

IMPORTS

```
import ROOT

# UHI tags from the ROOT uhi module
from ROOT.uhi import (
    loc,      # look up by value
    rebin,    # rebin action
    sum,      # sum action
    underflow, # underflow bin
    overflow, # overflow bin
)
```

CREATE AND FILL

1D

```
h1 = ROOT.TH1D("h", "title;x;counts", 100, -5.0, 5.0)

h1.Fill(1.5)
h1.Fill(2.0, 0.5) # with weight

# fill from numpy array directly
h1.Fill(np.ones(100))
```

2D

```
h2 = ROOT.TH2D("h2", "title;x;y", 50, -5.0, 5.0, 50, -5.0, 5.0)
h2.Fill(1.0, 2.0)
```

PROPERTIES

.values(flow, writable) all values as a numpy array:

- flow=True includes under/over flow bins
- writable=True returns a mutable array

.variances estimated variance of the accumulated values

.kind "COUNT" or "MEAN"

TAGS

sum action: sum an axis

rebin(n) action: rebin an axis by factor n

loc(x) bin of data coordinate x

under/overflow under/overflow bin

INDEXING

Bin access

<code>h[0]</code>	content of bin 0
<code>h2[2, 5]</code>	content of bin (2, 5)
<code>h[-1]</code>	content of last bin
<code>h[loc(1.5)]</code>	content of bin containing x=1.5
<code>h[underflow]</code>	content of underflow bin
<code>h2[overflow, overflow]</code>	content of overflow bin
<code>len(h)</code>	number of bins

Bin setting

Single bin

```
h[3] = 12.0 # set bin 3
h[loc(1.5)] = 5.0 # set bin at x=1.5
h[underflow] = 0.0 # clear underflow
```

Entire histogram

```
# array size = nbins → does NOT set flow bins
h[:] = np.ones(100)
h[...] = np.ones(100) # equivalent

# array size = nbins+2 → sets flow bins too
h2[:] = np.ones(102) # incl. under/overflow

# 2D: shape (nx, ny) or (nx+2, ny+2)
h2[:, :] = np.zeros((50, 50))
h2[...] = np.zeros((50, 50)) # equivalent
```

SERIALIZATION

Serialize / deserialize ROOT histograms to exchange data with any UHI-compatible library:

```
import json, uhi.io.json, hist

ob = json.dumps(h_root, default=uhi.io.json.default)

h_hist = hist.Hist(json.loads(ob),
                   object_hook=uhi.io.json.object_hook)
```

SLICING

<code>h2[2:8]</code>	bins 2-7 → TH1
<code>h[loc(-1):loc(1)]</code>	$x \in [-1,1] \rightarrow$ TH1
<code>h[underflow:overflow]</code>	all incl. flow
<code>h[:, :sum]</code>	sum all → float
<code>h[2:8:sum]</code>	sum bins 2-7 → float
<code>h[:, :rebin(2)]</code>	rebin $\times 2 \rightarrow$ TH1
<code>h[loc(-1):2:rebin(5)]</code>	slice + rebin

3	4	9	8
1	2	1	2
7	5	0	3

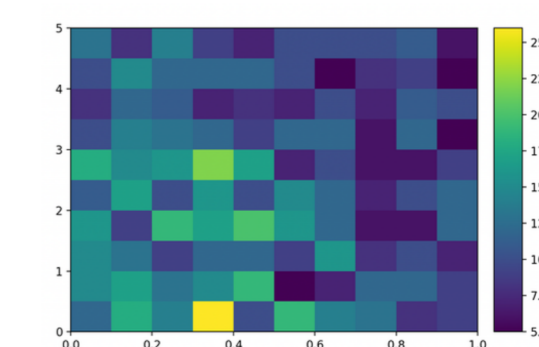
```
h2[:, 2:3]
# all x
# y bin 2
```

3	4	9	8
1	2	1	2
7	5	0	3

```
h2[:, 2, ...]
# x bins 0 and 1
# all y
```

PLOTTING

Plot ROOT histograms directly with ROOT graphics or your favorite Python plotting library



With ROOT graphics

```
h.Draw() # default style
h.Draw("HIST") # histogram bars
h2.Draw("COLZ") # 2D colour map
```

With external libraries

```
import matplotlib.pyplot as plt
import mplhep as hep

hep.histplot(h) # default style
hep.style.use("LHCb2") # choose style sheet
hep.hist2dplot(h2) # 2D
```