Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bundesministerium
für Bildung
und Forschung

ROOT tutorial
— part IV —

1st April 2015

Adrian Perieanu

ROOT tutorial: goals

* how to install it ✓

* how to find/read documentation ✓

* perform an interactive analysis with ROOT ✓

* design and write own analysis macros ✘

* how to store results of your analysis ✘

ROOT tutorial:
remote (batch mode) analysis

ingredients:

* own macro (with main function)

details:

* define the file
* define the root tree
* define the variables
* loop over the entries of the root tree

print out the variables

goal:
— run your own macro to perform the analysis

ROOT tutorial:
remote (batch mode) analysis

to do:

— give the correct file name

— same with the root tree name

— set the tree branches

— set a proper name for each variable, e.g. pT, eta, phi, m

— run the root macro

ROOT tutorial: first we need to edit

choose your favoured editor and pick your favoured name

I will go with
> nano extract_ROOTTree.C

> nano my_best_macro_ever.C

- general settings:

```
void extract_ROOTTree(){
  gStyle->SetPalette(1);
  gStyle->SetOptStat(0);
  // define the invariant mass histogram
  TH1D* myMass    = new TH1D( "myMass", "myMass", 64, 9.0, 10.6);
  myMass->GetXaxis()->SetTitle("M_{#mu^{+}#mu^{-}} [GeV/c^{2}]");
  myMass->GetYaxis()->SetTitle("Candidates/0.025 GeV/c^{2}");
  myMass->GetYaxis()->SetTitleOffset(1.3);
  // define the pT_vs_eta histogram
  TH2D* mypT_eta = new TH2D( "mypT_eta", "mypT_eta", 48, -2.4, 2.4, 25, 0., 50.);
  mypT_eta->GetXaxis()->SetTitle("#eta_{#mu^{+}#mu^{-}}");
  mypT_eta->GetYaxis()->SetTitle("p_{T #mu^{+}#mu^{-}} [GeV/c]");
  mypT_eta->GetZaxis()->SetTitle("Candidates/2 GeV/c / 0.1");
```

- open the root file and the root tree:

```
// open root file
TFile* file = new TFile("give_the_right_name.root","READ");
// get the tree
TTree* treeU1S = (TTree*)file->Get("give_the_right_path_and_name");
// define variables where to store the tree information
Float_t pTU1S;
Float_t etaU1S;
Float_t phiU1S;
Float_t mU1S;
```

ROOT tutorial: let's go now for details

- setup the root tree branches:

```
// define the tree branches
treeU1S->SetBranchAddress( "set_right_variable_name1",  &pTU1S);
treeU1S->SetBranchAddress( "set_right_variable_name2", &etaU1S);
treeU1S->SetBranchAddress( "set_right_variable_name3", &phiU1S);
treeU1S->SetBranchAddress( "set_right_variable_name4",   &mU1S);
```

- loop over all entries:

```
// loop over the entries in the tree
for( unsigned int i = 0; i < treeU1S->GetEntries(); ++i){
  treeU1S->GetEntry(i);
  cout<<"mU1S:"<<mU1S<<endl;
  myMass->Fill( mU1S);
  mypT_eta->Fill( etaU1S, pTU1S);
}// end of loop over the entries in the tree
```

ROOT tutorial: time to plot and save results

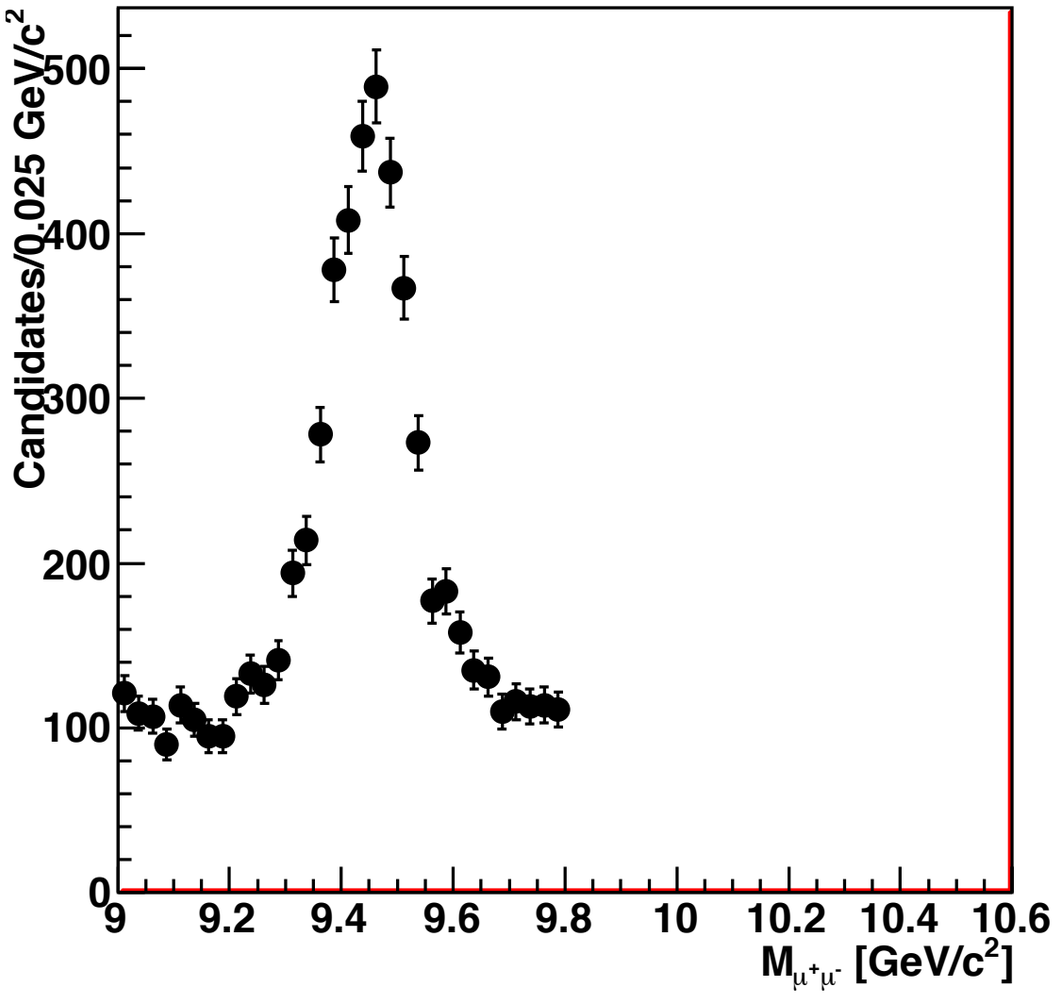
- plot & save the results :

```
  // define canvas
  TCanvas *myCanvas = new TCanvas("myCanvas", "myCanvas", 1020, 520);
  myCanvas->SetFillColor(0);
  myCanvas->SetBorderMode(0);
  myCanvas->SetLeftMargin(0.14);
  myCanvas->SetRightMargin(0.16);
  myCanvas->SetBottomMargin(0.14);
  // divide the canvas
  myCanvas->Divide(2,1);
  myCanvas->cd(1);
  // draw the histogram
  myMass->SetMarkerStyle(20);
  myMass->Draw("e1");
  myCanvas->cd(2);
  (myCanvas->cd(2))->SetRightMargin(  0.15);
  (myCanvas->cd(2))->SetBottomMargin( 0.11);
  // draw the histogram
  mypT_eta->Draw("colz");
  // save the histogram
  myCanvas->Print("extract_ROOTTree.eps");
}
```

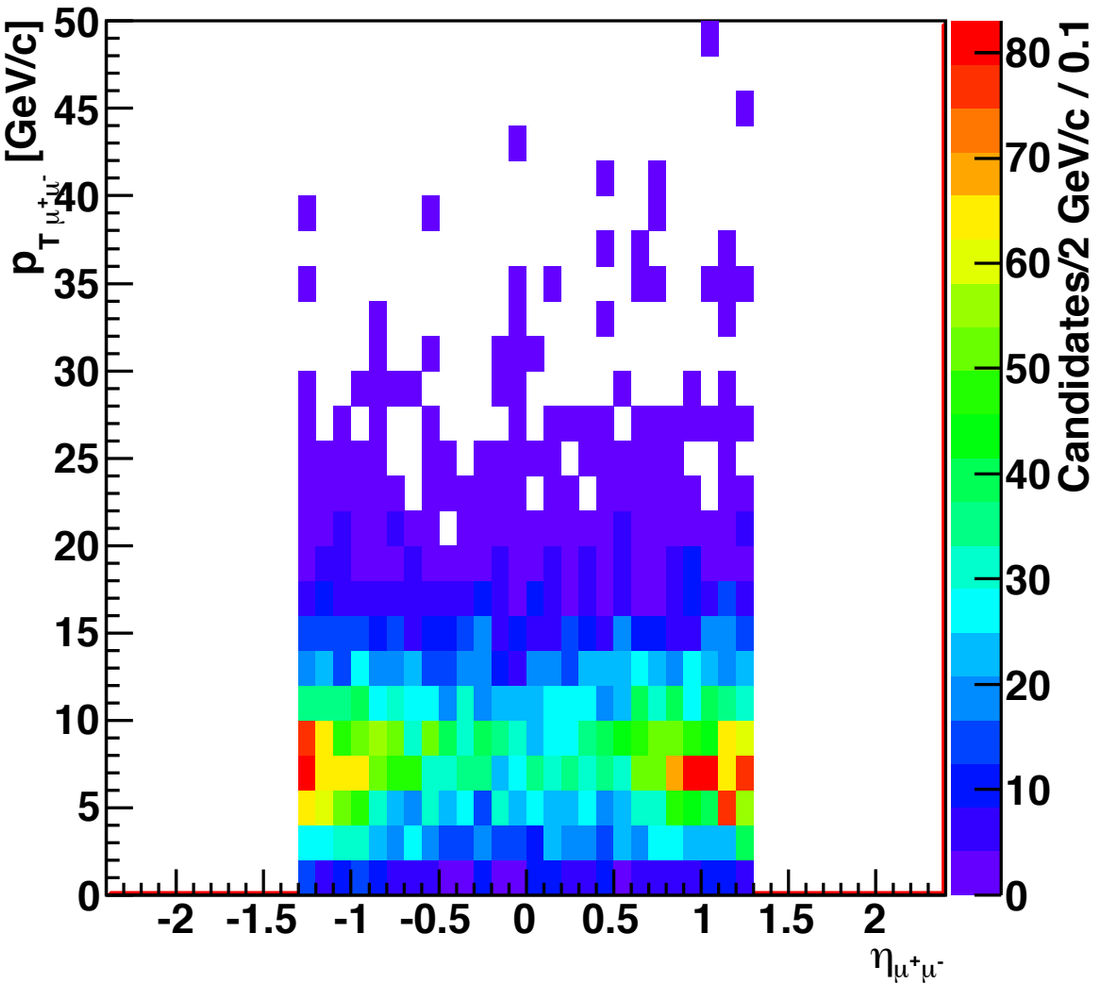
ROOT tutorial:
how it should come out

run:
> root -l -q extract_ROOTTree.C

myMass



mypT_eta



ROOT tutorial: how to store more? (because we can)

- include the right ROOT header:

```
// ROOT header files
#include "TFile.h"
#include "TH3D.h"
// C/C++ standard libraries
#include <iostream> // for input/output prints
#include <fstream>  // for reading the ASCII files
```

- define a TH3D histogram:
  (can store 3 variables, one on each axis)

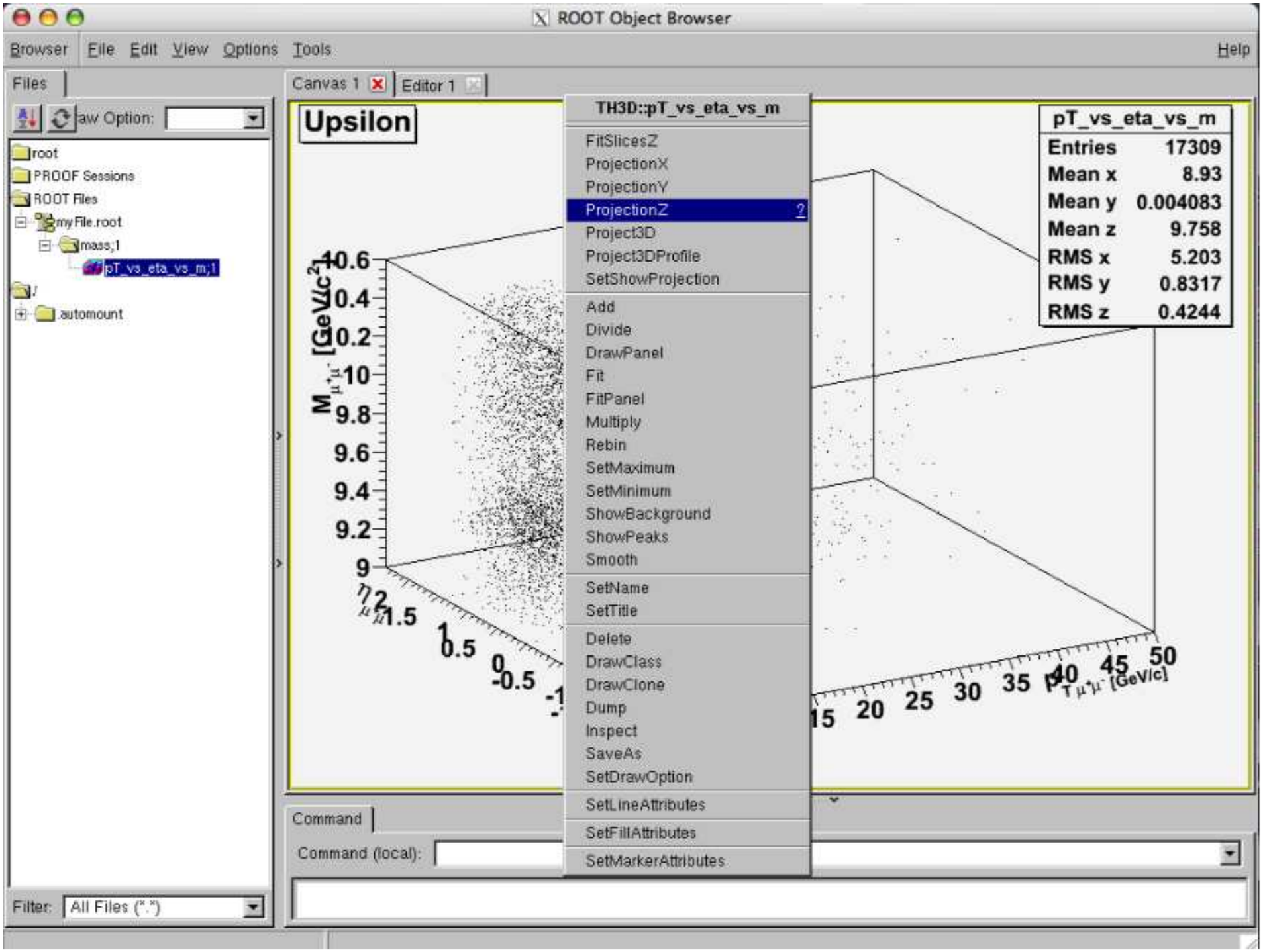
```
// define histograms
TH3D* pT_vs_eta_vs_m = new TH3D( "pT_vs_eta_vs_m", "Upsilon",
                                 25, 0., 50., 48, -2.4, 2.4, 64, 9.0, 10.6);
pT_vs_eta_vs_m->GetZaxis()->SetTitle("M_{#mu^{+}#mu^{-}} [GeV/c^{2}]");
pT_vs_eta_vs_m->GetYaxis()->SetTitle("#eta_{#mu^{+}#mu^{-}}");
pT_vs_eta_vs_m->GetXaxis()->SetTitle("p_{T #mu^{+}#mu^{-}} [GeV/c]");
```

- fill the histogram according to its definition:

```
// fill histogram
pT_vs_eta_vs_m->Fill( pTU, etaU, mU);
```

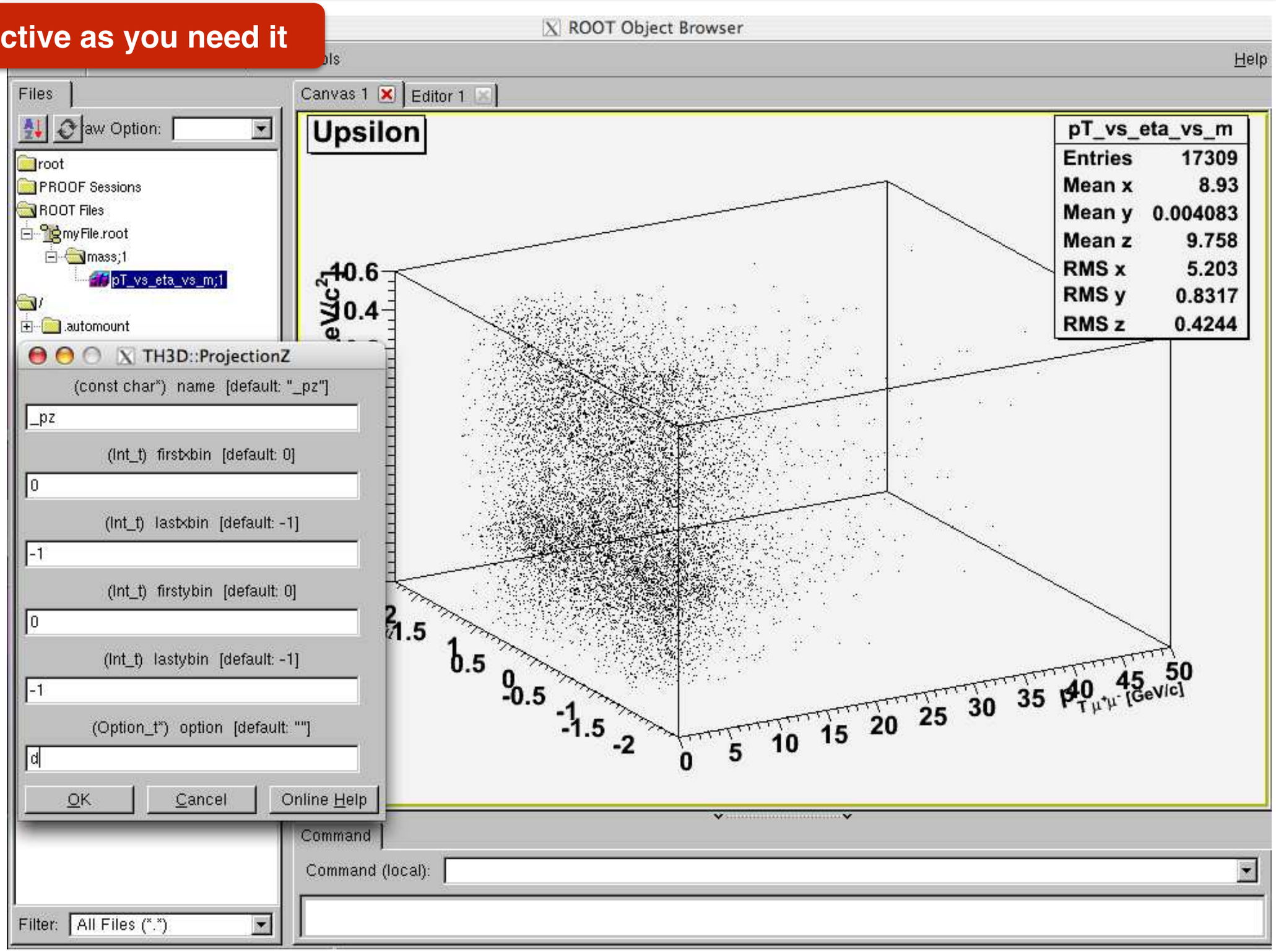
ROOT tutorial: how to extract more? (because we can)

be interactive as you need it



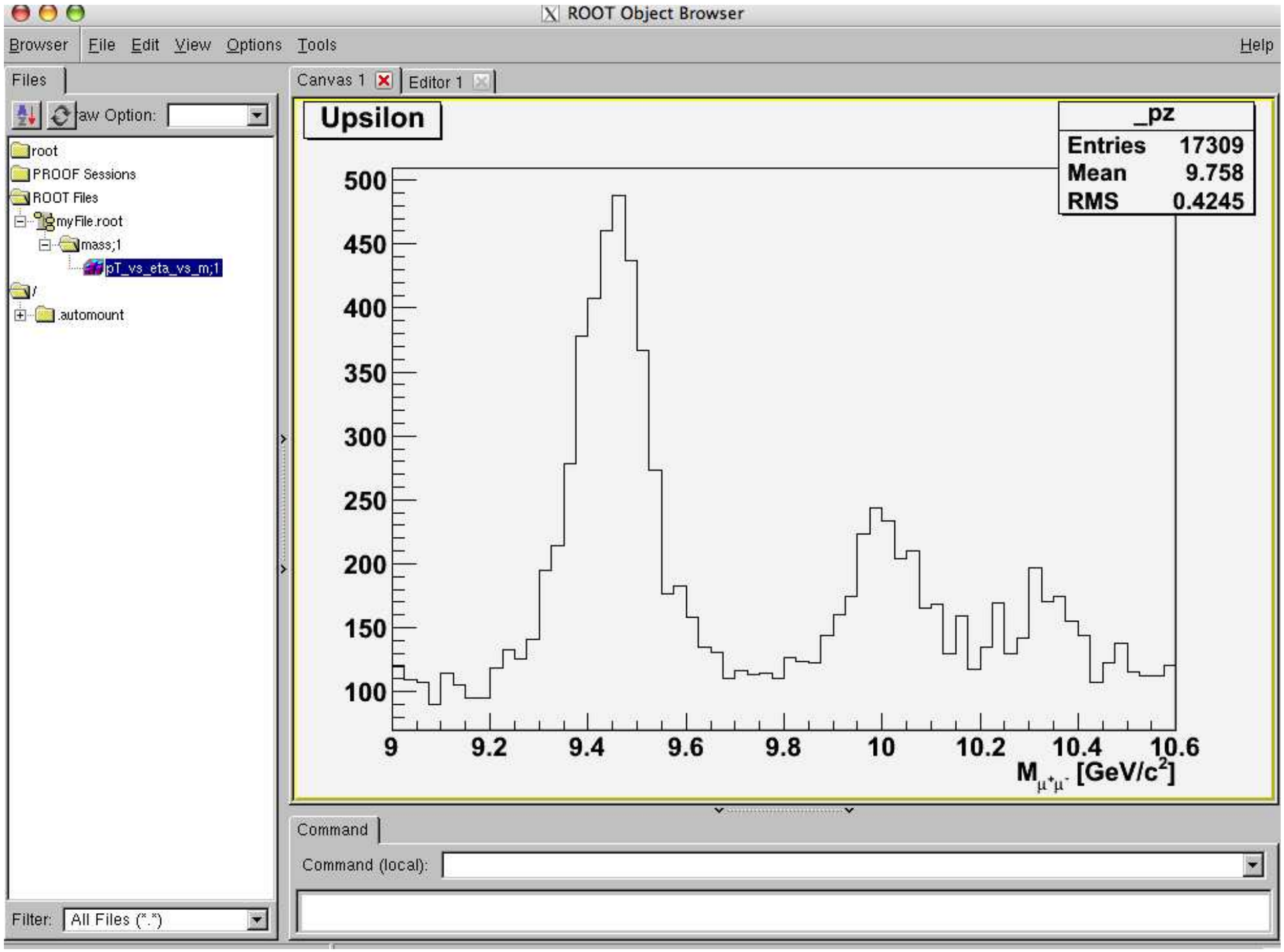
ROOT tutorial: how to extract more? (because we can)

be interactive as you need it



ROOT tutorial: how to extract more? (because we can)

be interactive as you need it



ROOT tutorial: how to extract more in batch mode

and then be cool — go remote

- general settings:

```
void extract_TH3D(){
  gStyle->SetPalette(1);
  gStyle->SetOptStat(0);
```

- open the root file & extract information (1D or 2D):

```
// open root file
TFile* file = new TFile("myFile.root","READ");
// open needed histograms
TH3D* myHisto = (TH3D*)file->Get("mass/pT_vs_eta_vs_m");
// extract variable on Z axis
TH1D* myZAxis = myHisto->ProjectionZ();
myZAxis->GetXaxis()->SetTitle("M_{#mu^{+}#mu^{-}} [GeV/c^{2}]");
myZAxis->GetYaxis()->SetTitle("Candidates/0.025 GeV/c^{2}");
// extract variables on Z and X axis
TH2D* myXYAxis = myHisto->Project3DProfile("xy");
myXYAxis->GetXaxis()->SetTitle("#eta_{#mu^{+}#mu^{-}}");
myXYAxis->GetYaxis()->SetTitle("p_{T #mu^{+}#mu^{-}} [GeV/c]");
myXYAxis->GetZaxis()->SetTitle("Candidates/2 GeV/c / 0.1");
```

ROOT tutorial: how to extract more in batch mode

and then be cool — go remote

- define a canvas (matter of taste):

```
// define canvas
TCanvas *myCanvas = new TCanvas("myCanvas", "myCanvas", 1020, 520);
myCanvas->SetFillColor(0);
myCanvas->SetBorderMode(0);
myCanvas->SetLeftMargin(0.14);
myCanvas->SetRightMargin(0.16);
myCanvas->SetBottomMargin(0.14);
```

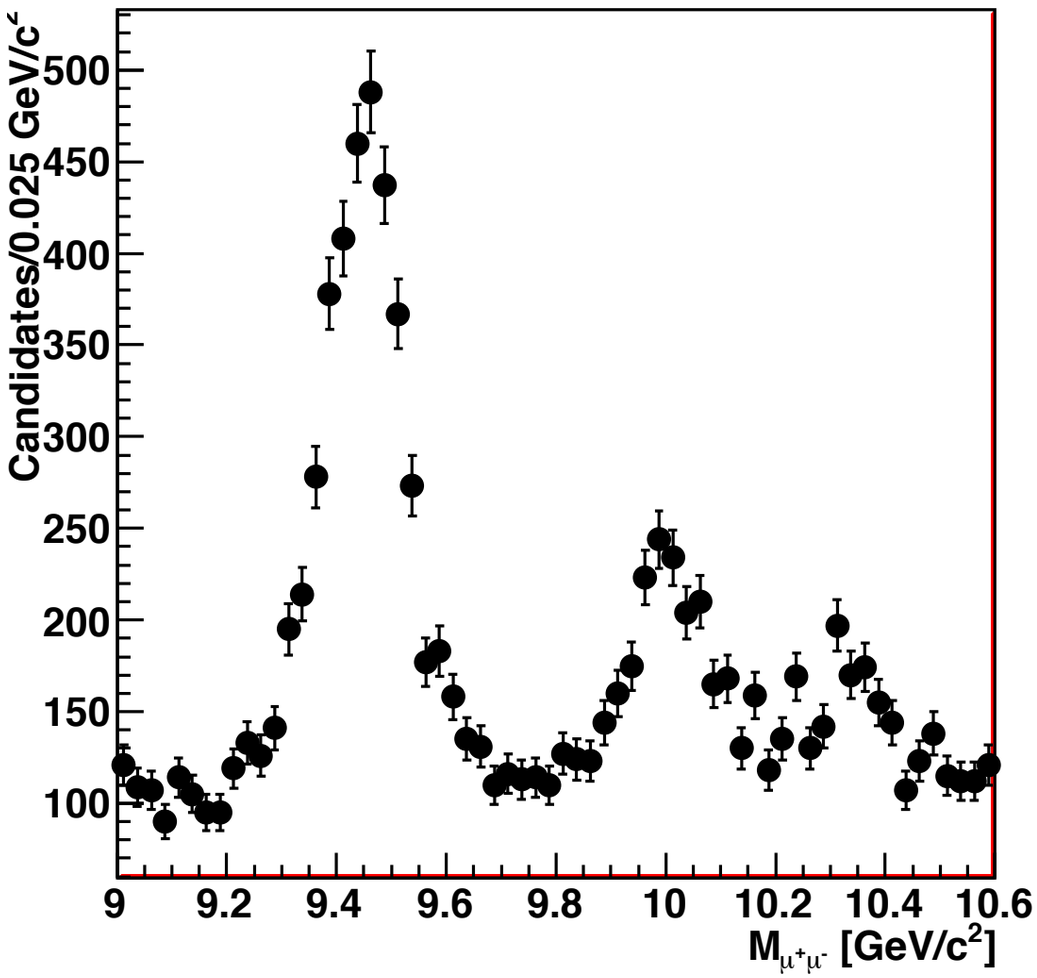
- plot & save the results:

```
  // divide the canvas
  myCanvas->Divide(2,1);
  myCanvas->cd(1);
  // draw the histogram
  myZAxis->SetMarkerStyle(20);
  myZAxis->Draw("e1");
  myCanvas->cd(2);
  (myCanvas->cd(2))->SetRightMargin(  0.15);
  (myCanvas->cd(2))->SetBottomMargin( 0.11);
  // draw the histogram
  myXYAxis->Draw("colz");
  // save the histogram
  myCanvas->Print("extract_TH3D.eps");
}
```

ROOT tutorial: how to extract more in batch mode

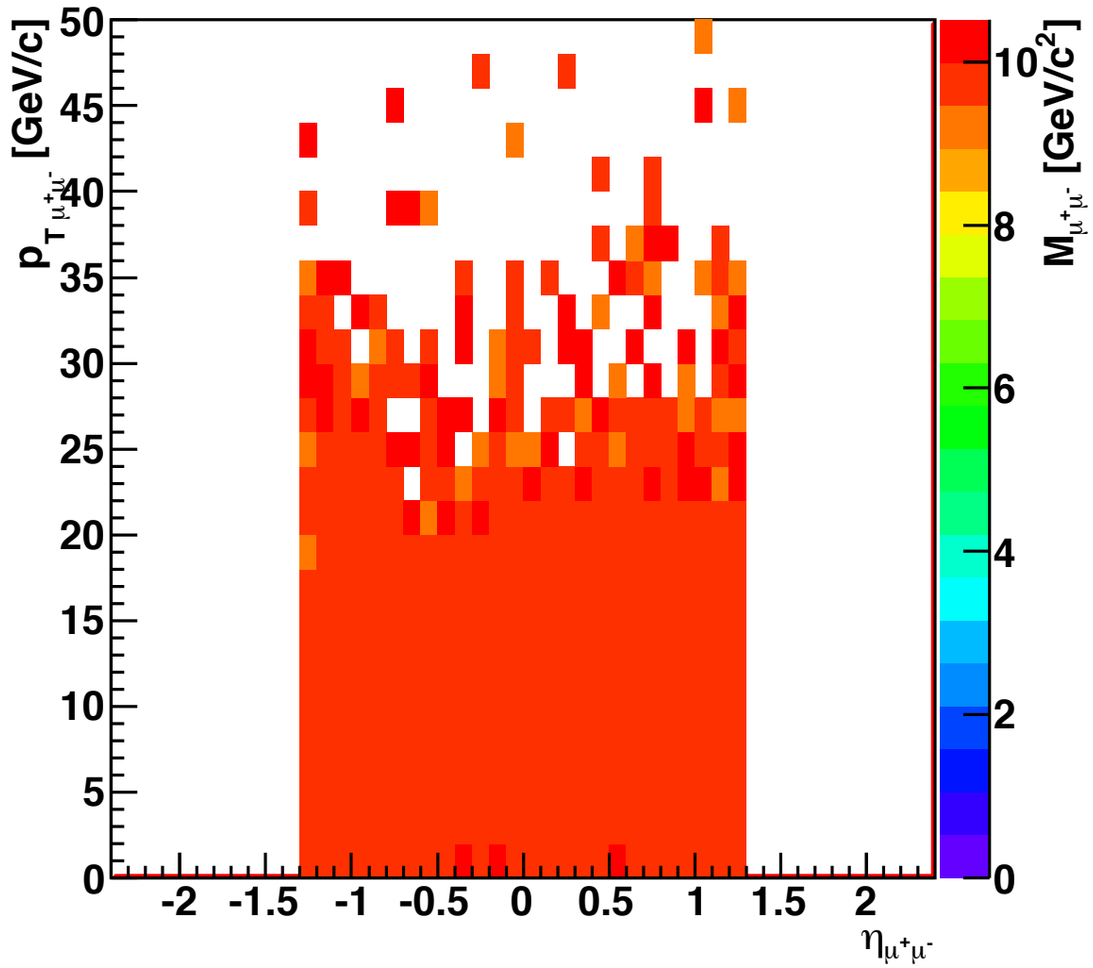
and then be cool — go remote

Upsilon



- information from the Z axis

Upsilon_xy



- information from the X & Y axis

summary: remote analysis

* now you can write your own macro

* can add more and extract more information

* store information in histograms and print them in your favoured format

time for a break and (a lot of) questions

each of you will become a Picasso