Introdução à Análise de dados em FAE **(Data: 17/11/2020)**

AULA 5: RooFit

Professores: Sandro Fonseca, Sheila Mara da Silva, Eliza Melo *Name:* Ana Maria Garcia Trzeciak

Os arquivos usados com a descrição completa do código estão no meu github: https://github.com/AnaTrzeciak/Curso-FAE.git

Exercícios referente à aula de RooFit

Problema 1:

Crie uma PDF Gaussiana, gere alguns dados e fite-a.

Para gerar uma PDF Gaussiana com o RooFit, primeiro definimos a quantidade observável e depois seus parâmetros. Os parâmetros de uma distribuição gaussiana são a média $\bar{x}$ e o desvio padrão σ. Vamos escolher os valores: $\bar{x} = 0$ e $\sigma = 3$.

```
//Observable
RooRealVar x("x","Observable", -10,10);

//Parameters
RooRealVar mean("mean", "B0 Mass",0);
RooRealVar sigma("sigma", "B0 Mass",3);
```

Depois precisamos criar a distribuição gaussiana.

```
//PDF Object
RooGaussian gaus("Gaussian Distribution", "Gaussian Distribution", x, mean, sigma);
```

Por último geramos os eventos e plotamos o gráfico.

```
//Generate Events
RooDataSet* data = gaus.generate(x,10000);


//Plot PDF
RooPlot* xframe = x.frame();
data->plotOn(xframe);
xframe->Draw();
```

O resultado é dado na figura 1.
O fit é feito pelo seguinte comando:

```
//Fit
gaus.fitTo(*data);
```

E precisamos agora plotar o gráfico com o fit, fazendo:

```
//Plot PDF
RooPlot* xframe = x.frame();
data->plotOn(xframe);
gaus.plotOn(xframe);
xframe->Draw();
```

O resultado é dado na figura 2.

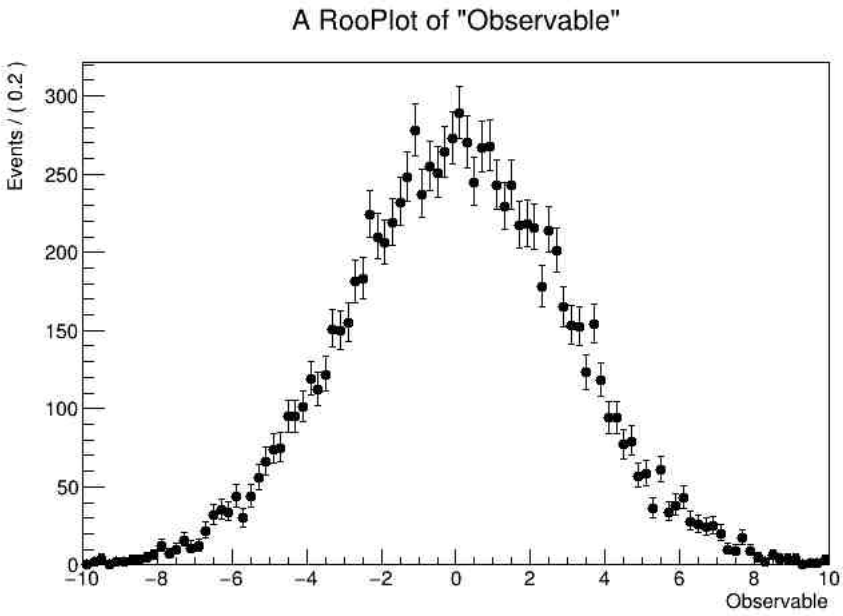

Figura 1: Distribuição gaussiana com $\bar{x} = 0$ e $\sigma = 3$.

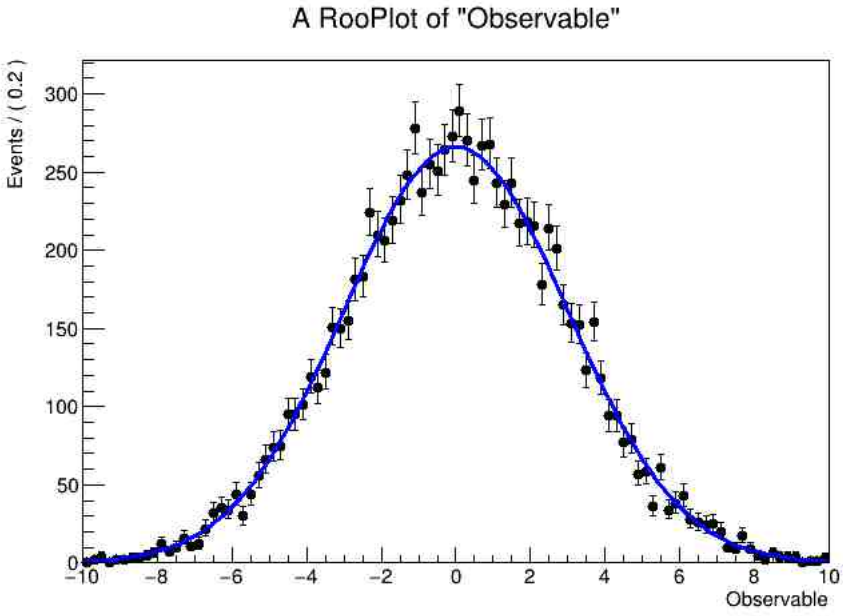

Figura 2: Distribuição e fit de uma distribuição de probabilidade gaussiana com $\bar{x} = 0$ e $\sigma = 3$.

Distribuição da Função Exponencial

Podemos fazer o plot da distribuição exponencial com os mesmos comandos usados acima. No entanto, a distribuição de probabilidade exponencial tem apenas um parâmetro. Eu usei o valor de $\lambda = 0.5$. Sendo assim temos:

```
1 //Observable
2 RooRealVar x("x","x", 0,5);
3
4 //Parameters
5 RooRealVar lambda("lambda", "Parameter",0.5);
6
7 //PDF Object
8 RooExponential expo("Exponential Distribution", "Exponential Distribution", x, lambda
    );
```

```
9
10 //Generate Events
11 RooDataSet* data = expo.generate(x,10000);
12
13 //Fit
14 expo.fitTo(*data);
15
16 //Plot PDF
17 RooPlot* xframe = x.frame();
18 data->plotOn(xframe);
19 expo.plotOn(xframe);
20 xframe->Draw();
21 }
```

O resultado é dado na figura 3.

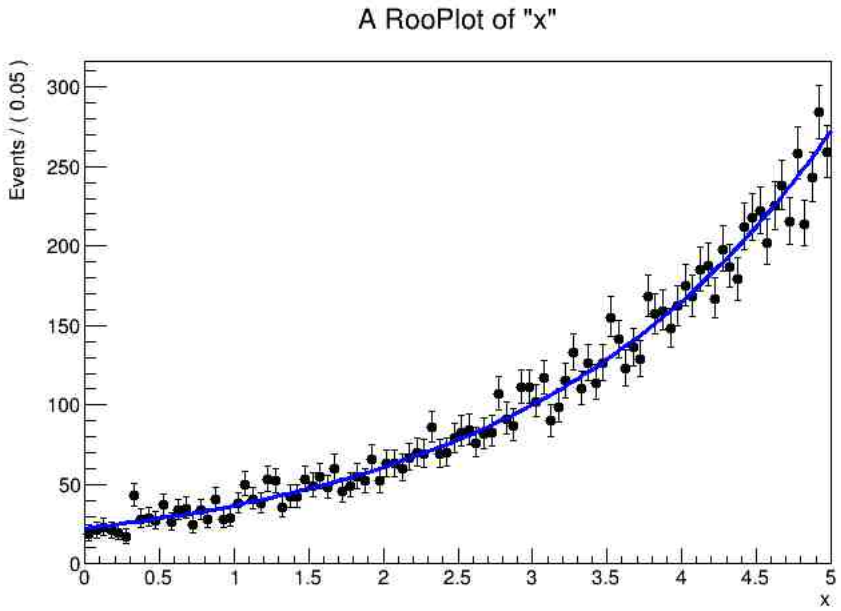

Figura 3: Distribuição exponencial com $\lambda = 0.5$.

Problema 2:

Construa uma PDF para J/Ψ + $\Psi(2S)$ + *background*.

- O sinal de J/Ψ deverá ser parametrizado por uma *Crystal Ball PDF*;
- O sinal de $\Psi(2S)$ deverá ser parametrizado por uma similar *Crystal Ball PDF* (uma PDF Gaussiana);
- O sinal do *background* deverá ser parametrizado por uma *Polynomial PDF*;

Primeiro vamos ler o arquivo de entrada, onde esta os eventos. O arquivo contêm 500 entradas e são eventos que diz respeito a medida da massa invariante de $\mu^+\mu^-$.

```
1 //Reading input file
2 TFile *f = new TFile("DataSet_lowstat.root");
3 RooDataSet *dataset = (RooDataSet*)f->Get("data");
```

Declaramos a nossa quantidade observável, que no caso é a massa invariante.

```
1 //Observable
2 RooRealVar mass("mass", "#mu^{+}#mu^{-} invariant mass", 2.,6., "GeV");
```

Agora vamos construir as PDF's para cada sinal e *background*. Primeiro vamos parametrizar o sinal de J/Ψ por uma *Crystal Ball*, que possui 4 parâmetros.

```
//Build a Crystal Ball PDF for J/Psi
RooRealVar meanJpsi("meanJpsi", "Mean of the Crystal Ball",3.1,2.8,3.2);
RooRealVar sigmaJpsi("sigmaJpsi", "The width of the Crystal ball", 0.3,0.0001,1.);
RooRealVar alphaJpsi("alphaJpsi", "Parameter Alpha", 1.5,-5,5);
RooRealVar nJpsi("nJpsi","Parameter n", 1.5,0.5,5);
RooCBShape CBJpsi("CBJpsi","The Crystal Ball Function", mass, meanJpsi, sigmaJpsi,
    alphaJpsi, nJpsi);
```

Agora o sinal de $\Psi(2S)$. Parametrizado por uma gaussiana, que possui dois parâmetros, a média e o desvio padrão. Vamos usar o desvio padrão σ igual ao definido na Crystal Ball e vamos definir apenas a média.

```
//Build a Gaussian PDF for Psi(2S)
RooRealVar meanPsi2s("meanPsi2s", "Mean of the Crystal Ball",3.7,3.6,3.8);
RooGaussian gaussPsi2s("gaussPsi2s", "Gaussian Distribution for Psi2s", mass,
    meanPsi2s, sigmaJpsi);
```

Por fim vamos parametrizar o *background* por uma PDF polinomial.

```
//Build a polynomial function for background
RooRealVar a1("a1", "Parameter a1 background", -0.7,-2.,2.);
RooRealVar a2("a2", "Parameter a2 background", 0.3,-2.,2.);
RooRealVar a3("a3", "Parameter a3 background", -0.03,-2.,2.);
RooPolynomial bkgPDF("bkgPDF", "Background function polynomial", mass, {a1,a2,a3});
```

Precisamos juntar todas as PDF's em um unico plot. Primeiro definimos o intervalo dos eventos, e logo em seguida unimos todas parametrizações.

```
//Construct a signal and background PDF
RooRealVar Njpsi("Njpsi","Jpsi signal events", 1500,0.1,10000);
RooRealVar Npsi("Npsi","Psi signal events", 100,0.1,5000);
RooRealVar Nbkg("Nbkg","Nbkg signal events", 5000,0.1,50000);
RooAddPdf model("model", "signal + background",RooArgList(CBJpsi,gaussPsi2s,bkgPDF),
    RooArgList(Njpsi,Npsi,Nbkg));
```

Fitamos nossos dados com o comando:

```
//Fit of model
model.fitTo(*dataset);
```

Por último, plotamos nosso o gráfico. A figura 4 mostra o resultado final.

```
//Plotting
RooPlot* xframe = mass.frame(Title("Mass Invariant #mu^{+} #mu^{-}"));
dataset->plotOn(xframe);
model.plotOn(xframe);
model.plotOn(xframe, LineStyle(ELineStyle::kDashed));
xframe->Draw();
```

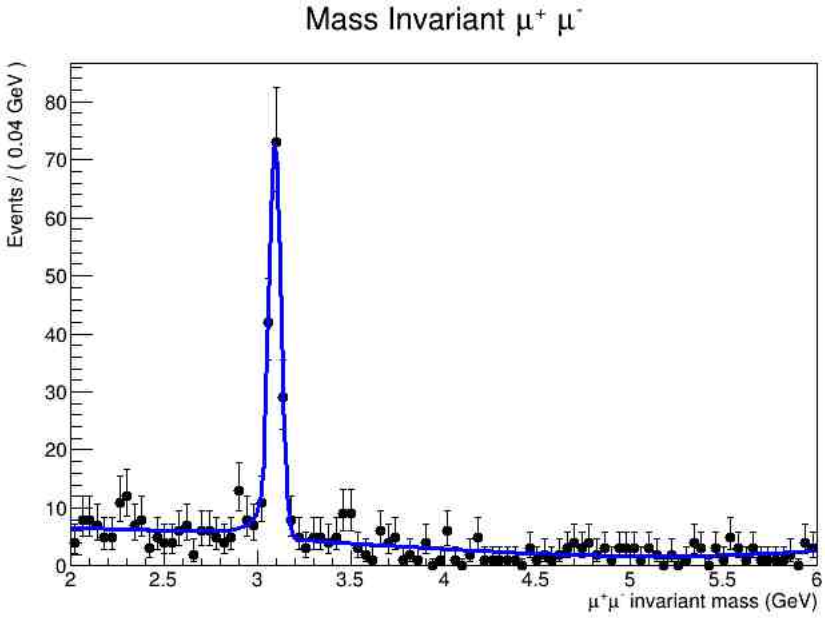

Figura 4: Massa invariante de $\mu^+\mu^-$

.