Introdução à Análise de dados em FAE **(DATA)**

Lista 3

Professores: Sandro, Mauricio e Eliza *Name:* Dalmo da Silva Dalto

https://github.com/Dalmomr/web-project/tree/new_branch

EXERCICIO 0:

```
    {
TCanvas *c1= new TCanvas("c1","c1",1500,800);

c1->Divide(3,1);


TFile *input = new TFile("DYJetsToLL.root","read");
TTree *t = (TTree *)input->Get("Events");

c1->cd(1);


t->Draw("Muon_mass");

c1->cd(2);

t->Draw("Muon_charge");


c1->cd(3);

t->Draw("run");

}
```

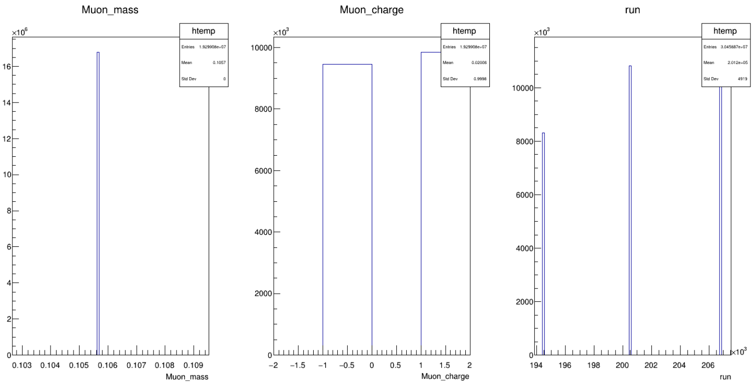

Figura 1: EXERCICIO 0

EXERCICIO 1:

```
{
```

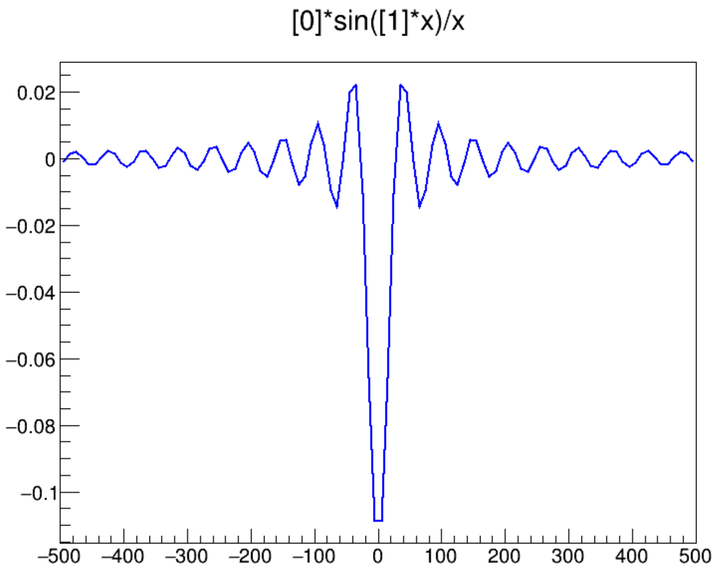

Figura 2: EXERCICIO 1

```

TF1 * f1 = new TF1("f1","[0]*sin([1]*x)/x", -500, +500);

f1->SetParameter(0, 1); // p0 = 1
f1->SetParameter(1, 2); // p1 = 2
f1->SetLineColor(kBlue);


TCanvas *c1 = new TCanvas("c1", "Function Plot", 800, 600);

f1->Draw();

double functionValue = f1->Eval(1);
printf("a. Function value for x = 1: %.4f\n", functionValue);

double functionDerivative = f1->Derivative(1);
printf("b. Function derivative for x = 1: %.4f\n", functionDerivative);

double integral = f1->Integral(0, 3);
printf("c. Integral of the function between 0 and 3: %.4f\n", integral);
}
```

A saída dessa macro resulta em um plot mostrado acima e os seguintes resultados:
a. Function value for x = 1: 0.9093
b. Function derivative for x = 1: -nan
c. Integral of the function between 0 and 3: 1.4247
O resultado da derivada é nan porque não tem derivada naquele ponto.

EXERCICIO 2:

```
    {

ifstream arq1;
ifstream arq2;

arq1.open("graphdata.txt");
arq2.open("graphdata_error.txt");


float x[10],y[10];
float x_1[10], ex[10],y_1[10], ey[10];

int i=0;

while(!arq1.eof() and !arq2.eof()){

arq1>>x[i]>>y[i];
arq2>>x_1[i]>>y_1[i]>>ex[i]>>ey[i];

cout<<x[i]<<" "<<y[i]<<endl;

cout<<x_1[i]<<" "<<y_1[i]<<" "<<ex[i]<<" "<<ey[i]<<endl;


i++;


}

TCanvas *c1= new TCanvas("c1","c1",1000,500);

c1->Divide(2,1);
c1->cd(1);
TGraph *t= new TGraph(10,x,y);
t->Draw();
t->SetTitle("Plot sem barra de erros");

c1->cd(2);
TGraphErrors *t1= new TGraphErrors(10,x_1,y_1,ex,ey);
t1->Draw();
t1->SetTitle("Plot com barra de erros");

}
```

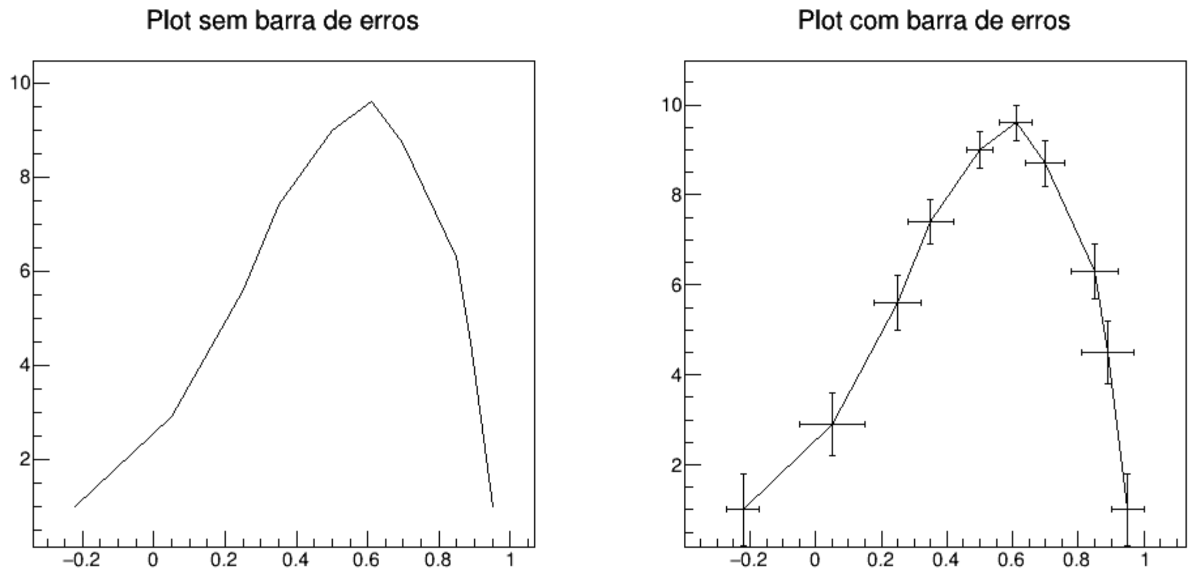

Figura 3: EXERCICIO 2

EXERCICIO 3:

```
    {

TRandom *t= new TRandom;
TH1F *h1= new TH1F("Fit gauss","Fit gauss",50,0,10);

for(int i=0; i<10000;i++){

        float random_gauss= t->Gaus(5,2);

        h1->Fill(random_gauss);
}

h1->Draw();
h1->Fit("gaus");
gStyle->SetOptFit(1111);
}
```

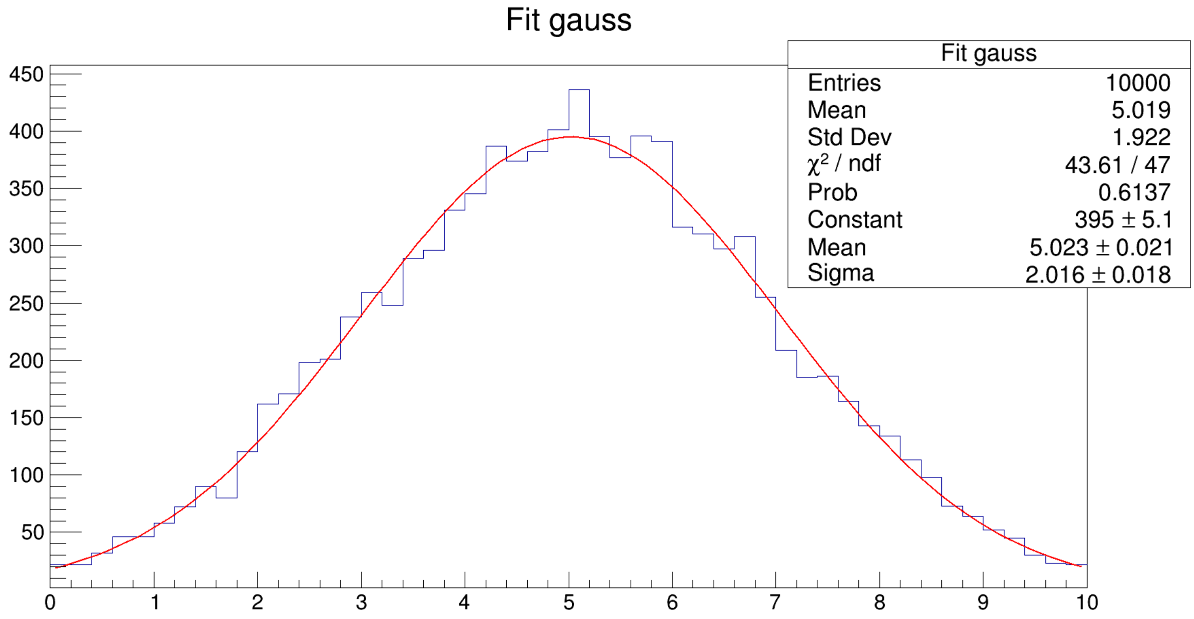

Figura 4: EXERCICIO 3

EXERCICIO 4:

```
    {
TCanvas *c1= new TCanvas("c1","c1",1500,800);

TFile *input = new TFile("tree.root","read");
TTree *t = (TTree *)input->Get("tree1");

gStyle->SetOptFit(1111);

TF1 *gauss = new TF1("g1","gaus",0,10);


c1->Divide(3,1);
c1->cd(1);
t->Draw("ebeam>>h1");
h1->Fit(gauss);

float mean=gauss->GetParameter(1);
float sigma=gauss->GetParameter(2);

c1->cd(2);

char cut[20];

sprintf(cut,"ebeam>%f",0.2 + mean);

t->Draw("px+py+pz>>h2",cut,"");


h2->Fit(gauss);
h2->SetTitle("px+py+pz (ebeam>0.2 + Mean_{ebeam})");

c1->cd(3);


t->Draw("px+py+pz>>h3","","");

h3->Fit(gauss);
h3->SetTitle("px+py+pz");

c1->Print("exercicio4.png","png");
}
```

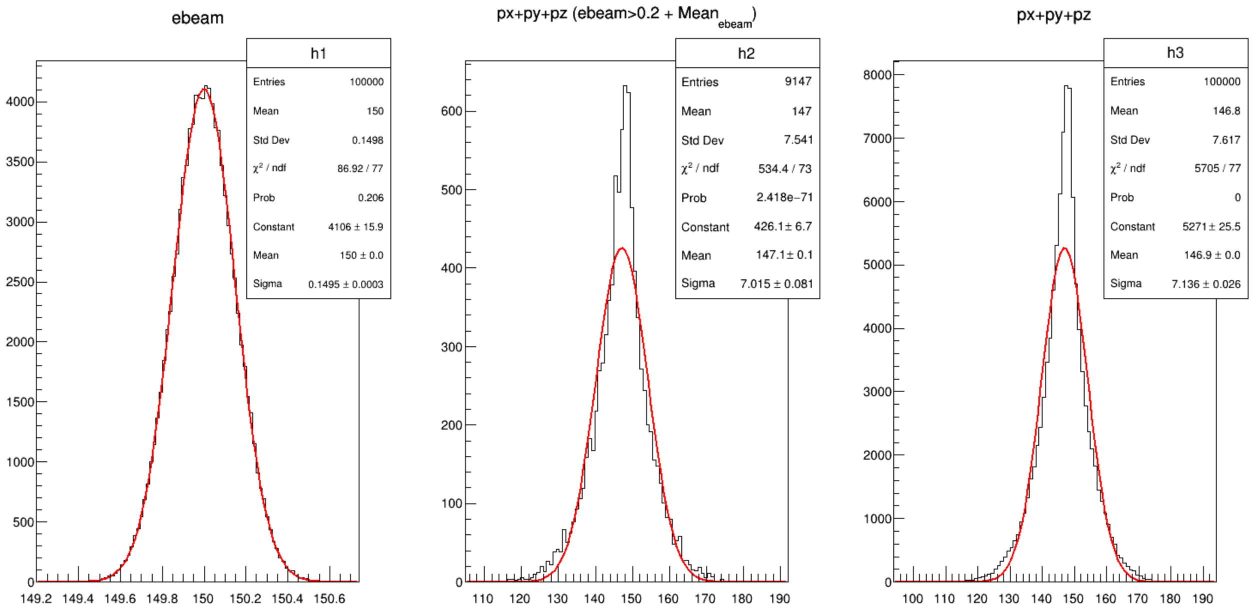

Figura 5: EXERCICIO 4