Introdução à Análise de dados em FAE **(02/11/2024)**

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 <TChain.h>
#include <TTreeReader.h>
#include <TTreeReaderArray.h>
#include <TCanvas.h>
#include <TH1F.h>
#include <TLorentzVector.h>
#include <RooRealVar.h>
#include <RooDataHist.h>
#include <RooFit.h>
#include <RooCrystalBall.h>
#include <RooPlot.h>
#include <RooAddPdf.h>
#include <RooPolynomial.h>
#include <TLegend.h>
#include <TLatex.h>
#include <iostream>
#include <vector>
#include <algorithm>

void analise_2() {
    std::vector<std::string> diretorios = {
        "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/
           UL2016_MiniAODv2_NanoAODv9-v1/100000/*.root",
        "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/
           UL2016_MiniAODv2_NanoAODv9-v1/1010000/*.root",
        "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/
           UL2016_MiniAODv2_NanoAODv9-v1/250000/*.root"
    };

    TChain chain("Events");
    for (const auto& path : diretorios) {
        chain.Add(path.c_str());
    }

    TTreeReader reader(&chain);
    TTreeReaderArray<float> Electron_pt(reader, "Electron_pt");
    TTreeReaderArray<float> Electron_eta(reader, "Electron_eta");
    TTreeReaderArray<float> Electron_phi(reader, "Electron_phi");

    TH1F* hInvariantMassElectron = new TH1F("hInvariantMassElectron", "Massa
       Invariante dos Dois Eletrons de Maior p_{T}; m_{ll} (GeV/c^{2}); Eventos",
       100, 80, 100);
    TCanvas *c2 = new TCanvas("c2", "Ajuste com Crystal Ball e Background Linear",
       800, 600);
```

```
    int eventos_analisados = 0;
    while (reader.Next()) {
        eventos_analisados++;
        if (eventos_analisados % 10000 == 0) {
            std::cout << "Eventos analisados: " << eventos_analisados << std::endl;
        }

        if (Electron_pt.GetSize() >= 2) {
            std::vector<TLorentzVector> electrons;
            for (int i = 0; i < Electron_pt.GetSize(); ++i) {
                if (Electron_pt[i] > 20.0 && fabs(Electron_eta[i]) < 2.4) {
                    TLorentzVector electron;
                    electron.SetPtEtaPhiM(Electron_pt[i], Electron_eta[i],
                        Electron_phi[i], 0.000511);
                    electrons.push_back(electron);
                }
            }
            std::sort(electrons.begin(), electrons.end(), [](const TLorentzVector& a,
                const TLorentzVector& b) {
                return a.Pt() > b.Pt();
            });

            if (electrons.size() >= 2) {
                TLorentzVector invMassElectron = electrons[0] + electrons[1];
                hInvariantMassElectron->Fill(invMassElectron.M());
            }
        }
    }

    // RooFit:
    RooRealVar mass("mass", "m_{e^{+}e^{-}} (GeV/c^{2})", 80, 100);
    RooDataHist dataHist("dataHist", "Dataset a partir do histograma", mass, RooFit::
        Import(*hInvariantMassElectron));

    RooRealVar mean("mean", "M dia", 90.5, 80, 100);
    RooRealVar sigma("sigma", "Desvio padr o", 1.5, 0.5, 10);
    RooRealVar alpha("alpha", "Alpha", 1.5, 0, 5);
    RooRealVar n("n", "n", 1, 0.1, 10);
    RooCrystalBall crystalBall("crystalBall", "Fun   o de Crystal Ball", mass, mean,
        sigma, alpha, n);

    RooRealVar a0("a0", "Coeficiente linear", 0, -10, 10);
    RooPolynomial background("background", "Fundo Linear", mass, RooArgList(a0));

    RooRealVar nsig("nsig", "N mero de eventos de sinal", 500, 0, 10000);
    RooRealVar nbkg("nbkg", "N mero de eventos de background", 200, 0, 5000);
    RooAddPdf model("model", "Modelo Combinado", RooArgList(crystalBall, background),
        RooArgList(nsig, nbkg));

    model.fitTo(dataHist);

    double meanVal = mean.getVal();
    double meanErr = mean.getError();
    double sigmaVal = sigma.getVal();
    double sigmaErr = sigma.getError();

    RooPlot* frame = mass.frame();
    dataHist.plotOn(frame);
    model.plotOn(frame, RooFit::LineColor(kBlue));
    model.plotOn(frame, RooFit::Components("background"), RooFit::LineStyle(kDashed),
        RooFit::LineColor(kRed));
    model.plotOn(frame, RooFit::Components("crystalBall"), RooFit::LineColor(kGreen))
        ;
```

```
57      frame->SetTitle("");
58      frame->Draw();
59
60      TLegend* legend = new TLegend(0.7, 0.6, 0.9, 0.9);
61      legend->SetTextSize(0.03);
62      legend->AddEntry(frame->getObject(0), "Dados", "P");
63      legend->AddEntry(frame->getObject(1), "Modelo Total", "L");
64      legend->AddEntry(frame->getObject(2), "Fundo Linear", "L");
65      legend->AddEntry(frame->getObject(3), "Crystal Ball", "L");
66
67      TLatex* latex = new TLatex();
68      latex->SetNDC();
69      latex->SetTextSize(0.035);
70      latex->DrawLatex(0.12, 0.78, Form("Media = %.3f #pm %.3f GeV/c^{2}", meanVal,
            meanErr));
71      latex->DrawLatex(0.12, 0.73, Form("Sigma = %.3f #pm %.3f GeV/c^{2}", sigmaVal,
            sigmaErr));
72
73      legend->Draw("SAME");
74
75      c2->Update();
76      c2->SaveAs("ajuste_crystal_ball_background.png");
77      c2->SaveAs("ajuste_crystal_ball_background.root");
78
79      delete latex;
80      delete c2;
81      delete hInvariantMassElectron;
82  }
```