Hands on - análise de dados básica

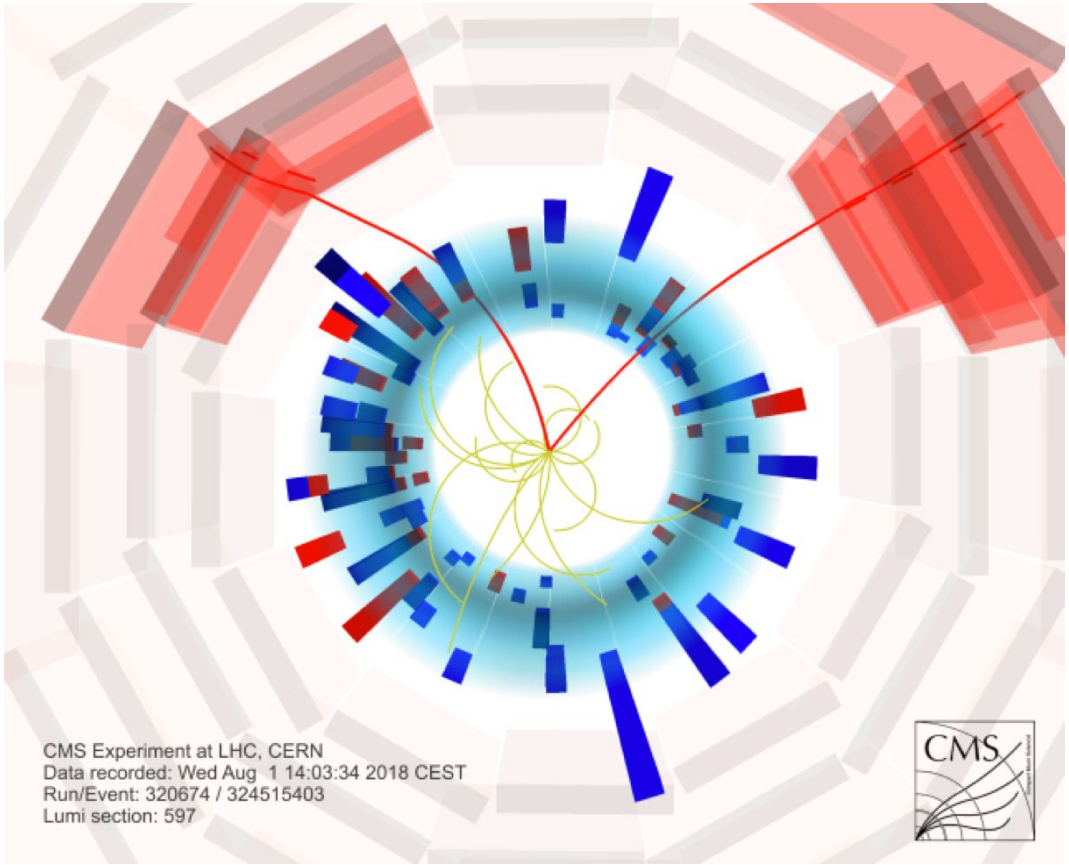
objetivos

- Realizar uma análise de dados simples
  - visualizar dados
  - manipular ntuple (TTree) de dados
  - produzir, processar e exibir histogramas de dados
    - selecionar sinal de física
    - plotar distribuições cinemáticas
  - critérios de seleção
  - extrair parâmetros físicos dos dados
    - medir os yields do sinal realizando um ajuste de probabilidade (likelihood fit) aos dados
    - incertezas estatísticas

Parte I
https://colab.research.google.com/drive/1sHo06MJh_yzv-n_s2WzfaMM6Jd9yPWey?usp=sharing
Parte II
https://colab.research.google.com/drive/1EmIghSAaxVjUp1HZAiOabbBmjZqZwW8b?usp=sharing

https://github.com/INCT-CERN-Brasil/Escola2024/blob/main/dia_01/ExercicioManipulandoDados_EscolaINCT_ParteI_v1.ipynb

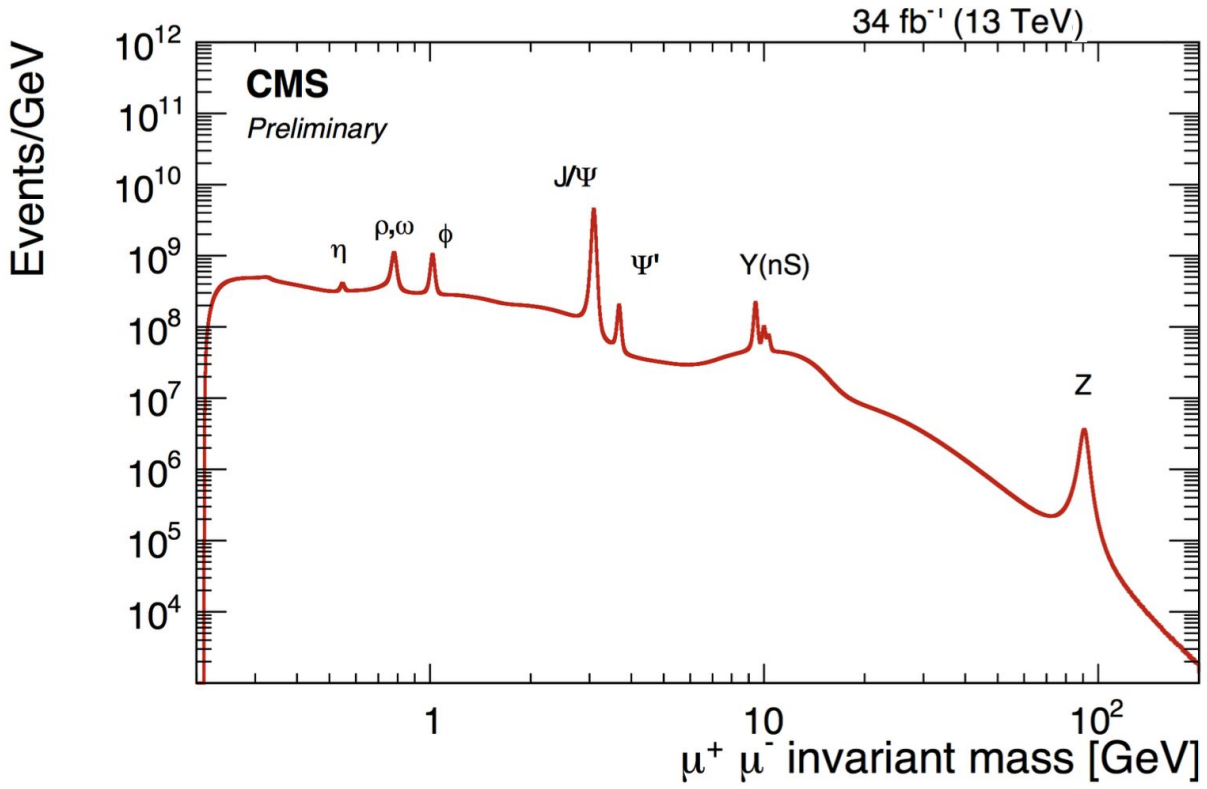
eventos de dimúons (x→μμ)



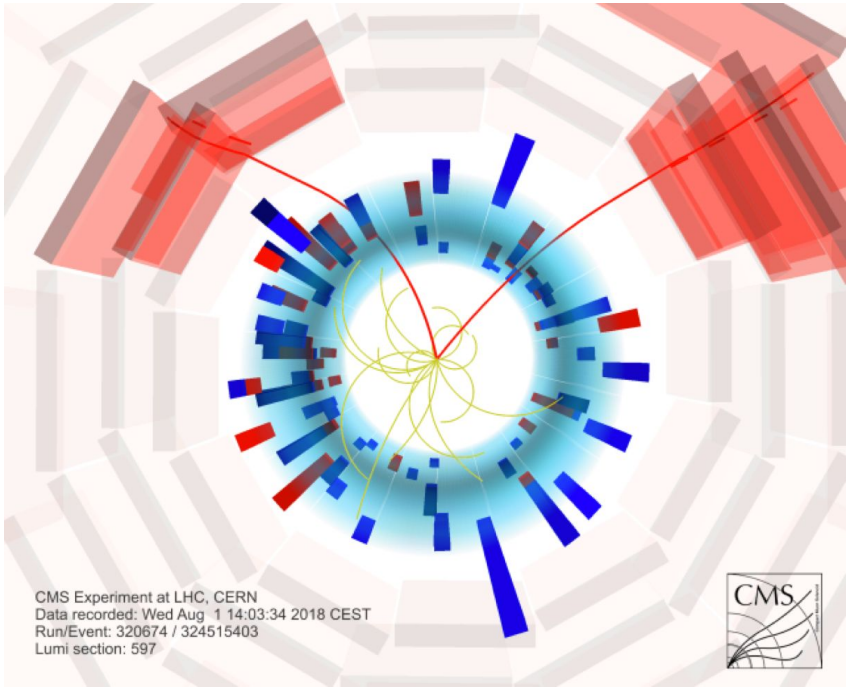
CMS-PHO-EVENTS-2022-031

o espetro de dimuon ($X \rightarrow \mu^+\mu^-$)

Décadas de descobertas em física de partículas... em um único gráfico!



massa invariante de dimúons



indentificação de partículas
- sinal nas câmaras de múons
→ é um múon!
➠ m = m(μ) ~ 106MeV/c2

trajetória da partícula
- câmaras de múons, mas especialmente o silicon tracker
➠ momentum linear , p=(px,py,pz)

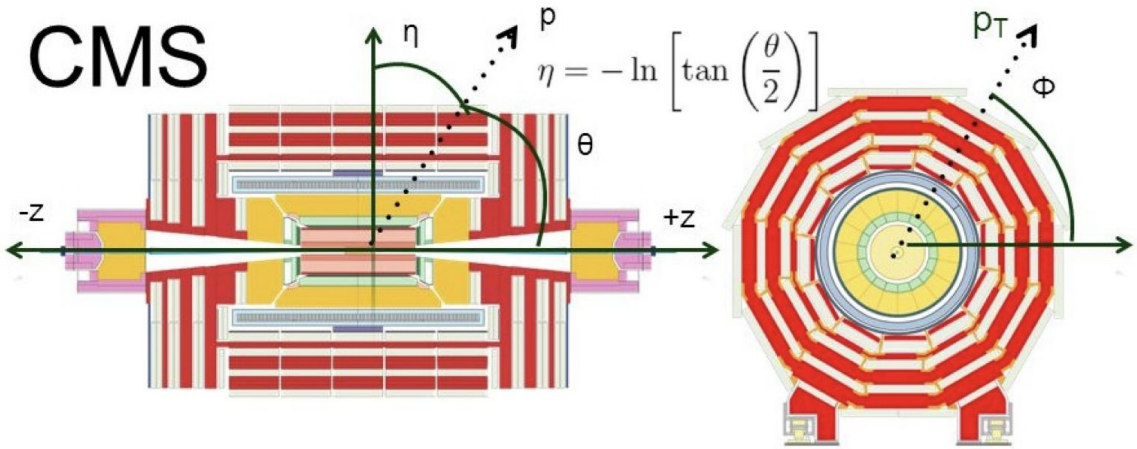
➠ 4-momentum de cada múon:

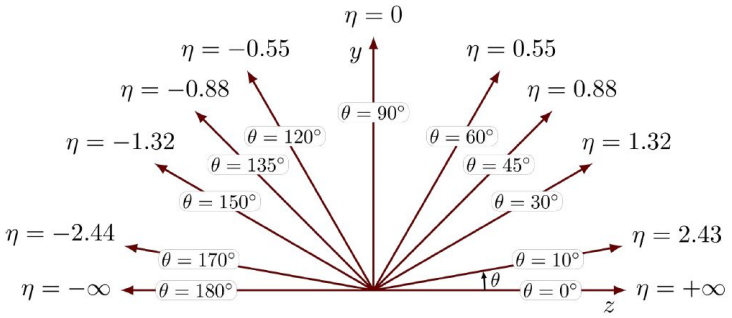
$\mathbf{P}=(E,p_x,p_y,p_z)$ ou $\mathbf{P}=(p_T,\boldsymbol{\eta},\phi,M)$ para E>>m

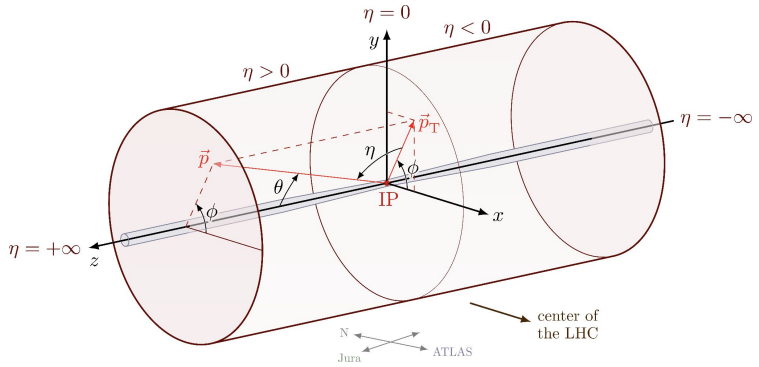
➠ para o par de dimuon: $\mathbf{P}_{\mu\mu}=\mathbf{P}_{\mu 1}+\mathbf{P}_{\mu 2}$

➠ massa invariante: $M_{\mu\mu} = (P_{\mu 1}+P_{\mu 2})^2$

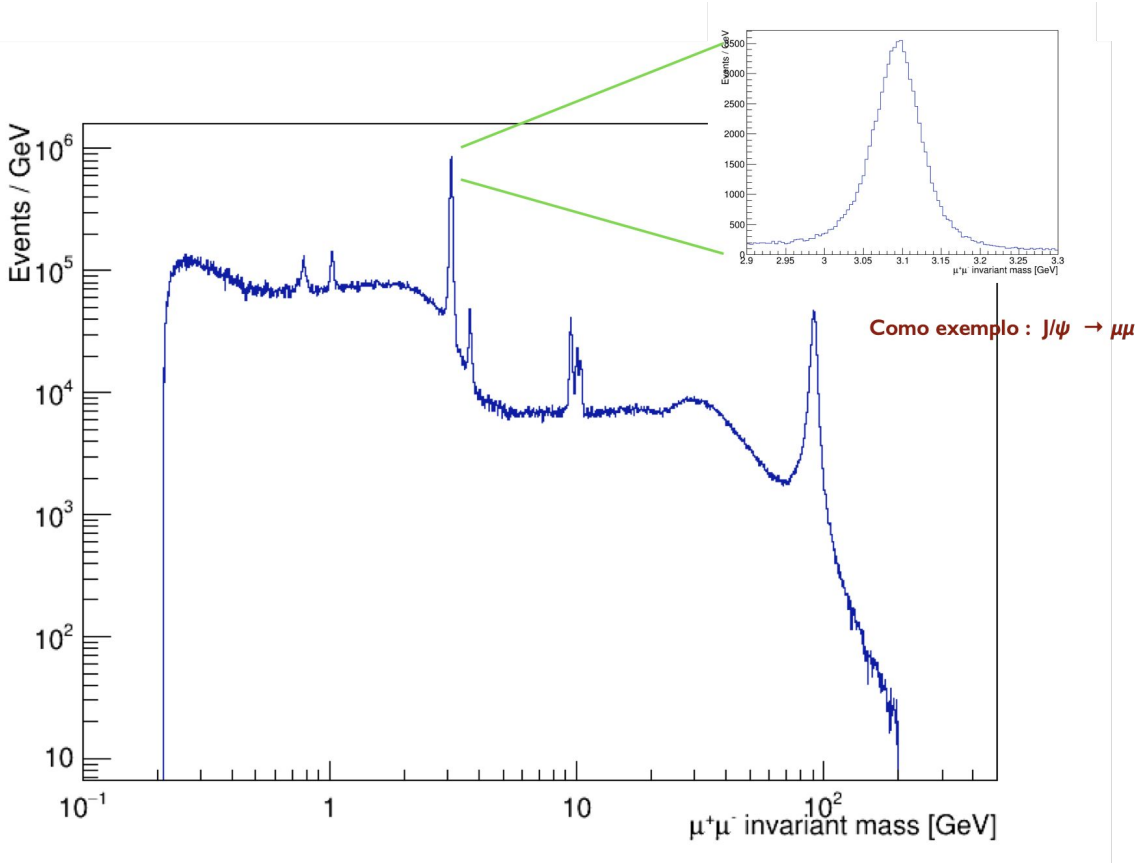
outras variáveis





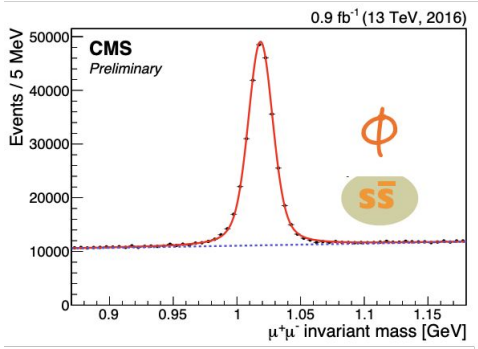


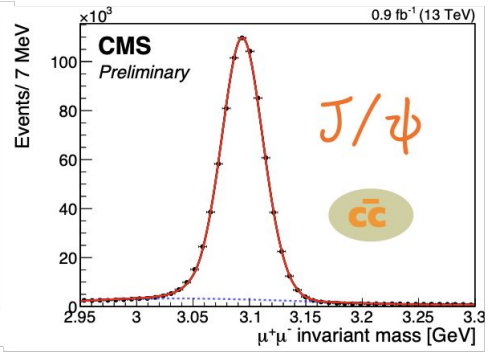
o espectro do dimuon reconstruído

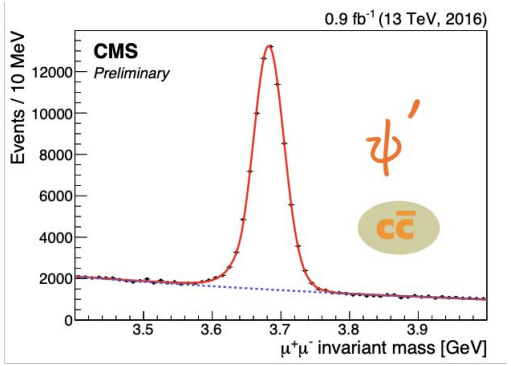


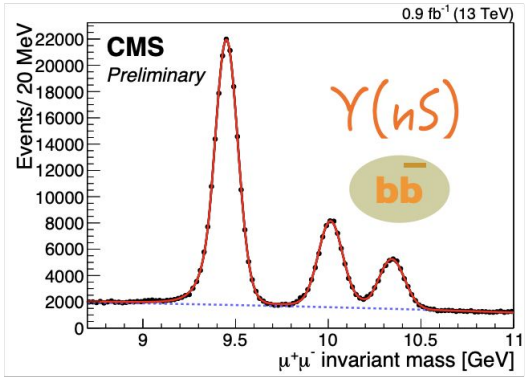
Como exemplo : $J/\psi \rightarrow \mu\mu$

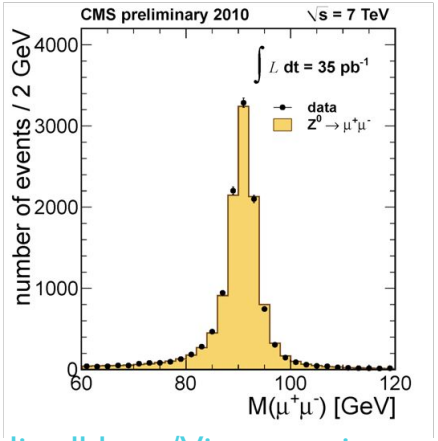
o que são os picos?





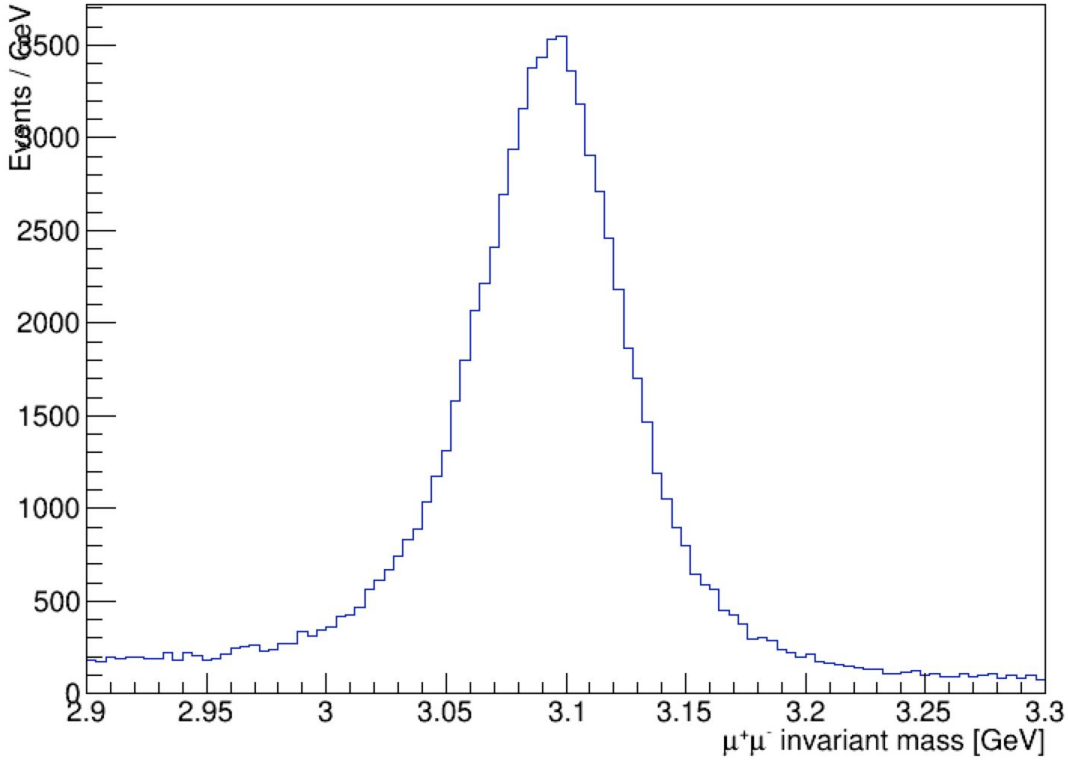




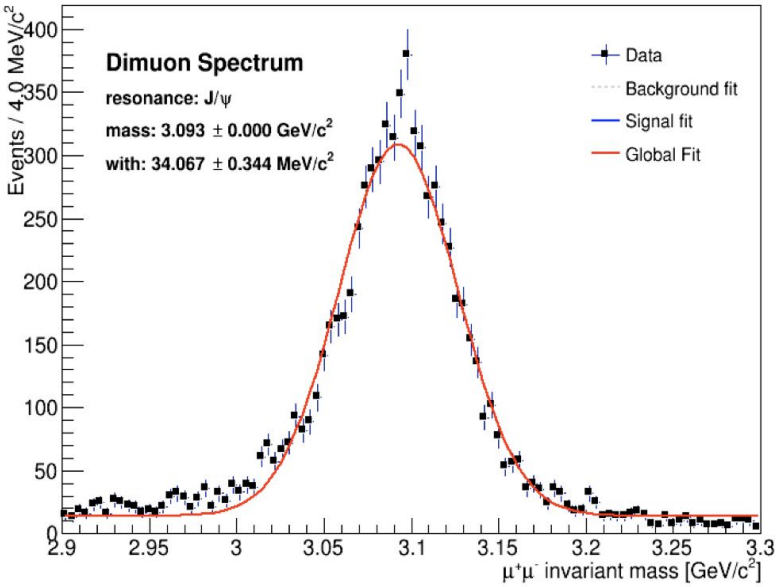


suas propriedades podem ser consultadas aqui: https://pdglive.lbl.gov/Viewer.action

zoom... e selecione um sinal



ajuste aos dados



Inspeção da qualidade do ajuste

o ajuste pode ser melhorado?

dica: a radiação do estado final (μ→μγ) pode distorcer a forma.

Passos:

Estabelecer um modelo de ajuste

sinal: gaussiana

fundo: polinomial

Extrair parâmetros do sinal

yields ($N \pm \boldsymbol{\sigma}_N$), massa ($m \pm \boldsymbol{\sigma}_m$)

Estimar as incertezas sistemáticos

a escolha do modelo afeta os resultados medidos?

quantificar as variações sistemáticas empregando diferentes modelos

Apresentar as medidas finais ($N \pm \boldsymbol{\sigma}_{esta} \pm \boldsymbol{\sigma}_{sis}$)

informações no pdg

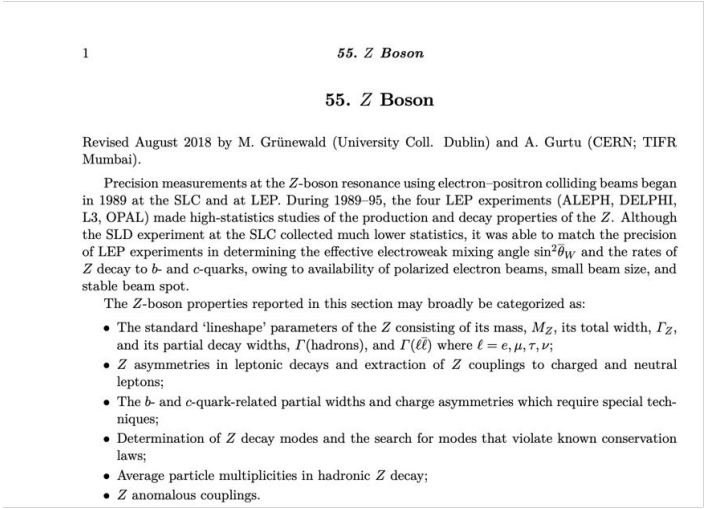
Z $J = 1$

See related reviews:

Z Boson	PDF
Anomalous $ZZ\gamma$, $Z\gamma\gamma$, and ZZV Couplings	PDF
Anomalous W/Z Quartic Couplings (QGCs)	PDF

▸ Expand all sections

Z MASS	91.1876 ± 0.0021 GeV
Z WIDTH	2.4955 ± 0.0023 GeV
▸ **AVERAGE PARTICLE MULTIPLICITIES IN HADRONIC Z DECAY**	
Z HADRONIC POLE CROSS SECTION	41.4802 ± 0.0325 nb



1 *55. Z Boson*

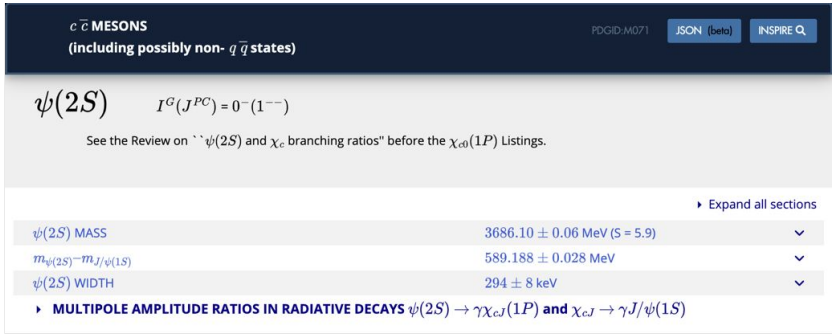
55. Z Boson

Revised August 2018 by M. Grünewald (University Coll. Dublin) and A. Gurtu (CERN; TIFR Mumbai).

Precision measurements at the Z-boson resonance using electron–positron colliding beams began in 1989 at the SLC and at LEP. During 1989–95, the four LEP experiments (ALEPH, DELPHI, L3, OPAL) made high-statistics studies of the production and decay properties of the Z. Although the SLD experiment at the SLC collected much lower statistics, it was able to match the precision of LEP experiments in determining the effective electroweak mixing angle $\sin^2\hat{\theta}_W$ and the rates of Z decay to b- and c-quarks, owing to availability of polarized electron beams, small beam size, and stable beam spot.

The Z-boson properties reported in this section may broadly be categorized as:

- The standard 'lineshape' parameters of the Z consisting of its mass, M_Z, its total width, Γ_Z, and its partial decay widths, Γ(hadrons), and $\Gamma(\ell\bar{\ell})$ where $\ell = e, \mu, \tau, \nu$;
- Z asymmetries in leptonic decays and extraction of Z couplings to charged and neutral leptons;
- The b- and c-quark-related partial widths and charge asymmetries which require special techniques;
- Determination of Z decay modes and the search for modes that violate known conservation laws;
- Average particle multiplicities in hadronic Z decay;
- Z anomalous couplings.



$c\bar{c}$ **MESONS**
(including possibly non- $q\bar{q}$ **states)**

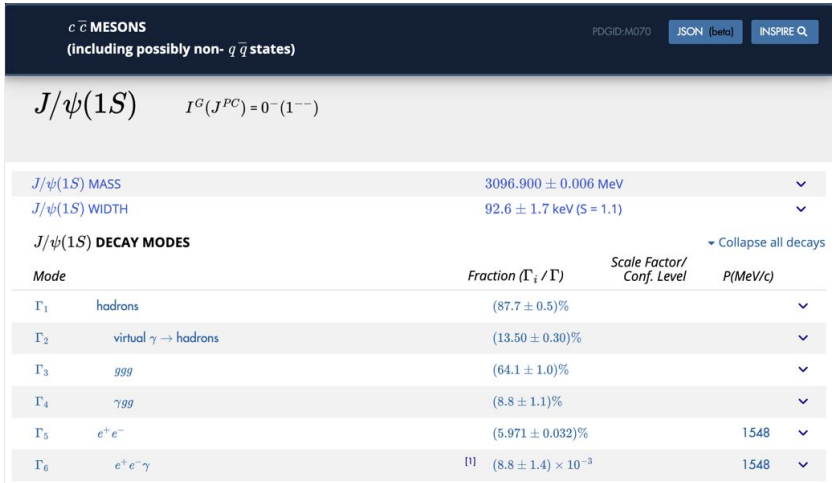
PDGID:M071 JSON (beta) INSPIRE

$\psi(2S)$ $I^G(J^{PC}) = 0^-(1^{--})$

See the Review on ``$\psi(2S)$ and χ_c branching ratios'' before the $\chi_{c0}(1P)$ Listings.

▸ Expand all sections

$\psi(2S)$ MASS	3686.10 ± 0.06 MeV (S = 5.9)
$m_{\psi(2S)} - m_{J/\psi(1S)}$	589.188 ± 0.028 MeV
$\psi(2S)$ WIDTH	294 ± 8 keV
▸ **MULTIPOLE AMPLITUDE RATIOS IN RADIATIVE DECAYS $\psi(2S) \to \gamma\chi_{cJ}(1P)$ and $\chi_{cJ} \to \gamma J/\psi(1S)$**	



$c\bar{c}$ **MESONS**
(including possibly non- $q\bar{q}$ **states)**

PDGID:M070 JSON (beta) INSPIRE

$J/\psi(1S)$ $I^G(J^{PC}) = 0^-(1^{--})$

$J/\psi(1S)$ MASS	3096.900 ± 0.006 MeV
$J/\psi(1S)$ WIDTH	92.6 ± 1.7 keV (S = 1.1)

$J/\psi(1S)$ DECAY MODES ▾ Collapse all decays

Mode		Fraction (Γ_i/Γ)	Scale Factor/ Conf. Level	P(MeV/c)
Γ_1	hadrons	$(87.7 \pm 0.5)\%$		
Γ_2	virtual $\gamma \to$ hadrons	$(13.50 \pm 0.30)\%$		
Γ_3	ggg	$(64.1 \pm 1.0)\%$		
Γ_4	γgg	$(8.8 \pm 1.1)\%$		
Γ_5	e^+e^-	$(5.971 \pm 0.032)\%$		1548
Γ_6	$e^+e^-\gamma$	[1] $(8.8 \pm 1.4) \times 10^{-3}$		1548

Análise de forma Colunar - uproot

- O uproot é um módulo Python que permite ler e escrever arquivos no formato ROOT.
- A dependência obrigatória, além do próprio Python, é o NumPy, que é a biblioteca mais popular para manipulação de arrays de dados em Python.
- Ele é capaz de processar grandes volumes de dados de forma rápida.


```
import uproot
file = uproot.open("data.root")
file
```

Saída: `<ReadOnlyDirectory '/' at 0x(some hexadecimal number here)>`

Análise de forma Colunar - uproot

Como navegar e acessar o conteúdo do arquivo ROOT usando o uproot?

Ex.:

Listar o conteúdo do arquivo:

```
file.keys()
```

```
# Saída: ['Events;1']
```

Verificar o tipo de objeto no arquivo:

```
file.classnames()
```

```
# Saída: {'Events;1': 'TTree'}
```

Acessar o conteúdo do arquivo (file['key']):

```
file['Events']
```

```
# Saída: <TTree 'Events' (6 branches) at 0x(hexadecimal number)>
```

Como acessar e manipular a TTree do arquivo ROOT usando o uproot?

```
tree = file['Events']
```

```
tree.keys()
```

```
#Saída: ['nMuon', 'Muon_pt', 'Muon_eta', 'Muon_phi', 'Muon_mass',
'Muon_charge']
```

```
tree.arrays()
```

```
#Saída: <Array [{nMuon: 2, Muon_pt: [10.8, ... -1, 1]}] type='100000 *
uint32,...'>
```

```
branches = tree.arrays()
```

```
branches['nMuon']
```

```
#Saída: <Array [2, 2, 1, 4, 4, 3, ... 0, 3, 2, 3, 2, 3] type='100000 *
uint32'>
```

Quando usamos tree.arrays(), ele retorna um objeto Awkward Array.

Como acessar e manipular a TTree do arquivo ROOT usando o uproot?

```
branches['Muon_pt']
```

```
#Saída: <Array [[10.8, 15.7], ... 11.4, 3.08, 4.97]] type='100000 * var * float32'>
```

São arrays irregulares (jagged array), pois o número de entradas (**'nMuon'**) varia por evento.

Para acessar os dados de um evento específico, você pode indexar o array como faria com qualquer array normal.

```
branches['Muon_pt'][0]
```

```
#Saída:<Array [10.8, 15.7] type='2 * float32'>
```

Análise de forma Colunar - Awkward Array

Os dados em altas energias são estruturados de forma irregulares:

- Não pode ser representado como uma tabela “retangular”.
  - Em diferentes eventos - números variáveis de objetos, como μ/e/jatos/…

A representação de Arrays Irregurares (Jagged Arrays):

Muon pt: table

Event 1	*40.2*	*25.6*	*10.2*
Event 2	*71.1*	*35.7*	
Event 3	*52.3*		
Event 4	*34.5*	*15.7*	

Muon pt: jagged array

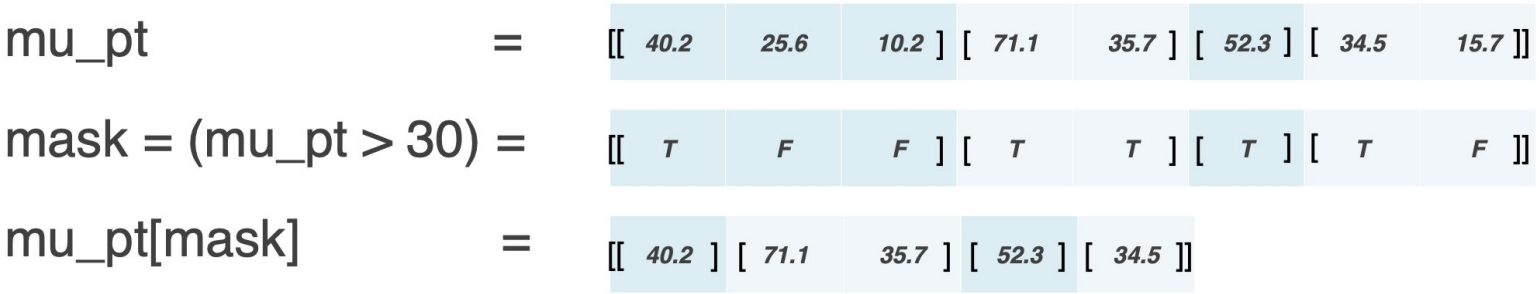
[[*40.2* *25.6* *10.2*] [*71.1* *35.7*] [*52.3*] [*34.5* *15.7*]]

Event 1 Event 2 Event 3 Event 4

Análise de forma Colunar - Awkward Array

Como aplicamos seleções em arrays irregulares (jagged arrays)?

- Usando máscaras.



A máscara foi aplicada a cada múon em cada evento de maneira vetorizada.

Análise de forma Colunar - Awkward Array

```
1 !pip install awkward
2 import awkward as ak
3 import numpy as np
```

Show hidden output

```
[9]  1 mu_pt = ak.Array([[40.2, 25.6, 10.2],  # Evento 1: três múons
     2                   [71.1, 35.7],        # Evento 2: dois múons
     3                   [52.3],              # Evento 3: um múon
     4                   [34.5, 15.7]])       # Evento 4: dois múons
```

```
[10] 1 mask_mupt = mu_pt > 30
     2 mu_pt_sel = mu_pt[mask_mupt]
```

```
[11] 1 print("Array original:", mu_pt)
     2 print("\nMáscara aplicada:")
     3 print(mask_mupt)
     4 print("\nArray após a aplicação da máscara:")
     5 print(mu_pt_sel)
```

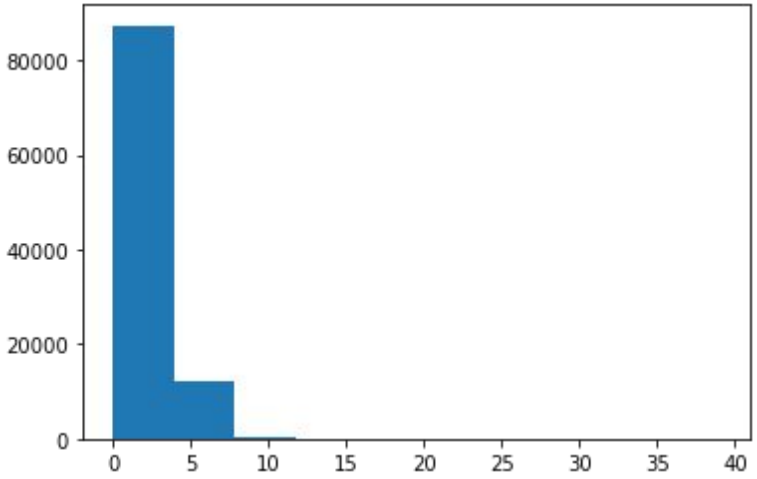
```
Array original: [[40.2, 25.6, 10.2], [71.1, 35.7], [52.3], [34.5, 15.7]]

Máscara aplicada:
[[True, False, False], [True, True], [True], [True, False]]

Array após a aplicação da máscara:
[[40.2], [71.1, 35.7], [52.3], [34.5]]
```

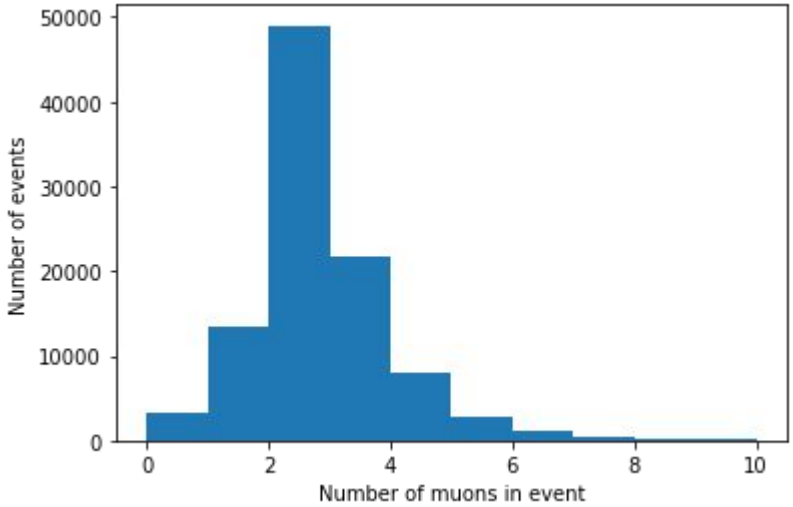
Análise de forma Colunar - Matplotlib


```
import matplotlib.pyplot as plt
plt.hist(branches['nMuon'])
#Saída:(array([8.7359e+04, 1.2253e+04, 3.5600e+02,
2.8000e+01, 2.0000e+00,1.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 1.0000e+00]),array([ 0.,3.9,
7.8, 11.7, 15.6, 19.5, 23.4, 27.3, 31.2, 35.1, 39. ]),
  <a list of 10 Patch objects>)
```



Análise de forma Colunar - Matplotlib

```
plt.hist(branches['nMuon'], bins=10, range=(0, 10))
plt.xlabel('Number of muons in event')
plt.ylabel('Number of events')
plt.show()
```

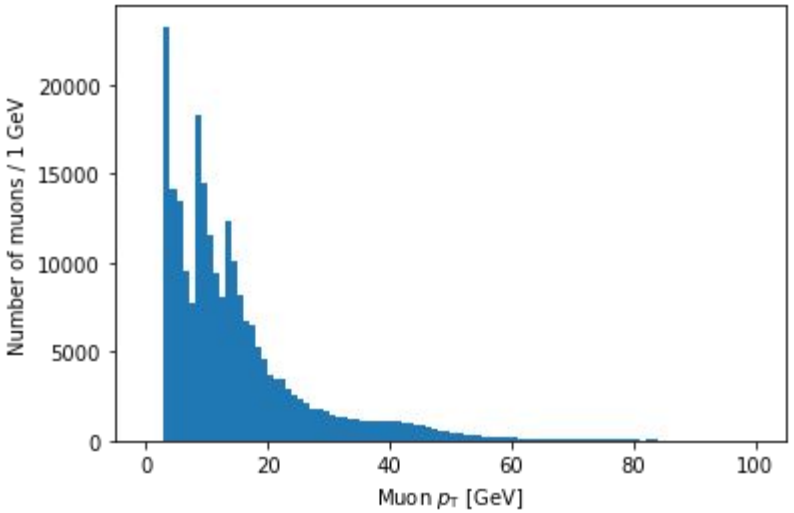


Análise de forma Colunar - Matplotlib

Para fazer um histograma de um array irregular (jagged array), precisamos converter o array 2D para 1D usando o Awkward Array com a função ak.flatten(). Ex.:

```
import awkward as ak

plt.hist(ak.flatten(branches['Muon_pt']), bins=100, range=(0, 100))
plt.xlabel('Muon $p_{\mathrm{T}}$ [GeV]')
plt.ylabel('Number of muons / 1 GeV')
plt.show()
```



Análise de forma Colunar - Contagem

```
len(branches['Muon_pt']) #número total de eventos
#Saída: 100000
len(ak.flatten(branches['Muon_pt'])) #número total de múons
#Saída: 235286
```

Análise de forma Colunar - Seleções

Para seleções:

```
branches['nMuon']== 1 # o resultado é array boleano
#Saída:<Array [False, False, True, ... False, False] type='100000 * bool'>
```

Para contar quantos eventos atendem a essa condição:

```
single_muon_mask = branches['nMuon']== 1
np.sum(single_muon_mask)  #Saída: 13447
```

Para selecionar o p_T dos múons em eventos com exatamente um múon, usamos a máscara single_muon_mask como um índice:

```
branches['Muon_pt'][single_muon_mask]
#Saída:<Array [[3.28], [3.84], ... [13.3], [9.48]] type='13447 * var * float32'>
len(branches['Muon_pt'][single_muon_mask])
#Saída: 13447
```

Análise de forma Colunar - Seleções

Construindo os 4-momenta dos múons:

```
two_muons_mask = branches['nMuon']== 2
```

Com o pacote Vector, podemos construir os quadrimomentos dos múons a partir das variáveis p_T, η, ϕ, e massa:

```
import vector
muon_p4 = vector.zip({'pt': branches['Muon_pt'],
                      'eta': branches['Muon_eta'],
                      'phi': branches['Muon_phi'],
                      'mass': branches['Muon_mass']})

two_muons_p4 = muon_p4[two_muons_mask]
```

Agora podemos acessar propriedades como p_T, η, ϕ, E, e mass dos múons filtrados.

Análise de forma Colunar - Seleções

Para calcular a massa invariante dos dois múons em cada evento, somamos os quadrivetores. Primeiro, selecionamos o quadrimomento do primeiro múon em cada evento usando o slicing [:, 0]:

```
first_muon_p4 = two_muons_p4[:, 0] #primeiro múon de cada evento
second_muon_p4 = two_muons_p4[:, 1] #segundo múon de cada evento
```

Agora podemos somar os quadrivetores e calcular a massa invariante dos dois múons para cada evento.

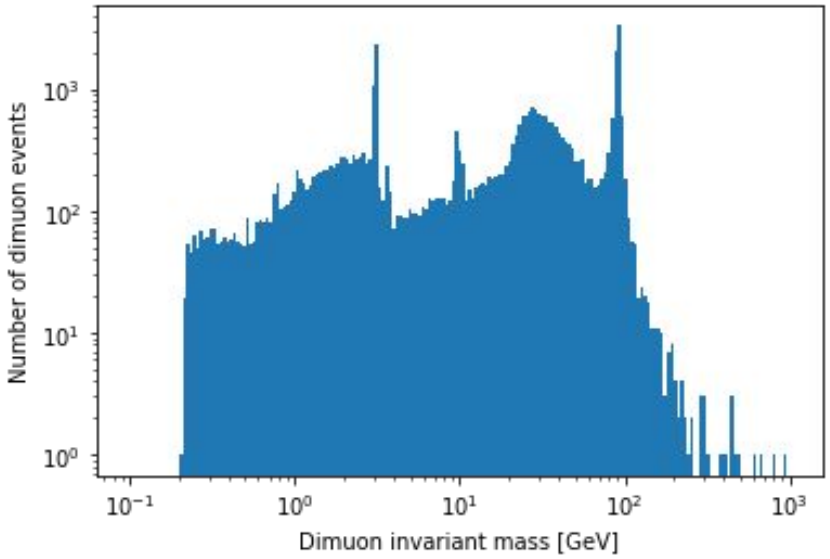
```
sum_p4 = first_muon_p4 + second_muon_p4 #o resultado é um array 1D
two_muons_charges = branches['Muon_charge'][two_muons_mask]
opposite_sign_muons_mask = two_muons_charges[:, 0] != two_muons_charges[:, 1]
```

Por fim, aplicamos essa máscara ao somatório dos quadrivetores para obter os eventos de múons de sinais opostos:

```
dimuon_p4 = sum_p4[opposite_sign_muons_mask]
```

Massa invariante dos pares de múons de cargas opostas

```
plt.hist(dimuon_p4.mass, bins=np.logspace(np.log10(0.1), np.log10(1000), 200))
plt.xlabel('Dimuon invariant mass [GeV]')
plt.ylabel('Number of dimuon events')
plt.xscale('log')
plt.yscale('log')
plt.show()
```



Referências

- N. Leonardo
- 16th Portuguese Language Teacher Programme
- Particle physics with the computer

Backup

Multiple files and large files

```
uproot.concatenate({'*.root': 'Events'})
```

Another method is to iterate through the entries across all files:

```
for events in uproot.iterate({'*.root': 'Events'}, step_size=100):
    # do something with events
```

HLT_IsoMu24: o múon deve ter um momento transversal (pT) superior a 24 GeV. Além de passar o corte de pT, o múon também deve satisfazer um critério de isolamento.

O isolamento é uma seleção que busca excluir múons que possam estar associados a jatos de partículas ou a decaimentos hadrônicos. Ele é feito somando o momento das partículas em um cone ao redor do múon; quanto menor essa soma em relação ao pT do múon, mais isolado ele é.

O que é o múon?

- Um múon é um lépton. É o primo mais pesado do elétron.
  - Massa do múon ~105,7 MeV ⇒ m_μ/m_e ~200.
  - O múon possui carga elétrica (+,-).
  - Participa das interações eletromagnéticas e fracas, mas não das interações fortes.
- No que os múons decaem?
  - BR(μ→eν̄ν) ~ 1. Como isso funciona?
  - E quanto ao decaimento μ→eγ?
- Se os múons decaem, como conseguimos observá-los no CMS?
  - A vida média do múon é $\tau \sim 2{,}2 \times 10^{-6}$ s ⇒ $c\tau \sim 660$ m!
  - Qual é a distância típica entre o ponto de colisão e o início das câmaras de múons?
- Como a vida média do múon se compara à de outras partículas?
  - Por que a vida média do múon é tão longa?
  - Qual é o impacto disso?

Por que nos importamos com os múons?

- Múons são produzidos em interações fracas
  - Se um múon está presente no estado final, um W, Z ou H deve ter estado envolvido.
  - O estudo da física eletrofraca é uma das principais missões do LHC e do CMS.
  - Ou uma nova física está próxima. A busca por física além do Modelo Padrão é a outra grande missão.
- Múons podem ser um indicador útil de física potencialmente interessante
  - Se conseguirmos diferenciar entre múons de W, Z ou H (múons pontuais) e aqueles de outras fontes.
  - Durante este exercício, discutiremos maneiras de fazer isso e como os múons podem ser usados em uma medida ou busca.
- O detector CMS foi projetado para detectar e identificar múons com alta eficiência e medir seu momento e carga em uma ampla faixa de parâmetros cinemáticos.

identificação de múons

Identificação de Múons para Diferenciar Fontes: A identificação de múons é essencial para distinguir entre diferentes origens de múons. Em um experimento como o CMS, múons podem ser produzidos por vários processos, como decaimentos de quarks pesados, decaimentos de partículas instáveis ou mesmo como parte do ruído de fundo. A escolha de critérios de identificação específicos ajuda a diferenciar entre esses casos.

IDs Baseados em Partícula Fluida (Particle Flow, PF):

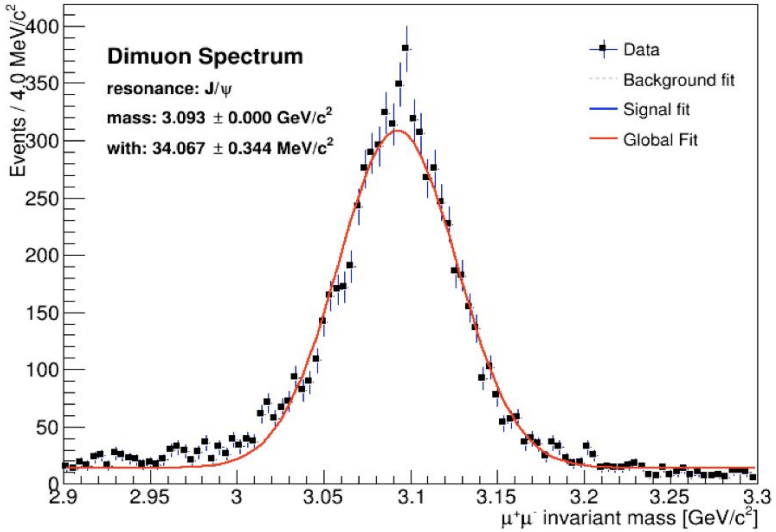
- **Loose ID**: Focado em uma alta eficiência para todos os tipos de múons, sem restrições rigorosas. É útil para quando se quer captar o máximo de múons possíveis, inclusive aqueles de fontes secundárias.
- **Medium ID** (CutBased e MVA-based): Alta eficiência para múons prompt (múons originados diretamente da colisão, não de decaimentos secundários) e decaimentos de quarks pesados (como o quark charm e o bottom).
- **Tight ID** (CutBased e MVA-based): Alta eficiência especificamente para múons prompt, sendo mais seletivo e usado quando há necessidade de maior certeza de que o múon seja "prompt".

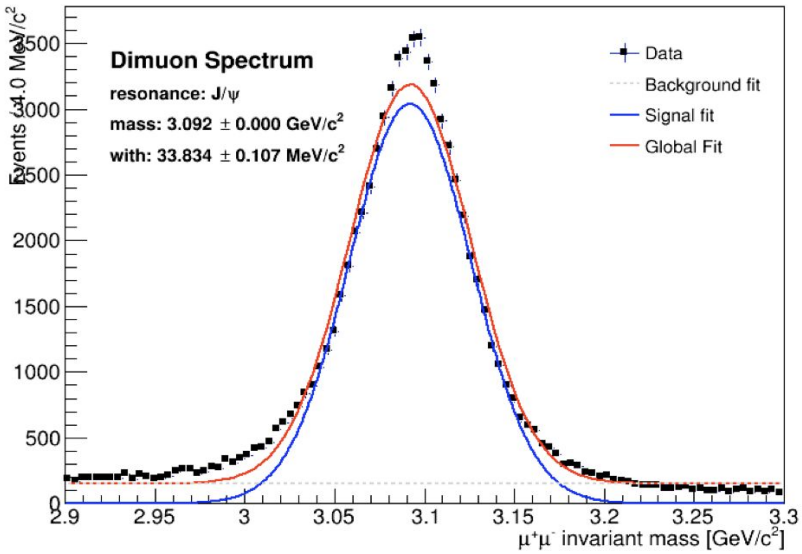
Outros Tipos de IDs Específicos:

- **High pT ID**: Projetado para selecionar múons de alto momento transversal (pT>200 GeVp_T > 200 \, \text{GeV}pT>200GeV) com alta eficiência e qualidade de reconstrução. Esses múons não são mais minimamente ionizantes (MIPs) e perdem energia ao passar pelo sistema de detecção de múons, o que torna a identificação um pouco mais complexa.
- **SoftID/SoftMVA ID**: Específico para a física de b-quarks (b-física), onde é comum lidar com múons de baixa energia resultantes de decaimentos de quarks pesados.
- **MVA ID**: Baseado em uma técnica de Machine Learning (MVA - Multi-Variate Analysis) para fornecer alta eficiência em identificar múons prompt e decaimentos de quarks pesados.
- **Prompt MVA**: Um identificador que combina critérios de isolamento, parâmetros de impacto e outras variáveis para selecionar múons prompt com grande pureza, essencial quando é necessário reduzir a presença de ruído de fundo.

ajuste aos dados

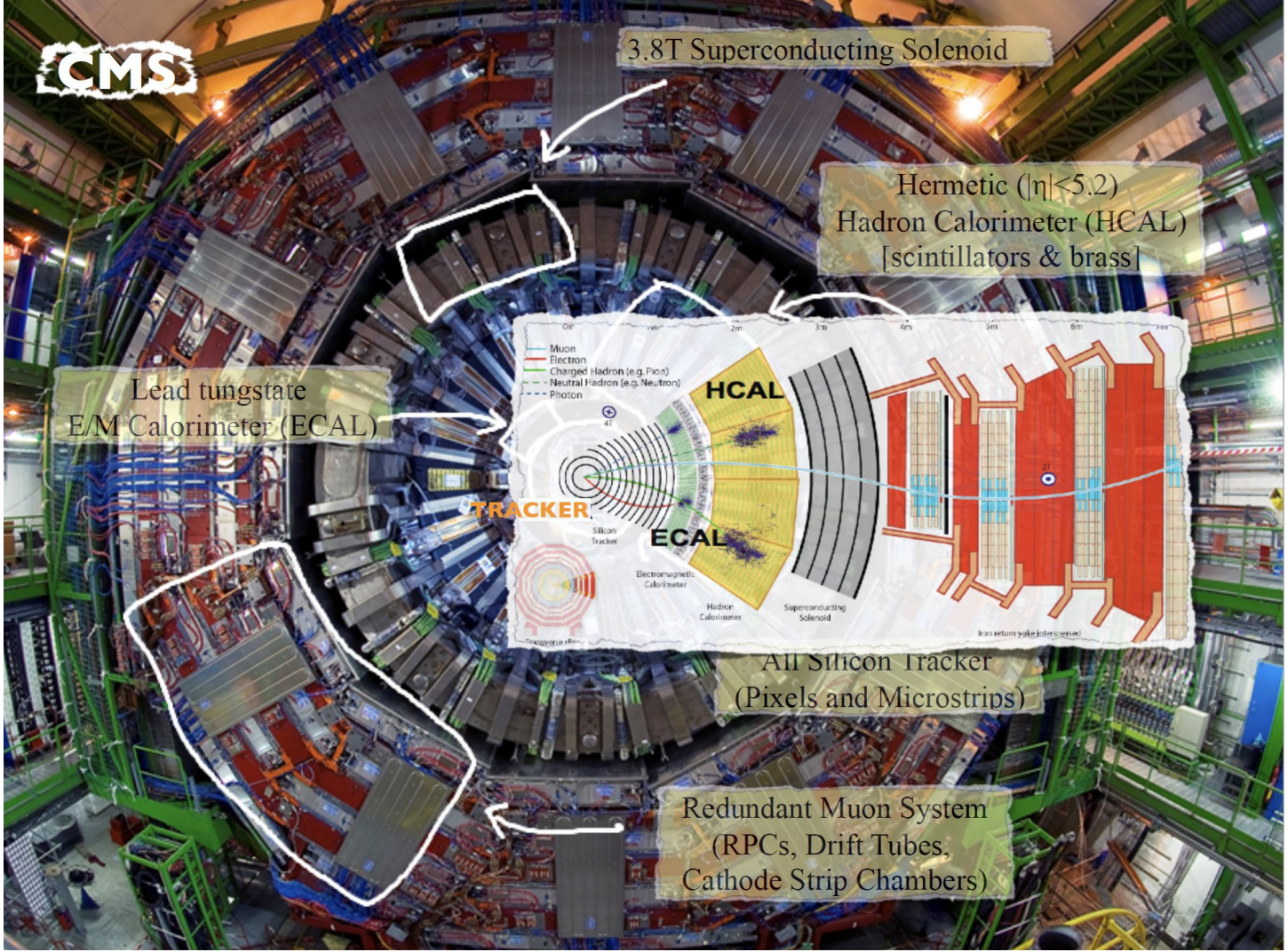
Modelo simplificado: Gaussiana (sinal) + polinominal (bckg)





Como poderíamos melhorar?
dica: final-state radiation

e essa distribuição com yield alto, não está bem descrita, por quê?


CMS
3.8T Superconducting Solenoid
Hermetic (|η|<5.2)
Hadron Calorimeter (HCAL)
[scintillators & brass]
Lead tungstate
E/M Calorimeter (ECAL)
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
HCAL
TRACKER
ECAL
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
All Silicon Tracker
(Pixels and Microstrips)
Redundant Muon System
(RPCs, Drift Tubes,
Cathode Strip Chambers)

Campo magnético: no coração de uma experiência

Cargas elétricas em movimento são sensíveis aos campos magnéticos

A partir da trajectória de uma partícula sujeita a B:

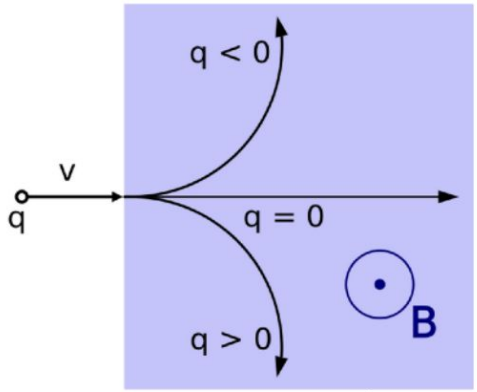
- *direcção*

  ⇒ medição da carga eléctrica

- *raio de curvatura*

  ⇒ medida de momento (conhecida a massa)

  ⇒ medida de massa (conhecida a velocidade)



$$p \cos \lambda = 0.3\, z\, B\, R$$

momento linear [GeV/c]

ângulo de inclinação

carga eléctrica [e]

intensidade de campo magnético [T]

Raio de curvatura [L]