

## Lista 7

Professores: Dilson Damião, Eliza Melo e Maurício Thiel

Nome: José Gonçalves Chaves Junior

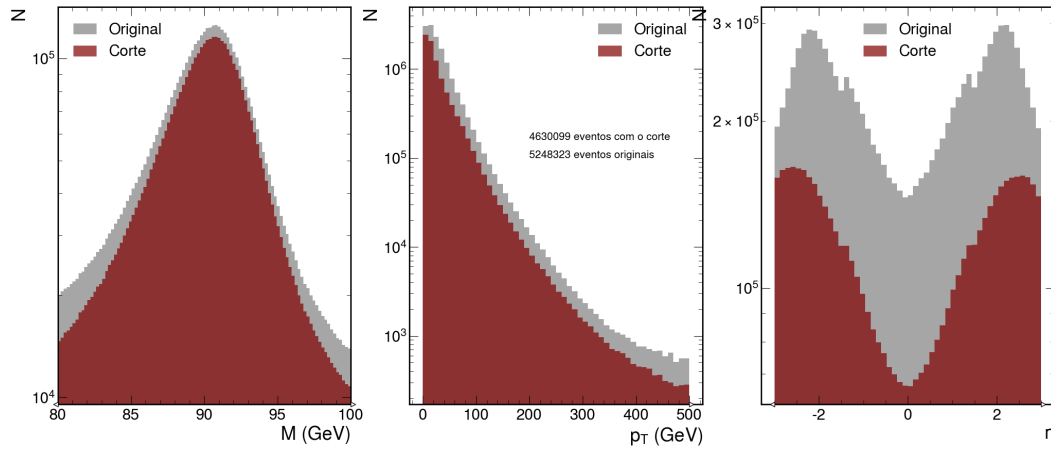


Figura 1: Histogramas para massa, momento transversal e  $\eta$  para a amostra original e a amostra com os cortes propostos na atividade.

Como podemos ver, tanto visualmente quanto logicamente, com a aplicação de um corte, para selecionarmos a amostra, existe uma diminuição na seleção de eventos. Mas, a estatística continua válida para a amostra, visto que ainda existe um número alto de eventos selecionados.

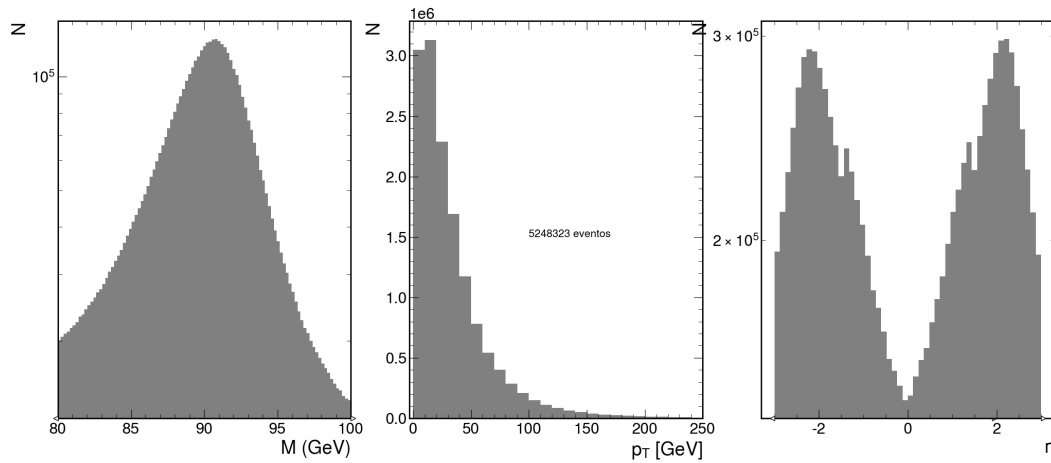


Figura 2: Histograma dos parâmetros físicos da amostra.

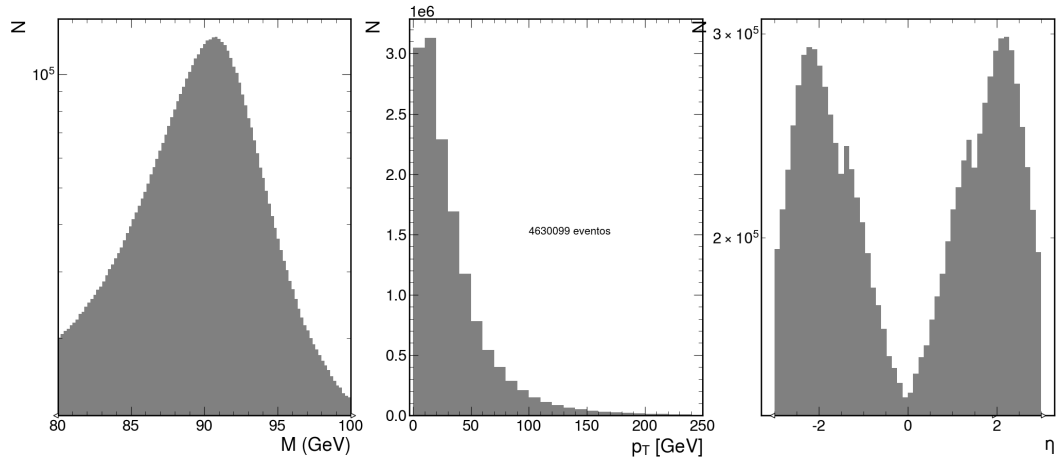


Figura 3: Histograma dos parâmetros físicos da amostra com os cortes.

```

1 import ROOT
2 import uproot
3 import awkward as ak
4 import matplotlib.pyplot as plt
5 import vector
6 import numpy as np
7 import glob
8 import hist
9 import matplotlib.colors as mcolors
10 import matplotlib.font_manager

```

```

1 files = [
2     "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOB/UL2016_MiniAODv2_NanoAODv9-
3     v1/100000/*.root:Events",
4     "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOB/UL2016_MiniAODv2_NanoAODv9-
5     v1/1010000/*.root:Events",
6     "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOB/UL2016_MiniAODv2_NanoAODv9-
7     v1/250000/*.root:Events"
8 ]
9
10 var = [
11     "nElectron",
12     "Electron_pt",
13     "Electron_eta",
14     "Electron_phi",
15     "Electron_mass",
16     "Electron_charge"
17 ]
18
19 arrays = uproot.concatenate(files, filter_name=var)
20
21 nElectron = arrays["nElectron"]
22 eletron_pt = arrays["Electron_pt"]
23 eletron_eta = arrays["Electron_eta"]
24 eletron_phi = arrays["Electron_phi"]
25 eletron_mass = arrays["Electron_mass"]
26 eletron_charge = arrays["Electron_charge"]

```

```

1 mask_di_eletron = nElectron == 2
2
3 eletron_p4 = vector.zip({'pt': eletron_pt,
4                          'eta': eletron_eta,

```

```

5         'phi': eletron_phi,
6         'mass': eletron_mass})
7 #cargas dos eletrons
8 two_eletron_p4 = eletron_p4[mask_di_eletron]
9 two_eletron_charges = eletron_charge[mask_di_eletron]
10 opposite_sign_eletron_mask = two_eletron_charges[:, 0] != two_eletron_charges[:, 1]
11 two_eletron_p4 = two_eletron_p4[opposite_sign_eletron_mask]

1     eletron_p4_0 = two_eletron_p4[:, 0]
2     eletron_p4_1 = two_eletron_p4[:, 1]
3     sum_eletron_p4 = eletron_p4_0 + eletron_p4_1
4     pt = sum_eletron_p4.pt
5     eta = sum_eletron_p4.eta
6     mass = sum_eletron_p4.mass
7     bins = np.linspace(80, 100, 100)
8     ###
9     di_eletron_mass_hist = hist.Hist(hist.axis.Variable(bins, label=r'M (GeV)'))
10    di_eletron_mass_hist.fill(mass)
11    ###
12    pt_hist = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ (GeV)'))
13    pt_hist.fill(pt)
14    ###
15    eta_hist = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
16    eta_hist.fill(eta)

1     fig, axs = plt.subplots(1, 3, figsize=(25, 10))
2     hep.style.use('CMS')
3     #plot da massa
4     hep.histplot(di_eletron_mass_hist, histtype='fill', ax=axs[0], color='grey', edgecolor=
5     'white')
6     axs[0].set_xlabel(r"M (GeV)")
7     axs[0].set_ylabel("N")
8     axs[0].set_xlim(80, 100)
9     axs[0].set_yscale('log')
10    #plot pt
11    hep.histplot(pt_hist, histtype='fill', ax=axs[1], color='grey', edgecolor='white')
12    axs[1].text(100, 1.5*10**6, f"{total_events} eventos", color='black', fontsize=15)
13    axs[1].set_xlabel(r"$p_T$ [GeV]")
14    axs[1].set_xlim(-3, 250)
15    axs[1].set_ylabel("N")
16    #plot eta
17    hep.histplot(eta_hist, histtype='fill', ax=axs[2], color='grey', edgecolor='white')
18    axs[2].set_xlabel(r"$\eta$")
19    axs[2].set_ylabel("N")
20    axs[2].set_yscale('log')
21    plt.rcParams['font.family'] = 'Times New Roman'
22    plt.tight_layout()

1     #Realizando os cortes para pt e eta
2     sel_pt_1 = eletron_p4_0.pt > 20
3     sel_pt_2 = eletron_p4_1.pt > 20
4
5     sel_eta_1 = eletron_p4_0.eta < np.abs(2.4)
6     sel_eta_2 = eletron_p4_1.eta < np.abs(2.4)
7
8     sel_pt = sel_pt_1 & sel_pt_2 & sel_eta_1 & sel_eta_2
9
10    eletron_p4_sel = two_eletron_p4[sel_pt]
11    eletron_p4_1 = eletron_p4_sel[:, 0]
12    eletron_p4_2 = eletron_p4_sel[:, 1]
13    di_eletron_p4_cut = eletron_p4_1 + eletron_p4_2

1     pt = di_eletron_p4_cut.pt
2     eta = di_eletron_p4_cut.eta

```

```

3 mass = di_eletron_p4_cut.mass
4 bins = np.linspace(80, 100, 100)
5 ###
6 di_eletron_mass_hist = hist.Hist(hist.axis.Variable(bins, label=r'M (GeV)'))
7 di_eletron_mass_hist.fill(mass)
8 ###
9 pt_hist = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ (GeV)'))
10 pt_hist.fill(pt)
11 ###
12 eta_hist = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
13 eta_hist.fill(eta)

1 hep.style.use('CMS')
2 fig, axs = plt.subplots(1, 3, figsize=(25, 10))
3
4 # plot da massa
5 hep.histplot(di_eletron_mass_hist, histtype='fill', ax=axs[0], color='grey', edgecolor='white')
6 axs[0].set_xlabel(r"M (GeV)")
7 axs[0].set_ylabel("N")
8 axs[0].set_ylabel("N", labelpad=20)
9 axs[0].set_xlim(80, 100)
10 axs[0].set_yscale('log')
11
12 # plot pt
13 hep.histplot(pt_hist, histtype='fill', ax=axs[1], color='grey', edgecolor='white')
14 axs[1].text(200, 10**5, f"{total_events} eventos", color='black', fontsize=15)
15 axs[1].set_xlabel(r"$p_T$ (GeV)")
16 axs[1].set_ylabel("N")
17 axs[1].set_yscale('log')
18
19 # plot eta
20 hep.histplot(eta_hist, histtype='fill', ax=axs[2], color='grey', edgecolor='white')
21 axs[2].set_xlabel(r"$\eta$")
22 axs[2].set_ylabel("N")
23 axs[2].set_yscale('log')
24 plt.rcParams['font.family'] = 'Times New Roman'
25 plt.tight_layout()

1 pt_no_cut = sum_eletron_p4.pt
2 eta_no_cut = sum_eletron_p4.eta
3 mass_no_cut = sum_eletron_p4.mass
4
5 pt_cut = di_eletron_p4_cut.pt
6 eta_cut = di_eletron_p4_cut.eta
7 mass_cut = di_eletron_p4_cut.mass
8 # Total de eventos
9 total_events_no_cut = int(di_eletron_mass_hist_no_cut.sum())
10 total_events_cut = int(di_eletron_mass_hist_cut.sum())

1 di_eletron_mass_hist_no_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^-e^+}$ [GeV]'))
2 di_eletron_mass_hist_no_cut.fill(mass_no_cut)
3
4 di_eletron_mass_hist_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^+e^-e^+}$ [GeV]'))
5 di_eletron_mass_hist_cut.fill(mass_cut)
6
7 # Criando os histogramas de pt
8 pt_hist_no_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
9 pt_hist_no_cut.fill(pt_no_cut)
10
11 pt_hist_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
12 pt_hist_cut.fill(pt_cut)

```

```

13
14 # Criando os histogramas de eta
15 eta_hist_no_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
16 eta_hist_no_cut.fill(eta_no_cut)
17
18 eta_hist_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
19 eta_hist_cut.fill(eta_cut)

1   fig, axs = plt.subplots(1, 3, figsize=(25, 10))
2
3   # Plot da massa
4   hep.histplot(di_eletron_mass_hist, histtype='fill', ax=axs[0], color='grey', edgecolor='white')
5   axs[0].set_xlabel(r"M (GeV)")
6   axs[0].set_ylabel("N")
7   axs[0].set_ylabel("N", labelpad=20)
8   axs[0].set_xlim(80, 100)
9   axs[0].set_yscale('log')
10
11  # Plot pt
12  hep.histplot(pt_hist, histtype='fill', ax=axs[1], color='grey', edgecolor='white')
13  axs[1].text(200, 10**5, f"{total_events_cut} eventos", color='black', fontsize=15)
14  axs[1].set_xlabel(r"$p_{T}$ (GeV)")
15  axs[1].set_ylabel("N")
16  axs[1].set_yscale('log')
17
18  # Plot eta
19  hep.histplot(eta_hist, histtype='fill', ax=axs[2], color='grey', edgecolor='white')
20  axs[2].set_xlabel(r"$\eta$")
21  axs[2].set_ylabel("N")
22  axs[2].set_yscale('log')
23  # Ajustando o layout e mostrando o gráfico
24  plt.rcParams['font.family'] = 'Times New Roman'
25  plt.savefig('hist_corte.png')
26  plt.tight_layout()

1   pt_no_cut = sum_eletron_p4.pt
2   eta_no_cut = sum_eletron_p4.eta
3   mass_no_cut = sum_eletron_p4.mass
4
5   pt_cut = di_eletron_p4_cut.pt
6   eta_cut = di_eletron_p4_cut.eta
7   mass_cut = di_eletron_p4_cut.mass
8   # Total de eventos
9   total_events_no_cut = int(di_eletron_mass_hist_no_cut.sum())
10  total_events_cut = int(di_eletron_mass_hist_cut.sum())
11
12  # Definindo os bins
13  bins = np.linspace(80, 100, 100)

1  ###
2  di_eletron_mass_hist_no_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^{-}e^{+}}$ [GeV]'))
3  di_eletron_mass_hist_no_cut.fill(mass_no_cut)
4
5  di_eletron_mass_hist_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^{-}e^{+}}$ [GeV]'))
6  di_eletron_mass_hist_cut.fill(mass_cut)
7
8  ###
9  pt_hist_no_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_{T}$ [GeV]'))
10 pt_hist_no_cut.fill(pt_no_cut)
11
12 pt_hist_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_{T}$ [GeV]'))

```

```

13 pt_hist_cut.fill(pt_cut)
14
15 ###
16 eta_hist_no_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
17 eta_hist_no_cut.fill(eta_no_cut)
18
19 eta_hist_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
20 eta_hist_cut.fill(eta_cut)

1 fig, axs = plt.subplots(1, 3, figsize=(25, 10))
2
3 # Massa (com e sem corte)
4 hep.histplot([di_eletron_mass_hist_no_cut, di_eletron_mass_hist_cut], histtype='fill',
5             , ax=axs[0], label=['Original', 'Corte'], color=['grey', 'maroon'], edgecolor=['
6             white', 'white'], alpha=0.7)
7 axs[0].set_xlabel(r"M (GeV)")
8 axs[0].set_ylabel("N")
9 axs[0].set_xlim(80, 100)
10 axs[0].set_yscale('log')
11 axs[0].legend()
12
13 # pt (com e sem corte)
14 hep.histplot([pt_hist_no_cut, pt_hist_cut], histtype='fill', ax=axs[1], label=['
15 Original', 'Corte'], color=['grey', 'maroon'], edgecolor=['white', 'white'], alpha
16 =0.7)
17 axs[1].text(200, 10**5, f"{total_events_no_cut} eventos originais", color='black',
18             fontsize=15)
19 axs[1].text(200, 10**5.2, f"{total_events_cut} eventos com o corte", color='black',
20             fontsize=15)
21 axs[1].set_xlabel(r"$p_T$ (GeV)")
22 axs[1].set_ylabel("N")
23 axs[1].set_yscale('log')
24 axs[1].legend()
25
26 # eta (com e sem corte)
27 hep.histplot([eta_hist_no_cut, eta_hist_cut], histtype='fill', ax=axs[2], label=['
28 Original', 'Corte'], color=['grey', 'maroon'], edgecolor=['white', 'white'], alpha
29 =0.7)
30 axs[2].set_xlabel(r"$\eta$")
31 axs[2].set_ylabel("N")
32 axs[2].set_yscale('log')
33 axs[2].legend()
34
35 # Ajustando o layout e mostrando o gráfico
36 plt.rcParams['font.family'] = 'Times New Roman'
37 plt.savefig('hist_comparacao.png')
38 plt.tight_layout()

```