

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

Name: Bruno Kron Guandalini e Pedro Oliveira

EXERCICIO 1

Crie uma p.d.f Crystall Ball, gere uma amostra de dados a partir dessa pdf e ajuste aos dados, adicione a caixa de informação estatística dos dados e do modelo. Quais foram os valores ajustados para os parâmetros dessa pdf?

RESPOSTA:

```
1  #include "RooRealVar.h"
2  #include "RooCBShape.h"
3  #include "RooDataSet.h"
4  #include "RooFitResult.h"
5  #include "RooPlot.h"
6  #include "TCanvas.h"
7  #include "TLegend.h"
8  #include "RooFit.h"
9  #include "TObject.h"
10 #include "TMath.h"
11
12
13 void generateCrystalBall() {
14
15   RooRealVar cbmean("cbmean", "cbmean" , 90.0, 20, 180.0) ;
16   RooRealVar cbsigma("cbsigma", "cbsigma" , 2.0, 1.0, 40.0) ;
17   RooRealVar cbsig("cbsig", "cbsignal", 10, 0, 1000000);
18   RooRealVar n("n", "", 5.1);
19   RooRealVar alpha("alpha", "", 1.3);
20   RooRealVar x("x", "x", 0, 80, 100);
21   RooCBShape cball("cball", "crystal ball", x, cbmean, cbsigma, alpha, n);
22
23
24   RooDataSet* data = cball.generate(RooArgSet(x), 10000);
25   RooFitResult* fit_result = cball.fitTo(*data, RooFit::Save());
26   fit_result->Print("v");
27
28   TCanvas* c1 = new TCanvas("c1", "Crystal Ball Event Generation", 800,600);
29
30
31   RooPlot* frame = x.frame();
32
33   data->plotOn(frame);
34   cball.plotOn(frame);
35
36   frame->Draw();
37
38
39   double chi2 = frame->chiSquare();
40   TLegend *leg = new TLegend(0.6, 0.7, 0.9, 0.9);
41   leg->SetTextSize(0.03);
```

```

42 leg->SetBorderSize(0);
43 leg->SetFillStyle(0);
44 leg->AddEntry((TObject*)0, Form("Mean = %.3f #pm %.3f",cbmean.getVal(), cbmean.
    getError()), "");
45 leg->AddEntry((TObject*)0, Form("Sigma = %.3f #pm %.3f",cbsigma.getVal(), cbsigma.
    getError()), "");
46 leg->AddEntry((TObject*)0, Form("#chi^2/ndf = %.2f", chi2), "");
47 leg->Draw();
48
49
50 c1->Draw();
51
52 c1->SaveAs("CrystalBall.pdf");
53 }

```

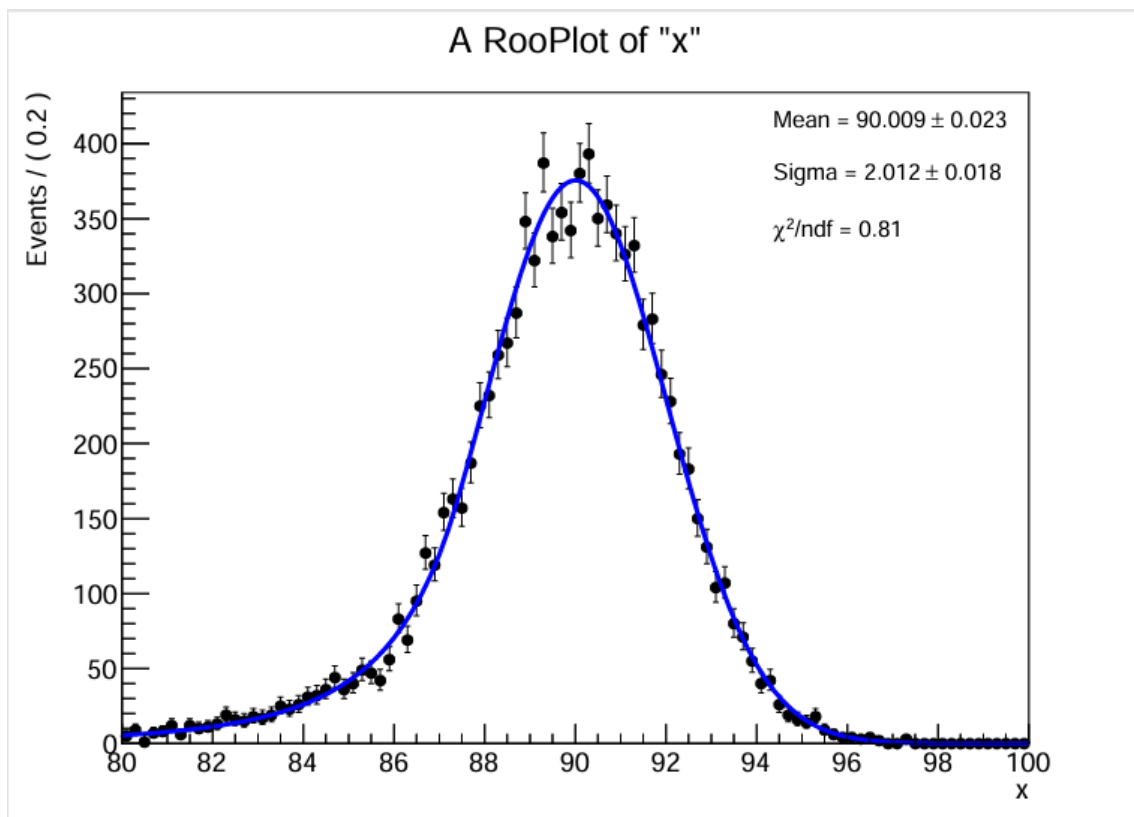


Figura 1: Crystalball

EXERCICIO 2

Defina uma variável contínua x no intervalo de 0 a 10. Defina uma função exponencial decrescente ($\exp(-\lambda x)$), onde λ é um parâmetro inicial com valor de 1 e limites de ajuste entre 0.1 e 2. Gere 1500 eventos simulados a partir dessa distribuição exponencial. Realize um ajuste estendido da função exponencial, ajustando tanto o parâmetro λ quanto o número total de eventos observados. Visualize o ajuste e exiba os resultados ajustados para λ e o rendimento total (número de eventos).

Responda: Qual é o valor ajustado para o parâmetro λ ? Qual é o número total de eventos de eventos ajustados? Compare os valores ajustados com os valores gerados. Eles estão dentro das expectativas?

RESPOSTA:

```

1 #include "TCanvas.h"
2 #include "RooRealVar.h"
3 #include "RooGaussian.h"
4 #include "RooPlot.h"

```

```

5  #include "RooFit.h"
6  #include "TH1.h"
7  #include "RooDataSet.h"
8  #include "RooArgSet.h"
9  #include "RooExponential.h"
10
11
12
13 void lambda(){
14     RooRealVar x("x","x",0,10);
15     RooRealVar lambda("lambda","lambda",-1,-2,-0.1);
16     RooExponential expo("expo", "Decaimento de Lambda",x,lambda);
17     RooRealVar N("N", "Numero de eventos", 1500, 0, 2000);
18
19     RooDataSet* data = expo.generate(RooArgSet(x), 1500);
20
21     RooExtendPdf extExp("extExp", "Extended", expo, N);
22
23     extExp.fitTo(*data ,RooFit::Save());
24
25     RooPlot* frame = x.frame();
26     data->plotOn(frame);
27     extExp.plotOn(frame);
28
29     RooFitResult* fitResult = extExp.fitTo(*data, RooFit::Save());
30
31
32     TCanvas *c =new TCanvas("c","Ajuste Exponencial",800,600);
33
34     frame->Draw();
35
36
37     double chi2 = frame->chiSquare();
38     TLegend *leg = new TLegend(0.1,0.7,0.48,0.9);
39     leg->SetTextSize(0.03);
40     leg->SetBorderSize(0);
41     leg->SetFillStyle(0);
42     leg->AddEntry((TObject*)0, Form("Lambda = %.3f #pm %.3f",lambda.getVal(), lambda.
43         getError()), "");
44     leg->AddEntry((TObject*)0, Form("Eventos ajustados = %.1f #pm %.1f",N.getVal(),N.
45         getError()), "");
46     leg->AddEntry((TObject*)0, Form("#chi^2/ndf = %.2f", chi2), "");
47     leg->Draw();
48
49     c->Draw();
50     c->SaveAs("Lambda.png");
51 }

```

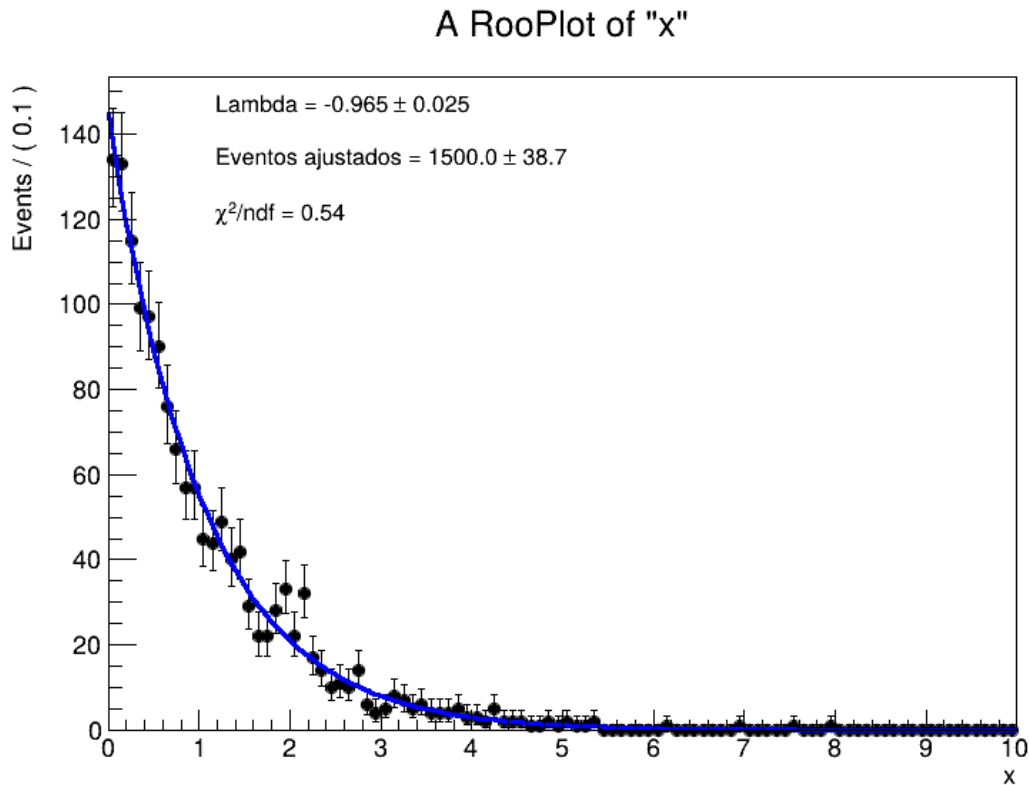


Figura 2: Lambda

EXERCICIO 3

Construa um modelo (sinal + background) e ajuste aos dados para a distribuição de massa da ressonância J/ψ . Utilize o arquivo que se encontra no link: <https://cernbox.cern.ch/index.php/s/DInqlmV9W52WPvY>. Faça o teste estatístico do seu ajuste, calculando o χ^2/ndf .

Use o método `paramOn()` para adicionar os parâmetros ajustados ao gráfico.

Informação: pico da massa do J/ψ 3,096916 GeV/c²

- a PDF do sinal no pico do J/ψ com uma função Crystal Ball
- a PDF do Background com uma polinomial

RESPOSTA:

```

1  #include <RooRealVar.h>
2  #include <RooDataSet.h>
3  #include <RooPlot.h>
4  #include <RooCBSShape.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 SinalBackground() {
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   RooRealVar mean("mean", "mean", 3.1, 3.05, 3.15);
23   RooRealVar sigma("sigma", "sigma", 0.1, 0.01, 0.15);
24   RooRealVar alpha("alpha", "alpha", 1.5, 0.5, 5);
25   RooRealVar N("N", "N", 2, -10, 10);
26   RooCBShape signal("signal", "signal", mass, mean, sigma, alpha, N);
27
28   RooRealVar lambda("lambda", "lambda", 5, 2, 7);
29   RooExponential background("background", "background", mass, lambda);
30   RooPolynomial pol("pol", "pol", mass);
31
32
33   RooRealVar frac("frac", "frac", 0.9, 0.8, 1);
34   RooAddPdf model("model", "model", RooArgList(signal, pol), RooArgList(frac));
35
36
37   model.fitTo(*data);
38
39   RooPlot* frame = mass.frame();
40   data->plotOn(frame);
41   model.plotOn(frame);
42   model.plotOn(frame, RooFit::Components("pol"), RooFit::LineStyle(kDashed), RooFit
       ::LineColor(kRed));
43   model.plotOn(frame, RooFit::Components("signal"), RooFit::LineStyle(kDotted),
       RooFit::LineColor(kGreen));
44
45   model.paramOn(frame, RooFit::Layout(0.1, 0.9, 0.9));
46
47   double chi2 = frame->chiSquare();
48   TCanvas* c = new TCanvas("c", "c", 800, 600);
49   frame->Draw();
50   TLegend *leg = new TLegend(-0.1, 0.2, 0.9, 0.9);
51   leg->SetTextSize(0.04);
52   leg->SetBorderSize(0);
53   leg->SetFillStyle(0);
54   leg->AddEntry((TObject*)0, Form("#chi^{2}/ndf = %.2f", chi2), "");
55   leg->Draw();
56   std::cout << "Chi^2 / ndf = " << chi2 << std::endl;
57   c->SaveAs("BackGround.pdf");
58
59 }

```

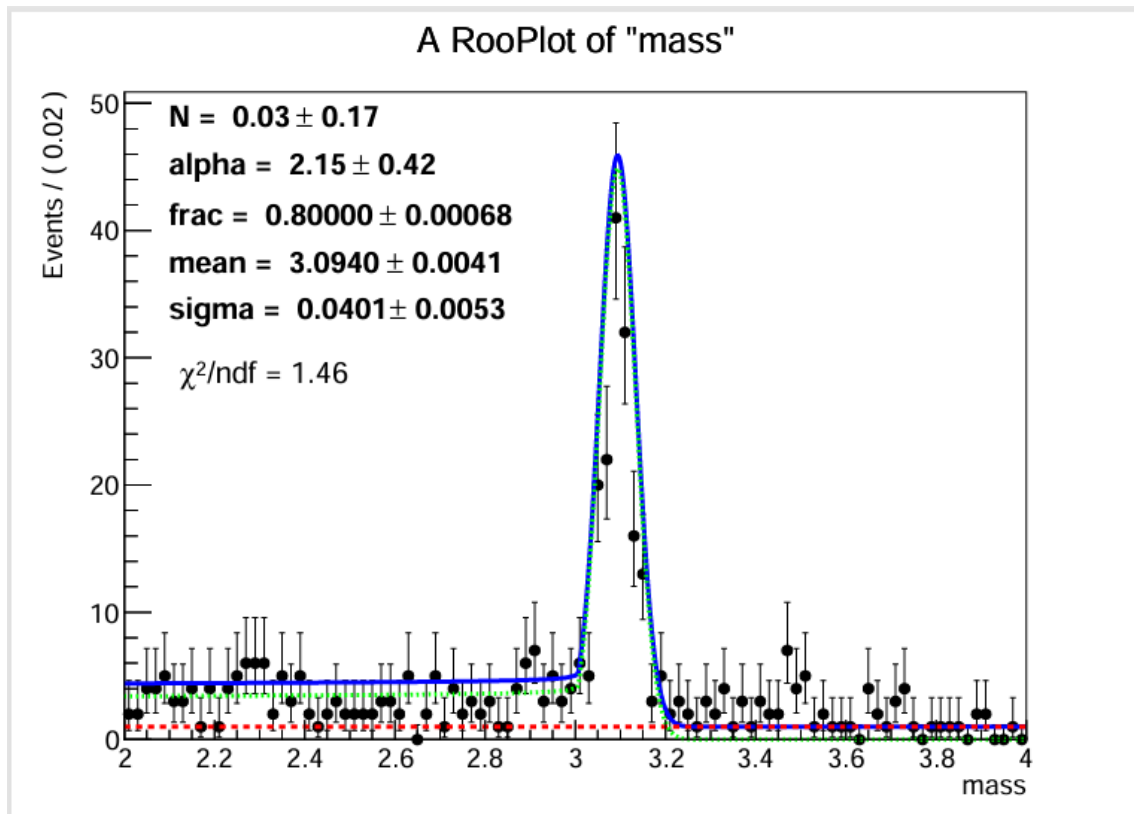


Figura 3: Background