Histogramming & more

Henry Schreiner

September 12, 2019
About me

Henry Schreiner
Computational Physicist / Research Software Engineer
Princeton University

- University of Texas at Austin
  - MayaMuon experiment - cosmic ray measurement, Belize pyramids
- University of Cincinnati
  - LHCb - Fast fitting in charm
- Author of iscinumpy.gitlab.io
- Some work on iris-hep.org
- Redesigned scikit-hep.org
- Desired hobby: Computer graphics/modeling
- Actual hobby: Raising a 3-year old
AS: boost-histogram (Python)

/scikit-hep/boost-histogram

- 0-dependency build (C++14 only)
- State-of-the-art PyBind11
- 280+ unit tests run on Azure on Linux, macOS, and Windows
- Binary wheels on Azure for all major platforms
  - Machinery already in use for iMinuit, helping other Scikit-HEP packages
- Read the docs (in progress)
Boost.Histogram C++14 (Part of Boost since 1.70)

- Multidimensional templated header-only histogram library: [boostorg/histogram](https://github.com/boostorg/histogram)
- Designed by Hans Dembinski, inspired by ROOT, GSL, and histbook

### Histogram
- Axes
- Storage

### Axes types
- Regular, Circular
- Variable
- Integer
- Category

![Diagram of histogram with axes, storage, and accumulator options]
Design demo

Note: Some details may change

```python
from pyforest import *
import boost.histogram as bh

# New histogram composition
hist = bh.histogram(
    bh.axis.regular(20, -1, 1),
    bh.axis.variable([-1, -0.3, 0, 0.2, 0.8, 1])
)

# Fill with arrays
hist.fill(
    np.random.normal(size=10_000),
    np.random.normal(size=10_000)
)

# Convert bins to numpy array
# Can keep or remove flow bins
nparr = hist.view(flow=True)

# UHI: Unified Histogram Indexing
pr = hist[:bh.rebin(2), ...]

# Scale
hist2 = hist * 2

# Sum
hist2.sum()
```

Henry Schreiner
Histogramming & more
September 12, 2019
22 axis types

- **regular**
  - uoflow, uflow, oflow, nflow, growth
- **regular_+**
  - log, sqrt, pow
- **circular**
- **integer**
  - uoflow, uflow, oflow, nflow, growth
- **variable**
  - uoflow, uflow, oflow, nflow
- **category**
  - int or str, growth

```python
bh.axis.regular(10,0,1)
```

```python
bh.axis.circular(8,0,2*np.pi)
```

```python
bh.axis.variable([0,.3,.5,1])
```

```python
bh.axis.integer(0,5)
```

```python
bh.axis.category([2,5,8,3,7])
```
hist is the ‘wrapper’ piece that does plotting and interacts with the rest of the ecosystem.

**Plans**

- Easy plotting adaptors (mpl-hep)
- Serialization formats (ROOT, HDF5)
- Auto-multithreading
- Statistical functions (Like TEfficiency)
- Multihistograms (HistBook)
- Interaction with fitters (ZFit, GooFit, etc)
- Bayesian Blocks algorithm from SciKit-HEP
- Command line histograms for stream of numbers
Future of histograms in Python

Core histogramming libraries

Universal adaptor

Front ends (plotting, etc)

boost-histogram

ROOT

Aghast

hist

mpl-hep

physt

others
• Python library for PDG particle information
• Intuitive interface, LaTeX/HTML names, and much more
• New: ID conversion tools, GeantID, more

• Python library for decay descriptions
• Reads DEC files, also AmpGen utilities
The fastest, most universal way to get ROOT on macOS or Linux, for Python and more!

### Packages
- ROOT 6.18/04
- Pythia8
- Uproot
- HEPMC2
- HEPMC3
- xrootd

And many more HEP tools!

- Full support for environments
  - no thisroot.sh
- Easy way to have Python 2 & 3
- Supports compiling, C++17, JupyROOT, and more
- Fully reproducible from environment.yml
- Runs on mybinder, docker, and more
IA: PV-Finder

- Custom 3D hits to 1D kernel
- Based on toy LHCb detector

- Developing CNN network to find PVs
- Achieving 93%+ accuracy
Very successful Bootcamp at LBNL!

Each day started with a general lesson, followed by a matching ATLAS lesson.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Name</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version Control</td>
<td>Intro to Git</td>
<td>Software Carpentries</td>
</tr>
<tr>
<td>Build Systems</td>
<td>Modern CMake</td>
<td>IRIS-HEP</td>
</tr>
<tr>
<td>Continuous Integration &amp;</td>
<td>CI/CD and Python Testing</td>
<td>Software Carpentries</td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td>IRIS-HEP</td>
</tr>
<tr>
<td>Containers</td>
<td>Intro to Docker</td>
<td>IRIS-HEP</td>
</tr>
</tbody>
</table>

Much of this material is reusable!