Exercícios Análise de Dados - Root

Lucas Brasil de Cerqueira

Outubro de 2024

1 Exercício 0

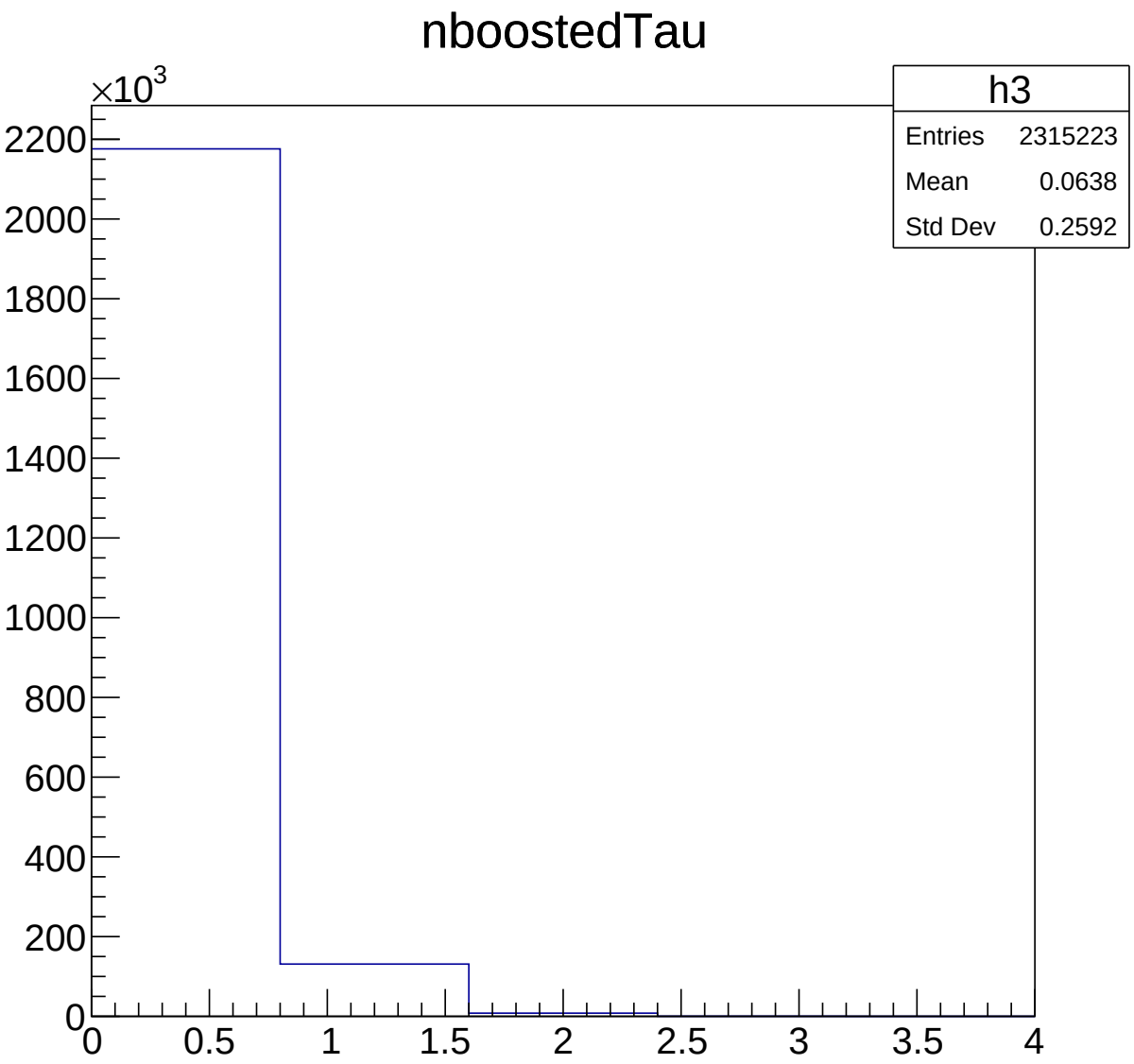
Exercício: Com a ntupla encontrada nas amostras de dados de seu grupo, use o MakeClass e faça o histograma de algumas distribuições.
Dica: https://youtu.be/P9aKV7W5zxg

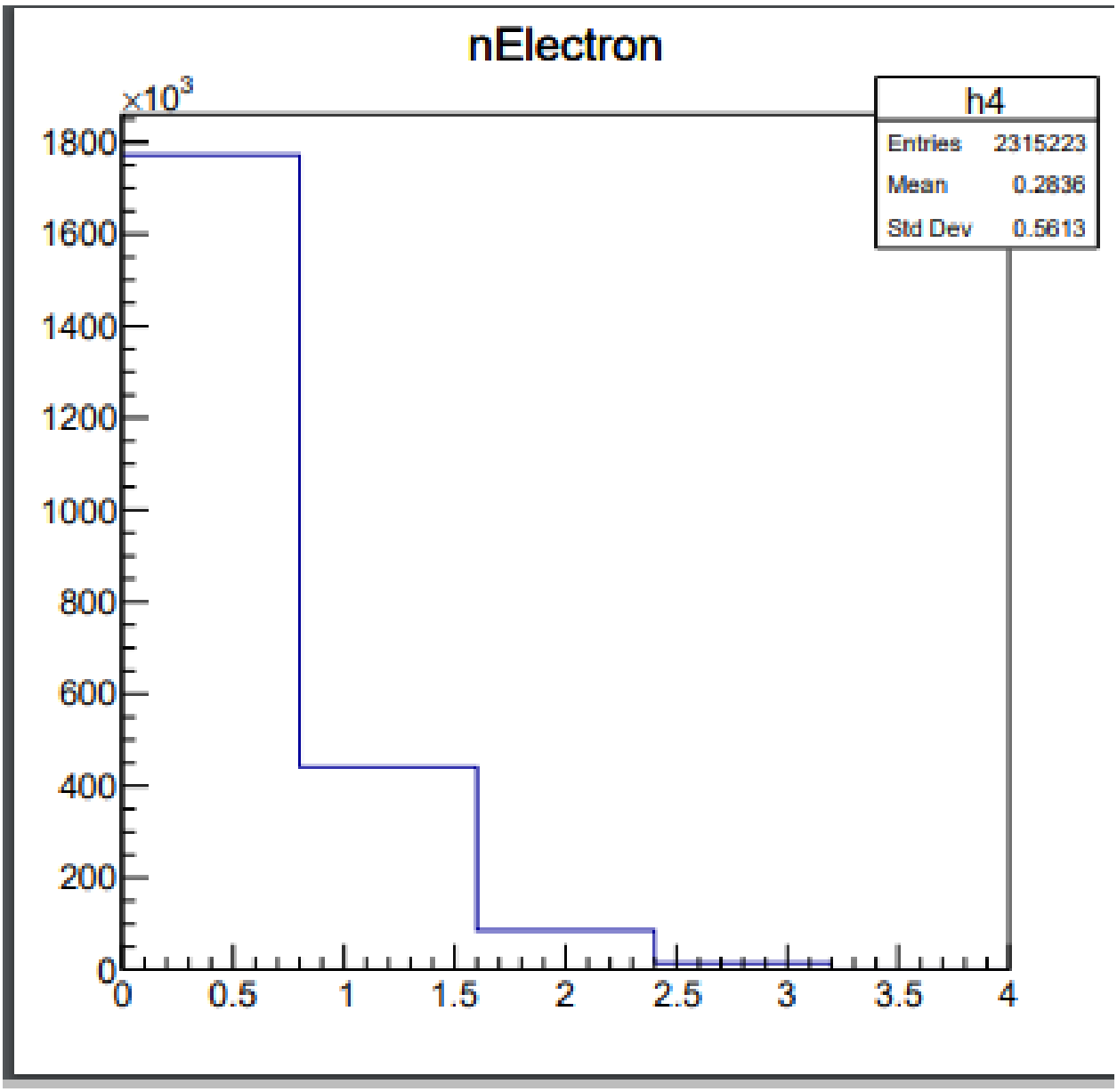
Primeiramente foi usado o MakeClass através do jupyter notebook GRID para acessar o repositório do professor Maurício onde estavam os arquivos de dados. Foi necessário indicar o caminho para o meu repositório da GRID. Assim foram gerados os arquivos que tiveram que sofrer modificações para, além de gerar os histogramas desejados também direcionar os caminhos para o jupyter notebook. Isso foi necessário pois os ntuplas gerados tinham o caminho do diretório do professor. Posteriormente foi feito cp no arquivo de dados e o mesmo foi manipulado através dos

MyTreeClass.C:

```
   if (fChain == 0) return;

   Long64_t nentries = fChain->GetEntriesFast();

   TH1F *h3 = new TH1F("h3","nboostedTau",6 ,-1 ,4);
   TH1F *h4 = new TH1F("h4","nElectron",5 ,0 ,4);

   Long64_t nbytes = 0, nb = 0;
   for (Long64_t jentry=0; jentry<nentries;jentry++) {
      Long64_t ientry = LoadTree(jentry);
      if (ientry < 0) break;
      nb = fChain->GetEntry(jentry);   nbytes += nb;
      // if (Cut(ientry) < 0) continue;

      h3->Fill(nboostedTau);
      h4->Fill(nElectron);
        }



TCanvas *c3 = new TCanvas("c3","c3",10,10,700,700);
TCanvas *c4 = new TCanvas("c4","c4",10,10,700,700);


c3->cd();
h3->Draw("hist");
c3->Print("output.pdf(");
```

E MyTreeClass.h que conterá todo o caminho de fChain e de Branchs para que se possa acessar o arquivo root de forma desejada. Desta forma foi feito o output dos seguintes histogramas:


nboostedTau
h3
Entries 2315223
Mean 0.0638
Std Dev 0.2592
×10^3
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
3
3.5
4



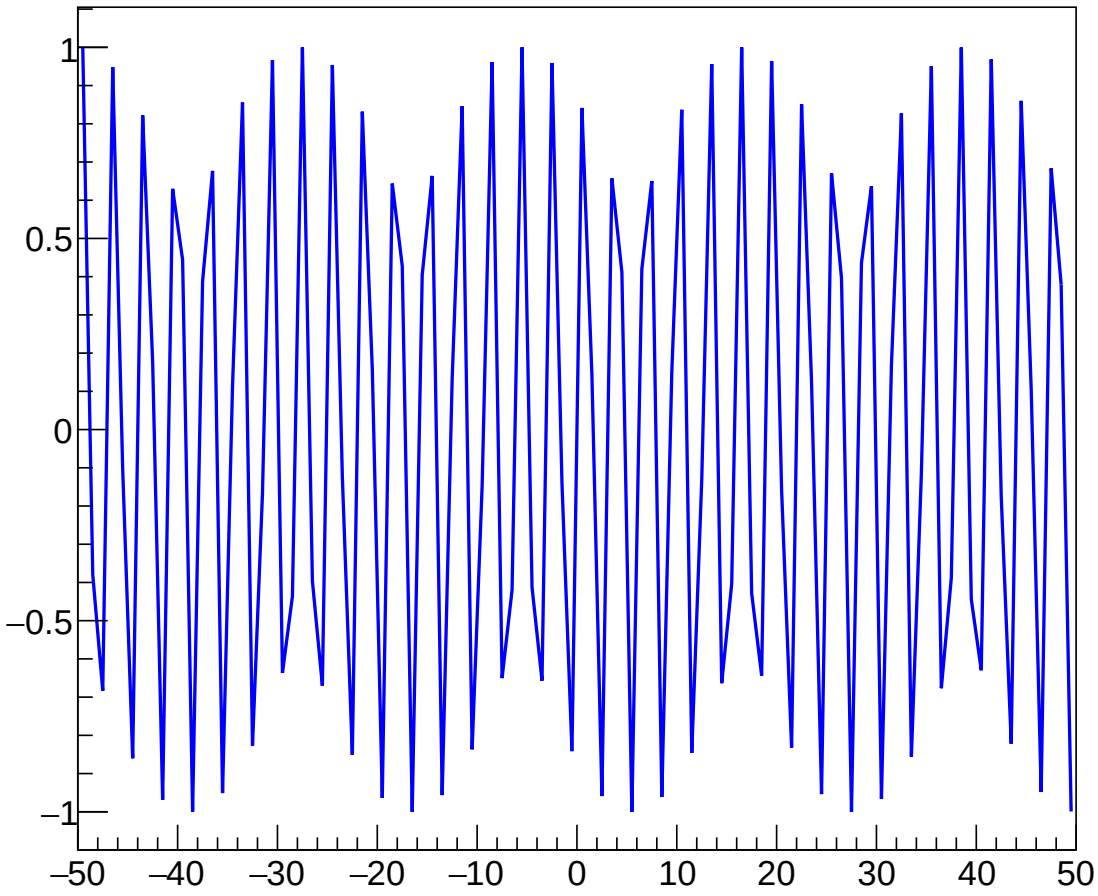
2 Exercício 1

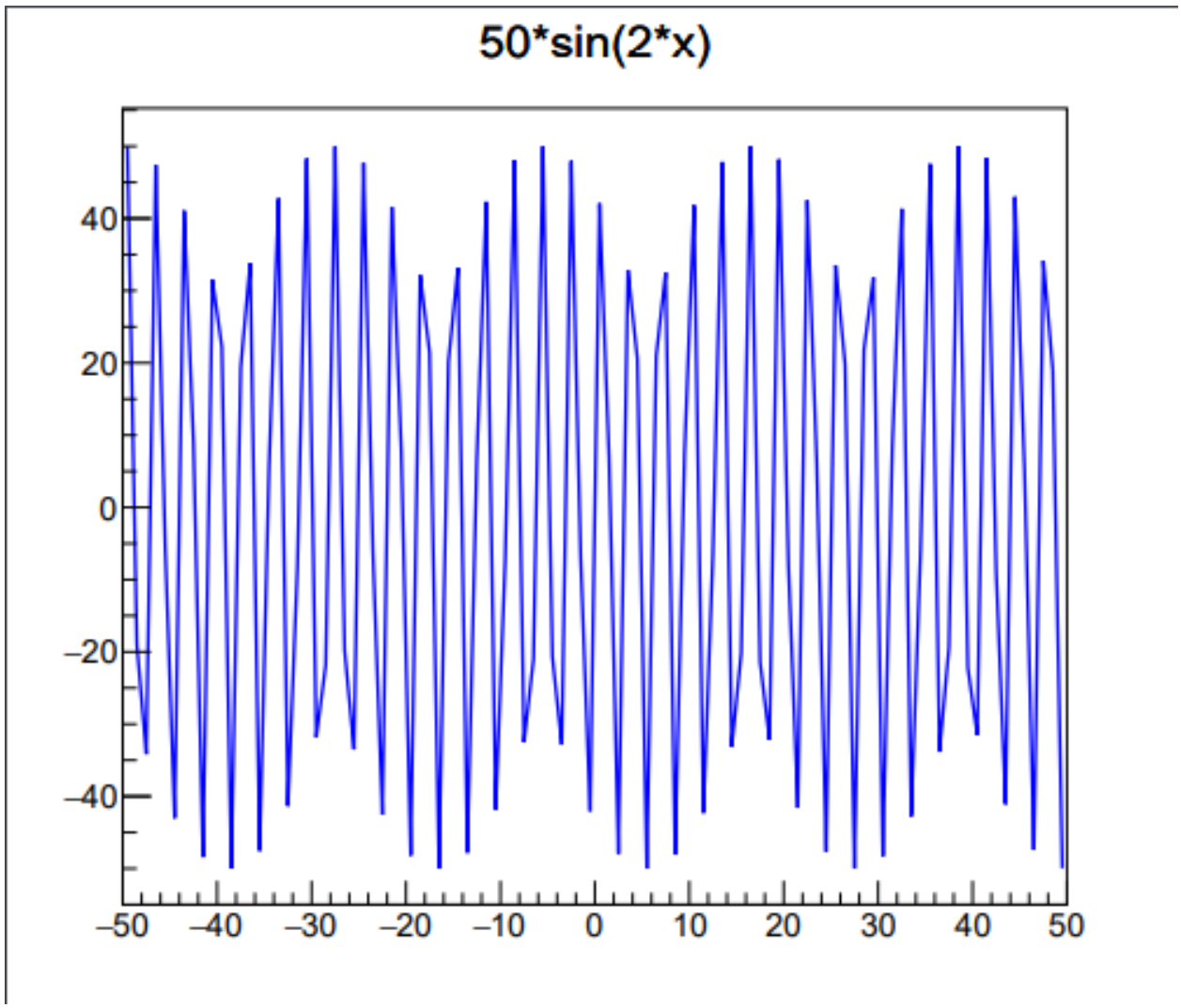
1. Create a function with parameters, p0 * sin (p1 * x) / x, and also draw it for different parameter values. Set the colour of the parametric function to blue. After having drawn the function, compute for the parameter values (p0 = 1, p1 = 2):
   a. Function value for x=1
   b. Function derivative for x=1
   c. Integral of the function between 0 and 3

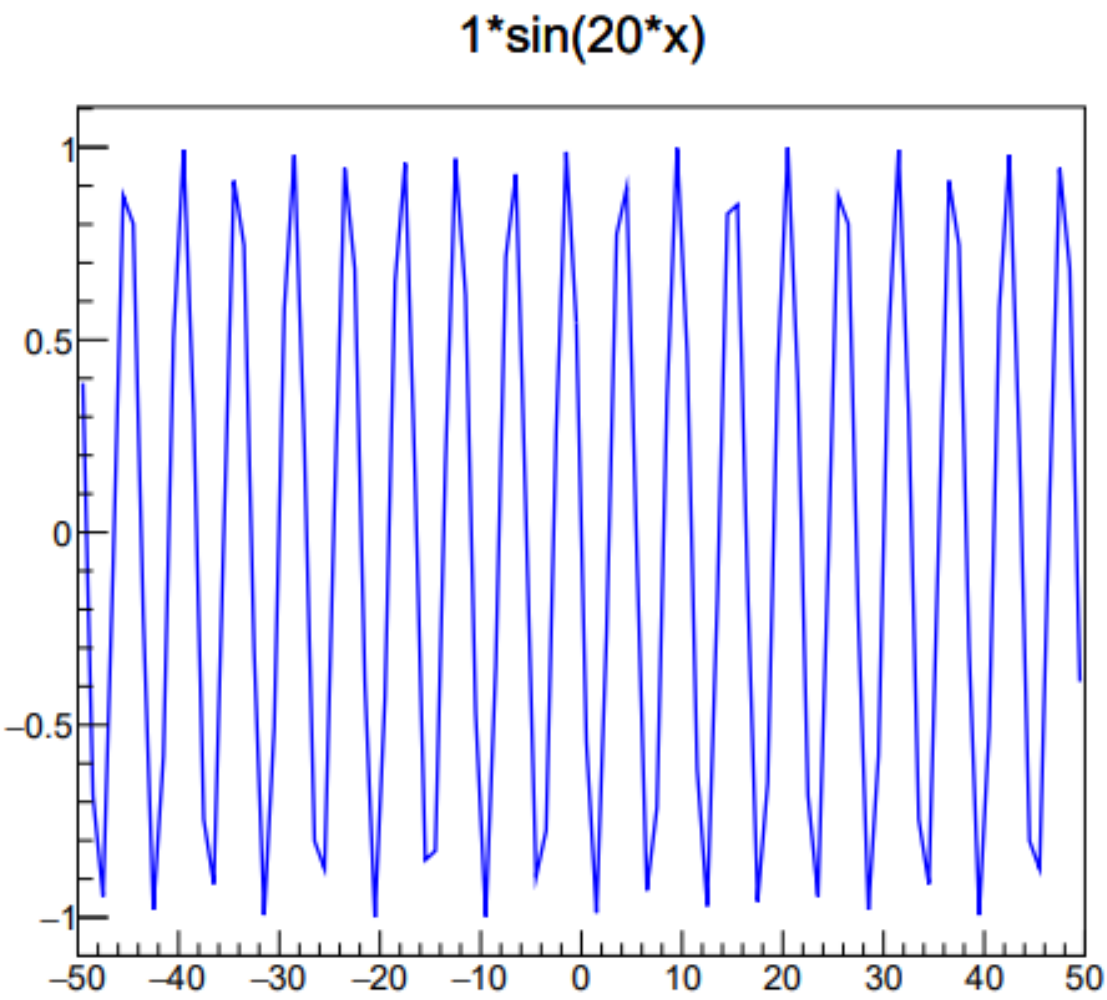
```
void ex1(){

    //criando a função e colocando a cor azul nela
    auto f1 = new TF1("f1", "1*sin(2*x)", -50, 50);
    f1->SetLineColor(kBlue);

    auto f2 = new TF1("f2", "50*sin(2*x)", -50, 50);
    f2->SetLineColor(kBlue);

    auto f3 = new TF1("f3", "1*sin(20*x)", -50, 50);
    f3->SetLineColor(kBlue);

    // Avaliando f1 em x = 1
    double val1 = f1->Eval(1);
    cout << "O valor de f1(1) é: " << val1 << endl;

    // Avaliando a derivada de f1 em x = 1
    double deriv1 = f1->Derivative(1);
    cout << "A derivada da função com p0=1 e p1=2 em x = 1 é: " << deriv1 << endl;

    // Calculando a integral de f1 no intervalo [0, 3]
    double integral1 = f1->Integral(0, 3);
    cout << "A integral da função com p0=1 e p1=2 de 0 a 3 é: " << integral1 << endl;

    //fazendo o canvas e Draw
    TCanvas *c1 = new TCanvas("c1", "c1", 10, 10, 800, 700);
    TCanvas *c2 = new TCanvas("c2", "c2", 10, 10, 800, 700);
    TCanvas *c3 = new TCanvas("c3", "c3", 10, 10, 800, 700);
```

```
lbrasil@c404fc8b068b:~/aula3-Root/exercicio 1$ root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.33.01                        https://root.cern |
  | (c) 1995-2024, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for linuxx8664gcc on Aug 06 2024, 17:41:19                 |
  | From heads/master@v6-31-01-2854-gd0fce0ae2a                      |
  | With c++ (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0                   |
  | Try '.help'/'.?', '.demo', '.license', '.credits', '.quit'/'.q'  |
   ------------------------------------------------------------------

root [0] .x ex1.C
O valor de f1(1) é: 0.909297
A derivada de f1 em x = 1 é: -0.832291
A integral de f1 de 0 a 3 é: 0.0199149
Info in <TCanvas::Print>: pdf file ex1.pdf has been created using the current canvas
Info in <TCanvas::Print>: Current canvas added to pdf file ex1.pdf
Info in <TCanvas::Print>: Current canvas added to pdf file ex1.pdf and file closed
root [1]
```

1*sin(2*x)



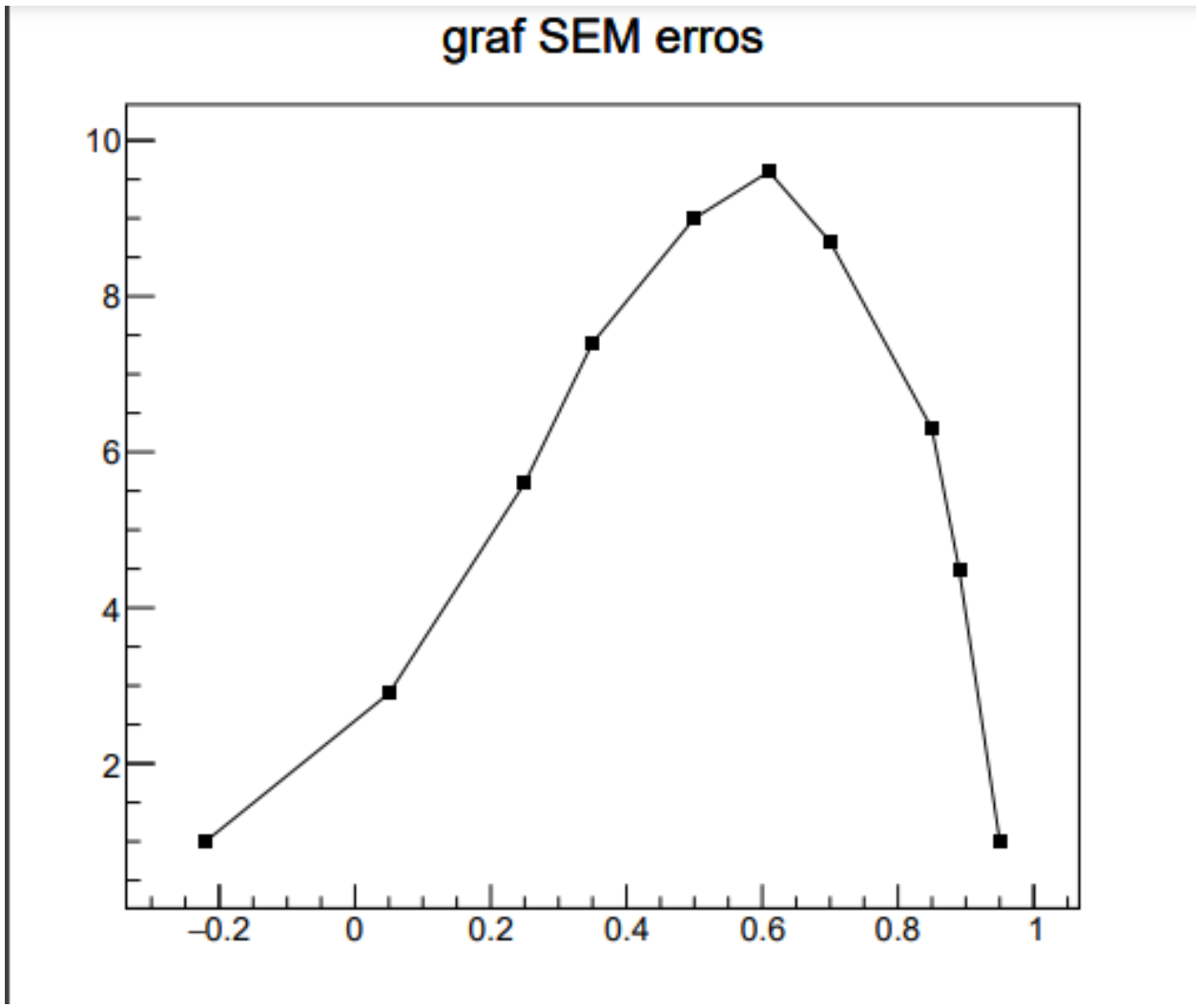

50*sin(2*x)
40
20
0
−20
−40
−50
−40
−30
−20
−10
0
10
20
30
40
50

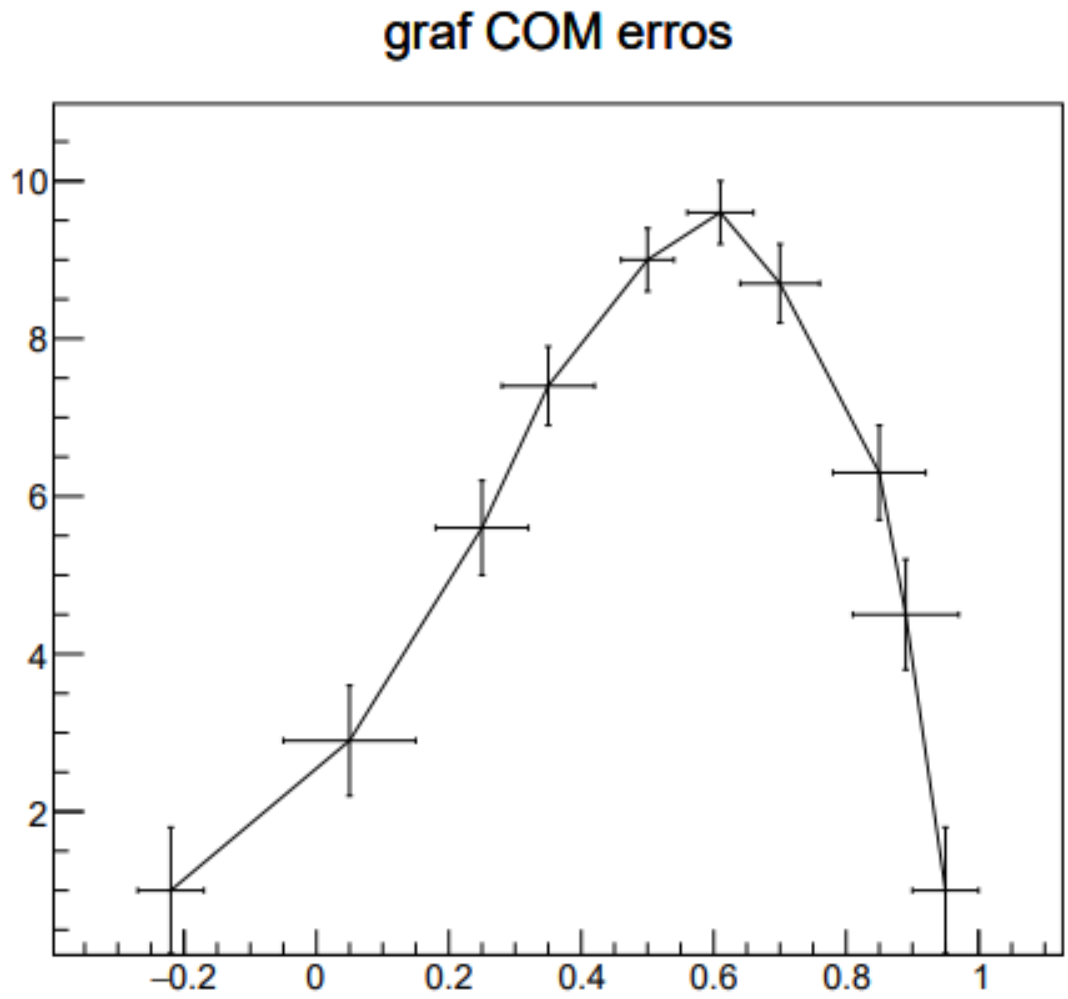


3 Exercício 2

2. Suppose you have this set of points defined in the attached file graphdata.txt. Plot these points using the TGraph class. Use as marker point a black box. Looking at the possible options for drawing the TGraph in TGraphPainter, plot a line connecting the points. Make a TGraphError and display it by using the attached data set, graphdata_error.txt, containing error in x and y.

```
void ex2(){

    //criando os grafícos
    TGraph *g1 = new TGraph("graphdata.txt", "%lg %lg");
    g1->SetTitle("graf SEM erros");
    g1->SetMarkerStyle(21);

    TGraphErrors *g2 = new TGraphErrors("graphdata_error.txt", "%lg %lg %lg %lg");
    g2->SetTitle("graf COM erros");

    //fazendo o canvas e Draw
    TCanvas *c1 = new TCanvas("c1", "c1", 10, 10, 800, 700);
    TCanvas *c2 = new TCanvas("c2", "c2", 10, 10, 800, 700);


    c1->cd();
    g1->Draw("APL");
    c1->Print("ex2.pdf(");



    c2->cd();
    g2->Draw("APL");
    c2->Print("ex2.pdf)");
```

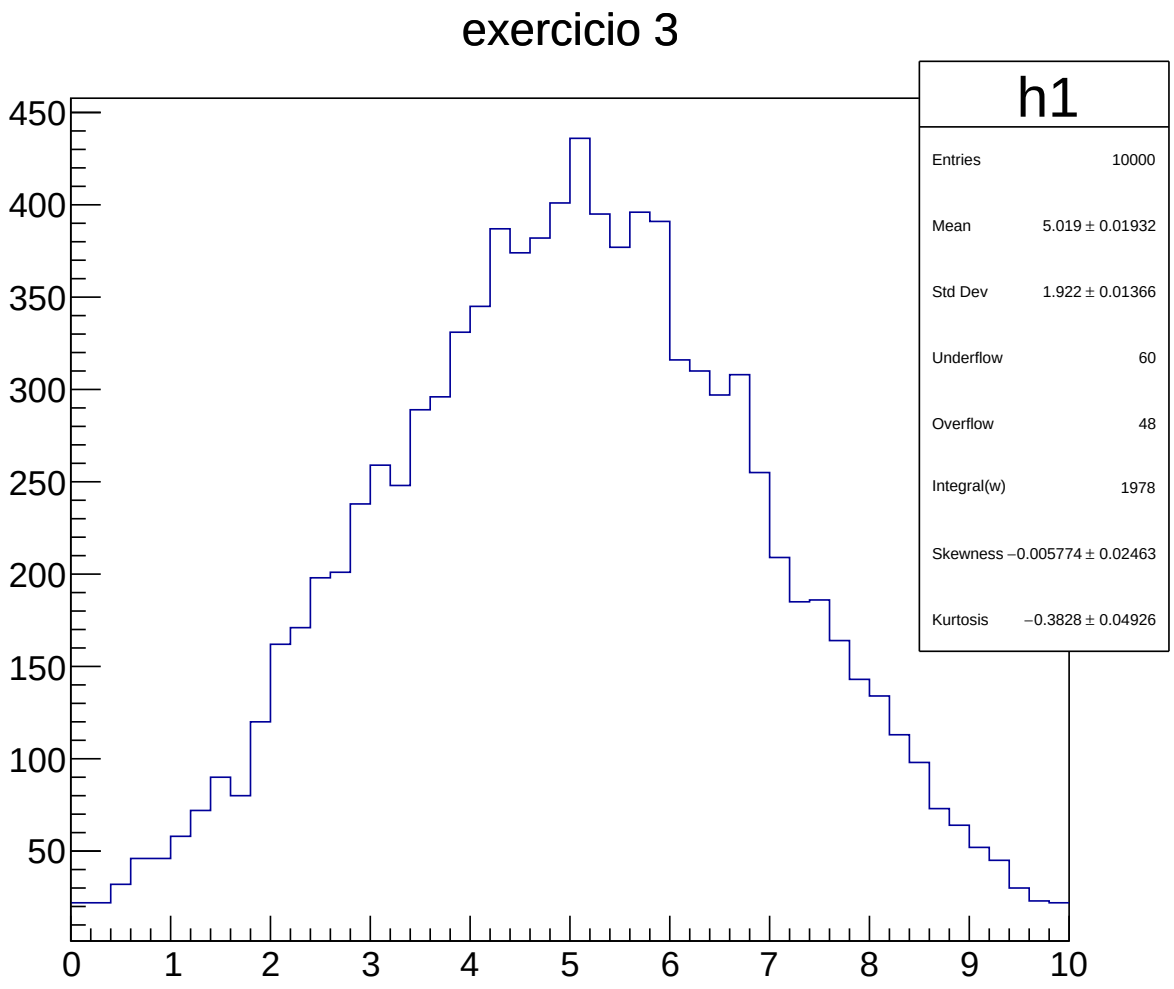

graf SEM erros
10
8
6
4
2
–0.2
0
0.2
0.4
0.6
0.8
1



4 Exercício 3

3. Create a one-dimensional histogram with 50 bins between 0 to 10, and fill it with 10000 gaussian distributed random numbers with mean 5 and sigma 2. Plot the histogram and, looking at the documentation in the THistPainter, show in the statistic box the number of entries, the mean, the RMS, the integral of the histogram, the number of underflows, the number of overflows, the skewness and the kurtosis.

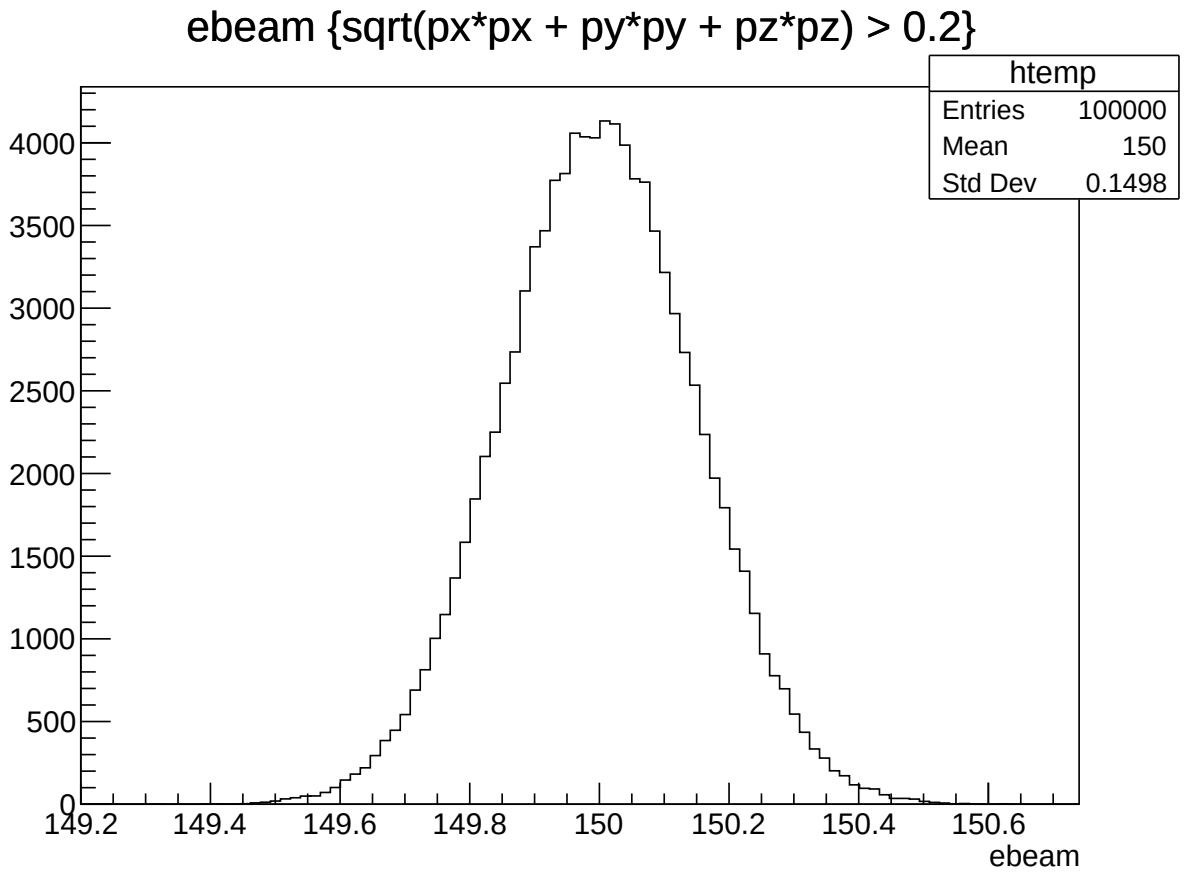
```
void ex3(){

    auto h1 = new TH1F("h1", "exercicio 3", 50, 0, 10);

    TRandom *rand = new TRandom();

    for (int i=0;i<10000;i++) {
        Double_t nrand = rand->Gaus(5,2);
        h1->Fill(nrand); }

    TCanvas *c1 = new TCanvas("c1", "c1", 10, 10, 800, 700);

    gStyle->SetOptStat(222112211);
    h1->Draw();
    c1->Print("ex3.pdf");
```



5 Exercício 4

4. Using the tree contained in tree.root make a distribution of the total momentum of each whose beam energy was outside of the mean by more than 0.2. Use TCut objects to make your events selections. Project this distribution into a histogram, draw it and save it to a file.

```
void ex4() {

    TFile *file = new TFile("tree.root");
    TTree *tree1 = (TTree*)file->Get("tree1");

    Float_t ebeam, px, py, pz;

    tree1->SetBranchAddress("ebeam", &ebeam);
    tree1->SetBranchAddress("px", &px);
    tree1->SetBranchAddress("py", &py);
    tree1->SetBranchAddress("pz", &pz);

    TCut *cut = new TCut("sqrt(px*px + py*py + pz*pz) > 0.2");

    TH1F *h1 = new TH1F("h1", "Momento Total", 100, 0, 300);

    tree1->Draw("px");

    TCanvas *c1 = new TCanvas("c1", "c1", 800, 600);
    h1->Draw();
    tree1->Draw("ebeam", *cut);

    c1->SaveAs("ex4.pdf");

    file->Close();
}
```


ebeam {sqrt(px*px + py*py + pz*pz) > 0.2}
htemp
Entries 100000
Mean 150
Std Dev 0.1498
4000
3500
3000
2500
2000
1500
1000
500
0
149.2
149.4
149.6
149.8
150
150.2
150.4
150.6
ebeam