

# IRIS-HEP Fellowship Project Proposal

---

## Features Extension, Inclusion & Rectification for Boost-Histogram

Project Area: Analysis Systems

### Jay Gohil

School of Technology PDEU (Fellow)

### Henry Schreiner and Hans Dembinski

Princeton University (Mentor) and TU Dortmund (Co-Mentor)

---

## Introduction:

[Boost-histogram](#) is a Python package that provides Python bindings for [Boost.Histogram](#) (a C++ library), and has proven to be transformative in producing Histograms using Python. It provides plotting tools, shortcuts and new ideas that include histogram operations, axes manipulation, variable storage options, indexing, subclassing, advanced visual object representations, UHI implementation, various accumulators, inbuilt analyses and comparison, complete histogram transformations and in-notebook representation while being compatible with [NumPy](#). Furthermore, boost-histogram was recently even included in the 0.20 release of [Pyodide](#) (a WebAssembly port of CPython).

Considering the powerful and extensive features provided by the tool, there are various improvements, feature additions and function changes that would result in making the tool more advanced while being viable and extensively usable. Therefore, the aim of the project is to dive deeper into the current [issues](#) and improvements to solve them and cater to the needs of the community while making the tool better and more robust.

## Project Proposal:

The major aim of the project proposal is to extend the development of boost-histogram tool through new features additions, core function changes, edge-case covers, bug fixes and more exhaustive documentation. This extension of the tool will result in making the tool better by improving performance, widening usability, easier readability (of documentation), covering of varied applications, and meticulous fine-tuning of functions. Therefore, here is the course of action (from 1st June to 30th September 2022; 2 months part time and full time each) broken down into timeframes for each proposed improvement that would be undertaken:

### I. Visualize or portray difference between histograms | Week 1 - 1.5

- A custom diff printer can be added for PyTest to visualize/see actual difference between different histograms. This would tremendously help in debugging while testing.
- Solves issue [#157](#).

## II. Allow slicing on axes | Week 1.5 - 3

- Slicing a histogram is common and works as intended.
- Thus, for consistency with Universal Histogram Interface, it should work on axes too.
- Solves issue [#281](#).

## III. Disallow non-integer argument input for weight fill on integer storage | Week 3 - 4.5

- The *weight* argument currently accepts both integers and doubles (yet converts them to integer) for integer storages.
- Thus, it shouldn't accept anything other than integers, (& throw an error otherwise).
- Solves issue [#289](#).

## IV. Add scalar support for histogram fill function | Week 4.5 - 6

- Currently, the fill operation accepts vectors as inputs in histogram.
- However, adding scalar value for sample argument throws a ValueError. Thus, add functionality to accept scalar values (and hence extend functionality).
- Solves issue [#646](#).

## V. Add operators for weighted histograms / accumulators | Week 7 - 8

- Add support for multiplication, division and subtraction for weighted histograms.
- Allow operations such as  $h1/h2$  or  $h1*h2$ , and thus extend functionality.
- Solves issue [#693](#) and partially [#352](#).

## VI. Add support for scaling and addition for mean views | Week 9 - 10

- The universal function's ndarray operation currently only supports scaling and addition for weight views.
- To extend functionality, the support for Mean or WeightedMean views must be added.
- Solves issue [#560](#).

## VII. Add collector and efficiency accumulator | Week 11 - 13

- Add an accumulator (collector) that holds all samples which ended up in a certain bin.
- Allows for operations such as unbinned fitting in each bin, median computation, kernel density estimate computation; and return (possibly) an awkward array.
- Add an efficiency accumulator in similar capacity as well.
- These will be added by the parallel (C++ based) Boost.Histogram IRIS-HEP fellowship project; only a Python wrapping is needed here.
- Solves issue [#353](#).

## VIII. Add in-doc notebook and extend the documentation | Week 14 - 16

- Add jupyter notebook inside [documentation](#) that portrays working of functions (and allows for running similar code post tutorial code).
- Extend the documentation in terms of guide sections, existing sections' content, exhaustive function argument information and code examples.

Solving these issues and pertaining problems will assuredly result in better performance, extended usability and wider applicability of the tool. Moreover, in case of a scenario where the aforementioned projects are completed before the deadline, the following issues will be taken into consideration as well:

- Support for addable weight view ([#276](#)).
- Vectorization of `_at`, `_at_set`, `__getitem__` and `__setitem__` ([#149](#)).
- Support for Full UHI ([#208](#)).