Subject: Different results for same information extracted in different ways
Posted by Mamen on Fri, 07 Nov 2014 12:42:11 GMT
View Forum Message <> Reply to Message

Hi guys,

I'm having a problem since yesterday with the way I extract de same information from a histogram. Doing it in different ways I get different results, but I don't understand why or what am I doing wrong. Maybe somebody can help me.
I'll try to explain my problem in the best way I can.

I have a root file, let's call it Input.root, with a structure with two n-tuples:

truthTuple:
- eppx // ep: positive electron, px: component x of momentum
- empx // em: negative electron, px: component x of momentum
- eppy // ep: positive electron, py: component y of momentum
- empy // em: negative electron, py: component y of momentum
- eppz// ep: positive electron, pz: component z of momentum
- empz // em: negative electron, pz: component z of momentum
- and others, but let's keep only these for the example

recoTuple:
- eppid // Particle Identification for positron hypothesis 0: Charged, 1:Very loose, 2: Loose, 3: Tight, 4: Very Thight
- empid // Particle Identification for electron hypothesis 0: Charged, 1:Very loose, 2: Loose, 3: Tight, 4: Very Thight
- feppx // ep: positive electron, px: component x of momentum
- fempx // em: negative electron, px: component x of momentum
- feppy // ep: positive electron, py: component y of momentum
- fempy // em: negative electron, py: component y of momentum
- feppz// ep: positive electron, pz: component z of momentum
- fempz // em: negative electron, pz: component z of momentum
- and others, but let's keep only these for the example

Ok, So I define two histograms

```
True = new TH1D ("True", "True", Nbins, Bin_min, Bin_max);
Reco = new TH1D ("Reco", "Reco", Nbins, Bin_min, Bin_max);
```

And after opening the root file I do:

```
        TFile *t=new TFile("Input.root");

   TTree *truthTuple = (TTree*)t->Get("truthTuple");
   TTree *recoTuple  = (TTree*)t->Get("recoTuple");

   // Projection of Real Statistics Files/* IT HAS TO BE DONE WITH THE REAL STATISTICS FILE*/
   truthTuple->Project("True", "variable", "eppid>3");
        recoTuple->Project("Reco",
"(sqrt((feppe+fempe)**2-(feppx+fempx)**2-(feppy+fempy)**2-(feppz+fempz)**2))**2",
```

"eppid>3");

After this, if I do

Reco->GetEntries();

I get 2323 entries

if I do:

```
int TotalEntries=0;
for (int i=0; i <Nbins;i++ )
{
TotalEntries=TotalEntries+Reco->GetBinContent(i+1);// histograms start on bin=1 not bin=0
}
cout << "Total Reco Entries: "<< TotalEntries<< endl;
```

I get "Total Reco Entries: 2297", when I think the result should be the same than Reco->GetEntries();
Does anybody have an idea of what can I be doing wrong?

On the other hand, if I fill the histogram differently, i.e. setting branch addresses and filling after applying a cut,
I also get a different number of entries in the histogram.
That would be something like that:

```
  // Variables to be read and filled recoTuple
  float feppx;
  float feppy;
  float feppz;
  float feppe;
  float fepp3;
  float fepcosth;
  int eppid, ntrk;

  recoTuple->SetBranchAddress("feppx", &feppx);
  recoTuple->SetBranchAddress("feppy", &feppy);
  recoTuple->SetBranchAddress("feppz", &feppz);
  recoTuple->SetBranchAddress("feppe", &feppe);
  recoTuple->SetBranchAddress("fepp3", &fepp3);
  recoTuple->SetBranchAddress("epcosth", &fepcosth);
  recoTuple->SetBranchAddress("eppid", &eppid);
  recoTuple->SetBranchAddress("ntrk", &ntrk);

  float fempx;
  float fempy;
  float fempz;
  float fempe;
  float femp3;
```

```
  float femcosth;
  int empid;

  recoTuple->SetBranchAddress("fempx", &fempx);
  recoTuple->SetBranchAddress("fempy", &fempy);
  recoTuple->SetBranchAddress("fempz", &fempz);
  recoTuple->SetBranchAddress("fempe", &fempe);
  recoTuple->SetBranchAddress("femp3", &femp3);
  recoTuple->SetBranchAddress("emcosth", &femcosth);
  recoTuple->SetBranchAddress("empid", &empid);


  float fpi0px;
  float fpi0py;
  float fpi0pz;
  float fpi0pe;

  recoTuple->SetBranchAddress("fpi0px", &fpi0px);
  recoTuple->SetBranchAddress("fpi0py", &fpi0py);
  recoTuple->SetBranchAddress("fpi0pz", &fpi0pz);
  recoTuple->SetBranchAddress("fpi0pe", &fpi0pe);


  TH1D * RecoFill;
  RecoFill = new TH1D("RecoFill",  "RecoFill", Nbins, Bin_max, Bin_max);

  long NEntriesReco=(long)recoTuple->GetEntries();
  int kkk=0;

  for (int k=0; k<NEntriesReco; k++)
   {

     recoTuple->GetEntry(k);


     if( eppid>3)
 {

        if (kkk % 10000 == 0 && kkk != 0)
         {
           cout<<"*** FILLING *** "<< kkk << " : " <<endl;
         }

        kkk++;


RecoFill->Fill((sqrt((feppe+fempe)**2-(feppx+fempx)**2-(feppy+fempy)**2-(feppz+fempz)**2))**
2);
       }
    }

  cout<< "RecoFill: "<<   RecoFill->GetEntries()<<endl;
```

In this case I get that the number of Entries is 2239, again a number different from the two previous ones.
I am representing always the same variable, and I am applying always the same cuts in different ways.
Does somebody know what am i doing wrong?
I've tried doing it with two different macros and also at the same time inside a unique macro. Always I get the discrepancies. I have also tested the cuts, the variables, tried different ones, the number of bins, and bin limits in the histograms... I am now puzzled, and I really don't know how to continue...

Thank you very much in advance.
Cheers,

Mamen