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

Exercício 4

Professores: NOME DOS PROFESSORES *Name:* Dalmo da Silva Dalto

TEXTO

EXERCICIO 1

```
{

using namespace RooFit;


TFile *fin = new TFile("example_data.root");
TNtuple *nt = (TNtuple*)fin->Get("nt");

// Definir a vari vel RooRealVar
RooRealVar mass("mass", "mass [GeV]", 0, 2.0);

// Criar o RooDataSet com a vari vel definida e o TNtuple
// Importar o dado unbinned
// -------------------------------------------------------------
// construtor do RooDataSet
RooDataSet rooData("data", "data", RooArgSet(mass), RooFit::Import(*nt));

// O m todo data.Print("v") imprime informa es sobre o RooDataSet,
// incluindo os nomes das vari veis e os valores dos eventos,
// para que voc  possa verificar se os dados foram importados corretamente.
rooData.Print("v");


RooPlot* frame3 = mass.frame(RooFit::Title("Imported Unbinned Mass"));
rooData.plotOn(frame3);

// Fit a Gaussian p.d.f to the data
RooRealVar mu("mu", "mean parameter", 1, 1.0, 1.0);
RooRealVar sigma("sigma", "Width of Gaussian", 0.05, 0.0, 0.05);

RooRealVar slope("slope", "Slope", -0.5, -10, 10);
RooRealVar intercept("intercept", "Intercept", 0, -10,10);

RooGaussian gauss("gaus", "Gaussian PDF", mass, mu, sigma);
RooPolynomial linear("linear", "Linear PDF", mass, RooArgList(intercept, slope),1);

RooRealVar sinal("sinal", "sinal", 0.5, 0, 1);

RooAddPdf sum_pdf("sum_pdf", "Sum of Gaussian and Linear", RooArgList(gauss, linear),
     RooArgList(sinal));


RooFitResult* result = sum_pdf.fitTo(rooData, RooFit::Save());

linear.plotOn(frame3, RooFit::LineColor(kRed));
sum_pdf.plotOn(frame3, RooFit::LineColor(kGreen));
gauss.plotOn(frame3, RooFit::LineColor(kBlack));


sum_pdf.fitTo(rooData);

// Estat sticas dos dados
//rooData.statOn(frame3);

// Criar a canvas para o plot
TCanvas *c3 = new TCanvas("exemplo03","exemplo03",800,400);
TPaveText *statistics = new TPaveText(0.6, 0.6, 0.9, 0.9, "NDC");
statistics->SetFillColor(0);
statistics->AddText(Form("Chi2/NdF = %.2f", frame3->chiSquare()));
statistics->AddText(Form("Linear Parameters:"));
statistics->AddText(Form("   Slope = %.3f", slope.getVal()));
statistics->AddText(Form("   Intercept = %.3f", intercept.getVal()));
statistics->AddText(Form("Gaussian Parameters:"));
statistics->AddText(Form("   Mean = %.3f", mu.getVal()));
statistics->AddText(Form("   Sigma = %.3f", sigma.getVal()));
statistics->AddText(Form("   amplitude = %.3f", sinal.getVal()));
statistics->Draw();

frame3->Draw();
c3->Draw();
statistics->Draw();
linear.Print("t")
}
```

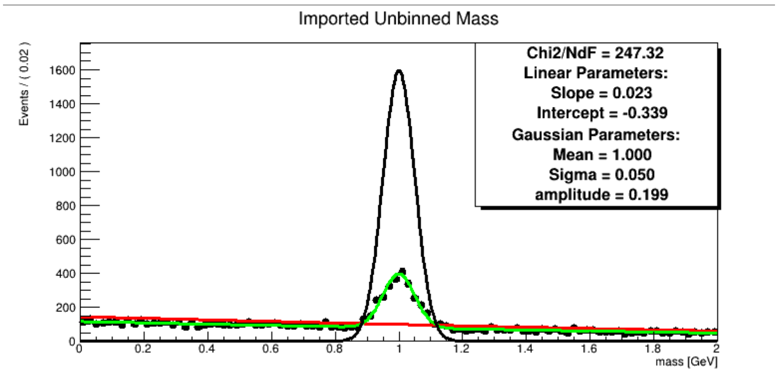

Figura 1: Exercício 0

EXERCICIO 2

```
    {
using namespace RooFit;


// Definindo vari vel e par metros da Crystal Ball
RooRealVar x("x", "x", -10, 10);
RooRealVar mean("mean", "mean", 0, -10, 10);
RooRealVar sigma("sigma", "sigma", 1, 0.1, 10);
RooRealVar alpha("alpha", "alpha", 1.5, 0.1, 10);
RooRealVar n("n", "n", 5, 0.1, 10);

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

// Gerando toy data
RooDataSet *data = crystalBall.generate(x, 1000);

// Ajustando os dados
RooFitResult *result = crystalBall.fitTo(*data, RooFit::Save(true));

// Criando o frame para o plot
RooPlot *frame = x.frame(RooFit::Title("Crystal Ball Fit"));

// Plotando os dados
data->plotOn(frame);

// Plotando a PDF ajustada
crystalBall.plotOn(frame);

// Criando a canvas para o plot
TCanvas *canvas = new TCanvas("canvas", "Crystal Ball Fit", 800, 600);

// Adicionando a caixa de informa  o estat stica para os dados e o modelo
TPaveText *statistics = new TPaveText(0.6, 0.6, 0.9, 0.9, "NDC");
statistics->SetFillColor(0);
statistics->AddText(Form("Data Statistics:"));
statistics->AddText(Form("Entries: %d", data->numEntries()));
statistics->AddText(Form("Model Statistics:"));
statistics->AddText(Form("Chi2/NdF: %.2f", frame->chiSquare()));
statistics->AddText(Form("Mean: %.2f", mean.getVal()));
statistics->AddText(Form("Sigma: %.2f", sigma.getVal()));
statistics->AddText(Form("Alpha: %.2f", alpha.getVal()));
statistics->AddText(Form("N: %.2f", n.getVal()));

// Desenhando o frame
frame->Draw();

// Desenhando a canvas
canvas->Draw();
statistics->Draw();


}
```

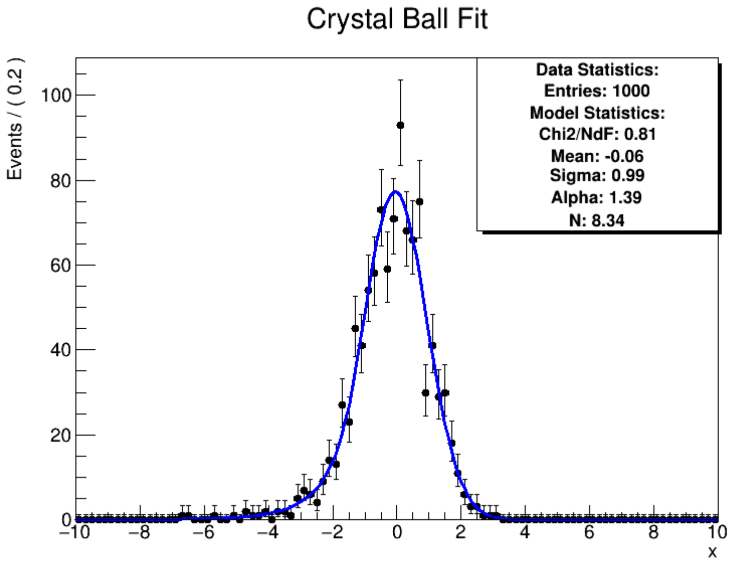

Figura 2: Exercício2

```

{
    // Definindo vari vel e par metros da Landau
    RooRealVar x("x", "x", -10, 10);
    RooRealVar mean("mean", "mean", 0, -10, 10);
    RooRealVar sigma("sigma", "sigma", 1, 0.1, 10);

    // Criando a PDF Landau
    RooLandau landau("landau", "Landau PDF", x, mean, sigma);

    // Gerando toy data
    RooDataSet *data = landau.generate(x, 1000);

    // Ajustando os dados
    RooFitResult *result = landau.fitTo(*data, RooFit::Save(true));

    // Criando o frame para o plot
    RooPlot *frame = x.frame(RooFit::Title("Landau Fit"));

    // Plotando os dados
    data->plotOn(frame);

    // Plotando a PDF ajustada
    landau.plotOn(frame);

    // Criando a canvas para o plot
    TCanvas *canvas = new TCanvas("canvas", "Landau Fit", 800, 600);

    // Adicionando a caixa de informa  o estat stica para os dados e o modelo
    TPaveText *statistics = new TPaveText(0.6, 0.6, 0.9, 0.9, "NDC");
    statistics->SetFillColor(0);
    statistics->AddText(Form("Data Statistics:"));
    statistics->AddText(Form("Entries: %d", data->numEntries()));
    statistics->AddText(Form("Mean: %.2f +/- %.2f", x.mean(), x.error()));
    statistics->AddText(Form("RMS: %.2f", x.rms()));
    statistics->AddText(Form(""));
    statistics->AddText(Form("Model Statistics:"));
    statistics->AddText(Form("Chi2/NdF: %.2f", frame->chiSquare()));
    statistics->AddText(Form("Mean: %.2f", mean.getVal()));
    statistics->AddText(Form("Sigma: %.2f", sigma.getVal()));
    statistics->Draw();

    // Desenhando o frame
    frame->Draw();

    // Desenhando a canvas
    canvas->Draw();
}
```

EXERCICIO 3

```

from ROOT import TFile
from ROOT import TLorentzVector
from ROOT import TH1F
from ROOT import TF1
import numpy as np
from ROOT import RooRealVar
from ROOT import RooFFTConvPdf
from ROOT import RooDataHist
from ROOT import RooDataSet
from ROOT import RooExponential
from ROOT import RooGaussian
from ROOT import RooPolynomial
from ROOT import RooVoigtian
```

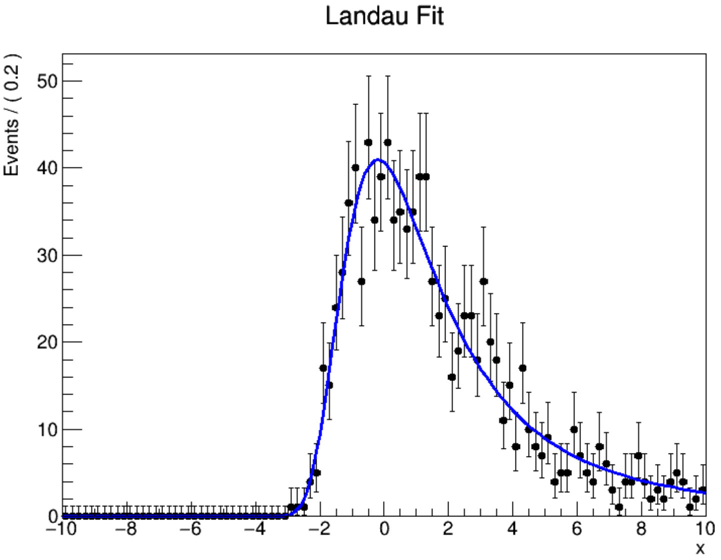

Figura 3: Exercício2 b)

```python
from ROOT import RooArgList
from ROOT import RooArgSet
from ROOT import RooAddPdf
from ROOT import RooPlot
from ROOT import TLegend
from ROOT import RooFit
from ROOT import TLatex
from ROOT import RooChi2Var
from ROOT import TStyle
from ROOT import TCanvas, TFile, TPaveText, TH1F, TLegend, TTree
from ROOT import gStyle, TGraphErrors, TF1, TGraph, gPad, gRandom
from ROOT import kRed, kBlue
from ROOT import TFitResultPtr, TMatrixD
from ROOT import RooGenericPdf
from ROOT import RooFitResult
from ROOT import RooArgusBG

fin = TFile("DataSet_lowstat.root")
nt = fin.Get("data")

mass = RooRealVar("mass", "mass [GeV]", 2, 6);
nt.Print("v")

frame3 = mass.frame(RooFit.Title(r"$\mu\bar{\mu}$"))
nt.plotOn(frame3)

c3 = TCanvas("exemplo03","exemplo03",800,400)
frame3.Draw()
c3.Draw()



import ROOT
kPurple = ROOT.TColor.GetColor(148, 0, 211)



slope = RooRealVar("slope", "slope", 0.0, -10.0, 10.0)  # Par metro da inclina  o da reta
intercept = RooRealVar("intercept", "intercept", 5, 5, 2.0)  # Par metro da interse  o com o eixo y
```

```
mean_gauss = RooRealVar("mean_gauss", "mean_gauss",3.1, 3, 3.2)
sigma_gauss = RooRealVar("sigma_gauss", "sigma_gauss", 0.1, 0.0, 0.05)
mean_gauss_1 = RooRealVar("mean_gauss_1", "mean_gauss_1",3.5, 3.5, 3.5)
sigma_gauss_1 = RooRealVar("sigma_gauss_1", "sigma_gauss_1", 0.05, 0.01, 0.06)
sinal=RooRealVar("sinal", "sinal", 0.5, 0, 0.3);
sinal_1=RooRealVar("sinal_1", "sinal_1", 0.1, 0, 0.05);
sinal_2=RooRealVar("sinal_2", "sinal_2", 0.2, 0,0.2);

argpar= RooRealVar("argpar","argus shape parameter",5,0,5) ;
argus= RooArgusBG("argus","Argus PDF",mass,5,argpar) ;


linear = RooPolynomial("linear", "linear", mass, RooArgList(intercept, slope))
gauss = RooGaussian("gauss", "gauss", mass, mean_gauss, sigma_gauss)
gauss_1= RooGaussian("gauss_1", "gauss_1", mass, mean_gauss_1, sigma_gauss_1)

# Soma das duas gaussianas e da linear multiplicada pelo  nico  par metro de amplitude
sum_pdf = RooAddPdf("sum_pdf", "Sum of Gaussian and Linear", RooArgList(gauss, gauss_1, argus), RooArgList(sinal,sinal_1,sinal_2))

# Ajusta a soma das duas gaussianas e da linear aos dados
sum_pdf.fitTo(nt, RooFit.Minimizer("Minuit2", "Migrad"), RooFit.PrintLevel(-1))

# Ajustando a gaussiana aos dados
gauss.fitTo(nt, RooFit.Minimizer("Minuit2", "Migrad"), RooFit.PrintLevel(-1))
gauss_1.fitTo(nt, RooFit.Minimizer("Minuit2", "Migrad"), RooFit.PrintLevel(-1))
# Ajustando a reta aos dados
linear.fitTo(nt, RooFit.Minimizer("Minuit2", "Migrad"), RooFit.PrintLevel(-1))
sum_pdf.fitTo(nt, RooFit.Minimizer("Minuit2", "Migrad"), RooFit.PrintLevel(-1))




# Plotando os resultados
frame3 = mass.frame(RooFit.Title(r"$\mu\bar{\mu}$"))
frame3.SetStats(0)  # Desativa o StatBox
nt.plotOn(frame3, RooFit.LineColor(kRed))
#gauss.plotOn(frame3, RooFit.LineColor(ROOT.kBlack))
#gauss_1.plotOn(frame3, RooFit.LineColor(kPurple))
#linear.plotOn(frame3, RooFit.LineColor(ROOT.kGreen))
sum_pdf.plotOn(frame3, RooFit.LineColor(ROOT.kBlue), RooFit.LineStyle(ROOT.kDashed))



frame3.SetStats(0)  # Desativa o StatBox
chi2_value = frame3.chiSquare()
legend = ROOT.TLegend(0.35, 0.5, 0.9, 0.9)
legend.SetFillColor(0)
legend.SetTextSize(0.04)
legend.AddEntry("", "#chi2/ndf = {:.3f}".format(chi2_value), "l")

# Adicionando as entradas   legenda
legend.AddEntry(sum_pdf, "J/ \psi: {:.2f} \pm: {:.2f} GeV, Amplitude: {:.2f}".format(mean_gauss.getVal(), sigma_gauss.getVal(), sinal.getVal()), "l")
legend.AddEntry(sum_pdf, " \psi (2S): {:.2f} \pm: {:.2f} GeV, Amplitude_1: {:.2f}".format(mean_gauss_1.getVal(), sigma_gauss_1.getVal(), sinal_1.getVal()), "l")
legend.AddEntry(sum_pdf, "BackGround: {:.2f}, Amplitude: {:.2f}".format(argpar.getVal(), sinal_2.getVal()), "l")
# Plotando os resultados
#nt.statOn(frame3)
c3 = TCanvas("exemplo03","exemplo03",800,400)
frame3.Draw()
```

```
legend.Draw()
c3.Draw()

sum_pdf.Print("t")
c3.Print("mass.png")
```

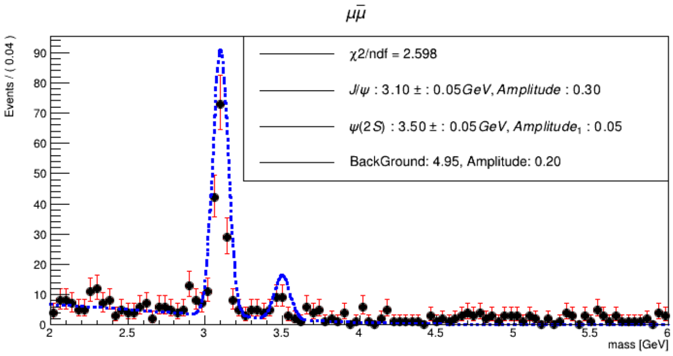

Figura 4: Exercício 3