Histogramming in Python

August 2025 to September 2026

PyHEP.dev · 9 September 2026

Release timeline

AUG 2025	FEB 2026	APR 2026	JUN 2026	SEP 2026
UHI 1.0	bh 1.7 · hist 2.10	bh 1.7.2	browser 0.9	browser 0.10
serialization spec	MultiCell, generics	wheel refresh	fuzzy branch finder	image plots, heatma

Work prompted last year by PyHEP.dev, then my Scikit-build grant is over now, so have a bit more time to focus on this. Also AI tooling advancements have helped work on some long standing issues / requests / features.

UHI

01

Since last year at PyHEP.dev

SAID IN JULY 2025	WHERE IT STANDS
Release in about two weeks	UHI 1.0 out on 20 August 2025
Nightly wheels needed to try it	On PyPI; nightlies still published
UHI tests nightly-only	Shipped in 1.0
Zarr backend, by Peter	Still in progress

What UHI covers

Indexing
Access, setting, slices, projections, rebinning.

Test framework released

Indexing+
Loc shortcuts, name-based syntax, rebin shortcut.

hist only

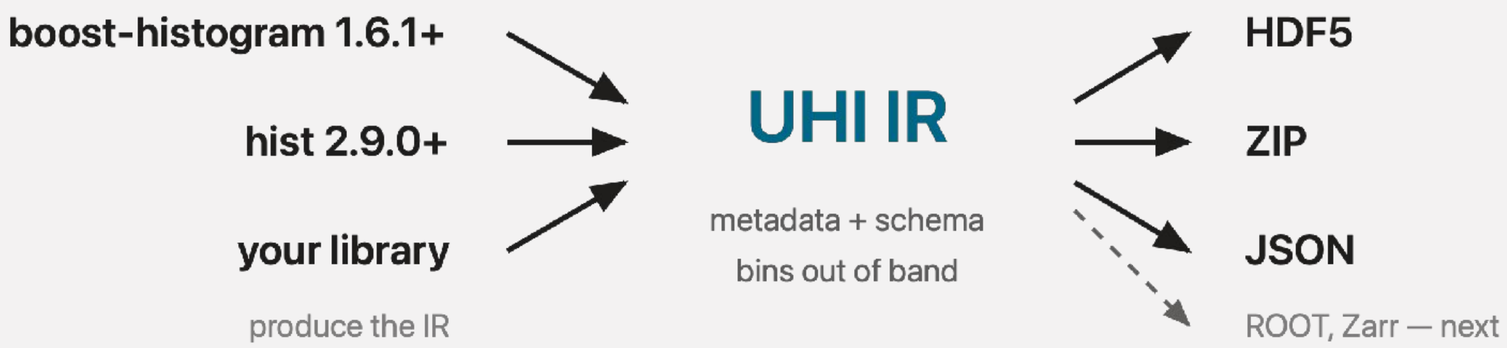
Plotting
The protocol for the basics — values, variances, kind.

Unchanged

Serialization
An IR plus readers and writers for three containers.

Released in 1.0

Producers, IR and containers



UHI 1.0 defines the representation; the container backends live in one place. A library can write its own reader instead — in any language.

A round trip

```
import json, uhi.io.json

data = json.dumps(hists, default=uhi.io.json.default)
hists = json.loads(data, object_hook=uhi.io.json.object_hook)

uhi.io.hdf5.write(h5_file.create_group(name), hist)
uhi.io.zip.read(zip_file, name)
```

Named histograms in, named histograms out. HDF5 keeps the native structure; ZIP holds JSON metadata beside binary values.

Who is on the spec

Test framework: PyROOT and babyyoda.
Sharing through UHI: uproot, histoprint, mplhep, PyROOT.
Producing the IR: boost-histogram 1.6.1+ and hist 2.9.0+.
PyROOT supports indexing now too

boost-histogram and hist

02

The MultiCell storage

```
from hist import Hist

h = Hist.new.Regular(10, 0, 20).MultiCell(3)
h.fill(1, weight=[[1, 2, 3]])

h.sum()              # [1.0, 2.0, 3.0]
h[0]                 # [1.0, 2.0, 3.0]
h.values().shape     # (3, 10)
```

A preset number of weights per bin — boost-histogram 1.7, hist 2.10.

Sparse storage: in progress

PR for double showing it works, waiting on upstream fixes in C++ Boost.Histogram.

Histograms are now generic

```
h = Hist.new.Integer(0, 10).Double()    # Hist[Double]

g = Hist.new.Boolean().Weight()         # Hist[Weight]
s = g.sum()                             # WeightedSum
print(f"{s.value=} {s.variance=}")      # no assert needed
```

The storage is part of the type, so returns narrow on their own.

Under strict checkers, bare Hist becomes Hist[Any].

Platform and build support

Python 3.14, 3.14t, 3.15 and 3.15t in, experimental 3.13t out.
New wheels: iOS, Windows ARM, GraalPy.
pybind11 3.0; Python 3.10 the floor since 1.7.
Diagnostics test inside the wheel — check a build without the test suite.

Smaller changes

Metadata

raw_metadata for the C++ side; __dict__ settable on a histogram, as on axes.

Indexing

Set a range with a scalar, a callable or a histogram — which also fixed *=.

Slicing in dicts

Use np.s_ for slices inside dict-style indexing.

Fewer copies

A clang-tidy sweep cut copies and refcount churn across the bindings.

hist keeps step

2.9, with serialization

The first version that reads and writes the UHI representation — inherited from the core, not reimplemented.

2.10, with the core's features

MultiCell through QuickConstruct, generic histograms, Python 3.10 the floor.

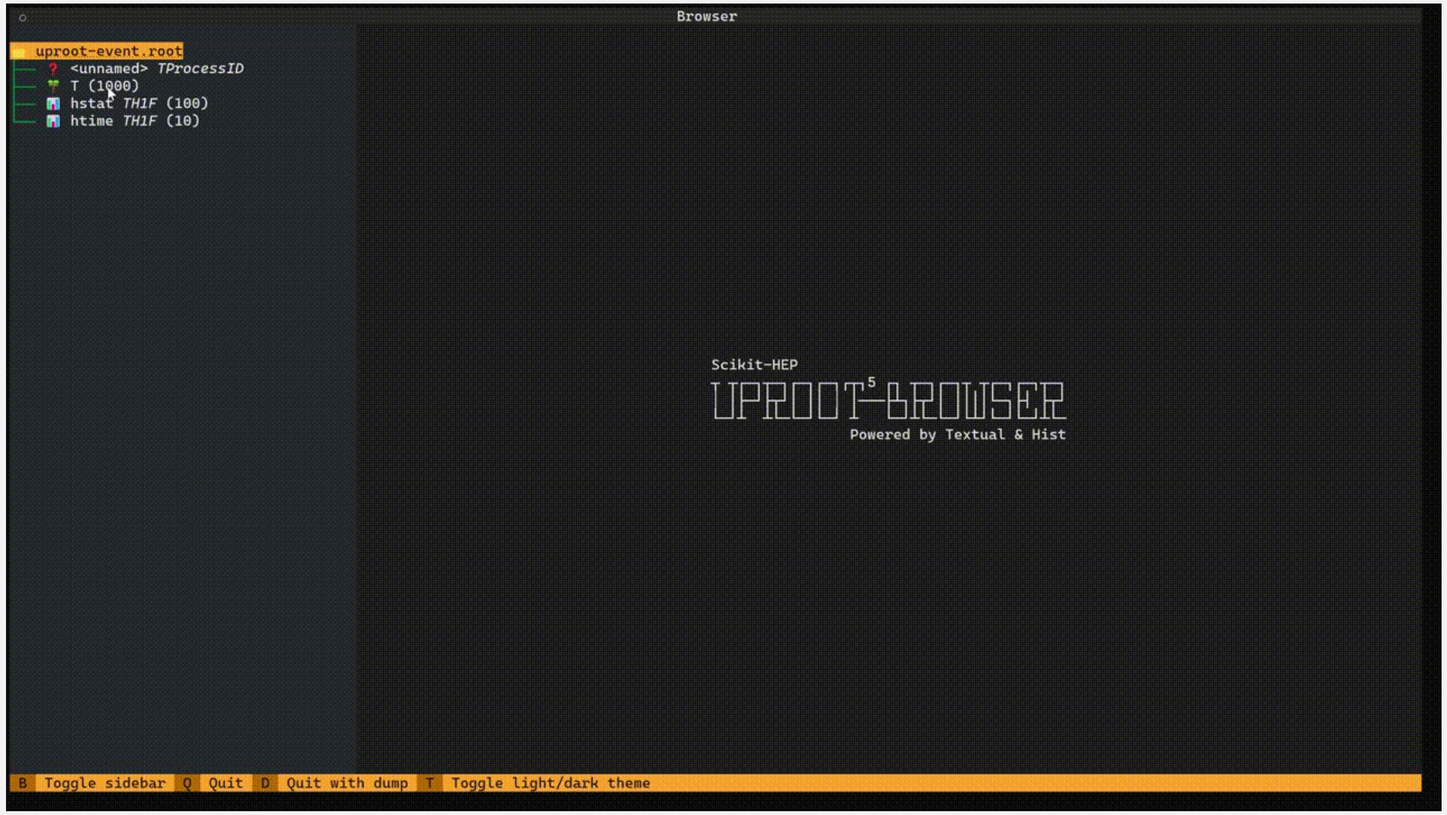
2.11, with ChunkedHist

hist.chunked.ChunkedHist holds a dense array per category key, so growing categories never reallocate and unfilled combinations cost nothing.

uproot-browser

LIVE DEMO

03



Before — uproot-browser 0.5, a year ago. The demo is what it looks like now.

PyHEP.dev based development cycle

Eleven months quiet, then four releases in three.
fzf-style fuzzy finder for jumping to a branch.
Tabs with settings, info, and entry box
Faster, lots of fixes and solved issues with Fable.

The command surface

```
uvx uproot-browser[testdata] --testdata uproot-Event.root

uproot-browser tree   --testdata uproot-Event.root
uproot-browser plot   --testdata uproot-Event.root:hstat

uproot-browser browse FILE --image     # new in 0.10
uproot-browser plot   FILE:T/fTracks.fPz --image
```

Breaking in 0.10: --iterm becomes --image , through textual-image; the extra is now [image].

Inside the TUI

Plots

2D histograms as heatmaps; image plots sized to the view and cached across re-renders.

Navigation

Vim-style count prefixes and scrolloff; unplottable entries grayed out.

plotext

0.9.1 capped it below 6; 0.10 has a compat layer for 5.2.8+ and 6+.

Internals

One PlotItem protocol behind both backends. Now tested on Python 3.15.

Related tools

04

cibuildwheel

Through 3.x: Android, a uv frontend, --clean-cache , GHA summaries.

4.0/4.1: Python 3.15(t), abi3audit, devlewheel, Pyodide 3.14/3.15

Dropped 3.8, 3.13t and GraalPy 3.11.

pyemscripten wheels now accepted on PyPI.

Distinct wheels it can build — six platforms, three interpreters, 3.9 through 3.15t.

scikit-build-core

1.0 at SciPy 2026: 140+ open issues down to 20, 90+ bugs fixed, new site and docs.

A CLI with project init , an env table, force-include, PEP 803 ABI3t, reproducible wheels, and much better editable installs.

dynamic-metadata is now a system other backends can use — PEP 808, new built-in providers.

PyTorch moved to scikit-build-core 1.0.

pybind11 and nanobind

pybind11 3.0.x

Four patch releases of fixes, plus a quiet performance pass: vectorcall dispatch, nanobind's two-level type lookup, no heap allocation for calls with ≤6 arguments. boost-histogram 1.6 builds against it.

pybind11 3.1.0 — August 2026

Python 3.9 is the floor, MSVC 2017 dropped, 3.15 in CI. PEP 484 numeric conversions (`float` takes `int`), reusable sub-interpreter thread states, `py::mod_gil_used()`.

nanobind 2.9–2.13

Object construction 1.42–3.2× faster, dispatch 1.2–1.5×, ndarray exchange up to 2.4×. An AI-assisted security audit fixed race conditions and memory corruption; stub generation in CMake and Array API support landed.

Questions

Slides made in Claude Design

Was it faster?

No.