

Professores: Dilson de Jesus Damião, Eliza Melo da Costa e Mauricio Thiel

Name: Bruno Kron Guandalini e Pedro Oliveira

EXERCICIO 1

Usando a ideia inicial do exercício da aula anterior que você calculou a M com a amostra do seu grupo:

Aplique cortes de seleção antes de plotar a massa, vamos exigir um limiar de p_T e η em cada objeto e depois salvar a figura da M no formato PNG.

1. O número de eventos é afetado?
2. Compare os plots de p_T e η antes e depois do corte.
3. Adicione as figuras e o código no git.

RESPOSTA:

```
1 import numpy as np
2 import uproot
3 import awkward as ak
4 import matplotlib.pyplot as plt
5 import vector
6 import coffea
7
8 path = '/opendata/eos/opendata/cms/mc/RunIISummer20UL16NanoAODv9/ZZ_TuneCP5_13TeV-
    pythia8/NANOASIM/106X_mcRun2_asymptotic_v17-v1/130000/AA03EB34-E055-F44D-873B-1
    E114A0B09AA.root'
9
10
11 file = uproot.open(path)
12
13 print('arquivo:', file.keys())
14 tree = file['Events;1']
15 tree.keys()
16
17
18 muon_pt = tree['Muon_pt'].array()
19 muon_phi = tree['Muon_phi'].array()
20 muon_eta = tree['Muon_eta'].array()
21 muon_mass = tree['Muon_mass'].array()
22 nMuon = tree['nMuon'].array()
23
24 muon_pt_val = ak.num(muon_pt) >= 4
25
26 mask_muon = (muon_pt[muon_pt_val][:, 0] > 20) & (np.abs(muon_eta[muon_pt_val][:, 0] <
    2.5))
27 mask_muon2 = (muon_pt[muon_pt_val][:, 1] > 20) & (np.abs(muon_eta[muon_pt_val][:, 1]
    < 2.5))
28 mask_muon3 = (muon_pt[muon_pt_val][:, 0] * muon_pt[muon_pt_val][:, 1]) < 0
29 mask_muon_comb = mask_muon & mask_muon2
30
```

```

31 muon_pt_sel = muon_pt[muon_pt_val][mask_muon_comb]
32 muon_eta_sel = muon_eta[muon_pt_val][mask_muon_comb]
33 muon_phi_sel = muon_phi[muon_pt_val][mask_muon_comb]
34 muon_mass_sel = muon_mass[muon_pt_val][mask_muon_comb]
35
36
37 muon_pt_clean = muon_pt_sel[ak.num(muon_pt_sel) > 0]
38 muon_eta_clean = muon_eta_sel[ak.num(muon_eta_sel) > 0]
39 muon_phi_clean = muon_phi_sel[ak.num(muon_phi_sel) > 0]
40 muon_mass_clean = muon_mass_sel[ak.num(muon_mass_sel) > 0]
41
42
43 muon_pt_flat = ak.flatten(muon_pt_clean)
44 muon_eta_flat = ak.flatten(muon_eta_clean)
45 muon_phi_flat = ak.flatten(muon_phi_clean)
46 muon_mass_flat = ak.flatten(muon_mass_clean)
47
48
49 muon_pt_before_cut = tree['Muon_pt'].array()
50 muon_pt_before_cut_clean = muon_pt_before_cut[ak.num(muon_pt_before_cut) > 0]
51 muon_pt_before = ak.flatten(muon_pt_before_cut_clean)
52
53 muon_phi_before_cut = tree['Muon_phi'].array()
54 muon_phi_before_cut_clean = muon_phi_before_cut[ak.num(muon_phi_before_cut) > 0]
55 muon_phi_before = ak.flatten(muon_phi_before_cut_clean)
56
57 muon_eta_before_cut = tree['Muon_eta'].array()
58 muon_eta_before_cut_clean = muon_eta_before_cut[ak.num(muon_eta_before_cut) > 0]
59 muon_eta_before = ak.flatten(muon_eta_before_cut_clean)
60
61 muon_mass_before_cut = tree['Muon_mass'].array()
62 muon_mass_before_cut_clean = muon_mass_before_cut[ak.num(muon_mass_before_cut) > 0]
63 muon_mass_before = ak.flatten(muon_mass_before_cut_clean)
64
65
66 muon1_4vecs = vector.zip({
67     "pt": muon_pt_clean[:, 0],
68     "eta": muon_eta_clean[:, 0],
69     "phi": muon_phi_clean[:, 0],
70     "mass": muon_mass_clean[:, 0]
71 })
72
73 muon2_4vecs = vector.zip({
74     "pt": muon_pt_clean[:, 1],
75     "eta": muon_eta_clean[:, 1],
76     "phi": muon_phi_clean[:, 1],
77     "mass": muon_mass_clean[:, 1]
78 })
79
80 dimuon = muon1_4vecs + muon2_4vecs
81 mass_dimuon = dimuon.mass
82
83 plt.figure(figsize=(10, 5))
84
85
86 plt.subplot(1, 2, 1)
87 plt.subplot(1, 2, 1)
88 plt.hist(muon_pt_before, bins=50, color='#7a21dd', range=(0, 300))
89
90 plt.title(r'Distribuição de $p_T$ antes do corte do Muon')
91 plt.xlabel(r"$p_{T}$")
92 plt.ylabel("Events")
93

```

```

94 plt.subplot(1, 2, 2)
95 plt.hist(muon_pt_flat, bins=50, color='#7a21dd', range=(0, 500))
96
97 plt.title(r'Distribuição de  $p_T$  depois do corte do Muon')#, loc = 'left')
98 plt.xlabel(r" $p_T$ ")
99 plt.ylabel("Events")
100
101 plt.tight_layout()
102 plt.show()
103
104
105
106 plt.figure(figsize=(10, 5))
107
108 plt.subplot(1, 2, 1)
109 plt.hist(muon_eta_before, bins=50, color='royalblue', range=(-2, 2))
110 plt.title("Distribuição de  $\eta$  antes do corte do Muon")
111 plt.xlabel(r" $\eta$ ")
112 plt.ylabel("Events")
113
114 plt.subplot(1, 2, 2)
115 plt.hist(muon_eta_flat, bins=50, color='royalblue', range=(-2, 2))
116 plt.title("Distribuição de  $\eta$  depois do corte do Muon")
117 plt.xlabel(r" $\eta$ ")
118 plt.ylabel("Events")
119
120 plt.tight_layout()
121 plt.show()
122
123 #plot phi
124
125
126 plt.figure(figsize=(10, 5))
127
128 plt.subplot(1, 2, 1)
129 plt.hist(muon_phi_before, bins=50, color='red', range=(-2, 2))
130 plt.title("Distribuição de  $\phi$  antes do corte do Muon")
131 plt.xlabel(r" $\phi$ ")
132 plt.ylabel("Events")
133
134 plt.subplot(1, 2, 2)
135 plt.hist(muon_phi_flat, bins=50, color='red', range=(-2, 2))
136 plt.title("Distribuição de  $\phi$  depois do corte do Muon")
137 plt.xlabel(r" $\phi$ ")
138 plt.ylabel("Events")
139
140 plt.tight_layout()
141 plt.show()
142
143
144
145 plt.figure(figsize=(10, 5))
146
147
148 plt.hist(mass_dimuon, bins=50, color='#3f90da', alpha=0.7, label='Dimuon', histtype='
step', range = (0,400))
149
150
151 plt.title("Distribuições da massa do Muon")
152 plt.xlabel(r"Massa ( $\text{GeV}/c^2$ )", loc='right')
153 plt.ylabel("Events", loc='top')
154 plt.legend(loc='best')
155 plt.grid(alpha=0.3)

```

```
plt.show()
```

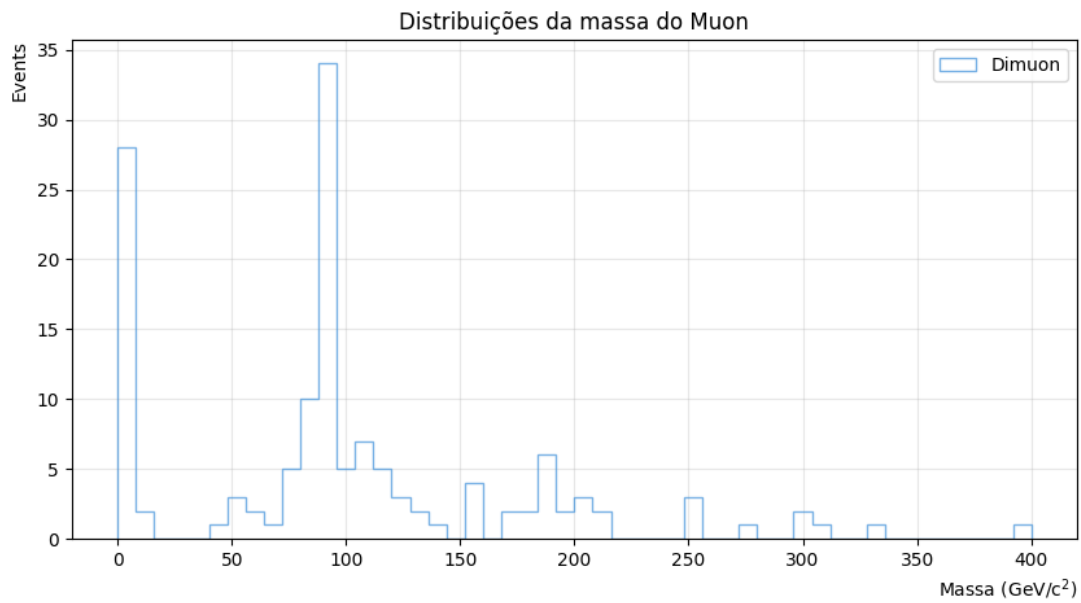
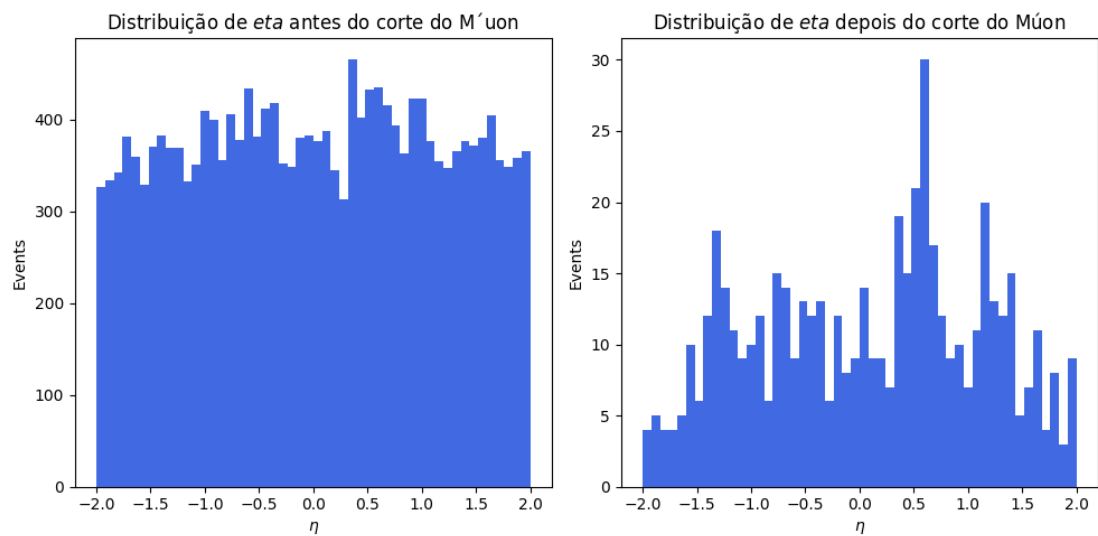
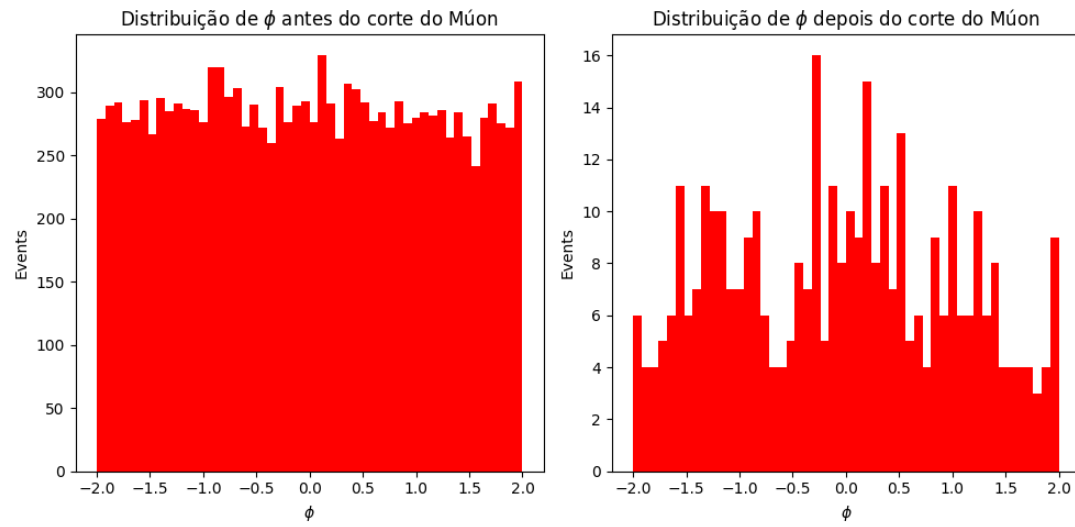
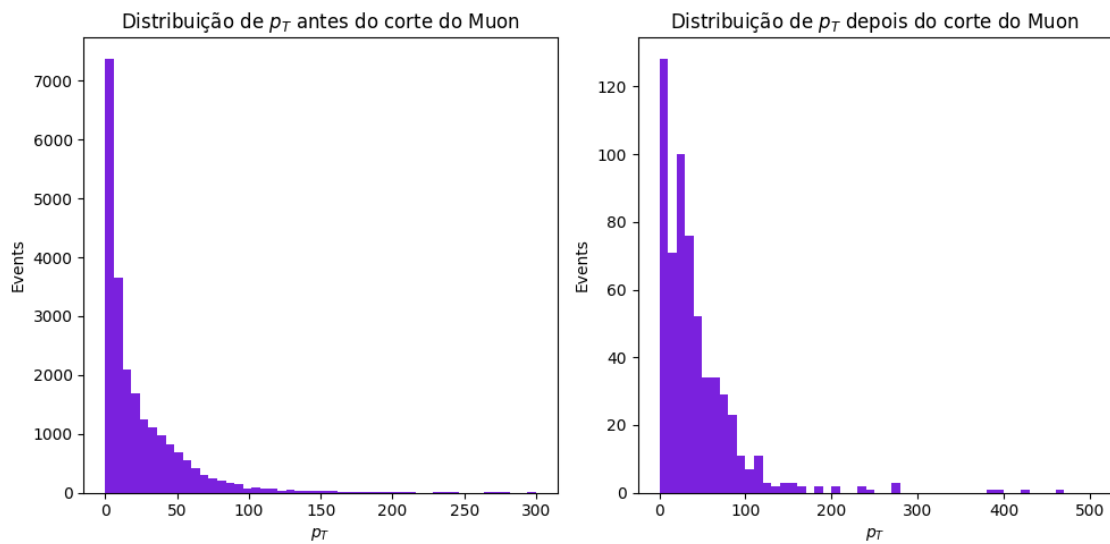


Figura 1: Massa do Múon

Figura 2: η do Múon

Figura 3: ϕ do MúonFigura 4: p_T do Múon