notebook-example

August 20, 2025

1 Introduction to ROOT & Documentation in Jupyter

1.1 Weekly Report - 1st week, by Gabriel

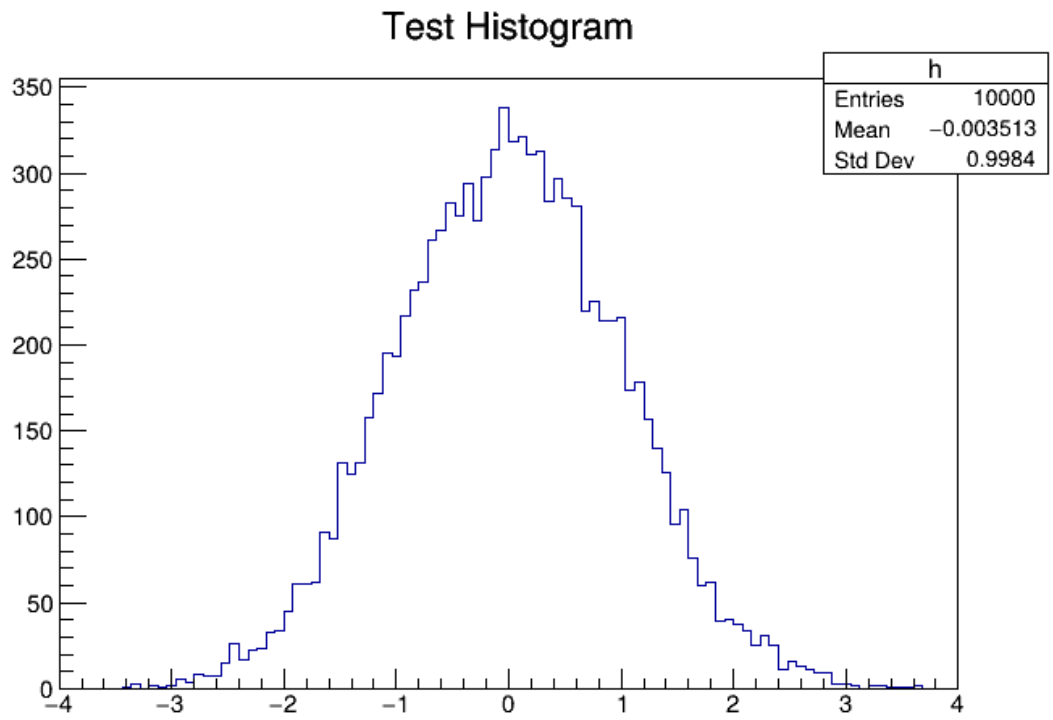
This week the goal was to make Jupyter (Python friendly environment) read and plot a ROOT macros (C++ based). We will use this data to fit a gaussian and visualize it. 1. Let's start by seting up ROOT:

[1]:
```
# 1. ROOT SETUP

import ROOT
from ROOT import TCanvas, TH1F, TF1, gStyle
```

2. Load your macro and plot observables

[4]:
```
# run your macro
ROOT.gROOT.ProcessLine('.x example.C')

# grab the current canvas
c = ROOT.gPad.GetCanvas()
c.SaveAs("plot.png")

# display in Jupyter
from IPython.display import Image
Image("plot.png")
```

[4]:




```
Warning in <TROOT::Append>: Replacing existing TH1: h (Potential memory leak).
Info in <TCanvas::Print>: png file plot.png has been created
```

3. Load your macro and call it's function (title of file).

[6]:
```
h1 = ROOT.gDirectory.Get("h")
print("Entries:", h1.GetEntries())
print("Mean:", h1.GetMean())
```

```
Entries: 10000.0
Mean: -0.003512854234856602
```

4. Define Gaussian (explicitly or call it's function) perform fit and draw.

[7]:
```
# Define a Gaussian function over the histogram range
gaus = ROOT.TF1("gaus", "gaus", -4, 4)
# Fit histogram with Gaussian
h1.Fit(gaus, "s")  # "R" = restrict to given range
```

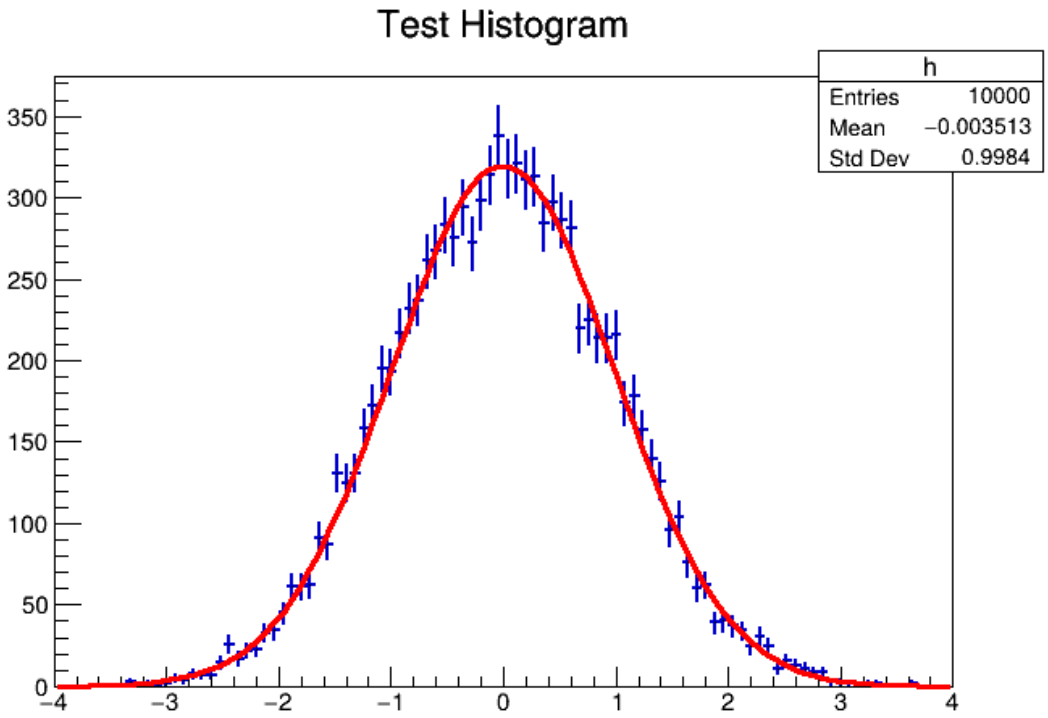
[7]:
```
<cppyy.gbl.TFitResultPtr object at 0x55b4faef5d20>
```

```
****************************************
Minimizer is Minuit2 / Migrad
Chi2                      =        71.15
NDf                       =           84
```

```
Edm                       =  3.13383e-07
NCalls                    =          55
Constant                  =      319.587   +/-   3.92696
Mean                      =  -0.00481863   +/-   0.0100074
Sigma                     =     0.992265   +/-   0.00709299      (limited)
```

[12]:
```
# Draw
c = ROOT.TCanvas()

# Draw histogram with a fill color
h1.SetLineColor(ROOT.kBlue+1)
h1.SetLineWidth(2)
h1.Draw("E")    # "E" draws error bars

# Style the Gaussian fit
gaus.SetLineColor(ROOT.kRed)
gaus.SetLineWidth(3)
gaus.Draw("same")
#Draw canvas
#c.Draw()
#Save

# grab the current canvas
c = ROOT.gPad.GetCanvas()
c.SaveAs("plotfit.png")

# display in Jupyter
from IPython.display import Image
Image("plotfit.png")
```

[12]:




```
Info in <TCanvas::Print>: png file plotfit.png has been created
```

2 Gaussian Fit Exercise

We generated random numbers following a Gaussian distribution with mean **0** and sigma **1**. The histogram shows the data, and the red curve is a Gaussian fit.

2.0.1 Questions

1. What is the mean value obtained from the fit?
   - Compare it with the expected mean (0).
2. What is the sigma (standard deviation) obtained from the fit?
   - Compare it with the expected sigma (1).
3. Does the fitted Gaussian describe the data well?
   - Look at the shape of the curve vs. the histogram.
4. How would the result change if we generated more entries?
   - Would the error on the fit parameters increase or decrease?

---

[9]:
```
# Show results and compare (...)
print("Mean:", gaus.GetParameter(1))
```

```
print("Sigma:", gaus.GetParameter(2))
```

```
Mean: -0.004818631265623609
Sigma: 0.9922654365849939
```

2.0.2 Conclusions

- The fit parameters are close to the expected values?
- With more statistics, the uncertainties become smaller.

- ROOT + Jupyter allows us to **document results**, including both the code and the interpretation.