**1 Overview**

Ice Nucleating Particles (INP) are … One in a million particles act as INP at -20C

They are crucial to determining various properties of clouds, including precipitation rates, lifetime, and shortwave reflectivity and longwave emissivity. Since the effects of aerosols on cloud optical properties and radiative forcing is the single most uncertain component of radiative forcing of Earth’s climate, this makes understanding INP massively important.

The field is nascent and fast moving. Not only that, but uncertainty is on the scale of orders of magnitude. Any time not spent wrangling and preprocessing data can be spent finding high impact results. Pyce Tools addresses these two problems. First, by offering a set of guidelines for INP data workup means less errors during data workup. Second, it’s easy to prepare, so you can spend less Spend less time working up your data, and more time finding high impact results.

This getting started guide briefly describes the Pyce Tools nomenclature, the processing workflow, and the additional data analysis steps included in the package. Further details are found in the code documentation (in the pyce\_tools.py file). Users can also reference the accompanying jupyter notebook tutorial file.

A final disclaimer: this code was written for the Sea2Cloud Tangaroa Cruise. As such, it assumes input files are organized in a certain way. I tried to keep things broad so that the code could be extended to future cases, but don’t be afraid tweak the source code as you see fit.

**2 Definitions**

The Pyce Toolsworkflow has a very strict naming convention. For this reason, it’s important to first define some terms. INP samples come in various **TYPES**, which can be collected from a range of **LOCATIONS,** and are subject to different **PROCESSES** before analysis with the LINDA instrument.

**TYPE:** *[aerosol, seawater, mq]* the category of the INP sample. This is most generally either seawater or aerosol. Note that if the sample is a seawater blank, it is defined as ‘mq’. If the sample is an aerosol blank, it is still classified as aerosol, since the blank used is a filter.

**LOCATION:** *[uway, wboatsml, wboatssw, bubbler, coriolis]* where the sample was collected. For instance, *seawate*r type samples can come from the boat’s underway (*uway*) or from workboat deployments (*wboat*). Workboat measurements are further specified as either *wboatsml* or *wboatssw* depending on where in the water column the sample originated. *Aerosol* type INP samples can come from a *bubbler*, *coriolis*, or *ambient* measurements. For blanks samples, the location options are *bubbler, Coriolis, mq, mq\_wboat*. This is kind of a weird naming convention. Sorry.

**PROCESS:** *[uf, f]* a technique used on an INP sample to gather more information. Possible processes include HEATING the sample or FILTERING it. Others include H2O2 but we do not do that in these samples. The way the code is written currently is that the process of either filtering or leaving samples unfiltered is described in the sample name, while heated and unheated is implicitly assumed to have been conducted during initial sample analysis. For this reason, only UF/F is specified in sample names.

These three terms defined above are used throughout the Pyce Tools workflow, which is described in the following section.

**3 INP Analysis Workflow**

Throughout the workflow of an INP sample, several file types are made. They are termed *raw, calculated,* and *cleaned.* In general, the workflow and how each file type fits into it is generalized as follows:

1. An INP sample of a specific type is collected from a specific location.
2. The sample then undergoes whatever post-collection processes the experimenter decides.
3. The sample is analyzed with LINDA and*raw* data files are saved to their appropriate folder with the appropriate naming convention.
4. Raw data files are then pre-processed into *calculated*report files using sample template spreadsheets. This step involves, among other things, subtraction of the blank (see **Section 3.3 Blank Correction**). *Calculated*report files are then saved to their appropriate folder.
5. *Calculated* report files from the calculation step are then cleaned and saved as *cleaned* output files, which are ready for analysis. During this step, error bars can be calculated and saved in separate files as well.

The specifics of these steps are further described in the following sections.

**3.1 Saving Raw Data Files**

It is imperative that raw files are saved in the correct location and with the correct names. The rules are described for the two sample types (i.e., *aerosol or seawater*).

Seawater Sample Type Naming Convention:

*seawater\_[location]\_[process]\_[DDMMYY]\_[HHmm].csv*

These files are saved into the following folder: ..\\*[PROJECT\_ROOT]\\data\\raw\\IN\\[SAMPLE\_TYPE]\\[FILE]*

Aerosol Sample Type Naming Convention:[[1]](#footnote-1)

*aerosol\_[location]\_[process]\_[size]\_[DDMMYYYY]\_[HHmm].csv*

These raw files are saved into the following folder: ..*\\[PROJECT\_ROOT]\\data\\raw\\IN\\[SAMPLE\_TYPE]\\[FILE]*

**3.2 Creating Calculated Sample Report Files from Raw Data**

*Calculated* report files are created by processing the raw data using either the *calculate\_raw\_blank()* or calculate\_raw() functions. Raw blank calculation is described in Section 3.3 below. Here we describe the process of creating a calculated sample report file for raw data, assuming that calculated blank data files are ready for use. [[2]](#footnote-2)

Raw data files are processed using template spreadsheets. There is a template for seawater and aerosol sample types which should be placed in your project root directory. These template spreadsheets have the necessary equations already inside of them so that the code simply needs to place the values in the correct locations.

The overall process of calculating report files is as follows:

1. Raw file is loaded.
2. Metadata is calculated and listed for raw data source, type, location, process, sample source name, sample collection date, sample analysis date, number of tubes, ml per tube, issues, and sigma. Sigma is used for error bar calculation and should be left at 1.96 for confidence intervals of 95%.
3. If the sample is *aerosol type*, additional metadata is calculated and used in INP calculation. These parameters are rinse volume, size, average flow, sample collection time (calculated from start and stop time) and total sampled air volume (calculated from sample collection time and average flow).
4. Raw data is loaded into the template spreadsheet and calculations are made.
5. A blank data file is loaded into the template file and used to subtract from raw data.

The calculated report file is then saved to its appropriate location according to the following convention:

*\\[PROJECT\_ROOT]\\data\\interim\\IN\\calculated\\[SAMPLE\_TYPE]\\[SAMPLE\_LOCATION]\\[type]\_[location]\_[process]\_[date]\_[time]\_calculated.xlsx*

You will want to check over the calculated report file yourself as the template may not calculate across all temperatures conducted in your specific experiment. Simply extending the equation further down to lower temperatures by dragging a cell should suffice. See the Tutorial in Section 6 for more information.

**3.3 Blank Correction**

As mentioned in the previous section, INP raw data files from the LINDA need to be blank corrected. To do this, blanks are collected, analyzed with LINDA, calculated, and finally subtracted from experiment raw data files during the calculate\_raw() function. The following functions are used in the preparation of blank files.

**calculate\_raw\_blank()** function to calculate raw data files of blanks. When conducting LINDA experiments, the name of the blank file should follow this naming template:

For seawater type samples:

*mq\_blank\_[process]\_[DDMMYY]*

For aerosol type samples:

*[type]\_blank\_[process]\_[size]\_[collection\_date]*

It is crucial that naming conventions are followed, so see specific function documentation for more details.

The calculated report file will include metadata on the following: raw data source, type, location, process, sample source name, sample collection date, sample analysis date, number of tubes, volume (ml) per tube.

If the sample is type aerosol, the metadata will also include rinse volume and size.

The calculated report file is then saved to the appropriate folder. See function documentation for specifics.

**3.4 Cleaning Calculated Report Files**

The **clean\_calculated\_in()** function loads and cleans each calculated report file into a format that’s easier to use in Python. Some processes include renaming columns and reshaping the dataframes. Most importantly, the function combines all the calculated report files in a folder into a single combined time series file. This means you will want to ensure each experiment has its own folder.

The output of this function is a CSV where each row is an observation and each column is a temperature. Seawater sample types will also have columns for datetime, time, process, type, location, and filtered/unfiltered indicator. The IN values are given in IN/mL of water. In addition to these columns, aerosol sample types from the bubbler or Coriolis will also have size (bubbler only), start\_date and stop\_date. IN values are given in INP/L of air.

Cleaned files are saved to: *\\[PROJECT\_ROOT]\\data\\interim\\IN\\cleaned\\combinedtimeseries\\[SAMPLE\_TYPE]\\[location]\_[start\_date]\_[end\_date].csv*

**3.4.1 Calculating Uncertainty**

Error bars are usually given as xxx. This is carried out using a combination of equations in cells in the calculated report file as well as some additional python code. It’s not pretty but it gets the job done. The output csv file is saved in the same location as the cleaned combined time series data file described in Section 3.4 and with the same naming convention, but with ‘wilson\_error’ appended to the end.

Lower and upper bounds for blank subtracted frozen fraction of tubes (upperBound, lowerBound) are calculated using subfunctions (wilsonLower() and wilsonUpper()). These fractions are then converted to a number of blank subtracted tubes that are frozen (upper\_N-BLNK, lower\_N-BLNK, respectively). These bounds are then converted into INP/tube upper and lower bounds. Then they are converted to IN/mL and IN/L upper and lower bounds. Finally, the difference between each bound and the original observed value is calculated to determine the size of the error bars and saved as error\_y and error\_minus\_y. The confidence interval of the uncertainty can be changed by using a different sigma value in the template spreadsheets.

For seawater samples, the units are INP/L seawater. For aerosol samples, the units are INP/L air.

**3.5 Loading and Final Pre-Preprocessing**

While Pyce Tools does a bulk of the necessary manipulations and pre-processing, there’s a few steps that should still be completed after loading the cleaned combined timeseries files. This is left to the user’s discretion as the specifics of each experiment will vary considerably. For reference, the preprocessing steps carried out for S2C data are shown below.

* Convert datetime column to a pandas datetime index
* Melt the dataframe
* Calculate different units
* Round to nearest hour
* Set to NZ time zone
* Merge uncertainty and concentration dataframes

The code for these steps can be found in the jupyter notebook that accompanies the **Section 4 Tutorial.**

**4 Handling Particle Size Distribution Data**

Particle size distribution data is crucial as it is needed to calculate surface area normalized INP concentrations of SSA. Pyce Tools includes some functions for loading, visualizing, and preparing size distribution data for normalization of INP.

Inverted data from the scanotron is cleaned and concatenated into a single combined timeseries file using **clean\_inverted()** function. For processing, all inverted files should be saved into a single folder. Here, we choose the following directory path:

*\\[PROJECT\_ROOT]\\data\\interim\\scanotron\\inverted\\pro\\[FILE]*

The out path for the cleaned size distribution file can be defined by the user. Here we choose the following:

*\\[PROJECT\_ROOT]\\data\\interim\\scanotron\\combinedtimeseries\\*

Concentrations are lognormalized. As such, the **clean\_inverted()** function accepts the number of size bins as a parameter.

After cleaning inverted scanotron data, it can be loaded using **load\_scano\_data()** and further manipulated. Parameters for the **load\_scano\_data()** function include dates, which is the name of the combined time series file you want to load, and instr, which tells where the file is located.

The pyce\_tools.py file includes further documentation on the rest of the functions, which include:

* Surface area can be calculated using **pt.surface\_area()**
* Magic CPC data can be cleaned using **pt.clean\_magic()**
* Create plots using **plot\_number\_dist()** and **plot\_surface\_dist()**

**5 Analysis**

**5.1 Creating INP Objects**

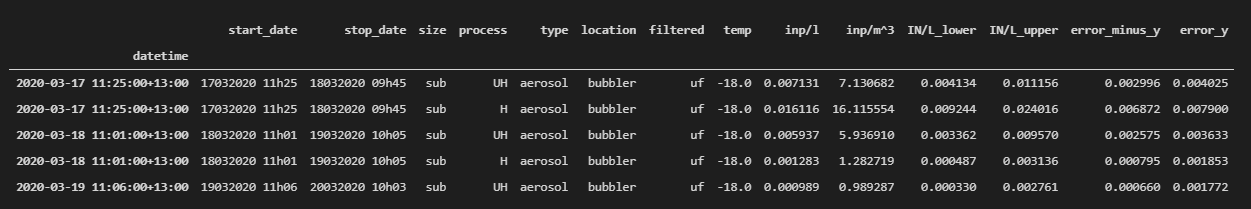
Analysis of INP data is achieved using the INP class. An INP class object consist of INP data of a single type from a single location, a uway\_bio dataframe consisting of observations from the ships underway, and a cyto\_bio dataframe which can have further biology data from any location. In this way, we have a final INP object consisting of INP data from a specific location and of a certain type, which will contain multiple processes and temperatures and filtered/unfiltered states.

Below are some examples of what is contained in an individual INP object:

* sml\_inp
  + type: seawater
  + location: sml
  + processes: uf; UH and H
  + temperatures: 0 through -18C
  + a uway\_bio dataframe
  + a cyto\_bio dataframe from workboat sml
* uway\_inp
  + type: seawater
  + location: uway
  + processes: uf; UH and H
  + temperatures: 0 through -18C
  + a uway\_bio dataframe
  + a cyto\_bio dataframe from uway

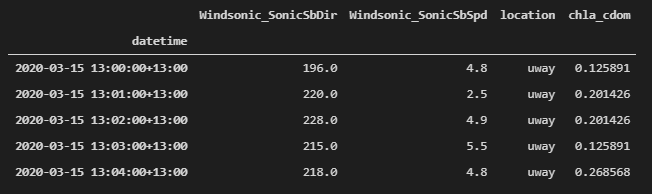
INP objects are instantiated when given an inp\_type, inp\_location, cyto\_location, a cyto\_data dataframe, a uway\_bio dataframe, and an inp\_data dataframe. **See the pyce\_tools.py file for in depth details.**

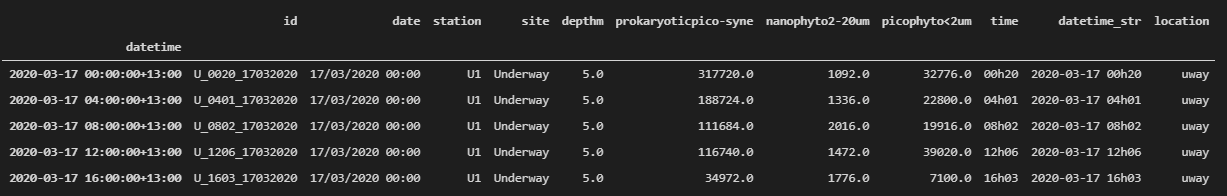
Here is an example of what the inp\_data dataframe should look like. This format is easily achieved if you’ve followed the steps outlined thus far.



This dataframe is basically just the original INP dataframe merged with the uncertainty dataframe. It is important that the index be titled ‘datetime’ and be a datetime object. Code for how to do this is found in the tutorial section. Note that you can pass in a dataframe consisting of multiple locations and types, but the code will automatically keep only the data that corresponds to your selected type and location as defined when instantiating the object.

The uwaybio dataframe can include any data. The only requirements are that the index is a datetime object with the name ‘datetime’ so that it can be lined up correctly with the INP observations and a location column is defined.



The cyto\_data dataframe (example below) will look similar to the uway\_bio dataframe. Again, you can pass in a cyto\_bio dataframe that contains a mixture of locations, but the function will automatically only keep data from your specified location. This means you need to make multiple objects if you have cyto data from multiple locations.

**5.2 Calculating Surface Area Normalized INP Concentrations**

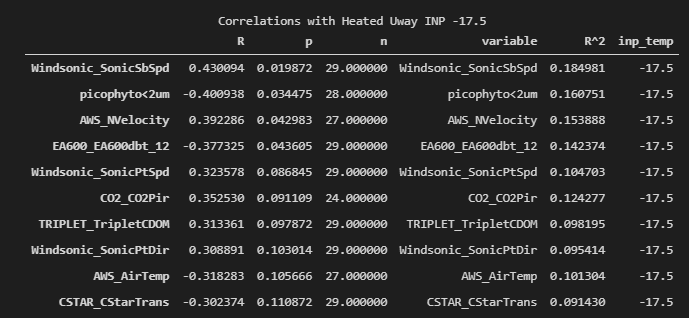
Surface area normalized INP concentrations are calculated using the inp object’s **sa\_normalize()** method. The dA\_total dataframe, which is returned from the **surface\_area()** function, is passed as a parameter. This function assumes you have already organized dA\_total to line up with your INP collection periods. See the tutorial for specifics.

**5.3 Plotting with error bars and previous studies**

A plot of aerosol INP vs literature values is done through the **plot\_ins\_inp()** method. Seawater INP (ssw and sml) plots are not object methods but rather Pyce Tools functions (**plot\_sml\_inp(), plot\_ssw\_inp()**).

**5.4 Correlations and correlation scatter plots**

Correlations are calculated using INP object’s correlations() method. A list of temperatures as strings are sent, as well as a specific process (H, or UH) and inp\_units string, which indicates the column containing your INP concentrations. Below is an example of the returned dataframe of correlations. See tutorial and code documentation for more details.



The correlations can also be viewed with scatter plots by using the **plot\_corr\_scatter()** method, which returns a figure object which can be further stylized. See the tutorial for examples.

**6 Tutorial**

As an example, we will go through the steps up to the calculation of clean INP data. We use data from samples collected during the S2C Tangaroa cruise. The sample type is seawater and was collected at the wboatssw sample location.

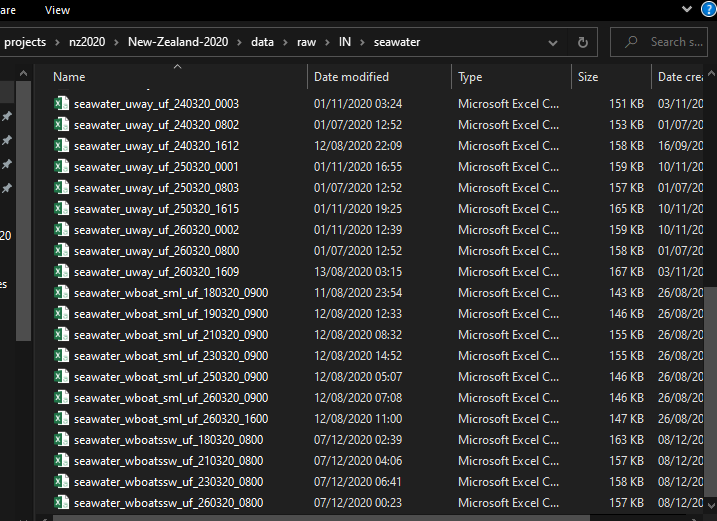
**Load Pyce Tools**

First, load pyce tools. You need to append your system path to wherever you have saved pyce tools (remember to import sys as well).

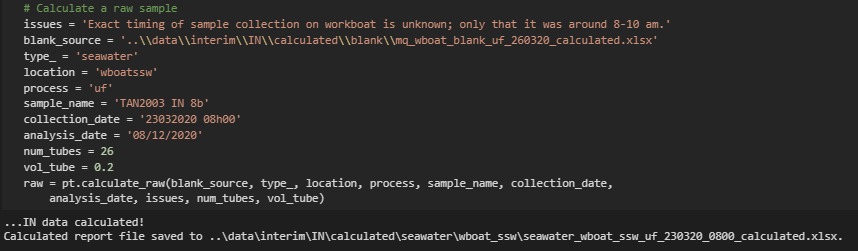
**Saving raw data files**

The following information should be logged into either a digital or hard copy file during LINDA analysis:

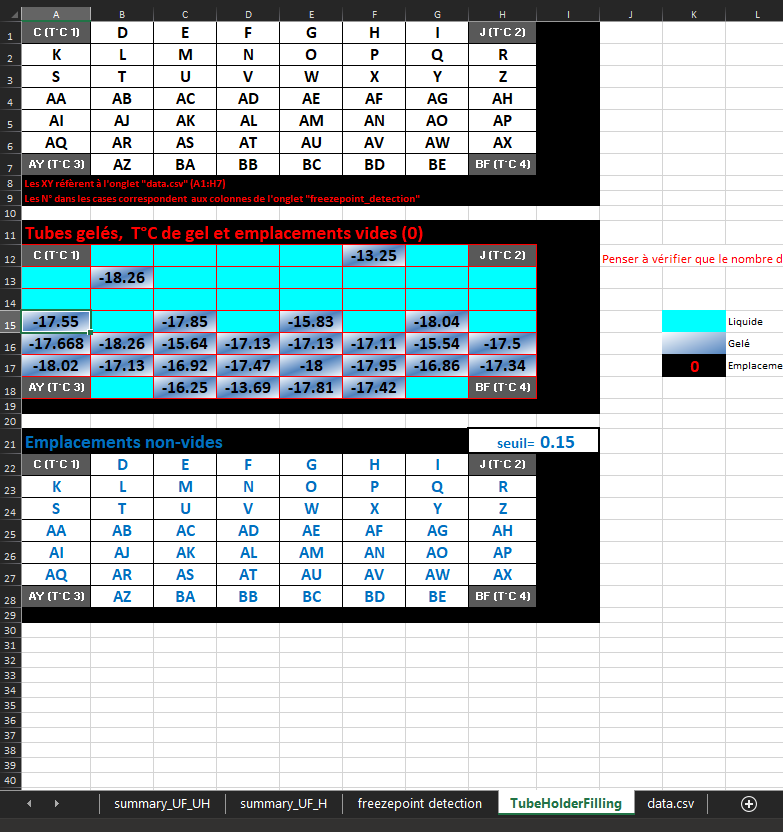
Type, location, process, sample name, collection date, analysis date, number of tubes and volume (mL) in each tube, issues. Filter aerosol samples need the following information in addition to the above: flow rate start, flow rate stop, rinsing volume (ml).

Below is an example of how the raw data files from the LINDA computer should be saved into your project workspace.

**Creating calculated report files**

 Assume we have a blank already made (for now we manually take the average of each blank file and create a new one which we point to in the calculation).

This will create a calculated report file in which the blank corrected values are given. The photo below shows an example of the report file. Here we are looking at the tubeholderfilling tab which gives an overview of all the cells. You should compare this with the final image from the LINDA to ensure the automated process correctly identified frozen vs nonfrozen cells.



Also note a separate tab for heated vs unheated samples is given (e.g., summary\_UF\_UH and summary\_UF\_H). The code is written to assume the bottom 26 cells are unheated and the top 26 cells are heated. You can alter this as needed. Each summary tab has calculations for blank corrected concentrations and uncertainty as well as metadata.

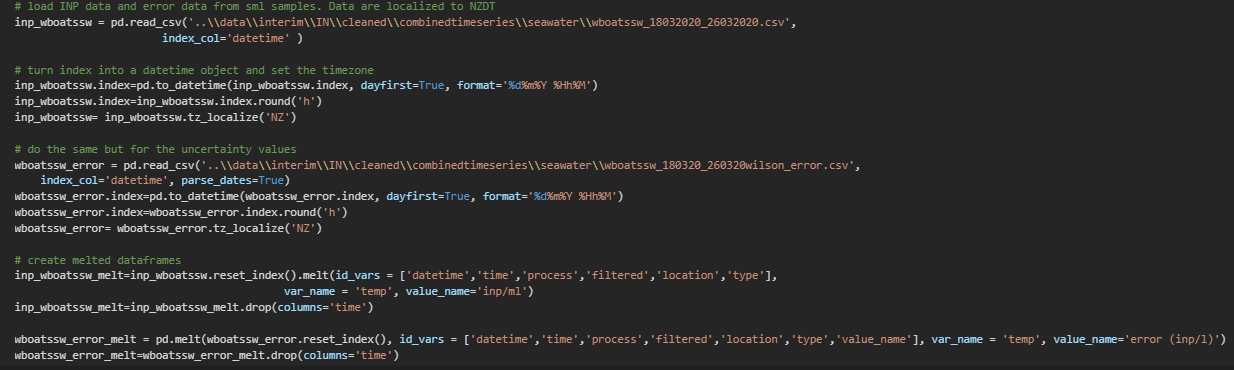
**Creating cleaned files and uncertainty files**

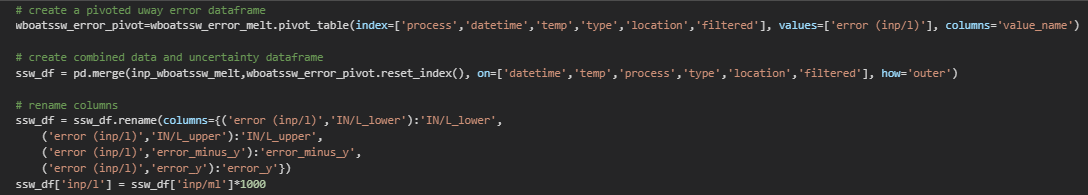
Cleaning the available calculated report files is as sample as calling **clean\_calculated\_in()** function and passing the parameters for INP type and location. The script will automatically find all relevant files in the folder and combine them into a single time series.



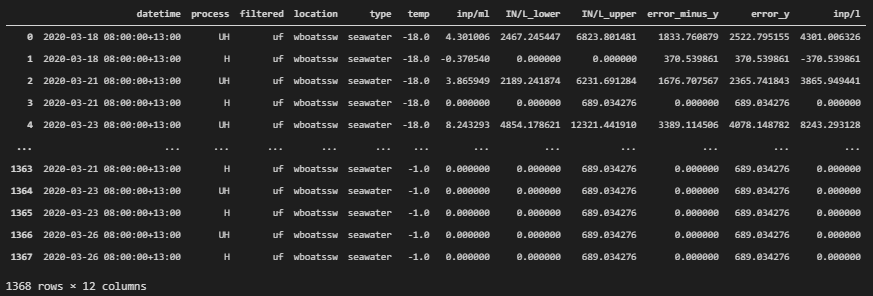
**Loading and preparing cleaned files for analysis.**

Load the cleaned INP data and uncertainty files. Do a final bit of preprocessing. This is unique to each experiment so has been left to the individual user.





Now we have a cleaned INP dataframe ready for analysis!



**Analysis**

See the attached jupyter notebook for an in-depth tutorial on loading and analysis of INP data.

1. When I initially began creating this workflow for aerosol sample types, I named files using ‘dayXX’. This was bad and I should not have done it. For this reason, there’s a section of code that uses a hash table to allocate dayXX with specific dates and times. Unless you are analyzing these specific samples where I did this (i.e., Coriolis samples from Tan2020 S2C), you can ignore that section of code. Going forward, files should be saved using the convention outlined below. [↑](#footnote-ref-1)
2. 2 Right now, only a single blank file is loaded. Eventually it would make sense to add functionality to average all relevant blank files into one file which is then subtracted from the data. For now, your best bet is to do this manually yourself by calculating several blank files individually (see Section 3.3 Blank Correction), and then averaging them into one blank file and passing that file as a parameter when calling the calculate\_raw() function. [↑](#footnote-ref-2)