Лекции. Практические занятия

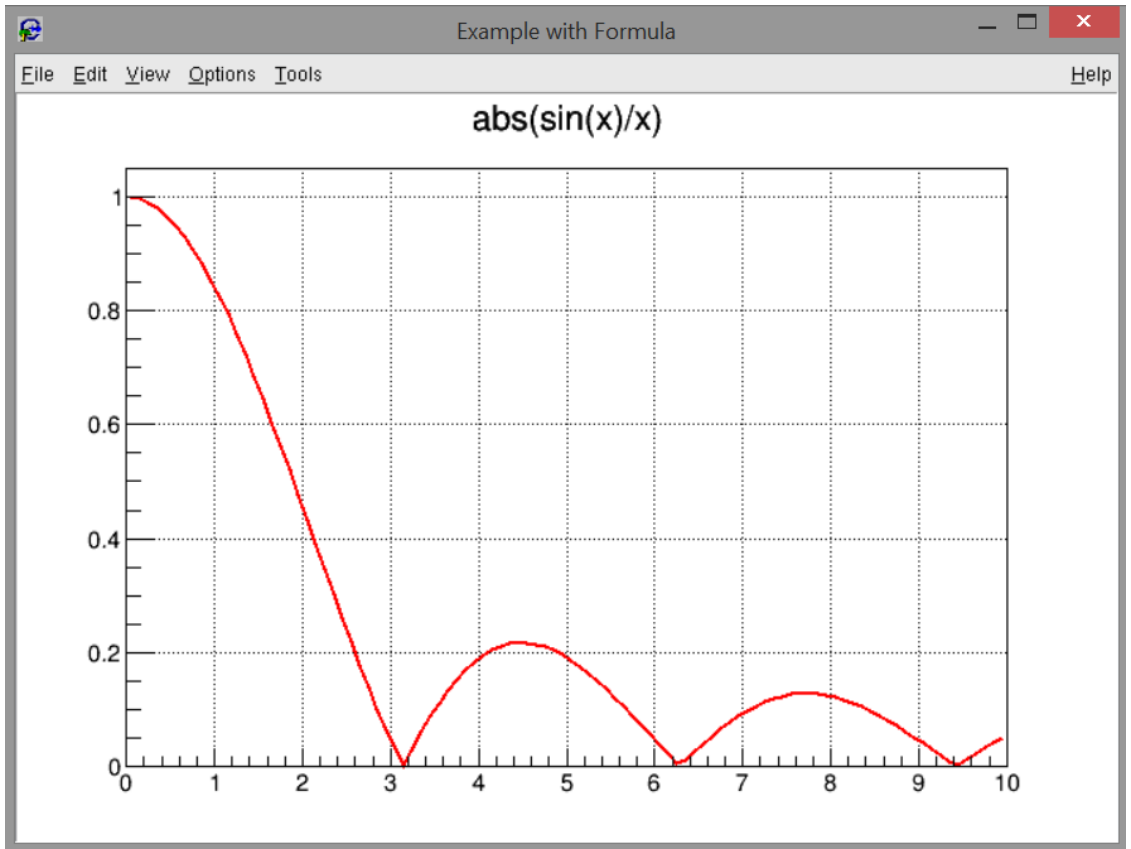
Солдатов Е.Ю.

2024 г.

ПРИМЕР СКРИПТА НА PyROOT

Настройки и инструкции по установке для **pyROOT**: https://root.cern.ch/pyroot
Попробуем предложенный скрипт:

```
from ROOT import gROOT, TCanvas, TF1
gROOT.Reset()
c1 = TCanvas( 'c1', 'Example with Formula', 200, 10, 700, 500 )
fun1 = TF1( 'fun1', 'abs(sin(x)/x)', 0, 10 )
c1.SetGridx()
c1.SetGridy()
fun1.Draw()
c1.Update()
```



PyROOT на lxfarm

$ ssh esoldato@ui03.lxfarm.mephi.ru

```
  GNU nano 2.3.1                                File: pyroot.py

from ROOT import gROOT,  TCanvas,  TF1
gROOT.Reset()
c1 = TCanvas( 'c1', 'Example with Formula', 200, 10, 700, 500 )
fun1 = TF1( 'fun1', 'abs(sin(x)/x)', 0, 10 )
c1.SetGridx()
c1.SetGridy()
fun1.Draw()
c1.Update()
input()
```

$ python3 pyroot.py

* gROOT->Reset() полезен в начале неименованного скрипта, который вы хотите выполнить несколько раз. Поскольку переменные, определенные в неименованном скрипте, находятся в глобальной области видимости, необходимо удалить переменные, созданные предыдущими выполнениями.

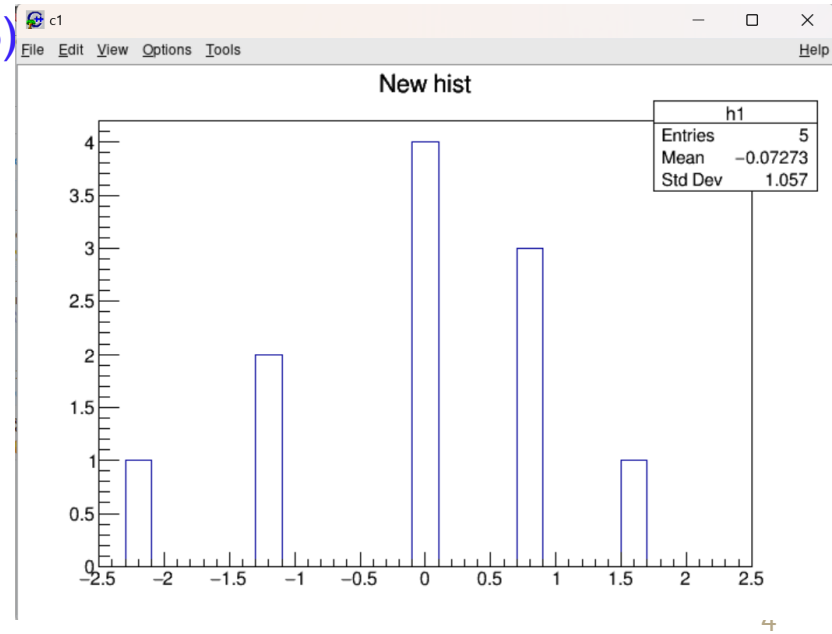
Гистограммы в PyROOT

C++:

```
TH1F *h1 = new TH1F("h1", "New hist",25,-2.5,2.5);
h1->Fill(-2.3);
h1->Fill(1.5);
h1->Fill(0,4);
h1->Fill(-1.2,2);
h1->Fill(0.8,3);
h1->Draw();
```

Python:

```
from ROOT import TH1F
h1 = TH1F("h1", "New hist",25,-2.5,2.5)
h1.Fill(-2.3)
h1.Fill(1.5)
h1.Fill(0,4)
h1.Fill(-1.2,2)
h1.Fill(0.8,3)
h1.Draw("HIST")
input()
```



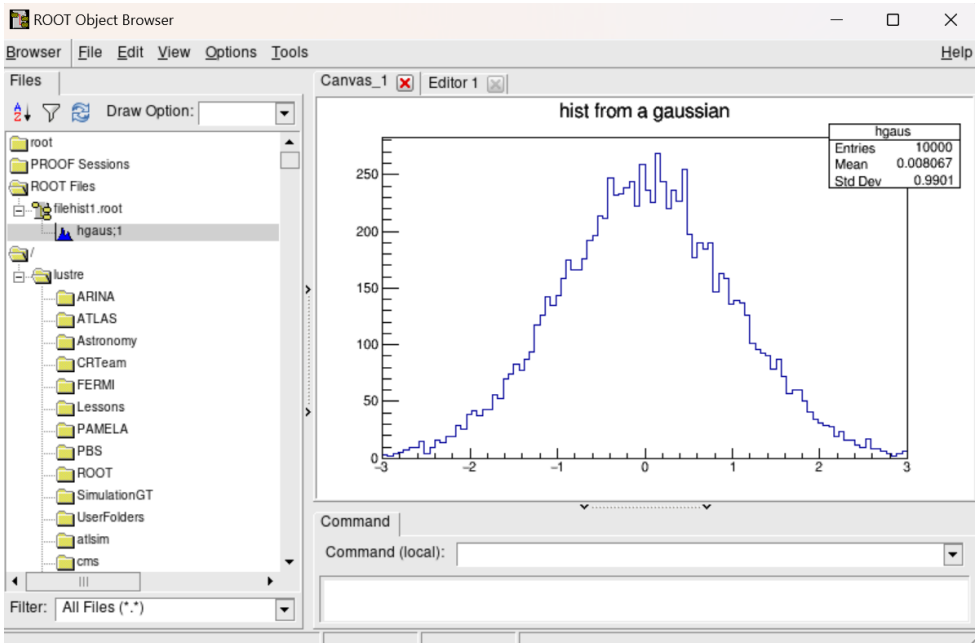
Запись в файл в PyROOT

C++:

```
TFile *file = new TFile("filehist1.root", "new");
TH1F *h1 = new TH1F("hgaus","hist from a gaussian",100,-3,3);
h1->FillRandom("gaus",10000);
h1->Write();
```

Python:

```
from ROOT import TH1F, TFile
file = TFile("filehist1.root", "new")
h1 = TH1F("hgaus","hist from a gaussian",100,-3,3)
h1.FillRandom("gaus",10000)
h1.Write()
```



Работа с деревьями в PyROOT

```
{TFile *f = new TFile("tree1.root", "recreate");
TTree *t1 = new TTree("t1", "Simple Tree");
Float_t px, py, pz;
Int_t ev;
t1->Branch("px", &px, "px/F");
t1->Branch("py", &py, "py/F");
t1->Branch("pz", &pz, "pz/F");
t1->Branch("ev", &ev, "ev/I");
for (Int_t i=0; i<10000; i++) {
gRandom->Rannor(px,py);
pz= px*px+ py*py;
ev= i;
t1->Fill();
}
f->Write();
f->Close();}
```

Проблема: ветвям сопоставляют адреса переменных!

Работа с деревьями в PyROOT

Будем создавать одномерные массивы с плавающей запятой в качестве переменных заполнения, таким образом, число с плавающей запятой:
px = array('f', [0])
массив же служит указателем, который можно передать в ветку.

Создайте ветки и назначьте им переменные заполнения как переменные типа Float:
t1.Branch("px", px, "px/F")

Работать с переменными для заполнения можно так:
ev[0]= i

Работа с деревьями в PyROOT

```
from ROOT import TTree, TFile, gRandom
from array import array
f = TFile("tree1.root", "recreate")
t1 = TTree("t1", "Simple Tree")
px = array('f', [0])
py = array('f', [0])
pz = array('f', [0])
ev = array('i', [0])
t1.Branch("px", px, "px/F")
t1.Branch("py", py, "py/F")
t1.Branch("pz", pz, "pz/F")
t1.Branch("ev", ev, "ev/I")
for i in range(10000):
    gRandom.Rannor(px,py)
    pz[0]= px[0]*px[0]+ py[0]*py[0]
    ev[0]= i
    t1.Fill()
f.Write()
f.Close()
```