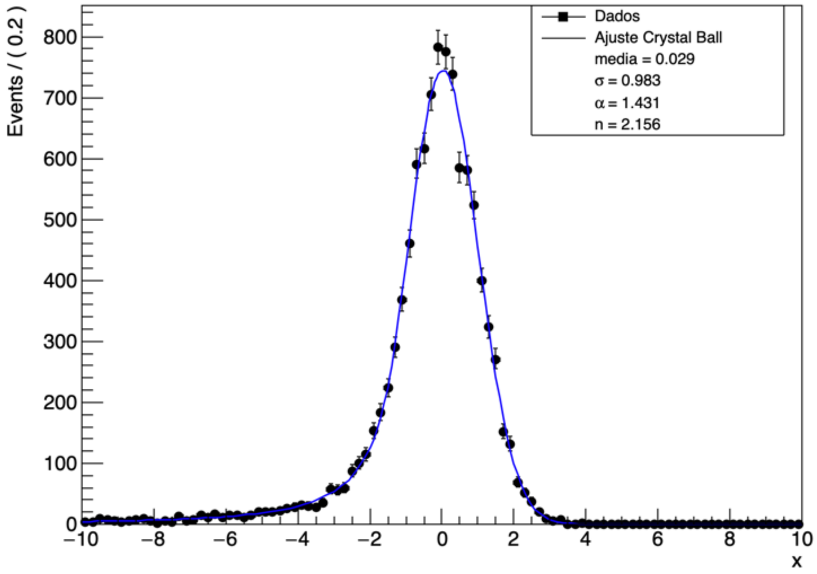
Introdução à Análise de dados em FAE **(13/10/2024)**

Exercicios de estatística para análise de dados em HEP

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

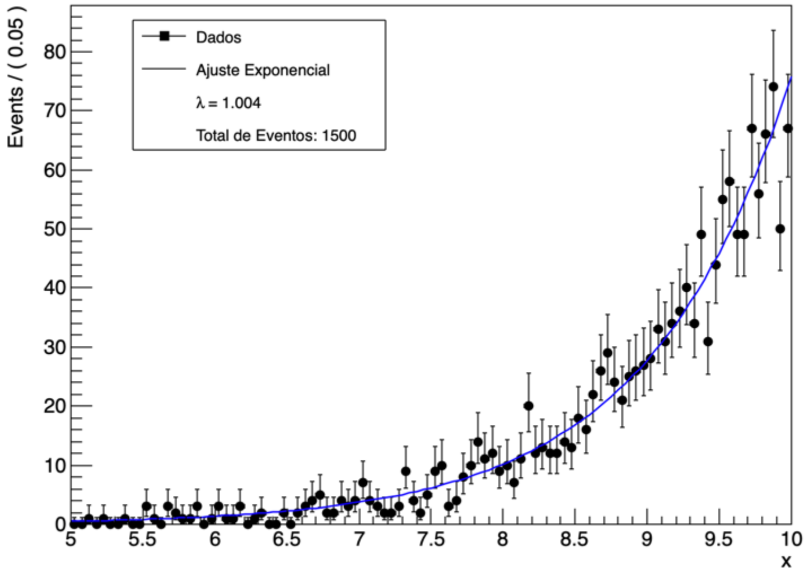
EXERCÍCIO 1

```
#include "RooRealVar.h"
#include "RooCBShape.h"
#include "RooDataSet.h"
#include "RooFitResult.h"
#include "RooPlot.h"
#include "RooAddPdf.h"
#include "TCanvas.h"
#include "TRandom3.h"

void exercicio_1() {

    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, 10);
    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);
    RooFitResult* fitResult = crystalball.fitTo(*data, RooFit::Save());
    RooPlot* xframe = x.frame(RooFit::Title("Ajuste da pdf Crystal Ball aos Dados"));
    data->plotOn(xframe);
    crystalball.plotOn(xframe);

    fitResult->Print();

    TLegend *leg = new TLegend(0.6, 0.7, 0.88, 0.9);
    leg->AddEntry(xframe->findObject("data"), "Dados", "PL");
    leg->AddEntry(xframe->findObject("crystalball"), "Ajuste Crystal Ball", "l");
    leg->AddEntry((TObject*)0, Form("media = %.3f", mean.getVal()), "");
    leg->AddEntry((TObject*)0, Form("#sigma = %.3f", sigma.getVal()), "");
    leg->AddEntry((TObject*)0, Form("#alpha = %.3f", alpha.getVal()), "");
    leg->AddEntry((TObject*)0, Form("n = %.3f", n.getVal()), "");

    xframe->SetTitle("");

    TCanvas* c1 = new TCanvas("c1", "Ajuste da pdf Crystal Ball", 800, 600);
    xframe->Draw();
    leg->Draw("SAME");
    c1->SaveAs("crystalball_fit.png");
}
```



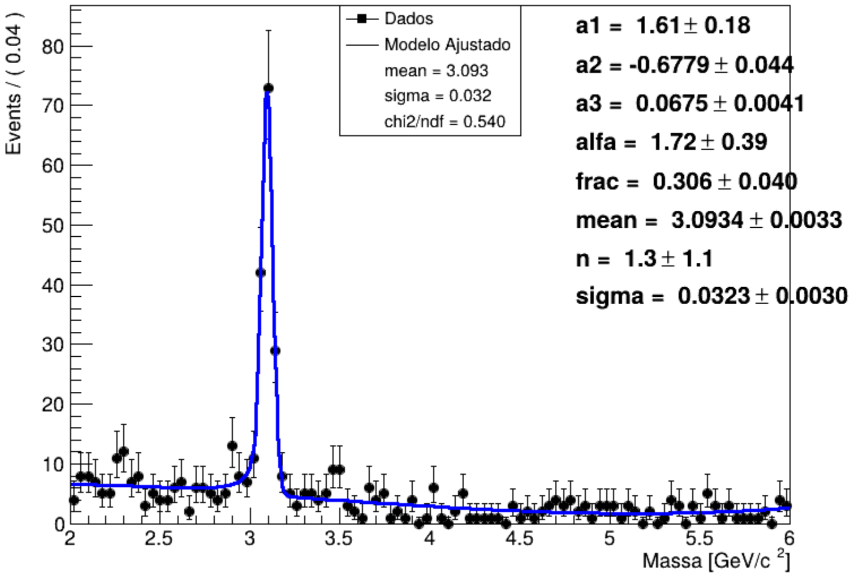
EXERCÍCIO 2

```cpp
    #include "RooRealVar.h"
#include "RooExponential.h"
#include "RooDataSet.h"
#include "RooExtendPdf.h"
#include "RooPlot.h"
#include "TCanvas.h"
#include "RooFitResult.h"

void exercicio_2() {

    RooRealVar x("x", "x", 5, 10);
    RooRealVar lambda("lambda", "lambda", 1, 0.1, 2);
    RooExponential exp("exp", "Exponential PDF", x, lambda);

    RooDataSet* data = exp.generate(x, 10000);
    RooFitResult* fitResult = exp.fitTo(*data, RooFit::Save());

    RooPlot* xframe = x.frame(RooFit::Title(""));
    data->plotOn(xframe);
    exp.plotOn(xframe);
    TCanvas* c = new TCanvas("c", "Ajuste Exponencial", 800, 600);
    xframe->Draw();

    fitResult->Print();
    c->SaveAs("exponential_fit.png");
}
```



EXERCÍCIO 3

```
    #include <iostream>
#include "TFile.h"
#include "TH1.h"
#include "TCanvas.h"
#include "RooRealVar.h"
#include "RooDataSet.h"
#include "RooAddPdf.h"
#include "RooCBShape.h"
#include "RooPolynomial.h"
#include "RooPlot.h"
#include "RooFitResult.h"
#include "RooFit.h"

void exercicio_3() {
    TFile *file = TFile::Open("DataSet_lowstat.root");
    RooDataSet *data = (RooDataSet*)file->Get("data");

    RooRealVar mass("mass", "Massa [GeV/c^{2}]", 2, 6);

    RooRealVar mean("mean", "mean", 3.1, 2.8, 3.2);
    RooRealVar sigma("sigma", "sigma", 0.3, 0.0001, 1.);
    RooRealVar alfa("alfa", "alfa", 1.5, -5., 5.);
    RooRealVar n("n", "n", 1.5, 0.5, 5.);
    RooCBShape CB("CB", "CB", mass, mean, sigma, alfa, n);

    RooRealVar a1("a1", "a1", -0.7, -2., 2.);
    RooRealVar a2("a2", "a2", 0.3, -2., 2.);
    RooRealVar a3("a3", "a3", -0.03, -2., 2.);
    RooPolynomial background("background", "The background PDF", mass, RooArgList(a1, a2, a3));

    RooRealVar frac("frac", "frac", 0.5, 0.0, 1.0);
    RooAddPdf model("model", "Modelo Sinal + Fundo", RooArgList(CB, background), RooArgList(frac));

    RooFitResult *fitResult = model.fitTo(*data, RooFit::Save());

    RooPlot *frame = mass.frame();
    data->plotOn(frame);
    model.plotOn(frame);
    model.paramOn(frame);

    double chi2 = frame->chiSquare();
    int ndf = data->numEntries() - fitResult->floatParsFinal().getSize();

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

    TLegend *leg = new TLegend(0.4, 0.7, 0.6, 0.9);
    leg->AddEntry(frame->findObject("data"), "Dados", "PL");
    leg->AddEntry(frame->findObject("model"), "Modelo Ajustado", "l");
    leg->AddEntry((TObject*)0, Form("mean = %.3f", mean.getVal()), "");
    leg->AddEntry((TObject*)0, Form("sigma = %.3f", sigma.getVal()), "");
    leg->AddEntry((TObject*)0, Form("chi2/ndf = %.3f", chi2), "");

    TCanvas *c = new TCanvas("c", "Ajuste JPsi", 800, 600);
    frame->SetTitle("");
    frame->Draw();
    leg->Draw();
    c->SaveAs("fit_result.pdf");
    file->Close();
}
```


Events / (0.04)
Massa [GeV/c ²]
Dados
Modelo Ajustado
mean = 3.093
sigma = 0.032
chi2/ndf = 0.540
a1 = 1.61 ± 0.18
a2 = -0.6779 ± 0.044
a3 = 0.0675 ± 0.0041
alfa = 1.72 ± 0.39
frac = 0.306 ± 0.040
mean = 3.0934 ± 0.0033
n = 1.3 ± 1.1
sigma = 0.0323 ± 0.0030