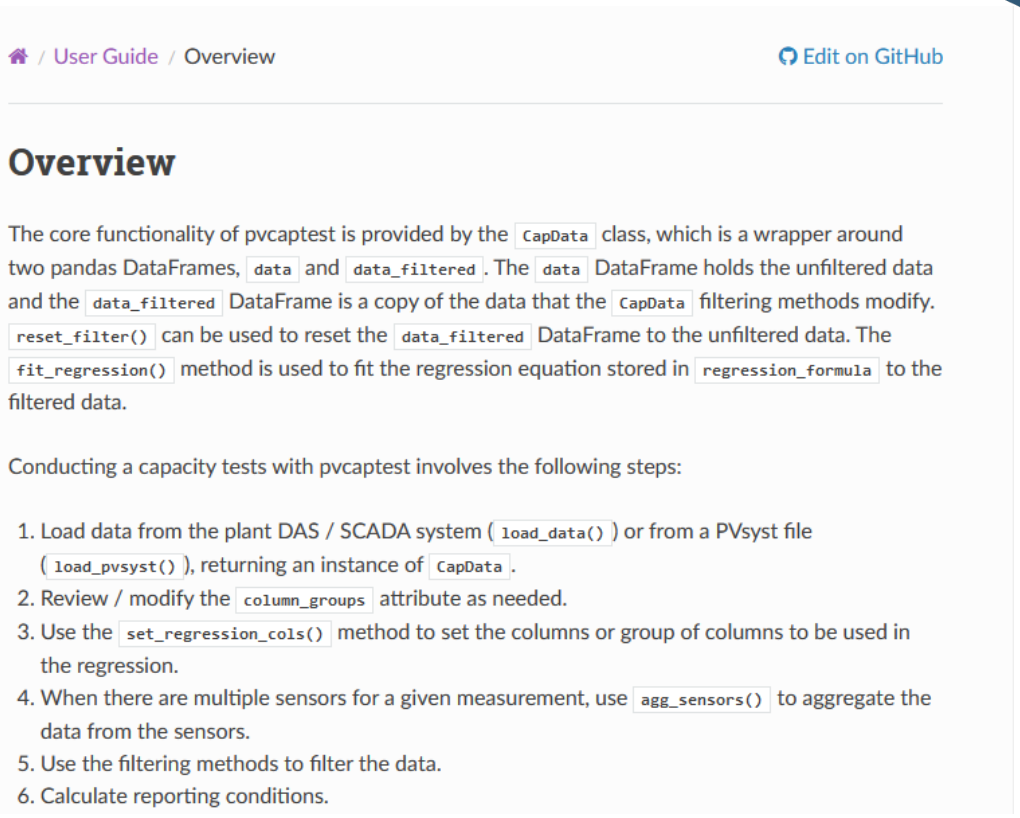
v0.11.2 – May 2023

- Project Name – pvcaptest
- Added User Guide to Documentation
- captest package
  - capdata, io, columngroups, prtest, util
- Loading Data
  - load_data and load_pvsyst in io module
  - load_data creates an instance of io.DataLoader
- Grouping Columns
  - moved to columngroups module
  - easy to specify explicitly with excel, json, or yaml
- Filtering Methods – 12 methods
  - filter_shade, filter_days, filter_power, filter_missing
- Re-factored reporting irradiance method
- Filtering output file (csv)
- Scatter Overlay of filtering steps and Timeseries output of filtering steps
- Work in progress

What's New – Documentation User Guide

- Added User Guide
  - Overview
  - Bifacial
- Updated
  - Installation
  - Examples





What's new – Loading Data

Prior to v0.11.*

- Create an instance of CapData and use load_data method

CapData
- Dataframe unfiltered data
- DataFrame for filtered data

New Functionality:

- Top level captest.load_data function, which returns an instance of CapData, like pd.read_csv returns a DataFrame
- io.DataLoader class
  - Provides a framework to specify location of file, load them, reindex, join
  - Used by the top level captest.load_data function
- Improved flexibility
  - Load individual file, the files in a directory, or a list of files
  - If the default function to read an individual file does not work, you can use a custom function
  - Load different type of files (xlsx, parquet), have to provide a function to read the file
  - Load files that have different column headings
- Defaults – sort data by time index, drop duplicate rows, and reindex so there are now missing timesteps

What's new – Grouping Columns

Prior to v0.11.*

- CapData._CapData__set_trans private method called by CapData.load_data

New Functionality:

- Column grouping algorithm moved to the columngroups module, but did not change
- Can specify column groupings in an excel, json, or yaml file explicitly
- Produce a template of excel column grouping file:

1. Load_data, set column_groups_template=True
2. Fill in Column A with column group names
3. Load_data, set column_groups_template=False and pass path of excel file to group_columns

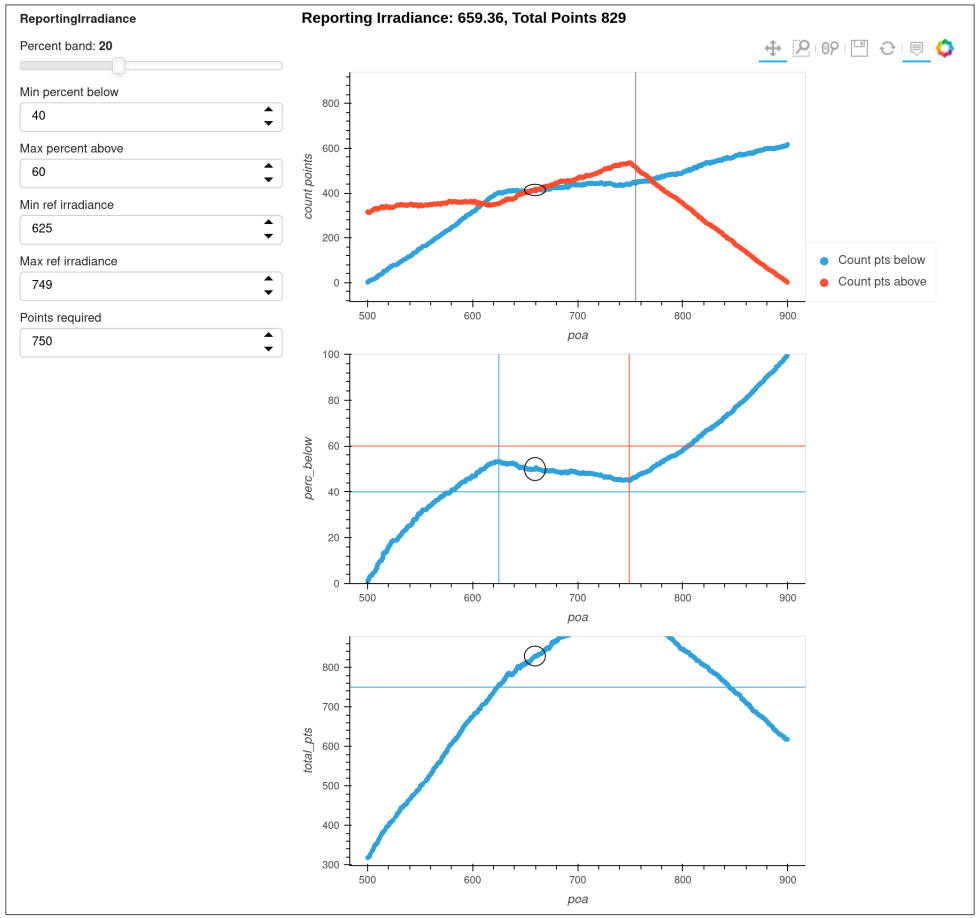
	A	B
1	irr-ghi-pyran	met1_ghi_pyranometer
2		met2_ghi_pyranometer
3	irr-poa-ref_cell	met1_poa_refcell
4		met2_poa_refcell
5	-mtr-	meter_power
6	wind--	met1_windspeed
7		met2_windspeed
8	temp-mod-	met1_mod_temp1
9		met1_mod_temp2
10		met2_mod_temp1
11		met2_mod_temp2
12	temp-amb-	met1_amb_temp
13		met2_amb_temp
14	-inv-	inv1_power
15		inv3_power

Use underscores in group names

What's new – Filtering Methods

- filter_shade – separated this functionality from the filter_pvsyst method
- filter_days – remove or keep all data from each day in a list of days from the data ['5/9/23', '5/10/23']
- filter_power – remove data above threshold, which can be an explicit value or a percent of a value
- filter_missing – Remove any time intervals where there is missing data within a given set of columns

What's new – Reporting Irradiance



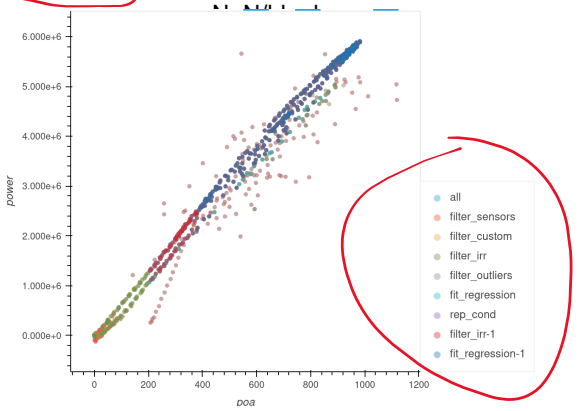
- irr_rc_balance function replaced by the ReportingIrradiance class
- Calculates reporting irradiance by selecting irradiance value from the filtered data where the quantity of measured irradiances above the reporting irradiance is equal to the quantity below
  - Save csv table of possible reporting irradiances
  - Save the plot as an html file
  - Interactive dashboard

What's new – Filtering Table to csv

CapData.get_filtering_table().to_csv('./path_to_save/file.csv')

	A	B	C	D	E	F	G	H	I	J
1	Timestamp	filter_sensors	filter_custom	filter_irr	filter_outliers	fit_regression	rep_cond	filter_irr-1	fit_regression-1	all_filters
74	10/9/1990 6:00	1								FALSE
77	10/9/1990 6:15	1								FALSE
84	10/9/1990 6:50	0	0	0	0	0	0	1		FALSE
85	10/9/1990 6:55	0	0	0	0	0	0	1		FALSE
86	10/9/1990 7:00	0	0	0	0	0	0	1		FALSE
87	10/9/1990 7:05	0	0	0	0	0	0	1		FALSE
88	10/9/1990 7:10	0	0	0	0	0	0	1		FALSE
89	10/9/1990 7:15	0	0	0	0	0	0	1		FALSE
90	10/9/1990 7:20	0	0	0	0	0	0	1		FALSE
91	10/9/1990 7:25	0	0	0	0	0	0	1		FALSE
92	10/9/1990 7:30	0	0	0	0	0	0	1		FALSE
93	10/9/1990 7:35	0	0	0	0	0	0	1		FALSE
94	10/9/1990 7:40	0	0	0	0	0	0	0	0	TRUE
95	10/9/1990 7:45	0	0	0	0	0	0	0	0	TRUE
96	10/9/1990 7:50	0	0	0	0	0	0	0	0	TRUE
97	10/9/1990 7:55	0	0	0	0	0	0	0	0	TRUE
98	10/9/1990 8:00	0	0	0	0	0	0	0	0	TRUE
99	10/9/1990 8:05	0	0	0	0	0	0	0	0	TRUE
100	10/9/1990 8:10	0	0	0	0	0	0	0	0	TRUE
101	10/9/1990 8:15	0	0	0	0	0	0	0	0	TRUE
102	10/9/1990 8:20	0	0	0	0	0	0	0	0	TRUE
103	10/9/1990 8:25	0	0	0	0	0	0	0	0	TRUE
104	10/9/1990 8:30	1								FALSE
105	10/9/1990 8:35	1								FALSE
106	10/9/1990 8:40	1								FALSE

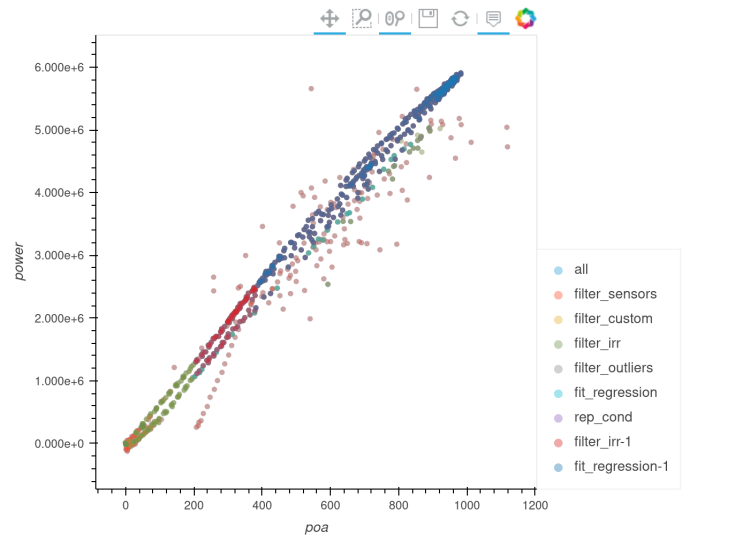
meas		pts_after_filter	pts_removed	filter_arguments
	filter_sensors	1245	195	Default arguments
	filter_custom	1240	5	DataFrame.dropna, ,
	filter_irr	424	816	200, 2000,
	filter_outliers	407	17	Default arguments
	fit_regression	385	22	filter: True, summary: False
	rep_cond	385	0	Default arguments
	filter_irr-1	293	92	0.5, 1.5, ref_val: self_val
	fit_regression-1	293	0	Default arguments



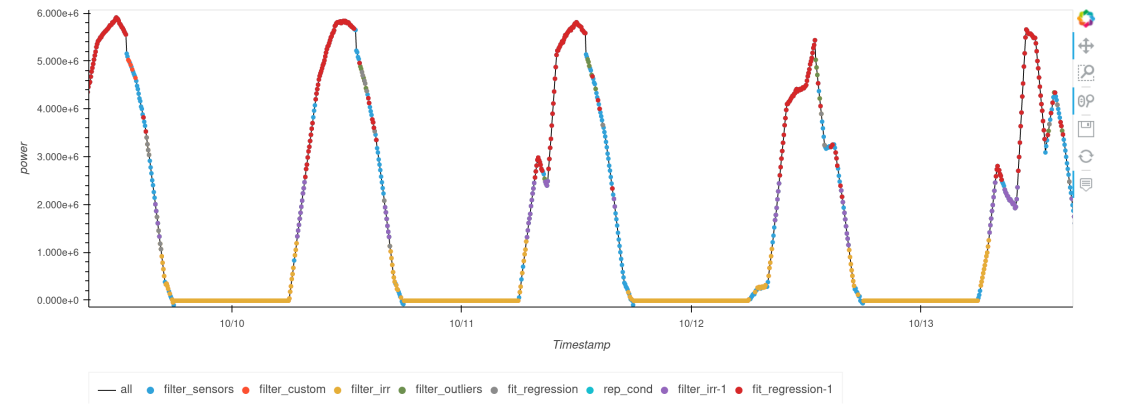
What's New – Visualizations

capdata.run_test – pass a CapData object and a list of filtering methods

```
das_scatter = das.scatter_filters()
das_scatter
```




```
das.timeseries_filters().opts(width=1200)
```



Used in the "Concise Capacity Test" example in the documentation.

Ongoing Work

Adding more filters- clear sky

Fix mybinder.org live examples

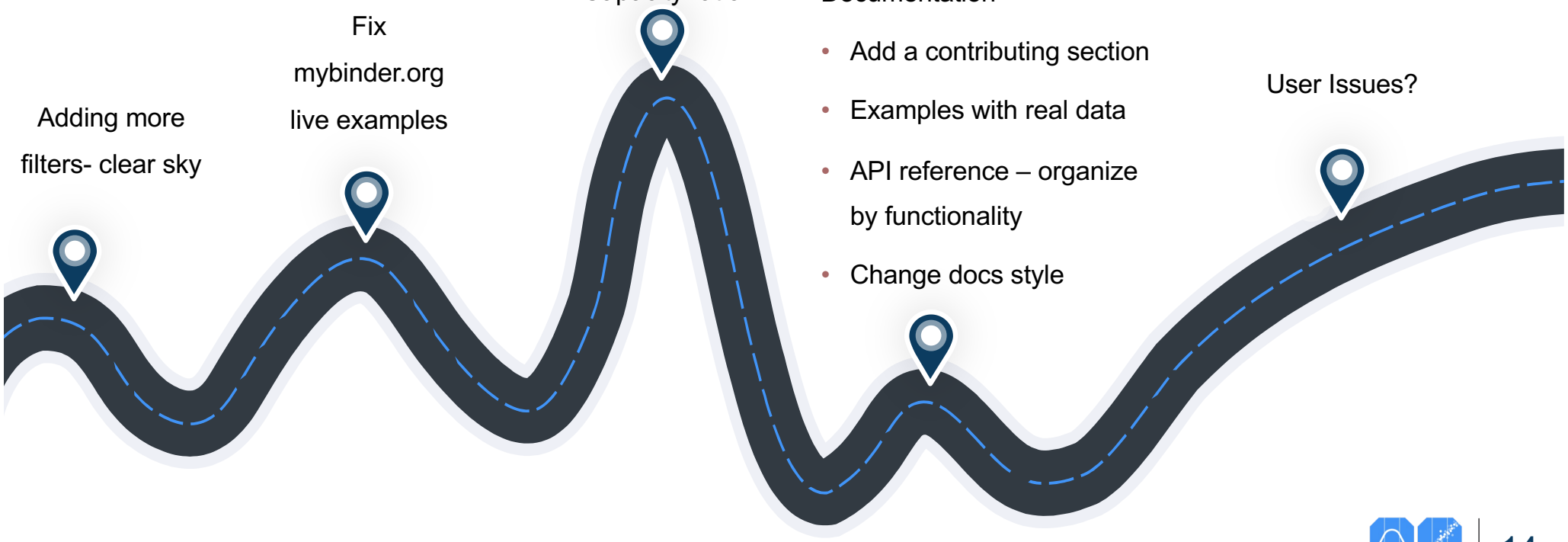
Uncertainty

- Predicted power
- Capacity ratio

Documentation

- Add a contributing section
- Examples with real data
- API reference – organize by functionality
- Change docs style

User Issues?



Resources / Questions

Contact me – ben@td.consulting
www.td.consulting

Open an Issue on Github

Source Code on Github

Documentation on Read the Docs