

Exercício - Cinemática Relativística em Física de Partículas

Professores: Dilson, Eliza & Maurício

Name: Diogo Gomes Santos Caffonso de Morais

EXERCICIO

O código utilizado está abaixo. Foi executado de dentro do JupyterNotebook.

```
1  #include <TFile.h>
2  #include <TTree.h>
3  #include <TTreeReader.h>
4  #include <TTreeReaderArray.h>
5  #include <TCanvas.h>
6  #include <TH1F.h>
7  #include <TMath.h>
8  #include <iostream>
9  #include <vector>
10 #include <filesystem>
11
12 double calcular_massa_invariante(const TTreeReaderArray<float>& pt, const
    TTreeReaderArray<float>& eta, const TTreeReaderArray<float>& phi) {
13     if (pt.GetSize() >= 2) {
14         return sqrt(2 * pt[0] * pt[1] * (TMath::CosH(eta[0] - eta[1]) - TMath::Cos(
            phi[0] - phi[1])));
15     }
16     return -1.0; // Valor inv lido caso n o haja pelo menos dois elementos
17 }
18
19 void analise() {
20     std::vector<std::string> diretorios = {
21         "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/
            UL2016_MiniAODv2_NanoAODv9-v1/100000",
22         "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/
            UL2016_MiniAODv2_NanoAODv9-v1/1010000",
23         "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/
            UL2016_MiniAODv2_NanoAODv9-v1/250000"
24     };
25
26     std::vector<double> e_massas_invariantes, m_massas_invariantes,
        t_massas_invariantes;
27
28     for (const auto& dir : diretorios) {
29         for (const auto& entry : std::filesystem::directory_iterator(dir)) {
30             std::string file_path = entry.path();
31             TFile file(file_path.c_str(), "READ");
32             if (!file.IsOpen()) continue;
33
34             TTreeReader reader("Events", &file);
35             TTreeReaderArray<float> Electron_pt(reader, "Electron_pt");
36             TTreeReaderArray<float> Electron_eta(reader, "Electron_eta");
37             TTreeReaderArray<float> Electron_phi(reader, "Electron_phi");
38             TTreeReaderArray<float> Muon_pt(reader, "Muon_pt");
39             TTreeReaderArray<float> Muon_eta(reader, "Muon_eta");
40             TTreeReaderArray<float> Muon_phi(reader, "Muon_phi");
41             TTreeReaderArray<float> Tau_pt(reader, "Tau_pt");
42             TTreeReaderArray<float> Tau_eta(reader, "Tau_eta");
43             TTreeReaderArray<float> Tau_phi(reader, "Tau_phi");
44
```

```

45     while (reader.Next()) {
46         if (Electron_pt.GetSize() >= 2) {
47             e_massas_invariantes.push_back(calcular_massa_invariante(
48                 Electron_pt, Electron_eta, Electron_phi));
49         }
50         if (Muon_pt.GetSize() >= 2) {
51             m_massas_invariantes.push_back(calcular_massa_invariante(Muon_pt,
52                 Muon_eta, Muon_phi));
53         }
54         if (Tau_pt.GetSize() >= 2) {
55             t_massas_invariantes.push_back(calcular_massa_invariante(Tau_pt,
56                 Tau_eta, Tau_phi));
57         }
58     }
59
60     TCanvas* canvas = new TCanvas("canvas", "Distribui es de Massas Invariantes",
61         800, 600);
62     TH1F* hEletron = new TH1F("hEletron", "", 50, 0, 200);
63     TH1F* hMuon = new TH1F("hMuon", "", 50, 0, 200);
64     TH1F* hTau = new TH1F("hTau", "", 50, 0, 200);
65
66     for (const auto& massa : e_massas_invariantes) if (massa >= 0) hEletron->Fill(
67         massa);
68     for (const auto& massa : m_massas_invariantes) if (massa >= 0) hMuon->Fill(massa);
69     for (const auto& massa : t_massas_invariantes) if (massa >= 0) hTau->Fill(massa);
70
71     hEletron->SetLineColor(kBlue);
72     hEletron->SetStats(0); // Desativa a caixa de estat sticas
73     hEletron->GetXaxis()->SetTitle("e_mass (GeV/c^{2})"); // R tulo do eixo X
74     hEletron->GetYaxis()->SetTitle("Eventos"); // R tulo do eixo Y
75     hEletron->Draw();
76     canvas->SaveAs("e_massa_invariante.png");
77
78     hMuon->SetLineColor(kBlue);
79     hMuon->SetStats(0); // Desativa a caixa de estat sticas
80     hMuon->GetXaxis()->SetTitle("#mu_mass (GeV/c^{2})"); // R tulo do eixo X
81     hMuon->GetYaxis()->SetTitle("Eventos"); // R tulo do eixo Y
82     hMuon->Draw();
83     canvas->SaveAs("m_massa_invariante.png");
84
85     hTau->SetLineColor(kBlue);
86     hTau->SetStats(0); // Desativa a caixa de estat sticas
87     hTau->GetXaxis()->SetTitle("#tau_mass (GeV/c^{2})"); // R tulo do eixo X
88     hTau->GetYaxis()->SetTitle("Eventos"); // R tulo do eixo Y
89     hTau->Draw();
90     canvas->SaveAs("t_massa_invariante.png");
91
92     delete hEletron;
93     delete hMuon;
94     delete hTau;
95     delete canvas;
96 }

```

Executado o código acima, os histogramas obtidos foram os publicados [neste repositório GitHub](#).