Introdução à Análise de dados em FAE **(05/09/2024)**

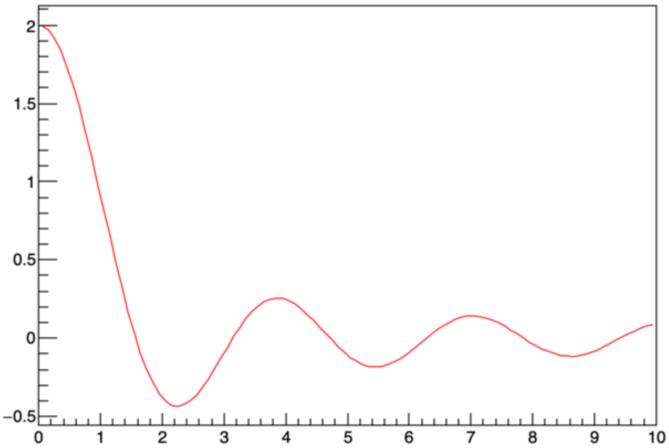
Exercicios de estatística para análise de dados em HEP

Professores: Eliza Melo, Dilson Damião e Mauricio Thiel *Name:* Thiago de Andrade Rangel Monteiro

EXERCÍCIO 1

```cpp

void exercicio_1(){

    TCanvas *c1 = new TCanvas();
    TF1 *f1 = new TF1("funcao", "([0]*sin([1]*x))/x", 0, 10);
    f1->SetTitle("");
    f1->SetParameters(1, 2);

    // Calculo de f(1)
    double valor_em_1 = f1->Eval(1);
    std::cout << "Valor da funcao em x=1: " << valor_em_1 << std::endl;
    // Calculo da derivada em x=1
    double derivada_em_1 = f1->Derivative(1);
    std::cout << "Derivada da funcao em x=1: " << derivada_em_1 << std::endl;
    // Calculo da integral entre 0 e 3
    double valor_integral = f1->Integral(0, 3);
    std::cout << "Integral da funcao entre 0 e 3: " << valor_integral << std::endl;

    f1->Draw();
    c1->Draw();
    c1->SaveAs("exercicio_1.png");
}
```

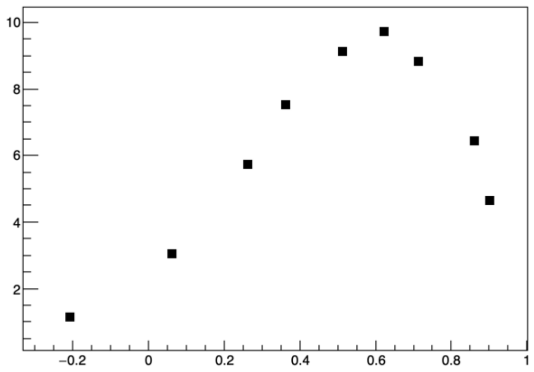
output:

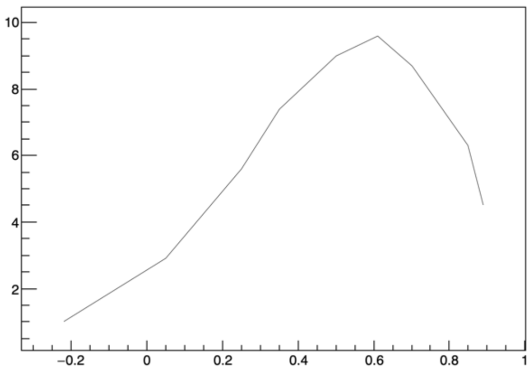


```
Valor da funcao em x=1: 0.909297
Derivada da funcao em x=1: -1.74159
Integral da funcao entre 0 e 3: 1.42469
```

EXERCÍCIO 2

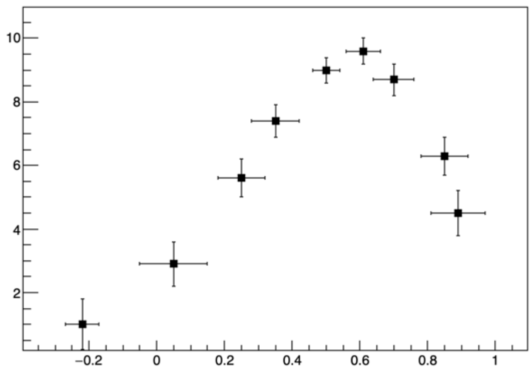
```
    void exercicio_2() {
    TCanvas *c1 = new TCanvas();
    TGraph *gr = new TGraph();

    std::ifstream file("/Users/thiagorangel/UERJ/Introducao_Analise_de_Dados_FAE/data/graphdata.txt"); // Abre o arquivo

    double x, y;

    while (1) {
        file >> x >> y;

        if (file.eof()) {break;}

        gr->SetPoint(gr->GetN(), x, y);
    }

    gr->SetMarkerSize(10);
    gr->SetMarkerStyle(1);
    gr->Draw("AP");
    c1->Draw();
    c1->SaveAs("graph_1.png");

    gr->SetMarkerSize(10);
    gr->SetMarkerStyle(1);
    gr->Draw("AL");
    c1->Draw();
    c1->SaveAs("graph_2.png");
}
```

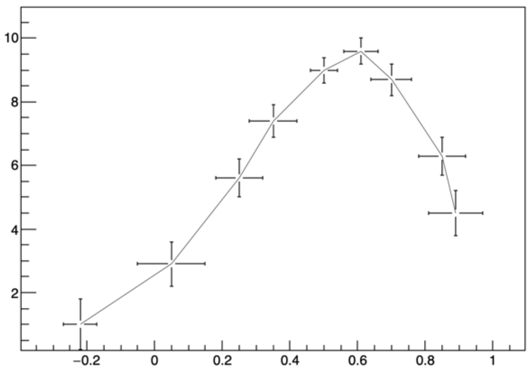




EXERCÍCIO 2

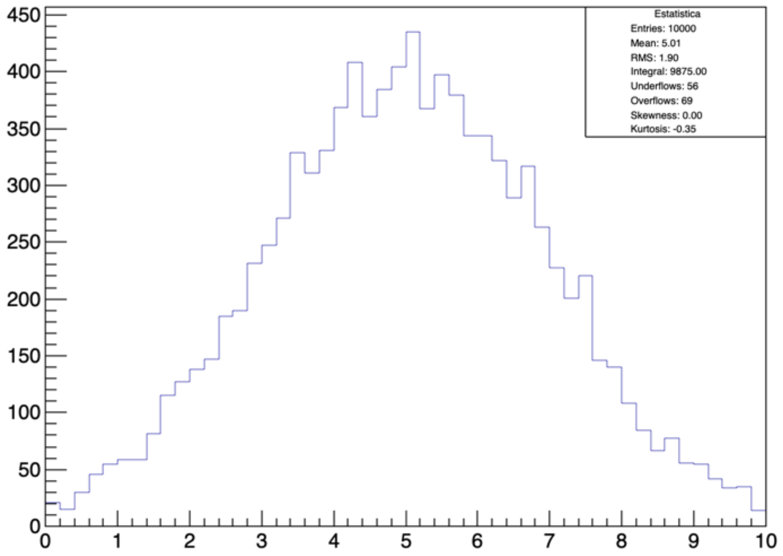
```cpp
    void exercicio_3() {

    TCanvas *c1 = new TCanvas();
    TGraphErrors *gr = new TGraphErrors();

    std::ifstream file("/Users/thiagorangel/UERJ/Introducao_Analise_de_Dados_FAE/data/graphdata_error.txt"); // Abre o arquivo

    double x, y, ex, ey;
    while (1) {
        file >> x >> y >> ex >> ey;

        if (file.eof()) {break;}

        gr->SetPoint(gr->GetN(), x, y);
        gr->SetPointError(gr->GetN() - 1, ex, ey);
    }

    gr->SetMarkerSize(1);
    gr->SetMarkerStyle(21);
    gr->Draw("AP");
    c1->SaveAs("graph_error_1.png");

    gr->SetMarkerSize(1);
    gr->SetMarkerStyle(21);
    gr->Draw("AP");
    c1->Draw();
    c1->SaveAs("graph_error_2.png");

    file.close();
}
```





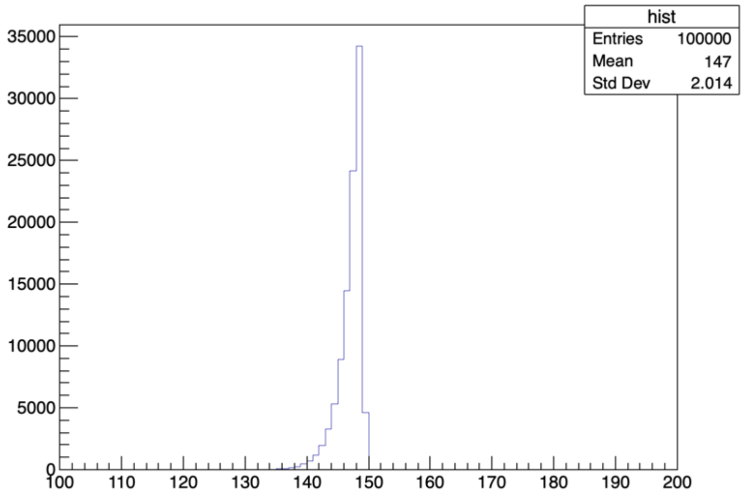
EXERCÍCIO 3

```cpp
    void exercicio_4() {

    TCanvas *c1 = new TCanvas("c1", "Canvas", 800, 600);
    TH1F *hist = new TH1F("hist", "", 50, 0, 10);
    TRandom3 *rand = new TRandom3();

    for (int i = 0; i < 10000; ++i) {
        double value = rand->Gaus(5, 2);
        hist->Fill(value);
    }

    hist->Draw();
    hist->SetStats(0);
    TLegend *legend = new TLegend(0.7, 0.7, 0.9, 0.9);
    legend->SetHeader("Estatistica", "C");

    legend->AddEntry((TObject*)0, Form("Entries: %d", (int)hist->GetEntries()), "");
    legend->AddEntry((TObject*)0, Form("Mean: %.2f", hist->GetMean()), "");
    legend->AddEntry((TObject*)0, Form("RMS: %.2f", hist->GetRMS()), "");
    legend->AddEntry((TObject*)0, Form("Integral: %.2f", hist->Integral()), "");
    legend->AddEntry((TObject*)0, Form("Underflows: %d", (int)hist->GetBinContent(0)), "");
    legend->AddEntry((TObject*)0, Form("Overflows: %d", (int)hist->GetBinContent(hist->GetNbinsX() + 1)), "");
    legend->AddEntry((TObject*)0, Form("Skewness: %.2f", hist->GetSkewness()), "");
    legend->AddEntry((TObject*)0, Form("Kurtosis: %.2f", hist->GetKurtosis()), "");
    legend->Draw();

    c1->SaveAs("histogram_estatistica.png");
}
```



EXERCÍCIO 4

```
    void exercicio_5()
{


    TCanvas *c1 = new TCanvas("c1", "Histograma de Momento", 800, 600);
    TFile *file = new TFile("/Users/thiagorangel/UERJ/Introducao_Analise_de_Dados_FAE/data/tree.root");
    TTree *tree = (TTree*)file->Get("tree1");
    TH1F *hist = new TH1F("hist", "", 100, 100, 200);
    TH1F *histEbeam = new TH1F("histEbeam", "", 100, 0, 1000);
    float px, py, pz, ebeam;

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

    Int_t nEntries = tree->GetEntries();

    for (Int_t i = 0; i < nEntries; i++) {
        tree->GetEntry(i);
        histEbeam->Fill(ebeam);
    }

    float meanEbeam = histEbeam->GetMean();
    for (Int_t i = 0; i < nEntries; i++) {
        tree->GetEntry(i);
        if (ebeam > meanEbeam * 0.2) {
            float pMagnitude = sqrt(px * px + py * py + pz * pz);
            hist->Fill(pMagnitude);
        }
    }

    hist->Draw();
    c1->SaveAs("histograma_momento.png");
}
```


hist
Entries 100000
Mean 147
Std Dev 2.014