Introdução à Análise de dados em FAE **(Data: 15/12/20)**

DiMuon Analysis with CMS Open Data

Professores: Sandro Fonseca, Sheila Mara, Eliza Melo *Name:* João Pedro Gomes Pinheiro

Exercício baseado no Tutorial sobre a Análise de DiMuon utilizando dados do CMS Open Data.

Os exercícios referentes ao Tutorial de CMS Open Data ministrado pela professora Sheila se encontram resolvidos e comentados no arquivo `exercises_cms_open_data.ipynb` do repositório `GitHub` Dimuon. O dataset utilizado (`DoubleMuRun2011A.csv`) pode ser encontrado no `GitHub` da disciplina Aula-Analysis-CMS-Opendata.

Os exercícios referentes ao Tutorial ministrado pela professora Eliza se encontram resolvidos e comentados aqui. Os arquivos e programas que serão citados estão disponíveis no repositório `GitHub` Dimuon.

Introdução:

O *dataset* `Skim.root` contém 3 conjuntos de 4-vetores: um conjunto para cada múon (positivo e negativo) e um para o par de múons reconstruídos. Podemos observar a massa invariante do par de múons em escala logarítmica na Figura 1. Podemos observar as várias ressonâncias das partículas que produzem, no seu decaimento, pares de múons. Para nossa análise, vamos escolher a região de 65 a 110 GeV, onde se encontra a ressonância do Bóson Z^0.

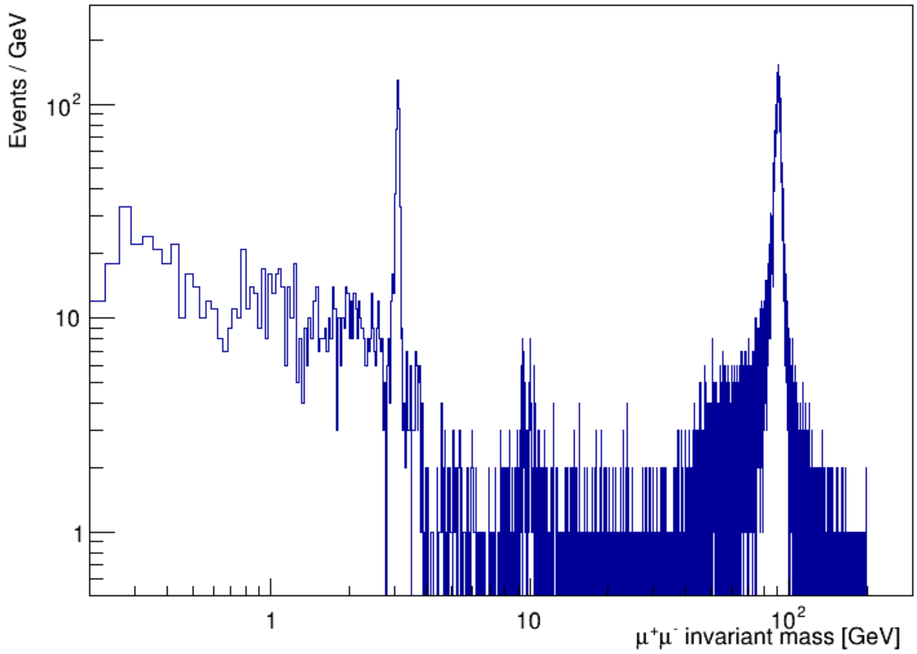

Figura 1: Histograma da massa invariante do par de múons em escala logarítmica.

Corte de Seleção:

Precisamos impor um cortes de seleção no valor do p_T de cada múon reconstruído. Nas Figuras 2 e 3 observamos a distribuição de p_T dos múons positivos e negativos do nosso *dataset*. Como esperamos que os múons que decaíram do Bóson Z^0 tenham maior p_T do que aqueles advindos de ressonâncias de menor massa invariante, **vamos impor um corte de seleção em $p_T > 20$ GeV para cada múon**. Estes plots foram produzidos simplesmente olhando os dados na *ntupla*, a partir do `TBrowser` do `ROOT`.

Na Figura 4, podemos observar o comportamento da massa invariante na região da ressonância do Z^0 quando aplicamos diferentes cortes no p_T de cada múon. Observamos que o corte no p_T de cada múon só se torna sensível nesta região para valores maiores de 20 GeV. A Figura 4 foi feita rodando o programa `dimuon.C` para diferentes cortes de p_T de cada múon ($>$ 0, 10, 20 e 30) GeV e depois desenhamos o histograma apresentado aqui com a ajuda do programa `read_hist.cpp`, presente no repositório `GitHub`.

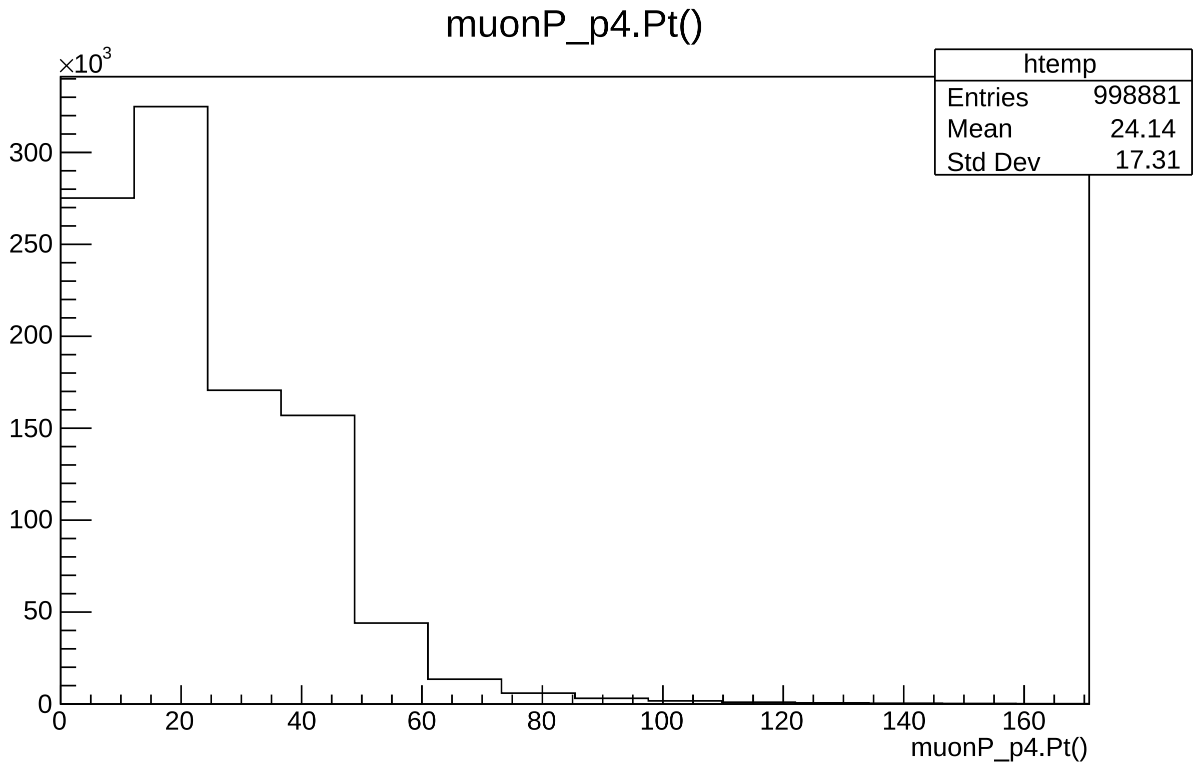

Figura 2: Distribuição de p_T dos múons positivos.

Resultados do Fit:

Com base no que foi discutido no item anterior, agora vamos realizar um ajuste na região de massa invariante entre 65 e 110 GeV, exigindo que cada múon tenha p_T maior que 20 GeV.

Para realizar o ajuste, vamos considerar uma curva gaussiana para o sinal e um ajuste linear para o *background*. O fit global será a soma dessas funções.

Sabemos que a distribuição gaussiana $G(x; A, \sigma, \mu)$ é dada pela Equação 0.1 e uma função linear $L(x; a, b)$ para representar o *background* é dada pela Equação 0.2:

$$G(x; A, \sigma, \mu) = A \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)\right] \tag{0.1}$$

$$L(x; a, b) = ax + b \tag{0.2}$$

Nosso primeiro ajuste global $F_1(x; a, b, A, \sigma, \mu)$, portanto, será dado pela Equação 0.3.

$$F_1(x; a, b, A, \sigma, \mu) = G(x; A, \sigma, \mu) + L(x; a, b) \tag{0.3}$$

Portanto, existem 5 parâmetros que devem ser ajustados, 3 ligados à gaussiana e 2 ligados à função linear, sendo apenas μ e σ os parâmetros de interesse.

No programa que utilizamos, `dimuons.C`, esta implementação é feita a partir das funções declaradas abaixo, sendo o fit global feito a partir da função `fitfun`:

```
Double_t signal(Double_t *x, Double_t *par) {
        //a simple gaussian
        return par[0]*exp(-0.5*TMath::Power(((x[0]-par[1])/(par[2])),2));
}
```

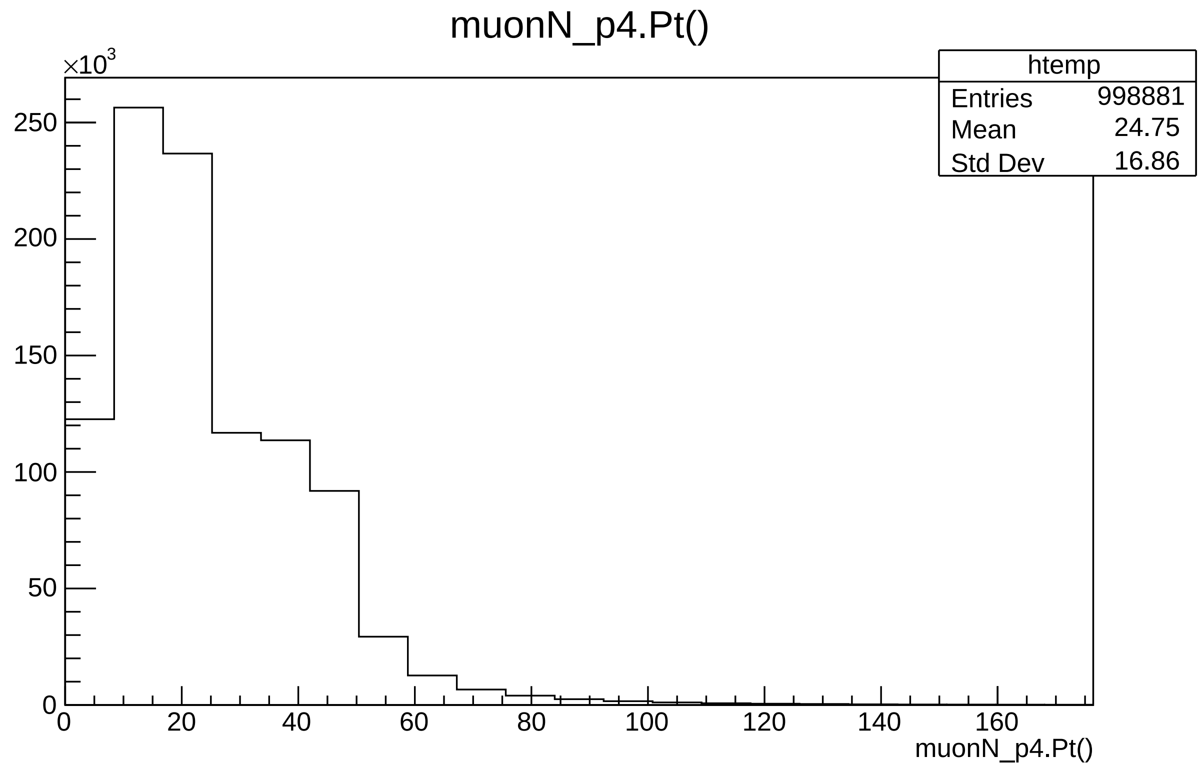

Figura 3: Distribuição de p_T dos múons negativos.

```
Double_t backgr(Double_t *x, Double_t *par) {
        //a simple polynomial
        return par[0]+par[2]*x[0];
}
Double_t fitfun(Double_t *x, Double_t *par) {
        //the total PDF function, sum of the above
        return signal(x,par) + backgr(x,&par[3]);
}
```

O resultado desse fit global pode ser visto na Figura 5. O valor da massa invariante estimada para o Z^0 e seu respectivo erro estatístico associado será:

$$M_Z = 90,731 \text{ GeV}$$

$$\sigma_{stat} = \frac{2,618}{\sqrt{33371}} = 0,014 \text{ GeV}$$

Efeitos Sistemáticos:

Afim de estimar os efeitos sistemáticos na nossa amostra, vamos implementar um modelo diferente para sinal e *background* e repetir o *fit*. O erro sistemático será a diferença entre o valor esperado obtido neste novo ajuste e o obtido do ajuste anterior.

Para o *background*, será mantido o modelo da função linear, dada pela Equação 0.2. Para o sinal, entretanto, utilizaremos a distribuição relativística de Breit-Wigner. Segundo a referência Relativistic Breit-Wigner distribuition, esta função é dada pela Equação 0.4.

$$B(x; M, \Gamma) = \frac{k}{(x^2 - M^2)^2 + M^2\Gamma^2} \tag{0.4}$$

sendo k uma constante de proporcionalidade, dada por: $k = \dfrac{2\sqrt{2}M\Gamma}{\pi\sqrt{M^2+\gamma}}\gamma$, sendo $\gamma = \sqrt{M^2(M^2+\Gamma^2)}$

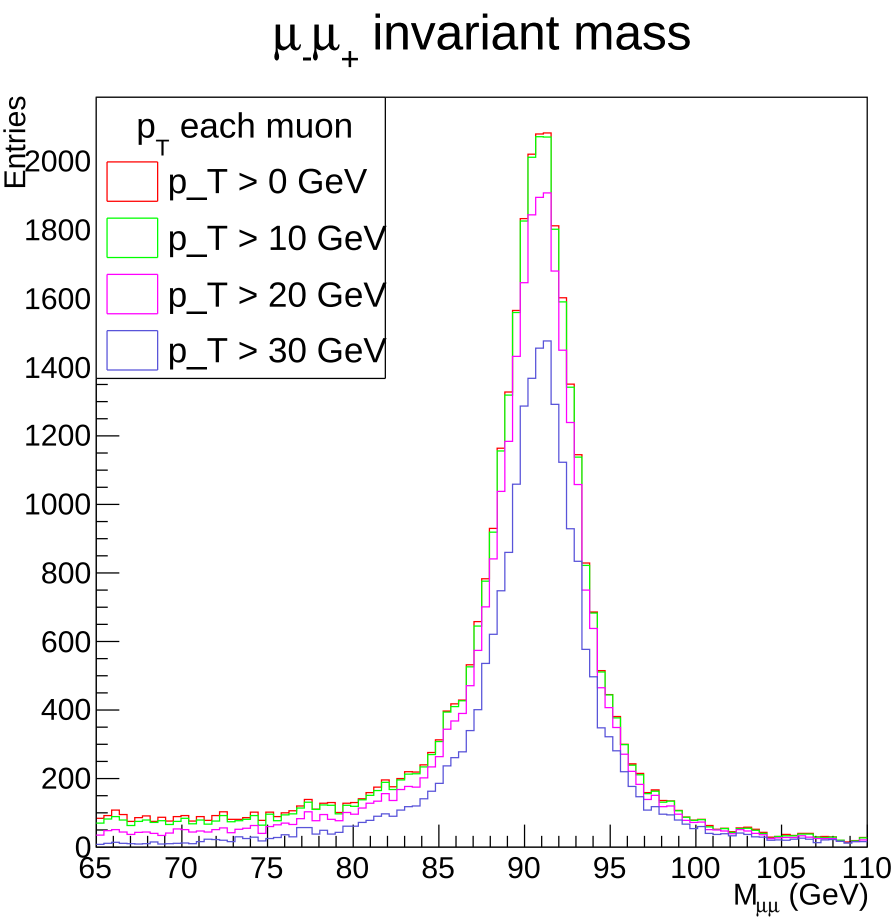

Figura 4: Distribuição de massa invariante do par de múons para diferentes cortes no p_T de cada múon.

O ajuste busca o melhor valor para os parâmetros: valor esperado da massa invariante (M) e largura do decaimento (Γ).

A implementação no código `dimuons.C` é feita com base na referência WorkBook: Fitting a Breit-Wigner, onde a nova distribuição para o sinal é descrita pela função `signal2` e o fit global será feito a partir da soma da função de Breit-Wigner com a função linear, como na Equação 0.5, implementada como a função `fitfun2`.

$$F_1(x; a, b, M, \Gamma) = B(x; M, \Gamma) + L(x; a, b) \tag{0.5}$$

```
Double_t signal2(Double_t *x, Double_t *par) {
        //a Breit-Wigner
        Double_t arg1 = 14.0/22.0; // 2 over pi
        Double_t arg2 = par[2]*par[2]*par[1]*par[1]; //Gamma=par[1]  M=par[2]
        Double_t arg3 = ((x[0]*x[0]) - (par[1]*par[1]))*((x[0]*x[0]) - (par[1]*par[1]));
        Double_t arg4 = x[0]*x[0]*x[0]*x[0]*((par[2]*par[2])/(par[1]*par[1]));
        return par[0]*arg1*arg2/(arg3 + arg4);
}
Double_t fitfun2(Double_t *x, Double_t *par) {
        //the total PDF function, sum of the above
        return signal2(x,par) + backgr(x,&par[3]);
}
```

O resultado deste novo ajuste pode ser visto na Figura 6.

A incerteza sistemática, portanto, será a diferença entre o valor esperado para a massa invariante obtido com os dois fits, ou seja:

$$\sigma_{sis} = |90,731 - 90,885| = 0,154 \text{ GeV}$$

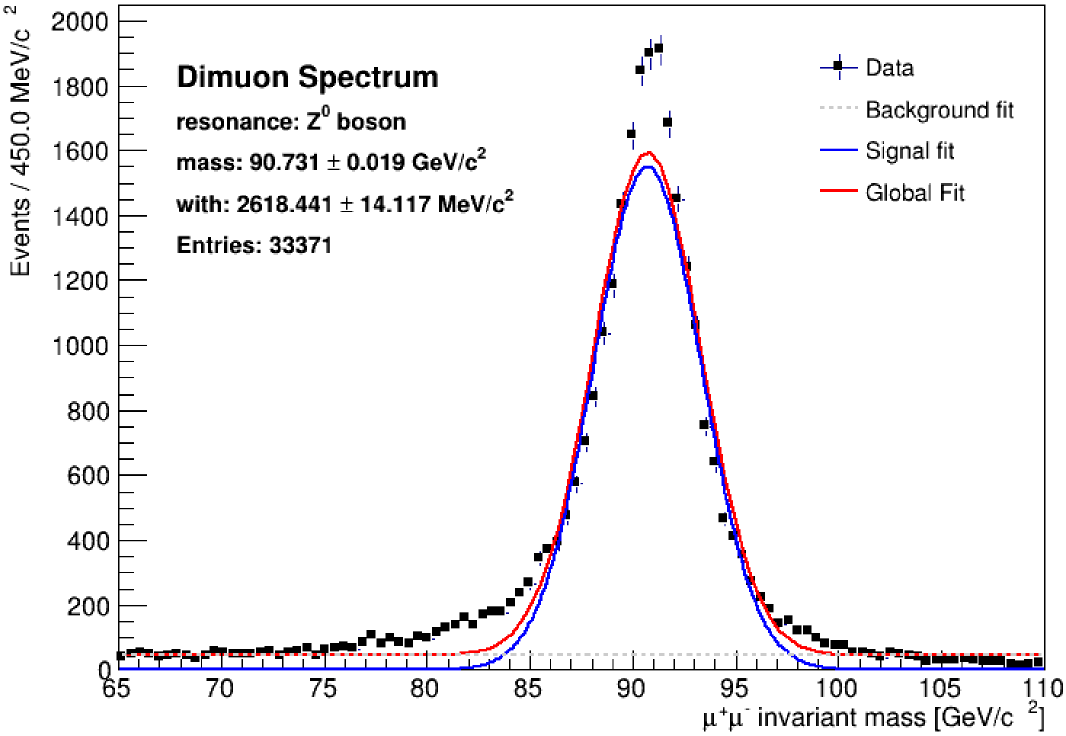

Figura 5: Fit global. Soma de uma gaussiana (sinal) com uma função linear (fundo).

Para efeito de comparação, implementamos um ajuste idêntico utilizando o `Roofit`. O código foi implementado no método `dimuon::FitPeak` e é transcrito abaixo:

```
RooRealVar x("x","x",hmin,hmax);
RooRealVar mean("mean","mean of gaussians",90.,70.,120.);
RooRealVar sigma("sigma","width of gaussians",5.,0.,120.);
RooRealVar width("width","width",5.0, 0.0, 120.0);

// Comment when fit breit-wigner
RooGaussian sig("sig","Signal component",x,mean,sigma);
// Comment when fit gaussian+pol
//RooBreitWigner pdf_sum("pdf_sum","pdf_sum",x,mean,sigma);

// Build Chebychev polynomial p.d.f.
RooRealVar a0("a0","a0",1.0,-1.,1);
RooRealVar a1("a1","a1",0.1,-1.,1);
RooChebychev bkg("bkg","Background",x,RooArgList(a0,a1)) ;

// Sum the components into a composite p.d.f (signal + bkg).
// Comment when run for breit-wigner
RooRealVar sigfrac("sigfrac","fraction of signal component",0.5,0.,1.);
RooAddPdf pdf_sum("pdf_sum","Gauss + pol",RooArgList(sig,bkg),RooArgList(sigfrac));

// The data
RooDataHist dh("dh","dh",x,Import(*hpeak));

// The plot
RooPlot* frame = x.frame(Name("xframe"), Title("Z mass")) ;

dh.plotOn(frame,MarkerColor(2),MarkerSize(0.9),MarkerStyle(21));  //this will show histogram data points on canvas
dh.statOn(frame);  //this will display hist stat on canvas
```

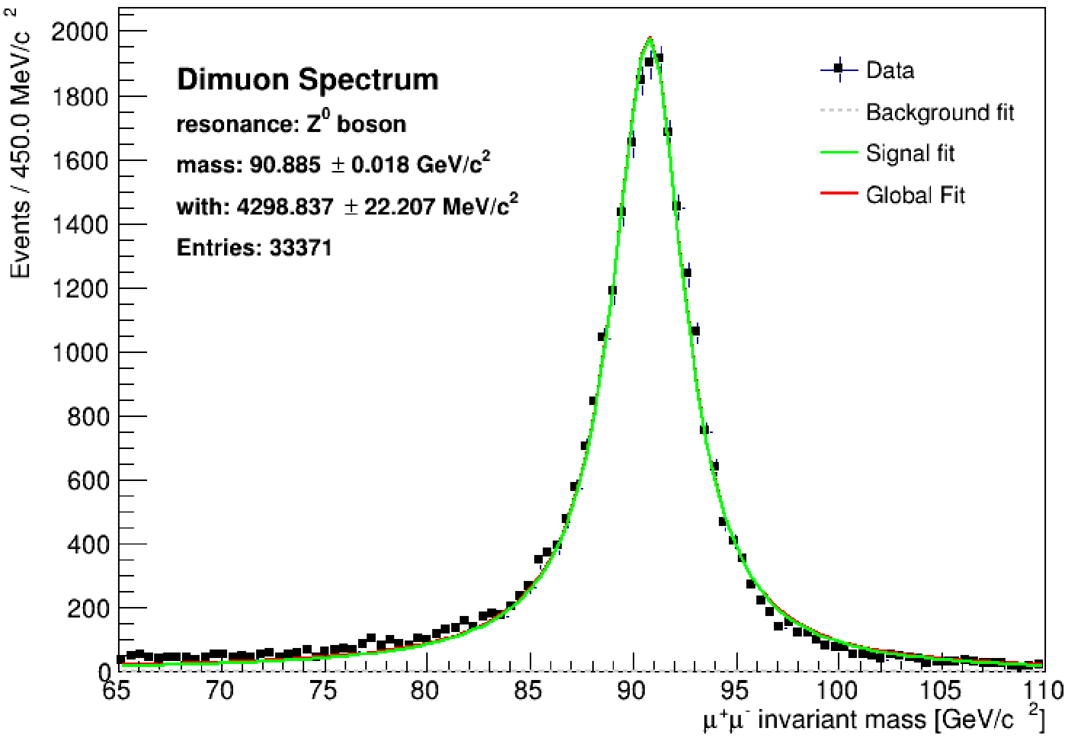

Figura 6: Fit global modelado de uma forma diferente. Soma de uma Breit-Wigner (sinal) com uma função linear (fundo).

```
RooFitResult* filters = pdf_sum.fitTo(dh,"qr"); // fit the pdf_sum to the data
pdf_sum.plotOn(frame);//this will show fit overlay on canvas
//sig.plotOn(frame);
pdf_sum.paramOn(frame,Title("Global Fit parameters"), Layout(0.175, 0.4, 0.9)); // this will display the fit parameters on canvas

pdf_sum.Print("t");// this will print the fit information

// Create canvas and save the results
TCanvas* c = new TCanvas("ZmassHisto","ZmassHisto",800,400) ;
c->cd(); gPad->SetLeftMargin(0.15);
c->SetTitle("Inv Mass with global fit (gauss+pol)");
frame->GetXaxis()->SetTitle("Inv mass (GeV/c^{2})");
frame->GetXaxis()->SetTitleOffset(1.2);
float binsize = hpeak->GetBinWidth(1); char Bsize[50];
frame->Draw();
c->Print("myZmaa.eps");
c->SaveAs("roofit_gauss.png");
c->Close();
c->Delete();
```

Para realizar os diferentes ajustes, tanto o Gauss+pol quanto o Breit-Wigner, basta comentar as linhas indicadas no código. Lembrando que um polinômio de Chebychev de grau 1 é uma função linear. Os resultados podem ser vistos nas Figuras 7 e 8. A informação nova e relevante que obtemos utilizando o `Roofit` é o parâmetro `sigfrac`, que mostra o quanto dos dados nesse intervalo de massa invariante ($65 < M < 110$ GeV) está na região de sinal. A função de Breit-Wigner já modela suficientemente bem os dados (sinal+fundo).

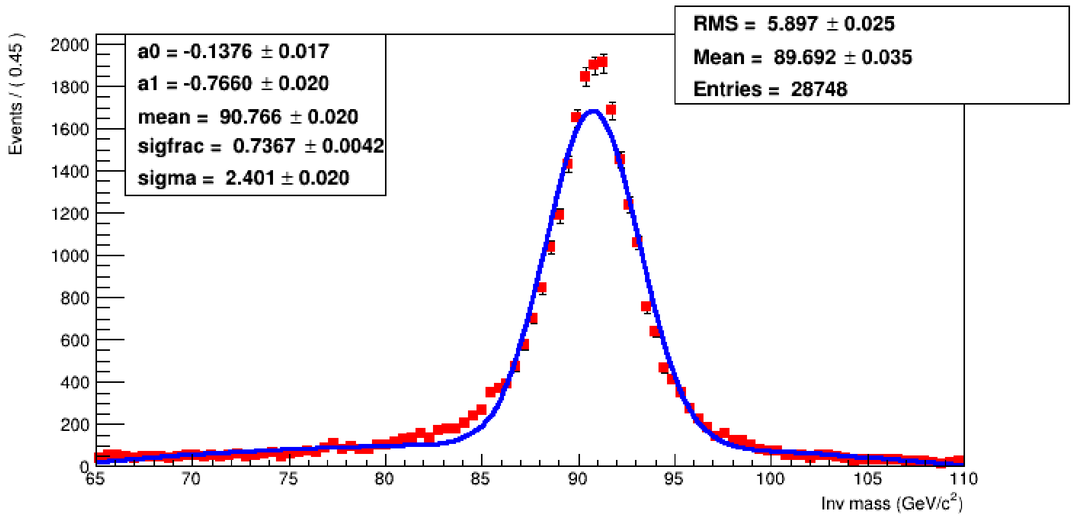

Figura 7: Fit global modelando os dados como a soma Gauss + função linear usando `Roofit`.

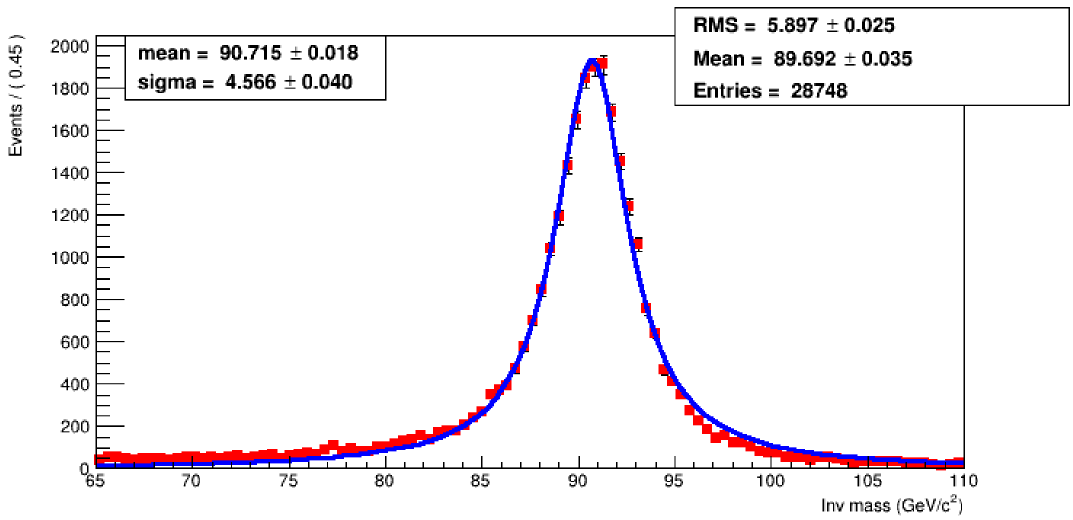

Figura 8: Fit global modelando os dados como uma Breit-Wigner usando `Roofit`.

Efeitos em p_T e η:

Para finalizar, vamos analisar os efeitos na reconstrução da massa invariante para valores distintos de p_T e diferentes regiões de η do par de múons. Estas informações são obtidas no quadrivetor do par de múons (`dimuon_p4`) através de `dimuon_p4.Pt()` e `dimuon_p4.Eta()`.

Para cada evento que passa no corte de seleção, vamos calcular o rendimento na reconstrução da massa em função de p_T e η do sistema $\mu^+\mu^-$. O rendimento será calculado a partir do erro relativo em relação ao valor obtido no primeiro ajuste ($M_Z = 90,731$ GeV), ou seja:

$$\sigma_{rel} \equiv \frac{|M - 90,731|}{90,731}$$

O resultado obtido pode ser visto na Figura 9. Não é possível ver nenhuma correlação entre os valores de p_T ou η com o erro relativo em relação ao valor esperado de massa.

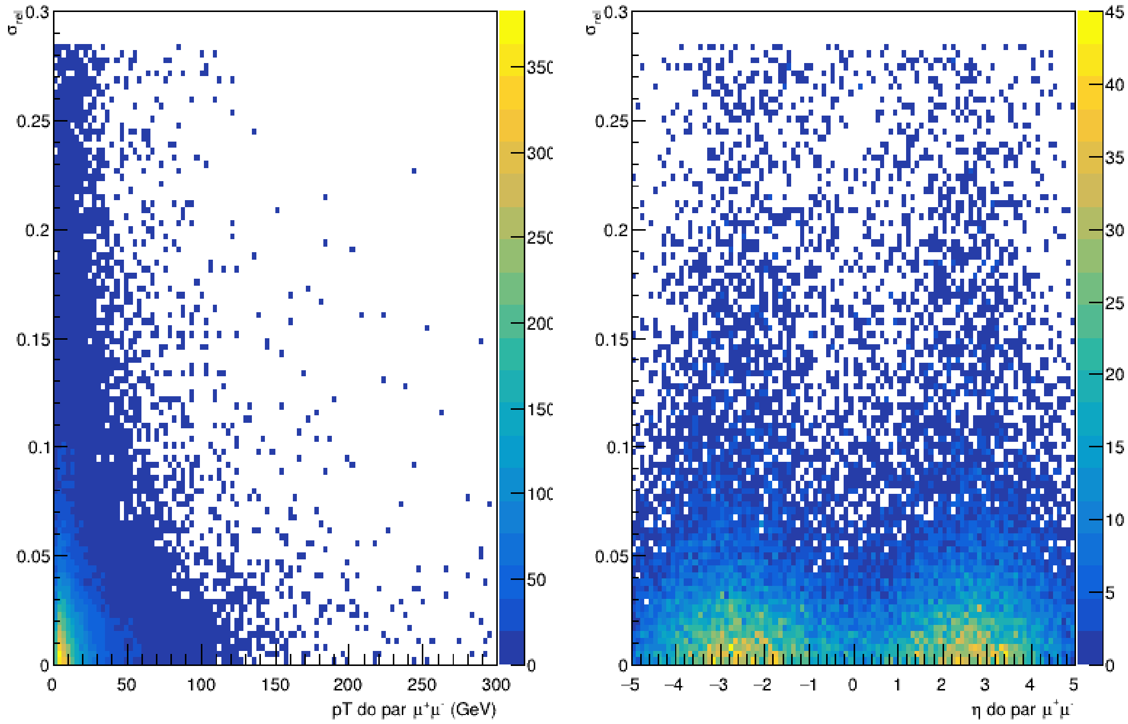

Figura 9: Espaço de fase na região do sinal ($65 < M_{\mu^+\mu^-} < 110$ GeV) em função do erro relativo σ_{rel}.