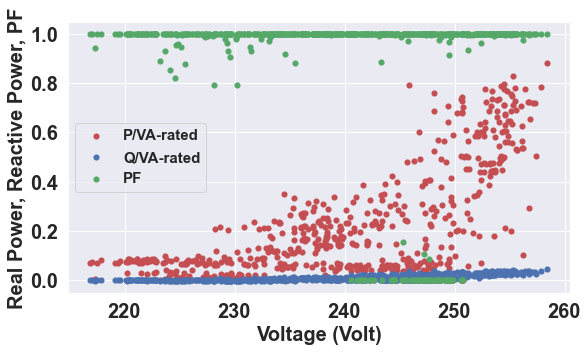
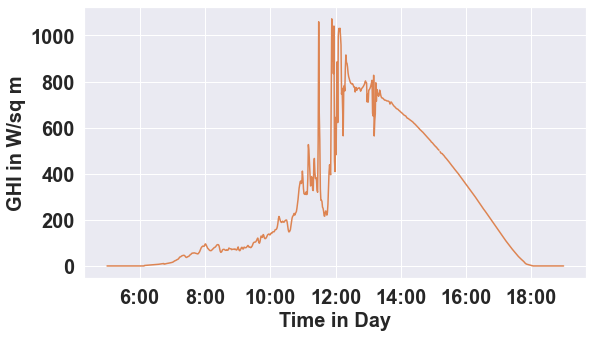
Solar Curtailment Project Progress Journal

# 13/09/2022

# Result on Tripping – non clear sky day (sample 1)

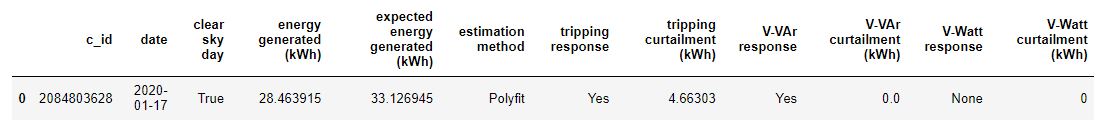
Graphical user interface, application, website

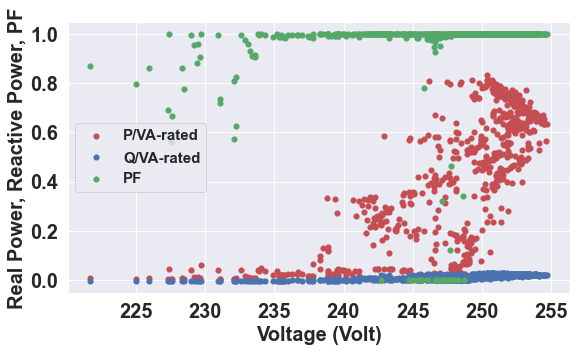
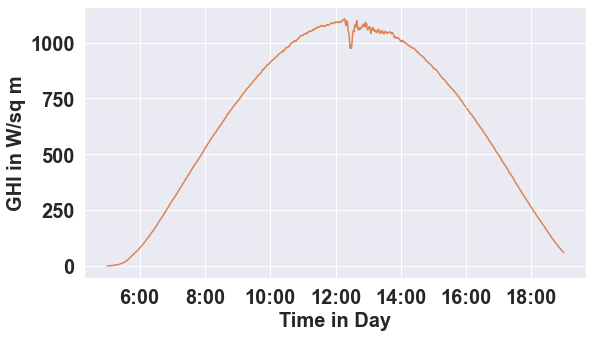
Description automatically generated

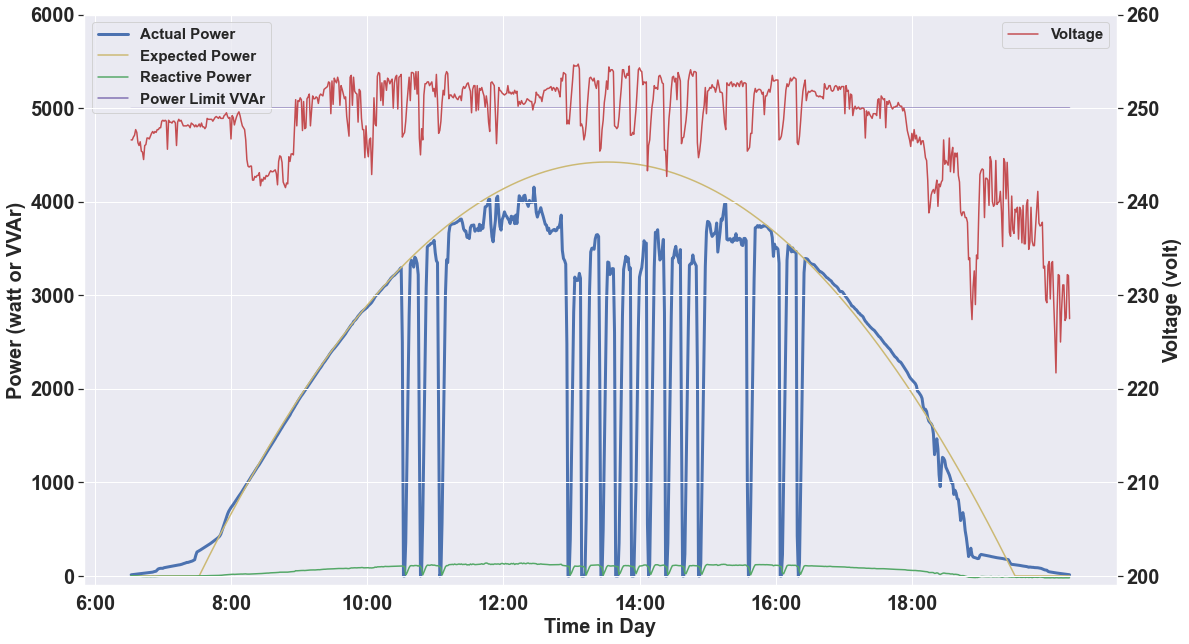




# Result on Tripping – clear sky day (sample 11)



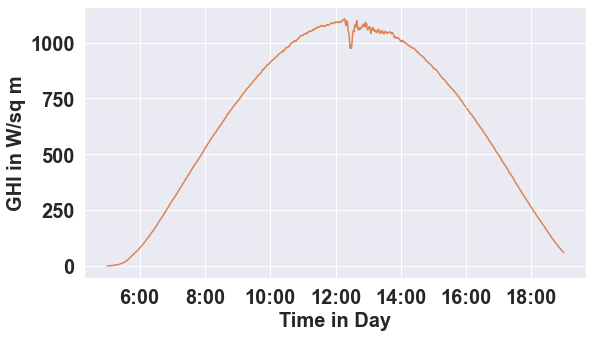
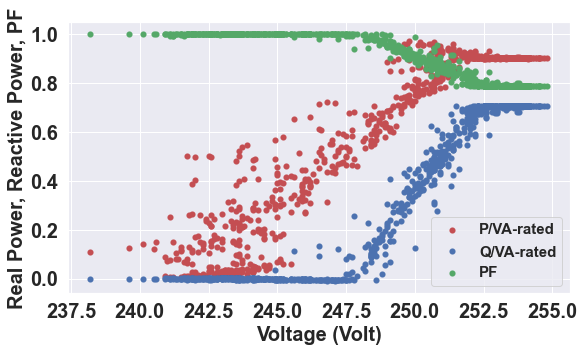


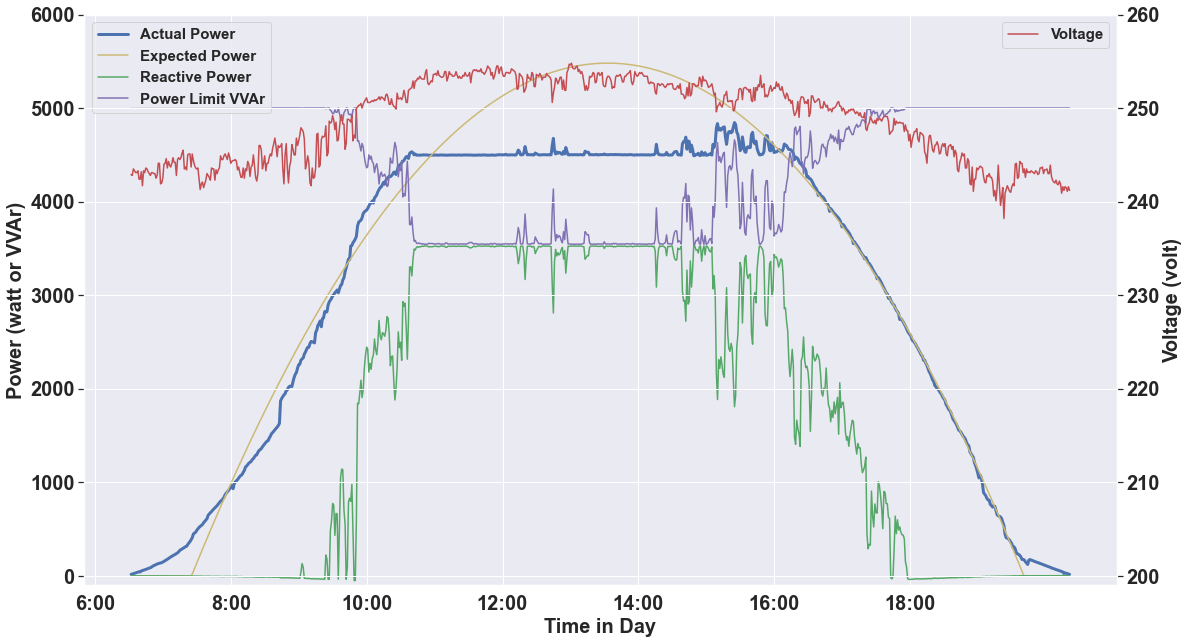


# Result on VVAr (sample 14)

Graphical user interface, application, Word

Description automatically generated

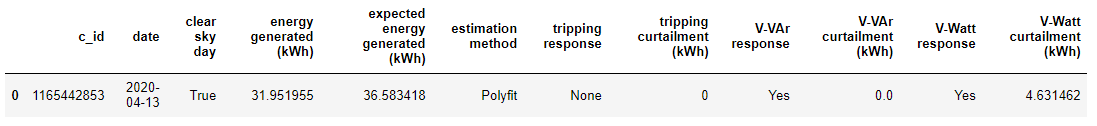
­­­­ ­­­­

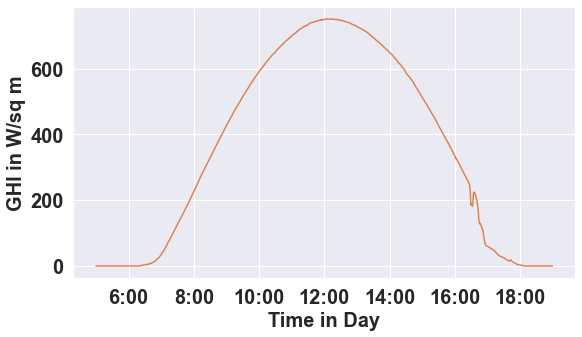
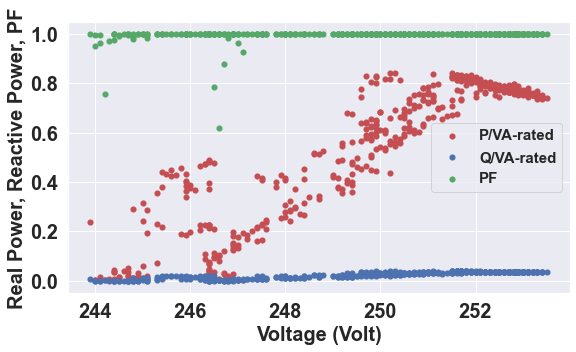


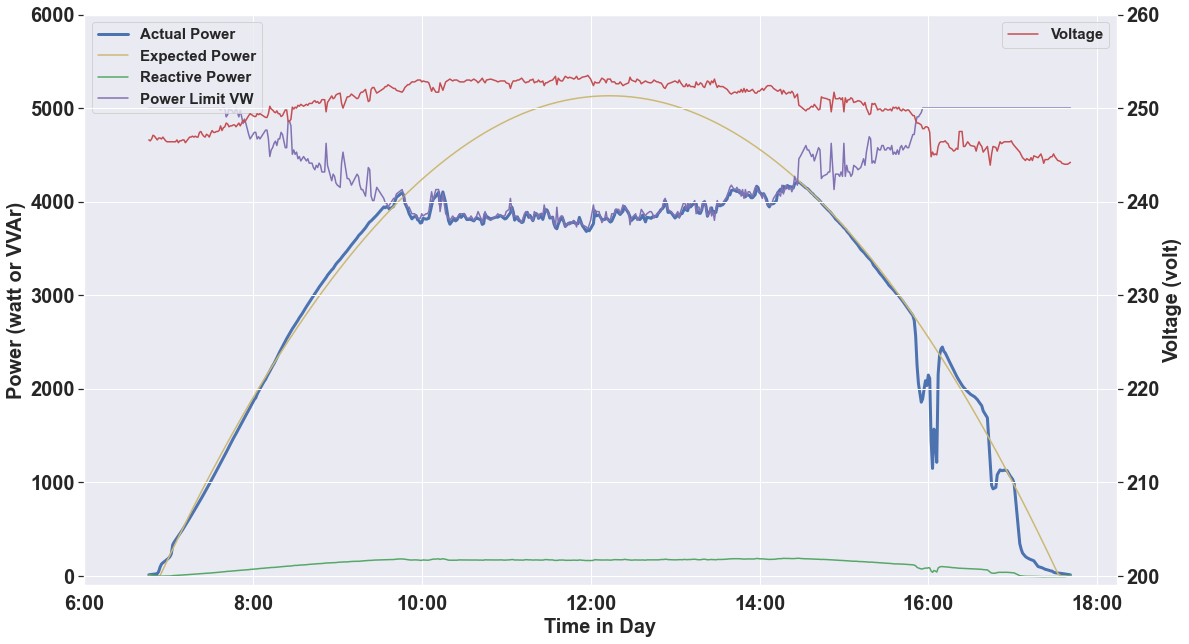
## There are two issues:

1. In the reactive power scatter plot, the Q/VA should have been negative. Problem with polarity correction?
2. In the power and voltage plot, the power limit vvar is below the actual power value. Probably the actual VA limit of the inverter is higher than the ac capacity of the inverter?

# Result on VWatt (sample 4)



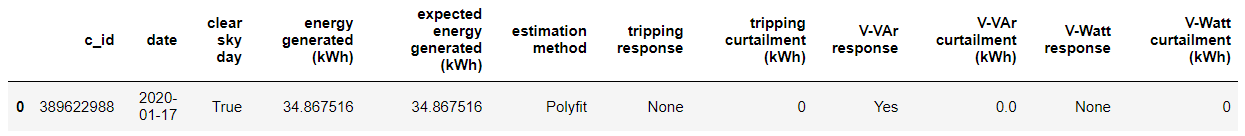


# Result on Incomplete Dataset (sample 5)



The tool will judge the dataset is incomplete only if there are less than 1000 rows in the data. The data should be more than that because the data resolution is either 60 or 5 seconds in SolA dataset.

# Result on No Curtailment Site in Clear Sky Day



Chart, line chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, line chart

Description automatically generated

# Polyfit with Constrain Idea Testing

When we implement the polyfit with constrain into a clear sky day without curtailment, it seems we get overestimating:

With constrain:

Chart, line chart

Description automatically generated

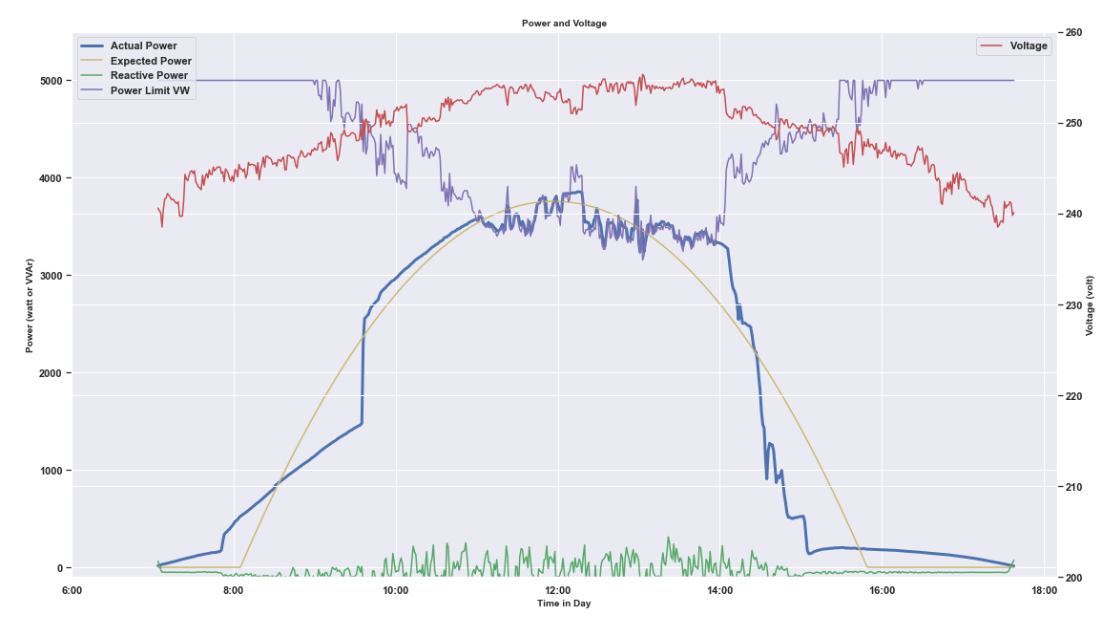
Without:

Chart, line chart

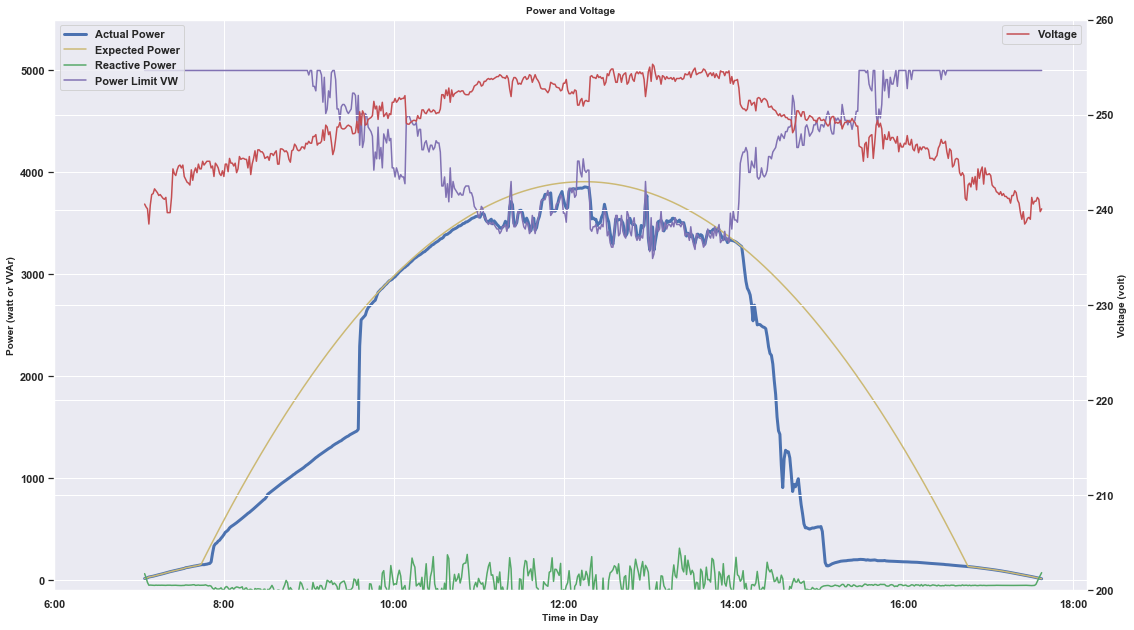
Description automatically generated

Should we give up on this? Or should we still think to optimize it since VWatt calculation is inaccurate and sometime it is underestimating, eg in sample 3:

Without constrain:



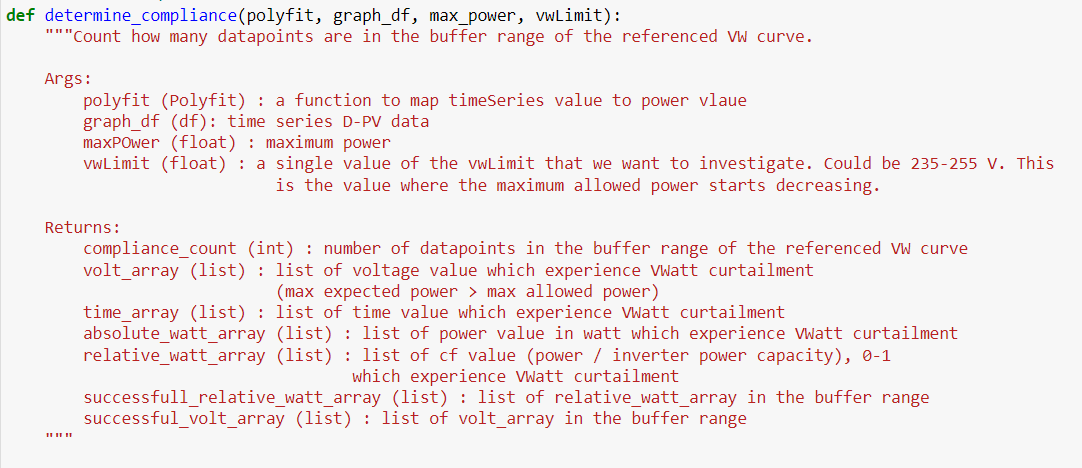
With constrain:



23/9 2022

Documentation

* Function and class docstring are done



* Naming convention edit is done:
  + Function, variable: lower\_case
  + Constant: UPPER\_CASE
  + Class: CamelCase
* Readme is done : https://github.com/mssamhan31/Solar-Curtailment
  + About
  + Getting started
  + Tool use demonstration: Screen capture of input, & output
  + High Level Explanation of How The Algorithm Works
  + Tool Limination & notes
  + Some Related Articles and Papers
  + Contributing
  + Project Partners
  + Authors
  + License
  + Contact
* Dataset information is done : <https://github.com/mssamhan31/Solar-Curtailment/blob/main/documentations/solar%20curtailment%20dataset%20information.docx>

# AC Capacity Curtailment

Chart

Description automatically generated with low confidenceA picture containing chart

Description automatically generated

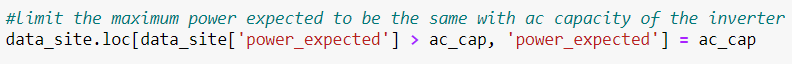
Chart

Description automatically generatedA picture containing chart

Description automatically generated

For c\_id 198317149, it is clear that the power is curtailed not because of tripping, VVAr, or VWatt response. Rather, it is because the poor sizing of the inverter. The ac capacity of the inverter is too low to accommodate the DC power generated by the PV array. Hence, the power is curtailed by the ac capacity of the inverter. In this case, the ac capacity is 5000 watt.

To accommodate this, we add one condition for the polyfit algorithm: make sure that the expected power generated is never more than the ac capacity of the inverter. Screenshot below:



Result for sample 4:

Chart, line chart

Description automatically generated

# Publish the script as a package

The main reference we use is <https://realpython.com/pypi-publish-python-package/> .

## Package Name

As discussed with Baran, we should try to name it with something general without CANVAS word. So, we name it ‘solar-curtailment’, which is still available in PyPI (python package index).

So later, we can install the package using this command.

pip install solar-curtailment

We will, however, try to publish it using the name trialsamhan2 for trial.

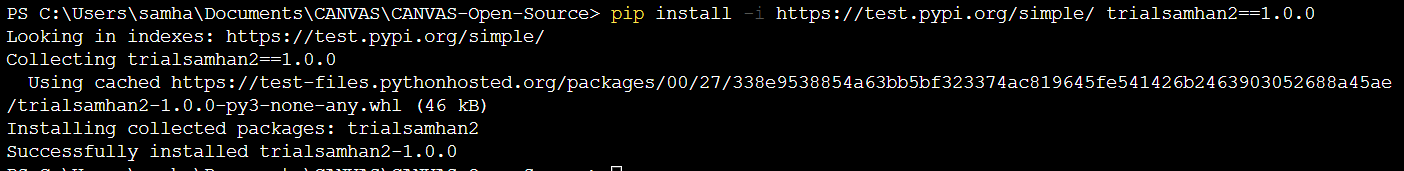
# Implementation to a Package

## Transformed the module into package

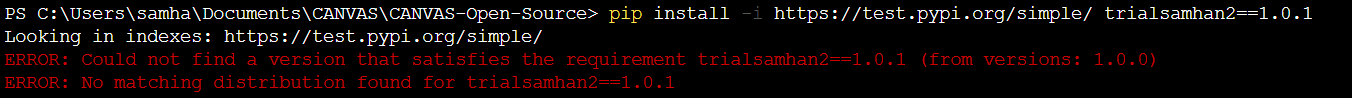
Graphical user interface, text

Description automatically generated with medium confidence

## Trial in TestPyPI



## Issue when changing the version in TestPyPI



Solved. This issue does not happen in pypi. Seems like only happen in TestPyPI.

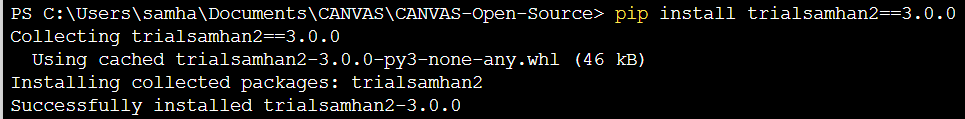
# Trial in PyPI

Install version 1.0.1

A screenshot of a computer

Description automatically generated

Install version 3.0.0



## Implementation

Text

Description automatically generated

# 6 Oct 2022

# Readme Baran Review

Followed up

# VVAr according to AS NZS 4777.2 2015 and 2020

Chart, line chart

Description automatically generated

## Allowed Range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | V1 (volt) | V2 (volt) | V3 (volt) | V4 (volt) |
| 2015 | 207 | 216-230 | 235-255 | 244-265 |
| 2020 | 180-230 | 180-230 | 230-265 | 230-265 |

However, according to Baran, this is not applicable because the site is indeed Australia B, as informed by the industry partner. Australia C site, for instance, is a remote area.

# VVAr detection algorithm

We say that it is VVAr if V3 in its range, V4 in its range, and the percentage of power scatter between V3 and V4 in the buffer range is higher than certain threshold, which is 80%.

Chart, scatter chart

Description automatically generated

# Script file separation

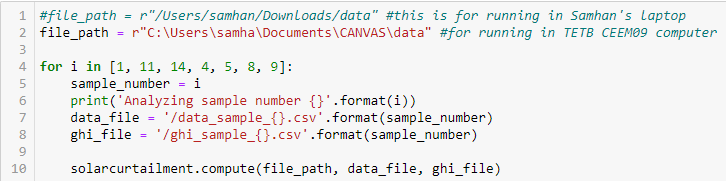
Graphical user interface, application

Description automatically generated

Merge to main?

# Current Implementation





# VVAr according to four different Standards:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| V-Var standards | VAr\_injection/  VA\_rated (%) | V1 (volt) | V2 (volt) | V3 (volt) | V4 (volt) | Var\_absorbtion/  VA\_rated(%) |
| SAPN TS-129 | 31 | 207 | 220 | 248 | 253 | 44 |
| AS/NZS 4777-2015 | 30 | 207 | 220 | 250 | 265 | 30 |
| ENA recommendation – 2019 | 41 | 207 | 220 | 240 | 258 | 60 |
| AS/NZS 4777 – 2020 (Australia B – small systems) | 30 | 205 | 220 | 235 | 255 | 40 |

Taken from CANVAS Final Report

It means, the possible range for V3 and V4 are:

V3: 235 – 250 V

V4: 253 – 268 V

However, we put 1.5 V margin to take some measurement random error into account, so the possible range for V3 and V4 are:

V3: 233.5 – 251.5 V

V4: 251.5 – 269.5 V

# Progress 11 October 2022

* Removed all functions that are not required
* Finished transforming the script to OOP using class. All functions are changed into class method
* Tested implementing from PyPI: pip install trialsamhan2==22.0.0

Implementation:



Text

Description automatically generated

Inside solarcurtailment.compute:

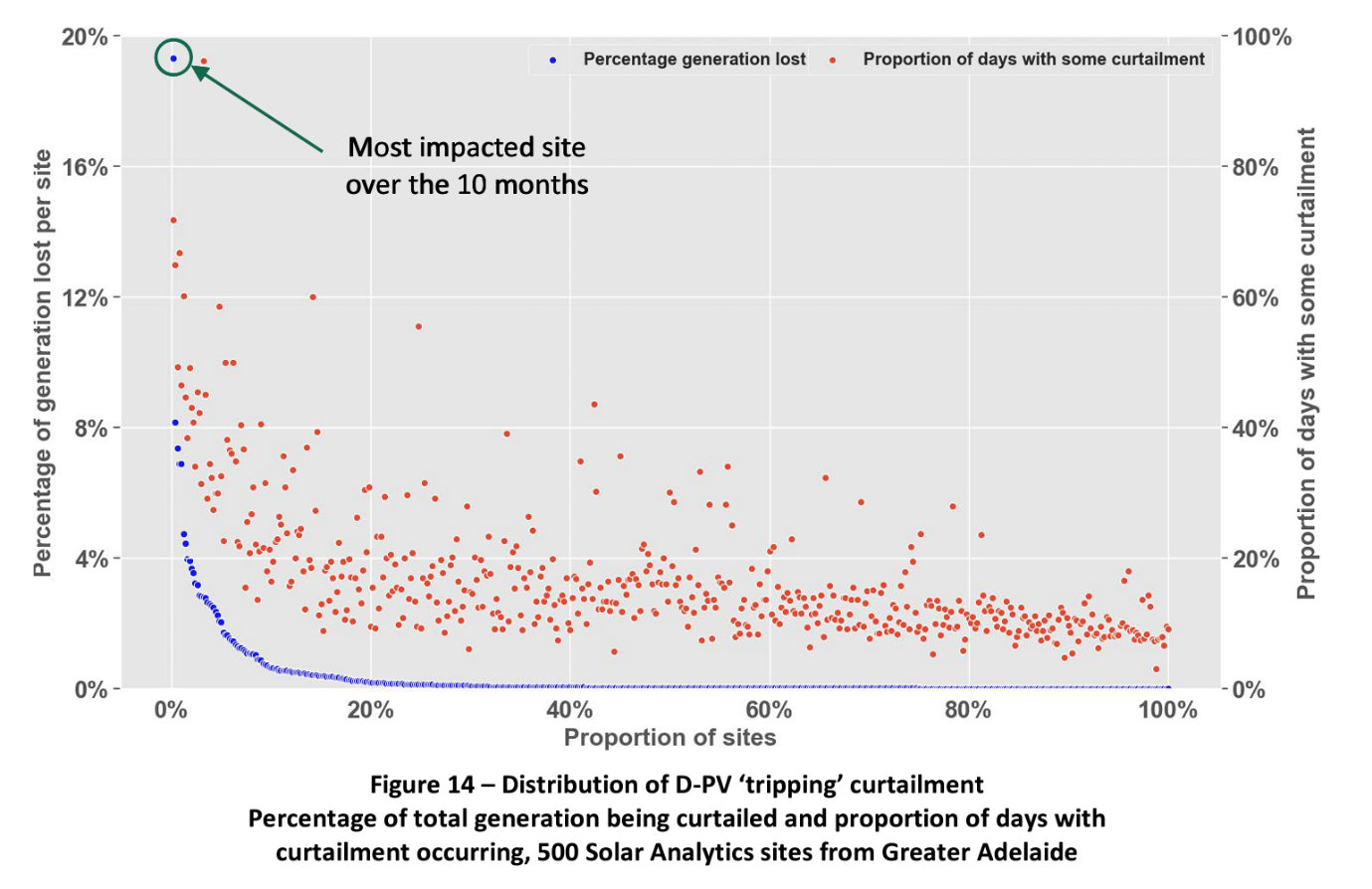
Text

Description automatically generated

# 20 October 2022

## 1. Fig. 14 plot for only clear sky days

This is figure 14 plot from the final report, showing the percentage of total generation being curtailed due to tripping and the proportion of days with curtailment.



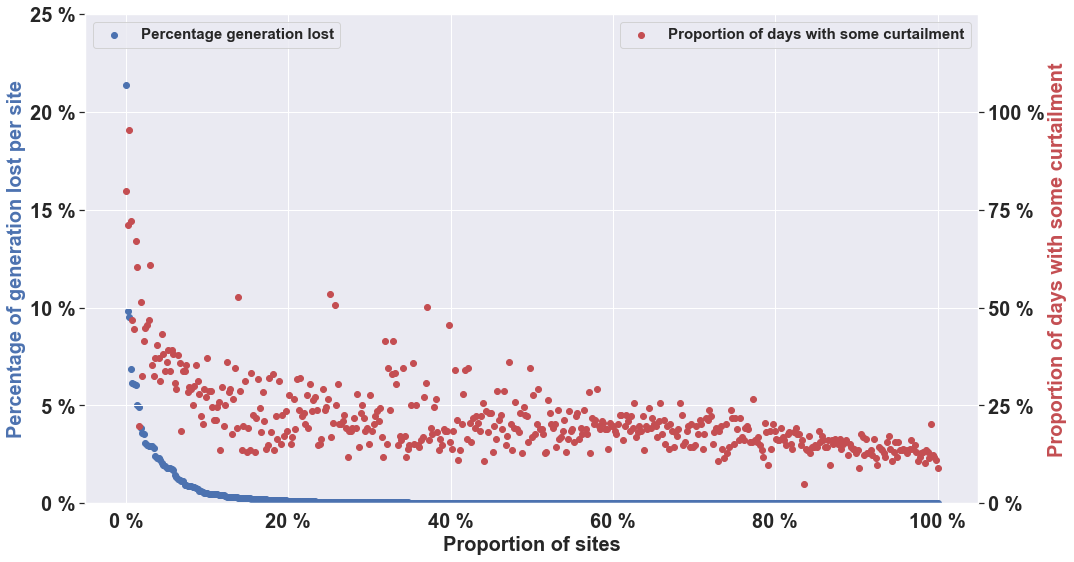
Next, this is similar plot but analyzing only the clear sky days.

Chart, scatter chart

Description automatically generated

The biggest percentage generation lost is increased from less than 20% to nearly 25%. We can also see that if we analyze only clear sky days, some sites do not experience curtailment at all for all days (last 20% of the data).

And then, this is a plot for tripping curtailment for all days, including the non clear sky date.



If we compare the scatter plot in this figure and figure from the final report (which was generated by Naomi), we can see that there is a difference. The maximum percentage of generation lost per site from the newsest calculation is more than 20%. In addition to that, if we see the actual number of energy generated in the spreadsheet, the calculated energy generated is much higher than naomi’s calculation. It is suspected that the days which Naomi’s analyse are fewer than 8 months of data.

Interestingly, the calculated energy generation by Samhan is also lower than Baran and Tim’s calculation. This may be because of there are some days, when Samhan does not analyze. He filters out days where there is no data between 7 and 17, also filters out when the total datapoint in a day is less than 1000.

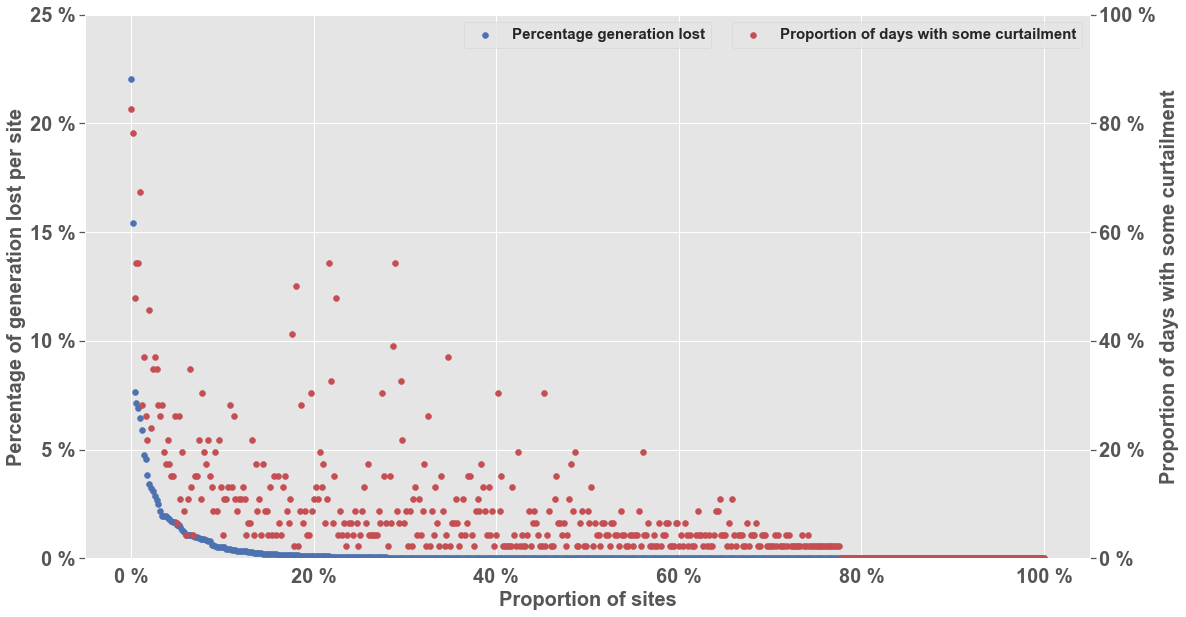
## 2. Updated summary spreadsheet with only clear sky days data

The summary spreadsheet has been updated with the file name SolA Curtailment Summary\_Final\_tripping.xlsx

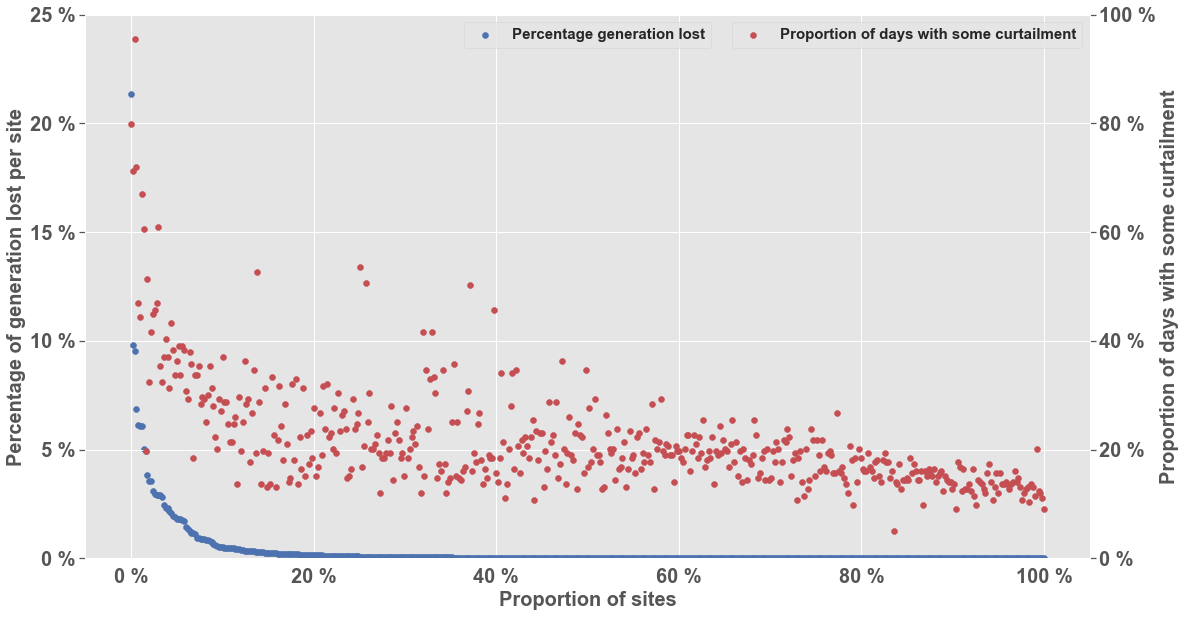
The script used to produce those 2 results are available in /other/Tripping in Clear Sky Days Analysis.ipynb

# November Week 1

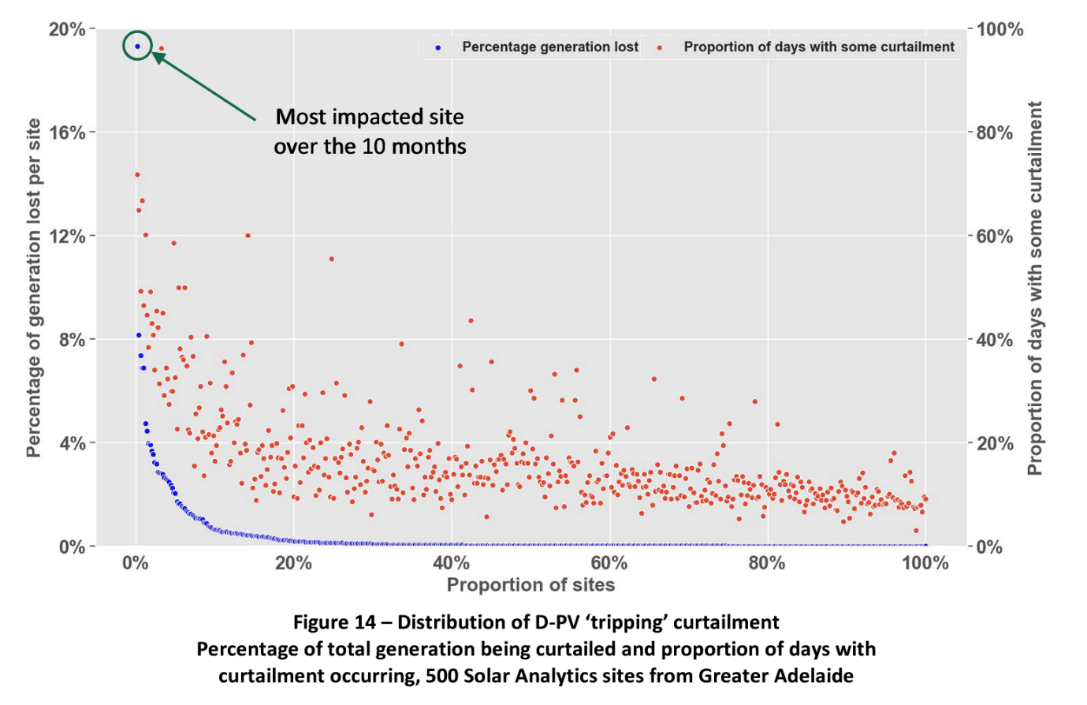
* Updated the tripping plot for clear sky days only and all days, with the legend put on the top right and top. Results:
* Only clear sky days:



* In all days:



* Comparison with naomi’s result (which is not clear only csd or all days):



Compared to Naomi’s result, our result shows higher percentage of generation lost per site which is more than 20%. Also, the plot for all days differs from the plot for only clear sky days in terms of the proportion of days with some curtailment. Surprisingly, in all days, there are more proportion of days with some curtailment compared to only in clear sky day. It means, tripping curtailment happens not only mainly in the clear sky day but also the non clear sky day in general.

* Made 5 plots for tripping in a non clear sky day for the most curtailed site, which is c\_id 1317822057. We chose to analyze October because in spring the PV production is not bad, but the load demand is low compared to summer and winter so the expected curtailment is high.
* Result:

1. Date 2019-10-08, Percentage curtailed = 37.03 %

Chart, histogram

Description automatically generated

1. Date 2019-10-09, Percentage curtailed = 30.39 %

Chart, histogram

Description automatically generated

1. Date 2019-10-10, Percentage curtailed = 35.75 %

Chart, histogram

Description automatically generated

1. Date 2019-10-11, Percentage curtailed = 62.23 %

Chart, histogram

Description automatically generated

1. Date 2019-10-15, Percentage curtailed = 21.66 %

Chart, histogram

Description automatically generated

Compared to the figure in the final report,

Chart, histogram

Description automatically generated

Seems like the final report uses lower data resolution. So, there are a lot of missing points. Regardless, the percentage of generation curtailed is more or less the same, 61% vs 62.23%.

* Comparison between Naomi’s CSD and Tim’s CSD

From 2019-09 until 2020-04, there are no days where Naomi judges as a clear sky day, but Tim judges as a non clear sky day. However, there are 14 days where **Tim judges as a clear sky day, but Naomi judges as a non clear sky day.** We analyse those 14 days in these tables, based on its ghi and power data:

|  |  |  |  |
| --- | --- | --- | --- |
| Date | GHI Plot | Power Plot | Visual Inspection of GHI |
| 2019-09-18 |  |  | Non CSD |
| 2019-09-26 |  |  | Non CSD |
| 2019-09-29 |  |  | Non CSD |
| 2019-09-30 |  |  | CSD |
| 2019-10-03 |  |  | Non CSD |
| 2019-10-29 |  |  | Non CSD |
| 2019-12-28 |  |  | Non CSD |
| 2020-01-17 |  |  | Non CSD |
| 2020-02-13 |  |  | Non CSD |
| 2020-02-23 |  |  | Non CSD |
| 2020-03-03 |  |  | Non CSD |
| 2020-03-10 |  |  | Non CSD |
| 2020-03-14 |  |  | Non CSD |
| 2020-04-08 |  |  | Non CSD |

Based on visual inspection on the GHI plot, 13 dates are actually a non clear sky day. So, probably it is worth to later think about improvement to Tim’s algorithm.

Regardless, it seems like the difference between the calculated energy generation from Naomi and Samhan is not due to this clear sky day. It is more because Samhan filters out days when the data is incomplete. Will analyse this further.

* Analyzing c\_id 1317822057, Samhan’s algorithm skipped analyzing these dates:

2019-09-01, 2019-10-01, 2019-11-01, 2019-12-01, 2020-01-01, 2020-02-01, 2020-03-01, 2020-03-21, 2020-03-22, 2020-03-23, 2020-04-01.

First day of the month is always skipped due to the monthly data shifting with the timezone.

However, 2020-03-21 until 2020-03-23, we should analyse it further. Turns out it because the algorithm used filters out days when there is no power more than 300 watt. We changed it into 5% of the ac capacity of the inverter so it is still consistent with naomi’s script, and now we analyzed 2020-03-21 and 2020-03-23. Screenshot:

Text

Description automatically generated

I modified the script already to make sure those dates are included, but it is still not working because the data is too big. Probably will try to do it again later.

# Energy Calculated Difference

Problem: There is difference in the total energy generated calculation from Tim & Baran vs Samhan. For instance, for c\_id 1317822057, which is the most curtailed site, the total energy generated from September 2019 to April 2020 is:

|  |  |  |
| --- | --- | --- |
| Tim (kWh) | Baran (kWh) | Samhan (kWh) |
| 5719 | 5510 | 5017 |

For other sites, we found that on average, Samhan’s result is only 87.5 % of Baran’s result.

## Hypothesis 1: There is difference in the calculation for each day

We checked already for date 2019-09-03, c\_id 1317822057, that there is no significance difference:

|  |  |  |
| --- | --- | --- |
| Tim (kWh) | Baran (kWh) | Samhan (kWh) |
| 11.17 | 11.14 | 11.18 |

In fact, the energy calculated by Samhan is bigger than Tim and Baran. Details in https://github.com/mssamhan31/Solar-Curtailment/issues/2

## Hypothesis 2: There are some days which are not computed by Samhan

We checked Samhan’s algorithm and found that he does not calculate every first day of the month, which are 2019-09-01, 2019-10-01, …, 2020-04-01. This might be due to the time zone difference. However, when we see the original data at UTC (before converting to Adelaide time), there is in fact missing data for every 1st day of the month. The data collection always start at 13.00 UTC at the 1st day of the month. Meaning, in Adelaide, it is already 22:30 so the day cannot be analyzed.

### September 2019

Table

Description automatically generated

## October 2019

Table

Description automatically generated

## November 2019

Table

Description automatically generated

December 2019

Table

Description automatically generated

## January 2020

Table

Description automatically generated

### February 2020

Table

Description automatically generated

### March 2020

Table

Description automatically generated

### April 2020

Table

Description automatically generated

## Hypothesis 3: There is missing month analyzed

In Samhan’s script, there are 8 months analyzed which are September 2019 until April 2020. Turns out in the final report, Baran and Tim analyze 10 months which are July 2019 until April 2020. Now, if we analyze 10 months using Samhan’s script, we get this result for c\_id 1317822057:

|  |  |  |
| --- | --- | --- |
| Tim (kWh) | Baran (kWh) | Samhan (kWh) |
| 5719 | 5510 | 5729 |

**That’s it!**

# Missing Site Id

Problem: There are only 499 sites in the UniqueCids file, while in the report there are 500 sites.

The missing site id is:

|  |  |
| --- | --- |
| c\_id | site\_id |
| 1471916197 | 853897796 |

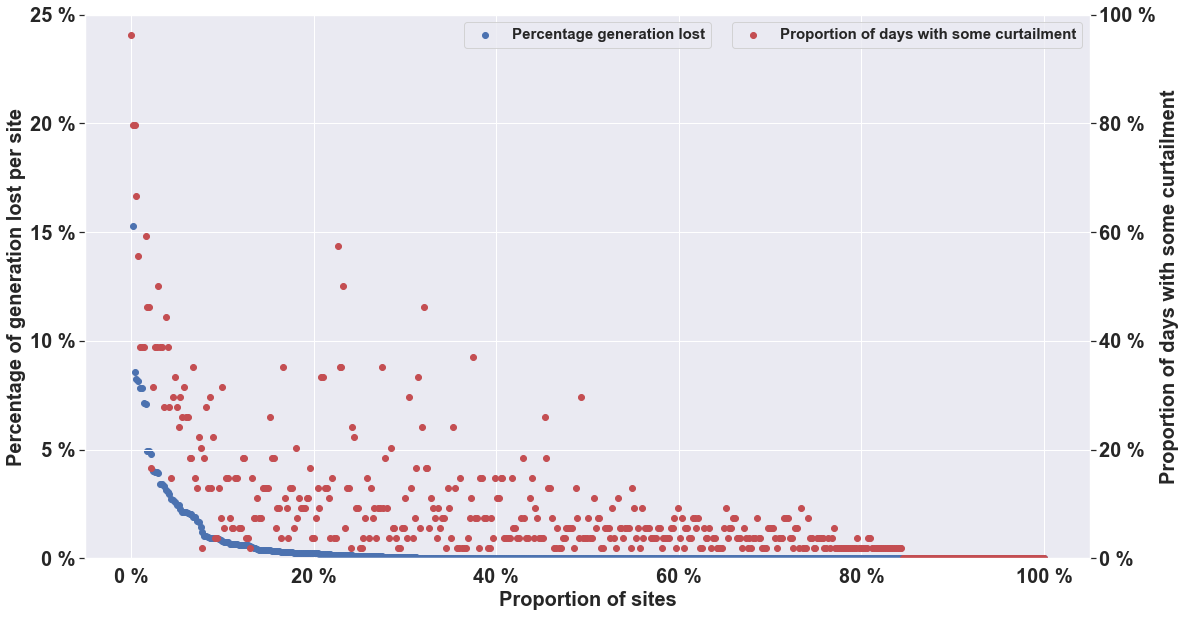
I will make a new list called UniqueCids500, which includes this site. Also, the package will use this file for further analysis.

# Revision of Tripping Analysis

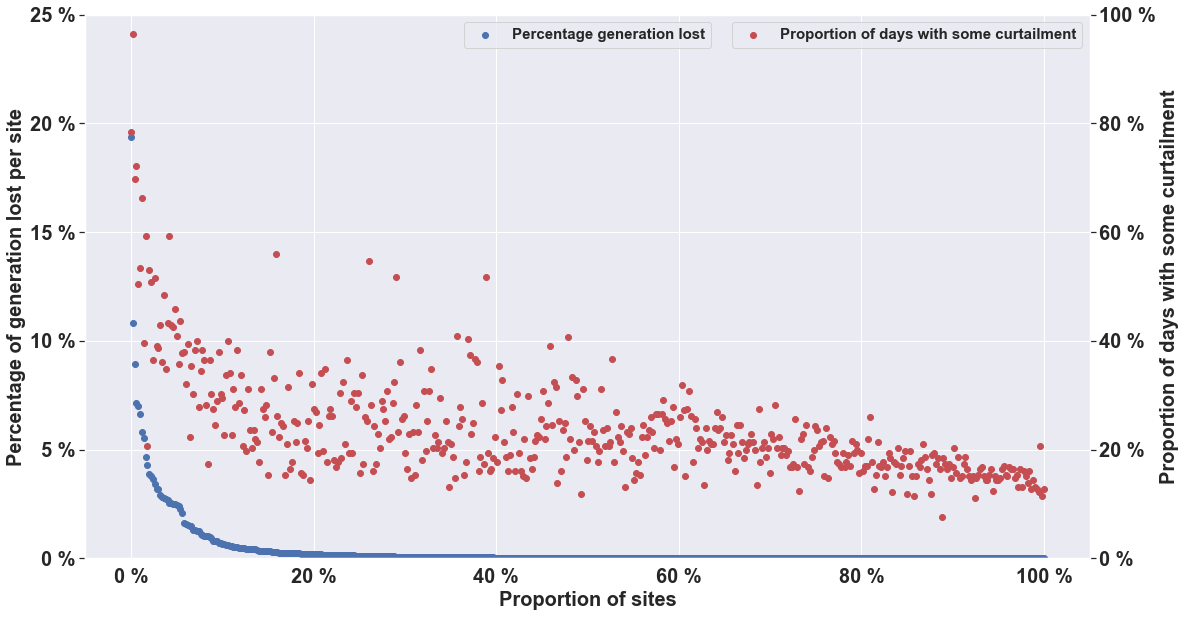
Because earlier we did not analyse 10 months, we will redo the computation again for 10 months for CSD only and for all days.

We will divide each computation into 10 batches each for 10 sites, so the checklist will be like this:

|  |  |
| --- | --- |
| Name | Checklist |
| CSD batch 1 | V |
| CSD batch 2 | V |
| CSD batch 3 | V |
| CSD batch 4 | V |
| CSD batch 5 | V |
| Create summary CSD spreadsheet | V |
| Plot CSD result | V |
| All days batch 1 |  |
| All days batch 2 |  |
| All days batch 3 |  |
| All days batch 4 |  |
| All days batch 5 | V |
| Create summary All days spreadsheet |  |
| Plot All days result |  |

Now, the result for CSD only: 

The result for All days:



Naomi’s result:

Chart, scatter chart

Description automatically generated

Not much different from Naomi’s result.

# Reference

Packaging project https://packaging.python.org/en/latest/tutorials/packaging-projects/