**SNOWPYLOT: A PYTHON LIBRARY FOR WORKING WITH SNOWPILOT**

Mary Kate Connelly1

##### ABSTRACT

SnowPilot (https://snowpilot.org/) is a free, open-source software designed to help users graph, record, and store snowpit data. The SnowPilot database currently contains data from over 65,000 snowpits, collected by avalanche professionals and snow scientists around the world. It is particularly popular among avalanche professionals in the United States.

Despite SnowPilot’s widespread use to graph and record snowpit data, the SnowPilot database has been underutilized as a research resource. This is due in part to the complexity of the data and lack of available tools for accessing specific snowpit properties.

To address these challenges and improve accessibility to the database, we have developed SnowPylot, an open source Python library that enables researchers to import and structure data from the SnowPilot database within Python, facilitating the use of Python tools and methods for analysis. (Keywords: SnowPilot, SnowPylot, CAAML, Python, Snowpit)

##### INTRODUCTION

Snow pit observations are fundamental to avalanche safety and research, providing critical information about snowpack structure and stability. Since its inception, SnowPilot has served as a crucial platform for collecting and sharing these observations within the avalanche community. However, while SnowPilot has successfully amassed a vast database of snow pit observations, the tools for accessing and analyzing this data have remained limited. Exploratory analysis of the information in the database was difficult and time consuming.

The development of the SnowPylot Python library (https://github.com/connellymk/snowpylot) addresses this gap by providing a suite of tools for processing and analyzing snow pit data from SnowPilot. This library, designed to work with files exported in the Canadian Avalanche Association Markup Language (CAAML) format, enables researchers and practitioners to process snow pit observations at both individual and large-scale levels.

The information in the CAAML files is parsed into Python objects with a structure that mirrors the CAAML format. This allows the user to leverage Python tools and libraries for analysis and makes it possible to examine patterns and trends across thousands of snow pit observations, potentially revealing new insights about snowpack behavior and avalanche formation.

##### DATA STRUCTURE AND EXAMPLE APPLICATION

The SnowPylot Python library implements an object-oriented design that mirrors the CAAML data structure. Additional detail is available in the README and other package documentation on GitHub and PyPI.

For the purposes of demonstration, we exported CAAML.xml files from SnowPilot for the 2020-2024 water years, and used the SnowPylot library to gather summary statistics about the snow pits in the dataset. A jupyter notebook called “demo\_2020-2024.ipynb” contains the full analysis and is in the “demos” folder of the SnowPylot repository on GitHub.

SnowPit

The SnowPit class serves as the container for all snow pit observations and consists of four main components: core\_info, snowProfile, stabilityTests and whumpData. Each CAAML.xml file in the dataset is parsed into a snowPit object.

CoreInfo

The CoreInfo class encapsulates Core attributes of a snow pit, including the pit's identification details (pit\_id, pit\_name), temporal information (date), and descriptive elements (comment, caaml\_version). There are additional nested objects within the CoreInfo class including a User object that holds information about the person who created the profile, a Location object that contains geographical and environmental details of where the snow pit was dug, and a WeatherConditions object that describes the atmospheric conditions at the time of the profile creation.

Some summary statistics about the 2020-2024 dataset using CoreInfo attributes:

*Summary of some CoreInfo attributes for the 2020-2024 dataset*

|  |  |
| --- | --- |
| Total pits | 31,170 |
| Unique Users | 3,854 |
| Pits submitted by Avalanche Professionals | 19,891 |
| Pits submitted by non-professionals | 11,279 |

*Summary of some Location Info for the 2020-2024 dataset*

|  |  |
| --- | --- |
| Unique Countries | 30 |
| Pits Near and Avalanche | 945 |
| Pit on Avalanche Crown | 480 |
| Pits on Avalanche Flank | 240 |

SnowProfile

The SnowProfile class contains basic profile information like measurement direction, profile depth, and height of the snowpack. The class manages three main types of observations: layers (stored in a list of Layer objects), temperature observations (stored in a list of TempObs objects), and density observations (stored in a list of DensityObs objects). It also includes a surface condition (SurfaceCondition) and indicates the layer of concern if specified.

Some summary statistics about the 2020-2024 dataset using SnowProfile attributes:

*Summary of SnowProfile Data Available for 2020-2024 dataset*

|  |  |
| --- | --- |
| Total Pits | 31,170 |
| Layers | 232,718 |
| Layers with Primary Grain Form | 190,805 |
| Layers with Primary Grain Size | 109,034 |
| Pits with Temperature Profile | 14,449 |
| Temperature Observations | 134,518 |
| Pits with Density Profile | 891 |
| Density Observations | 7,731 |
| Pits with Foot Penetration Observation | 19,811 |
| Pits with Ski Penetration Observation | 11,061 |

StabilityTests

The StabilityTests class is a dataclass that serves as a container for different types of snow stability tests. It contains four lists as attributes, each representing a different type of stability test (Extended Column Test, Compression Test, Rutschblock Test and Propagation Saw Test):

*Available Data for All Test Types*

|  |  |
| --- | --- |
| Total Pits | 31,170 |
| Pits with Stability Test Results | 28,151 |
| Percentage of Pits with Stability Test Results | 90.31% |

*Test Results by Test Type*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ECT | CT | PST | RBlock |
| Total Pits with Test Results | 21,092 | 18,022 | 3,102 | 121 |
| Total Test results | 29,247 | 32,150 | 3,719 | 134 |
| Percentage of Pits with Test Results | 67.67 | 57.82% | 9.95% | 0.39% |

WhumpfData

WhumpfData is a custom, SnowPilot-specific set of fields that was created for a study and allows certain users to capture information about whumpf observations. At the time of this writing, there are 306 records in the SnowPilot database that contain WhumpfData.

##### EXAMPLE ANALYSES

## The python code for both of these analyses is available in the Jupyter Notebook “demo\_2020-2024.ipynb” in the “demos” folder of the SnowPylot repository on GitHub (https://github.com/connellymk/snowpylot/blob/main/demos/demo\_2020-2024.ipynb).

## Example 1: *How often is a Q1 Facture in a Compression Test associated with Propagation on the same layer in an Extended Column Test*

Steps in Analysis:

1. Parse all files in dataset into SnowPit objects
2. Find SnowPits that have ECT and ECT results
3. For every combination of ECT and CT result, find pits where failure occurs on the same layer and ECT has propagation
4. Record fracture character for these pits
5. Map fracture character to common schema
6. Analyze and plot results

Results:

Our analysis shows an inconsistency in the SnowPilot data where two different schemas for fracture character were used. To account for this, we mapped the fracture character to consistent shear quality groups, as shown in the treemap in Figure 1.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure . Tree Map of CT Shear Quality Results in Pits with ECTP Results where failure occurred on the same layer.

After performing this mapping we found the following results:

|  |  |
| --- | --- |
| Tests with CT failure on the same layer as ECTP | 3,442 |
| CT results with Q1 failure | 2,617 |
| **Percentage of Results with Q1 failure** | **76.03%** |

Example 2: What is the relationship between Hand Hardness and Primary Grain Form of Snowpit layers?

Steps in Analysis:

1. Parse all files in dataset into SnowPit objects
2. Use SnowPit objects to create a list of layer objects for all layers in all pits
3. Record hand hardness and primary grain form for each layer
4. Map hand hardness to hand hardness group
5. Plot results

Results:

SnowPilot allows the user to enter hand hardness on a scale that include “ – “and “+” for each group, for example: F-, F, and F+. We found that the +/- designations were used less and chose to map all sub designations to the main hand hardness value.

Figure 2. shows a clear correlation between the Hand Hardness and Primary Grain Form, which matches our understanding and observations in the field.

A screenshot of a graph

AI-generated content may be incorrect.

Figure . Heatmap of Hand Hardness Group and Primary Grain Form

##### DISCUSSION AND FUTURE WORK

Potential Applications

Limitations and Potential Improvements

Future Development Plans

Invitation for Community Involvement

##### REFERENCES

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Paper presented Western Snow Conference 2025

1 Mary Connelly, Montana State University, Bozeman, MT, 59718, connellymarykate@gmail.co