

AULA 4: ROOT

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 as aulas de ROOT

Problema 1

Revise the `Analyze::Terminate` method in `Analyze.C` to draw the histograms with error bars.

Para plotarmos um histograma-1D com barras de erro, basta apenas adicionar a letra *E1* no argumento da função *Draw* da seguinte forma:

```
1 chi2Hist->Draw("E1");
```

O resultado é dado na figura 1.

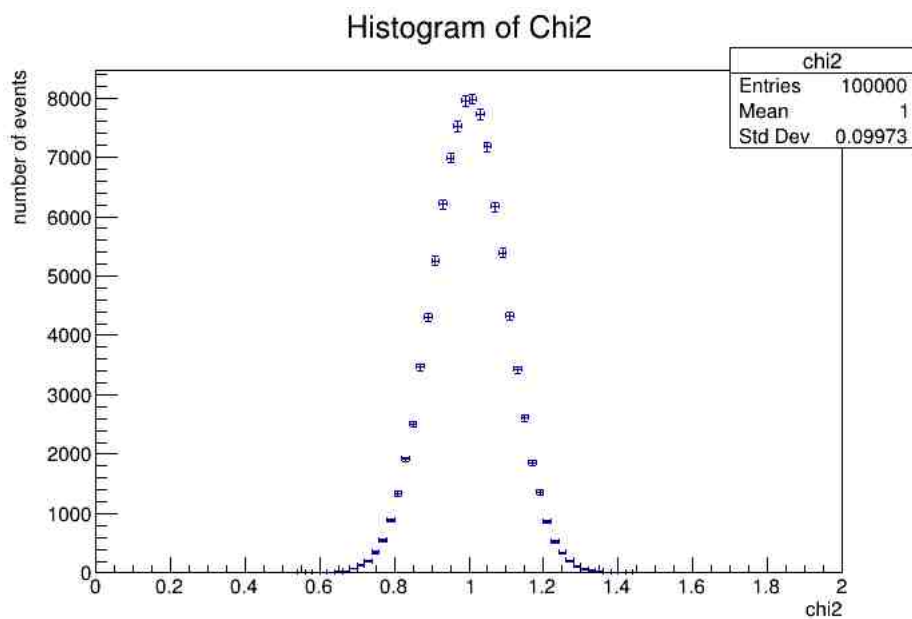


Figura 1: Histograma do χ^2 com barras de erro.

Problema 2

Revise `Analyze.C` to create, fill and display an additional histogram of the variable `ebeam` (with error bars and axis label)

Primeiro criamos o histograma:

```
1 //##### Histogram ebeam #####
2 ebeamHist = new TH1D("ebeamHist", "Histogram of ebeam", 100, 149.0, 151.0);
3 ebeamHist->GetXaxis()->SetTitle("ebeam");
4 ebeamHist->GetYaxis()->SetTitle("number of events");
```

Depois lemos o *branch* e preenchemos o histograma:

```
1 GetEntry(entry);
2 ebeamHist->Fill(*ebeam);
```

E por último plotamos o histograma com barras de erro:

```
1 ebeamHist->Draw("E1");
```

O resultado é dado na figura 2.

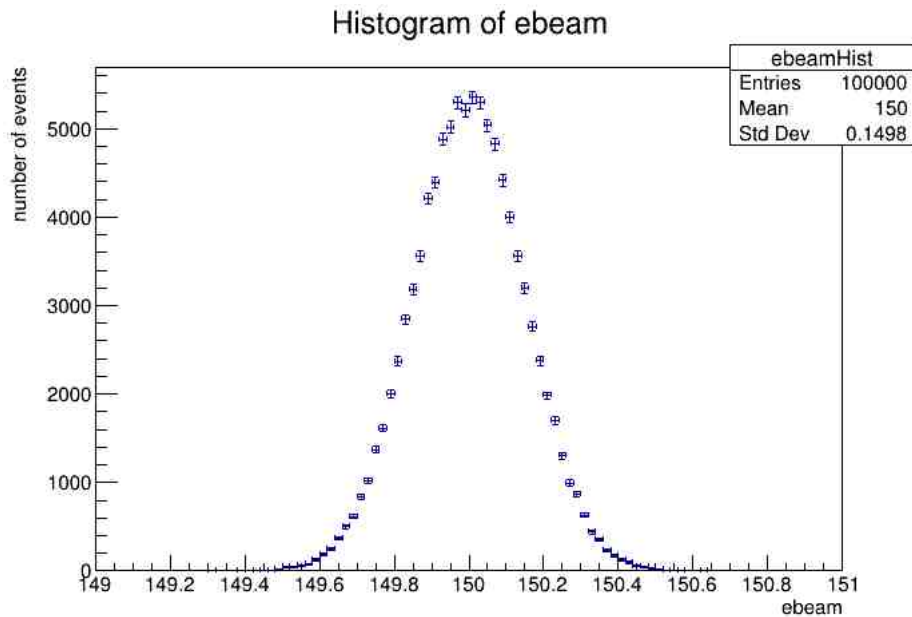


Figura 2: Histograma do *branch ebeam* com barras de erro.

Problema 3

Fit the ebeam histogram to a gaussian distribution.

Para fazer o *fit* do histograma com uma distribuição gaussiana fazemos:

```
1 ebeamHist->Fit("gaus");
2 ebeamHist->Draw("E1");
```

O resultado é dado na figura 3.

Problema 4

Add another plot: a scatterplot of chi2 versus ebeam. Don't forget to label the axes!

Primeiro criamos o *scatterplot* utilizando a classe [TH2](#).

```
1 scattHist = new TH2F("scattHist", "Scatterplot chi2 x ebeam", 100, 149.0, 151.0, 100, 0, 2)
;
2 scattHist->GetYaxis()->SetTitle("chi2");
3 scattHist->GetXaxis()->SetTitle("ebeam");
```

Lemos as duas quantidades:

```
1 GetEntry(entry);
2 scattHist->Fill(*ebeam,*chi2);
```

E plotamos o gráfico:

```
1 scattHist->Draw();
```

O resultado é dado na figura 4.

Problema 5

Calculate p_t in an analysis macro and make a histogram of the variable.

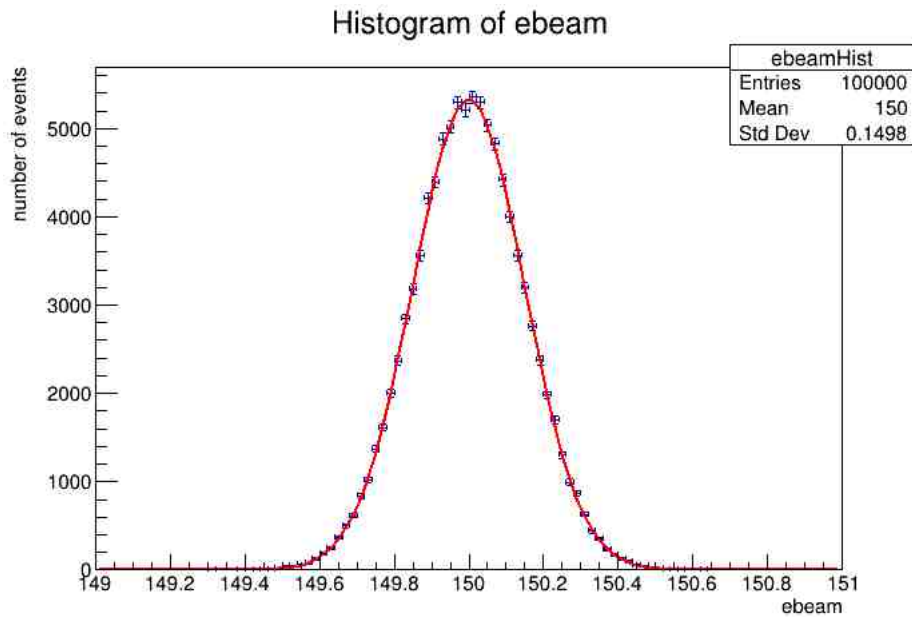


Figura 3: Histograma do *branch ebeam* com barras de erro e *fitado* por uma função gaussiana.

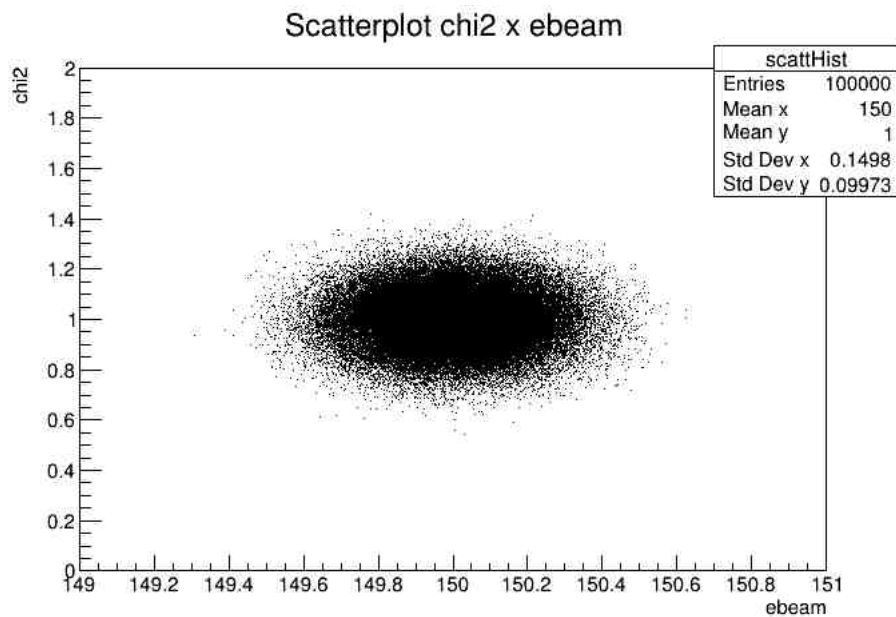


Figura 4: Scatterplot de χ^2 x *ebeam*.

A variável p_t é definida como: $p_t = \sqrt{p_x^2 + p_y^2}$.

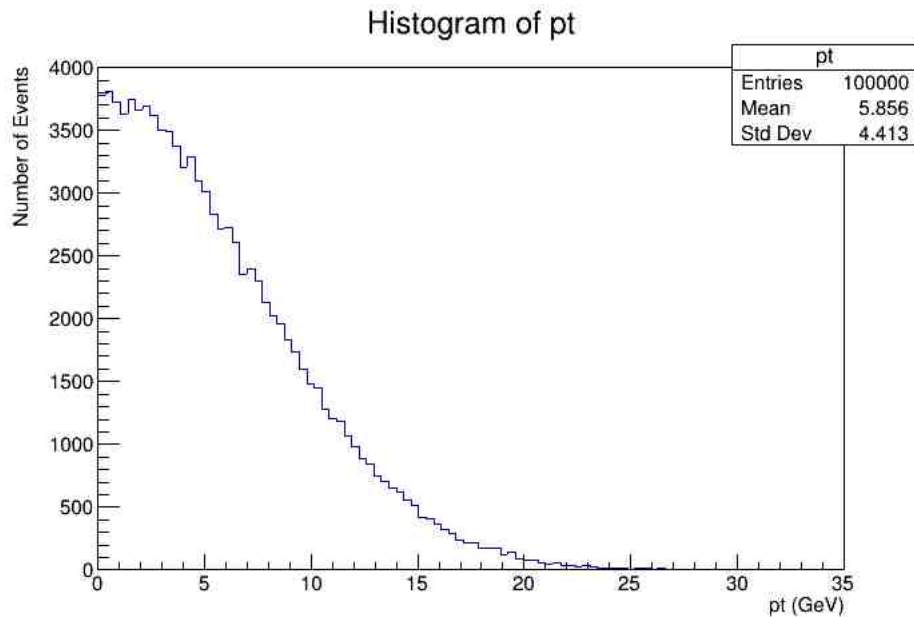
Primeiro criamos o histograma referente à p_t .

```
1 ptHist = new TH1D("pt", "Histogram of pt", 100, 0, 35);
2 ptHist->GetXaxis()->SetTitle("pt (GeV)");
3 ptHist->GetYaxis()->SetTitle("Number of Events");
```

Depois definimos a como calcular p_t e preenchemos o histograma:

```
1 //Fill histogram of pt
2 Double_t pt;
3 pt = TMath::Sqrt((*px)*(*px)+(*py)*(*py));
4 ptHist->Fill(pt);
```

O resultado pode ser visto na figura 5.

Figura 5: Histograma da quantidade p_t .**Problema 6**

Include a histogram of theta, using the math function `TMath::ATan2(y,x)`.

A variável θ é definida como: $\theta = \arctan(p_t/p_z)$ e é o ângulo que o feixe faz com a direção z .

Primeiro criamos o histograma:

```
1  thetaHist = new TH1D("theta", "Histogram of theta", 100, -0.4, 0.4);
2  thetaHist->GetXaxis()->SetTitle("theta");
3  thetaHist->GetYaxis()->SetTitle("Number of Events");
```

Depois definimos θ e preenchemos o histograma:

```
1  Double_t theta;
2  theta = TMath::ATan2((pt),(*pz));
3  thetaHist->Fill(theta);
```

O resultado pode ser visto na figura 6.

Problema 7

Apply a cut in your analysis macro. Your goal is to count the number of events for which p_z is less than 145 GeV, and then display the value.

Primeiro defini uma variável global que irá contar o número de $p_z < 145$ GeV no início do código, antes de todas as seções.

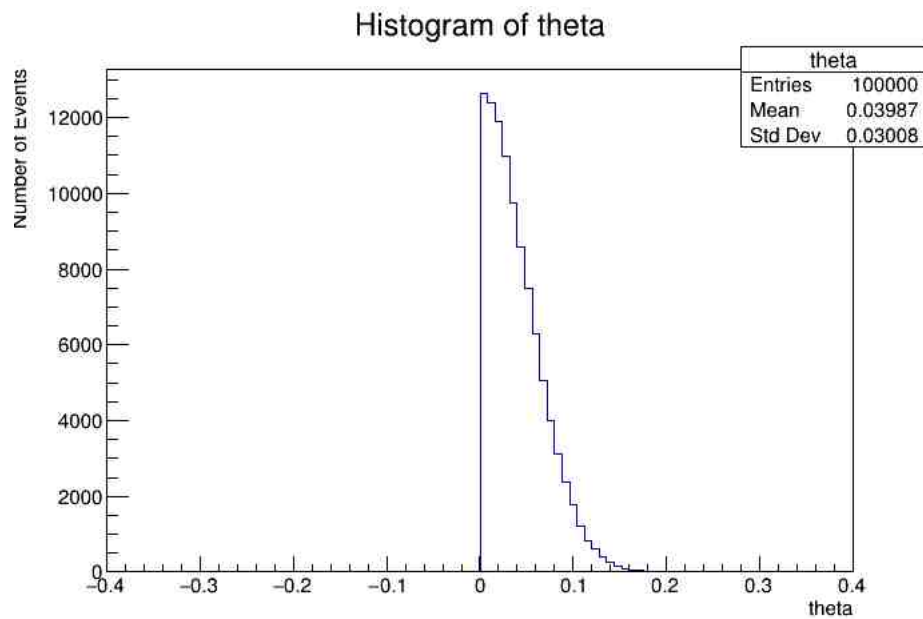
```
1  Int_t cont =0;
```

Logo em seguida, implementei a instrução que irá implementar o corte.

```
1  if(*pz < 145.0){
2      cont++;
3  }
```

E na seção *terminate*, imprimi o valor final da minha variável contadora.

```
1  std::cout << "Number of events with pz is less than 145 GeV: " << cont << endl;
```

Figura 6: Histograma de θ .**Problema 7**

Revise your code to write the histograms to a file.

Para salvar todos os histogramas em um único arquivo, primeiro implementamos a instrução ROOT que cria o arquivo de saída.

```
1 TFile *f = new TFile("histograms_cpp.root","RECREATE");
```

E na seção *terminate* salvamos todos os histogramas dentro deste arquivo com os comandos:

```
1 chi2Hist->Write();
2 ebeamHist->Write();
3 scattHist->Write();
4 ptHist->Write();
5 thetaHist->Write();
6 f->Write();
7 f->Close();
```