Introdução à Análise de dados em FAE **(12/07/2024)**

EXERCÍCIOS - ANÁLISE DE DADOS

Professores: Sandro Fonseca, Maurício Thiel, Eliza Melo, Diego Machado *Name:* Isis Prazeres Mota

EXERCICIO 1

Cópia do arquivo.

EXERCICIO 2

Seleção das variáveis:

```
root [3] data_Weighted->Scan("B_m:B_charge:s31_DTF:s23_DTF")
```

EXERCICIO 3

Calculando e exibindo a massa invariante do méson B e verificando se esse valor está de acordo com a massa invariante correta do $B^{\pm}$.

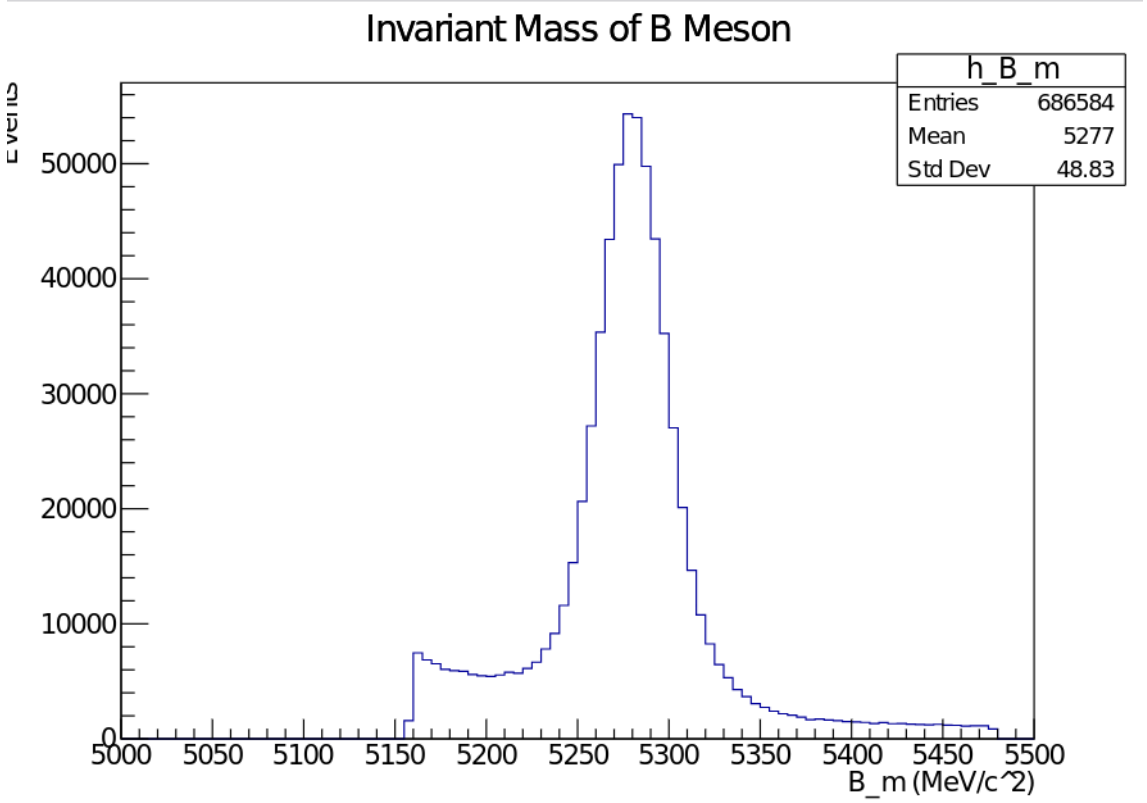
```
    // Variaveis para armazenar os dados
    double B_m;

    // Configura a branch
    tree->SetBranchAddress("B_m", &B_m);

    // Cria histograma
    TH1F *h_B_m = new TH1F("h_B_m", "Invariant Mass of B Meson; B_m (MeV/c^2); Events
        ", 100, 5000, 5500);

    // Loop sobre os eventos
    Long64_t nentries = tree->GetEntries();
    for (Long64_t i = 0; i < nentries; i++) {
        tree->GetEntry(i);
        h_B_m->Fill(B_m);
    }

    // Calcula a media e o desvio padrao
    double mean = h_B_m->GetMean();
    double std_dev = h_B_m->GetStdDev();

    // Verifica se a massa invariante esta de acordo com a massa esperada do B+-
    bool in_agreement = (mean >= B_m_expected - std_dev) && (mean <= B_m_expected +
        std_dev);

    std::cout << "Media da massa invariante do meson B: " << mean << " MeV/c^2" <<
        std::endl;
    std::cout << "Desvio padrao da massa invariante do meson B: " << std_dev << " MeV
        /c^2" << std::endl;
    std::cout << "A massa invariante do meson B esta de acordo com a massa esperada
        do B+-? " << (in_agreement ? "Sim" : "Nao") << std::endl;

    // Cria canvas para desenhar o histograma
    TCanvas *c1 = new TCanvas("c1", "Invariant Mass of B Meson", 800, 600);
    h_B_m->Draw();

    // Salva o grafico
```

```
33      c1->SaveAs("B_meson_mass.pdf");
34  }
```

Mostrando os seguintes resultados no terminal:

- Média da massa invariante do méson B : $5277.38 MeV/c^2$
- Desvio padrão da massa invariante do méson B : $48.8346 MeV/c^2$
- A massa invariante do méson B está de acordo com a massa esperada do B±? **Sim**

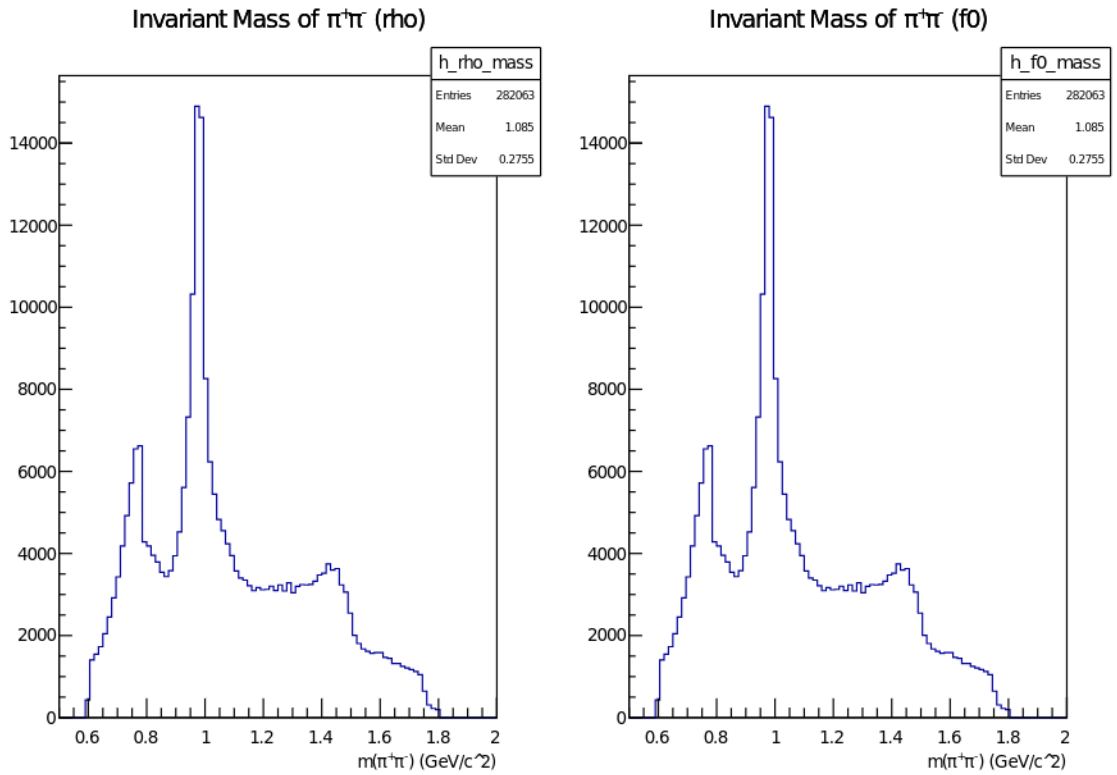
Plot:

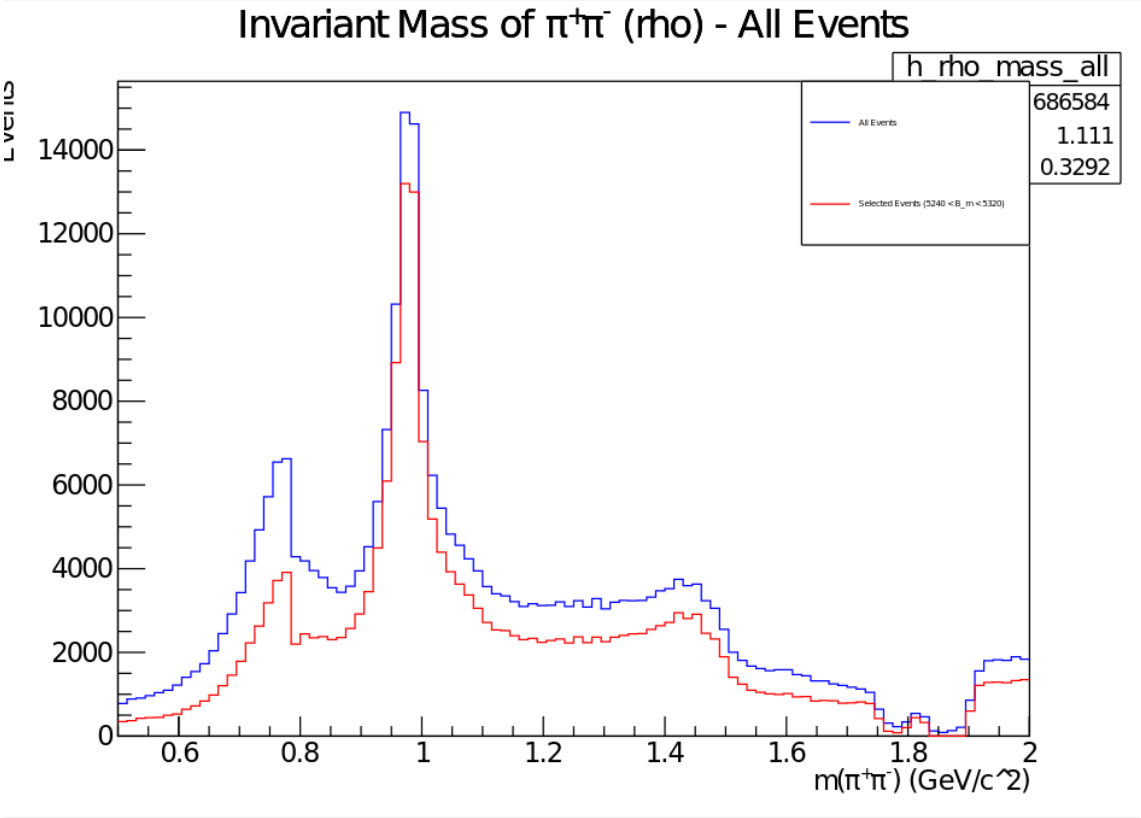


Calculando e exibindo a massa invariante das ressonâncias $\rho(770)$ e $f_0(980)$ usando $\sqrt{s31\_DTF}$ para calcular $m(\pi^+\pi^-)$. Verificando, também, se há alguma mudança na forma das ressonâncias ao selecionar eventos dentro do intervalo $5240 < B\_m[MeV/c^2] < 5320$).

```
// Variaveis para armazenar os dados
    double B_m, s31_DTF;

    // Configura as branches
    tree->SetBranchAddress("B_m", &B_m);
    tree->SetBranchAddress("s31_DTF", &s31_DTF);

    // Cria histogramas para os dois casos
    TH1F *h_rho_mass_all = new TH1F("h_rho_mass_all", "Invariant Mass of #pi^{+}#pi
        ^{-} (rho) - All Events; m(#pi^{+}#pi^{-}) (GeV/c^2); Events", 100, 0.5, 2.0)
        ;
    TH1F *h_rho_mass_selected = new TH1F("h_rho_mass_selected", "Invariant Mass of #
        pi^{+}#pi^{-} (rho) - Selected Events; m(#pi^{+}#pi^{-}) (GeV/c^2); Events",
        100, 0.5, 2.0);

    // Loop sobre os eventos
    Long64_t nentries = tree->GetEntries();
    for (Long64_t i = 0; i < nentries; i++) {
        tree->GetEntry(i);

```

```
17          // Calcula a massa invariante
18          double pi_pi_mass = std::sqrt(s31_DTF);
19
20          // Preenche o histograma com todos os eventos
21          h_rho_mass_all->Fill(pi_pi_mass);
22
23          // Preenche o histograma com eventos selecionados
24          if (B_m > 5240 && B_m < 5320) {
25              h_rho_mass_selected->Fill(pi_pi_mass);
26          }
27      }
28
29      // Cria canvas para desenhar os histogramas
30      TCanvas *c1 = new TCanvas("c1", "Invariant Mass of Resonances", 800, 600);
31      h_rho_mass_all->SetLineColor(kBlue);
32      h_rho_mass_selected->SetLineColor(kRed);
33
34      h_rho_mass_all->Draw();
35      h_rho_mass_selected->Draw("SAME");
36
37      // Adiciona legenda
38      TLegend *legend = new TLegend(0.7, 0.7, 0.9, 0.9);
39      legend->AddEntry(h_rho_mass_all, "All Events", "l");
40      legend->AddEntry(h_rho_mass_selected, "Selected Events (5240 < B_m < 5320)", "l")
            ;
41      legend->Draw();
42
43      // Salva o grafico
44      c1->SaveAs("resonances_mass_comparison.pdf");
```



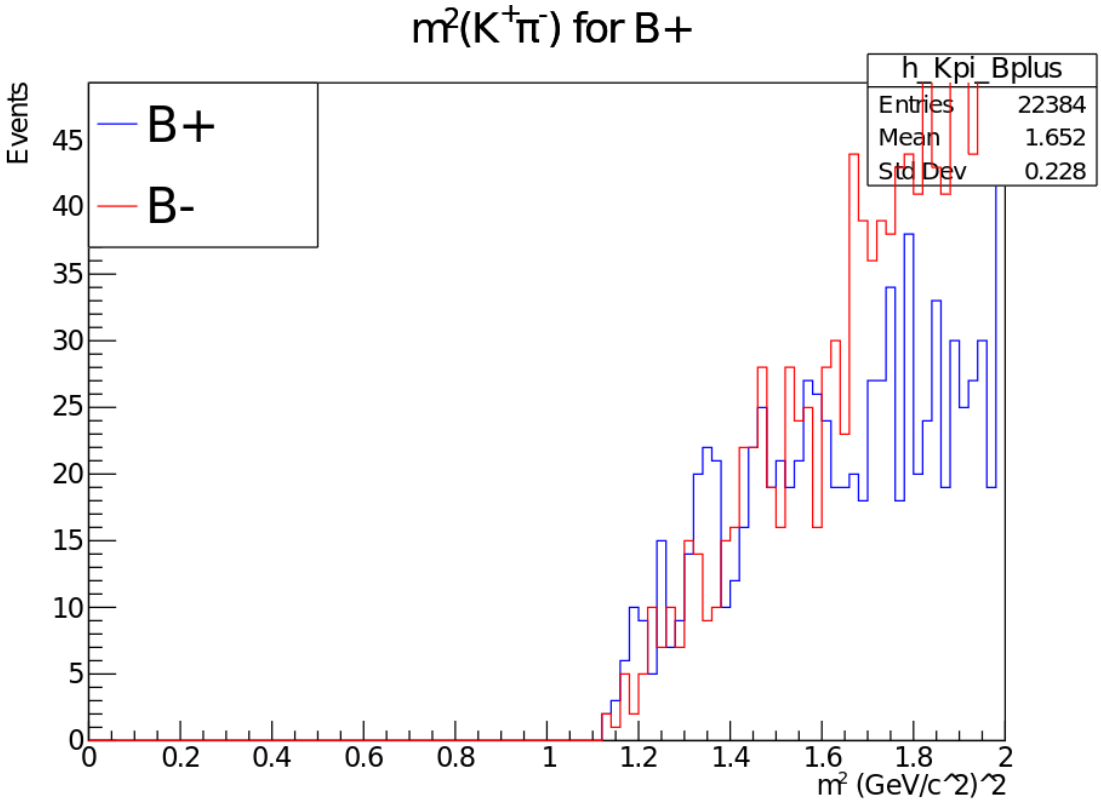


Avaliando a violação direta de CP em decaimentos $B \to PV$ ao selecionar uma janela estreita ao redor da ressonância $\rho(770)$. Calculando $m^2(K^+\pi^-)$ para B^+ e B^-, fazendo um ajuste quadrático e calculando a assimetria A_{CP}.

```
    // Constantes para a janela ao redor da ressonancia rho(770)
const double rho_mass_min = 0.6; // GeV/c^2
const double rho_mass_max = 0.8; // GeV/c^2

\\\\\\\\\\\\\\\int main\\\\\\\\\\\\\\\\\\\\\\\\\\\\

// Variaveis para armazenar os dados
    double B_m, B_charge, s31_DTF, s23_DTF;

    // Configura as branches
    tree->SetBranchAddress("B_m", &B_m);
    tree->SetBranchAddress("B_charge", &B_charge);
    tree->SetBranchAddress("s31_DTF", &s31_DTF);
    tree->SetBranchAddress("s23_DTF", &s23_DTF);

    // Histogramas para B+ e B-
    TH1F *h_Kpi_Bplus = new TH1F("h_Kpi_Bplus", "m^{2}(K^{+}#pi^{-}) for B+; m^{2} (
        GeV/c^2)^2; Events", 100, 0, 2);
    TH1F *h_Kpi_Bminus = new TH1F("h_Kpi_Bminus", "m^{2}(K^{+}#pi^{-}) for B-; m^{2}
        (GeV/c^2)^2; Events", 100, 0, 2);

    // Loop sobre os eventos
    Long64_t nentries = tree->GetEntries();
    for (Long64_t i = 0; i < nentries; i++) {
        tree->GetEntry(i);

        // Calcula m^2(K^+pi^-)
        double m2_Kpi = s23_DTF;

        // Preenche os histogramas dependendo da carga do B
        if (B_charge == 1) { // B+
```

```
            if (sqrt(s31_DTF) >= rho_mass_min && sqrt(s31_DTF) <= rho_mass_max) {
                h_Kpi_Bplus->Fill(m2_Kpi);
            }
        } else if (B_charge == -1) { // B-
            if (sqrt(s31_DTF) >= rho_mass_min && sqrt(s31_DTF) <= rho_mass_max) {
                h_Kpi_Bminus->Fill(m2_Kpi);
            }
        }
    }

    // Cria canvas para desenhar os histogramas
    TCanvas *c1 = new TCanvas("c1", "m^{2}(K^{+}#pi^{-}) for B+ and B-", 800, 600);
    h_Kpi_Bplus->SetLineColor(kBlue);
    h_Kpi_Bminus->SetLineColor(kRed);

    h_Kpi_Bplus->Draw();
    h_Kpi_Bminus->Draw("SAME");

    // Adiciona legenda
    TLegend *legend = new TLegend(0.7, 0.7, 0.9, 0.9);
    legend->AddEntry(h_Kpi_Bplus, "B+", "l");
    legend->AddEntry(h_Kpi_Bminus, "B-", "l");
    legend->Draw();

    // Salva o grafico
    c1->SaveAs("Kpi_mass_squared.pdf");

    // Ajuste quadratico e obtencao dos termos quadraticos
    TF1 *fit_Bplus = new TF1("fit_Bplus", "pol2", 0, 2);
    h_Kpi_Bplus->Fit(fit_Bplus, "Q");

    TF1 *fit_Bminus = new TF1("fit_Bminus", "pol2", 0, 2);
    h_Kpi_Bminus->Fit(fit_Bminus, "Q");

    double p2_Bplus = fit_Bplus->GetParameter(2);
    double p2_Bminus = fit_Bminus->GetParameter(2);

    // Calcula a assimetria A_CP
    double A_CP = (p2_Bminus - p2_Bplus) / (p2_Bminus + p2_Bplus);
    std::cout << "A_CP: " << A_CP << std::endl;
```

Mostrando no terminal o resultado: $A_{CP} : -13.7462$
Isso indica uma assimetria significativa entre as duas configurações de decaimento B^+ e B^-, o que é um indício de violação de paridade e pode ser relacionado a processos de violação de CP.


m²(K⁺π⁻) for B+
h_Kpi_Bplus
Entries 22384
Mean 1.652
Std Dev 0.228
B+
B-
Events
m² (GeV/c^2)^2