# `tsp` ("teaspoon"): a python library for ground temperature data

*Nick Brown* [1,2]

*1. NSERC PermafrostNet* *2. Carleton University*

## Ground temperature data are essential to understand permafrost

Ground temperature data are foundational to permafrost science and engineering. These data represent measurements of ground temperature over time (time series) at one or more depths in the ground. Environmental monitoring, hypothesis testing and model development all rely on ground temperature data which may be obtained from a range of different sources including sensors, permafrost databases or model output. Unfortunately, cleaning, organizing and sharing data continues to be an obstacle in part because of the heterogenaeity of the data [1].

Members of the permafrost community have described continued challenges accessing standardized permafrost data [1]. Reusing existing tools has been suggested as a way to conserve resources and avoid the duplication of effort between research groups or initiatives. However, there are few existing tools designed specifically to handle permafrost or ground temperature data.

## Data formatting is not consistent

A survey of 37 data-producing institutions worldwide suggests that more than 50% of environmental science datasets use 'in-house' text formats to store and distribute data [2]. Consequently, most researchers rely on ad-hoc scripts to manage data, and new data sources commonly take a day or more to integrate into workflows. This is also true for permafrost science where there are multiple databases each with their own way of formatting ground temperature data which may differ in both structure and nomenclature.

Similarly, dataloggers for environmental science data can be configured to output text files in any number of heterogenous structures (Figure 1). Differences in date format, separator, and metadata structure confound attempts to develop reusable scripts to easily open datalogger files for further processing. A script that one scientist uses to read their files can fail when trying to open files that were configured differently, even when data are from a sensor of the same make and model. This makes it harder to share data or to develop common tools and workflows for quality control and analysis of observational data.


```
1,08/18/10 02:00:00  PM,69.557,69.771
2,08/18/10 03:00:00  PM,68.313,68.571
3,08/18/10 04:00:00  PM,43.624,43.900
4,08/18/10 05:00:00  PM,46.256,47.109
5,08/18/10 06:00:00  PM,45.534,45.806
6,08/18/10 07:00:00  PM,42.755,42.892
```

```
10-08-18,14:00:00.0,20.865,20.984
10-08-18,15:00:00.0,20.174,20.317
10-08-18,16:00:00.0,6.458,6.611
10-08-18,17:00:00.0,7.920,8.394
10-08-18,18:00:00.0,7.519,7.670
10-08-18,19:00:00.0,5.975,6.051
```

Figure 1: Sensor data downloaded from the same datalogger and exported to a text file can differ significantly in formatting depending on how it is configured. In this example, ground temperature data from an Onset HOBO device have different date formatting, different units, and different numbers of columns. `tsp` provides datalogger-specific file readers to handle this heterogenaeity.

## Easily read, write, summarize and explore data with `tsp`

`tsp` ("teaspoon") is a python library intended to simplify how permafrost researchers work with temperature time series (Figure 2). It does this by meeting three objectives:

1. Increase the ease with which ground temperature time series data from any source can be read into python.
2. Provide reuseable functions to perform the most common visualization and analysis conducted on permafrost data.
3. Improve permafrost data interoperability by providing methods to output data in common text file formats used by the permafrost community and as more standardized, self-documenting formats such as netCDF.

The software includes a collection of file readers designed to handle different model outputs (GEOtop), datalogger exports (HoboWare and Geoprecision) and files from territorial, national, and global permafrost databases (Northwest Territories Geological Survey and GTN-P). tsp also includes functions to produce common data visualizations to speed up the data exploration phase of a project.

`tsp` can be used as a standalone library for interactive data analysis and exploration or can be integrated into other programs and workflows.

## Why the name?

- The data represent the **T**hermal **S**tate of **P**ermafrost (or alternatively the **T**emperature du **S**ol à la **P**rofondeur).
- The data are represented by the "**T**ime **S**eries of **P**rofiles" CF discrete sampling geometry.
- It's short and memorable!

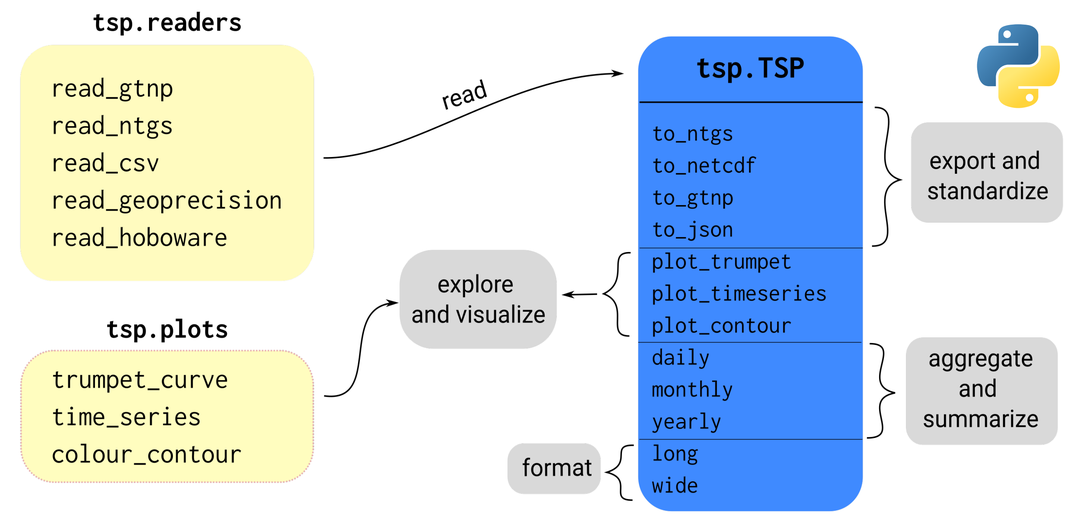


Figure 2: An overview of the tsp package showing the main functions (yellow) and classes (blue) that are included.

## "Re-use" instead of "re-do"

Researchers often rewrite the same code for visualizing ground temperature data in successive projects as input data changes slightly or as old files are lost. With `tsp`, the focus is on bringing data into a unified structure that can be visualized the same way regardless of the input data. With `tsp`, it is possible to start exploring data in as few as 3 lines of code (Figure 3).
Graphics for publications are often highly specialized and tailored for a specific purpose. For this reason, visualizations in `tsp` are targeted more towards data exploration instead of the creation of publication-ready figures.

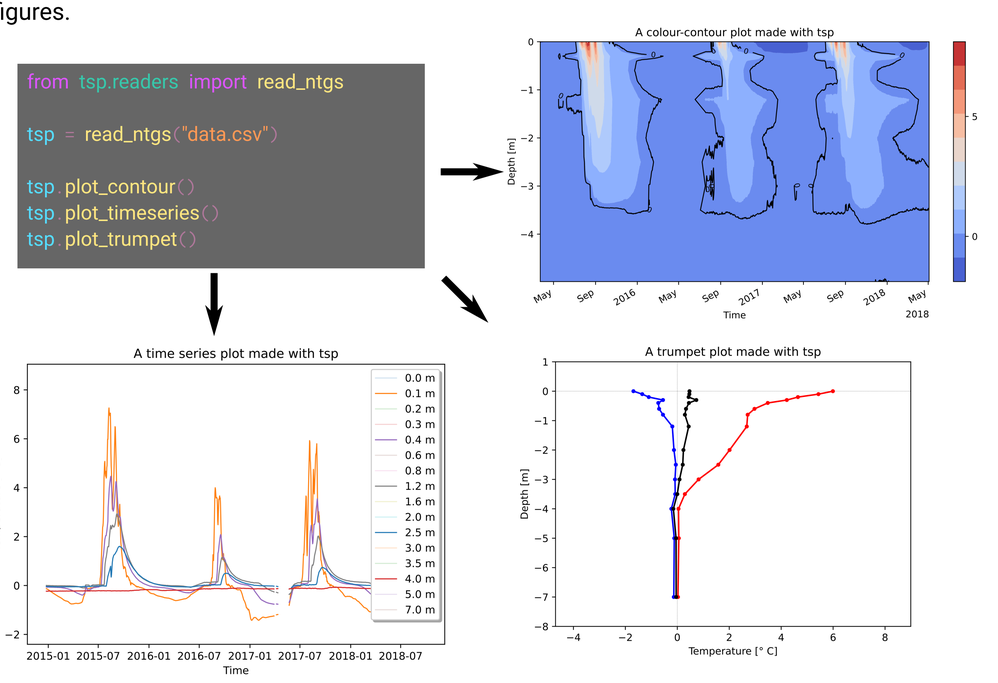


Figure 3: Once data have been loaded into the common data structure, tsp provides common visualizations using the matplotlib library. These can be used for interactive data exploration, or integrated into other dashboards or workflows. Plotting windows can be panned, zoomed, and saved.

## Discover, comment, collaborate

`tsp` was designed with participation in mind to make the tools useful as a shared community resource. Contributions are encouraged and the code is distributed under an open-source license (GPLv3). Next steps for the project may include capabilities to work with two or more datasets at once—as model 'ensembles' or as comparisons of model output with observations—and to develop an online dashboard to enable the use of the tools without requiring any coding experience.

To contribute to the project, or to find out more, use your phone's camera to scan the QR codes below

**Documentation**

`permafrostnet.gitlab.io/teaspoon/`

**Share a suggestion**

`user:` **cpa-agm-2022**
`password:` **Permafrost**

**Code**

`gitlab.com/permafrostnet/teaspoon`

## References and acknowledgements

**[1]** Brown, N., Gruber, S., Pulsifer, P., Stewart-Jones, E. (2020) Permafrost Data Workshop Final Report. NSERC PermafrostNet. May 27-29, 2020: Ottawa, Canada. http://doi.org/10.22215/pn/10120001

**[2]** Bavay, M., Fiddes, J. and Godøy, Ø., 2020. Automatic Data Standardization for the Global Cryosphere Watch Data Portal. Data Science Journal, 19(1), p.6.DOI:http://doi.org/10.5334/dsj-2020-006

This project was enabled in part by support provided by Westgrid (westgrid.ca) and the Digital Research Alliance of Canada (alliancecan.ca).

CANADIAN PERMAFROST ASSOCIATION
ASSOCIATION CANADIENNE DU PERGÉLISOL

PermafrostNet
NSERC | CRSNG

Digital Research Alliance of Canada | Alliance de recherche numérique du Canada