

INTRODUÇÃO À ANÁLISE DE DADOS EM FÍSICA DE ALTAS ENERGIAS

Exercícios de ROOT

Professores: Dilson de Jesus Damião, Mauricio Thiel e Eliza Melo

Aluno: Thiago Henrique de Sousa

(22/10/2024)

EXERCÍCIO 1

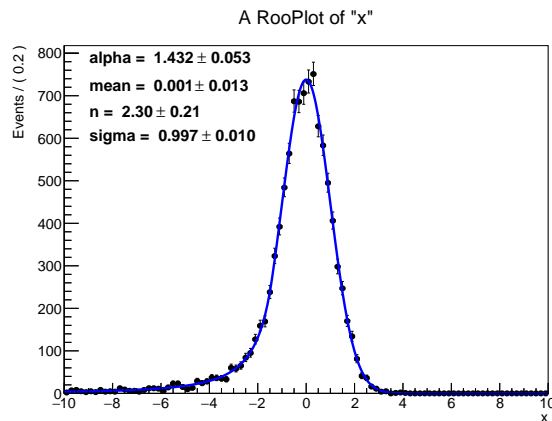
```
1  #include <RooRealVar.h>
2  #include <RooCBSShape.h>
3  #include <RooDataSet.h>
4  #include <RooPlot.h>
5  #include <TCanvas.h>
6  #include <TStyle.h>
7  #include "TH1F.h"
8
9  void exercicio4_1() {
10     // Definir a variável independente (x)
11     RooRealVar x("x", "x", -10, 10);
12
13     // Parâmetros da Crystal Ball
14     RooRealVar mean("mean", "Mean", 0, -10, 10);
15     RooRealVar sigma("sigma", "Sigma", 1, 0.1, 5);
16     RooRealVar alpha("alpha", "Alpha", 1.5, 0.1, 5);
17     RooRealVar n("n", "n", 2, 0.1, 10);
18
19     // Criando função Crystal Ball
20     RooCBSShape crystalBall("crystalBall", "Crystal Ball PDF", x, mean, sigma, alpha, n);
21
22     // Gerar uma amostra de 10.000 dados
23     RooDataSet* data = crystalBall.generate(x, 10000);
24
25     // Ajustar a PDF aos dados
26     crystalBall.fitTo(*data);
27
28     // Criar um frame para plotar a variável x
29     RooPlot* xframe = x.frame();
30
31     // Adicionar os dados e a função ajustada no gráfico
32     data->plotOn(xframe);
33     crystalBall.plotOn(xframe);
34     crystalBall.paramOn(xframe, RooFit::Layout(0.1, 0.95, 0.9));
35
36     TCanvas* c1 = new TCanvas("c1", "Ajuste Crystal Ball", 800, 600);
37     xframe->Draw();
38 }
```

```

39 // Caixa de informações estatísticas
40 gStyle->SetOptStat(222112211);
41 // gStyle->SetOptFit(1111);
42 c1->SaveAs("crystalBall_roofit.pdf");
43 }

```

É possível ver no gráfico os valores das diversas pdfs:



Fonte: O autor.

Exercício 2

```

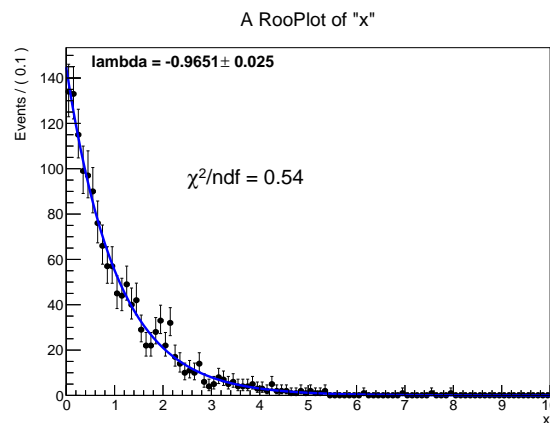
1  #include <RooRealVar.h>
2  #include <RooExponential.h>
3  #include <RooDataSet.h>
4  #include <RooPlot.h>
5  #include <TCanvas.h>
6  #include <TStyle.h>
7
8  void exercicio4_2() {
9      // Definir variável x em um intervalo de 0 a 10
10     RooRealVar x("x", "x", 0, 10);
11
12     // Definir o parâmetro lambda com valor inicial de 1
13     RooRealVar lambda("lambda", "Decay constant", -1, -2, -0.1);
14
15     // Definir a função exponencial
16     RooExponential expo("expo", "Exponential PDF", x, lambda);
17
18     // Gerar 1500 eventos simulados a partir da distribuição exponencial
19     RooDataSet* data = expo.generate(x, 1500);
20
21     // Ajusta lambda e o número de eventos
22     RooFitResult* fitResult = expo.fitTo(*data, RooFit::Save(), RooFit::Extended());
23
24     // Criar um frame para plotar a variável x
25     RooPlot* xframe = x.frame();
26
27     // Plotar os dados e a função ajustada
28     data->plotOn(xframe);

```

```

29     expo.plotOn(xframe);
30     expo.paramOn(xframe, RooFit::Layout(0.1, 0.95, 0.9));
31
32     TCanvas* c1 = new TCanvas("c1", "Ajuste Exponencial", 800, 600);
33     xframe->Draw();
34     gStyle->SetOptStat(222112211);
35     double chi2 = xframe->chiSquare();
36     TLegend *leg = new TLegend(0.1, 0.3, 0.9, 0.9);
37     leg->SetTextSize(0.05);
38     leg->SetBorderSize(0);
39     leg->SetFillStyle(0);
40     leg->AddEntry((TObject*)0, Form("#chi^{2}/ndf = %.2f", chi2), "");
41     leg->Draw();
42
43     c1->SaveAs("exercicio4_2.pdf");
44
45
46     fitResult->Print("v");
47
48     // Mostrar os valores ajustados de lambda e do número total de eventos
49     std::cout << "Lambda ajustado: " << lambda.getVal() << std::endl;
50     // std::cout << "Número total de eventos ajustados: " << fitResult->Ndf() << std::endl;
51 }

```



Fonte: O autor.

Respostas das perguntas da tarefa 2:

- $\lambda = -0.9651 \times 0.025$;
- Foram 1500 eventos ajustados;
- Os valores estão dentro da expectativa, $\lambda < 1$.

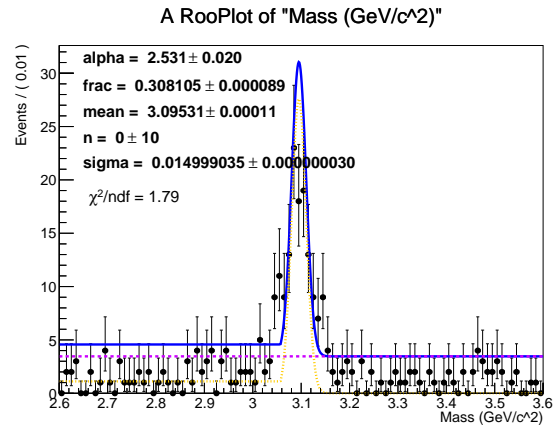
Exercício 3

```
1  #include <RooRealVar.h>
2  #include <RooDataSet.h>
3  #include <RooPlot.h>
4  #include <RooCBShape.h>
5  #include <RooExponential.h>
6  #include <RooPolynomial.h>
7  #include <RooAddPdf.h>
8  #include <RooFitResult.h>
9  #include <TCanvas.h>
10 #include <TAxis.h>
11 #include <TFile.h>
12 #include <TH1.h>
13
14 void ex3() {
15
16     TFile *file = TFile::Open("DataSet_lowstat.root");
17     //TH1F hist = (TH1F)file->Get("data");
18     RooDataSet data = (RooDataSet)file->Get("data");
19
20     RooRealVar mass("mass", "mass", 2, 4);
21
22     //signal
23     RooRealVar mean("mean", "mean", 3.1, 3.05, 3.15);
24     RooRealVar sigma("sigma", "sigma", 0.1, 0.01, 0.15);
25     RooRealVar alpha("alpha", "alpha", 1.5, 0.5, 5);
26     RooRealVar N("N", "N", 2, -10, 10);
27     RooCBShape signal("signal", "signal", mass, mean, sigma, alpha, N);
28
29     // background
30     //RooRealVar lambda("lambda", "lambda", 5, 2, 7);
31     //RooExponential background("background", "background", mass, lambda);
32     RooPolynomial pol("pol", "pol", mass);
33
34
35     RooRealVar frac("frac", "frac", 0.9, 0.8, 1);
36     RooAddPdf model("model", "model", RooArgList(signal, pol), RooArgList(frac));
37     // RooDataHist data("data", "data", RooArgList(mass), hist);
38
39     model.fitTo(*data);
40
41     RooPlot* frame = mass.frame();
42     data->plotOn(frame);
43     model.plotOn(frame);
44     model.plotOn(frame, RooFit::Components("pol"), RooFit::LineStyle(kDashed), RooFit::LineColor(kRed));
45     model.plotOn(frame, RooFit::Components("signal"), RooFit::LineStyle(kDotted), RooFit::LineColor(kBlue));
46
47     model.paramOn(frame, RooFit::Layout(0.1, 0.9, 0.9));
48
49     double chi2 = frame->chiSquare();
50     TCanvas* c = new TCanvas("c", "c", 800, 600);
51     frame->Draw();
52     TLegend *leg = new TLegend(-0.1, 0.2, 0.9, 0.9);
53     leg->SetTextSize(0.04);
54     leg->SetBorderSize(0);
```

```

55     leg->SetFillStyle(0);
56     leg->AddEntry((TObject*)0, Form("#chi^{2}/ndf = %.2f", chi2), "");
57     leg->Draw();
58     std::cout << "Chi^2 / ndf = " << chi2 << std::endl;
59     c->SaveAs("ex3.pdf");
60 }

```



Fonte: O autor.