Introdução à Análise de dados em FAE **(15/10/2024)**

Exercícios - Aula ROOT

Professores: Dilson, Eliza & Maurício *Name:* Diogo Gomes Santos Caffonso de Morais

EXERCICIO 1

Para o primeiro exercício, foi desenvolvido o código que segue:

```
#include <RooRealVar.h>
#include <RooCBShape.h>
#include <RooDataSet.h>
#include <RooPlot.h>
#include <TCanvas.h>
#include <TStyle.h>
#include <TPaveText.h>

void exercicio1() {
        RooRealVar x("x", "x", -10, 10);
        RooRealVar mean("mean", "Mean", 0, -10, 10);
        RooRealVar sigma("sigma", "Sigma", 1, 0.1, 5);
        RooRealVar alpha("alpha", "Alpha", 1.5, 0.1, 5);
        RooRealVar n("n", "n", 2, 0.1, 10);

        RooCBShape crystalBall("crystalBall", "Crystal Ball PDF", x, mean, sigma,
            alpha, n);

        RooDataSet* data = crystalBall.generate(x, 10000);

        crystalBall.fitTo(*data);

        RooPlot* xframe = x.frame();
        data->plotOn(xframe);
        crystalBall.plotOn(xframe);

        TCanvas* c1 = new TCanvas("c1", "Ajuste Crystal Ball", 800, 600);
        xframe->Draw();

        gStyle->SetOptStat(1111);
        gStyle->SetOptFit(1111);

        TPaveText* paramsText = new TPaveText(0.6, 0.7, 0.9, 0.9, "NDC");
        paramsText->AddText(Form("Mean ajustado = %.3f", mean.getVal()));
        paramsText->AddText(Form("Sigma ajustado = %.3f", sigma.getVal()));
        paramsText->AddText(Form("Alpha ajustado = %.3f", alpha.getVal()));
        paramsText->AddText(Form("n ajustado = %.3f", n.getVal()));
        paramsText->SetFillColor(0);
        paramsText->SetTextSize(0.03);
        paramsText->Draw();

        c1->SaveAs("exercicio1.png");
}
```

Executado o código acima, tive o seguinte output:

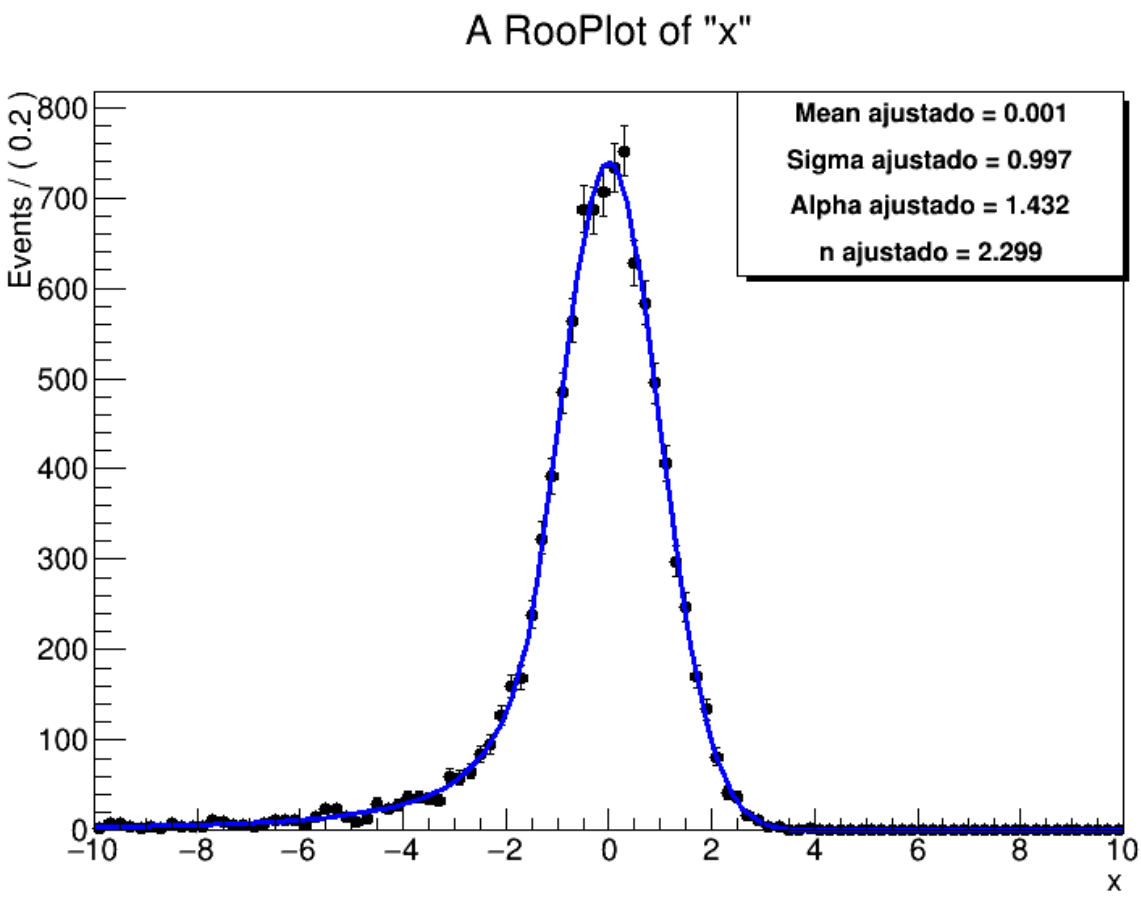

Figura 1: Gráfico da função de fit.

EXERCICIO 2

Para o segundo exercício, foi desenvolvido o código que segue:

```
#include <iostream>
#include <cmath>
#include "TH1F.h"
#include "TF1.h"
#include "TCanvas.h"
#include "TRandom.h"

void exercicio2() {

        int nEvents = 1500;
        double lambdaTrue = 1.0;
        double x[nEvents];

        for (int i = 0; i < nEvents; i++) {
                x[i] = -log(1 - gRandom->Uniform()) / lambdaTrue;
        }

        TH1F *hist = new TH1F("hist", "Histograma de eventos simulados", 50, 0, 10);
        for (int i = 0; i < nEvents; i++) {
                hist->Fill(x[i]);
        }

        TF1 *fitFunction = new TF1("fitFunction", "[0] * exp(-[1] * x)", 0, 10);
        fitFunction->SetParameters(nEvents, 1.0);
        fitFunction->SetParLimits(1, 0.1, 2);
        hist->Fit("fitFunction", "R");

        double lambdaAdjusted = fitFunction->GetParameter(1);
        double totalYield = fitFunction->Integral(0, 10);

        TCanvas *c1 = new TCanvas("c1", "Ajuste da fun   o exponencial", 800, 600);
        hist->Draw();
        fitFunction->Draw("same");
        std::cout << "Valor ajustado para lambda: " << lambdaAdjusted << std::endl;

        c1->SaveAs("exercicio2.png");
}
```

Que gerou o gráfico:

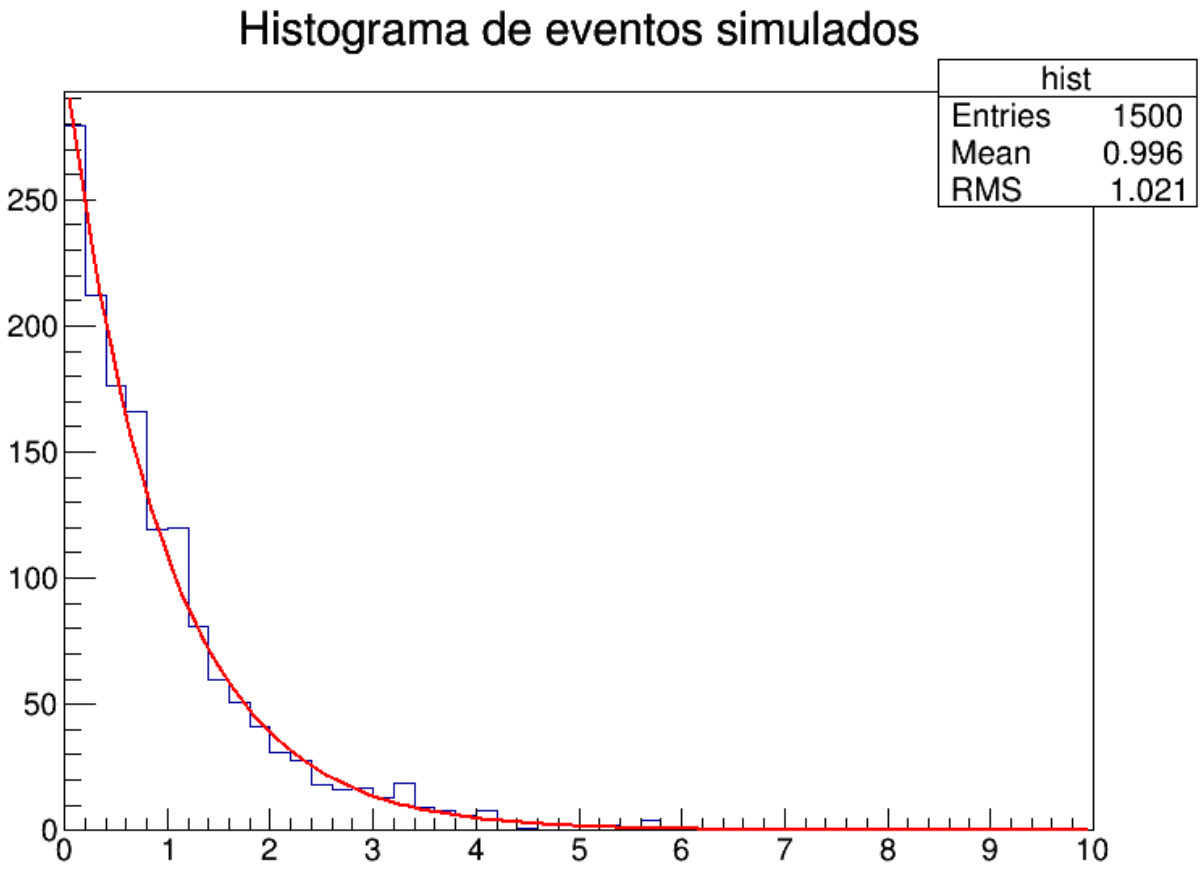

Figura 2: Histograma com função de fit.

Respondendo às perguntas:
$\lambda = 1,03$
$Número total de eventos ajustados$: 1500
Estão dentro das expectativas, já que não desviam muito dos valores propostos.

EXERCICIO 3

Para o terceiro exercício, foi desenvolvido o código que segue:

```
#include <iostream>
#include "RooDataSet.h"
#include "RooRealVar.h"
#include "RooCBShape.h"
#include "RooPolynomial.h"
#include "RooAddPdf.h"
#include "RooPlot.h"
#include "TFile.h"
#include "TCanvas.h"
#include "TLegend.h"

void exercicio3() {
        TFile *file = TFile::Open("/home/cms-opendata/FAE_Repo/Trabalho4/ex3/
            DataSet_lowstat.root");
        RooDataSet *data = dynamic_cast<RooDataSet*>(file->Get("data"));

        RooRealVar mass("mass", "Massa", 2.8, 3.5);
        RooRealVar mean("mean", "M dia", 3.096916, 3.0, 3.2);
        RooRealVar sigma("sigma", "Desvio Padr o", 0.05, 0.01, 0.1);
        RooRealVar alpha("alpha", "Alpha", 1.5);
        RooRealVar n("n", "n", 0.5, 0, 5);
        RooCBShape signal("signal", "Sinal J/  ", mass, mean, sigma, alpha, n);

        RooRealVar a0("a0", "a0", 0, -10, 10);
        RooRealVar a1("a1", "a1", 1000, 0, 10000);
        RooPolynomial background("background", "Fundo", mass, RooArgList(a0, a1));

        RooRealVar nsig("nsig", "N mero de eventos sinal", 1000, 0, 10000);
        RooRealVar nbkg("nbkg", "N mero de eventos fundo", 1000, 0, 10000);
        RooAddPdf model("model", "Modelo Sinal + Fundo", RooArgList(signal,
            background), RooArgList(nsig, nbkg));

        model.fitTo(*data);
        RooPlot *frame = mass.frame();
        frame->SetXTitle("Massa (GeV/c^{2})");

        data->plotOn(frame);
        model.plotOn(frame);
        model.paramOn(frame);

        double chi2 = frame->chiSquare();
        int nParams = model.getParameters(*data)->getSize();
        int nPoints = data->numEntries();
        int ndf = nPoints - nParams;

        std::cout << "Chi2/ndf: " << chi2 / ndf << std::endl;

        TCanvas *c1 = new TCanvas("c1", "Ajuste da resson ncia J/  ", 800, 600);
        frame->Draw();

        c1->SaveAs("exercicio3.png");
}
```

Que gerou o gráfico:

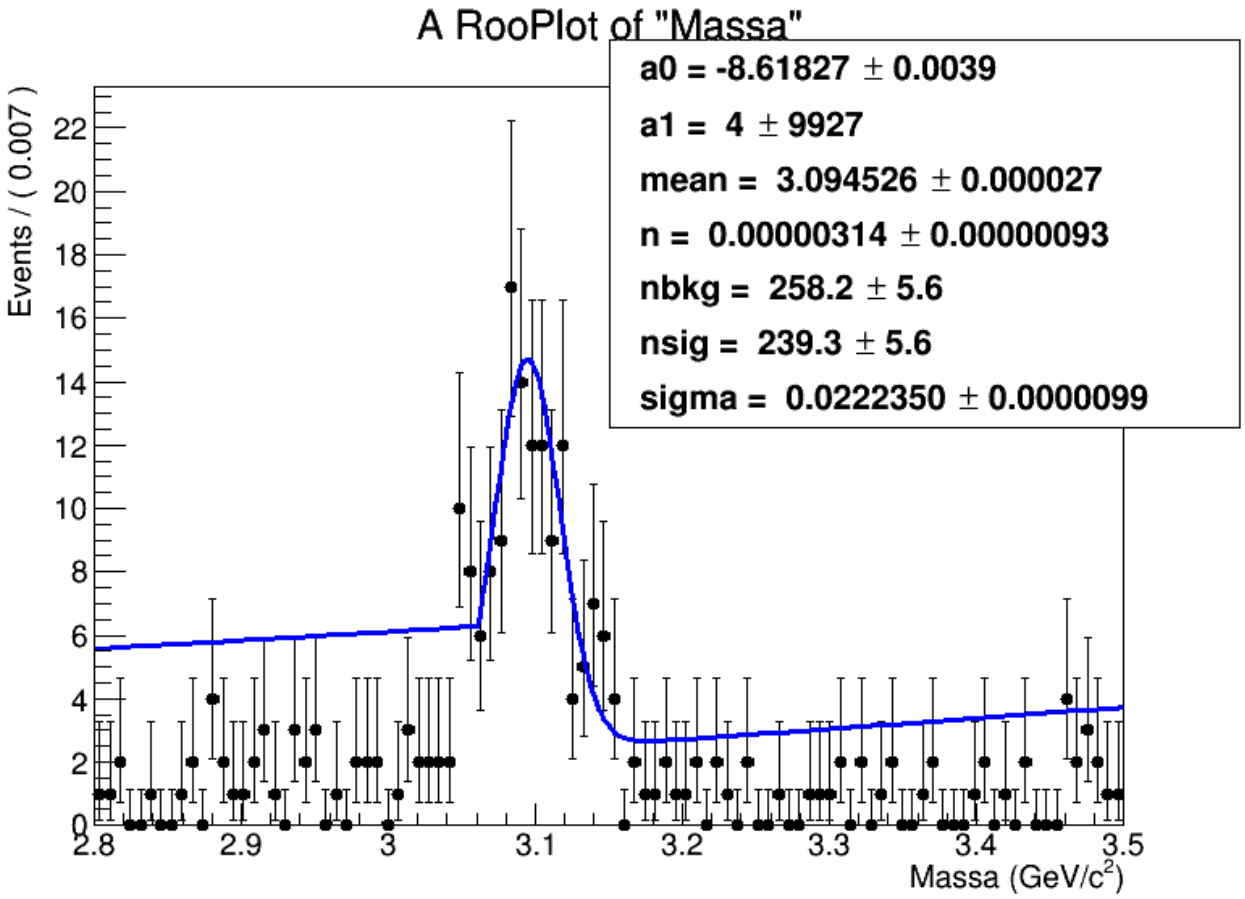

Figura 3: Gráfico de dispersão função de fit.

$$\chi^2/ndf = 2,68753 \times 10^{-3}$$

Obs.: A razão χ^2/ndf está muito pequena, o que pode levar a um "excesso"de ajuste.