> "下载文章"

Lesson 0 install root and software on your computer

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

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.

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  #make a directory
rm file #remove
rm -r directory  #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

---

Get start!

```
$ vi test.C
```

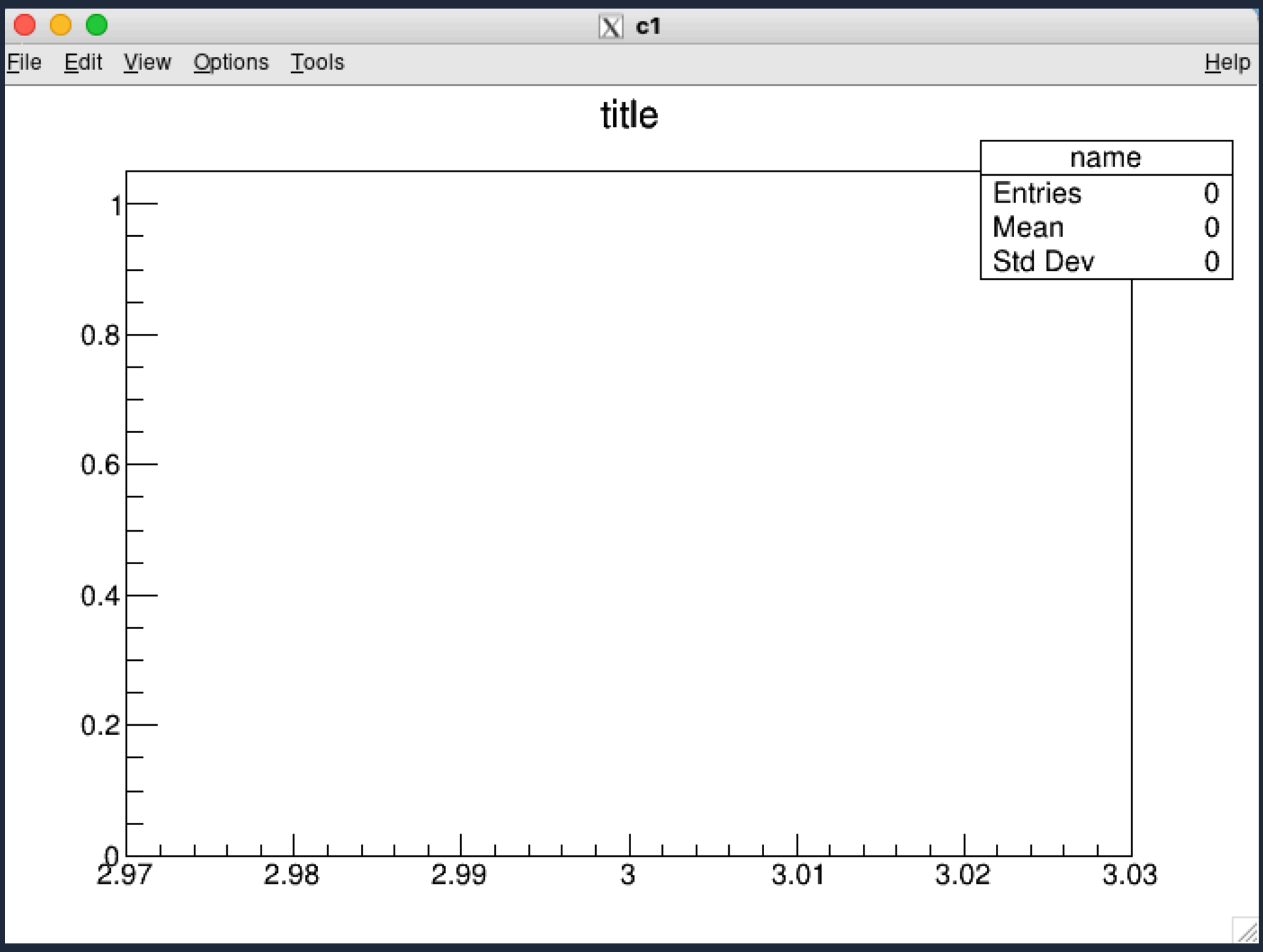
//push `i` to enter the insert mode//

```
1  void test() //be sure it is similar with your file name
2
3   {
4
5    TH1F* graph=new TH1F("name","title",200,2.97,3.03);
6    //TH1F* graph name=new TH1F("name","title”,bins, low, high)
7
8    graph->Draw();  //draw your graph
9
10   }
```

//push ESC to end the insert mode//

:wq //save and exit

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

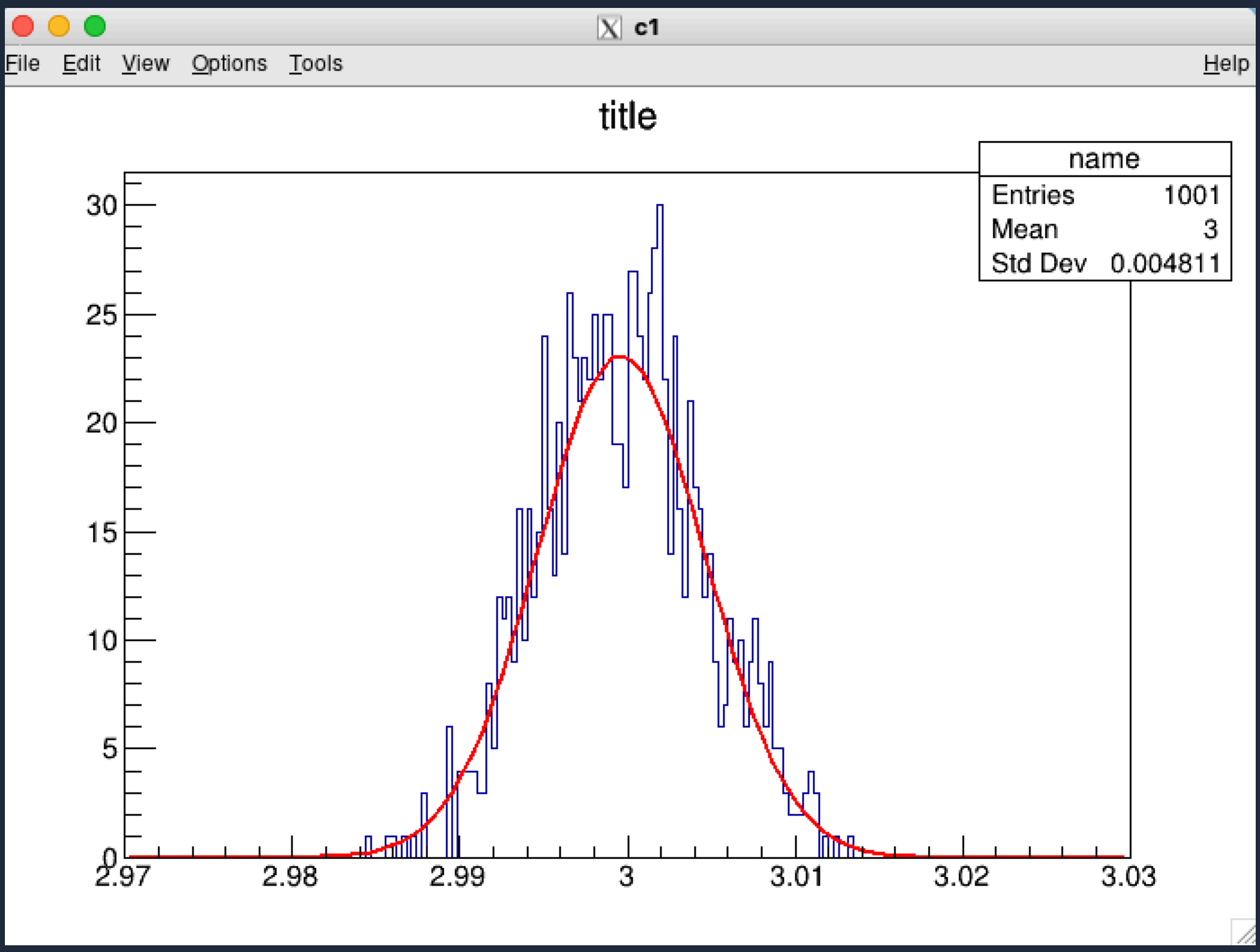


///successful!///

.q //exit the program

Let’s add something into your graph!

```
1  void test(){
2
3   //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
4
5   //c1->SetGrid();  // set grid
6
7   TH1F* graph=new TH1F("name","title",200,2.97,3.03);
8
9   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!
10
11  TRandom n; // define a random variable n
12
13  for (int i=0;i<1000;++i){
14
15  graph->Fill(n.Gaus(3,0.005));  //Determine variable n using Gaussian distribution and fill
   to graph
16
17  }
18
19  graph->Draw();
20
21  graph->Fit("gaus");  //fit your graph with Gaussian function
22
23  }
```

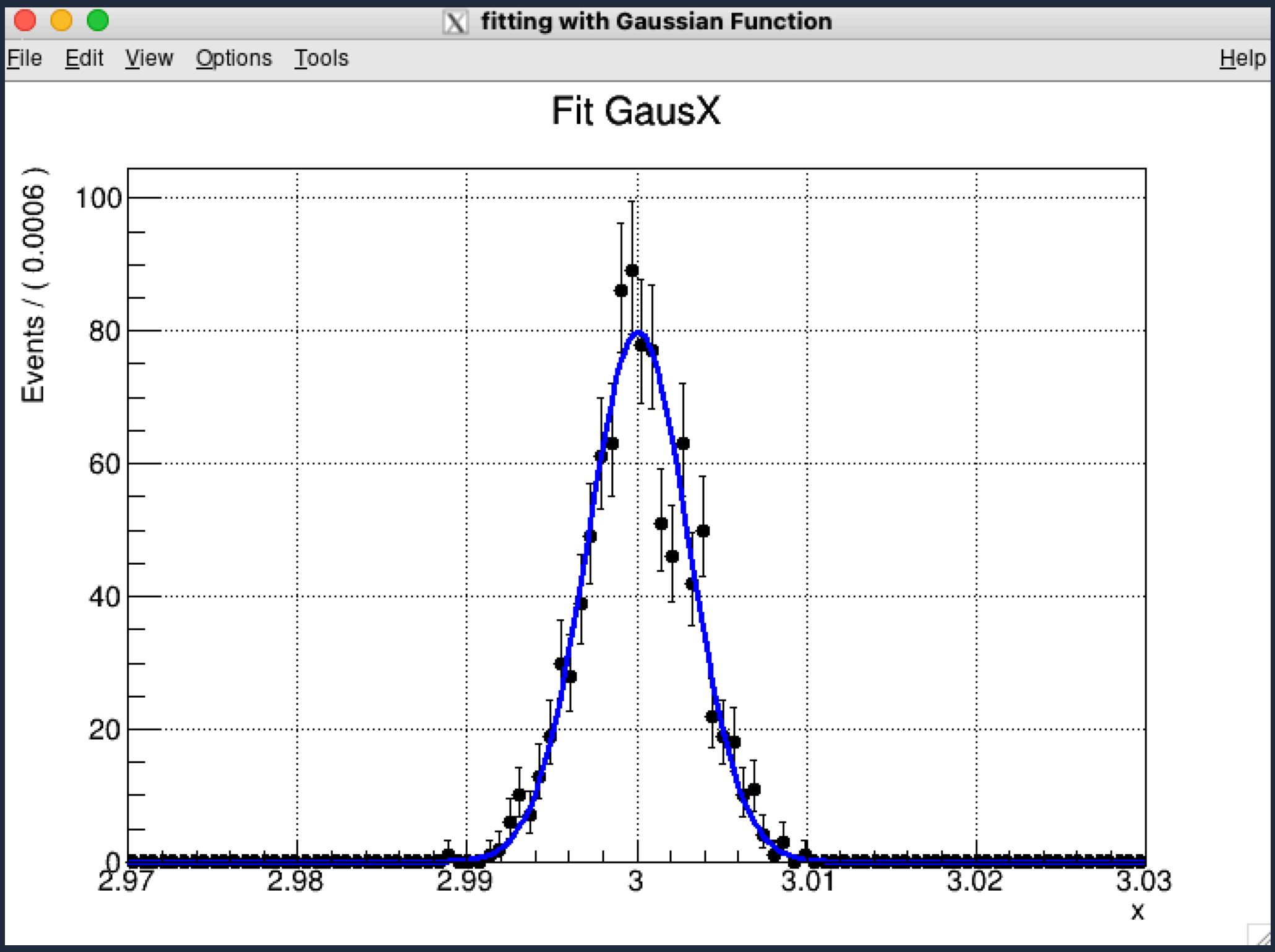


Wow! now you can use root to do some simple fit works. but we usually use RooFit to do more complex job! So, ready for more programs!

Lesson 2 Gaussian fitting using RooFit

```
1  void test(){
2  TCanvas* c1=new TCanvas("c1","fitting with Gaussian function");
3  c1->SetGrid();
4  //build a gaussian pdf (probability density function)
5  /*   RooRealVar: define a variable
6  RooRealVar::RooRealVar(const char* name, const char* title,Double_t minValue,Double_t
   maxValue, const char* unit = "" )    */
7  RooRealVar x("x","x",2.97,3.03);//observable
8  RooRealVar mean("mean","mean",3.0,2.8,3.2);//mean
9  RooRealVar sigma("sigma","sigma",0.003,0.002,0.003);//sigma
10 RooAbsPdf* gaus=new RooGaussian("gaus","gaus",x,mean,sigma);//representing pdf
11 //or: RooGaussian gaus("gaus","gaus",x,mean,sigma) <- this code used more commonly
12 RooRealVar n("n","n",0,0,50000);
13 RooExtendPdf*exp=new RooExtendPdf("exp","exp",*gaus,n);
14 RooAddPdf total("total","total",RooArgList(*exp),RooArgList(n));
```

```
15
16  //    RooAddPdf: define your total PDF
17
18      RooAddPd::RooAddPdf(const char* name, const char* title, const RooArgList& pdfList,
    const RooArgList& coefList, bool recursiveFraction = false)
19
20      //coefList->coefficient List
21
22      //so if you have many pdfs or coefficients, you can use following code
23
24      /* RooArgList pdfList;
25
26       pdfList.add(pdf1);
27
28       pdfList.add(pdf2);
29
30         ……
31
32     RooArgList numList;
33
34       numList.add(num1);
35
36      numList.add(num2);
37
38         ……
39
40     RooAddPdf mytotpdf("mytotpdf", "mytotpdf",pdfList,numList);
41
42    */
43
44  //generate a toy data
45  RooDataSet*data;
46  data=gaus->generate(RooArgSet(x),1000);
47  //fit
48  RooFitResult*result=total.fitTo(*data,Save(),"mer");
49  RooPlot* xframe=x.frame(RooFit::Title("Fit GausX"));
50  data->plotOn(xframe);  //data point
51  global.plotOn(xframe);  //fit curve
52  xframe->Draw();
53  }
```



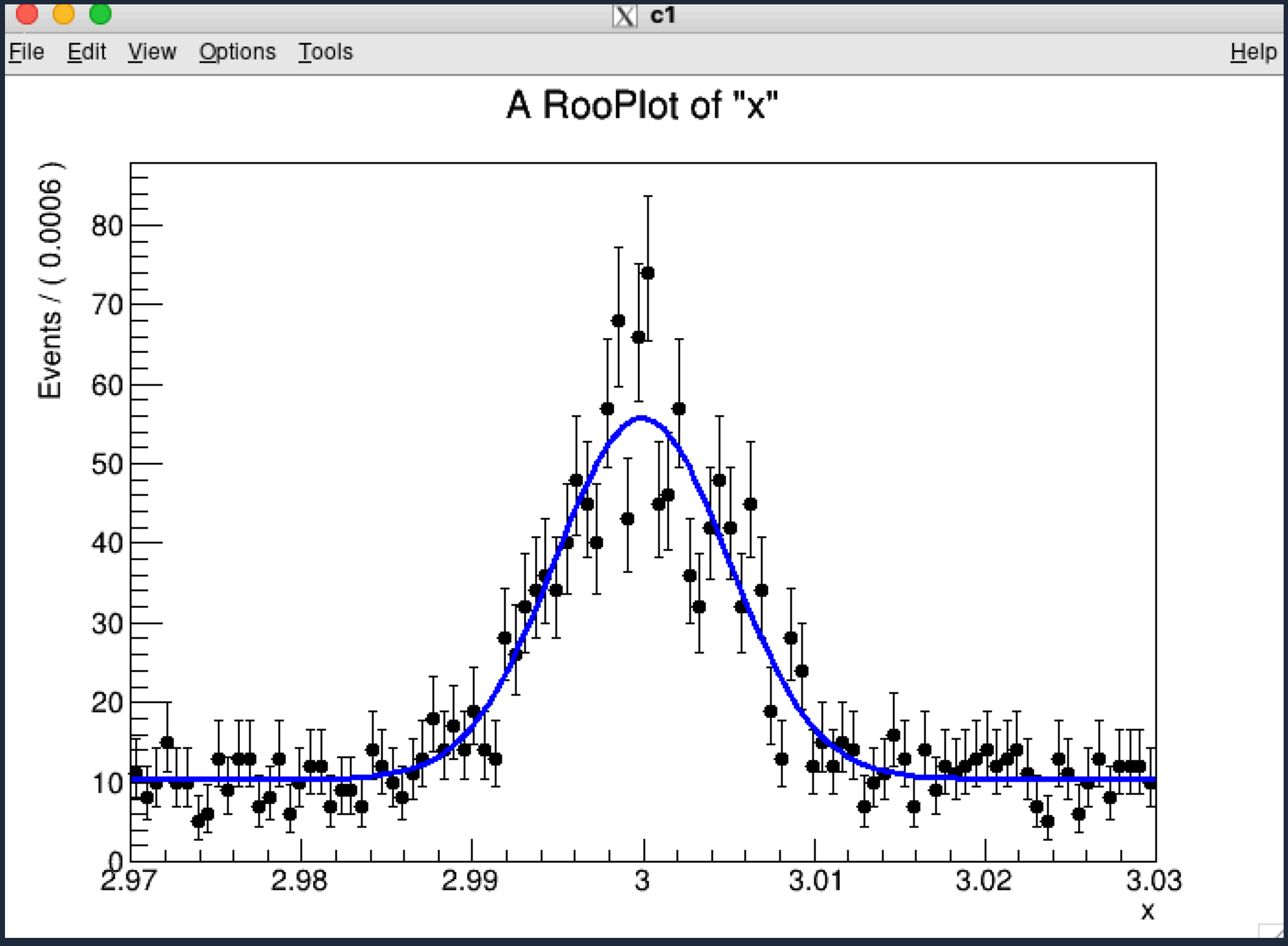
> *"Homework: fitting with two PDFs(Gaussian and Chebyshev polynomials) "*

> *"Hint: Chebyshev polynomials need one variable"*

```
void back(){
using namespace RooFit;
//build gaussian and shev pdf
RooRealVar x("x","x",2.97,3.03);
RooRealVar mean("mean","mean",3.0,2.8,3.2);
RooRealVar sigma("sigma","sigma",0.005,0.003,0.007);
RooRealVar a0("a0","a0",0.005,0.004,0.006);
RooAbsPdf* gaus=new RooGaussian("gaus","gaus",x,mean,sigma);
RooAbsPdf* shev=new RooChebychev("shev","shev",x,a0);
RooRealVar n1("n1","n1",0,0,50000);
RooRealVar n2("n2","n2",0,0,50000);
RooExtendPdf*signal1=new RooExtendPdf("sig1","sig1",*gaus,n1);
RooExtendPdf*signal2=new RooExtendPdf("sig2","sig2",*shev,n2);
RooAddPdf totalPdf("total","total",RooArgList(*gaus,*shev),RooArgList(n1,n2));

```

```
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(),"mer");
25 RooPlot*xframe=x.frame();
26 data1->plotOn(xframe);
27 totalPdf.plotOn(xframe);
28 xframe->Draw();
29 }
```



Lesson 3 2D fitting

Just as the title says, this time we will try 2 dimensions fitting. Relax, it will be very easy if you master what you have learnt before. Additionally, let's make your graph more "beautiful"!

```
1 #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 fit(){
//create two canvases, because you have to know two dimensions fitting.
TCanvas*c1 =new TCanvas("c1","Fitting X");
TCanvas*c2 =new TCanvas("c2","Fitting Y");
c1->SetGrid();
c2->SetGrid();
//build gaussian pdf in two dimension
RooRealVar x("x","x",2.7,11);

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",2.7,11);

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",0.002,0.001,0.003);
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);
RooRealVar n_x2y1("n3","n3",0,0,50000);
```

```
51
52 RooExtendPdf*signal1=new RooExtendPdf("sig1","sig1",*gausx1y1,n_x1y1);
53 RooExtendPdf*signal2=new RooExtendPdf("sig2","sig2",*gausx1y2,n_x1y2);
54 RooExtendPdf*signal3=new RooExtendPdf("sig3","sig3",*gausx2y1,n_x2y1);
55
56 RooAddPdf
   totalPdf("total","total",RooArgList(*gausx1y1,*gausx1y2,*gausx2y1),RooArgList(n_x1y1,n_x1y
   2,n_x2y1));
57
58 //generate data point
59
60 RooDataSet*data1;
61 RooDataSet*data2;
62 RooDataSet*data3;
63 data1=gausx1y1->generate(RooArgSet(x,y),1000);
64 data2=gausx1y2->generate(RooArgSet(x,y),1000);
65 data3=gausx2y1->generate(RooArgSet(x,y),1000);
66
67 data1->append(*data2);
68 data1->append(*data3);
69
70 //fit
71 RooFitResult*result= totalPdf.fitTo(*data1,Save(),"mer");
72 //x dimension
73
74 c1->cd();
75 RooPlot*xframe=x.frame(RooFit::Title("Fit GausX"));
76 data1->plotOn(xframe);
77 totalPdf.plotOn(xframe,Name("fullModel"));
78 totalPdf.plotOn(xframe,Components(*gausx1y1),LineColor(2),LineStyle(1),Name("JpsiJpsi"));
79 totalPdf.plotOn(xframe,Components(*signal2),LineColor(6),LineStyle(1),Name("JpsiPsi2S"));
80 totalPdf.plotOn(xframe,Components(*gausx2y1),LineColor(7),LineStyle(1),Name("Psi2SJpsi"));
81
82 TLegend leg(0.7, 0.7, 0.9, 0.9);
83 leg.AddEntry(xframe->findObject("fullModel"), "Full Model", "L");
84 leg.AddEntry(xframe->findObject("JpsiJpsi"), "J/#psi_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
85 leg.AddEntry(xframe->findObject("JpsiPsi2S"), "J/#psi_{#mu1#mu2}#psi(2S)_{#mu3#mu4}",
   "L");
86 leg.AddEntry(xframe->findObject("Psi2SJpsi"), "#psi(2S)_{#mu1#mu2}J/#psi_{#mu3#mu4}",
   "L");
87
88 xframe->Draw();
89 leg.DrawClone();
90
91
92
93 //y dimension
94
95 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(*gausx1y1),LineColor(2),LineStyle(1),Name("JpsiJpsi"));
totalPdf.plotOn(yframe,Components(*signal2),LineColor(6),LineStyle(1),Name("JpsiPsi2S"));
totalPdf.plotOn(yframe,Components(*gausx2y1),LineColor(7),LineStyle(1),Name("Psi2SJpsi"));


TLegend leg2(0.7, 0.7, 0.9, 0.9);
leg2.AddEntry(yframe->findObject("data1"), "Data", "pe");
leg2.AddEntry(yframe->findObject("fullModel"), "Total Fit", "L");
leg2.AddEntry(yframe->findObject("JpsiJpsi"), "J/#psi_{#mu1#mu2}J/#psi_{#mu3#mu4}", "L");
leg2.AddEntry(yframe->findObject("JpsiPsi2S"), "J/#psi_{#mu1#mu2}#psi(2S)_{#mu3#mu4}",
"L");
leg2.AddEntry(yframe->findObject("Psi2SJpsi"), "#psi(2S)_{#mu1#mu2}J/#psi_{#mu3#mu4}",
"L");

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