> "下载文章"

Lesson 0 install root and software on your compute

> "If you get the THU's computer cluster account, use following commands to get into.
>
> `$ ssh -XY -p 48571 -o ServerAliveInterval=5  yourname@hepthu.com`
>
> You can also run code on your pc if you have installed root, it's up to you. "

homebrew Click this link to go to the Homebrew page, follow the instructions to install Homebrew on your Mac. Homebrew is a software package manager that allows you to download many applications on your Mac!
After finishing the installation, paste the following command in a macOS terminal to install CERN ROOT software: `$ brew install root` . This way, you can also install applications like `firefox` which you can't find in the App Store.
Additionally, `$` indicates a shell command, so drop the `$` sign when you copy the command.

Lesson 1 base commands for Linux

Some simple commands you can have a try!

```
pwd  #print work directory
cd /path/to/directory #change directory
cd ..  #back to parent directory
ls    #list directory contents
mkdir test #make a directory
touch test.txt #creat a new txt file
cp test.txt copy.txt #copy and rename a file
mv copy.txt copy1.txt #move and rename a file
rm copy.txt #remove
rm -r test  #delete a directory
vi filename    #open or make a file with vim editor
```

Vim editor

```
i     #insert, edit mode
Esc    #exit edit mode
:wq    #save and exit
:q    #exit, will mention you if you have modified the file
:q!    #exit !without! save
```

you will use the next commands from time to time
vi somefile -> press i into edit mode -> do something ->esc exit edit mode -> :wq

Lesson 2 root 基础练习

进入root环境

```
$ root
```

```
% root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.32.06                              https://root.cern |
  | (c) 1995-2024, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for macosxarm64 on Sep 21 2024, 18:21:53                   |
  | From tags/6-32-06@6-32-06                                        |
  | With Apple clang version 15.0.0 (clang-1500.1.0.2.5)             |
  | Try '.help'/'.?', '.demo', '.license', '.credits', '.quit'/'.q'  |
   ------------------------------------------------------------------

root [0]
```

```
root [0] .q 退出root
```

root能输入一句解释一句

```
root [0] 1+1
(int) 2
```

也可以边解释边执行或者编译后执行一个程序文件，用 `root yourprogram.C`

例如：可以在root中输入下面的程序用来创建一个空白的直方图

```
TH1F* graph=new TH1F("name","title",200,2.97,3.03);
graph->Draw();
```

但一个空白的直方图是很无趣的，你还需要用 `Fill` 来填入数据

```
% root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.32.06                              https://root.cern |
  | (c) 1995-2024, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for macosxarm64 on Sep 21 2024, 18:21:53                   |
  | From tags/6-32-06@6-32-06                                        |
  | With Apple clang version 15.0.0 (clang-1500.1.0.2.5)             |
  | Try '.help'/'.?', '.demo', '.license', '.credits', '.quit'/'.q'  |
   ------------------------------------------------------------------

root [0] TH1F* graph=new TH1F("name","title",200,2.97,3.03);
root [1] graph->Draw();
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [2] graph->Fill(2.99);
root [3] graph->Draw();
root [4]
```

此外我们还能直接创建一个程序文件，以放入更长的程序

Get start!

```
$ vi test.C
```

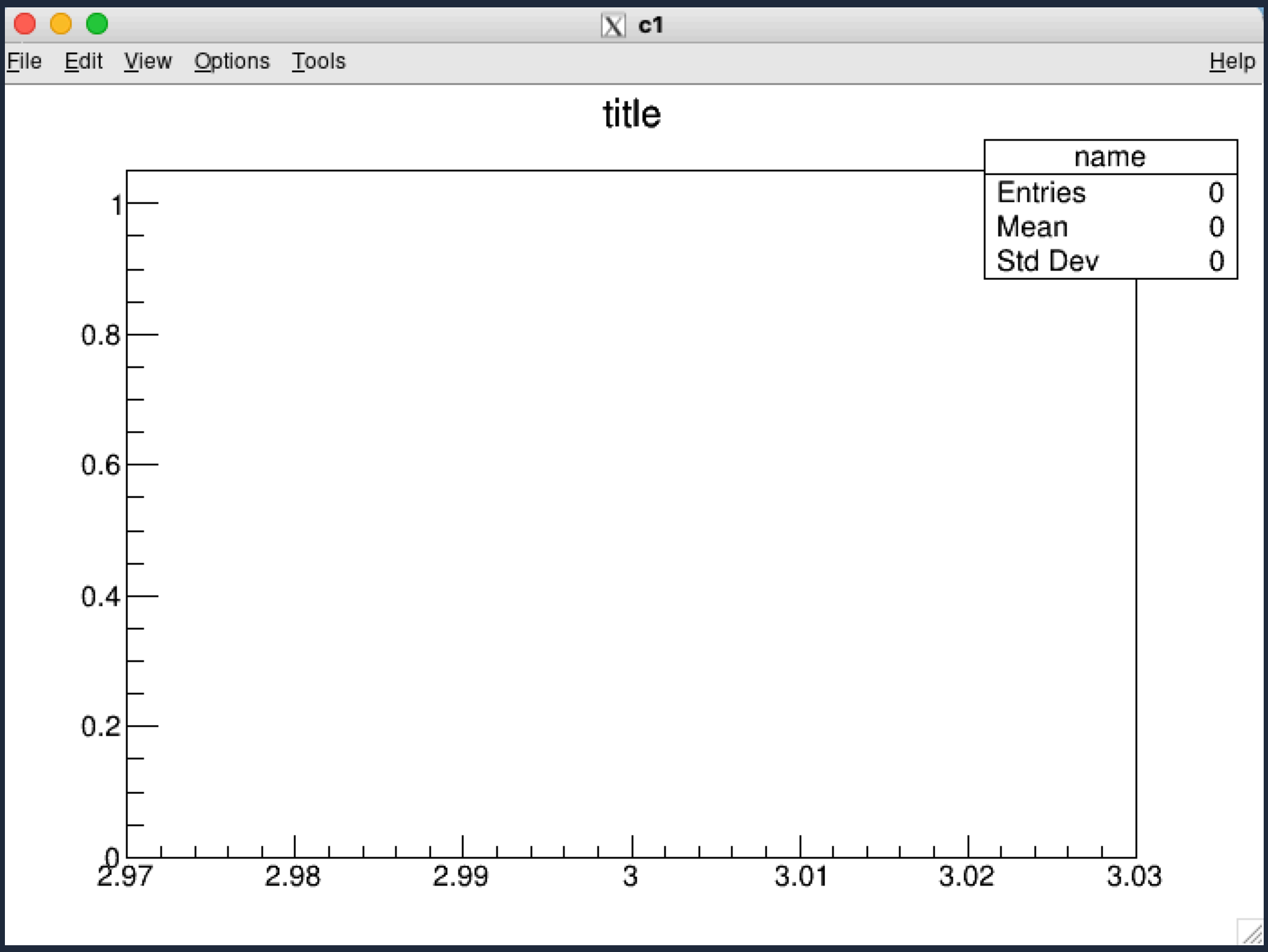
//push `i` to enter the insert mode//

```
void test() //be sure it is similar with your file name

 {

  TH1F* graph=new TH1F("name","title",200,2.97,3.03);
  //TH1F* graph name=new TH1F("name","title”,bins, low, high)

  graph->Draw();  //draw your graph

  }
```

//push ESC to end the insert mode//

:wq //save and exit

```
$ root test.C  //run your code
```

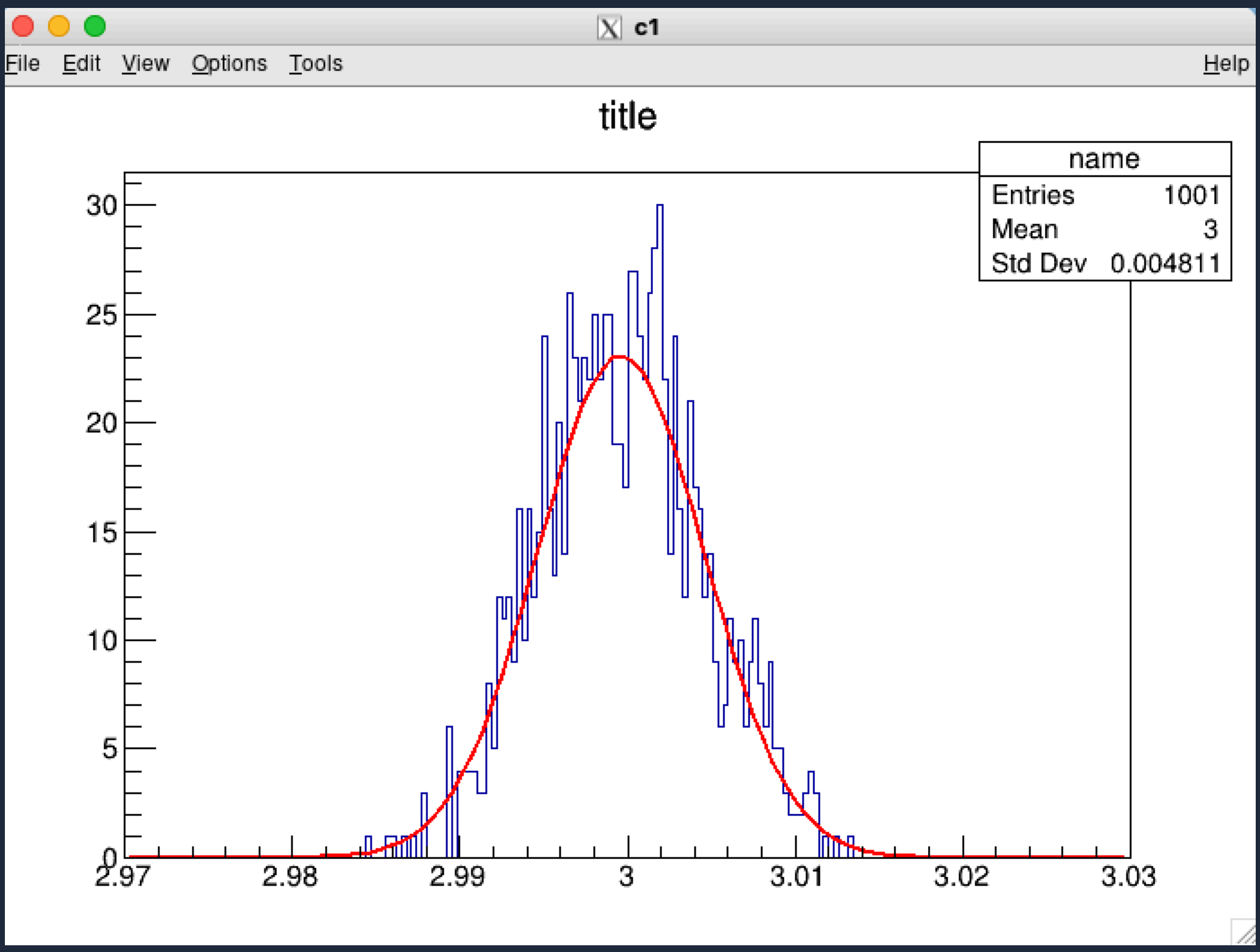


///successful!///

.q //exit the program

Let’s add something into your graph!

```
void test(){

 //TCanvas* c1=new TCanvas("c1","fitting with Gaussian function");  //if you don't set the canvas's name, it will created default canvas with name c1

 //c1->SetGrid();  // set grid

 TH1F* graph=new TH1F("name","title",200,2.97,3.03);

 graph->Fill(2.99); // you can add just one data in your graph using "Fill" command and try to draw your graph.It's boring, right? We need more!

 TRandom n; // define a random variable n

 for (int i=0;i<1000;++i){

 graph->Fill(n.Gaus(3,0.005));  //Determine variable n using Gaussian distribution and fill to graph

 }

 graph->Draw();

 graph->Fit("gaus");  //fit your graph with Gaussian function

 }
```



Wow! now you can use root to do some simple fit works. but we usually use RooFit to do more complex job! So, ready for more programs!

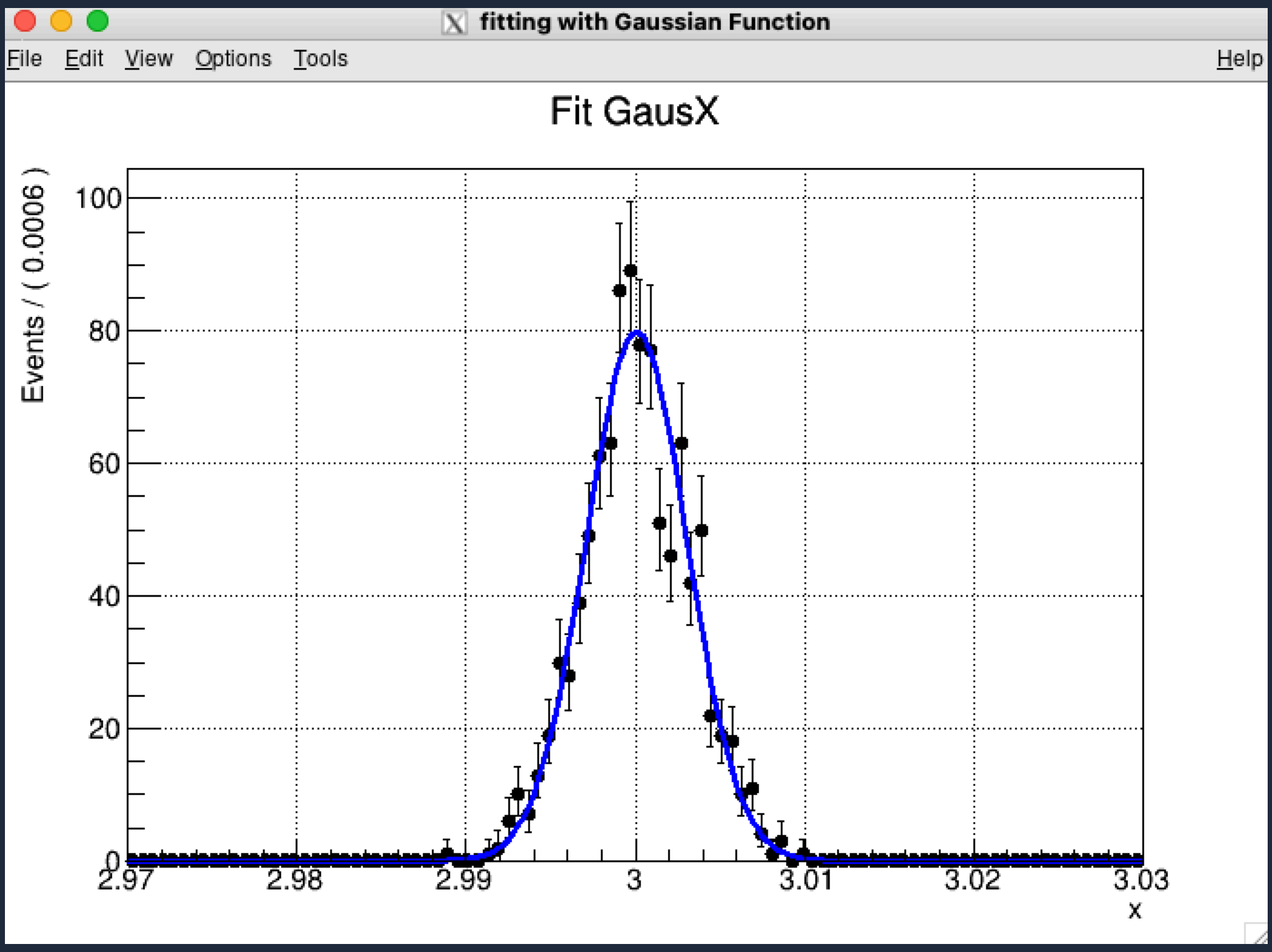
以上只是一些基础的拟合，很多参数我们并不能自己去定义，没有灵活性， `RooFit` 中则提供了更加专业的拟合函数。

Lesson 2 Gaussian fitting using RooFit

```
#include "RooAbsReal.h"
#include "RooRealVar.h"
#include "RooGaussian.h"
#include "RooChebychev.h"
#include "RooAddPdf.h"
//#include "RooProdPdf.h"
#include "RooDataSet.h"
//#include "RooDataHist.h"
//#include "RooFitResult.h"
#include "RooPlot.h"
//#include "RooArgList.h"
//#include "RooArgSet.h"
//#include "RooRandom.h"
//#include "RooPrintable.h"
using namespace RooFit;
```

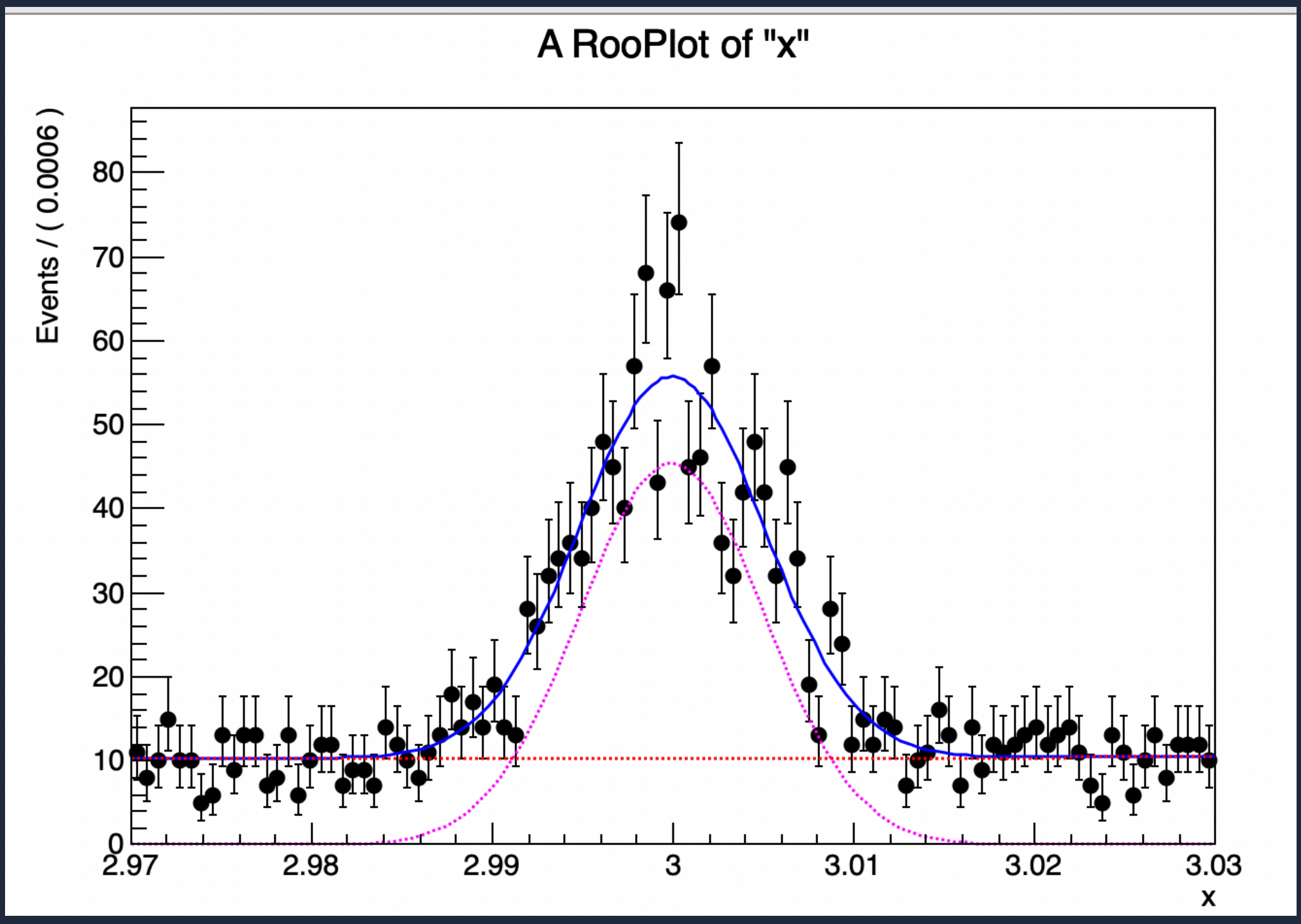
```
16 void test() {
17     RooRealVar x("x", "x", 2.97, 3.03);
18     // Declares a real-valued variable 'x' with a range from 2.97 to 3.03.
19 
20     RooRealVar mean("mean", "mean", 3.0, 2.8, 3.2);
21     // Declares a real-valued variable 'mean' with an initial value of 3.0 and a range from
   2.8 to 3.2.
22 
23     RooRealVar sigma("sigma", "sigma", 0.003, 0.002, 0.003);
24     // Declares a real-valued variable 'sigma' with an initial value of 0.003 and a range
   from 0.002 to 0.003.
25 
26     RooAbsPdf* gaus = new RooGaussian("gaus", "gaus", x, mean, sigma);
27     // Creates a Gaussian probability density function (PDF) named 'gaus' with 'x' as the
   observable, 'mean' as the mean, and 'sigma' as the standard deviation.
28 
29     RooRealVar n("n", "n", 0, 0, 50000);
30     // Declares a real-valued variable 'n' with an initial value of 0 and a range from 0 to
   50000.
31 
32     RooExtendPdf* exp = new RooExtendPdf("exp", "exp", *gaus, n);
33     // Creates an extended PDF named 'exp' that combines the Gaussian PDF 'gaus' with the
   variable 'n'.
34 
35     RooAddPdf total("total", "total", RooArgList(*exp), RooArgList(n));
36     // Creates a composite PDF named 'total' that consists of the extended PDF 'exp' and
   the variable 'n'.
37 
38     RooDataSet* data;
39     // Declares a pointer to a RooDataSet object named 'data'.
40 
41     data = gaus->generate(RooArgSet(x), 1000);
42     // Generates a dataset 'data' with 1000 events based on the Gaussian PDF 'gaus' and the
   observable 'x'.
43 
44     RooFitResult* result = total.fitTo(*data, Save());
45     // Fits the composite PDF 'total' to the dataset 'data' and saves the fit result in
   'result'.
46 
47     TCanvas* c1 = new TCanvas("c1", "fitting with Gaussian function");
48     // Creates a new canvas named 'c1' with the title "fitting with Gaussian function" for
   plotting.
49 
50     RooPlot* xframe = x.frame(RooFit::Title("Fit GausX"));
51     // Creates a frame for the observable 'x' with the title "Fit GausX".
52 
53     data->plotOn(xframe);
54     // Plots the dataset 'data' on the frame 'xframe'.
55 
```

```
56      total.plotOn(xframe);
57      // Plots the composite PDF 'total' on the frame 'xframe'.
58
59      xframe->Draw();
60      // Draws the frame 'xframe' on the canvas.
61  }
```




```
1   void test(){
2   using namespace RooFit;
3   //build gaussian and shev pdf
4   RooRealVar x("x","x",2.97,3.03);
5   RooRealVar mean("mean","mean",3.0,2.8,3.2);
6   RooRealVar sigma("sigma","sigma",0.005,0.003,0.007);
7   RooRealVar a0("a0","a0",0.005,0.004,0.006);
8   RooAbsPdf* gaus=new RooGaussian("gaus","gaus",x,mean,sigma);
9   RooAbsPdf* shev=new RooChebychev("shev","shev",x,a0);
10  RooRealVar n1("n1","n1",0,0,50000);
11  RooRealVar n2("n2","n2",0,0,50000);
12  RooExtendPdf*signal1=new RooExtendPdf("sig1","sig1",*gaus,n1);
13  RooExtendPdf*signal2=new RooExtendPdf("sig2","sig2",*shev,n2);
```

```
14 RooAddPdf totalPdf("total","total",RooArgList(*signal1,*signal2),RooArgList(n1,n2));//totalPdf(x)=signal1(x)+signal2(x)
15
16 //generate data point
17 RooDataSet*data1;
18 RooDataSet*data2;
19 data1=gaus->generate(RooArgSet(x),1000);
20 data2=shev->generate(RooArgSet(x),1000);
21 data1->append(*data2);
22
23 //fit
24 RooFitResult*result= totalPdf.fitTo(*data1,Save());
25 RooPlot*xframe=x.frame();
26 data1->plotOn(xframe);
27 totalPdf.plotOn(xframe);
28 totalPdf.plotOn(xframe, Components(*signal1),LineStyle(kDashed),LineColor(6));
29 totalPdf.plotOn(xframe, Components("sig2"),LineStyle(kDashed),LineColor(2));//two different ways set the Components name
30 xframe->Draw();
31 }
```



> *"以上是一维拟合，但很多时候我们会研究两个不同组分，这样就需要进行二维拟合。二维拟合就不单单是把两个PDF加和在一起了，而是需要乘积，可以理解为同时出现两个组分的概率。"*

Lesson 3 2D fitting

下面展示了两个二维拟合

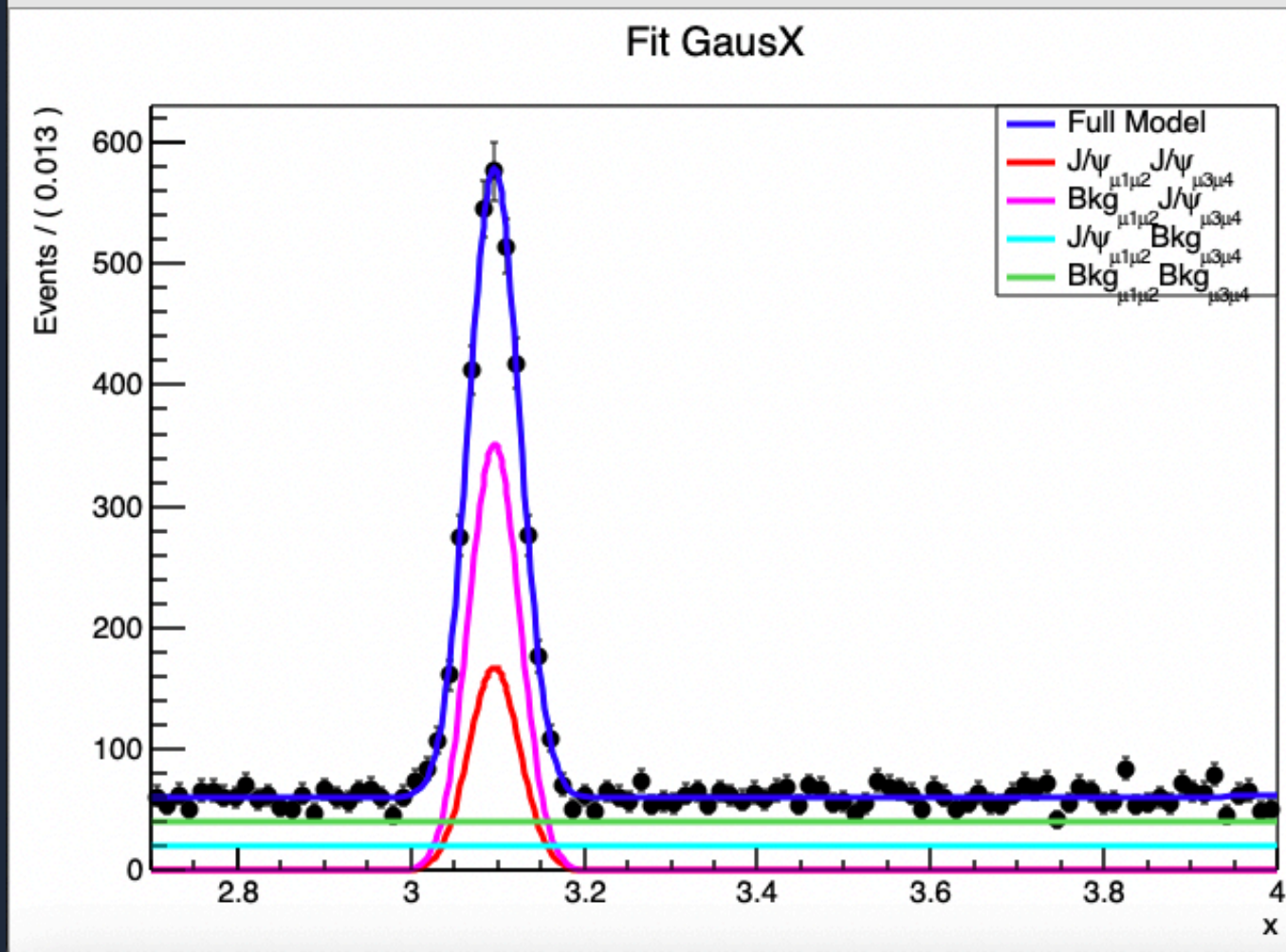
```
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <iostream>
#include <fstream>
#include "TF1.h"
#include "TText.h"
#include "TLorentzVector.h"
#include "TVector3.h"
#include "TLorentzRotation.h"
#include "RooNumIntConfig.h" //no matter what, copy directly!

using namespace RooFit;
void test(){
//build gaussian pdf in two dimension
RooRealVar x("x","x",2.7,4);

RooRealVar meanx("meanx","meanx",3.0967,2.9,3.2);
RooRealVar sigmax("sigmax","sigmax",0.03,0.01,0.06);
RooAbsPdf* gausx=new RooGaussian("gausx","gausx",x,meanx,sigmax);

RooRealVar ax0("ax0","ax0",0.005,0.004,0.006);
RooRealVar ax1("ax1","ax1",0.01,0.01,0.02);
RooAbsPdf* shevx=new RooChebychev("shevx","shevx",x,RooArgSet(ax0,ax1));

RooRealVar y("y","y",2.7,4);

RooRealVar meany("meany","meany",3.0967,2.9,3.2);
RooRealVar sigmay("sigmay","sigmay",0.03,0.01,0.06);
RooAbsPdf* gausy=new RooGaussian("gausy","gausy",y,meany,sigmay);

RooRealVar ay0("ay0","ay0",0.005,0.004,0.006);
RooRealVar ay1("ay1","ay1",0.01,0.01,0.02);
RooAbsPdf* shevy=new RooChebychev("shevy","shevy",y,RooArgSet(ay0,ay1));

//constructor with 2 pdf
RooProdPdf* sigxsigy=new RooProdPdf("sigxsigy","sigxsigy",*gausx,*gausy);
RooProdPdf* bkgxsigy=new RooProdPdf("bkgxsigy","bkgxsigy",*shevx,*gausy);
RooProdPdf* sigxbkgy=new RooProdPdf("sigxbkgy","sigxbkgy",*gausx,*shevy);
RooProdPdf* bkgxbkgy=new RooProdPdf("bkgxbkgy","bkgxbkgy",*shevx,*shevy);
```

```
41
42  RooRealVar n_sigxsigy("n_sigxsigy","n_sigxsigy",0,0,50000);
43  RooRealVar n_bkgxsigy("n_bkgxsigy","n_bkgxsigy",0,0,50000);
44  RooRealVar n_sigxbkgy("n_sigxbkgy","n_sigxbkgy",0,0,50000);
45  RooRealVar n_bkgxbkgy("n_bkgxbkgy","n_bkgxbkgy",0,0,50000);
46
47
48  RooExtendPdf*signal1=new RooExtendPdf("signal1","signal1",*sigxsigy,n_sigxsigy);
49  RooExtendPdf*signal2=new RooExtendPdf("signal2","signal2",*sigxbkgy,n_sigxbkgy);
50  RooExtendPdf*signal3=new RooExtendPdf("signal3","signal3",*bkgxsigy,n_bkgxsigy);
51  RooExtendPdf*signal4=new RooExtendPdf("signal4","signal4",*bkgxbkgy,n_bkgxbkgy);
52
53  RooAddPdf
    totalPdf("total","total",RooArgList(*signal1,*signal2,*signal3,*signal4),RooArgList(n_sigx
    sigy,n_sigxbkgy,n_bkgxsigy,n_bkgxbkgy));
54
55  //generate data point
56
57  RooDataSet*data1;
58  RooDataSet*data2;
59  RooDataSet*data3;
60  RooDataSet*data4;
61  data1=sigxsigy->generate(RooArgSet(x,y),1000);
62  data2=sigxbkgy->generate(RooArgSet(x,y),2000);
63  data3=bkgxsigy->generate(RooArgSet(x,y),2000);
64  data4=bkgxbkgy->generate(RooArgSet(x,y),4000);
65
66  data1->append(*data2);
67  data1->append(*data3);
68  data1->append(*data4);
69
70  //fit
71  RooFitResult*result= totalPdf.fitTo(*data1,Save());
72  //x dimension
73
74  //create two canvases, because you have to know two dimensions fitting.
75  TCanvas*c1 =new TCanvas("c1","Fitting X");
76  c1->cd();
77  RooPlot*xframe=x.frame(RooFit::Title("Fit GausX"));
78  data1->plotOn(xframe);
79  totalPdf.plotOn(xframe,Name("fullModel"));
80  totalPdf.plotOn(xframe,Components(*signal1),LineColor(2),LineStyle(1),Name("JpsiJpsi"));
81  totalPdf.plotOn(xframe,Components(*signal2),LineColor(6),LineStyle(1),Name("BkgJpsi"));
82  totalPdf.plotOn(xframe,Components(*signal3),LineColor(7),LineStyle(1),Name("JpsiBkg"));
83  totalPdf.plotOn(xframe,Components(*signal4),LineColor(8),LineStyle(1),Name("BkgBkg"));
84
85  TLegend leg(0.7, 0.7, 0.9, 0.9);
86  leg.AddEntry(xframe->findObject("fullModel"), "Full Model", "L");
87  leg.AddEntry(xframe->findObject("JpsiJpsi"), "J/#psi_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
```

```
leg.AddEntry(xframe->findObject("BkgJpsi"), "Bkg_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
leg.AddEntry(xframe->findObject("JpsiBkg"), "J/#psi_{#mu1#mu2}Bkg_{#mu3#mu4}", "L");
leg.AddEntry(xframe->findObject("BkgBkg"), "Bkg_{#mu1#mu2}Bkg_{#mu3#mu4}", "L");

xframe->Draw();
leg.DrawClone();

//y dimension
TCanvas*c2 =new TCanvas("c2","Fitting Y");
c2->cd();
RooPlot*yframe=y.frame(RooFit::Title("Fit GausY"));
data1->plotOn(yframe);
totalPdf.plotOn(yframe);
totalPdf.plotOn(yframe,Name("fullModel"));
totalPdf.plotOn(yframe,Components(*signal1),LineColor(2),LineStyle(1),Name("JpsiJpsi"));
totalPdf.plotOn(yframe,Components(*signal2),LineColor(6),LineStyle(1),Name("BkgJpsi"));
totalPdf.plotOn(yframe,Components(*signal3),LineColor(7),LineStyle(1),Name("JpsiBkg"));
totalPdf.plotOn(yframe,Components(*signal4),LineColor(8),LineStyle(1),Name("BkgBkg"));

TLegend leg2(0.7, 0.7, 0.9, 0.9);
leg2.AddEntry(yframe->findObject("fullModel"), "Full Model", "L");
leg2.AddEntry(yframe->findObject("JpsiJpsi"), "J/#psi_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
leg2.AddEntry(yframe->findObject("BkgJpsi"), "Bkg_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
leg2.AddEntry(yframe->findObject("JpsiBkg"), "J/#psi_{#mu1#mu2}Bkg_{#mu3#mu4}", "L");
leg2.AddEntry(yframe->findObject("BkgBkg"), "Bkg_{#mu1#mu2}Bkg_{#mu3#mu4}", "L");

yframe->Draw();
leg2.DrawClone();

TCanvas*c3 =new TCanvas("c3","2D plot");
c3->cd();
TH2F* mytwojpsimass = new TH2F("mytwojpsimass","mytwojpsimass",40,2.7,3.5,40,2.7,3.5);
data1->fillHistogram(mytwojpsimass,RooArgList(x,y));
mytwojpsimass->Draw("colz");

}
```

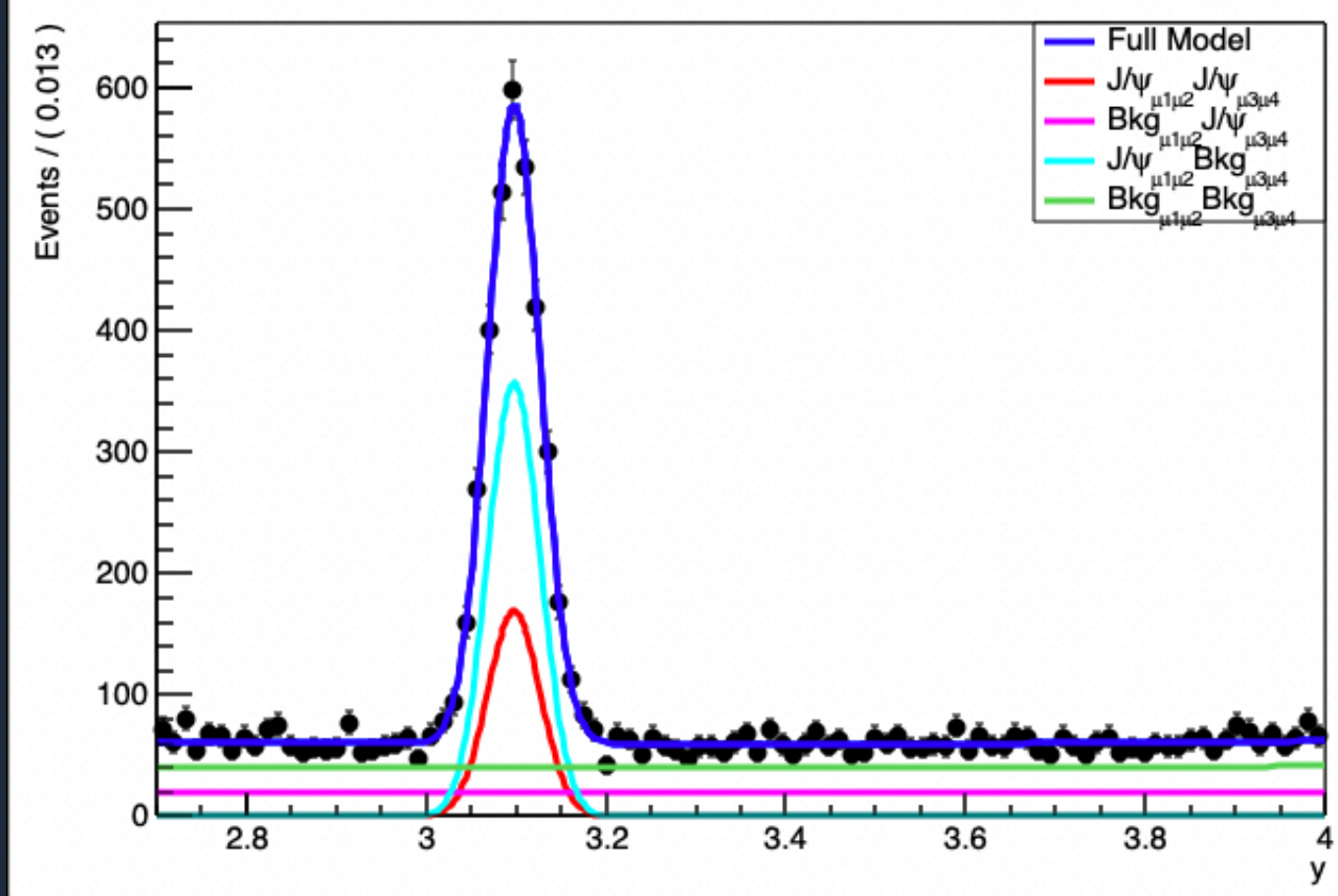
Fitting X

File Edit View Options Tools Help



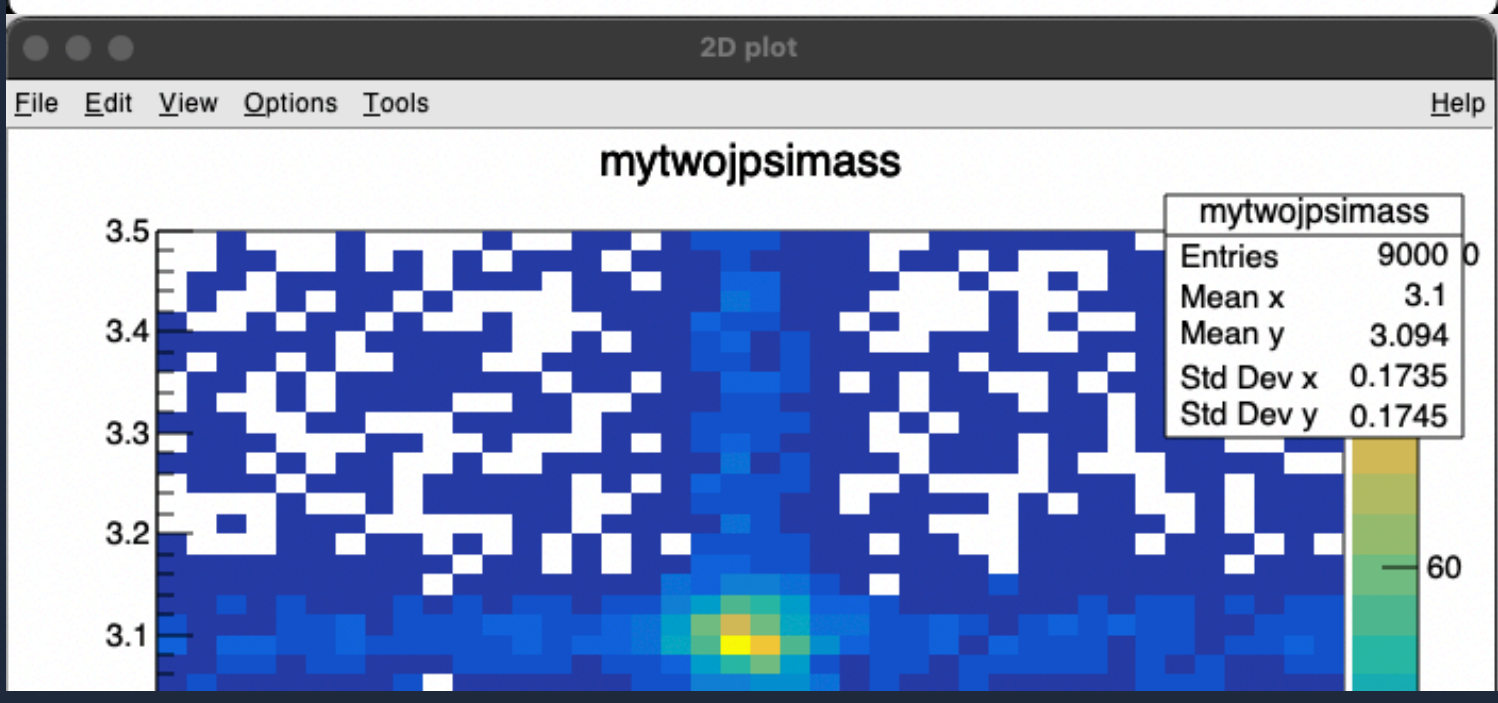
Fitting Y

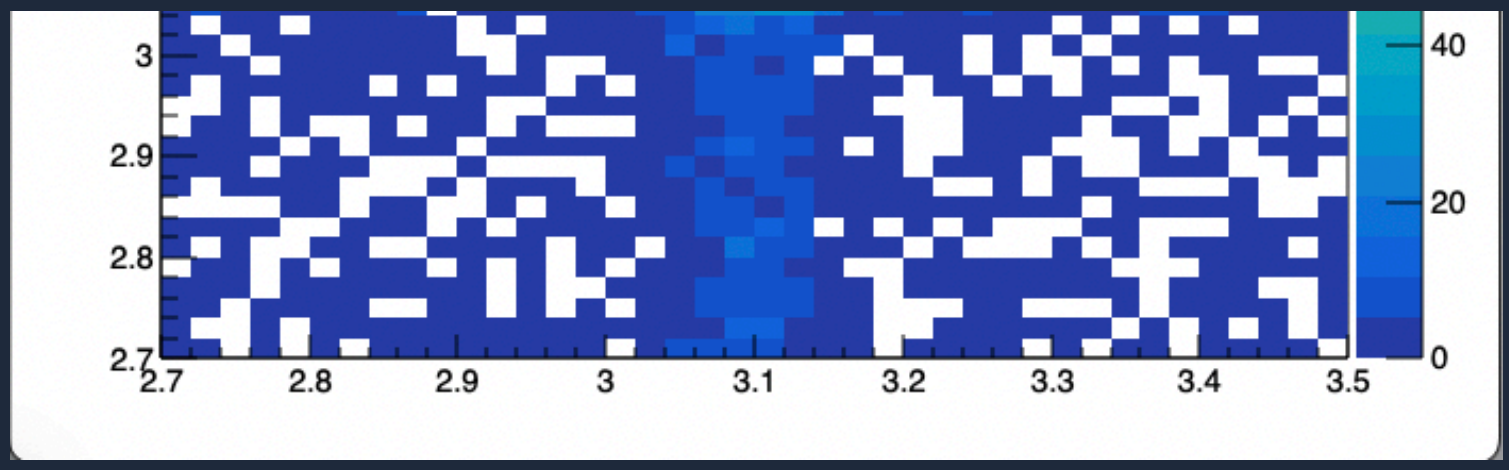
File Edit View Options Tools Help



2D plot

File Edit View Options Tools Help






```
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <iostream>
#include <fstream>
#include "TF1.h"
#include "TText.h"
#include "TLorentzVector.h"
#include "TVector3.h"
#include "TLorentzRotation.h"
#include "RooNumIntConfig.h" //no matter what, copy directly!
using namespace RooFit;
void test(){
//build gaussian pdf in two dimension
RooRealVar x("x","x",2.7,4);
RooRealVar meanx1("meanx1","meanx1",3.0967,2.9,3.2);
RooRealVar sigmax1("sigmax1","sigmax1",0.03,0.01,0.06);
RooAbsPdf* gausx1=new RooGaussian("gausx1","gausx1",x,meanx1,sigmax1);

RooRealVar meanx2("meanx2","meanx2",3.686,3.5,3.8);
RooRealVar sigmax2("sigmax2","sigmax2",0.1,0.01,0.2);
RooAbsPdf* gausx2=new RooGaussian("gausx2","gausx2",x,meanx2,sigmax2);

RooRealVar y("y","y",2.7,4);

RooRealVar meany1("meany1","meany1",3.0967,2.9,3.2);
RooRealVar sigmay1("sigmay1","sigmay1",0.03,0.01,0.06);
RooAbsPdf* gausy1=new RooGaussian("gausy1","gausy1",y,meany1,sigmay1);

RooRealVar meany2("meany2","meany2",3.686,3.5,3.8);
RooRealVar sigmay2("sigmay2","sigmay2",0.1,0.01,0.2);
RooAbsPdf* gausy2=new RooGaussian("gausy2","gausy2",y,meany2,sigmay2);

//constructor with 2 pdf
RooProdPdf* gausx1y1=new RooProdPdf("gausx1y1","gausx1y1",*gausx1,*gausy1);
RooProdPdf* gausx1y2=new RooProdPdf("gausx1y2","gausx1y2",*gausx1,*gausy2);
RooProdPdf* gausx2y1=new RooProdPdf("gausx2y1","gausx2y1",*gausx2,*gausy1);

RooRealVar n_x1y1("n1","n1",0,0,50000);
RooRealVar n_x1y2("n2","n2",0,0,50000);
```

```
41 RooRealVar n_x2y1("n3","n3",0,0,50000);
42
43 RooAddPdf
   totalPdf("total","total",RooArgList(*gausx1y1,*gausx1y2,*gausx2y1),RooArgList(n_x1y1,n_x1y
   2,n_x2y1));
44
45 //generate data point
46
47 RooDataSet*data1;
48 RooDataSet*data2;
49 RooDataSet*data3;
50 data1=gausx1y1->generate(RooArgSet(x,y),1000);
51 data2=gausx1y2->generate(RooArgSet(x,y),1000);
52 data3=gausx2y1->generate(RooArgSet(x,y),1000);
53
54 data1->append(*data2);
55 data1->append(*data3);
56
57 //fit
58 RooFitResult*result= totalPdf.fitTo(*data1,Save());
59 //x dimension
60 //create two canvases, because you have to know two dimensions fitting.
61 TCanvas*c1 =new TCanvas("c1","Fitting X");
62 c1->cd();
63 RooPlot*xframe=x.frame(RooFit::Title("Fit GausX"));
64 data1->plotOn(xframe,Name("data1"));
65 totalPdf.plotOn(xframe,Name("fullModel"));
66 totalPdf.plotOn(xframe,Components(*gausx1y1),LineColor(2),LineStyle(1),Name("JpsiJpsi"));
67 totalPdf.plotOn(xframe,Components(*gausx1y2),LineColor(6),LineStyle(1),Name("JpsiPsi2S"));
68 totalPdf.plotOn(xframe,Components(*gausx2y1),LineColor(7),LineStyle(1),Name("Psi2SJpsi"));
69
70 TLegend* leg= new TLegend(0.7, 0.7, 0.9, 0.9);
71 leg->AddEntry(xframe->findObject("data1"), "Data", "pe");
72 leg->AddEntry(xframe->findObject("fullModel"), "Total Fit", "L");
73 leg->AddEntry(xframe->findObject("JpsiJpsi"), "J/#psi_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
74 leg->AddEntry(xframe->findObject("JpsiPsi2S"), "J/#psi_{#mu1#mu2}#psi(2S)_{#mu3#mu4}",
   "L");
75 leg->AddEntry(xframe->findObject("Psi2SJpsi"), "#psi(2S)_{#mu1#mu2}J/#psi_{#mu3#mu4}",
   "L");
76
77 xframe->Draw();
78 leg->Draw();
79
80
81
82 //y dimension
83 TCanvas*c2 =new TCanvas("c2","Fitting Y");
84 c2->cd();
85 RooPlot*yframe=y.frame(RooFit::Title("Fit GausY"));
```

```
86  data1->plotOn(yframe,Name("data1"));
87  totalPdf.plotOn(yframe,Name("fullModel"));
88  totalPdf.plotOn(yframe,Components(*gausx1y1),LineColor(2),LineStyle(1),Name("JpsiJpsi"));
89  totalPdf.plotOn(yframe,Components(*gausx1y2),LineColor(6),LineStyle(1),Name("JpsiPsi2S"));
90  totalPdf.plotOn(yframe,Components(*gausx2y1),LineColor(7),LineStyle(1),Name("Psi2SJpsi"));
91
92
93  TLegend leg2(0.7, 0.7, 0.9, 0.9);
94  leg2.AddEntry(yframe->findObject("data1"), "Data", "pe");
95  leg2.AddEntry(yframe->findObject("fullModel"), "Total Fit", "L");
96  leg2.AddEntry(yframe->findObject("JpsiJpsi"), "J/#psi_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
97  leg2.AddEntry(yframe->findObject("JpsiPsi2S"), "J/#psi_{#mu1#mu2}#psi(2S)_{#mu3#mu4}",
    "L");
98  leg2.AddEntry(yframe->findObject("Psi2SJpsi"), "#psi(2S)_{#mu1#mu2}J/#psi_{#mu3#mu4}",
    "L");
99
100 yframe->Draw();
101 leg2.DrawClone();
102
103 TCanvas*c3 =new TCanvas("c3","2D plot");
104 c3->cd();
105 TH2F* mytwojpsimass = new TH2F("mytwojpsimass","mytwojpsimass",40,2.7,4,40,2.7,4);
106 data1->fillHistogram(mytwojpsimass,RooArgList(x,y));
107 mytwojpsimass->Draw("colz");
```