

Introdução à Análise de Dados em FAE
(24/11/2024)

Exercícios de Manipulação de dados com ROOT
Part.II(aula 7)

Professores:
Eliza Melo, Dilson Damião e Mauricio Thiel

Nome:
Matheus da Costa Geraldês

1 Aviso

os códigos, as imagens e assim como os arquivos usados se encontram no meu Git:
<https://github.com/Geraldes-Matheus/cruso-analise-de-dados-2024-2/tree/main/exercicios7/data>

2 Estrutura do Código

O código foi desenvolvido de maneira a seguir as etapas a seguir:

2.1 1. Carregamento dos Dados

Utilizamos o pacote `uproot` para abrir o arquivo ROOT, que contém as informações de diferentes variáveis físicas dos eventos simulados. O código realiza a leitura da árvore `Events`, que contém as variáveis dos objetos fundamentais, como:

- p_T , η , ϕ e massa dos múons, elétrons, taús e jets.
- Variáveis adicionais como o número de partículas de cada tipo no evento ($nMuon$, $nElectron$, $nTau$, $nJet$) e o sinal de disparo (`HLT_IsoMu24`).

2.2 2. Aplicação dos Cortes de Seleção

Foram aplicados cortes para garantir que os objetos tivessem valores físicos dentro de uma faixa realista. Especificamente, selecionamos os seguintes critérios de p_T e η :

- Para os múons: $p_T > 12 \text{ GeV}$ para o primeiro múon e $p_T > 4 \text{ GeV}$ para o segundo, com η dentro de ± 2.4 .
- Para os elétrons: $p_T > 12 \text{ GeV}$ para o primeiro elétron e $p_T > 4 \text{ GeV}$ para o segundo, com η dentro de ± 2.4 .

- Para os taús: $p_T > 12 \text{ GeV}$ para o primeiro taú e $p_T > 4 \text{ GeV}$ para o segundo, com η dentro de ± 2.4 .
- Para os jets: $p_T > 30 \text{ GeV}$ para o primeiro jet e $p_T > 30 \text{ GeV}$ para o segundo, com η dentro de ± 2.4 .

Além disso, realizamos uma verificação de que cada evento tenha pelo menos 2 partículas do tipo relevante (múons, elétrons, taús ou jets).

2.3 3. Cálculo das Massas Invariantes

Depois de aplicar os cortes, calculamos a massa invariante dos pares de partículas selecionados (múons, elétrons, taús e jets). A massa invariante de dois objetos pode ser calculada pela soma dos quatro-vetores p^μ dos dois objetos, e é dada pela fórmula:

$$M = \sqrt{(E_1 + E_2)^2 - (\vec{p}_1 + \vec{p}_2)^2}$$

onde E_1 e E_2 são as energias dos dois objetos, e \vec{p}_1 e \vec{p}_2 são os vetores de momento.

- As distribuições de p_T e η são feitas com os histogramas de cada partícula, antes e depois dos cortes.
- A massa invariante dos pares de partículas também é plotada para diferentes tipos de partículas.

3 Resultados

A seguir, temos os resultados da análise, incluindo as comparações das distribuições de p_T , η e as massas invariantes para os pares de partículas.

3.1 1. Número de Eventos

A aplicação dos cortes de p_T e η afetou diretamente o número de eventos disponíveis para a análise. O número de eventos foi reduzido devido à exclusão de eventos que não satisfazem os critérios de seleção. Isso é esperado, pois estamos impondo condições adicionais sobre as partículas em cada evento.

Número total de eventos: 168000

Número de eventos após os cortes para muons: 35109

Número de eventos após os cortes para elétrons: 31863

Número de eventos após os cortes para taús: 45856

Número de eventos após os cortes para jets: 43491

3.2 2. Comparação dos Plots de p_T e η

Para as distribuições de p_T e η , podemos observar uma redução no número de eventos para os valores mais baixos de p_T , uma vez que os eventos com partículas de baixo p_T foram cortados. Já a distribuição de η foi mais afetada para as partículas com η fora da faixa de ± 2.4 .

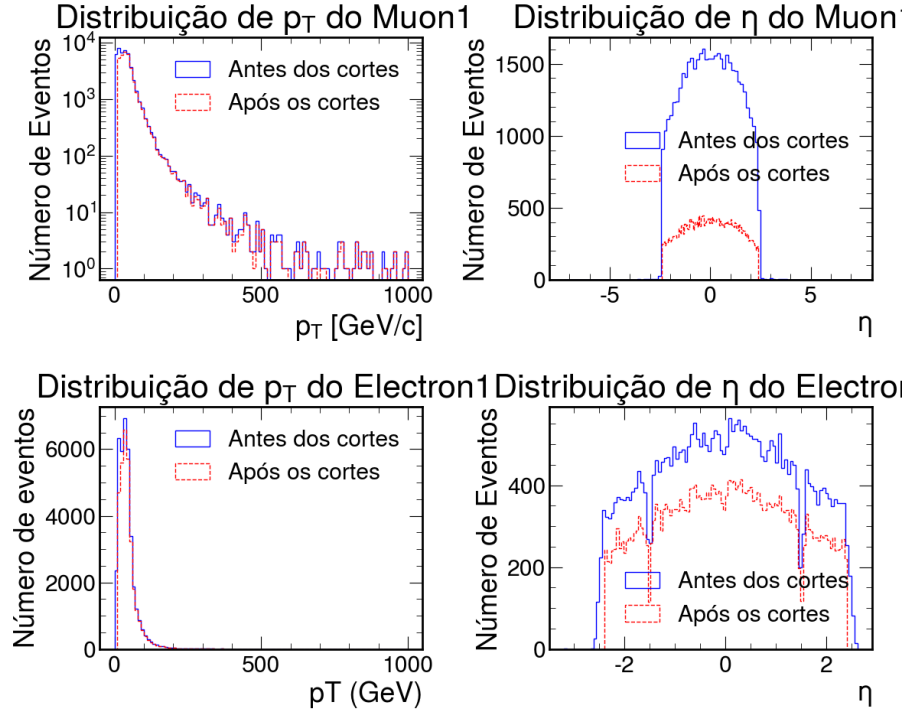


Figure 1: Comparação entre antes e depois do corte aplicado no Pt e η do Múon e Elétron

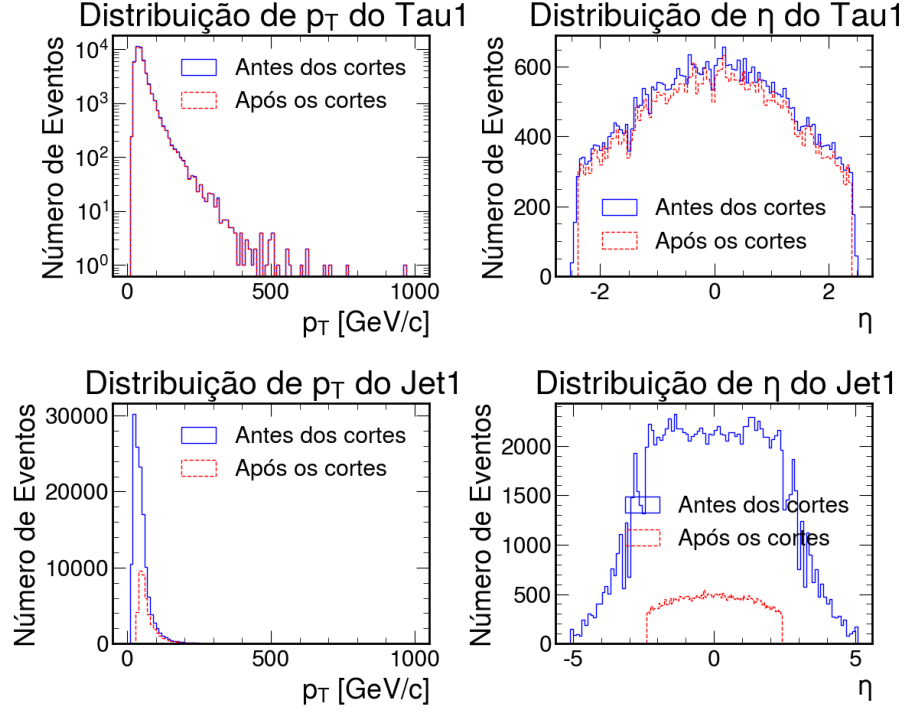


Figure 2: Comparação entre antes e depois do corte aplicado no p_T e η dos Jatos e Tau

3.3 3. Massas Invariantes

A distribuição de massas invariantes dos pares de partículas mostra picos para as massas do Z e outras ressonâncias. A aplicação dos cortes também afetou a forma dessas distribuições, filtrando eventos com partículas de p_T muito baixo, o que resultou em uma distribuição mais concentrada na região de interesse física.

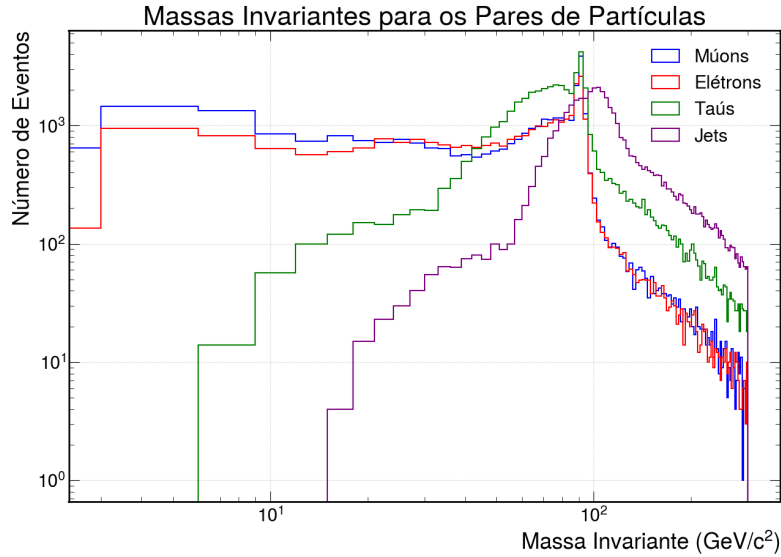


Figure 3: Plot da Massa invariante dos pares de partículas após os cortes no P_t e η dos Jatos, Tau, Múon e Elétron

4 Código Implementado

Código 1

```

1 import uproot
2 import awkward as ak
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import vector
7 import hist
8 import mplhep as hep
9
10 print(uproot.__version__)
11 print(ak.__version__)
12
13 # Abrir o arquivo de dados
14 data = uproot.open("/opendata/eos/opendata/cms/mc/RunIISummer20UL16NanoAODv9/
ZZTo4L_TuneCP5_13TeV_powheg_pythia8/NANOADSIM/106
X_mcRun2_asymptotic_v17-v1/2430000/051E9D22-4F30-8E49-8477-644E65768282.
root")
15 data.keys()
16
17 # Verificar o conte do da rvore de eventos
18 tree = data['Events;1']
19
20 # Vari veis cinem ticas dos m ons , el trons , ta s e jets
21 muon_pt = tree['Muon_pt'].array()
22 muon_eta = tree['Muon_eta'].array()
23 muon_phi = tree['Muon_phi'].array()
24 muon_mass = tree['Muon_mass'].array()

```

```

25
26 electron_pt = tree['Electron_pt'].array()
27 electron_eta = tree['Electron_eta'].array()
28 electron_phi = tree['Electron_phi'].array()
29 electron_mass = tree['Electron_mass'].array()
30
31 tau_pt = tree['Tau_pt'].array()
32 tau_eta = tree['Tau_eta'].array()
33 tau_phi = tree['Tau_phi'].array()
34 tau_mass = tree['Tau_mass'].array()
35
36 jet_pt = tree['Jet_pt'].array()
37 jet_eta = tree['Jet_eta'].array()
38 jet_phi = tree['Jet_phi'].array()
39 jet_mass = tree['Jet_mass'].array()
40
41 hlt_isomu24 = tree['HLT_IsoMu24'].array()
42 nmuon = tree['nMuon'].array()
43 nelectron = tree['nElectron'].array()
44 ntau = tree['nTau'].array()
45 njet = tree['nJet'].array()
46
47 # Sele o do trigger
48 cut_trigger = (hlt_isomu24 == True)
49
50 # Sele o de pt e eta para os muons
51 valid_events = ak.num(muon_pt) >= 2 # Garante que temos pelo menos 2 muons
    por evento
52
53 mask_mu1 = (muon_pt[valid_events][:, 0] > 12.0) & (np.abs(muon_eta[
    valid_events][:, 0]) < 2.4)
54 mask_mu2 = (muon_pt[valid_events][:, 1] > 4.0) & (np.abs(muon_eta[
    valid_events][:, 1]) < 2.4)
55
56 cut_cinematico_muons = mask_mu1 & mask_mu2
57
58 muon_pt_sel = muon_pt[valid_events][cut_cinematico_muons]
59 muon_eta_sel = muon_eta[valid_events][cut_cinematico_muons]
60
61 # Sele o de pt e eta para os el trons
62 valid_events_elec = ak.num(electron_pt) >= 2
63 mask_elec = (electron_pt[valid_events_elec][:, 0] > 12.0) & (np.abs(
    electron_eta[valid_events_elec][:, 0]) < 2.4)
64 mask_elec2 = (electron_pt[valid_events_elec][:, 1] > 4.0) & (np.abs(
    electron_eta[valid_events_elec][:, 1]) < 2.4)
65
66 cut_cinematico_elec = mask_elec & mask_elec2
67
68 electron_pt_sel = electron_pt[valid_events_elec][cut_cinematico_elec]
69 electron_eta_sel = electron_eta[valid_events_elec][cut_cinematico_elec]
70
71 # Sele o de pt e eta para os ta s
72 valid_events_tau = ak.num(tau_pt) >= 2
73 mask_tau = (tau_pt[valid_events_tau][:, 0] > 12.0) & (np.abs(tau_eta[
    valid_events_tau][:, 0]) < 2.4)
74 mask_tau2 = (tau_pt[valid_events_tau][:, 1] > 4.0) & (np.abs(tau_eta[
    valid_events_tau][:, 1]) < 2.4)
75
76 cut_cinematico_tau = mask_tau & mask_tau2
77
78 tau_pt_sel = tau_pt[valid_events_tau][cut_cinematico_tau]
79 tau_eta_sel = tau_eta[valid_events_tau][cut_cinematico_tau]
80
81 # Sele o de pt e eta para os jets
82 valid_events_jet = ak.num(jet_pt) >= 2
83 mask_jet = (jet_pt[valid_events_jet][:, 0] > 30.0) & (np.abs(jet_eta[
    valid_events_jet][:, 0]) < 2.4)

```

```

84 mask_jet2 = (jet_pt[valid_events_jet][:, 1] > 30.0) & (np.abs(jet_eta[
    valid_events_jet][:, 1]) < 2.4)
85
86 cut_cinematico_jet = mask_jet & mask_jet2
87
88 jet_pt_sel = jet_pt[valid_events_jet][cut_cinematico_jet]
89 jet_eta_sel = jet_eta[valid_events_jet][cut_cinematico_jet]
90
91 # Gerar os plots para cada partícula (m ons, el trons, ta s, jets)
92 fig, axes = plt.subplots(2, 2, figsize=(12, 10))
93
94 # Plot para muon1
95 axes[0, 0].hist(muon_pt[valid_events][:, 0], bins=100, range=(0, 1000),
    histtype='step', color='blue', label='Antes_dos_cortes')
96 axes[0, 0].hist(muon_pt_sel[:, 0], bins=100, range=(0, 1000), histtype='step',
    color='red', linestyle='--', label='Ap os_cortes')
97 axes[0, 0].set_title('Distribui o de p_T$do_Muon1')
98 axes[0, 0].set_xlabel(r'$p_T$[GeV/c]')
99 axes[0, 0].set_ylabel('N mero de Eventos')
100 axes[0, 0].set_yscale('log')
101 axes[0, 0].legend()
102
103 axes[0, 1].hist(muon_eta[valid_events][:, 0], bins=100, color='blue',
    histtype='step', label='Antes_dos_cortes')
104 axes[0, 1].hist(muon_eta_sel[:, 0], bins=100, color='red', histtype='step',
    linestyle='--', label='Ap os_cortes')
105 axes[0, 1].set_title('Distribui o de $\eta$do_Muon1')
106 axes[0, 1].set_xlabel(r'$\eta$')
107 axes[0, 1].set_ylabel('N mero de Eventos')
108 axes[0, 1].legend()
109
110 # Plot para electron1
111 axes[1, 0].hist(electron_pt[valid_events_elec][:, 0], bins=100, range=(0,
    1000), histtype='step', color='blue', label='Antes_dos_cortes')
112 axes[1, 0].hist(electron_pt_sel[:, 0], bins=100, range=(0, 1000), histtype='
    step', color='red', linestyle='--', label='Ap os_cortes')
113 axes[1, 0].set_title('Distribui o de p_T$do_Electron1')
114 axes[1, 0].set_xlabel('p_T(GeV)')
115 axes[1, 0].set_ylabel('N mero de eventos')
116 axes[1, 0].legend()
117
118 axes[1, 1].hist(electron_eta[valid_events_elec][:, 0], bins=100, color='blue'
    , histtype='step', label='Antes_dos_cortes')
119 axes[1, 1].hist(electron_eta_sel[:, 0], bins=100, color='red', histtype='step
    ', linestyle='--', label='Ap os_cortes')
120 axes[1, 1].set_title('Distribui o de $\eta$do_Electron1')
121 axes[1, 1].set_xlabel(r'$\eta$')
122 axes[1, 1].set_ylabel('N mero de Eventos')
123 axes[1, 1].legend()
124
125 plt.tight_layout()
126 plt.show()

```

Código 2

```
1 import uproot
2 import awkward as ak
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import vector
7 import mplhep as hep
8
9 # Abrir o arquivo de dados
10 data = uproot.open("/opendata/eos/opendata/cms/mc/RunIISummer20UL16NanoAODv9/
    ZZTo4L_TuneCP5_13TeV_powheg_pythia8/NANOAODSIM/106
    X_mcRun2_asymptotic_v17-v1/2430000/051E9D22-4F30-8E49-8477-644E65768282.
    root")
11 data.keys()
12
13 # Verificar o conte do da rvore de eventos
14 tree = data['Events;1']
15
16 # Vari veis cinem ticas dos m ons , el trons , ta s e jets
17 muon_pt = tree['Muon_pt'].array()
18 muon_eta = tree['Muon_eta'].array()
19 muon_phi = tree['Muon_phi'].array()
20 muon_mass = tree['Muon_mass'].array()
21
22 electron_pt = tree['Electron_pt'].array()
23 electron_eta = tree['Electron_eta'].array()
24 electron_phi = tree['Electron_phi'].array()
25 electron_mass = tree['Electron_mass'].array()
26
27 tau_pt = tree['Tau_pt'].array()
28 tau_eta = tree['Tau_eta'].array()
29 tau_phi = tree['Tau_phi'].array()
30 tau_mass = tree['Tau_mass'].array()
31
32 jet_pt = tree['Jet_pt'].array()
33 jet_eta = tree['Jet_eta'].array()
34 jet_phi = tree['Jet_phi'].array()
35 jet_mass = tree['Jet_mass'].array()
36
37 hlt_isomu24 = tree['HLT_IsoMu24'].array()
38 nmuon = tree['nMuon'].array()
39 nelectron = tree['nElectron'].array()
40 ntau = tree['nTau'].array()
41 njet = tree['nJet'].array()
42
43 # Sele o do trigger
44 cut_trigger = (hlt_isomu24 == True)
45
46 # Sele o de pt e eta para os muons
47 valid_events = ak.num(muon_pt) >= 2 # Garante que temos pelo menos 2 muons
    por evento
48
49 mask_mu1 = (muon_pt[valid_events][:, 0] > 12.0) & (np.abs(muon_eta[
    valid_events][:, 0]) < 2.4)
50 mask_mu2 = (muon_pt[valid_events][:, 1] > 4.0) & (np.abs(muon_eta[
    valid_events][:, 1]) < 2.4)
51
52 cut_cinematico_muons = mask_mu1 & mask_mu2
53
54 muon_pt_sel = muon_pt[valid_events][cut_cinematico_muons]
55 muon_eta_sel = muon_eta[valid_events][cut_cinematico_muons]
56 muon_phi_sel = muon_phi[valid_events][cut_cinematico_muons]
57 muon_mass_sel = muon_mass[valid_events][cut_cinematico_muons]
58
59 # Cria o dos quadri-vetores dos m ons ap s os cortes
60 muon1_4vecs = vector.zip({
```



```

61     "pt": muon_pt_sel[:, 0],
62     "eta": muon_eta_sel[:, 0],
63     "phi": muon_phi_sel[:, 0],
64     "mass": muon_mass_sel[:, 0]
65 })
66
67 muon2_4vecs = vector.zip({
68     "pt": muon_pt_sel[:, 1],
69     "eta": muon_eta_sel[:, 1],
70     "phi": muon_phi_sel[:, 1],
71     "mass": muon_mass_sel[:, 1]
72 })
73
74 # Somar os 4-vetores do par de muons
75 dimuon = muon1_4vecs + muon2_4vecs
76 mass_dimu = dimuon.mass
77
78 # Seleção de pt e eta para os el trons
79 valid_events_elec = ak.num(electron_pt) >= 2
80 mask_elec = (electron_pt[valid_events_elec][:, 0] > 12.0) & (np.abs(
81     electron_eta[valid_events_elec][:, 0]) < 2.4)
82
83 mask_elec2 = (electron_pt[valid_events_elec][:, 1] > 4.0) & (np.abs(
84     electron_eta[valid_events_elec][:, 1]) < 2.4)
85
86 cut_cinematico_elec = mask_elec & mask_elec2
87
88 electron_pt_sel = electron_pt[valid_events_elec][cut_cinematico_elec]
89 electron_eta_sel = electron_eta[valid_events_elec][cut_cinematico_elec]
90 electron_phi_sel = electron_phi[valid_events_elec][cut_cinematico_elec]
91 electron_mass_sel = electron_mass[valid_events_elec][cut_cinematico_elec]
92
93 # Criação dos quadri-vetores dos el trons após os cortes
94 electron1_4vecs = vector.zip({
95     "pt": electron_pt_sel[:, 0],
96     "eta": electron_eta_sel[:, 0],
97     "phi": electron_phi_sel[:, 0],
98     "mass": electron_mass_sel[:, 0]
99 })
100
101 electron2_4vecs = vector.zip({
102     "pt": electron_pt_sel[:, 1],
103     "eta": electron_eta_sel[:, 1],
104     "phi": electron_phi_sel[:, 1],
105     "mass": electron_mass_sel[:, 1]
106 })
107
108 # Somar os 4-vetores do par de el trons
109 dielectron = electron1_4vecs + electron2_4vecs
110 mass_dielelectron = dielectron.mass
111
112 # Seleção de pt e eta para os taus
113 valid_events_tau = ak.num(tau_pt) >= 2
114 mask_tau = (tau_pt[valid_events_tau][:, 0] > 12.0) & (np.abs(tau_eta[
115     valid_events_tau][:, 0]) < 2.4)
116
117 mask_tau2 = (tau_pt[valid_events_tau][:, 1] > 4.0) & (np.abs(tau_eta[
118     valid_events_tau][:, 1]) < 2.4)
119
120 cut_cinematico_tau = mask_tau & mask_tau2
121
122 tau_pt_sel = tau_pt[valid_events_tau][cut_cinematico_tau]
123 tau_eta_sel = tau_eta[valid_events_tau][cut_cinematico_tau]
124 tau_phi_sel = tau_phi[valid_events_tau][cut_cinematico_tau]
125 tau_mass_sel = tau_mass[valid_events_tau][cut_cinematico_tau]
126
127 # Criação dos quatro-vetores dos taus após os cortes
128 tau1_4vecs = vector.zip({
129     "pt": tau_pt_sel[:, 0],
130     "eta": tau_eta_sel[:, 0],

```

```

125     "phi": tau_phi_sel[:, 0],
126     "mass": tau_mass_sel[:, 0]
127 })
128
129 tau2_4vecs = vector.zip({
130     "pt": tau_pt_sel[:, 1],
131     "eta": tau_eta_sel[:, 1],
132     "phi": tau_phi_sel[:, 1],
133     "mass": tau_mass_sel[:, 1]
134 })
135
136 # Somar os 4-vetores do par de ta s
137 ditau = tau1_4vecs + tau2_4vecs
138 mass_ditau = ditau.mass
139
140 # Sele o de pt e eta para os jets
141 valid_events_jet = ak.num(jet_pt) >= 2
142 mask_jet = (jet_pt[valid_events_jet][:, 0] > 30.0) & (np.abs(jet_eta[
    valid_events_jet][:, 0]) < 2.4)
143 mask_jet2 = (jet_pt[valid_events_jet][:, 1] > 30.0) & (np.abs(jet_eta[
    valid_events_jet][:, 1]) < 2.4)
144
145 cut_cinematico_jet = mask_jet & mask_jet2
146
147 jet_pt_sel = jet_pt[valid_events_jet][cut_cinematico_jet]
148 jet_eta_sel = jet_eta[valid_events_jet][cut_cinematico_jet]
149 jet_phi_sel = jet_phi[valid_events_jet][cut_cinematico_jet]
150 jet_mass_sel = jet_mass[valid_events_jet][cut_cinematico_jet]
151
152 # Cria o dos quadri-vetores dos jets ap s os cortes
153 jet1_4vecs = vector.zip({
154     "pt": jet_pt_sel[:, 0],
155     "eta": jet_eta_sel[:, 0],
156     "phi": jet_phi_sel[:, 0],
157     "mass": jet_mass_sel[:, 0]
158 })
159
160 jet2_4vecs = vector.zip({
161     "pt": jet_pt_sel[:, 1],
162     "eta": jet_eta_sel[:, 1],
163     "phi": jet_phi_sel[:, 1],
164     "mass": jet_mass_sel[:, 1]
165 })
166
167 # Somar os 4-vetores do par de jets
168 dijet = jet1_4vecs + jet2_4vecs
169 mass_dijet = dijet.mass
170
171 # Plotando as massas invariantes
172 plt.figure(figsize=(15, 10))
173 hep.style.use('CMS')
174
175 plt.hist(mass_dimu, bins=100, range=(0, 300), histtype='step', label='M ons',
    color='blue', linewidth=1.5)
176 plt.hist(mass_dielelectron, bins=100, range=(0, 300), histtype='step', label='
    El trons', color='red', linewidth=1.5)
177 plt.hist(mass_ditau, bins=100, range=(0, 300), histtype='step', label='Ta s',
    color='green', linewidth=1.5)
178 plt.hist(mass_dijet, bins=100, range=(0, 300), histtype='step', label='Jets',
    color='purple', linewidth=1.5)
179
180 plt.xlabel(r'MassaInv(GeV/$c^2$)')
181 plt.ylabel('N merodeEventos')
182 plt.legend()
183 plt.xscale('log')
184 plt.yscale('log')
185 plt.grid(True)
186 plt.title('MassasInvariantes para os ParesdePart culas')

```

```
187
188 plt.savefig('Inv-Mass.png', format='png')
189
190 plt.text(100.0, 2.4e3, r'$Z$', fontsize=16, color='black')
191
192 plt.show()
```