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

Subject: Re: Different results for same information extracted in different ways
Posted by Klaus Götzen on Fri, 07 Nov 2014 12:55:22 GMT
View Forum Message <> Reply to Message

Hi Mamen,

concerning the discrepancy between the first two approaches (GetEntries() and summing the bin contents), I think a TH1 histogram in reality has n+2 bins, the n bins plotted and an overflow and underflow bin. Therefore you should actually try to loop like

```
int TotalEntries=0;
for (int i=0; i <Nbins+2;i++ )
{
  TotalEntries=TotalEntries+Reco->GetBinContent(i);// histograms start on bin=1 not bin=0
}
cout << "Total Reco Entries: "<< TotalEntries<< endl;
```

In the last case with the loop over the tree itself, I'm not sure. What I could imagine is, that e.g. the variable you set the branch address to has wrong type and then behaves unpredictable. I.e. check whether eppid really is a int branch and not a float branch. Of course you again don't count over- and underflow with this method.

Best,
Klaus

Thanks a lot Klaus,

Looping over Nbins+2 in the histogram created using "->Project();" helps me getting the same number of events than using ->GetEntries();

However I still need to understand why I don't get the same number of entries looping over the tree or doing a Project.
I checked out the values that were being filled in the tree like that:

```
 if (kk % 10000 == 0 && kk != 0)
          {
            cout<<"*** FILLING *** "<< kk << " : "<< cosThetaP << "\t"<< Pz << " / "
<<(sqrt(pow(Px, 2)+pow(Py,2)+pow(Pz,2)))<< endl;
          }

kk++;
```

(cosThetaP, Pz, Px and Py are defined from the branch addressed variables, and cosThetaP is the variable I want at the end fill in the Histogram)
and I saw that costThetaP, as well as Pz and sqrt(...) were showing sometimes "-nan" value.

So I added a variable that counted how many times I was getting nan:

```
   if (isnan(cosThetaP))
            {
              fail++;
               //cout << "cosThe is nan"<<endl;
            }
```

However, the number of entries still don't match.

In addition I have also got the number of entries from the histogram filled looping over the Tree, summing up over Nbins, summing up over Nbins+2 and using GetEntries(), and the results are very strange.

For different files I get:

File	Reco->GetEntries()	Sum=Sum+Reco->GetBinContent(i) from bin=1 to bin=Nbins	Sum=Sum+Reco->GetBinContent(i) from bin=0 to bin=Nbins+1	Nr. -nan events value of fail
1	493075	564152	487270	21476
2	379395	510432	465690	14101

3	646731	373536	0	11467
4	505865	380024	0	22111

==========

Before filling the histogram looping over the tree I've initialized them using

```
for (int b=0; b<Nbins+2; b++)
    {
      Eff->SetBinContent(b, 0);
      Eff->SetBinError(b, 0);

      Reco->SetBinContent(b, 0);
      Reco->SetBinError(b, 0);
    }
```

Can the over/underflow bins have negative values??? even after initializing them to 0 values???

Subject: Re: Different results for same information extracted in different ways
Posted by Klaus Götzen on Fri, 07 Nov 2014 16:14:41 GMT
View Forum Message <> Reply to Message

Hi Mamen,

without hands on the n-tuple (or the root-files) itself, it's hard to find a diagnosis for your problem. According to your table you again have a discrepancy between Project() (1st column) and summing yourself (3rd column), what I don't understand. To reset the histograms you might better use TH1F::Reset(). I don't know exactly how (or whether) the bin weights influence the number of entries.

Best,
Klaus

Subject: Re: Different results for same information extracted in different ways
Posted by Mamen on Fri, 07 Nov 2014 16:26:35 GMT
View Forum Message <> Reply to Message

Dear Klaus,

thank you again for your help.
I will try reseting the histogram using the method you propose.
By now I am trying to write a small test macro that makes the following:

1.- Reads only one file and projects it into a histogram using Projec().
2.- Loops over the histogram using GetBinContent from 0 to NBins+1 (Total Nbins+2)
3.- Loops over the histogram using GetBinContent from bin 1 to NBins (Total Nbins)
4.- GetEntries for the histogram.
5.- Compare all that values
6.- Then I close the file.

Up to now this is working and the values match.

Now I'm trying to:

1.- Open the same file inside the same macro using a different TFile.
2.- Set Branch addresses.
3.- loop over the tree
4.- Fill a histogram applying a cut, in principle the same as with Project()
5.- Repeat points 2 to 6 of the previous block with the new histogram.

This second part is right now not working, even commenting the first part of the code out, the histogram its not filled... :-S
I'm working on it... I'll keep you informed of my progress.

Thanks again for your help.
Mamen

Subject: Re: Different results for same information extracted in different ways
Posted by Mamen on Mon, 10 Nov 2014 14:30:38 GMT
View Forum Message <> Reply to Message

Dear All,

I'm getting more and more confuse about the behavior of root. In order to clarify how a histogram is filled using "->Project()" or looping over a TTree and using "->Fill()" I created a small macro:

```
// c/c++
//#include <stdlib.h>
#include <iostream>


//root

#include "TFile.h"
#include "TTree.h"
#include "TH1D.h"
#include "TH1F.h"
```

```
void FillHisto_test(){
  char Filename[256];
  char *directionName;
  char *Pi0FW;

  double W2=5.;
  int FW=1;
  int NBINS=100;
  int BINMIN=-2;
  int BINMAX=2;

  if(FW==0)
    {
      Pi0FW="bw";
      directionName="backward";
    }
  else if (FW==1)
    {
      Pi0FW="fw";
      directionName="forward";
    }
    sprintf(Filename,
"/home/moraespi/VariableCosThetaGammaStar/Rootfiles/SimuJan2014/electrons/epempi0-W2
-%.0f-Delta0-%s-LargeQ2-Merged.root", W2, Pi0FW);

  cout << "File: "<< Filename<< endl;

  // TCanvas *myCanvas;
  // myCanvas=new TCanvas("C", "C", 1);
  TFile *t=new TFile(Filename);

  TTree *epempi0Tuple  = (TTree*)t->Get("epempi0Tuple");

  // // Project method
  TH1D *Reco;
  Reco = new TH1D ("Reco", "Reco", NBINS, BINMIN, BINMAX);
  epempi0Tuple->Project("Reco", "feppx");//, "eppid>3");
  // Reco->Draw();

  double Total_int=0;
  double Integral=0;
  double GetEntries=0;
  for (int i=0; i<NBINS+2; i++ )
    {
      Total_int=Total_int+Reco->GetBinContent(i);
    }

  for (int j=1; j<NBINS+1; j++ )
    {
      Integral=Integral+Reco->GetBinContent(j);
    }
```

```
  GetEntries=Reco->GetEntries();
  cout << "Number of events in the Histogram using Project: "<<endl;
  cout << " Total_int (0-9)\t Integral (1-8)\t\t GetEntries"<<endl;
  cout << Total_int <<"\t\t\t"<<Integral<<"\t\t\t"<<GetEntries<<endl;
    t->Close();


// // Branch Addresses method

  TFile *t2=new TFile(Filename);

  TTree *epempi0TupleReco  = (TTree*)t2->Get("epempi0Tuple");

  float feppx;
  int eppid;

  epempi0TupleReco->SetBranchAddress("feppx", &feppx);
  epempi0TupleReco->SetBranchAddress("eppid", &eppid);

  TH1F *Reco2;
  Reco2 = new TH1F("Fill", "Fill", NBINS, BINMIN, BINMAX);

  long NEntriesReco=(long)epempi0TupleReco->GetEntries();

   double value=0;
   for (int k=0; k<NEntriesReco; k++)
     {
       epempi0TupleReco->GetEntry(k);
       if (k % 100000 == 0 && k != 0)
  {
    cout<<"*** Reco Loop *** Getting Entry: "<< k << endl;
  }


      // if (eppid>3)
      //  {
   value=feppx;
     Reco2->Fill(value);
   //   }
     }

  double Total_intFill=0;
  double IntegralFill=0;
  double GetEntriesFill=0;
  for (int i=0; i<NBINS+2; i++ )
    {
      Total_intFill=Total_intFill+Reco2->GetBinContent(i);
    }

  for (int j=1; j<NBINS+1; j++ )
    {
      IntegralFill=IntegralFill+Reco2->GetBinContent(j);
```

```
   }
  GetEntriesFill=Reco2->GetEntries();

  cout << "Number of events in the Histogram looping over TTree: "<<endl;
  cout << " Total_intF (0-9)\t IntegralF (1-8)\t\t GetEntriesF"<<endl;
  cout << Total_intFill <<"\t\t\t"<<IntegralFill<<"\t\t\t"<<GetEntriesFill<<endl;

  // //myCanvas->cd();
  // Reco2->SetLineColor(kRed);
  // Reco2->Draw();
  // //myCanvas->SaveAs("Test_Canvas.eps");

    t2->Close();

}
```

As you can see, changing the value of "W2" and "FW" I can select which of the four root files to read:

epempi0-W2-10-Delta0-bw-LargeQ2-Merged.root
epempi0-W2-10-Delta0-fw-LargeQ2-Merged.root
epempi0-W2-5-Delta0-bw-LargeQ2-Merged.root
epempi0-W2-5-Delta0-fw-LargeQ2-Merged.root

I have many problems with this little macro:

1.- If I run

> root FillHisto_test.C+

it crashes for all four root files.

2.- If I run:

> root FillHisto_test.C

it runs only for the file epempi0-W2-5-Delta0-fw-LargeQ2-Merged.root.

3.- I don't get the same number of events in the Histograms using ->Project() or looping over the TTree and doing ->Fill().
(I principally want to understand why, but problems 1.- and 2.- are difficulting me the process).

I am using the following root version:

```
  ****************************************
  *                                      *
  *       W E L C O M E  to  R O O T      *
  *                                      *
  *   Version   5.34/05  14 February 2013   *
  *                                      *
  *  You are welcome to visit our Web site  *
```

```
*        http://root.cern.ch          *
*                                     *
*****************************************
```

ROOT 5.34/05 (tags/v5-34-05@48582, Nov 05 2013, 17:07:52 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.

and I attach to this post:

a.- The macro FillHisto_test.C.

b.- The four root files I want to read, downloadable under:
https://fileshare.zdv.uni-mainz.de/7lGNaukAXiRNxqfPERB7yw.repo
Username: PandaRoot
Password: v3UyxFX8

c.- A file Output.txt with the outputs of the macro run with the four files, compiled and not compiled.

If somebody has any ideas, or can help me in some way, I would be very, very thankful.

Thanks a lot in advance.
Mamen

File Attachments

1) FillHisto_test.C, downloaded 453 times
2) Output.txt, downloaded 396 times

Subject: Re: Different results for same information extracted in different ways
Posted by Mamen on Mon, 10 Nov 2014 15:41:04 GMT
View Forum Message <> Reply to Message

Dear All,

I'm getting more and more confuse about the behavior of root. In order to clarify how a histogram is filled using "->Project()" or looping over a TTree and using "->Fill()" I created a small macro:

```
// c/c++
//#include <stdlib.h>
#include <iostream>
```

```
//root

#include "TFile.h"
#include "TTree.h"
#include "TH1D.h"
#include "TH1F.h"

void FillHisto_test(){
  char Filename[256];
  char *directionName;
  char *Pi0FW;

  double W2=5.;
  int FW=1;
  int NBINS=100;
  int BINMIN=-2;
  int BINMAX=2;

  if(FW==0)
    {
      Pi0FW="bw";
      directionName="backward";
    }
  else if (FW==1)
    {
      Pi0FW="fw";
      directionName="forward";
    }
    sprintf(Filename,
"/home/moraespi/VariableCosThetaGammaStar/Rootfiles/SimuJan2014/electrons/epempi0-W2
-%.0f-Delta0-%s-LargeQ2-Merged.root", W2, Pi0FW);

  cout << "File: "<< Filename<< endl;

  // TCanvas *myCanvas;
  // myCanvas=new TCanvas("C", "C", 1);
  TFile *t=new TFile(Filename);

  TTree *epempi0Tuple  = (TTree*)t->Get("epempi0Tuple");

  // // Project method
  TH1D *Reco;
  Reco = new TH1D ("Reco", "Reco", NBINS, BINMIN, BINMAX);
  epempi0Tuple->Project("Reco", "feppx");//, "eppid>3");
  // Reco->Draw();

  double Total_int=0;
  double Integral=0;
  double GetEntries=0;
  for (int i=0; i<NBINS+2; i++ )
    {
      Total_int=Total_int+Reco->GetBinContent(i);
```

```
  }

 for (int j=1; j<NBINS+1; j++ )
  {
    Integral=Integral+Reco->GetBinContent(j);
  }
 GetEntries=Reco->GetEntries();
 cout << "Number of events in the Histogram using Project: "<<endl;
 cout << " Total_int (0-9)\t Integral (1-8)\t\t GetEntries"<<endl;
 cout << Total_int <<"\t\t\t"<<Integral<<"\t\t\t"<<GetEntries<<endl;
  t->Close();


// // Branch Addresses method

 TFile *t2=new TFile(Filename);

 TTree *epempi0TupleReco  = (TTree*)t2->Get("epempi0Tuple");

 float feppx;
 int eppid;

 epempi0TupleReco->SetBranchAddress("feppx", &feppx);
 epempi0TupleReco->SetBranchAddress("eppid", &eppid);

 TH1F *Reco2;
 Reco2 = new TH1F("Fill", "Fill", NBINS, BINMIN, BINMAX);

 long NEntriesReco=(long)epempi0TupleReco->GetEntries();

  double value=0;
  for (int k=0; k<NEntriesReco; k++)
   {
     epempi0TupleReco->GetEntry(k);
     if (k % 100000 == 0 && k != 0)
{
  cout<<"*** Reco Loop *** Getting Entry: "<< k << endl;
}


    // if (eppid>3)
    //  {
  value=feppx;
    Reco2->Fill(value);
  //   }
   }

 double Total_intFill=0;
 double IntegralFill=0;
 double GetEntriesFill=0;
 for (int i=0; i<NBINS+2; i++ )
  {
```

```
    Total_intFill=Total_intFill+Reco2->GetBinContent(i);
   }

  for (int j=1; j<NBINS+1; j++ )
   {
    IntegralFill=IntegralFill+Reco2->GetBinContent(j);
   }
  GetEntriesFill=Reco2->GetEntries();

  cout << "Number of events in the Histogram looping over TTree: "<<endl;
  cout << " Total_intF (0-9)\t IntegralF (1-8)\t\t GetEntriesF"<<endl;
  cout << Total_intFill <<"\t\t\t"<<IntegralFill<<"\t\t\t"<<GetEntriesFill<<endl;

  // //myCanvas->cd();
  // Reco2->SetLineColor(kRed);
  // Reco2->Draw();
  // //myCanvas->SaveAs("Test_Canvas.eps");

    t2->Close();

}
```

As you can see, changing the value of "W2" and "FW" I can select which of the four root files to read:

epempi0-W2-10-Delta0-bw-LargeQ2-Merged.root
epempi0-W2-10-Delta0-fw-LargeQ2-Merged.root
epempi0-W2-5-Delta0-bw-LargeQ2-Merged.root
epempi0-W2-5-Delta0-fw-LargeQ2-Merged.root

I have many problems with this little macro:

1.- If I run

> root FillHisto_test.C+

it crashes for all four root files.

2.- If I run:

> root FillHisto_test.C

it runs only for the file epempi0-W2-5-Delta0-fw-LargeQ2-Merged.root.

3.- I don't get the same number of events in the Histograms using ->Project() or looping over the TTree and doing ->Fill().
(I principally want to understand why, but problems 1.- and 2.- are difficulting me the process).

I am using the following root version:

```
  ****************************************
```

```
 *                          *
 *      W E L C O M E  to  R O O T      *
 *                          *
 *   Version   5.34/05  14 February 2013   *
 *                          *
 *  You are welcome to visit our Web site  *
 *         http://root.cern.ch          *
 *                          *
 ****************************************
```

ROOT 5.34/05 (tags/v5-34-05@48582, Nov 05 2013, 17:07:52 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.

and I attach to this post:

a.- The macro FillHisto_test.C

b.- The four root files I want to read, can be found under:
https://fileshare.zdv.uni-mainz.de/7lGNaukAXiRNxqfPERB7yw.repo
Username: PandaRoot
Password: v3UyxFX8

c.- A file Output.txt with the outputs of the macro run with the four files, compiled and not compiled.

If somebody has any ideas, or can help me in some way, I would be very, very thankful.

Thanks a lot in advance.
Mamen

File Attachments

1) FillHisto_test.C, downloaded 380 times
2) Output.txt, downloaded 415 times

Subject: Re: Different results for same information extracted in different ways
Posted by Klaus Götzen on Mon, 10 Nov 2014 16:13:46 GMT
View Forum Message <> Reply to Message

Hi Mamen,

you are setting the branch addresses with wrong type, feppx and eppid are arrays[ncnd]. If you check, you see

root [5] epempi0Tuple.Print("feppx*")

```
*******************************************************************************
*Br    0 :feppx     : feppx[ncnd]/F                                          *
*Entries :   477980 : Total  Size=    3999009 bytes  File Size  =    2757478 *

root [6] epempi0Tuple.Print("eppid*")
*******************************************************************************
*Br    0 :eppid     : eppid[ncnd]/I                                          *
*Entries :   477980 : Total  Size=    3999002 bytes  File Size  =    1021207 *
```

The number 477980 is exactly the number of events you see in your second approach.

You should do the loop like this:

```
  float feppx[100];
  int eppid[100];
  int ncnd;

  epempi0TupleReco->SetBranchAddress("feppx", &feppx);
  epempi0TupleReco->SetBranchAddress("eppid", &eppid);
  epempi0TupleReco->SetBranchAddress("ncnd", &ncnd);

  TH1F *Reco2;
  Reco2 = new TH1F("Fill", "Fill", NBINS, BINMIN, BINMAX);

  long NEntriesReco=(long)epempi0TupleReco->GetEntries();

  for (int k=0; k<NEntriesReco; k++)
  {
    epempi0TupleReco->GetEntry(k);

    if (k % 100000 == 0 && k != 0) cout<<"*** Reco Loop *** Getting Entry: "<< k << endl;

    for (int kk=0;kk<ncnd;++kk)
       Reco2->Fill(feppx[kk]);
   }
```

With that I get the output:

```
File: epempi0-W2-10-Delta0-bw-LargeQ2-Merged.root
Number of events in the Histogram using Project:
 Total_int (0-9)        Integral (1-8)         GetEntries
500104                 500104                 500104
*** Reco Loop *** Getting Entry: 100000
*** Reco Loop *** Getting Entry: 200000
*** Reco Loop *** Getting Entry: 300000
*** Reco Loop *** Getting Entry: 400000
Number of events in the Histogram looping over TTree:
 Total_intF (0-9)       IntegralF (1-8)              GetEntriesF
500104                 500104                 500104
```

Best,
Klaus

Subject: Re: Different results for same information extracted in different ways
Posted by Stefan Pflueger on Mon, 10 Nov 2014 16:41:05 GMT
View Forum Message <> Reply to Message

Hi Mamen,

so when I run your macro in the compiled mode (.C+) it runs fine (sometimes I get errors but all the way at the end, when everything is clean up). From what I can see in your Output file, your crashes appear also all the way at the end so all results etc should be fine. The segfault is annoying nevertheless... I don't see why it crashes at this point, but since it is all the way at the end it has to do with cleanup of your objects (destructors are called). I would guess that somehow the memory management of root screws up there.
When I run the macros in interpreter mode (.C) it runs fine on some files and crashes on other files. In my case it crashes when filling the Reco2 histogram at some defined entry (not the last) of the tree. Debugging this nearly impossible (afaik), as more or less non logical stuff happens. However I traced it back to the #include statements. So removing the them should make your macro run just just like in the compiled mode (crashes at the end of the macro when cleanup happens, but that is no biggie). Hope that helps...

Small remark: I would recommend to replace the char array and sprint statements with stringstreams as they are a bit safer, so something like this:

```
int W2=10;
std::string FWBW="bw";
int NBINS=100;
int BINMIN=-2;
int BINMAX=2;

std::stringstream ss;
ss<<"epempi0-W2-"<<W2<<"-Delta0-"<<FWBW<<"-LargeQ2-Merged.root";
```

Cheers,

Stefan

Subject: Re: Different results for same information extracted in different ways
Posted by Mamen on Mon, 10 Nov 2014 16:51:02 GMT
View Forum Message <> Reply to Message

Dear Klaus,

thank you, thank you, thank you SO MUCH!!!

I have been struggling with this problem since last week.
Addressing the variables as arrays seems to solve the problem in the small macro. Also the crashes are gone
Now I will try to implement that also in my original macro, where first the inconsistency between the number of events in the Histogram filled with one or the other method appeared.
I hope I can solve the problem also there.

Thank you again, for your help and your fast answer!
Cheers,

Mamen.

Subject: Re: Different results for same information extracted in different ways
Posted by Mamen on Mon, 10 Nov 2014 16:54:02 GMT
View Forum Message <> Reply to Message

Dear Stefan,

thank you very much for your suggestions.
I followed the instructions given by Klaus in the previous post, and it seems the crashes are gone and I get the same number of events in the histograms filled with the two different methods. So in principle the problems are solved.

However, I will also take your suggestions about the string-streams into account.

Cheers,
Mamen
