

Moller2021_V0

October 15, 2024

```
[1]: import sys
import ROOT as R
R.EnableImplicitMT()
%jsroot on
```

```
[2]: R.gSystem.Load("/data/HPS/lib/libMoller.dylib")
Moller = R.Moller()
print("Version of Moller Class:",Moller.GetVersion())
```

Version of Moller Class: 0.3.3

```
[3]: #data_files_pattern = "/data/HPS/data/physrun2021/Moller_MC/
↳moller_beam_*_minidst.root"
data_files_pattern = "/data/HPS/data/physrun2021/Moller_MC/moller-beam_all.root"
ch_moller = R.TChain("MiniDST")
ch_moller.Add(data_files_pattern)
print("Number of entries in the chain:",ch_moller.GetEntries())
df = R.RDataFrame(ch_moller)
ch_moller2 = R.TChain("MiniDST")
ch_moller2.Add("/data/HPS/data/physrun2021/Moller_MC/moller-beam_all_devel.
↳root")
print("Number of entries in the chain2:",ch_moller2.GetEntries())
df2 = R.RDataFrame(ch_moller2)
ch_moller3 = R.TChain("MiniDST")
ch_moller3.Add("/data/HPS/data/physrun2021/Moller_MC/moller-beam_all_devel2.
↳root")
print("Number of entries in the chain3:",ch_moller3.GetEntries())
df3 = R.RDataFrame(ch_moller3)
```

Number of entries in the chain: 271987

Number of entries in the chain2: 271987

Number of entries in the chain3: 271987

```
[4]: h_v0_type = df.Histo1D(("h_v0_type","Vertex type", 17 ,-0.5,16.5),"v0_type")
h_v0_type2 = df2.Histo1D(("h_v0_type2","Vertex type", 17 ,-0.5,16.5),"v0_type")
h_v0_type3 = df3.Histo1D(("h_v0_type3","Vertex type", 17 ,-0.5,16.5),"v0_type")
```

```

h_v0_n_type4 = df.Define("nt4","int out=0; for(int i=0; i<v0_type.size();
↳++i){if(v0_type[i]==4) ++out;} return out;").Histo1D(("h_v0_n_type4","Number_
↳of type 4 vertexes",6,-0.5,5.5),"nt4")
h_v0_n_type4_2 = df2.Define("nt4","int out=0; for(int i=0; i<v0_type.size();
↳++i){if(v0_type[i]==4) ++out;} return out;").Histo1D(("h_v0_n_type4","Number_
↳of type 4 vertexes",6,-0.5,5.5),"nt4")
h_v0_n_type4_3 = df3.Define("nt4","int out=0; for(int i=0; i<v0_type.size();
↳++i){if(v0_type[i]==4) ++out;} return out;").Histo1D(("h_v0_n_type4","Number_
↳of type 4 vertexes",6,-0.5,5.5),"nt4")

h_v0_time_diff = df.Define("dt","vector<double> out; for(int i=0; i<v0_type.
↳size();++i){if(v0_type[i]==4){double_
↳tr_dtime=(track_time[v0_em_track[i]]-track_time[v0_ep_track[i]]); out.
↳push_back(tr_dtime);} } return out;").Histo1D(("h_v0_time_diff","V0 type 4_
↳Track time diff",500,-100,100),"dt")
h_v0_time_diff2 = df2.Define("dt","vector<double> out; for(int i=0; i<v0_type.
↳size();++i){if(v0_type[i]==4){double_
↳tr_dtime=(track_time[v0_em_track[i]]-track_time[v0_ep_track[i]]); out.
↳push_back(tr_dtime);} } return out;").Histo1D(("h_v0_time_diff","V0 type 4_
↳Track time diff",500,-100,100),"dt")
h_v0_time_diff3 = df3.Define("dt","vector<double> out; for(int i=0; i<v0_type.
↳size();++i){if(v0_type[i]==4){double_
↳tr_dtime=(track_time[v0_em_track[i]]-track_time[v0_ep_track[i]]); out.
↳push_back(tr_dtime);} } return out;").Histo1D(("h_v0_time_diff","V0 type 4_
↳Track time diff",500,-100,100),"dt")

cc00 = R.TCanvas("cc00","Canvas 00", 1200, 1000)
cc00.Divide(2,2)
cc00.cd(1)
h_v0_type.SetStats(0)
h_v0_type.Draw()
h_v0_type2.SetLineColor(R.kGreen)
h_v0_type2.Draw("same")
h_v0_type3.SetLineColor(R.kRed)
h_v0_type3.Draw("same")
leg = R.TLegend(0.6,0.7,0.9,0.9)
leg.AddEntry(h_v0_type.GetPtr(),"Original code","l")
leg.AddEntry(h_v0_type2.GetPtr(),"New code w timecut","l")
leg.AddEntry(h_v0_type3.GetPtr(),"New code w/o timecut","l")
leg.Draw()

pad2 = cc00.cd(2)
# pad2.SetLogy()
h_v0_n_type4.Draw()
h_v0_n_type4_2.SetLineColor(R.kGreen)
h_v0_n_type4_2.Draw("same")

```

```

h_v0_n_type4_3.SetLineColor(R.kRed)
h_v0_n_type4_3.Draw("same")

pad3 = cc00.cd(3)
h_v0_time_diff3.SetStats(0)
h_v0_time_diff3.SetLineColor(R.kRed)
h_v0_time_diff3.Draw("same")
h_v0_time_diff.Draw("same")
h_v0_time_diff2.SetLineColor(R.kGreen)
h_v0_time_diff2.Draw("same")

cc00.Draw()

```

<IPython.core.display.HTML object>

```

[5]: df_uc = Moller.Select_El_Pairs_v0(R.RDF.AsRNode(df), 4, 0., 2.
↳,"electron_pairs") \
    .Define("n_moller_pair","return electron_pairs.size();").
↳Filter("n_moller_pair >= 1")

df3_uc = Moller.Select_El_Pairs_v0(R.RDF.AsRNode(df3), 4, 0., 2.
↳,"electron_pairs") \
    .Define("n_moller_pair","return electron_pairs.size();").
↳Filter("n_moller_pair >= 1")

data_set = [df_uc, df3_uc]
for i in range(len(data_set)):
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,↳
↳"electron_pairs","part_","p4")
    data_set[i] = Moller.Refine_El_Pairs_1(R.RDF.AsRNode(data_set[i]), 1.7, 1.
↳92*0., 1.92*2., "electron_pairs", "el_pairs_r1")
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,↳
↳"el_pairs_r1","part_","r1_p4")
    data_set[i] = Moller.Refine_El_Pairs_2(R.RDF.AsRNode(data_set[i]),↳
↳"el_pairs_r1", "el_pairs_r2")
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,↳
↳"el_pairs_r2","part_","r2_p4")
    data_set[i] = Moller.Refine_El_Pairs_3(R.RDF.AsRNode(data_set[i]), 0.01,↳
↳"el_pairs_r2", "r2_", "el_pairs_r3")
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,↳
↳"el_pairs_r3","part_","r3_p4")
    data_set[i] = Moller.Refine_El_Pairs_1(R.RDF.AsRNode(data_set[i]), 2.0, 1.
↳92*0.85, 1.92*1.1, "el_pairs_r3", "el_pairs_r4")
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,↳
↳"el_pairs_r4","part_","r4_p4")

```

```

    # These cuts should not be done.
    data_set[i] = Moller.Refine_El_Pairs_X(R.RDF.AsRNode(data_set[i]), 0.040, 0.
↪048, "r2_p4t1", "r2_p4t2", "el_pairs_r3", "el_pairs_rx1")
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,
↪"el_pairs_rx1", "part_", "rx1_p4")
    data_set[i] = Moller.Refine_El_Pairs_X2(R.RDF.
↪AsRNode(data_set[i]), "el_pairs_r3", "el_pairs_rx2")
    data_set[i] = Moller.Add_Four_Vectors(R.RDF.AsRNode(data_set[i]), -0.0302,
↪"el_pairs_rx2", "part_", "rx2_p4")

df_uc, df3_uc = data_set

```

```

[6]: h_n_part = df_uc.Define("n_part", "return part_pdg.size()").
↪Histo1D(("h_n_part", "Number of particles", 11, -0.5, 10.5), "n_part")
h_n_neg_track = df_uc.Define("n_neg_track", "int count=0; for(int i=0;
↪i<track_omega.size(); ++i){if(track_omega[i]>0 && track_type[i]==1) count++;}
↪return count;").Histo1D(("h_n_neg_track", "Number of - tracks", 11, -0.5, 10.
↪5), "n_neg_track")
h_n_electron = df_uc.Define("n_electron", "int out=0; for(int i=0; i<part_pdg.
↪size(); ++i){if(part_pdg[i]==11) out++;} return out;").
↪Histo1D(("h_n_electron", "Number of e-", 11, -0.5, 10.5), "n_electron")
h_n_moller = df_uc.Define("n_moller", "return electron_pairs.size()").
↪Histo1D(("h_n_moller", "Number of Moller pairs", 11, -0.5, 10.5), "n_moller")

h_track_time_diff = df_uc.Define("track_time_diff", "vector<double> out;
↪for(auto p: electron_pairs){out.push_back(track_time[part_track[p.first]] -
↪track_time[part_track[p.second]]);} return out;").
↪Histo1D(("h_track_time_diff", "track_time_diff", 500, -50., 50.
↪), "track_time_diff")
hx_track_time_diff = df3_uc.Define("track_time_diff", "vector<double> out;
↪for(auto p: electron_pairs){out.push_back(track_time[part_track[p.first]] -
↪track_time[part_track[p.second]]);} return out;").
↪Histo1D(("hx_track_time_diff", "track_time_diff", 500, -50., 50.
↪), "track_time_diff")

h_track_chi2 = df_uc.Define("t_chi2", "vector<double> out; std::set<int> tracks;
↪for(auto p: electron_pairs){tracks.insert(part_track[p.first]); tracks.
↪insert(part_track[p.second]);}; for(auto t: tracks){out.
↪push_back(track_chi2[t]);} return out;").
↪Histo1D(("h_track_chi2", "track_chi2", 500, 0., 10.), "t_chi2")
h_track_nhit = df_uc.Define("t_nhit", "vector<int> out; std::set<int> tracks;
↪for(auto p: electron_pairs){tracks.insert(part_track[p.first]); tracks.
↪insert(part_track[p.second]);}; for(auto t: tracks){out.
↪push_back(track_n_hits[t]);} return out;").
↪Histo1D(("h_track_nhit", "track_nhit", 15, -0.5, 14.5), "t_nhit")

```

```

h_mom_sum = df_uc.Define("mom_sum","vector<double> out; for(int i=0; i<p4v1.
↳size(); ++i){out.push_back((p4v1[i].Vect()+p4v2[i].Vect()).Mag());} return_
↳out;").Histo1D(("h_mom_sum","mom sum",500,0.,6.0),"mom_sum")
h_mom_sum_r1 = df_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r1_p4v1.size(); ++i){out.push_back((r1_p4v1[i].Vect()+r1_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("h_mom_sum_r1","mom sum r1",500,0.,6.
↳0),"mom_sum")
h_mom_sum_r2 = df_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r2_p4v1.size(); ++i){out.push_back((r2_p4v1[i].Vect()+r2_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("h_mom_sum_r2","mom sum r2",500,0.,6.
↳0),"mom_sum")
h_mom_sum_r3 = df_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r3_p4v1.size(); ++i){out.push_back((r3_p4v1[i].Vect()+r3_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("h_mom_sum_r3","mom sum r3",500,0.,6.
↳0),"mom_sum")
h_mom_sum_r4 = df_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r4_p4v1.size(); ++i){out.push_back((r4_p4v1[i].Vect()+r4_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("h_mom_sum_r4","mom sum r4",500,0.,6.
↳0),"mom_sum")

hx_mom_sum = df3_uc.Define("mom_sum","vector<double> out; for(int i=0; i<p4v1.
↳size(); ++i){out.push_back((p4v1[i].Vect()+p4v2[i].Vect()).Mag());} return_
↳out;").Histo1D(("hx_mom_sum","mom sum",500,0.,6.0),"mom_sum")
hx_mom_sum_r1 = df3_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r1_p4v1.size(); ++i){out.push_back((r1_p4v1[i].Vect()+r1_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("hx_mom_sum_r1","mom sum r1",500,0.,6.
↳0),"mom_sum")
hx_mom_sum_r2 = df3_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r2_p4v1.size(); ++i){out.push_back((r2_p4v1[i].Vect()+r2_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("hx_mom_sum_r2","mom sum r2",500,0.,6.
↳0),"mom_sum")
hx_mom_sum_r3 = df3_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r3_p4v1.size(); ++i){out.push_back((r3_p4v1[i].Vect()+r3_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("hx_mom_sum_r3","mom sum r3",500,0.,6.
↳0),"mom_sum")
hx_mom_sum_r4 = df3_uc.Define("mom_sum","vector<double> out; for(int i=0;_
↳i<r4_p4v1.size(); ++i){out.push_back((r4_p4v1[i].Vect()+r4_p4v2[i].Vect()).
↳Mag());} return out;").Histo1D(("hx_mom_sum_r4","mom sum r4",500,0.,6.
↳0),"mom_sum")

h_beam_theta_x = df_uc.Histo1D(("h_beam_theta_x","Beam Angle_
↳#theta_{x}",1000,-0.1,0.1),"p4beam_theta_x")
hx_beam_theta_x = df3_uc.Histo1D(("hx_beam_theta_x","Beam Angle_
↳#theta_{x}",1000,-0.1,0.1),"p4beam_theta_x")

```

```

h_ee_thtE1 = df_uc.Histo2D(("h_ee_thtE1", "EE theta-E 1;#theta_{1} [rad];E_{1} [GeV]", 500, 0., 0.1, 500, 0., 2.5), "p4tht1", "p4E1")
h_ee_thtE2 = df_uc.Histo2D(("h_ee_thtE2", "EE theta-E 2;#theta_{2} [rad];E_{2} [GeV]", 500, 0., 0.1, 500, 0., 2.5), "p4tht2", "p4E2")
h_ee_thth = df_uc.Histo2D(("h_ee_thth", "EE theta-theta;#theta_{1} [rad];#theta_{2} [rad]", 500, 0., 0.1, 500, 0., 0.1), "p4tht1", "p4tht2")

```

```

[7]: cc0 = R.TCanvas("cc0", "cc0", 1200, 800)
cc0.Divide(3, 2)
cc0.cd(1)
h_n_moller.SetTitle("Particle counts in Moller events")
h_n_moller.SetStats(0)
h_n_moller.SetLineWidth(2)
h_n_moller.SetLineColor(R.kCyan)
h_n_moller.Draw("same")
h_n_electron.SetLineWidth(3)
h_n_electron.SetLineColor(R.kBlue)
h_n_electron.Draw("same")
h_n_neg_track.SetLineColor(R.kRed)
h_n_neg_track.Draw("same")
h_n_part.SetLineWidth(1)
h_n_part.SetLineColor(R.kGreen+2)
h_n_part.Draw("same")
legend = R.TLegend(0.7, 0.7, 0.9, 0.9)
legend.AddEntry(h_n_part.GetPtr(), "any particle", "l")
legend.AddEntry(h_n_neg_track.GetPtr(), "neg. tracks", "l")
legend.AddEntry(h_n_electron.GetPtr(), "electron", "l")
legend.AddEntry(h_n_moller.GetPtr(), "Moller", "l")
legend.Draw()
cc0.cd(2)
h_track_time_diff.SetStats(0)
h_track_time_diff.SetLineWidth(1)
h_track_time_diff.SetLineColor(R.kGreen+2)
h_track_time_diff.Draw()
hx_track_time_diff.SetLineColor(R.kRed)
hx_track_time_diff.Draw("same")

# hm_track_time_diff.SetLineColor(R.kAzure)
# hm_track_time_diff.Draw("same")
pad6 = cc0.cd(3)
pad6.SetLogy()

h_beam_theta_x.SetStats(0)
h_beam_theta_x.SetLineWidth(1)
h_beam_theta_x.SetLineColor(R.kGreen+2)
h_beam_theta_x.Draw("same")
hx_beam_theta_x.SetLineColor(R.kRed)

```

```

hx_beam_theta_x.Draw("same")

leg2 = R.TLegend(0.7,0.7,0.9,0.9)
leg2.AddEntry(h_beam_theta_x.GetPtr(),"Moller","l")
leg2.AddEntry(hx_beam_theta_x.GetPtr(),"Background","l")
leg2.Draw()

pad2 = cc0.cd(4)
pad2.SetLogz()
h_ee_thth.SetStats(0)
h_ee_thth.Draw("colz")
l1 = R.TLine(0.,0.04,0.04,0.)
# l1.SetLineColor(R.kRed)
# l1.SetLineWidth(2)
# l1.Draw()
# l2 = R.TLine(0.,0.048,0.048,0.)
# l2.SetLineColor(R.kRed)
# l2.SetLineWidth(2)
# l2.Draw()
f1 = R.TF1("f1","2*asin(0.00011108696/sin(x/2))",0.0044,0.1)
f1.SetLineColor(R.kGreen)
f1.Draw("same")

pad3 = cc0.cd(5)
pad3.SetLogz()
h_ee_thtE1.SetStats(0)
h_ee_thtE1.Draw("colz")
f2 = R.TF1("f2","2.3/(1+ (2.3/0.000511)*(1-cos(x)))",0.005,0.1)
f2.SetLineColor(R.kRed)
f2.Draw("same")

pad4 = cc0.cd(6)
#f1 = R.TF1("f1","2*asin(0.11108696/sin(x/2))",0.0001,0.1)
pad4.SetLogz()
h_ee_thtE2.SetStats(0)
h_ee_thtE2.Draw("colz")
f3 = R.TF1("f3","2.3/(1+ (2.3/0.000511)*(1-cos(x)))",0.005,0.1)
f3.SetLineColor(R.kRed)
f3.Draw("same")

cc0.Draw()

```

<IPython.core.display.HTML object>

```

[8]: cc1 = R.TCanvas("cc1","cc1",1200,400)
      cc1.Divide(2,1)

```

```

cc1.cd(1)
h_mom_sum.SetStats(0)
h_mom_sum.SetTitle("Momentum sum, Moller pairs")
h_mom_sum.SetLineWidth(1)
h_mom_sum.SetLineColor(R.kBlue)
h_mom_sum.Draw()
h_mom_sum_r1.SetStats(0)
h_mom_sum_r1.SetLineColor(R.kGreen+2)
h_mom_sum_r1.Draw("same")
h_mom_sum_r2.SetStats(0)
h_mom_sum_r2.SetLineColor(R.kRed)
h_mom_sum_r2.Draw("same")
h_mom_sum_r3.SetStats(0)
h_mom_sum_r3.SetLineColor(R.kOrange)
h_mom_sum_r3.Draw("same")
h_mom_sum_r4.SetStats(0)
h_mom_sum_r4.SetLineColor(R.kCyan)
h_mom_sum_r4.Draw("same")
legend2 = R.TLegend(0.7,0.7,0.95,0.9)
legend2.AddEntry(h_mom_sum.GetPtr(),"Moller","l")
legend2.AddEntry(h_mom_sum_r1.GetPtr(),"Moller r1","l")
legend2.AddEntry(h_mom_sum_r2.GetPtr(),"Moller r1+r2","l")
legend2.AddEntry(h_mom_sum_r3.GetPtr(),"Moller r1+r2+r3","l")
legend2.AddEntry(h_mom_sum_r4.GetPtr(),"Moller ...r3+r4","l")
legend2.Draw()

cc1.cd(2)
hx_mom_sum.SetStats(0)
hx_mom_sum.SetTitle("Momentum sum, New code")
hx_mom_sum.SetLineWidth(1)
hx_mom_sum.SetLineColor(R.kBlue)
hx_mom_sum.Draw()
hx_mom_sum_r1.SetStats(0)
hx_mom_sum_r1.SetLineColor(R.kGreen+2)
hx_mom_sum_r1.Draw("same")
hx_mom_sum_r2.SetStats(0)
hx_mom_sum_r2.SetLineColor(R.kRed)
hx_mom_sum_r2.Draw("same")
hx_mom_sum_r3.SetStats(0)
hx_mom_sum_r3.SetLineColor(R.kOrange)
hx_mom_sum_r3.Draw("same")
hx_mom_sum_r3.Draw("same")
hx_mom_sum_r4.SetStats(0)
hx_mom_sum_r4.SetLineColor(R.kCyan)
hx_mom_sum_r4.Draw("same")

legend3 = R.TLegend(0.7,0.7,0.95,0.9)

```



```

legend3.AddEntry(hx_mom_sum.GetPtr(),"New code","1")
legend3.AddEntry(hx_mom_sum_r1.GetPtr(),"New code r1","1")
legend3.AddEntry(hx_mom_sum_r2.GetPtr(),"New code r2","1")
legend3.AddEntry(hx_mom_sum_r3.GetPtr(),"New code r2+r3","1")
legend3.AddEntry(hx_mom_sum_r4.GetPtr(),"New code ...r3+r4","1")
legend3.Draw()

cc1.Draw()

```

<IPython.core.display.HTML object>

```

[9]: h_ee_mass_r1 = df_uc.Histo1D(("h_ee_mass_r1","EE inv mass R1; M_{inv} [GeV]",500,0.,0.2),"r1_p4minv")
      ↪ [GeV]",500,0.,0.2),"r1_p4minv")
hx_ee_mass_r1 = df3_uc.Histo1D(("hx_ee_mass_r1","EE inv mass R1; M_{inv} [GeV]",500,0.,0.2),"r1_p4minv")
      ↪ [GeV]",500,0.,0.2),"r1_p4minv")
#hm_ee_mass_r1 = dfm.Histo1D(("hm_ee_mass_r1","EE inv mass R1; M_{inv} [GeV]",500,0.,0.2),"r1_p4minv")
      ↪ [GeV]",500,0.,0.2),"r1_p4minv")

h_ee_mass_r2 = df_uc.Histo1D(("h_ee_mass_r2","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r2_p4minv")
      ↪ [GeV]",500,0.,0.2),"r2_p4minv")
hx_ee_mass_r2 = df3_uc.Histo1D(("hx_ee_mass_r2","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r2_p4minv")
      ↪ [GeV]",500,0.,0.2),"r2_p4minv")
#hm_ee_mass_r2 = dfm.Histo1D(("hm_ee_mass_r2","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r2_p4minv")
      ↪ [GeV]",500,0.,0.2),"r2_p4minv")

h_ee_mass_r3 = df_uc.Histo1D(("h_ee_mass_r3","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r3_p4minv")
      ↪ [GeV]",500,0.,0.2),"r3_p4minv")
hx_ee_mass_r3 = df3_uc.Histo1D(("hx_ee_mass_r3","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r3_p4minv")
      ↪ [GeV]",500,0.,0.2),"r3_p4minv")
#hm_ee_mass_r3 = dfm.Histo1D(("hm_ee_mass_r3","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r3_p4minv")
      ↪ [GeV]",500,0.,0.2),"r3_p4minv")

h_ee_mass_r4 = df_uc.Histo1D(("h_ee_mass_r4","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r4_p4minv")
      ↪ [GeV]",500,0.,0.2),"r4_p4minv")
hx_ee_mass_r4 = df3_uc.Histo1D(("hx_ee_mass_r4","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r4_p4minv")
      ↪ [GeV]",500,0.,0.2),"r4_p4minv")
#hm_ee_mass_r4 = dfm.Histo1D(("hm_ee_mass_r4","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"r4_p4minv")
      ↪ [GeV]",500,0.,0.2),"r4_p4minv")

h_ee_mass_rx1 = df_uc.Histo1D(("h_ee_mass_rx1","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"rx1_p4minv")
      ↪ [GeV]",500,0.,0.2),"rx1_p4minv")
hx_ee_mass_rx1 = df3_uc.Histo1D(("hx_ee_mass_rx1","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"rx1_p4minv")
      ↪ [GeV]",500,0.,0.2),"rx1_p4minv")
#hm_ee_mass_rx1 = dfm.Histo1D(("hm_ee_mass_rx1","EE inv mass; M_{inv} [GeV]",500,0.,0.2),"rx1_p4minv")
      ↪ [GeV]",500,0.,0.2),"rx1_p4minv")

```

```

h_ee_mass_rx2 = df_uc.Histo1D(("h_ee_mass_rx2", "EE inv mass; M_{inv}
↳ [GeV]", 500, 0., 0.2), "rx2_p4minv")
hx_ee_mass_rx2 = df3_uc.Histo1D(("hx_ee_mass_rx2", "EE inv mass; M_{inv}
↳ [GeV]", 500, 0., 0.2), "rx2_p4minv")
#hm_ee_mass_rx2 = dfm.Histo1D(("hm_ee_mass_rx2", "EE inv mass; M_{inv}
↳ [GeV]", 500, 0., 0.2), "rx2_p4minv")

```

```

[10]: cc2 = R.TCanvas("cc2", "cc2", 1200, 400)
cc2.Divide(2, 1)
pad1 = cc2.cd(1)
# pad1.SetLogy()
h_ee_mass_r1.SetTitle("M_{inv} Moller")
h_ee_mass_r1.SetStats(0)
h_ee_mass_r1.SetLineColor(R.kGreen+2)
h_ee_mass_r1.SetLineWidth(2)
#h_ee_mass_r1.Draw()

h_ee_mass_r2.SetStats(0)
h_ee_mass_r2.SetLineColor(R.kRed)
h_ee_mass_r2.SetLineWidth(2)
h_ee_mass_r2.Draw("same")

h_ee_mass_r3.SetStats(0)
h_ee_mass_r3.SetLineColor(R.kOrange)
h_ee_mass_r3.SetLineWidth(2)
h_ee_mass_r3.Draw("same")

h_ee_mass_r4.SetStats(0)
h_ee_mass_r4.SetLineColor(R.kCyan)
h_ee_mass_r4.SetLineWidth(2)
h_ee_mass_r4.Draw("same")

legend = R.TLegend(0.6, 0.7, 0.9, 0.9)
legend.AddEntry(h_ee_mass_r1.GetPtr(), "Moller r1", "l")
legend.AddEntry(h_ee_mass_r2.GetPtr(), "Moller r1+r2", "l")
legend.AddEntry(h_ee_mass_r3.GetPtr(), "Moller ...r2+r3", "l")
legend.AddEntry(h_ee_mass_r4.GetPtr(), "Moller ...r3+r4", "l")
legend.Draw()

pad2 = cc2.cd(2)

hx_ee_mass_r1.SetTitle("M_{inv} New Code")
hx_ee_mass_r1.SetStats(0)
hx_ee_mass_r1.SetLineColor(R.kGreen+2)
hx_ee_mass_r1.SetLineWidth(2)
hx_ee_mass_r1.Draw()

```

```

hx_ee_mass_r2.SetStats(0)
hx_ee_mass_r2.Draw("same")

hx_ee_mass_r2.SetStats(0)
hx_ee_mass_r2.SetLineColor(R.kRed)
hx_ee_mass_r2.SetLineWidth(2)
hx_ee_mass_r2.Draw("same")

hx_ee_mass_r3.SetStats(0)
hx_ee_mass_r3.SetLineColor(R.kOrange)
hx_ee_mass_r3.SetLineWidth(2)
hx_ee_mass_r3.Draw("same")

hx_ee_mass_r4.SetStats(0)
hx_ee_mass_r4.SetLineColor(R.kCyan)
hx_ee_mass_r4.SetLineWidth(2)
hx_ee_mass_r4.Draw("same")

legend2 = R.TLegend(0.6,0.7,0.9,0.9)
legend2.AddEntry(hx_ee_mass_r1.GetPtr(),"New Code r1","l")
legend2.AddEntry(hx_ee_mass_r2.GetPtr(),"New Code r1+r2","l")
legend2.AddEntry(hx_ee_mass_r3.GetPtr(),"New Code ...r2+r3","l")
legend2.AddEntry(hx_ee_mass_r4.GetPtr(),"New Code ...r3+r4","l")

legend2.Draw()

cc2.Draw()

```

<IPython.core.display.HTML object>