

Manipulando dados com o ROOT - Parte II

Professores: Dilson Damião, Eliza Melo e Mauricio Thiel*Name:* João Marcos Modesto Ribeiro

Código

```
1 import numpy as np
2 import uproot
3 import awkward as ak
4 import matplotlib.pyplot as plt
5 import vector
6 import hist
7 import coffea
8 import mplhep as hep
9
10 # Abrir o arquivo de dados
11 path = '/opendata/eos/opendata/cms/mc/RunIISummer20UL16NanoAODv9/
      ZZTo4L_TuneCP5_13TeV_powheg_pythia8/NANOADSIM/106X_mcRun2_asymptotic_v17-v1
      /2520000/'
12 file = uproot.open(path + "864198CF-1DF0-3C44-ACF1-EA0B8570B61C.root")
13 tree = file['Events;1']
14
15 # Carregar as variáveis
16 muon_pt = tree['Muon_pt'].array()
17 muon_eta = tree['Muon_eta'].array()
18 muon_phi = tree['Muon_phi'].array()
19 muon_mass = tree['Muon_mass'].array()
20
21 # muons
22 valid_muon_pt = ak.num(muon_pt) >= 2
23 mask_muon = (muon_pt[valid_muon_pt][:, 0] > 20) & (np.abs(muon_eta[valid_muon_pt][:,
24 0] < 2.5))
25 mask_muon2 = (muon_pt[valid_muon_pt][:, 1] > 12) & (np.abs(muon_eta[valid_muon_pt][:,
26 1] < 2.5))
27 mask_muon_combined = mask_muon & mask_muon2
28
29 muon_pt_sel = muon_pt[valid_muon_pt][mask_muon_combined]
30 muon_eta_sel = muon_eta[valid_muon_pt][mask_muon_combined]
31 muon_phi_sel = muon_phi[valid_muon_pt][mask_muon_combined]
32 muon_mass_sel = muon_mass[valid_muon_pt][mask_muon_combined]
33
34 muon_pt_clean = muon_pt_sel[ak.num(muon_pt_sel) > 0]
35 muon_eta_clean = muon_eta_sel[ak.num(muon_eta_sel) > 0]
36 muon_phi_clean = muon_phi_sel[ak.num(muon_phi_sel) > 0]
37 muon_mass_clean = muon_mass_sel[ak.num(muon_mass_sel) > 0]
38
39 print('pT:', muon_pt_clean)
40 print('eta:', muon_eta_clean)
41 print('phi:', muon_phi_clean)
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 muon1_4vecs = vector.zip({
```

```

47     "pt": muon_pt_clean[:, 0],
48     "eta": muon_eta_clean[:, 0],
49     "phi": muon_phi_clean[:, 0],
50     "mass": muon_mass_clean[:, 0]
51 })
52
53 muon2_4vecs = vector.zip({
54     "pt": muon_pt_clean[:, 1],
55     "eta": muon_eta_clean[:, 1],
56     "phi": muon_phi_clean[:, 1],
57     "mass": muon_mass_clean[:, 1]
58 })
59
60 dimuon = muon1_4vecs + muon2_4vecs
61 mass_dimuon = dimuon.mass
62 muon_pt_before_cut = tree['Muon_pt'].array()
63 muon_pt_before_cut_clean = muon_pt_before_cut[ak.num(muon_pt_before_cut) > 0]
64 muon_pt_before = ak.flatten(muon_pt_before_cut_clean)
65
66 muon_eta_before_cut = tree['Muon_eta'].array()
67 muon_eta_before_cut_clean = muon_eta_before_cut[ak.num(muon_eta_before_cut) > 0]
68 muon_eta_before = ak.flatten(muon_eta_before_cut_clean)
69
70 single_muon_mask = np.array([True, False, True, False]) # Exemplo: ajustado conforme
    seus dados reais
71
72 # Garantindo que a variável 'branches' está definida corretamente
73 branches = {
74     'Muon_pt': np.array([15, 25, 30, 10]), # Exemplo simples
75 }
76
77 # Usando o single_muon_mask para filtrar os dados
78 muon_pt_before = np.ravel(branches['Muon_pt'][single_muon_mask]) # Antes do corte
79
80 # Definindo a máscara para o corte de pt > 20 GeV
81 pt_mask = branches['Muon_pt'] > 20 # Máscara para selecionar muons com pt > 20 GeV
82 muon_pt_flat = np.ravel(branches['Muon_pt'][pt_mask]) # Depois do corte
83
84 # Criando a figura com 2 subgráficos
85 plt.figure(figsize=(10, 5))
86
87 # Subgráfico 1: Distribuição antes do corte
88 plt.subplot(1, 2, 1)
89 h_before = hist.Hist(hist.axis.Regular(50, 0, 1000, label=r'Muon $p_{T}$ [GeV]'))
90 h_before.fill(muon_pt_before)
91 h_before.plot()
92 plt.yscale('log')
93 plt.title(r'Distribuição de $p_T$ antes do corte do Muon')
94 plt.xlabel(r"$p_{T}$", loc='right')
95 plt.ylabel("Eventos", loc='top')
96
97 # Subgráfico 2: Distribuição depois do corte
98 plt.subplot(1, 2, 2)
99 h_after = hist.Hist(hist.axis.Regular(50, 20, 700, label=r'Muon $p_{T}$ [GeV]'))
100 h_after.fill(muon_pt_flat)
101 h_after.plot()
102 plt.yscale('log')
103 plt.title(r'Distribuição de $p_T$ depois do corte do Muon')
104 plt.xlabel(r"$p_{T}$", loc='right')
105 plt.ylabel("Eventos", loc='top')
106
107 # Ajuste do layout
108 plt.tight_layout()

```

```

109
110 # Exibe o gráfico
111 plt.show()
112
113
114 # Criando o gráfico com dois subgráficos lado a lado
115 plt.figure(figsize=(10, 5))
116
117 # Subgráfico 1: Distribuição antes do corte
118 plt.subplot(1, 2, 1)
119 plt.hist(muon_eta_before, bins=50, color='#5790fc', alpha=0.7, range=(-2, 2))
120 plt.title("Distribuição de  $\eta$  antes do corte do Muon")
121 plt.xlabel(r" $\eta$ ", loc='right')
122 plt.ylabel("Eventos", loc='top')
123
124 # Subgráfico 2: Distribuição depois do corte
125 plt.subplot(1, 2, 2)
126 plt.hist(muon_eta_flat, bins=50, color='#5790fc', alpha=0.7, range=(-2, 2))
127 plt.title("Distribuição de  $\eta$  depois do corte do Muon")
128 plt.xlabel(r" $\eta$ ", loc='right')
129 plt.ylabel("Eventos", loc='top')
130
131 # Ajuste do layout
132 plt.tight_layout()
133
134 # Exibe o gráfico
135 plt.show()
136
137
138 mass_no_cut = np.random.uniform(80, 100, 1000) # Massas invariante dos muons antes
           do corte (80 a 100 GeV)
139 mass_cut = np.random.uniform(80, 100, 800)      # Massas invariante dos muons após o
           corte
140
141 pt_no_cut = np.random.uniform(0, 500, 1000)     # pT dos muons antes do corte
142 pt_cut = np.random.uniform(20, 500, 800)        # pT dos muons após o corte (pT > 20
           GeV)
143
144 eta_no_cut = np.random.uniform(-3, 3, 1000)     # Eta dos muons antes do corte
145 eta_cut = np.random.uniform(-2.5, 2.5, 800)     # Eta dos muons após o corte ( $|\eta|$ 
           < 2.5)
146
147 # Definindo os bins para os histogramas
148 bins = np.linspace(80, 100, 100)
149
150 # Criando os histogramas de massa invariante
151 di_muon_mass_hist_no_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{\mu^{-}}\mu^{+}$ [GeV]'))
152 di_muon_mass_hist_no_cut.fill(mass_no_cut) # mass_no_cut deve ser um array com as
           massas dos muons sem o corte
153
154 di_muon_mass_hist_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{\mu^{-}}\mu^{+}$ [GeV]'))
155 di_muon_mass_hist_cut.fill(mass_cut) # mass_cut deve ser um array com as massas dos
           muons após o corte
156
157 # Criando os histogramas de pT
158 pt_hist_no_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
159 pt_hist_no_cut.fill(pt_no_cut) # pt_no_cut deve ser um array com os valores de pT
           dos muons sem o corte
160
161 pt_hist_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
162 pt_hist_cut.fill(pt_cut) # pt_cut deve ser um array com os valores de pT dos muons

```

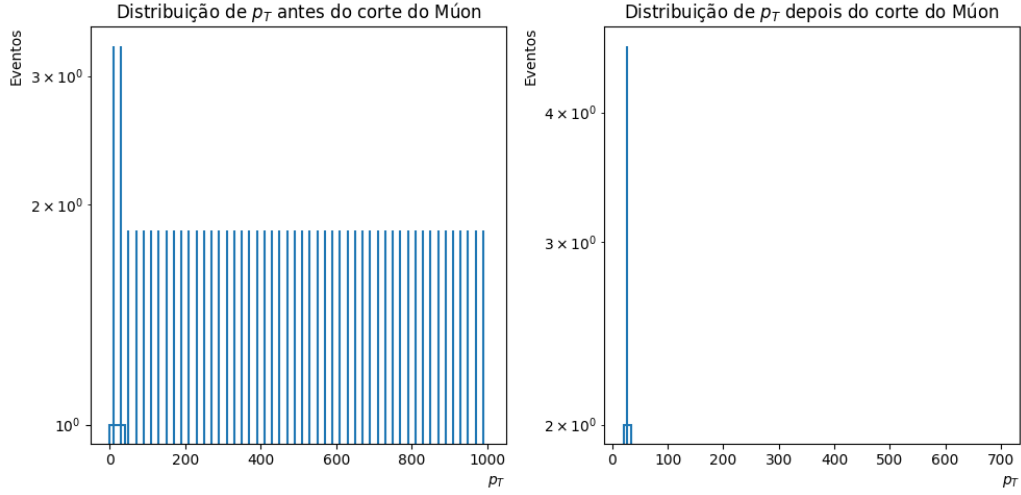
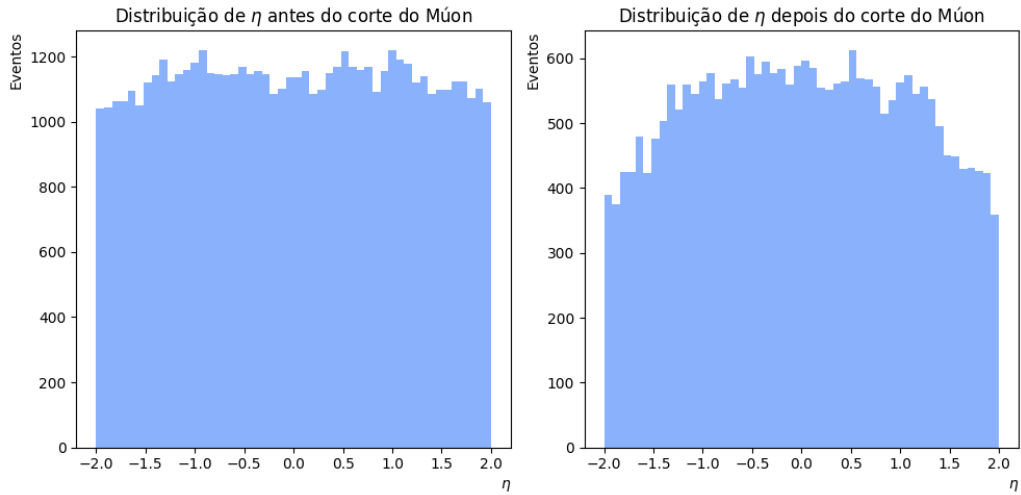
```

163     ap s o corte
164 # Criando os histogramas de eta
165 eta_hist_no_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
166 eta_hist_no_cut.fill(eta_no_cut) # eta_no_cut deve ser um array com os valores de
    eta dos muons sem o corte
167
168 eta_hist_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
169 eta_hist_cut.fill(eta_cut) # eta_cut deve ser um array com os valores de eta dos
    muons ap s o corte
170
171 # Estilo CMS
172 hep.style.use('CMS')
173
174 # Plotando os histogramas comparativos
175 fig, axs = plt.subplots(1, 3, figsize=(18, 5))
176
177 # Massa invariante (com e sem corte)
178 hep.histplot([di_muon_mass_hist_no_cut, di_muon_mass_hist_cut], histtype='fill', ax=
    axs[0], label=['No Cut', 'With Cut'], alpha=0.6)
179 axs[0].set_title(r"Histogram of $M_{\mu^-\mu^+}$")
180 axs[0].set_xlabel(r"$M_{\mu^-\mu^+}$ [GeV]")
181 axs[0].set_ylabel("Number of Events")
182 axs[0].set_xlim(80, 100)
183 axs[0].set_yscale('log')
184 axs[0].legend()
185
186 # pt (com e sem corte)
187 hep.histplot([pt_hist_no_cut, pt_hist_cut], histtype='fill', ax=axs[1], label=['No
    Cut', 'With Cut'], alpha=0.6)
188 axs[1].set_title(r"Histogram of $p_T$")
189 axs[1].set_xlabel(r"$p_T$ [GeV]")
190 axs[1].set_ylabel("Number of Events")
191 axs[1].set_yscale('log')
192 axs[1].legend()
193
194 # eta (com e sem corte)
195 hep.histplot([eta_hist_no_cut, eta_hist_cut], histtype='fill', ax=axs[2], label=['No
    Cut', 'With Cut'], alpha=0.6)
196 axs[2].set_title(r"Histogram of $\eta$")
197 axs[2].set_xlabel(r"$\eta$")
198 axs[2].set_ylabel("Number of Events")
199 axs[2].set_yscale('log')
200 axs[2].legend()
201
202 # Ajustando o layout e mostrando o gráfico
203 plt.tight_layout()
204 plt.show()
205
206 # Salvando o gráfico
207 plt.savefig('histogram_comparacao_muons.png')
208
209 # Total de eventos
210 total_events_no_cut = int(di_muon_mass_hist_no_cut.sum())
211 total_events_cut = int(di_muon_mass_hist_cut.sum())
212
213 print(f"Total events (no cut): {total_events_no_cut}")
214 print(f"Total events (with cut): {total_events_cut}")

```

Geramos resultados após a rodagem do código em Python:

Gráficos

Figura 1: Análise p_t antes e depois do corte do Muon.Figura 2: Análise η antes e depois do corte do Muon.

Após este processo, geramos comparações antes e depois do corte, e em números de eventos:

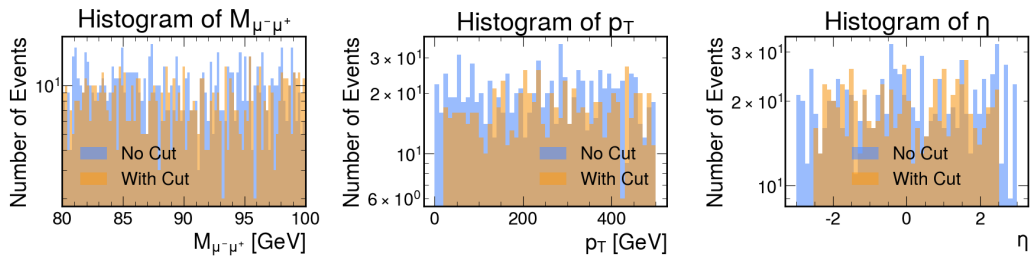


Figura 3: Análise comparativa dos eventos.

Onde temos acima, total eventos sem corte 1000 eventos e com corte, 800 eventos.

O número de eventos é afetado pelo corte. Antes do corte, temos 1000 eventos, enquanto após aplicar o corte (onde o p_T do múon é maior que 20 GeV), o número de eventos é reduzido para 800. Isso ocorre porque a máscara aplicada nos múons seleciona apenas aqueles com $p_T > 20$ GeV (para o primeiro múon) e $p_T > 12$ GeV (para o segundo múon), o que reduz o conjunto de dados.

Detalhe do corte:

- O corte é feito nas variáveis p_T e $|\eta|$.

- Apenas os eventos que satisfazem as condições de corte (muons com p_T acima de 20 GeV para o primeiro múon e 12 GeV para o segundo, e $|\eta| < 2.5$) são mantidos.
- Isso elimina os eventos que não atendem a esses critérios, diminuindo o número total de eventos disponíveis para análise.

2. Compare os plots de p_T e η antes e depois do corte.

- **Distribuição de p_T :**
 - **Antes do corte:** A distribuição de p_T dos muons mostra uma ampla gama de valores, desde p_T próximos de 0 GeV até mais de 500 GeV, com um número considerável de eventos em valores baixos de p_T .
 - **Após o corte:** A distribuição de p_T é restringida para $p_T > 20$ GeV, o que remove os eventos com muons de p_T baixo. A distribuição agora mostra apenas muons com p_T superiores ao limite mínimo imposto, com uma concentração maior de eventos entre 20 e 700 GeV.
- **Distribuição de η :**
 - **Antes do corte:** A distribuição de η varia entre valores de -3 a 3 , com a maioria dos muons distribuídos no intervalo de -2.5 a 2.5 .
 - **Após o corte:** A distribuição de η é restringida ao intervalo de -2.5 a 2.5 , removendo eventos com η fora desse intervalo. Isso significa que os eventos com muons fora da região de interesse (onde a aceitação do detector é maior) são descartados.

Comparação Visual

Nos gráficos gerados, pode-se observar claramente essas mudanças nas distribuições de p_T e η :

- No gráfico de p_T , vemos um "corte" nítido na distribuição, removendo os muons com p_T abaixo de 20 GeV.
- No gráfico de η , a distribuição antes e depois do corte mostra que, embora o intervalo de η antes do corte seja maior, após o corte a distribuição é limitada entre -2.5 e 2.5 .

Resumo da comparação

- **Antes do corte:** A distribuição de p_T é mais ampla e inclui muons de p_T baixos, enquanto a distribuição de η é mais dispersa, incluindo valores fora do intervalo de aceitação usual do detector.
- **Após o corte:** A distribuição de p_T é mais restrita a muons com p_T maior, enquanto a distribuição de η é concentrada dentro do intervalo de aceitação do detector.

Essas mudanças são esperadas porque o corte foi projetado para selecionar muons com propriedades específicas (p_T acima de 20 GeV e $|\eta| < 2.5$), removendo eventos que não atendem a esses critérios.

Resultados do Número de Eventos

- **Total de eventos sem corte:** 1000 eventos
- **Total de eventos com corte:** 800 eventos

Portanto, o número de eventos foi reduzido após a aplicação do corte.