

# 粒子物理与核物理实验中的 数据分析

---

杨振伟  
清华大学

第五讲：ROOT在数据分析中的应用(3)

# 上讲摘要

- ROOT的**TTree**类

```
TTree *tree = new TTree("tree","mytree");  
tree->Branch("br1",&br1,"br1/F");
```

- **填充tree, 将tree写入root文件**

```
tree->Fill(); //填充  
TFile *f = new TFile("f1.root","recreate");  
tree->Write(); //写入root文件
```

- **查看root文件中tree的信息**

```
TFile *f = new TFile("f1.root");  
f->ls();  
TTree *tree = (TTree*)f->Get("tree");  
tree->Scan(), tree->Show(i), tree->Print()
```

- **处理tree格式相同的多个文件root文件: TChain**

```
TChain *chain = new TChain();  
chain->Add("f1.root/treename");
```

# 本讲要点

---

- **直方图的运算**

  - 加减乘除： Add, Divide, ...

  - 归一化： Scale

- **ROOT中直方图拟合**

  - `h1->Fit();`

- **含参数的ROOT脚本**

- **ROOT小结**

# 直方图归一化(1)

<http://root.cern.ch/root/html522/TH1.html#TH1:Scale>

直方图的归一化

```
void TH1::Scale(Double_t c1, Option_t *option)
```

默认 $c1=1$ ，把直方图每个区间的值(BinContent)乘以 $c1$

假设  $sum=h1->Integral()$

$h1->Scale(c1)$ 之后，

$$h1->Integral() = c1 * sum$$

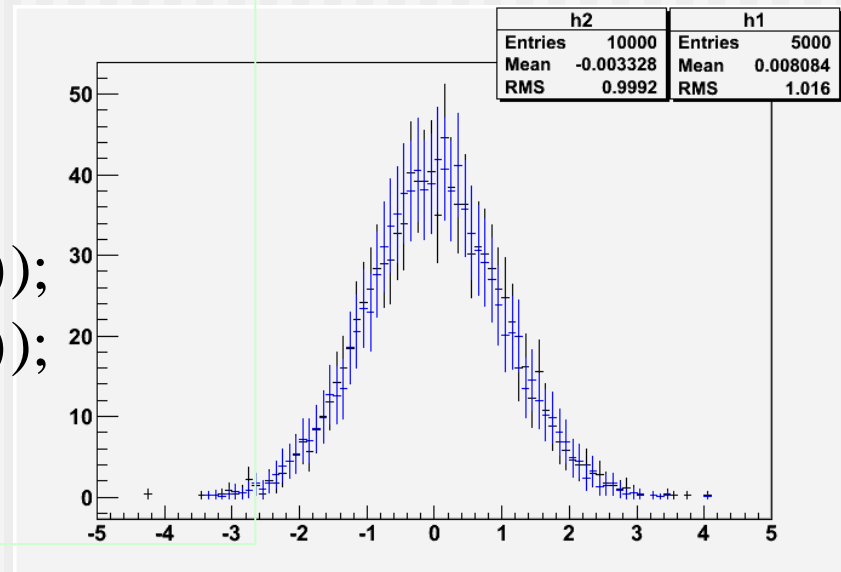
不加参数时， $h1->Scale()$  没有变化(默认 $c1=1$ )

"归一化"后，不仅BinContent变化了， BinError也变化了

# 直方图的归一化(2)

归一化常用于比较两种分布，找出区别。  
所以，将2个直方图归一化到积分相同进行比较才直观。

```
root[0]TH1F *h1=new TH1F("h1","",100,-5,5);
root[1]TH1F *h2=new TH1F("h2","",100,-5,5);
root[2]h1->FillRandom("gaus",5000);
root[3]h2->FillRandom("gaus",10000);
root[4]float norm=1000;
root[5]h1->Scale(norm/h1->Integral());
root[6]h2->Scale(norm/h2->Integral());
root[7]h1->Draw("e");
root[8]h2->Draw("esames");
```



注意Draw()函数的选项

"归一化"之后， $h1$ 或 $h2 \rightarrow Integral() = norm$   
在同一张图上可以看出比较2个分布的差别。

# 直方图四则运算(1)

重要提示:

1. 对直方图进行四则运算操作, 一定要想明白运算的意义  
比如两个直方图的相加与两个随机变量的卷积有什么区别
2. 两个直方图的四则运算, 区间大小和区间数相同才有意义  
四则运算"加减乘除"分别对应

统计量(BinContent)的相加、相减、相乘、相除

3. 如果需要正确处理统计误差, 需要在对ROOT脚本中调用TH1的某个静态成员函数, 即

```
TH1::SetDefaultSumw2();
```

```
void SetDefaultSumw2(Bool_t sumw2 = kTRUE)  
//static function. When this static function is called with  
sumw2=kTRUE, all new histograms will automatically  
activate the storage of the sum of squares of errors,  
ie TH1::Sumw2 is automatically called.
```

# 直方图的四则运算(2)

**相加：** 常用于相同实验的数据叠加，增加统计量。

.....

```
root[1]TH1::SetDefaultSumw2();
```

```
root[1]TH1F *h3=new TH1F(*h1);
```

```
root[2]h3->Add(h1,h2,a,b);
```

结果： h3的BinContent被 $a*h1+b*h2$ 替换，一般 $a=b=1$

**相减：** 常用于从实验测量的分布中扣除本底。

.....

```
root[1]TH1F *h3=new TH1F(*h1);
```

```
root[2]h3->Sumw2();//也可在定义h3前TH1::SetDefaultSumw2();
```

```
root[3]h3->Add(h1,h2,a,-b);
```

结果： h3的BinContent被 $a*h1+b*h2$ 替换，一般 $a=-b=1$

误差：  $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{n_1 + n_2}$  (假设h1和h2独立)

# 直方图的四则运算 (3)

相除

常用于效率的计算。

```
root[1] TH1F *h3=new TH1F(*h1  
root[2] h3->Sumw2();  
root[3] h3->Divide(h1,h2,a,b);  
root[4] h3->Divide(h1,h2,a,b);
```

$$\sigma = \frac{n_1}{n_2} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \quad (\text{h1和h2独立})$$

思考：如果h1和h2不独立，怎么办？比如h1包含于h2

```
root[4] h3->Divide(h1,h2,a,b,"B");
```

$$\sigma = \sqrt{\frac{\frac{n_1}{n_2} (1 - \frac{n_1}{n_2})}{n_2}}$$

二项分布误差

相乘

常用于对分布进行诸如效率等的修正。

```
root> TH1F *h3=new TH1F(*h1);  
root> h3->Sumw2();  
root> h3->Multiply(h1,h2,a,b);
```

$$\sigma = n_1 n_2 \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$



# 直方图四则运算的误差处理

💣\*虽然**ROOT**都提供了较完善的一维直方图运算功能，但对最终结果的误差一定要仔细检查。很多情况下，用户需要从图中读出各频数数值与误差值，并确认运算无误。

包括归一化和加减乘除在内，

如果希望使用直方图的误差，都需要调用

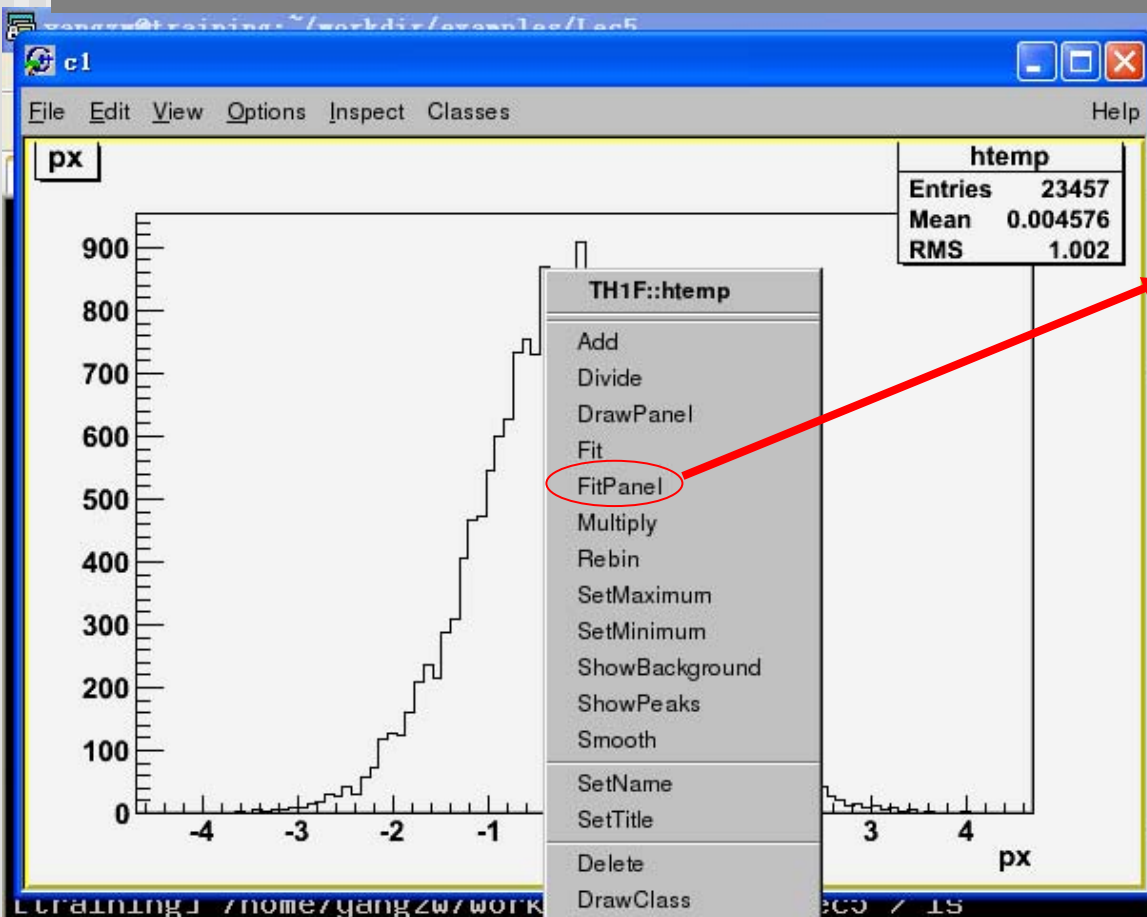
```
TH1::SetDefaultSumw2();
```

或者，对每个直方图(如**hist**)调用

```
hist->Sumw2();
```

# 拟合直方图(1)

将鼠标放到直方图上，右键，出现直方图操作选项，选择**FitPanel**，可以在**FitPanel**中选择拟合的各个选项，比如用什么**函数**拟合，拟合的**区间**，等等。



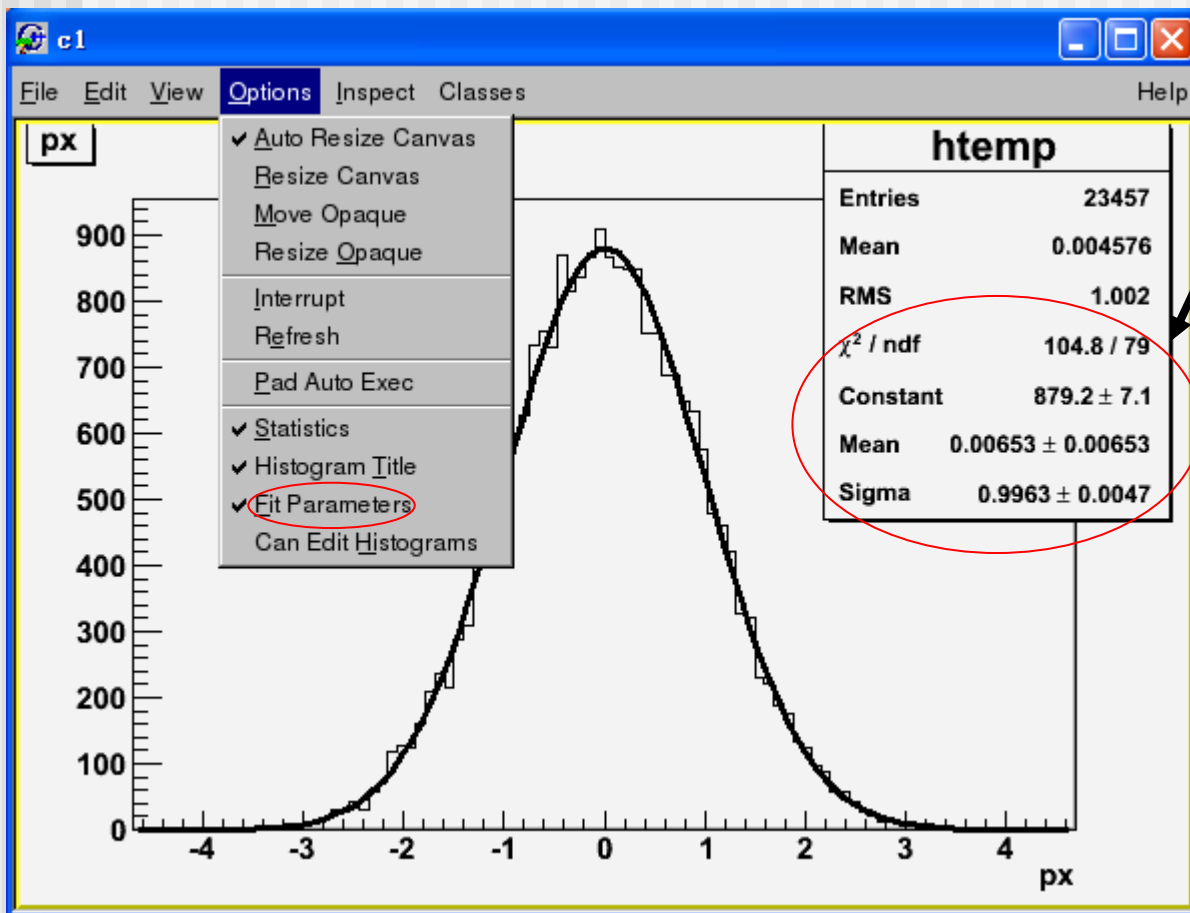
The 'New Fit Panel' dialog box is shown with the following settings:

- Current selection: **htemp::TH1F**
- General tab selected
- Function: **gaus** (Predefined dropdown)
- Operation: **Nop** (radio button selected)
- Selected: **gaus**
- Set Parameters...** button circled in blue
- Fit Settings: **Chi-square** (Method dropdown)
- Fit Options:  Integral,  Best errors,  All weights = 1,  Empty bins, weights=1
- Draw Options:  SAME,  No drawing,  Do not store/draw
- Fit** button circled in blue

At the bottom of the dialog, the following options are visible: LIB Minuit, MIGRAD, Itr: 5000, Pm: DEF.

# 拟合直方图(2)

用默认的高斯拟合，并在**Options**菜单中选中**Fit Parameters**选项，可以看到拟合的结果。



拟合结果给出了高斯分布的**3**个参数：  
常系数、均值、均方差，以及拟合的好坏**chi2/ndf**

并不推荐这种拟合方式：

- 1) 不适合自定义函数拟合
- 2) 不适合批处理

# 下载本讲的例子

---

```
cd <your workding directory>  
cp -r ~yangzw/examples/Lec5 .
```

或者下载到自己本地机器上:

```
scp -r  
username@166.111.32.64:/home/yangzw/  
examples/Lec5 .
```

```
wget  
  hep.tsinghua.edu.cn/~yangzw/CourseData  
  Ana/examples/Lec5.tgz  
tar -zxvf Lec5.tgz
```

# 拟合直方图(3)

/home/yangzw/examples/Lec5/ex51.C

```
hpx->Fit("gaus");
```

```
hpx->Fit("gaus","","",-3,3);
```

自定义拟合函数

```
TF1 *fcn = new TF1("fcn","gaus",-3,3);
```

```
hpx->Fit(fcn,"R");
```

```
gStyle->SetOptFit(); //设置拟合选项
```

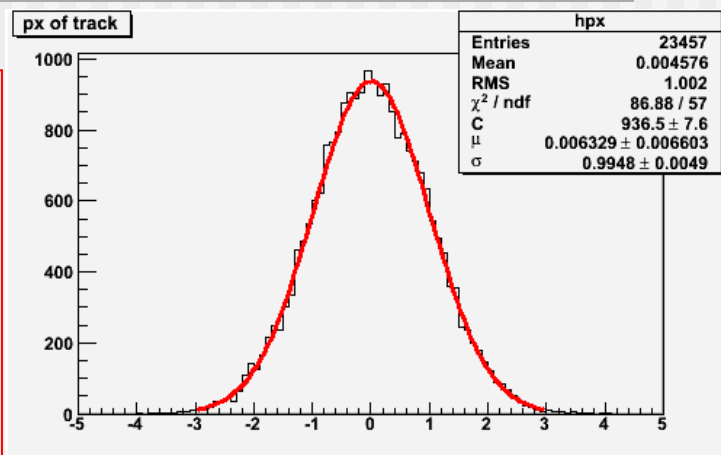
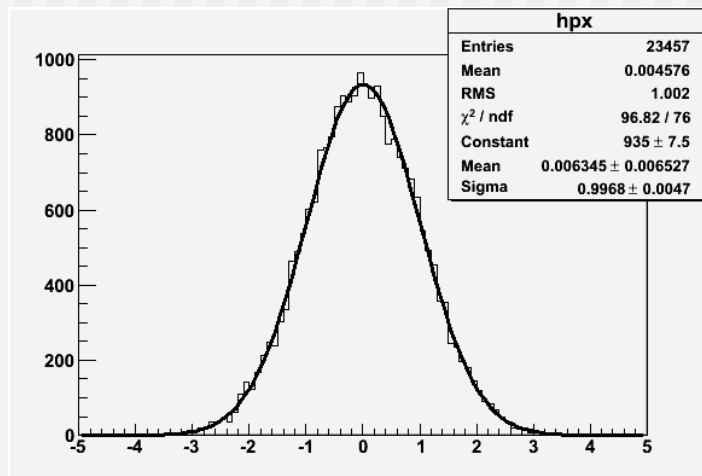
拟合前往往需要给出合理的参数初值

```
fcn->SetParameters(500,mean,sigma);
```

拟合后取出拟合得到的参数

```
Double_t mypar[3];
```

```
fcn->GetParameters(&mypar[0]);
```



```
运行: root -l  
root [0] .L ex51.C  
root [1] ex51r()  
root [2] ex51r2()
```

用自定义的函数拟合直方图

# 拟合直方图(3)

/home/yangzw/examples/Lec5/ex52.C

共振峰(Breit-Wigner分布)加上二次函数本底的拟合(一共6个参数)

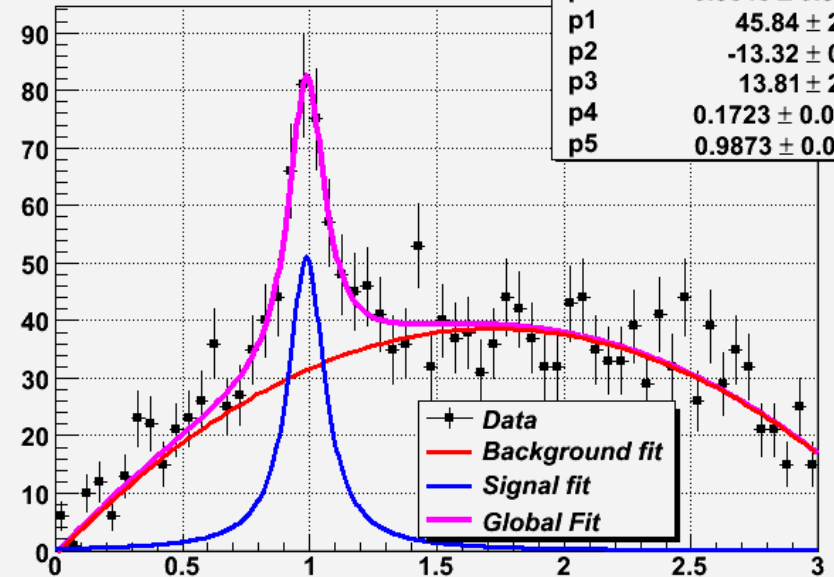
这是下面例子的简化版:

`$ROOTSYS/tutorials/fit/FittingDemo.C`

先自定义本底函数(background)和共振峰函数(lorentianPeak), 再定义这两个函数的和为拟合函数:fitFunction

利用fitFunction定义TF1函数

Lorentzian Peak on Quadratic Bkgd



```
TF1 *fitFcn = new TF1("fitFcn",fitFunction,0,3,6);
```

这里指定函数区间为0-3, 6个参数

fitFcn->SetParameter(4, 0.2); 为某个参数设初值(width)

fitFcn->SetParLimits(5, 0.6, 1.4); 为某参数设置取值范围

```
运行: root -l  
root [0] .L ex52.C
```

注意TLegend的使用

# ROOT小结

- 设定ROOT环境变量:  
ROOTSYS, PATH, LD\_LIBRARY\_PATH
- 绘制各种直方图, 散点图, 数学函数  
TH1F, TH2F, TF1, ...
- 随机数产生子, 各种分布  
gRandom->Rndm, Uniform, Gaus, Exp, ...
- 创建、保存root文件  
TFile \*f = new TFile("myfile.root", "recreate");  
f->Write();
- TTree, TChain的使用  
TTree \*mytree = new TTree("mytree", "my tree");  
mytree->Branch(...);  
用TChain分析相同格式的数据文件。
- 直方图的运算, 拟合  
h1->Fit("function\_name");

# ROOT的重要功能或用法(1)

- ROOT手册13、14章分别是数学库和线性代数，提供很多数学功能，比如Lorentz矢量的操作，特殊函数，矩阵求解运算，求极值等等
- ROOT手册第4章介绍Graphs，适用于不等距数据的图形分析(当然也可以构造不等bin的直方图)
- RooFit，最大似然法拟合等
- 神经网络分析方法, TMVA(多元数据分析)
- ROOT中使用PYTHIA、Geant3/4
- 图形接口...



# ROOT的重要功能或用法(2)

## ■ MakeClass, MakeSelector的运用

比如当前/home/yangzw/examples/Lec5/目录下有文件ex51.root, 其中含有复杂的tree。可以用MakeClass或MakeSelector自动产生分析文件和头文件:

```
root [0] TFile f("ex51.root");
root [1] .ls
TFile**      ex51.root
TFile*       ex51.root
KEY: TTree  t4;1  Reconst events
root [2] t4->MakeClass();
或:       t4->MakeClass("MyClass");
```

自动产生以t4.h和t4.C文件,  
或MyClass.h和MyClass.C  
文件。

类的定义以及Branch地址设  
定、分析框架都已经自动完成。

MakeSelector的用法类似:

```
root [0] TFile f("ex51.root");
root [1] t4->MakeSelector();
或       t4->MakeSelector("MySelector");
```

# ROOT的重要功能或用法(3)

## ■ 独立编译程序进行ROOT分析

尽管在ROOT环境中运行ROOT脚本很方便，但如果分析处理的东西比较复杂，需要长时间运行，独立编译运行比在ROOT环境中运行要快很多，大约有数量级的差别。

/home/yangzw/examples/Lec5/standalone目录是独立编译运行ROOT的例子。这实际上是SDA习题3.7c的一部分。

进入standalone目录后，gmake进行编译就可以运行。

比较make后运行可执行文件所需时间与直接  
`root -l ex37c.C`的运行时间差别。

# 习题

练习需要的root文件都存放在下面目录里：

/home/yangzw/examples/Lec5/exercise/

1. 查看该double\_gaus.root文件。其中存储了名为tree1的TTree。画出tree1中pz的分布，并对该分布进行拟合，在图上显示出拟合的结果，并在屏幕上打印出拟合结果。  
(提示：该分布为两个高斯的叠加，可以自定义一个包含6个参数的TF1进行拟合，分布比较复杂的时候，需要先估计参数的大概值，为拟合函数预设估计值。)  
思考：假设函数
$$\text{fun} = p_0 * \exp(-(x-p_1)^2 / 2 / p_2^2) + p_3 * \exp(-(x-p_4)^2 / 2 / p_5^2)$$
由拟合得到的结果，比较两个高斯的份额
2. hist.root中有两个直方图，对这两个直方图进行加减乘除运算。除法时，查看用"B"选项和不用"B"选项时误差的不同。  
(提示：h1的事例包含于h2的事例，计算误差需要用"B"选项)
3. 利用1.root和2.root，将其中的px分别画到两个直方图h1，h2中。对h1，h2进行加减乘除运算，查看误差情况。比较调用与不调用Sumw2()的差别。

# 参考资料

---

- ROOT手册第5章: Fitting Histogram
- \$ROOTSYS/tutorials/fit目录中的例子
- <http://root.cern.ch/tutorials.html>  
中与Fit有关的例子
- <http://root.cern.ch/howto.html>  
中与Fit有关的例子