

**Name : Shardul Khade**

**Task - Exercise2**

**Project: Histogram Pythonization**

**Improve Python Interface for Histograms**

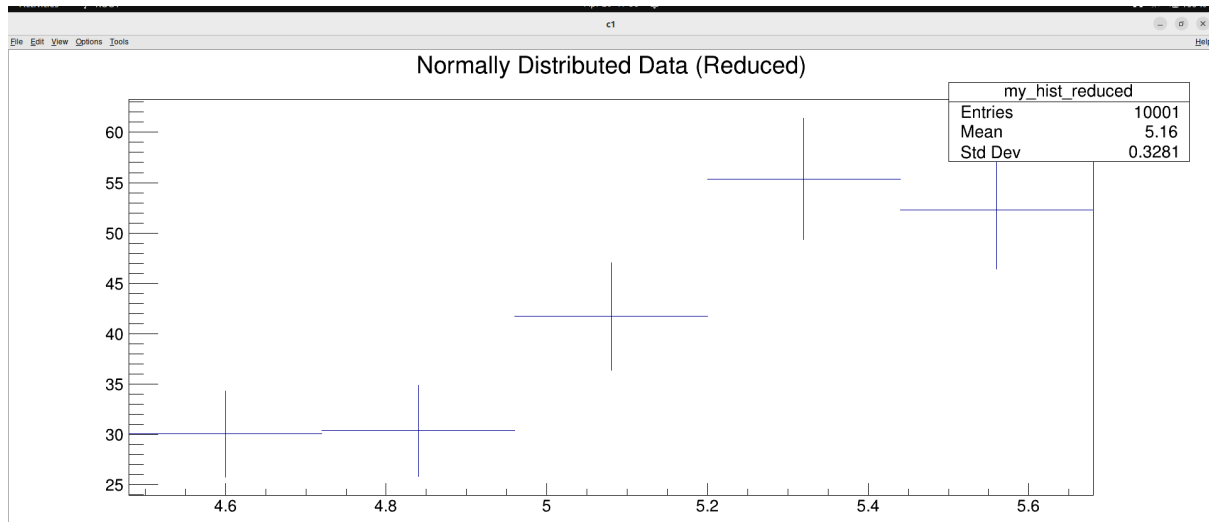
- 1) Create a new histogram in a reduced range, given the histogram h, and implement the code creating a reduced (zoomed) histogram. The corresponding UHI function is  $h[\text{ifirst}:\text{ilast}]$ , where ifirst and ilast are arbitrary values provided by the user.

**Corresponding code file to be found here :** <root\tutorials\tasks\Exercise2\test3.py>

Approach:

I am using the hist generated in **Exercise 1** by the '.Clone' method and reducing (zooming) it to the range(ifirst,ilast).

Output: ifirst= 3 ilast= 7 ; Total bins =  $7-3+1$  (since inclusive) = 5;



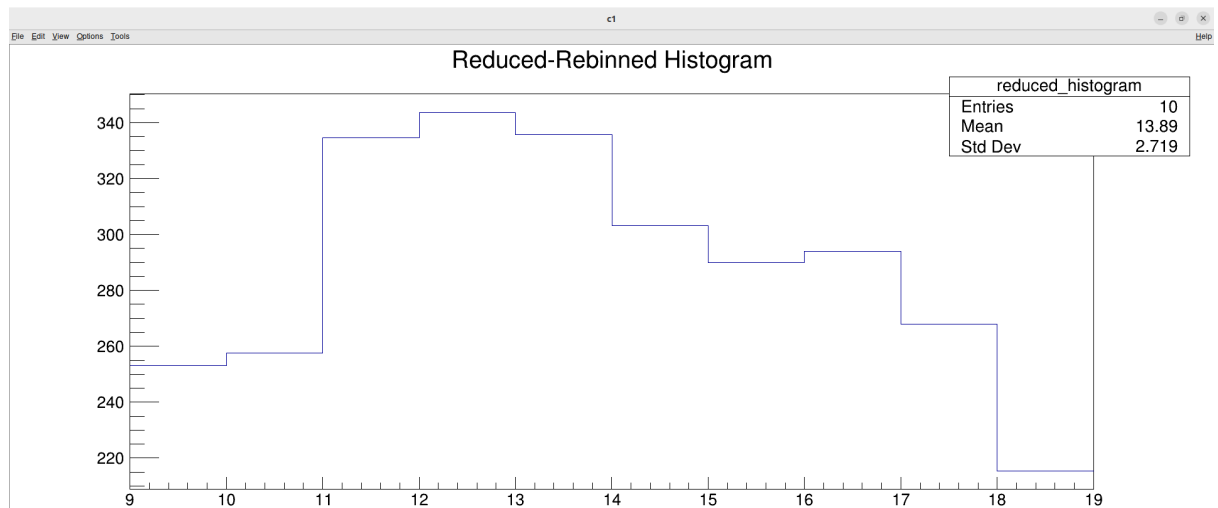
## reduce\_his() function:

```
1  from ROOT import *
2  import numpy as np
3
4  def reduce_his(h, ifirst, ilast):
5
6
7      # Check for valid range
8      print(h.GetNbinsX()) #Range should have less bins than total bins.
9
10
11     # Create the new histogram with the reduced range
12     h_reduced = h.Clone(h.GetName() + "_reduced")
13     h_reduced.SetBinContent(h_reduced.GetNbinsX(), h.GetBinContent(ifirst, ilast))
14     h_reduced.SetBinError(h_reduced.GetNbinsX(), h.GetBinError(ifirst, ilast))
15
16     # Set axis ranges and labels for the reduced histogram
17     h_reduced.GetAxis().SetRange(ifirst, ilast)
18     h_reduced.GetAxis().SetTitle(h.GetAxis().GetTitle())
19     h_reduced.SetTitle(h.GetTitle() + " (Reduced)")
20
21     return h_reduced
22
```

```
4  def reduce_his(h, ifirst, ilast):
20
21     return h_reduced
22
23
24     num_data = 10000
25     data = np.random.normal(loc=10, scale=3, size=num_data)
26     weights = np.random.rand(num_data)
27
28     # Create ROOT histogram (using efficient TH1D constructor)
29     hist = TH1F("my_hist", "Normally Distributed Data", 50, 4, 16) # 50 bins from 4 to 16
30     for i in range(num_data):
31         hist.Fill(data[i], weights[i])
32     h_reduced = reduce_his(hist, 3, 7)
33     cl=TCanvas()
34     h_reduced.Draw()
35     input()
36
```

- 2) Same operation as before, but adding a rebin of 2, i.e. merge the bins by 2 to reduce the total number of bins by 2: `h[first:ilast:rebin(2)]`

Corresponding code file to be found here : <root\tutorials\tasks\Exercise2\test4.py>



The output histogram is reduced(zoomed) in the range of 11 to 30 inclusive. Bins are refactored by 2. I am logging the no. of bins in the new hist after Rebin. Initially there were 50 bins but after rebinning, it became 25 which was expected.

```
shardul@shardul:~/Downloads$ python3 test4.py
25
```

### reduce\_rebin\_Hist() : Code

```
5
6 def reduce_rebin_Hist(h, ifirst, ilast, rebin_factor):
7     # Clone the original histogram
8     h_clone = h.Clone("h_clone")
9
10    # Rebin the cloned histogram
11    h_clone.Rebin(rebin_factor)
12    reduced_histogram = TH1F("reduced_histogram", "Reduced-Rebinned Histogram", ilast - ifirst + 1, ifirst, ilast + 1)
13    print(h_clone.GetNbinsX())
14    # Filling the new histogram with the summed bin contents from the original histogram
15    for i in range(ifirst, ilast + 1):
16        reduced_histogram.SetBinContent(i - ifirst + 1, h_clone.GetBinContent(i))
17    return reduced_histogram
18
19 num_data = 10000
20 data = np.random.normal(loc=10, scale=3, size=num_data)
21 weights = np.random.rand(num_data)
22
23 # Create ROOT histogram (using efficient TH1D constructor)
24 hist = TH1F("my_hist", "Normally Distributed Data", 50, 4, 16) # 50 bins from 4 to 16
25 for i in range(num_data):
```

```

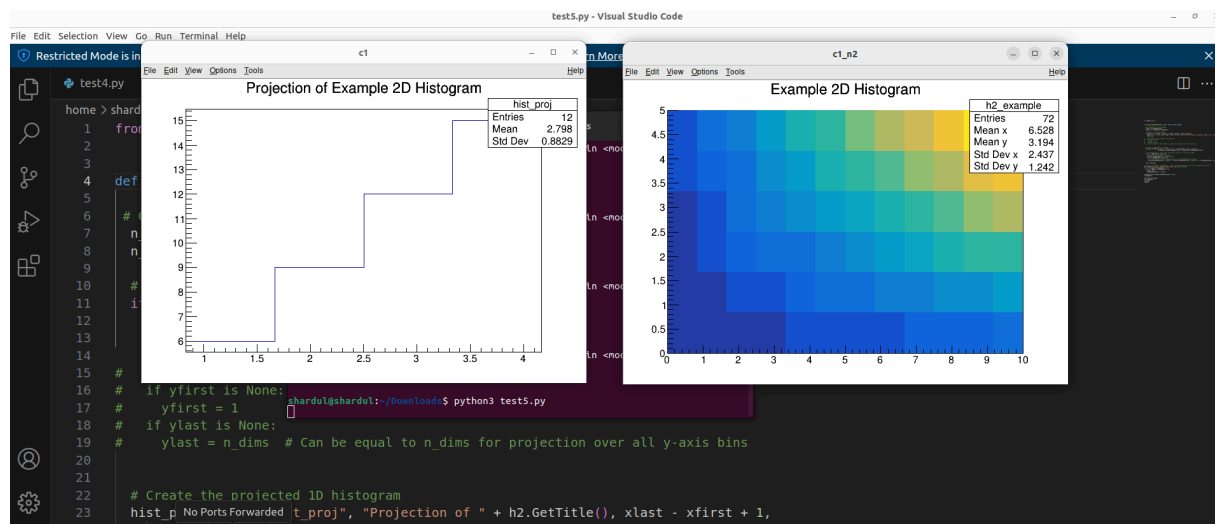
21 weights = np.random.rand(num_data)
22
23 # Create ROOT histogram (using efficient TH1D constructor)
24 hist = TH1F("my_hist", "Normally Distributed Data", 50, 4, 16) # 50 bins from 4 to 16
25 for i in range(num_data):
26     hist.Fill(data[i], weights[i])
27 h_reduced = reduce_rebin_Hist(hist, 9, 18, 2)
28 c1 = TCanvas()
29 h_reduced.Draw()
30 input()
31

```

- 3) Given a multidimensional histogram, h2 implement a function performing a projection in one-dimensional histogram, doing the following: a projection in the first axis (x) between xfirst and xlast by summing the bins in the y axis between yfirst and ylast:  
`h[xfirst:xlast, yfirst:ylast:sum]`

Corresponding code file to be found here : [root\tutorials\tasks\Exercise2\test5.py](#)

**Output: 2D Hist created and displayed along with its 1D projection.**



## Project\_histogram\_1d() ; Code

```
test5.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

test4.py test5.py test3.py
home > shardul > Downloads > test5.py
1 from ROOT import *
2
3
4 def project_histogram_1d(h2, xfirst, xlast, yfirst, ylast):
5
6     # Get dimensions of the histogram
7     n_dims = h2.GetDimension()
8     n_bins_x = h2.GetAxis().GetNbins()
9
10    # Check for valid x-axis range
11    if xfirst < 1 or xfirst > n_bins_x or xlast < xfirst or xlast > n_bins_x:
12        print("Error: Invalid x-axis range. Please provide values within the original histogram's range (1 to", n_bins_x, ").")
13        return None
14
15
16
17    # Create the projected 1D histogram
18    hist_proj = TH1F("hist proj", "Projection of " + h2.GetTitle(), xlast - xfirst + 1,
19                    h2.GetAxis().GetBinLowEdge(xfirst), h2.GetAxis().GetBinUpEdge(xlast))
20
21    # Loop through bins in the x-axis range and sum counts over the y-axis range
22    for i in range(xfirst, xlast + 1):
23        # Sum the content of the corresponding bin in the multidimensional histogram
24        hist_proj.SetBinContent(i - xfirst + 1, h2.GetBinContent(i, yfirst, ylast))
25
26    return hist_proj
```

```
test5.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

test4.py test5.py test3.py
home > shardul > Downloads > test5.py
4 def project_histogram_1d(h2, xfirst, xlast, yfirst, ylast):
5
6     # Get dimensions of the histogram
7     n_dims = h2.GetDimension()
8     n_bins_x = h2.GetAxis().GetNbins()
9
10    # Check for valid x-axis range
11    if xfirst < 1 or xfirst > n_bins_x or xlast < xfirst or xlast > n_bins_x:
12        print("Error: Invalid x-axis range. Please provide values within the original histogram's range (1 to", n_bins_x, ").")
13        return None
14
15
16
17    # Create the projected 1D histogram
18    hist_proj = TH1F("hist proj", "Projection of " + h2.GetTitle(), xlast - xfirst + 1,
19                    h2.GetAxis().GetBinLowEdge(xfirst), h2.GetAxis().GetBinUpEdge(xlast))
20
21    # Loop through bins in the x-axis range and sum counts over the y-axis range
22    for i in range(xfirst, xlast + 1):
23        # Initialize the projected bin content
24        hist_proj.SetBinContent(i - xfirst + 1, 0)
25        for j in range(yfirst, ylast + 1):
26            # Sum the content of the corresponding bin in the multidimensional histogram
27            hist_proj.SetBinContent(i - xfirst + 1, hist_proj.GetBinContent(i - xfirst + 1) + h2.GetBinContent(i, j))
28
29    return hist_proj
30
31 # Create a 2D histogram. Following h2: X axis has 12bins ranging 0 to 10; Y axis has 6bins from 0 to 5.
32 h2 = TH2F("h2 example", "Example 2D Histogram", 12, 0, 10, 6, 0, 5)
33 for i in range(1, h2.GetNbinsX() + 1):
34     for j in range(1, h2.GetNbinsY() + 1):
35         value = i * j
36         h2.SetBinContent(i, j, value)
37
38 projected_hist = project_histogram_1d(h2, 2, 5, 1, 2)
39 c1 = TCanvas()
40
Ln 14, Col 1 Spaces: 4 UTF-8 LF Python
```