



中国科学技术大学
University of Science and Technology of China

ROOT and RooFit

彭高乐

From 宋海林, 王咪, 王博, 秦龙宇

2025年7月2日



- ROOT 介绍
- 一些例子
- RooFit 介绍
- 一些例子
- 总结

ROOT 介绍



中国科学技术大学
University of Science and Technology of China



ROOT
Data Analysis Framework

[About](#)[Learn](#)[Install](#)[Forum & Help](#)[Manual](#)[Blog Posts](#)[Contribute](#)[For Developers](#)

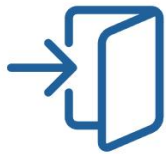
ROOT: analyzing petabytes of data, scientifically.

An open-source data analysis framework used by high energy physics and others.

[Learn more](#)[Install v6.32.02](#)

基于面向对象设计的数据分析软件框架！

ROOT的主页，由CERN维护。



Learn



Reference



Forum



Gallery

$\sqrt{-1}$

ROOT enables *statistically sound* scientific analyses and visualization of large amounts of data: today, more than 1 exabyte (1,000,000,000 gigabyte) are stored in ROOT files. The Higgs was found with ROOT!



As *high-performance* software, ROOT is written mainly in C++. You can use it on Linux, macOS, or Windows; it works out of the box. ROOT is [open source](#): use it freely, [modify it](#), [contribute to it](#)!

\$ _

ROOT comes with an [incredible C++ interpreter](#), ideal for *fast prototyping*. Don't like C++? ROOT integrates super-smoothly with Python thanks to its [unique dynamic and powerful Python \$\rightleftharpoons\$ C++ binding](#). Or what about using ROOT in a Jupyter notebook?



- Framework for large scale data handling
- Provides, among others,
 - an efficient data storage, access and query system (PetaBytes)
 - advanced statistical analysis: histogramming, fitting, minimization and multi-variate analysis algorithms
 - scientific visualization: 2D and 3D graphics, Postscript, PDF, LaTeX
 - geometrical modeler
 - PROOF parallel query engine
- An Open Source Project
- User Interface
 - GUI: Browsers, Panels, Tree Viewer
 - Command Line interface: C++ interpreter CINT
 - Script Processor (C++ compiled C++ interpreted)

ROOT 介绍



中国科学技术大学
University of Science and Technology of China

- Start:

```
$ root          Start the ROOT
$ root -l       Does not show the ROOT banner
root [0] .q     quit the ROOT
root [1] .x <file_name>      .x name.cxx (execute)
root [2] .L <file_name>     .L name.cxx (load)

$ root -l name.root  open file
root [0] .ls         look list
root [1] treename->Print()    list all branches in this tree
root [2] treename->Draw("branch")  draw branch
root [3] treename->Scan()      print message
```

- Calculator root

```
[0] 1+1
root [1] 2*(50+91)/12.
root [2] sqrt(3.)
root [3] TMath::Pi()
```

```
> root -h
```

Usage: root [-l] [-b] [-n] [-q] [file1.C ... fileN.C] Options:

-b : run in batch mode without graphics

以批处理形式运行不显示图，远程操作需要，减少等待时间和流量。

-n : do not execute logon and logoff macros as specified in .rootrc

-q : exit after processing command line script files

运行完脚本文件后自动退出，非常有用，在脚本中运行root需要。

-l : do not show the image logo (splash screen)

```
*****
*
*           W E L C O M E  t o  R O O T
*
*   Version  5.34/09           26 June 2013
*
*   You are welcome to visit our Web site
*   http://root.cern.ch
*
*****

ROOT 5.34/09 (v5-34-09@v5-34-09, Jun 26 2013, 17:10:36 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 { }.
```

Creating ROOT macros

The name of the ROOT macro and the file name (without file extension) in which the macro is saved must match.

1. Create a new file in your preferred text editor.
2. Use the following general structure for the ROOT macro, preferably with a function that has the same name as the file:

```
void MacroName() {
    ...

    your lines of C++ code
    code line ends with ;

    ...
}
```

1. Save the file ROOT macro, using the macro name as file name: **MacroName.C**

Note

It's not necessary to #include anything in the ROOT macros. Everything in the include paths is automatically included. Note that you can type .i in the ROOT prompt to see the include paths, and .I [path] to add an extra path.



- \$ROOTSYS/tutorials: contains **many example scripts**. They assume some basic knowledge of C++ and ROOT

copy them to your home directory and try!

- Some Reference: <https://root.cern/learn/>

- The **ROOT beginners' guide** (aka "Primer")
- The **Manual** where more in depth information can be found
- The full **API Documentation**
- The **Tutorials**

```
-bash-4.2$ cd /scratchfs/bes/wwangbo/draw/draw/cxx/  
-bash-4.2$ ls  
draw1.cxx draw2.cxx opeTree.cxx
```

- You can find sample program of next two paths:

/scratchfs/bes/wwangbo/draw/draw/cxx/ (ihepc)

/ustcfs/BES3User/2021/bwang/draw/ (ustc)

draw1.cxx and **draw2.cxx** are used to draw figures

opetree.cxx is used to modify tree in the root file

读文件建立直方图

```
void draw1(){  
  
    gStyle->SetOptStat(0);  
    //information in statistics via parameter mode  
  
    TFile *File = new TFile("../rhopi3097_001.root");  
    //read file  
    TTree *Tree = (TTree*)File->Get("fit5c");  
    //read tree in the file  
    TFile *File1 = new TFile("../rhopi3097_002.root");  
    TTree *Tree1 = (TTree*)File1->Get("fit5c");  
    TFile *File2 = new TFile("../rhopi3097_003.root");  
    TTree *Tree2 = (TTree*)File2->Get("fit5c");  
    TFile *File3 = new TFile("../rhopi3097_004.root");  
    TTree *Tree3 = (TTree*)File3->Get("fit5c");  
  
    TCut cut = "";  
    //selection  
    //cut1+cut2=(cut1)&&(cut2)      cut1||cut2=(cut1)||(cut2)  
  
    double bin(100), lo(0), up(3);  
    //bins and the range  
    TString val = "mrh0";  
    //name of variable  
  
    TH1F *h = new TH1F("h", "", bin, lo, up);  
    //H for histogram, F for float  
    TH1F *H[5];  
    //Hist in array form  
    for(int i=0; i<5 ;i++)  
    {
```

输入需要绘图的root
文件

输入root中待分析的
tree

创建一个新的直方图, bin
指的是区间数lo,up指的是
上下限

建立画布 将变量加进直方图

```
TH1F *H[5];
//Hist in array form
for(int i=0; i<5 ;i++)
{
H[i] = new TH1F(Form("H%d", i), "", bin, lo, up);
}

TCanvas *c = new TCanvas("c","c",10,10,800,600);
//define a canvas
Tree->Draw(val+">>h",cut);
//consider as signal
Tree1->Draw(val+">>H0",cut);
Tree2->Draw(val+">>H1",cut);
Tree3->Draw(val+">>H2",cut);
//consider as data

h->SetYTitle("Events");
h->SetXTitle("M_{#rho^{0}}");
h->GetYaxis()->SetLabelSize(0.043);
//size of label of Yaxis
h->GetYaxis()->SetRangeUser(0,10000);
//range of Yaxis

h->SetLineWidth(3); //TAttLine Class
h->SetLineColor(kRed);
h->Scale(2); //scale

H[2]->SetMarkerSize(0.9); //TAttMarker Class
H[2]->SetMarkerStyle(8);
H[2]->SetLineWidth(2);
H[2]->Scale(2*h->GetEntries()/H[2]->GetEntries());
```

将Tree中的 mrho画
在画布上

设置图像的具体
细节，比如线的
粗细，点的大小，
坐标轴名称之类
的

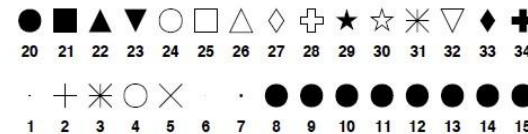
设置堆积图 一般用来描述本底

```
THStack *hs = new THStack("hs", "");
H[0] -> SetLineColor(kOrange);
H[0] -> SetLineWidth(2);
H[0] -> SetFillColor(kOrange);
H[0] -> SetFillStyle(3344);
H[1] -> SetLineColor(kBlue);
H[1] -> SetLineWidth(2);
H[1] -> SetFillColor(kBlue);
H[1] -> SetFillStyle(3344);
hs -> Add(H[0]);
//add hist
hs -> Add(H[1]);
```

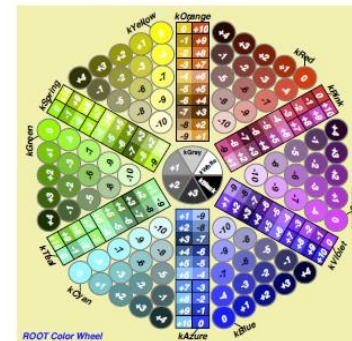
画出直方图

```
h -> Draw("hist");
hs -> Draw("same");
H[2] -> Draw("same hist ep");
//e for error, p for point
```

Available markers in ROOT



Available Colors



you can access them from the Canvas View Menu

Graphs can be drawn with the following options:

Option	Description
"A"	Axis are drawn around the graph
"L"	A simple polyline is drawn
"F"	A fill area is drawn ('CF' draw a smoothed fill area)
"C"	A smooth Curve is drawn
"*"	A Star is plotted at each point
"P"	The current marker is plotted at each point
"B"	A Bar chart is drawn
"I"	When a graph is drawn as a bar chart, this option makes the bars start from the bottom of the pad. By default they start at 0.
"X+"	The X-axis is drawn on the top side of the plot.
"Y+"	The Y-axis is drawn on the right side of the plot.

```

TLegend *leg = new TLegend(0.6, 0.7, 0.85, 0.85);
//legend
leg->SetFillColor(0);
leg->SetNColumns(2);
//2 columns
leg->SetBorderSize(0);
leg->AddEntry(h, "signal", "l");
//l for line, f for fill, e for error, p for point
leg->AddEntry(H[2], "data", "ep");
leg->AddEntry(H[0], "back1", "f");
leg->AddEntry(H[1], "back2", "f");
leg->Draw();

```

```

TArrow *arrow = new TArrow(1.2, 200, 1.2, 0, 0.02, ">");

```

```

//TArrow Class

```

```

arrow->SetFillColor(1);
arrow->SetFillStyle(1001);
arrow->SetLineColor(kGreen+2);
arrow->SetLineWidth(2);
arrow->Draw();

```

```

TPaveText *pt = new TPaveText(0.5, 0.5, 0.85, 0.7, "BRNDC");

```

```

pt->SetFillColor(10);

```

```

pt->SetBorderSize(10);

```

```

pt->SetTextAlign(22);

```

```

TString par1 = "show show way";

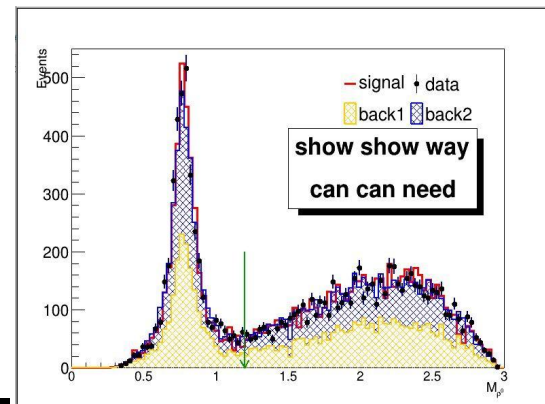
```

```

TString par2 = "can can need";

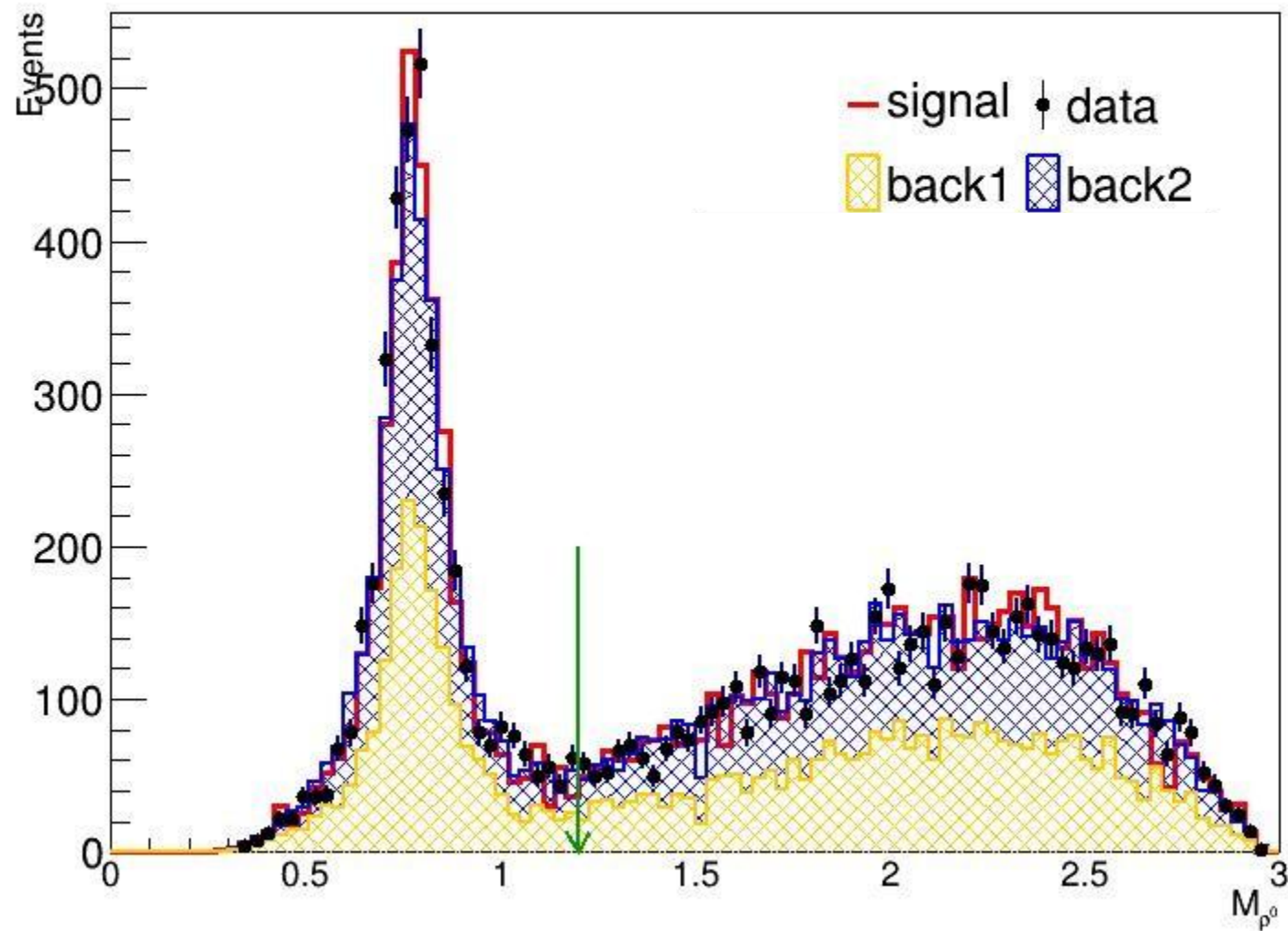
```

设置图例



设置箭头

设置文本框



另一种读取root的方式

```
TChain *chain = new TChain("fit5c");  
chain->Add("../rhopi3097_001.root");  
chain->Add("../rhopi3097_005.root");
```

```
double m1,m2,m3,m4,chisq;  
chain->SetBranchAddresses("mrh0",&m1);  
chain->SetBranchAddresses("chi2",&chisq);  
Tree1->SetBranchAddresses("mrh0",&m2);  
Tree2->SetBranchAddresses("mrh0",&m3);  
Tree3->SetBranchAddresses("mrh0",&m4);  
  
for(int i=0; i<chain->GetEntries(); i++){  
    chain->GetEntry(i);  
    /*if(chisq<50)*/ sig->Fill(m1);  
}  
for(int i=0; i<Tree1->GetEntries(); i++){  
    Tree1->GetEntry(i);  
    data->Fill(m2);  
}  
for(int i=0; i<Tree2->GetEntries(); i++){  
    Tree2->GetEntry(i);  
    back1->Fill(m3);  
}  
for(int i=0; i<Tree3->GetEntries(); i++){  
    Tree3->GetEntry(i);  
    back2->Fill(m4);
```

这样可以一次性读取多个root文件

另一种将变量填入直方图的方式，这里的data, back1, 2都是前面的定义过的直方图

将画布分为两部分，打开第一部分（内容与draw1.cxx类似）

```
TCanvas *c = new TCanvas("c","c",10,10,1200,500);  
c->Divide(2,1);  
c->cd(1);
```

```
c->cd(2);  
double x[10], y[10];  
for(int i=0; i<10; i++){  
    x[i] = i+1;  
    y[i] = sin(x[i])/x[i]+0.1;  
}  
TGraph *gr = new TGraph(10,x,y);  
  
gr->SetLineWidth(3);  
gr->SetLineColor(2);  
gr->SetMarkerStyle(4);  
gr->SetMarkerColor(3);  
  
gr->GetYaxis()->SetRangeUser(-0.3,1.2);  
gr->Draw("ACP");
```

→ 打开第二部分，使用已知数组画图，这里等于画了一个函数的散点图。

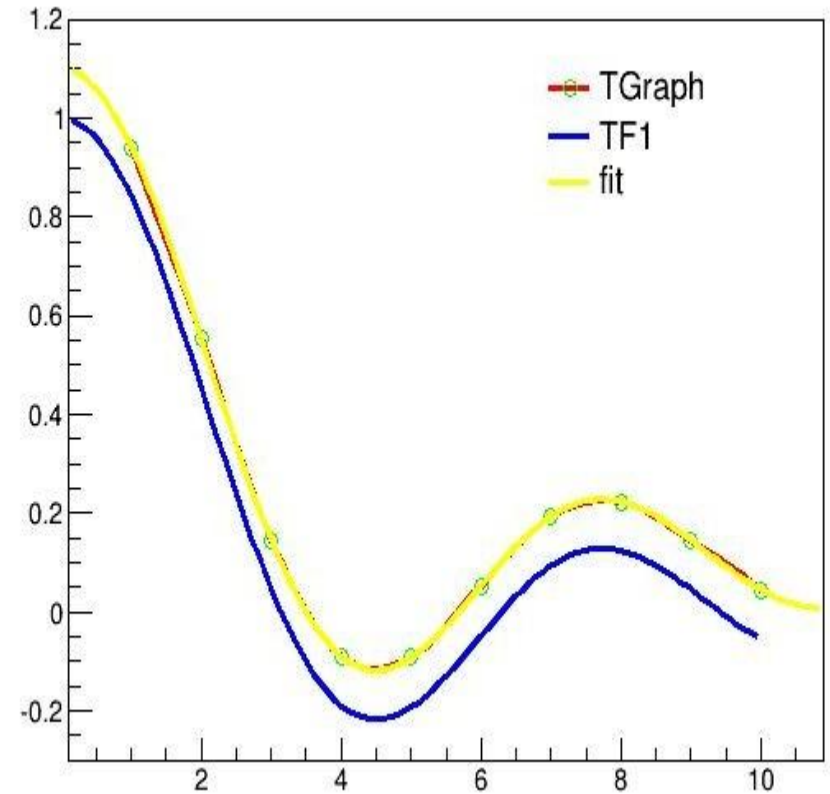
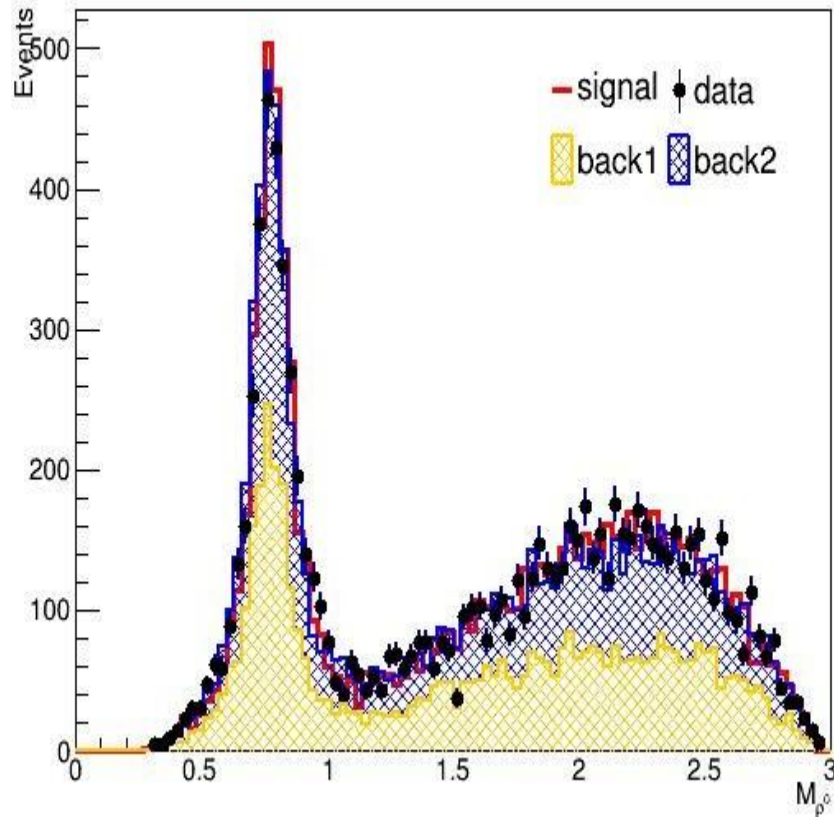
使用公式画图与超简单拟合

```
TF1 *f = new TF1("fun", "sin(x)/x", 0, 10);  
f->SetLineWidth(3);  
f->SetLineColor(4);  
f->Draw("same");  
  
TF1 *f_fit = new TF1("f_fit", "sin(x)/x+[0]", 0, 10);  
f_fit->SetLineWidth(3);  
f_fit->SetLineColor(5);  
f_fit->SetParameter(0, 1);  
gr->Fit("f_fit");  
f_fit->Draw("same");
```

新定义一个函数
绘制其连续
图像

对刚刚绘制的函数
散点图进行拟合

Graph



读取文件和tree

```
TFile *infile = new TFile("../rhopi3097_010.root");  
TTree *intree1 = (TTree*)infile->Get("vxyz");  
TTree *intree2 = (TTree*)infile->Get("photon");  
TTree *intree3 = (TTree*)infile->Get("dedx");  
TTree *intree4 = (TTree*)infile->Get("tof1");  
TTree *intree5 = (TTree*)infile->Get("tof2");  
TTree *intree6 = (TTree*)infile->Get("pid");  
TTree *intree7 = (TTree*)infile->Get("etot");  
TTree *intree8 = (TTree*)infile->Get("fit4c");  
TTree *intree9 = (TTree*)infile->Get("fit5c");  
TTree *intree10 = (TTree*)infile->Get("geff");  
TTree *intree11 = (TTree*)infile->Get("tofe");
```


综合性操作 加减branch 添加cut条件

```
//fit5c
double chisq, mrh0_1, mrhp_1, mrhm_1;
intree9->SetBranchAddresses("mrh0",&mrh0_1);
//read branch
intree9->SetBranchAddresses("mrhp",&mrhp_1);
intree9->SetBranchAddresses("chi2",&chisq);
intree9->SetBranchStatus("mrhm",0);
//0 means deactive; don't clone this branch while cloning in tree intree9

TFile *newfile = new TFile("../test.root", "recreate");
TTree *newfit5c = intree9->CloneTree(0);
//clone without value in the branch

double sum;
TBranch *Br1 = newfit5c->Branch("sum",&sum);
//built a new branch of tree newfit5c

for(int i=0; i<intree9->GetEntries(); i++)
{
    intree9->GetEntry(i);
    sum = mrh0_1+mrhp_1;
    if(chisq<50) newfit5c->Fill();
}
newfit5c->Print();
//newfit5c->Write();
```

```
//define the new variable
//add cut
```

```
//Copy tree
intree1->SetBranchStatus("*", 0);
//all deactive
intree1->SetBranchStatus("vx0", 1);
//1 mean active

TTree *newtree1 = intree1->CloneTree();
//clone all without fill
newtree1->Print();
//newtree1->Write();

//Cut tree
TTree *newfit4c = intree8->CopyTree("chi2<50");
newfit4c->Print();
//newfit4c->Write();

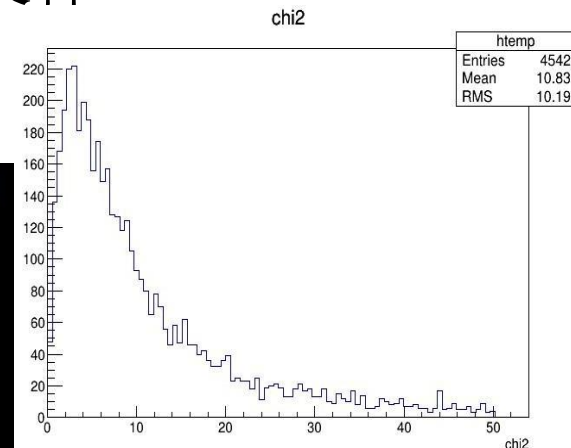
newfile->Write();
//write the file
//delete newfile;
```

复制tree

添加cut条件

写入新文件

```
[wangyijing@lxslc705 ~/draw]$ root -l test.root
^[[Aroot [0]
Attaching file test.root as _file0...
root [1] .ls
TFile**          test.root
TFile*           test.root
KEY: TTree       fit5c;1 ks N-Tuple example
KEY: TTree       vx0;1 ks N-Tuple example
KEY: TTree       fit4c;1 ks N-Tuple example
root [2]
```





- 什么是拟合
 - ✓ 我们的数据一般是一些离散的点，我们用比如极大似然值法，得到一个最好的参数化公式去描述数据。
- 为什么拟合
 - ✓ 我们得到的参数化公式往往可能蕴含某种物理规律。比如粒子物理最常见的去拟合不变质量谱得到一个**BW**函数，就是研究一个我们感兴趣的共振态的最直接有效的方法。

Fit Methods:

Maximum Likelihood (ML) Fit

χ^2 Fit



What is it?

- A powerful toolkit for modeling the expected distribution(s) of events in a physics analysis
- Primarily targeted to high-energy physicists using ROOT
- Originally developed for the BaBar collaboration by Wouter Verkerke and David Kirkby.
- Included with ROOT v5.xx

Documentation:

- <http://root.cern.ch/root/Reference.html> – for latest class descriptions. RooFit classes start with “Roo”.
- <http://roofit.sourceforge.net> – for documentation and tutorials

Tutorials:

- Dig \$ROOTSYS/tutorials/rootfit



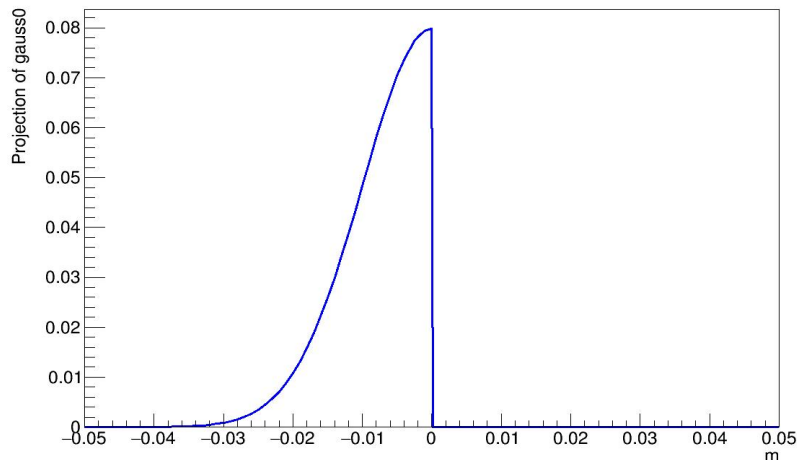
- Mathematical objects are represented as C++ objects

Mathematical concept			RooFit class
variable	x	➔	<code>RooRealVar</code>
function	$f(x)$	➔	<code>RooAbsReal</code>
PDF	$f(x)$	➔	<code>RooAbsPdf</code>
space point	$\vec{x}$	➔	<code>RooArgSet</code>
integral	$\int_{x_{\min}}^{x_{\max}} f(x) dx$	➔	<code>RooRealIntegral</code>
list of space points		➔	<code>RooAbsData</code>



- RooGaussian
- RooExponential
- RooPolynomial
- RooChebychev
- RooBreigWigner
- RooArgusBG
- RooBiFurGauss
- RooCrystalBall
- Gaussian
- Exponential
- Polynomial
- Chebychev polynomial
- Breit-wigner
- Argus
- Bifurcated Gaussian
- Crystal Ball

这些都算作前面提到的RooAbsPdf! !

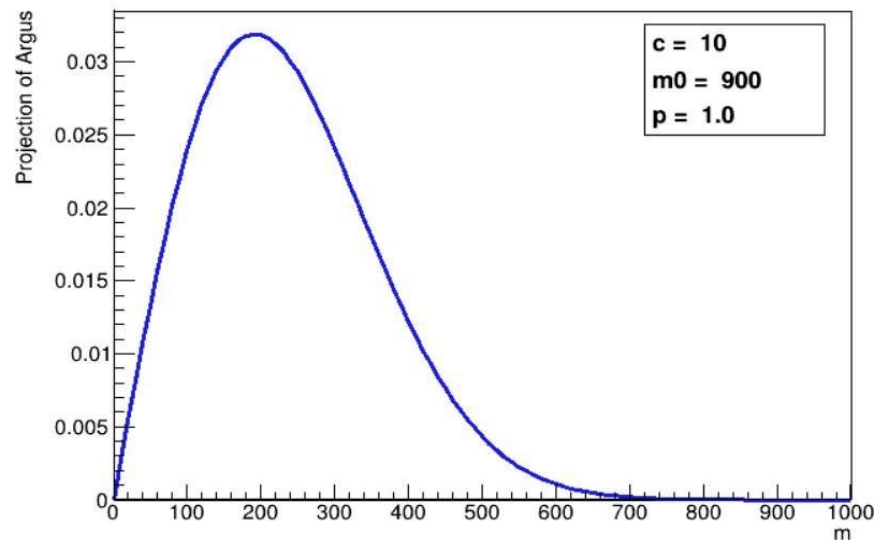


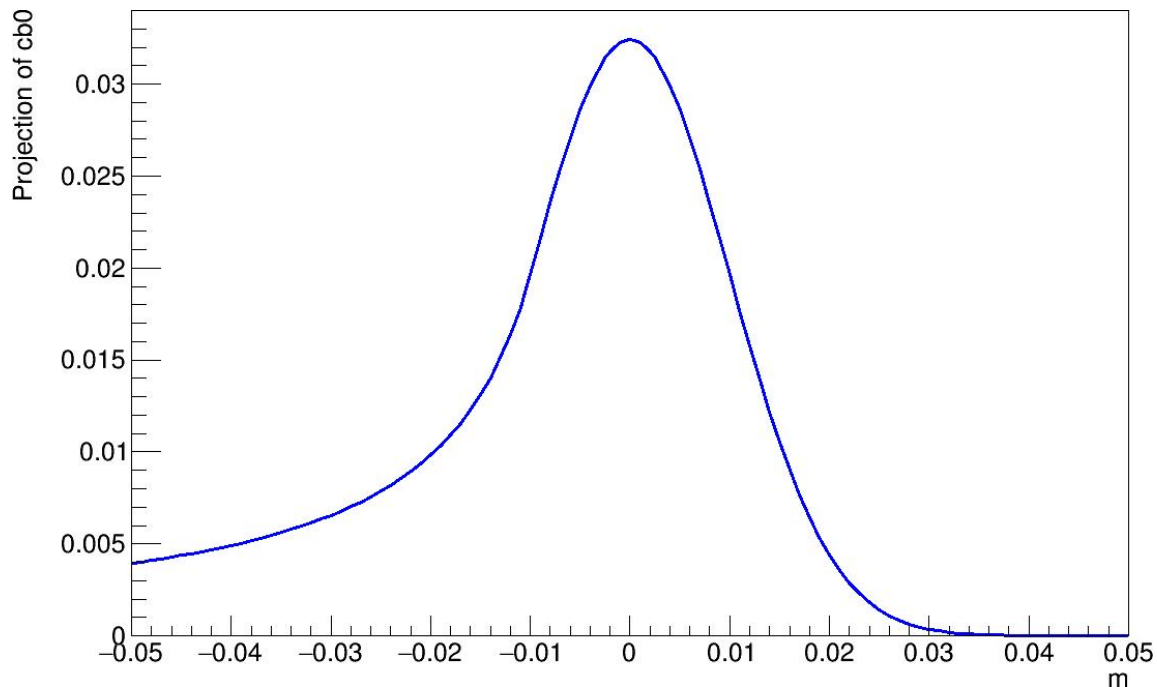
Bifurcated Gaussian

Argus

$$\text{Argus}(m, m_0, c, p) = \mathcal{N} \cdot m \cdot \left[1 - \left(\frac{m}{m_0} \right)^2 \right]^p \cdot \exp \left[c \cdot \left(1 - \left(\frac{m}{m_0} \right)^2 \right) \right]$$

A RooPlot of "m"





Crystal Ball

$$f(m; m_0, \sigma, \alpha_L, n_L, \alpha_R, n_R) = \begin{cases} A_L \cdot (B_L - \frac{m-m_0}{\sigma_L})^{-n_L}, & \text{for } \frac{m-m_0}{\sigma_L} < -\alpha_L \\ \exp\left(-\frac{1}{2} \cdot \left[\frac{m-m_0}{\sigma_L}\right]^2\right), & \text{for } \frac{m-m_0}{\sigma_L} \leq 0 \\ \exp\left(-\frac{1}{2} \cdot \left[\frac{m-m_0}{\sigma_R}\right]^2\right), & \text{for } \frac{m-m_0}{\sigma_R} \leq \alpha_R \\ A_R \cdot (B_R + \frac{m-m_0}{\sigma_R})^{-n_R}, & \text{otherwise,} \end{cases}$$

$$A_i = \left(\frac{n_i}{|\alpha_i|}\right)^{n_i} \cdot \exp\left(-\frac{|\alpha_i|^2}{2}\right)$$

$$B_i = \frac{n_i}{|\alpha_i|} - |\alpha_i|$$

- 最终我们要进行的操作是：把一个用参数 (RooRealVar) 描述的 **PDF(RooAbsPdf)** (比如一个用mean值和sigma值描述的gaussian分布)，去拟合到一个数据集 (**RooDataSet**) 上面，让程序去根据自己的算法，找到这些参数最好的值，来让这个PDF **最精确**地表述这个数据集。

Estimator !!!

- A perfect estimator is (一致性, 无偏性, 有效性)
 - Consistent: $\lim_{n \rightarrow \infty} (\hat{a}) = a$
 - Unbiased – With finite statistics you get the right answer on average
 - Efficient $V(\hat{a}) = \langle (\hat{a} - \langle \hat{a} \rangle)^2 \rangle$ ← This is called the **最小方差界**
Minimum Variance Bound

No perfect estimators !!

复制一下先！



- 先进到自己的主目录下随便创个文件夹，进去之后cp:
- `cp -r /besfs5/groups/jpsi/jpsigroup/user/qinlongyu/shareble/roofit ./`
- `cd roofit`
- `source ROOT6_24.csh`
 - ✓ 对于某些复杂的拟合，root版本的差别对拟合结果会有挺大的影响(建议用root6)
- `vi gauss1.cxx`

一个简单的例子



- vi gauss1.cxx

```
14 void gauss1()
15 {
16     //define my variable
17     RooRealVar x("x","x",-10,10) ;
18
19     //define a gaussian distribution
20     RooRealVar mean("mean","mean of gaussian",1,-10,10) ;
21     RooRealVar sigma("sigma","width of gaussian",1,0.1,10) ;
22     RooGaussian gauss("gauss","gaussian PDF",x,mean,sigma) ;
23
24     // Generate a dataset of 1000 events in x from gauss
25     RooDataSet* data = gauss.generate(x,10000) ;
26
27     //define another gaussian distribution
28     RooRealVar mean1("mean1","mean1 of gaussian",2,-10,10) ;
29     RooRealVar sigma1("sigma1","width1 of gaussian",2,0.1,10) ;
30     RooGaussian gauss1("gauss1","gaussian PDF1",x,mean1,sigma1) ;
31
32
33
34
35
36     // Draw all frames on a canvas
37     RooPlot* xframe = x.frame() ;
38     RooPlot* xframe2 = x.frame() ;
39     TCanvas* c = new TCanvas("c","c",800,400) ;
40
41     c->Divide(2) ;
42     c->cd(1) ;
43     data->plotOn(xframe) ;
44     gauss.plotOn(xframe) ;
45     xframe->Draw() ;
46
47     c->cd(2) ;
48     //this line means fit pdf to dataset
49     gauss1.fitTo(*data) ;
50     data->plotOn(xframe2) ;
51     gauss1.plotOn(xframe2) ;
52     xframe2->Draw() ;
53
54 }
```

先定义一个高斯pdf (gauss)

用这个高斯pdf产生了一个dataset

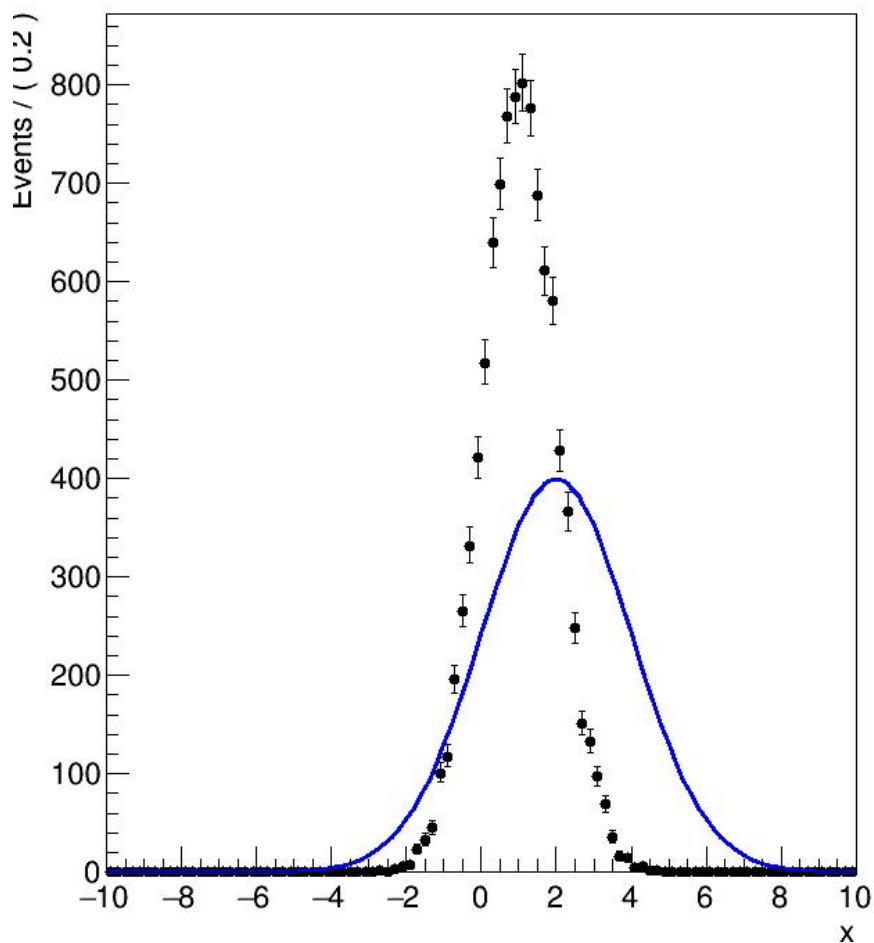
定义参数不一样的高斯pdf (gauss1)

定义两个plot(类似于TH1F) 和一个画布

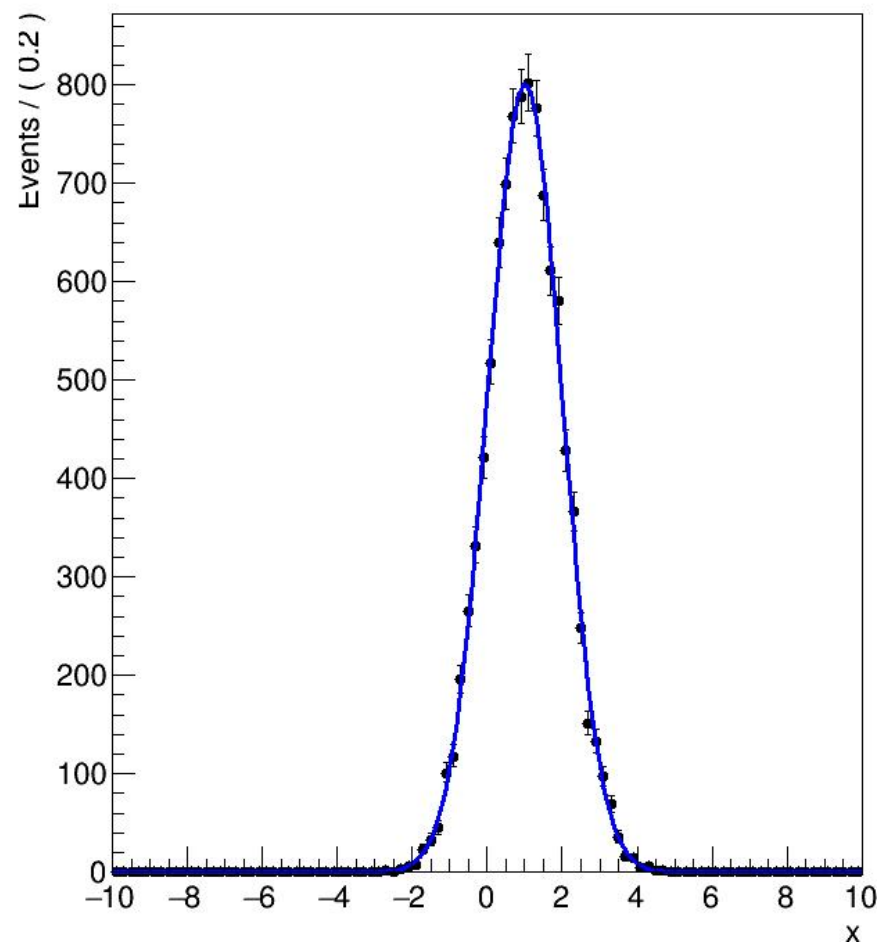
在左边的画布上画出前面产生的dataset, 以及初始条件下的gauss1

在右边的画布上画出前面产生的dataset, 以及拟合后的gauss1

A RooPlot of "x"



A RooPlot of "x"



Estimate Distance to Minimum, 越小越好

OK表示拟合成功

卡方最小值

拟合结果

相关系数

```
*****
**  18 **HESSE      1000
*****
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=14162.8 FROM HESSE STATUS=OK
EDM=3.21094e-07 STRATEGY= 1 ERROR MATRIX ACCURATE
10 CALLS      66 TOTAL
EXT PARAMETER      INTERNAL      INTERNAL
NO.  NAME      VALUE      ERROR  STEP SIZE      VALUE
1  mean1      1.00665e+00  9.97346e-03  1.64750e-05  1.00836e-01
2  sigma1      9.97346e-01  7.05227e-03  8.15530e-06 -9.59175e-01
ERR DEF= 0.5
EXTERNAL ERROR MATRIX.  NDIM= 25  NPAR= 2  ERR DEF=0.5
9.947e-05  8.427e-09
8.427e-09  4.973e-05
PARAMETER CORRELATION COEFFICIENTS
NO.  GLOBAL      1      2
1  0.00012  1.000  0.000
2  0.00012  0.000  1.000
```



- 在实际工作当中，我们的dataset一般是会从root文件里面引入
- 而且dataset的构成会比较复杂，一般不会是一个高斯分布就能很轻易描述
- 需要各种pdf按不同的权重求和(RooAddPdf)，构成新的pdf以描述更复杂的dataset.



- vi omegabw.cxx

```
1 using namespace RooFit;
2 void omegabwfit()
3 {
4
5     //here import my data
6     TFile *f_data = new TFile("sample.root");
7     TTree *t_data = (TTree*)f_data->Get("GammaOmegaPhi");
8     ///////////////////////////////////////////////////
9     RooRealVar momega("momega","M_{#pi^{+}#pi^{-}#pi^{0}} (GeV/c^{2})",0.582,0.982);
10    RooDataSet data("data","data",t_data,momega);
11    ///////////////////////////////////////////////////bw conv gaus/////////////////////////////////
12    RooRealVar mean0("mean0","mean0",0,-0.1,0.1);
13    RooRealVar sigma0("sigma0","sigma0",0.1,0,0.1);
14    RooGaussian gauss0("gauss0","gauss0",momega,mean0,sigma0);
15    RooRealVar m0("m0","m0",0.782,0.582,0.982);
16    m0.setConstant();
17    RooRealVar g0("g0","g0",0.00868,0.0,0.020);
18    g0.setConstant();
19    RooBreitWigner bw0("bw0","bw0",momega,m0,g0);
20    RooFFTConvPdf conv1("conv1","conv1",momega,bw0,gauss0);
21    ///////////////////////////////////////////////////poly/////////////////////////////////
22    RooRealVar p0("p0","",1.3,-2,2);
23    RooRealVar p1("p1","",0.6,-2,2);
24    RooRealVar p2("p2","",0.2,-2,2);
25    RooChebychev bkgo0("bkgo0","bkgo0",momega,RooArgList(p0,p1,p2));
26    ///////////////////////////////////////////////////
27    RooRealVar nsig("nsig","nsig",19000,0,50000);
28    RooRealVar nbkg("nbkg","nbkg",13000,0,50000);
29
30    RooAddPdf summomega("sumomega","sumomega",RooArgList(conv1,bkgo0),RooArgList(nsig,nbkg));
31    summomega.fitTo(data);
32
33
34
35    RooPlot* momegaframe = momega.frame();
36    TCanvas *c1 = new TCanvas("c1","a canvas",0,0,800,600);
37    ///////////////////////////////////////////////////
38    c1->cd();
39    data.plotOn(momegaframe);
40    summomega.plotOn(momegaframe,LineColor(4));
41    summomega.plotOn(momegaframe,Components(conv1),LineStyle(4),LineColor(2),LineWidth(3));
42    summomega.plotOn(momegaframe,Components(bkgo0),LineStyle(2),LineColor(7),LineWidth(3));
43    momegaframe->Draw();
44 }
```

读取ROOT文件

定义gauss0, bw0

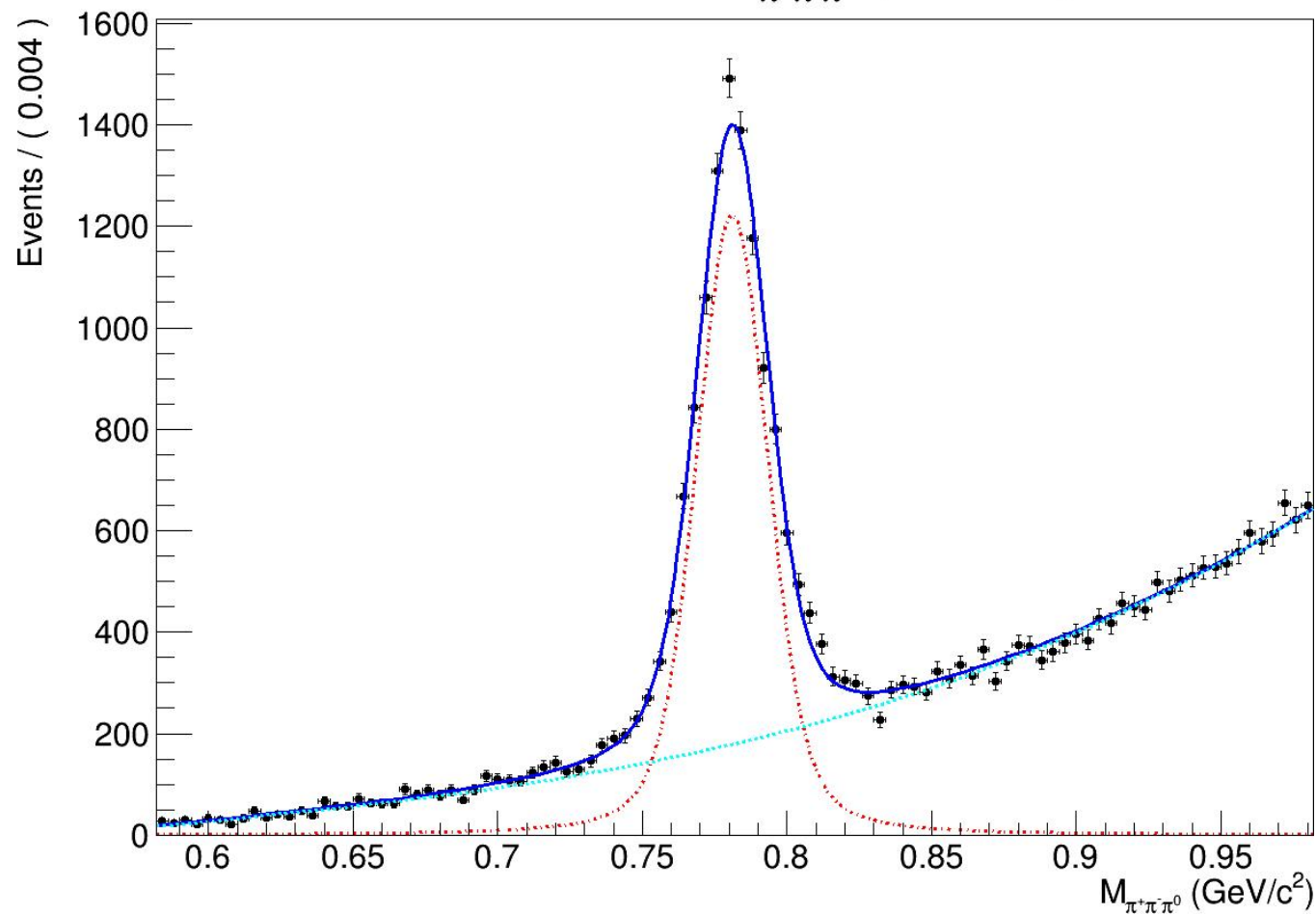
卷积gauss0、bw0为conv1
(描述信号)

定义切比雪夫多项式
(描述本底)

将 conv1 和 bkgo0 相加
RooArgList为信号和本底
的事例数，可以通过拟合
得到具体的事例数

画出拟合结果

A RooPlot of " $M_{\pi^+\pi^-\pi^0}$ (GeV/c²)"




```

*****
**      9 **HESSE          3500
*****
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=-360148 FROM HESSE      STATUS=OK          62 CALLS          314 TOTAL
                EDM=2.59207e-05  STRATEGY= 1      ERROR MATRIX ACCURATE

EXT PARAMETER          INTERNAL          INTERNAL
NO.   NAME      VALUE      ERROR      STEP SIZE      VALUE
1  mean0      -6.92184e-04  1.78943e-04  7.32605e-04  -6.92190e-03
2  nbkg       2.30344e+04  1.78863e+02  2.69154e-03  -7.87064e-02
3  nsig       1.05995e+04  1.39871e+02  2.43186e-03  -6.13852e-01
4  p0         1.17803e+00  7.98795e-03  1.47813e-03  6.29839e-01
5  p1         2.94004e-01  1.16601e-02  1.41953e-03  1.47537e-01
6  p2         4.54275e-02  8.42510e-03  1.23438e-03  2.27157e-02
7  sigma0     1.03433e-02  2.07064e-04  2.55629e-03  -2.22566e+00
                ERR DEF= 0.5
EXTERNAL ERROR MATRIX.   NDIM= 25  NPAR= 7  ERR DEF=0.5
3.202e-08 -1.886e-03  1.888e-03  3.484e-08  1.246e-07  1.607e-07 -4.077e-09
-1.886e-03  3.199e+04 -8.953e+03 -3.401e-01 -6.805e-01 -3.206e-01  1.003e-02
1.888e-03 -8.953e+03  1.956e+04  3.401e-01  6.805e-01  3.206e-01 -1.004e-02
3.484e-08 -3.401e-01  3.401e-01  6.381e-05  6.390e-05  3.638e-05 -3.736e-07
1.246e-07 -6.805e-01  6.805e-01  6.390e-05  1.360e-04  6.861e-05 -7.696e-07
1.607e-07 -3.206e-01  3.206e-01  3.638e-05  6.861e-05  7.098e-05 -3.710e-07
-4.077e-09  1.003e-02 -1.004e-02 -3.736e-07 -7.696e-07 -3.710e-07  4.288e-08
PARAMETER CORRELATION COEFFICIENTS
NO.  GLOBAL      1      2      3      4      5      6      7
1  0.15548  1.000 -0.059  0.075  0.024  0.060  0.107 -0.110
2  0.42533 -0.059  1.000 -0.358 -0.238 -0.326 -0.213  0.271
3  0.51472  0.075 -0.358  1.000  0.304  0.417  0.272 -0.347
4  0.69237  0.024 -0.238  0.304  1.000  0.686  0.541 -0.226
5  0.81386  0.060 -0.326  0.417  0.686  1.000  0.698 -0.319
6  0.70743  0.107 -0.213  0.272  0.541  0.698  1.000 -0.213
7  0.42079 -0.110  0.271 -0.347 -0.226 -0.319 -0.213  1.000
    
```



- 在实际工作当中，有时候某些pdf是没法用公式去描述的，找不到或者不存在公式去描述(或者说我就是不想用公式)，而我们需要这些PDF，我们就会想办法在物理上产生相近的PDF。比如我们可以从信号MC，收集一些事例构成dataset，然后由dataset去产生这个PDF。这种方法可以由任意的 dataset 来生成对应的PDF。

复杂例子1.1



- vi omegasigmcfit1.cxx

```
1 using namespace RooFit;
2 void omegasigmcfit1()
3 {
4
5     //here import my data
6     TFile *f_data = new TFile("sample.root");
7     TTree *t_data = (TTree*)f_data->Get("GammaOmegaPhi");
8     ///////////////////////////////////////////////////
9     RooRealVar momega("momega", "M_{#pi^{+}#pi^{-}}#pi^{0}} (GeV/c^{2})", 0.582, 0.982);
10    RooDataSet data("data", "data", t_data, momega);
11    ///////////////////////////////////////////////////sigpdf////////////////////////////////////
12    TFile *f_mc = new TFile("myshape.root");
13    TTree *t_mc = (TTree*)f_mc->Get("GammaOmegaPhi");
14    RooDataSet signalmc("signalmc", "signalmc", t_mc, momega);
15    RooKeysPdf somegapdf("sigPdf", "sigpdf", momega, signalmc, RooKeysPdf::MirrorBoth, 1);
16
17
18
19    RooRealVar mean0("mean0", "mean0", 0, -0.1, 0.1);
20    RooRealVar sigma0("sigma0", "sigma0", 0.1, 0, 0.1);
21    RooGaussian gauss0("gauss0", "gauss0", momega, mean0, sigma0);
22    RooFFTConvPdf conv1("conv1", "conv1", momega, somegapdf, gauss0);
23    //////////////////////////////////////////////////poly////////////////////////////////////
24    RooRealVar p0("p0", "", 1.3, -2, 2);
25    RooRealVar p1("p1", "", 0.6, -2, 2);
26    RooRealVar p2("p2", "", 0.2, -2, 2);
27    RooChebychev bkgo0("bkgo0", "bkgo0", momega, RooArgList(p0, p1, p2));
28    ///////////////////////////////////////////////////
29    RooRealVar nsig("nsig", "nsig", 19000, 0, 50000);
30    RooRealVar nbkg("nbkg", "nbkg", 13000, 0, 50000);
31
32    RooAddPdf sumomega("sumomega", "sumomega", RooArgList(conv1, bkgo0), RooArgList(nsig, nbkg));
33    sumomega.fitTo(data);
34
35
36
37
38
39    RooPlot* momegaframe = momega.frame();
40
41
42    TCanvas *c1 = new TCanvas("c1", "a canvas", 0, 0, 1200, 800);
43
```

这里和上面的例子的区别是，把BW分布，换成了由我们自己的dataset生成的一个PDF

复杂例子1.2



- vi omegasigmcfits.cxx

```
1 using namespace RooFit;
2 void omegasigmcfits()
3 {
4
5     //here import my data
6     TFile *f_data = new TFile("sample.root");
7     TTree *t_data = (TTree*)f_data->Get("GammaOmegaPhi");
8     ///////////////////////////////////////////////////
9     RooRealVar momega("momega", "M_{#omega} (GeV/c^{2})", 0.582, 0.982);
10    RooDataSet data("data", "data", t_data, momega);
11    ///////////////////////////////////////////////////
12    TH1F *moss = new TH1F("moss", "", 200, 0.582, 0.982); //momega signalmc shape
13    TFile *f_mc = new TFile("myshape.root");
14    TTree *t_mc = (TTree*)f_mc->Get("GammaOmegaPhi");
15    double smomega;
16    t_mc -> SetBranchAddress("momega", &smomega);
17    int EntNum = t_mc -> GetEntries();
18    for(int i = 0; i < EntNum; i++)
19    {
20        t_mc -> GetEntry(i);
21        moss -> Fill(smomega);
22    }
23    RooDataHist somegahist("somegahist", "somegahist", momega, moss);
24    RooHistPdf somegapdf("somegapdf", "somegapdf", momega, somegahist, 4);
25
26
27
28    RooRealVar mean0("mean0", "mean0", 0, -0.1, 0.1);
29    RooRealVar sigma0("sigma0", "sigma0", 0.1, 0.0, 0.1);
30    RooGaussian gauss0("gauss0", "gauss0", momega, mean0, sigma0);
31    RooFFTConvPdf conv1("conv1", "conv1", momega, somegapdf, gauss0);
32    ///////////////////////////////////////////////////
33    RooRealVar p0("p0", "", 1.3, -2, 2);
34    RooRealVar p1("p1", "", 0.6, -2, 2);
35    RooRealVar p2("p2", "", 0.2, -2, 2);
36    RooChebychev bkgo0("bkgo0", "bkgo0", momega, RooArgList(p0, p1, p2));
37    ///////////////////////////////////////////////////
38    RooRealVar nsig("nsig", "nsig", 19000, 0, 50000);
39    RooRealVar nbkg("nbkg", "nbkg", 13000, 0, 50000);
40
41    RooAddPdf summomega("sumomega", "sumomega", RooArgList(conv1, bkgo0), RooArgList(nsig, nbkg));
42    summomega.fitTo(data);
43
```

这是另一种生成PDF的方法，即先把数据填到一个hist里面，再由这个hist来生成一个PDF。该方法较上一页的方法相比稍微粗糙一些，但是快，在生成PDF的数据量很大的时候适用。

- 拟合还有很多可以的操作，比如
 - ✓ 需要拟合的数据量巨大的时候，可以把数据分成bin拟合，速度会大大加快
 - ✓ 如果需要同时进行多个量拟合的时候，会有二维单位的拟合，即那种联合概率密度函数。
 - ✓ **dataset**很难拟合上的话，可以进行多次迭代拟合(拟合本身就是一种迭代！)，以得到更有的结果。
 - ✓ 还可以自己编写PDF函数，写成头文件或者写成包之类的，后面可以自己用。
 - ✓ **blablabla**

- 需要进行资料查阅的话，可以到ROOT的官网里面的reference里面去搜索，也可以直接在bing搜索某个函数。有问题解决不了的话，不要自己一直纠结，狠狠抓师兄们！(root作为工具我们的要求是熟练地用，而不需要自己去浪费太多时间)
- ROOT自带的tutorial就很好用，里面有很多东西,大家可以把整个tutorial目录cp到自己的目录下，随便运行进行尝试(里面有roofit的部分，也有各种别的部分):
 - ✓ `cp $ROOTSYS/tutorials ./`