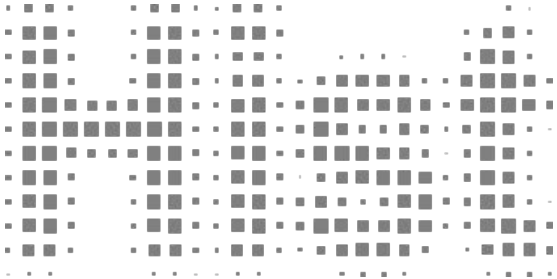


Advancing the Hist library

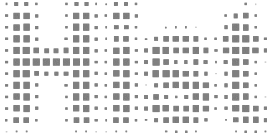
IRIS-HEP Fellowship

Aman Goel, University of Delhi

Mentor: Henry Schreiner, Princeton University

About Me

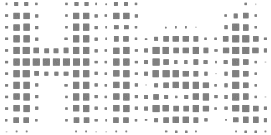
- Undergraduate Student of Engineering at *Cluster Innovation Centre, University of Delhi*
- Majoring in *Information Technology & Mathematical Innovations*
- IRIS-HEP Fellow, Summer 2021
- Github: amangoel185
- E-mail: aman.goel185@gmail.com



Background

Hist is a powerful Histogramming tool for analysis based on *boost-histogram* (the *Python* binding of the Histogram library in Boost). It is a friendly analysis-focused project that uses *boost-histogram* as a backend to do the work, but provides plotting tools, shortcuts, and new ideas.

The latest release of *Hist* is **Version 2.5.1** which includes a variety of features and improvements.



Stacked Histograms

A histogram stack holds multiple 1-D histograms into a stack, whose axes are required to match. The most common way to create one is with a categorical axes.

```
ax = hist.axis.Regular(25, -5, 5, flow=False, name="x")
cax = hist.axis.StrCategory(["signal", "upper", "lower"], name="c")
full_hist = Hist(ax, cax)

full_hist.fill(x=np.random.normal(size=600), c="signal")
full_hist.fill(x=2 * np.random.normal(size=500) + 2, c="upper")
full_hist.fill(x=2 * np.random.normal(size=500) - 2, c="lower")

 s = full_hist.stack("c")
```

Stacked Histograms

We can also build it using shortcuts such as:

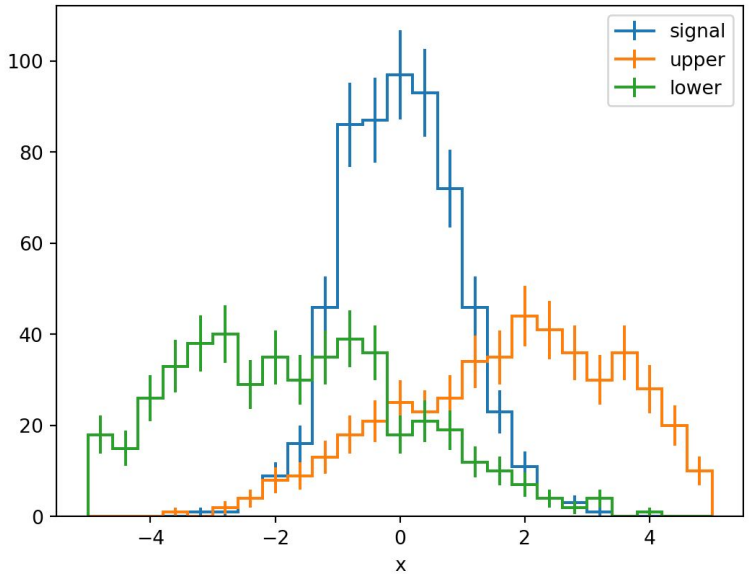
```
hist.Stack(h1, h2, h3)
```

or

```
hist.Stack.from_iter([h1, h2, h3])
```

or

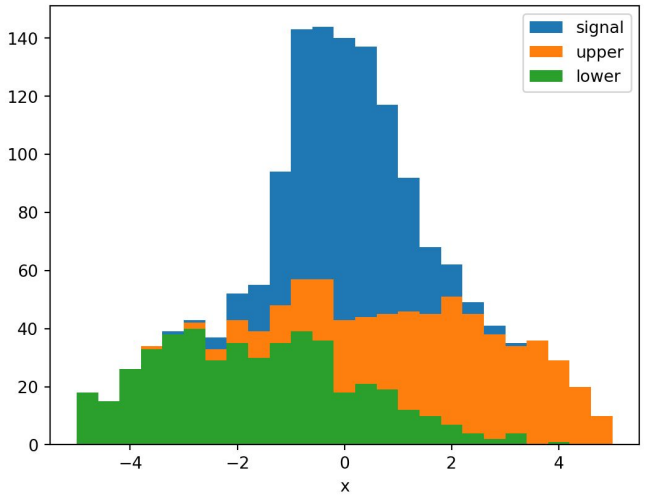
```
hist.Stack.from_dict({"signal": h1,
"lower": h2, "upper": h3})
```



Stacked Histograms

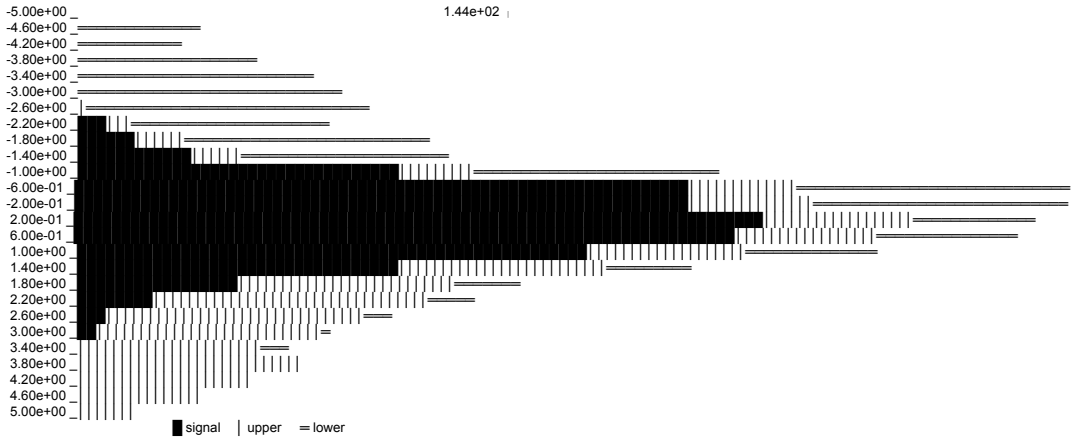
We can also obtain the “stacked” style of plot via *mplhep*. Although we need to use slicing to obtain the correct order since it gets reversed.

```
s[::-1].plot(stack=True, histtype="fill")
plt.legend()
plt.show()
```



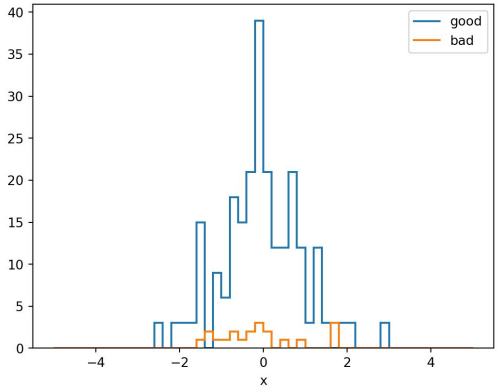
Stacked Histograms

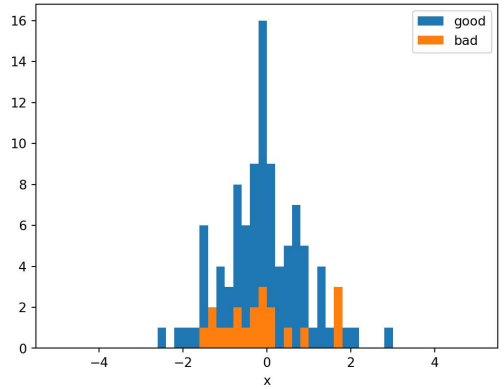
We can also print the stacked histograms to the console via `histoprint` which can be accessed via `.show()` which gives us the following:

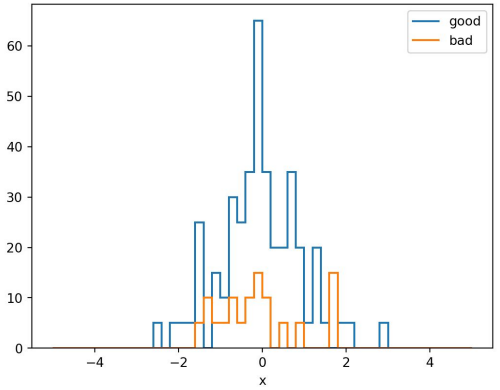


Stacked Histograms

- Stacks can also save names of the histograms that are present in it.
- They can be scaled too or an item in the stack can be scaled inplace.
- Various math operations can be performed on stacks.



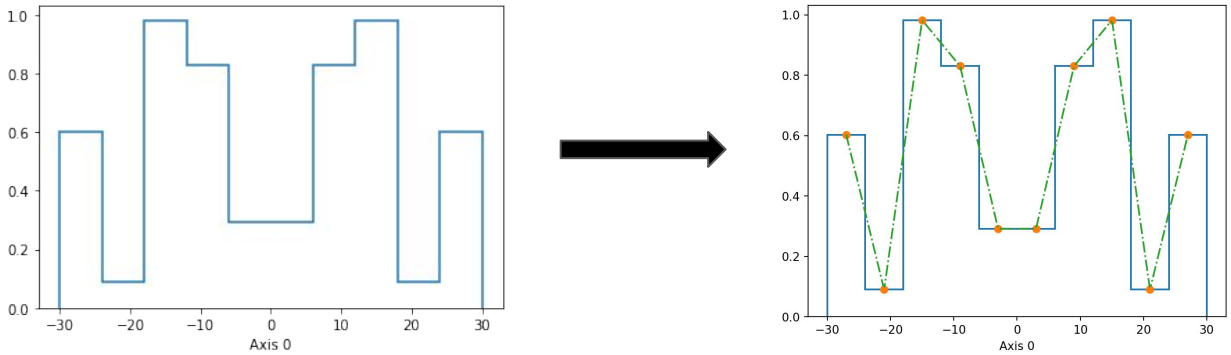




Interpolation

We can perform interpolation in *Hist* using *SciPy* in the following way:

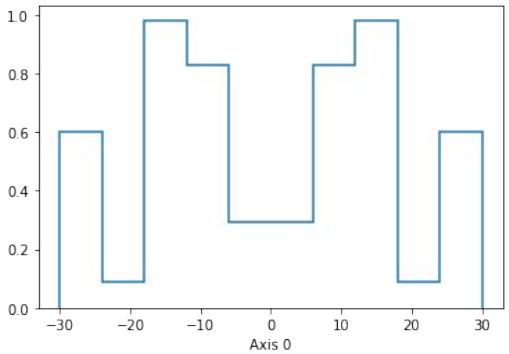
```
x = np.linspace(-27, 27, num=250, endpoint=True)

linear_interp = interpolate.interp1d(h.axes[0].centers, h.values(), kind="linear")
```

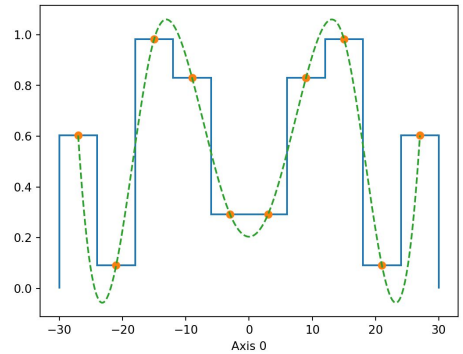


Interpolation

We can perform interpolation in *Hist* using *SciPy* in the following way:

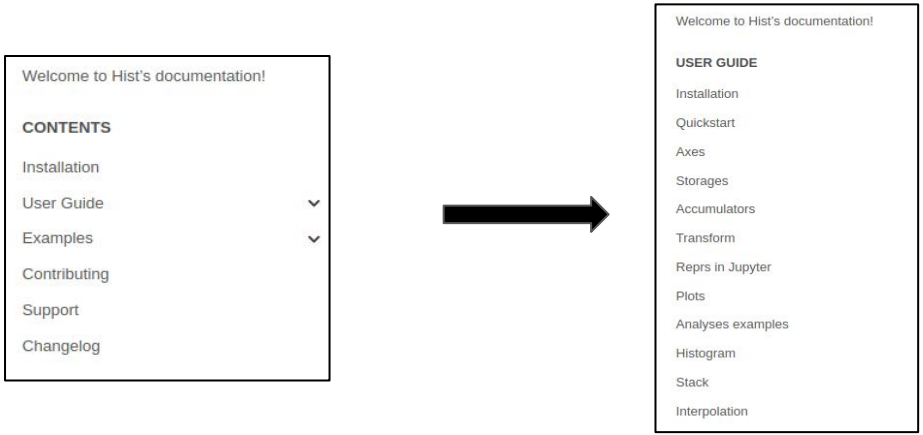
```
x = np.linspace(-27, 27, num=250, endpoint=True)

cubic_interp = interpolate.interp1d(h.axes[0].centers, h.values(), kind="cubic")
```

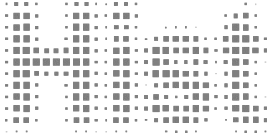




Documentation

Since *Hist* builds upon the features of *boost-histogram*, the documentation has been updated to include a similar structure to that of *boost-histogram*. It uses the same conventions and features and adds on to them where the only major difference is the dependency on `hist` rather than `bh`.



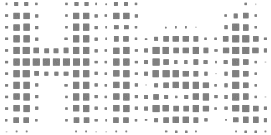


Other Features

- QuickConstruct in *Hist* can directly pass `name=` and `label=` as arguments while defining the histogram.

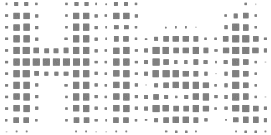
```
data = np.array([18, 27, 9, 36, 1, 8, 45, 7, 2, 108])
h = Hist.new.Regular(10, 0, 90, name="x").Double(name="h", label="y", data = data)
```

- There is now a `hist.new` alias for `hist.Hist.new`.



Other Features

- Dropped *Python* 3.6 support.
- Uses *boost-histogram* 1.2.x series, includes all features and fixes, and *Python* 3.10 support.
- No longer require `scipy` or `iminuit` unless actually needed.
- Improve and clarify treatment of confidence intervals in intervals submodule.
- Use *NumPy* 1.21 for static typing.
- Support running tests without plotting requirements.

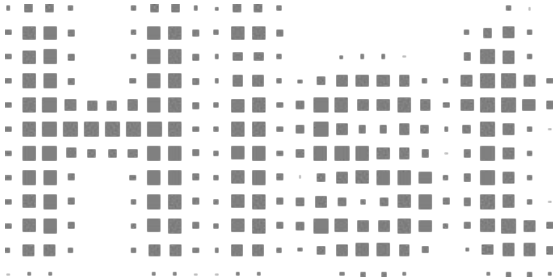


Acknowledgement

- I would like to extend my sincere gratitude to Henry Schreiner who guided me throughout the project and helped me understand new concepts while being kind enough to answer all my queries and doubts. It has been a wonderful experience to have him as my mentor.

- I would also like to thank Shuo Liu for helping me whenever I got stuck and using his experience with the project to make sure I am able to understand everything.

- And finally, I am really grateful the whole IRIS-HEP community for being warm, welcoming and supportive throughout the fellowship, right from the application process.



Thank You!

Aman Goel, University of Delhi