Introdução à Análise de dados em FAE **(27 de Novembro de 2024)**

Lista 7

Professores: Dilson Damião, Eliza Melo e Maurício Thiel *Nome:* José Gonçalves Chaves Junior

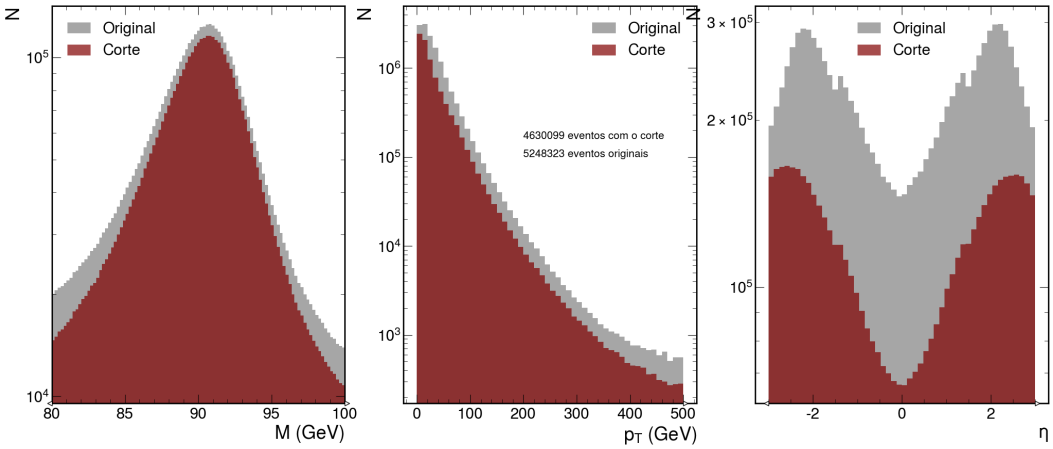

Figura 1: Histogramas para massa, momento transversal e /*eta* para a amostra original e a amostra com os cortes propostos na atividade.

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.

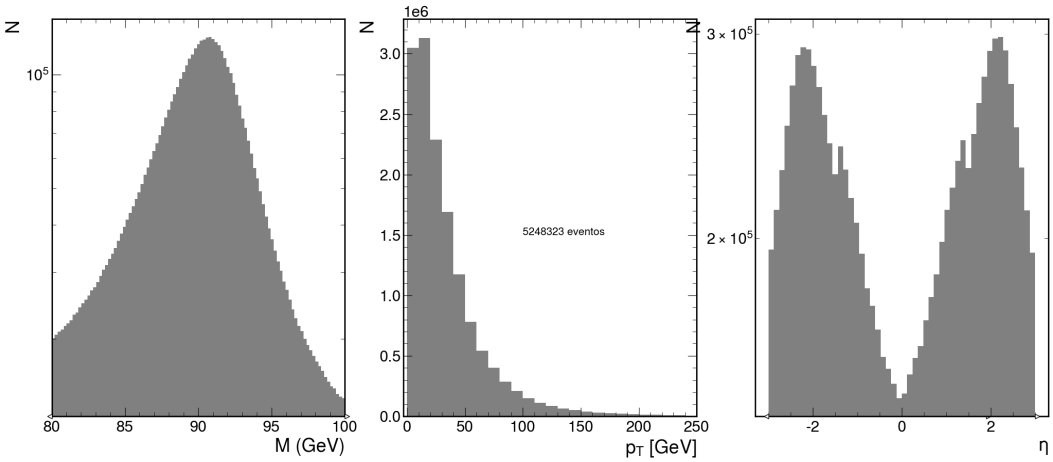

Figura 2: Histograma dos parâmetros físicos da amostra.

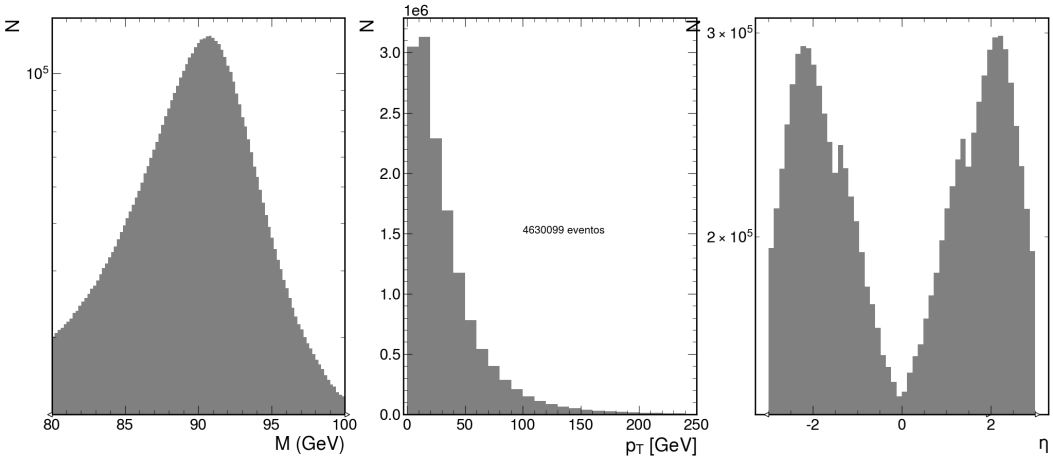

Figura 3: Histograma dos parâmetros físicos da amostra com os cortes.

```
import ROOT
import uproot
import awkward as ak
import matplotlib.pyplot as plt
import vector
import numpy as np
import glob
import hist
import matplotlib.colors as mcolors
import matplotlib.font_manager
```

```
    files = [
    "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/UL2016_MiniAODv2_NanoAODv9-
        v1/100000/*.root:Events",
    "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/UL2016_MiniAODv2_NanoAODv9-
        v1/1010000/*.root:Events",
    "/opendata/eos/opendata/cms/Run2016G/DoubleEG/NANOAOD/UL2016_MiniAODv2_NanoAODv9-
        v1/250000/*.root:Events"
]

var = [
    "nElectron",
    "Electron_pt",
    "Electron_eta",
    "Electron_phi",
    "Electron_mass",
    "Electron_charge"
]

arrays = uproot.concatenate(files, filter_name=var)

nElectron = arrays["nElectron"]
eletron_pt = arrays["Electron_pt"]
eletron_eta = arrays["Electron_eta"]
eletron_phi = arrays["Electron_phi"]
eletron_mass = arrays["Electron_mass"]
eletron_charge = arrays["Electron_charge"]
```

```
mask_di_eletron = nElectron == 2

eletron_p4 = vector.zip({'pt': eletron_pt,
                         'eta': eletron_eta,
```

```
                        'phi': eletron_phi,
                        'mass': eletron_mass})
#cargas dos eletrons
two_eletron_p4 = eletron_p4[mask_di_eletron]
two_eletron_charges = eletron_charge[mask_di_eletron]
opposite_sign_eletron_mask = two_eletron_charges[:, 0] != two_eletron_charges[:, 1]
two_eletron_p4 = two_eletron_p4[opposite_sign_eletron_mask]
```

```
    eletron_p4_0 = two_eletron_p4[:, 0]
eletron_p4_1 = two_eletron_p4[:, 1]
sum_eletron_p4 = eletron_p4_0 + eletron_p4_1
pt = sum_eletron_p4.pt
eta = sum_eletron_p4.eta
mass = sum_eletron_p4.mass
bins = np.linspace(80, 100, 100)
###
di_eletron_mass_hist = hist.Hist(hist.axis.Variable(bins, label=r'M (GeV)'))
di_eletron_mass_hist.fill(mass)
###
pt_hist = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ (GeV)'))
pt_hist.fill(pt)
###
eta_hist = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
eta_hist.fill(eta)
```

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

```
    #Realizando os cortes para pt e eta
    sel_pt_1 = eletron_p4_0.pt > 20
sel_pt_2 =  eletron_p4_1.pt > 20

sel_eta_1 = eletron_p4_0.eta < np.abs(2.4)
sel_eta_2 =  eletron_p4_1.eta < np.abs(2.4)

sel_pt = sel_pt_1 & sel_pt_2 & sel_eta_1 & sel_eta_2

eletron_p4_sel = two_eletron_p4[sel_pt]
eletron_p4_1 = eletron_p4_sel[:,0]
eletron_p4_2 = eletron_p4_sel[:,1]
di_eletron_p4_cut = eletron_p4_1 + eletron_p4_2
```

```
    pt = di_eletron_p4_cut.pt
eta = di_eletron_p4_cut.eta
```

```
mass = di_eletron_p4_cut.mass
bins = np.linspace(80, 100, 100)
###
di_eletron_mass_hist = hist.Hist(hist.axis.Variable(bins, label=r'M (GeV)'))
di_eletron_mass_hist.fill(mass)
###
pt_hist = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ (GeV)'))
pt_hist.fill(pt)
###
eta_hist = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
eta_hist.fill(eta)
```

```
    hep.style.use('CMS')
fig, axs = plt.subplots(1, 3, figsize=(25, 10))

# plot da massa
hep.histplot(di_eletron_mass_hist, histtype='fill', ax=axs[0],color='grey',edgecolor=
    'white')
axs[0].set_xlabel(r"M (GeV)")
axs[0].set_ylabel("N")
axs[0].set_ylabel("N", labelpad=20)
axs[0].set_xlim(80, 100)
axs[0].set_yscale('log')

# plot pt
hep.histplot(pt_hist, histtype='fill', ax=axs[1],color='grey',edgecolor='white')
axs[1].text(200,10**5,f"{total_events} eventos",color='black',fontsize=15)
axs[1].set_xlabel(r"$p_T$ (GeV)")
axs[1].set_ylabel("N")
axs[1].set_yscale('log')

# plot eta
hep.histplot(eta_hist, histtype='fill', ax=axs[2],color='grey',edgecolor='white')
axs[2].set_xlabel(r"$\eta$")
axs[2].set_ylabel("N")
axs[2].set_yscale('log')
plt.rcParams['font.family'] = 'Times New Roman'
plt.tight_layout()
```

```
    pt_no_cut = sum_eletron_p4.pt
eta_no_cut = sum_eletron_p4.eta
mass_no_cut = sum_eletron_p4.mass

pt_cut = di_eletron_p4_cut.pt
eta_cut = di_eletron_p4_cut.eta
mass_cut = di_eletron_p4_cut.mass
# Total de eventos
total_events_no_cut = int(di_eletron_mass_hist_no_cut.sum())
total_events_cut = int(di_eletron_mass_hist_cut.sum())
```

```
    di_eletron_mass_hist_no_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e
        ^{-}e^{+}}$ [GeV]'))
di_eletron_mass_hist_no_cut.fill(mass_no_cut)

di_eletron_mass_hist_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^{-}e^{+}}
    $ [GeV]'))
di_eletron_mass_hist_cut.fill(mass_cut)

# Criando os histogramas de pt
pt_hist_no_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
pt_hist_no_cut.fill(pt_no_cut)

pt_hist_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
pt_hist_cut.fill(pt_cut)
```

```

# Criando os histogramas de eta
eta_hist_no_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
eta_hist_no_cut.fill(eta_no_cut)

eta_hist_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
eta_hist_cut.fill(eta_cut)
```

```
    fig, axs = plt.subplots(1, 3, figsize=(25, 10))

# Plot da massa
hep.histplot(di_eletron_mass_hist, histtype='fill', ax=axs[0],color='grey',edgecolor='white')
axs[0].set_xlabel(r"M (GeV)")
axs[0].set_ylabel("N")
axs[0].set_ylabel("N", labelpad=20)
axs[0].set_xlim(80, 100)
axs[0].set_yscale('log')

# Plot pt
hep.histplot(pt_hist, histtype='fill', ax=axs[1],color='grey',edgecolor='white')
axs[1].text(200,10**5,f"{total_events_cut} eventos",color='black',fontsize=15)
axs[1].set_xlabel(r"$p_T$ (GeV)")
axs[1].set_ylabel("N")
axs[1].set_yscale('log')

# Plot eta
hep.histplot(eta_hist, histtype='fill', ax=axs[2],color='grey',edgecolor='white')
axs[2].set_xlabel(r"$\eta$")
axs[2].set_ylabel("N")
axs[2].set_yscale('log')
# Ajustando o layout e mostrando o gr fico
plt.rcParams['font.family'] = 'Times New Roman'
plt.savefig('hist_corte.png')
plt.tight_layout()
```

```
    pt_no_cut = sum_eletron_p4.pt
eta_no_cut = sum_eletron_p4.eta
mass_no_cut = sum_eletron_p4.mass

pt_cut = di_eletron_p4_cut.pt
eta_cut = di_eletron_p4_cut.eta
mass_cut = di_eletron_p4_cut.mass
# Total de eventos
total_events_no_cut = int(di_eletron_mass_hist_no_cut.sum())
total_events_cut = int(di_eletron_mass_hist_cut.sum())

# Definindo os bins
bins = np.linspace(80, 100, 100)
```

```
###
di_eletron_mass_hist_no_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^{-}e^{+}}$ [GeV]'))
di_eletron_mass_hist_no_cut.fill(mass_no_cut)

di_eletron_mass_hist_cut = hist.Hist(hist.axis.Variable(bins, label=r'$M_{e^{-}e^{+}}$ [GeV]'))
di_eletron_mass_hist_cut.fill(mass_cut)

###
pt_hist_no_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
pt_hist_no_cut.fill(pt_no_cut)

pt_hist_cut = hist.Hist(hist.axis.Regular(50, 0, 500, label=r'$p_T$ [GeV]'))
```

```
pt_hist_cut.fill(pt_cut)

###
eta_hist_no_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
eta_hist_no_cut.fill(eta_no_cut)

eta_hist_cut = hist.Hist(hist.axis.Regular(50, -3, 3, label=r'$\eta$'))
eta_hist_cut.fill(eta_cut)
```

```
fig, axs = plt.subplots(1, 3, figsize=(25, 10))

# Massa (com e sem corte)
hep.histplot([di_eletron_mass_hist_no_cut, di_eletron_mass_hist_cut], histtype='fill', ax=axs[0], label=['Original', 'Corte'], color=['grey','maroon'],edgecolor=['white','white'],alpha=0.7)
axs[0].set_xlabel(r"M (GeV)")
axs[0].set_ylabel("N")
axs[0].set_xlim(80, 100)
axs[0].set_yscale('log')
axs[0].legend()

# pt (com e sem corte)
hep.histplot([pt_hist_no_cut, pt_hist_cut], histtype='fill', ax=axs[1], label=['Original', 'Corte'], color=['grey','maroon'],edgecolor=['white','white'],alpha=0.7)
axs[1].text(200,10**5,f"{total_events_no_cut} eventos originais",color='black', fontsize=15)
axs[1].text(200,10**5.2,f"{total_events_cut} eventos com o corte",color='black', fontsize=15)
axs[1].set_xlabel(r"$p_T$ (GeV)")
axs[1].set_ylabel("N")
axs[1].set_yscale('log')
axs[1].legend()

# eta (com e sem corte)
hep.histplot([eta_hist_no_cut, eta_hist_cut], histtype='fill', ax=axs[2], label=['Original', 'Corte'],color=['grey','maroon'],edgecolor=['white','white'],alpha=0.7)
axs[2].set_xlabel(r"$\eta$")
axs[2].set_ylabel("N")
axs[2].set_yscale('log')
axs[2].legend()

# Ajustando o layout e mostrando o gr fico
plt.rcParams['font.family'] = 'Times New Roman'
plt.savefig('hist_comparacao.png')
plt.tight_layout()
```