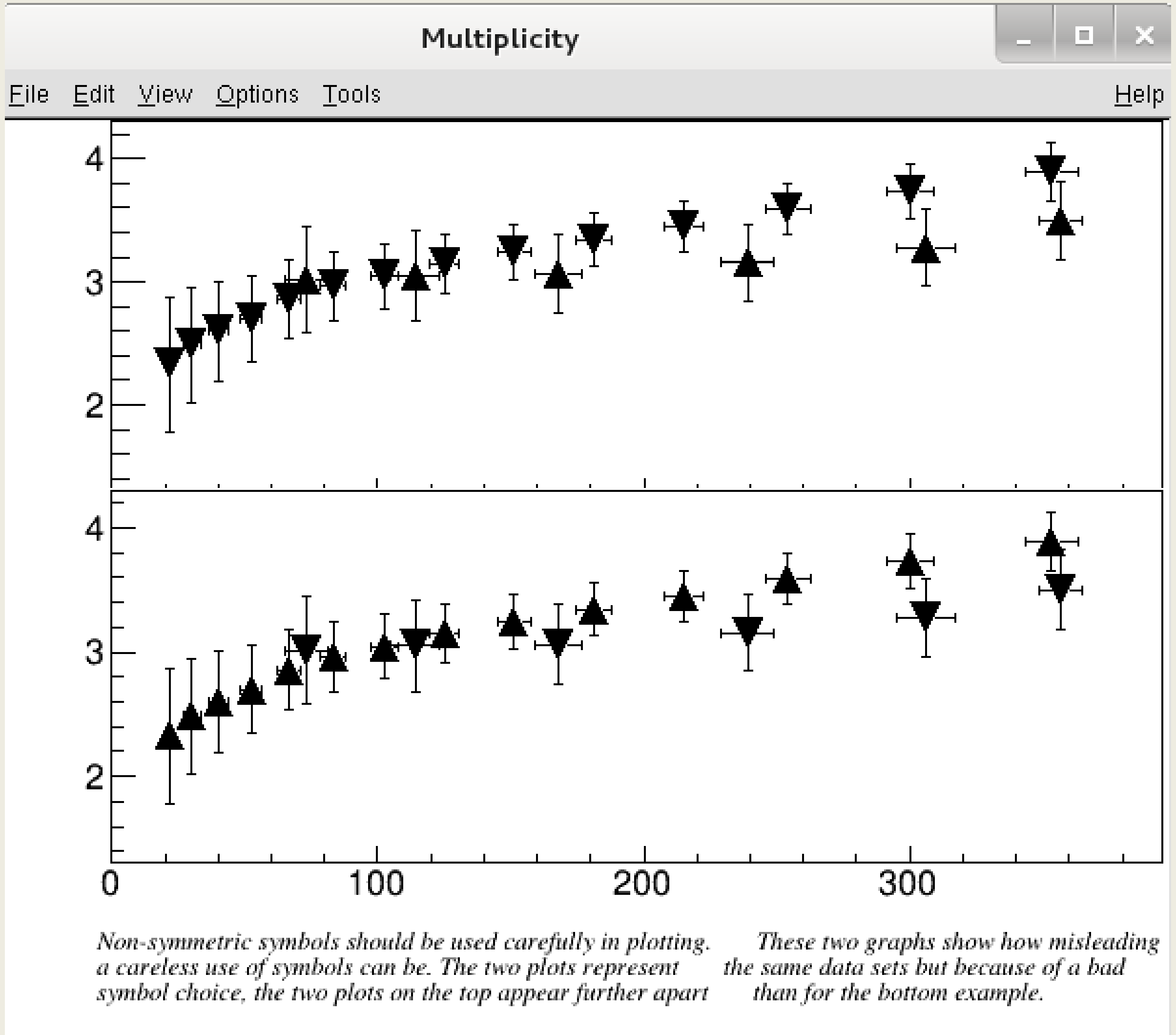
markerwarning.C

ROOT自带的例子，在$ROOTSYS/tutorials/graphics目录下



上下是同样的数据，但给人的感觉是差异不同：
上图差异大；下图差异小

```
// This script illustrates the danger of using asymmetric symbols.
// Non-symmetric symbols should be used carefully in plotting.
// These two graphs show how misleading a careless use of symbols can be.
// The two plots represent the same data sets but because of a bad symbol
// choice, the two plots on the top appear further apart than for the bottom
// example.
//Author: Olivier Couet

void markerwarning()
{
   const int Nph = 14;
   double np_ph[Nph]  = {353.4,300.2,254.3,215.2,181.0,151.3,125.2,102.7,
      83.3, 66.7, 52.5, 40.2, 30.2, 22.0};
   double nc_ph[Nph]  = {3.890,3.734,3.592,3.453,3.342,3.247,3.151,3.047,
      2.965,2.858,2.701,2.599,2.486,2.328};
   double npe_ph[Nph] = {10.068,9.004,8.086,7.304,6.620,6.026,5.504,5.054,
      4.666,4.334,4.050,3.804,3.604,3.440};
   double nce_ph[Nph] = {0.235,0.217,0.210,0.206,0.213,0.223,0.239,0.260,
      0.283,0.318,0.356,0.405,0.465,0.545};

   const int Nbr = 6;
   double np_br[Nbr]  = {357.0,306.0,239.0,168.0,114.0, 73.0};
   double nc_br[Nbr]  = {3.501,3.275,3.155,3.060,3.053,3.014};
   double npe_br[Nbr] = {8.000,11.000,10.000,9.000,9.000,8.000};
   double nce_br[Nbr] = {0.318,0.311,0.306,0.319,0.370,0.429};
```

```
TGraphErrors *phUP = new TGraphErrors(Nph,np_ph,nc_ph,npe_ph,nce_ph);
TGraphErrors *phDN = new TGraphErrors(Nph,np_ph,nc_ph,npe_ph,nce_ph);
TGraphErrors *brUP = new TGraphErrors(Nbr,np_br,nc_br,npe_br,nce_br);
TGraphErrors *brDN = new TGraphErrors(Nbr,np_br,nc_br,npe_br,nce_br);

float Top_margin   = 0.;
float Left_margin  = 0.025;
float Right_margin = 0.005;
float maxPlotPart  = 395;
float Marker_Size  = 1.3;
int   Marker_Style = 8;

float Et_200_Min   = 0.71;
float Et_200_Max   = 3.80;
float Et_130_Min   = 1.21;
float Et_130_Max   = 3.29;

float Nc_200_Min   = 1.31;
float Nc_200_Max   = 4.30;
float Nc_130_Min   = 1.51;
float Nc_130_Max   = 3.89;
```

左边的4个变量没有用到

左边的2个变量没有用到

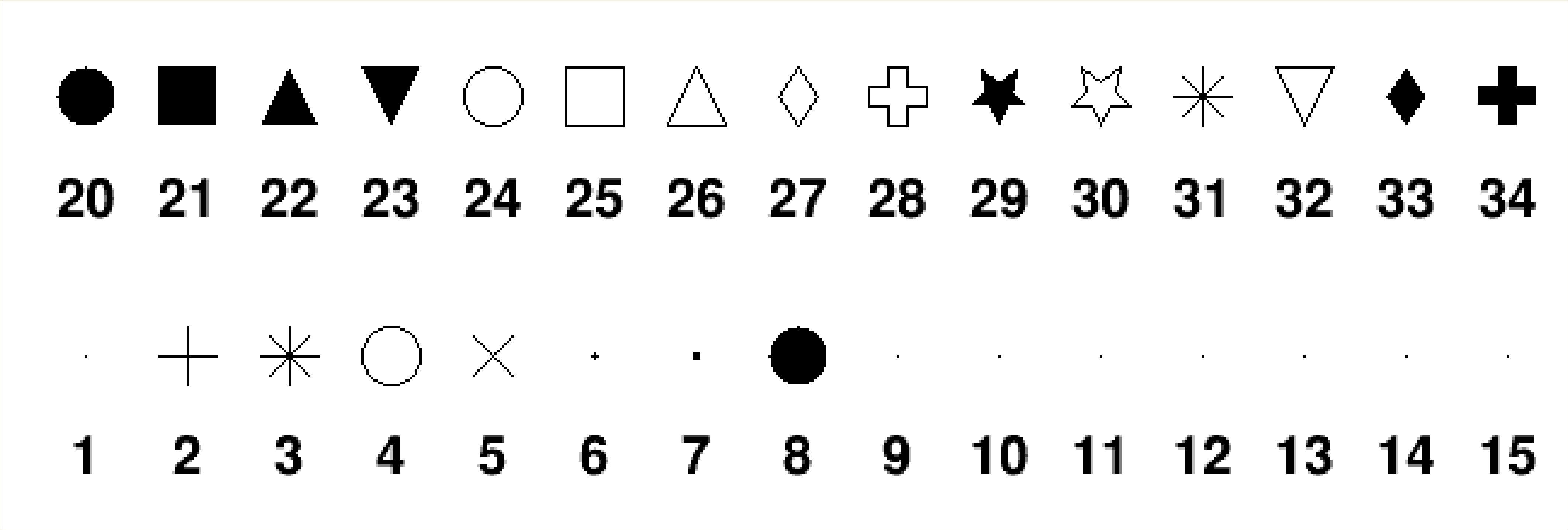
```
// Primitives in Nc200 pad
TPad *padNcUP = new TPad("padNcUP","200 GeV",0.07,0.60,1.,1.00);
padNcUP->Draw();
padNcUP->cd();
padNcUP->SetFillColor(10);
padNcUP->SetFrameFillColor(10);
padNcUP->SetBorderSize(0);
padNcUP->SetLeftMargin(Left_margin);
padNcUP->SetRightMargin(Right_margin);
padNcUP->SetTopMargin(Top_margin+0.005);
padNcUP->SetBottomMargin(0.00);

TH1F* frameNcUP = new TH1F("frameNcUP","",100,0,maxPlotPart);
frameNcUP->GetYaxis()->SetLabelOffset(0.005);
frameNcUP->GetYaxis()->SetLabelSize(0.10);
frameNcUP->SetMinimum(Nc_200_Min);
frameNcUP->SetMaximum(Nc_200_Max);
frameNcUP->SetNdivisions(505,"Y");
frameNcUP->SetNdivisions(505,"X");
frameNcUP->Draw();

brUP->SetMarkerStyle(22);
brUP->SetMarkerSize (2.0);
brUP->Draw("P");
phDN->SetMarkerStyle(23);
phDN->SetMarkerSize (2);
phDN->Draw("P");
```

```
canvasNc->cd();

// Primitives in Nc130 pad
TPad *padNcDN = new TPad("padNcDN","130 GeV",0.07,0.02,1.,0.60);
padNcDN->Draw();
padNcDN->cd();
padNcDN->SetFillColor(10);
padNcDN->SetFrameFillColor(10);
padNcDN->SetBorderSize(0);
padNcDN->SetLeftMargin(Left_margin);
padNcDN->SetRightMargin(Right_margin);
padNcDN->SetTopMargin(Top_margin+0.005);
padNcDN->SetBottomMargin(0.30);

TH1F* frameNcDN = new TH1F("frameNcDN","",100,0,maxPlotPart);
frameNcDN->GetYaxis()->SetLabelOffset(0.005);
frameNcDN->GetYaxis()->SetLabelSize(0.07);
frameNcDN->GetXaxis()->SetLabelOffset(0.005);
frameNcDN->GetXaxis()->SetLabelSize(0.07);
frameNcDN->SetMinimum(Nc_200_Min);
frameNcDN->SetMaximum(Nc_200_Max);
frameNcDN->SetNdivisions(505,"Y");
frameNcDN->SetNdivisions(505,"X");
frameNcDN->Draw();

brDN->SetMarkerStyle(23);
brDN->SetMarkerSize (2.0);
brDN->Draw("P");
```

```
    phUP->SetMarkerStyle(22);
    phUP->SetMarkerSize (2);
    phUP->Draw("P");

    TLatex t1;
    t1.SetTextFont(12); t1.SetTextSize(0.0525);
    t1.DrawLatex(-5,0.6,"Non-symmetric symbols should be used carefully in plotting. \
       These two graphs show how misleading");
    t1.DrawLatex(-5,0.4,"a careless use of symbols can be. The two plots represent \
       the same data sets but because of a bad");
    t1.DrawLatex(-5,0.2,"symbol choice, the two plots on the top appear further apart \
       than for the bottom example.");

    canvasNc->cd();
}
```

改变不同的MarkerStyle

root [0] TMarker d
root [1] d.DisplayMarkerTypes()

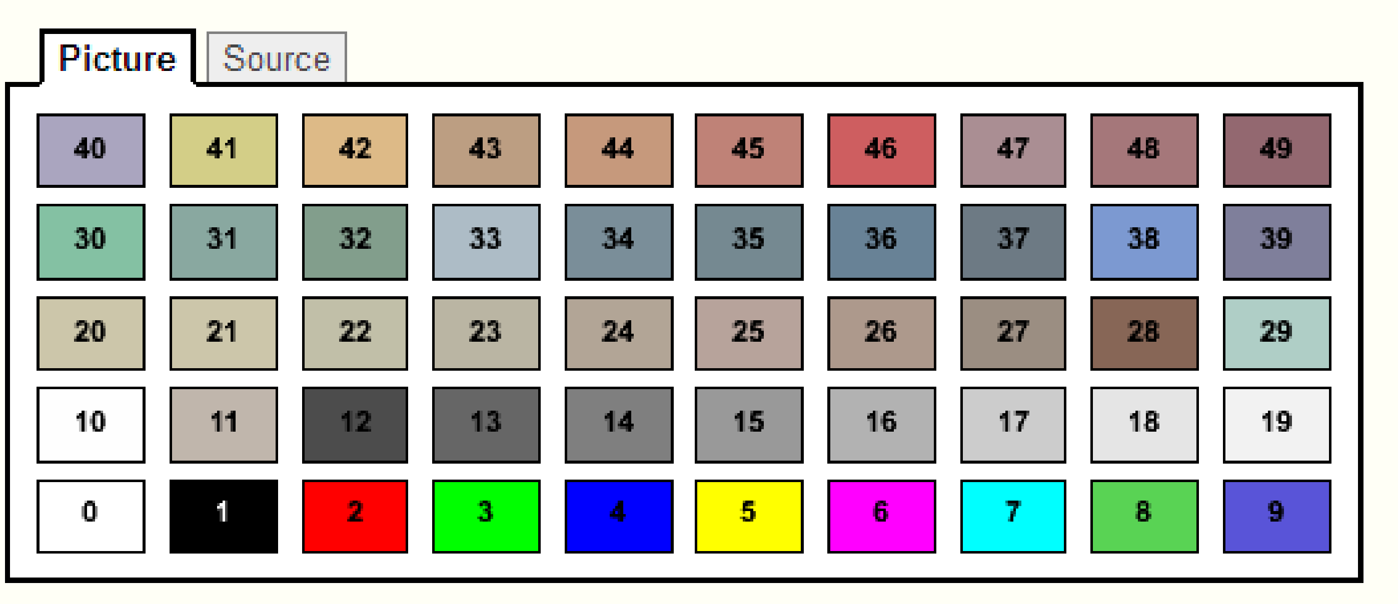


brDN->SetMarkerStyle(8)

改变不同的MarkerColor

d.SetMarkerColor(2)

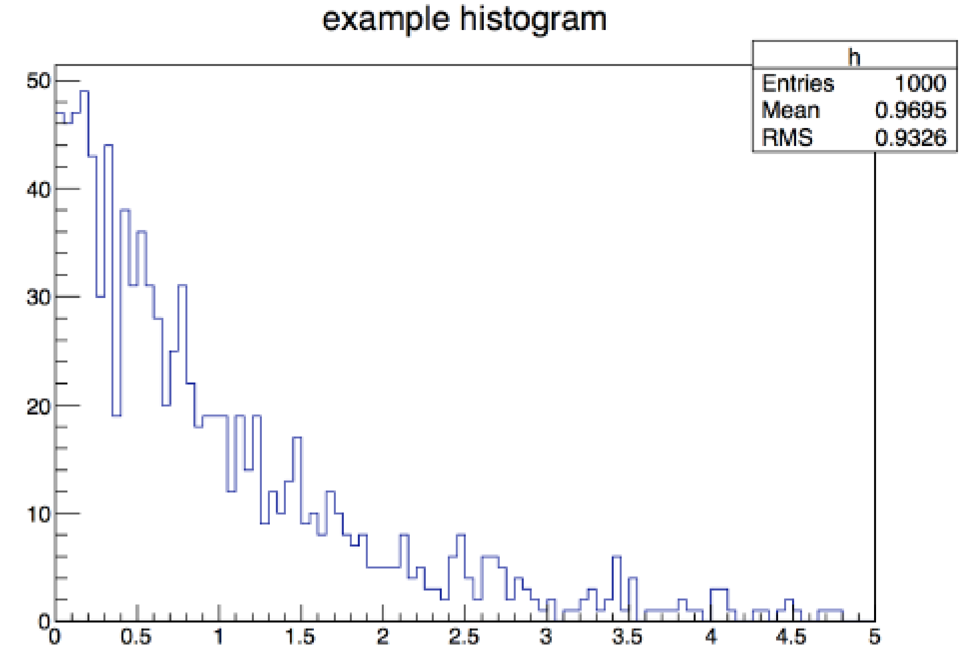
root [0] TCanvas d
root [1] d.DrawColorTable()



siguang@pku.edu.cn

直方图（Histograms）

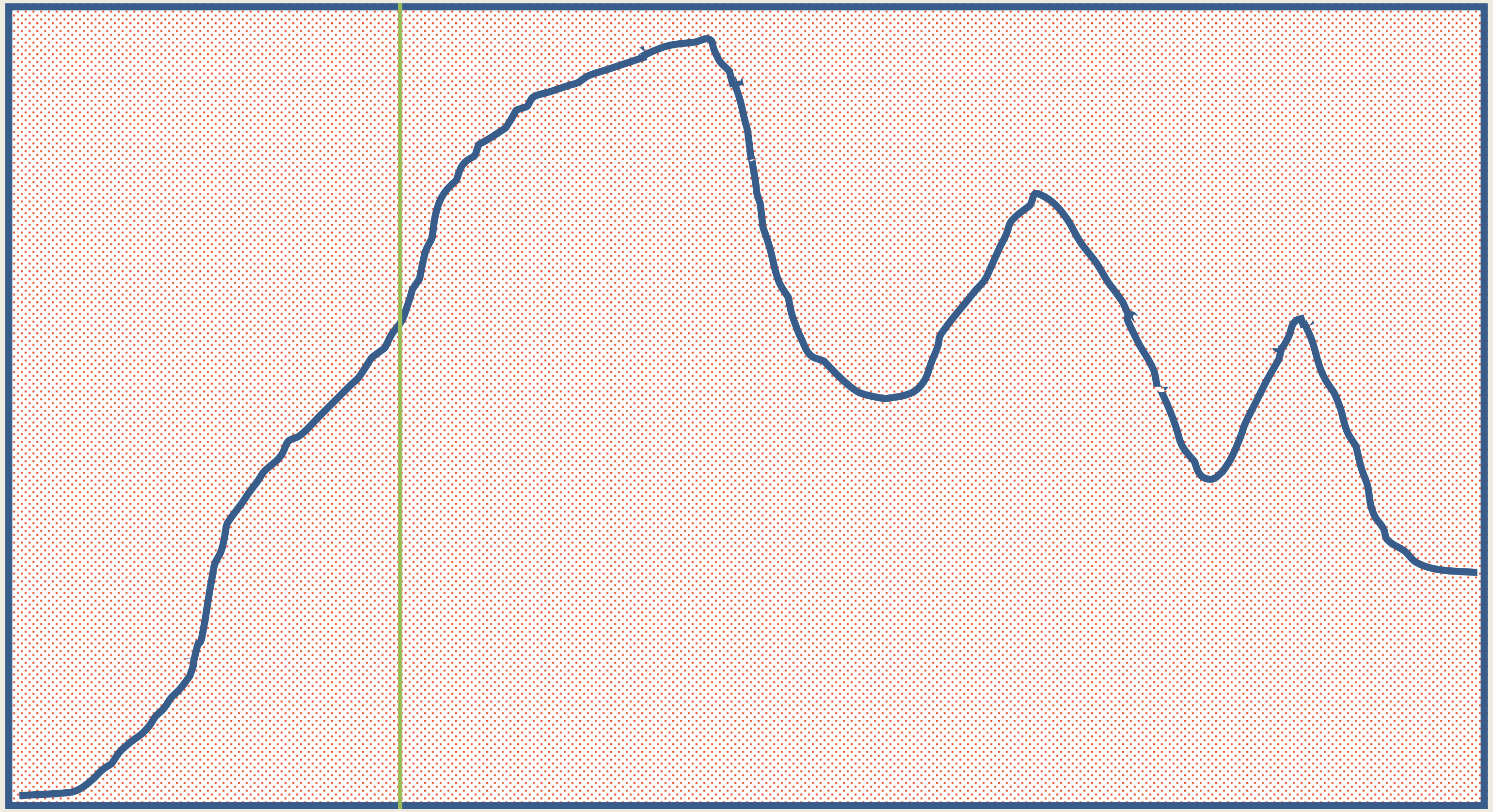
```
root [0] TF1 efunc("efunc","exp([0]+[1]*x)",0.,5.);
root [1] efunc.SetParameter(0,1);
root [2] efunc.SetParameter(1,-1);
root [3] TH1F* h=new TH1F("h","example histogram",100,0.,5.);
root [4] for (int i=0;i<1000;i++) {h->Fill(efunc.GetRandom());}
root [5] h->Draw();
```



在调用efunc.GetRandom()前最好调用efunc. SetNpx(fNpx)

- 尤其是对于变化比较剧烈的函数（有尖锐的峰），SetNpx(fNpx)中的fNpx应该大一些（例如对于1-d函数，4—100000之间的任何数，缺省为100；）
- GetRandom的工作原理：
  - 对函数区间按fNpx的数值分区（bin），并将各区间的高度值相加归1
  - 在[0,1)产生一随机数r
  - 查找r所对应的归1化后的区间，使得r落在两bin之间，然后内插，给出具体的X（X满足从函数左边界到X之间的积分值为r）

一精确的函数产生方法



1)随机产生(x,y)坐标
2)判断y是否在线以下

不要用TRandom而要用TRandom3

- 周期
  - Trandom 2^{31}
  - TRandom3 10^{6000}
- TRandom存在相关性缺陷，故不要用！

TRandom3

周期10^{6000}

```
public:
                    TRandom3 (UInt_t seed = 4357)
                    TRandom3 (const TRandom3&)
            virtual ~TRandom3 ()
     static TClass* Class ()
    virtual TClass* IsA () const
         TRandom3& operator= (const TRandom3&)
   virtual Double_t Rndm (Int_t i = 0)
       virtual void RndmArray (Int_t n, Float_t* array)
       virtual void RndmArray (Int_t n, Double_t* array)
       virtual void SetSeed (UInt_t seed = 0)
       virtual void ShowMembers (TMemberInspector& insp, char* parent)
       virtual void Streamer (TBuffer& b)
               void StreamerNVirtual (TBuffer& b)
```

在]0,1]产生均匀分布的机器无关的随机数：

TRandom3   r;

r.SetSeed(0);

//0---自动随机；>0确保每次运行产生的随机数序列一样

Double_t val = r.Rndm();

TRandom3的常用分布函数

Distributions	Description
`Double_t Uniform(Double_t x1,Double_t x2)`	Uniform random numbers between `x1,x2`
`Double_t Gaus(Double_t mu,Double_t sigma)`	Gaussian random numbers. Default values: `mu=0`, `sigma=1`
`Double_t Exp(Double_t tau)`	Exponential random numbers with mean tau.
`Double_t Landau(Double_t mean,Double_t sigma)`	Landau distributed random numbers. Default values: `mean=0`, `sigma=1`
`Double_t BreitWigner(Double_t mean, Double_t gamma)`	Breit-Wigner distributed random numbers. Default values `mean=0`, `gamma=1`
`Int_t Poisson(Double_t mean)` `Double_t PoissonD(Double_t mean)`	Poisson random numbers
`Int_t Binomial(Int_t ntot,Double_t prob)`	Binomial Random numbers
`Circle(Double_t &x,Double_t &y,Double_t r)`	Generate a random 2D point `(x,y)` in a circle of radius `r`
`Sphere(Double_t &x,Double_t &y, Double_t &z,Double_t r)`	Generate a random 3D point `(x,y,z)` in a sphere of radius `r`
`Rannor(Double_t &a,Double_t &b)`	Generate a pair of Gaussian random numbers with `mu=0` and `sigma=1`

读文件填充TH1F例子

```
root [1] TH1F* h=new TH1F("h","example histogram",100,0.,5.);
root [2] ifstream inp; double x;
root [3] inp.open("expo.dat");
root [4] while (inp >> x) { h->Fill(x); }
root [5] h->Draw();
root [6] inp.close();
```

expo.data
为每行一个数的文本文件：

```
1.22
1.44
2.44
4.33
5.92
1.23
```

rootlogon.C

$ROOTSYS/tutorials/rootlogon.C 的内容如下

```
{
   printf("\nWelcome to the ROOT tutorials\n\n");
   printf("\nType \".x demos.C\" to get a toolbar from which to execute the demos\n");
   printf("\nType \".x demoshelp.C\" to see the help window\n\n");
   printf("==> Many tutorials use the file hsimple.root produced by hsimple.C\n");
   printf("==> It is recommended to execute hsimple.C before any other script\n\n");
}
```

要想导入库函数，在rootlogon.C文件内可加入：
gSystem->Load(“xxx.so”)

rootlogon.C

```
// This is the file rootlogon.C
{
   TStyle *myStyle  = new TStyle("MyStyle","My Root Styles");

   // from ROOT plain style
   myStyle->SetCanvasBorderMode(0);
   myStyle->SetPadBorderMode(0);
   myStyle->SetPadColor(0);
   myStyle->SetCanvasColor(0);
   myStyle->SetTitleColor(1);
   myStyle->SetStatColor(0);

   myStyle->SetLabelSize(0.03,"xyz"); // size of axis values

   // default canvas positioning
   myStyle->SetCanvasDefX(900);
   myStyle->SetCanvasDefY(20);
   myStyle->SetCanvasDefH(550);
   myStyle->SetCanvasDefW(540);

   myStyle->SetPadBottomMargin(0.1);
   myStyle->SetPadTopMargin(0.1);
   myStyle->SetPadLeftMargin(0.1);
   myStyle->SetPadRightMargin(0.1);
   myStyle->SetPadTickX(1);
   myStyle->SetPadTickY(1);
   myStyle->SetFrameBorderMode(0);

   // Din letter
   myStyle->SetPaperSize(21, 28);
```

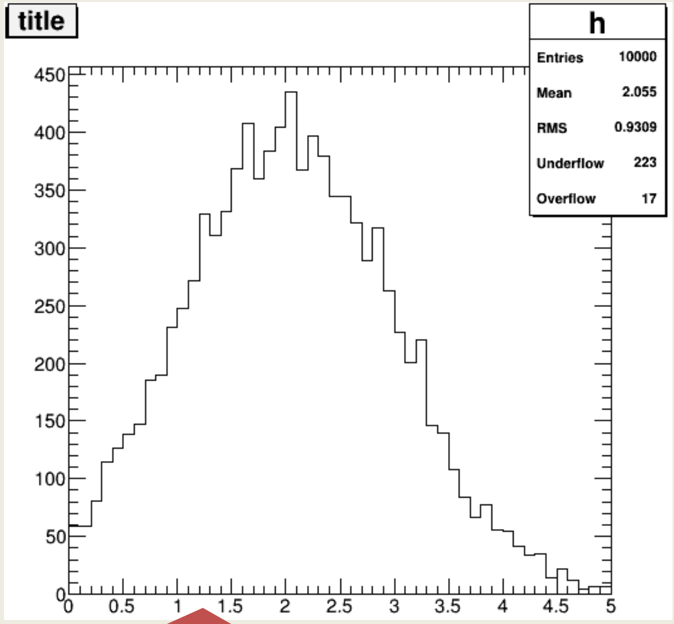
rootlogon.C [continue]

```
    myStyle->SetOptStat(111111);// Show overflow and underflow as well
    myStyle->SetOptFit(1011);
    myStyle->SetPalette(1);

    // apply the new style
    gROOT->SetStyle("MyStyle"); //uncomment to set this style
    gROOT->ForceStyle(); // use this style, not the one saved in root files

    printf("\n Beginning new ROOT session with private TStyle \n");

}
```

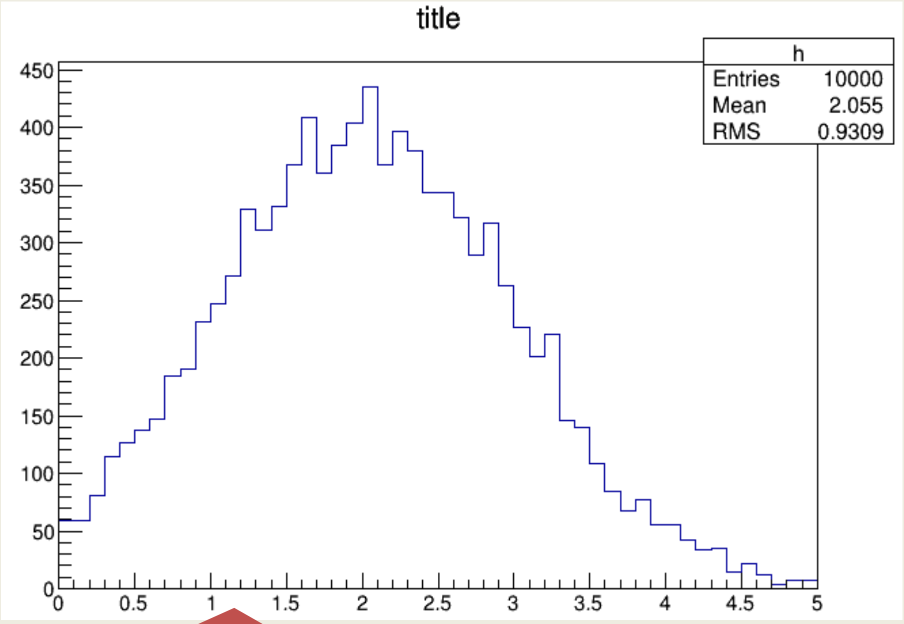
siguang@pku.edu.cn

例子

```
void test(){
  TH1F *h = new TH1F("h","title",50,0,5);
  TRandom3 r;
  Double_t mean=2;
  Double_t sigma = 1;
  for(Int_t i=0; i<10000; i++) h->Fill(r.Gaus(mean,sigma));
  h->Draw();
}
```



运行的目录含上述rootlogon.C



运行的目录无上述rootlogon.C

.rootrc

实现方式，按顺序检查：

- .rootrc //local directory
- $HOME/.rootrc //user directory
- $ROOTSYS/etc/system.rootrc //global ROOT directory

内有

Rint (interactive ROOT executable) specific alias, logon and logoff macros.

Rint.Load: rootalias.C
Rint.Logon: rootlogon.C
Rint.Logoff: rootlogoff.C

Tracking Memory Leaks

在当前目录建立：.rootrc文件，文件内容：
Root.MemStat: 1
Root.ObjectStat: 1

或在ROOTSYS/etc/system.rootrc内修改

在程序中加入：
gObjectTable->Print();例如 create.C文件中：

```
void create(){
  TMemStat mm("gnubuiltin");
  TH1F *h;
  for(Int_t i=0;i<10; i++){
    h = new TH1F("h","",100,0,10);
    gObjectTable->Print();
  }
}
```

Tracking Memory Leaks

```
Warning in <TROOT::Append>: Replacing existing TH1: h (Potential memory leak).

Object statistics
class                          cnt    on heap     size    total size    heap size
=================================================================================
```

每次循环打出一个表，其中TH1F对应的最后3个表中的信息：

```
TH1F                             8          8      976          7808         7808
```

```
TH1F                             9          9      976          8784         8784
```

```
TH1F                            10         10      976          9760         9760
```

ROOT的命令历史记录

cat ~/.root_hist

```
.x create.C
.x create.C
.q
.x create.C
.q
.x create.C
.q
```

wsg@debian:~$ cd
wsg@debian:~$ pwd
/home/wsg
cat .root_hist
与上同样的结果

将常用的代码放一个文件中以提高效率

```
#include <TStyle.h>

// Set the general style options
void SetSgStyle(){
  // No Canvas Border
  gStyle->SetCanvasBorderMode(0);
  gStyle->SetCanvasBorderSize(0);
  // White BG
  gStyle->SetCanvasColor(10);
  // Format for axes
  gStyle->SetLabelFont(22,"xyz");
  gStyle->SetLabelSize(0.06,"xyz");
  gStyle->SetLabelOffset(0.01,"xyz");
  gStyle->SetNdivisions(510,"xyz");
  gStyle->SetTitleFont(22,"xyz");
  gStyle->SetTitleColor(1,"xyz");
  gStyle->SetTitleSize(0.06,"xyz");
  gStyle->SetTitleOffset(0.91);
  gStyle->SetTitleYOffset(1.1);
  // No pad borders
  gStyle->SetPadBorderMode(0);
  gStyle->SetPadBorderSize(0);
  // White BG
  gStyle->SetPadColor(10);
  // Margins for labels etc.
  gStyle->SetPadLeftMargin(0.15);
  gStyle->SetPadBottomMargin(0.15);
  gStyle->SetPadRightMargin(0.05);
  gStyle->SetPadTopMargin(0.06);
  // No error bars in x direction
  gStyle->SetErrorX(0);
  // Format legend
  gStyle->SetLegendBorderSize(0);
  // gStyle->SetLegendFont(22); not In root5.28
  gStyle->SetFillStyle(0);
}
```

用法：
1)在一个文件中例如useful.h敲入左边代码
2) emacs testStyle.C &

```
#include "useful.h"
void testStyle(){
  TH1F *h1 = new TH1F("h1","",100,-10,10);
  SetSgStyle();
  TH1F *h2 = new TH1F("h2","",100,-10,10);
  h1->FillRandom("gaus",1000);
  h2->FillRandom("gaus",1000);
  TCanvas *c1 = new TCanvas("c1","");
  c1->Divide(2,1);
  c1->cd(1);
  h1->Draw();
  c1->cd(2);
  h2->Draw();
}
```

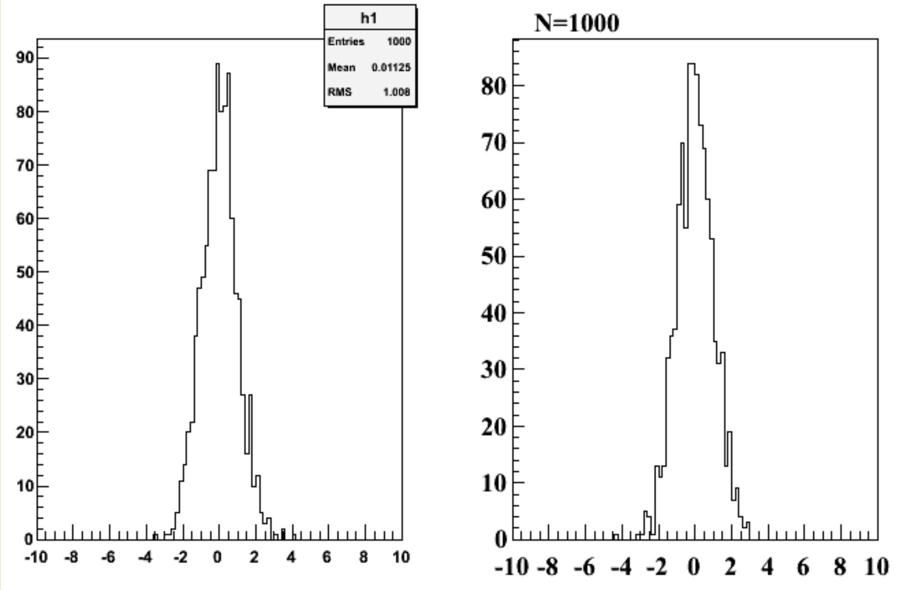
txtN

在useful.h中加入：

```
void txtN(Double_t x0, Double_t y0,TH1 *h,Char_t sName[]="N=%.0f",Double_t sizeTxt=0.06){
  h->SetStats(kFALSE);
  TLatex *ltx = new TLatex();
  ltx->SetNDC(kTRUE);
  ltx->SetTextColor(h->GetLineColor());
  ltx -> SetTextFont(22);
  ltx->SetTextSize(sizeTxt);
  ltx->DrawLatex(x0,y0,Form(sName,h->GetEntries()));
  gPad->Modified();
  gPad->Update();
}
```

在testStyle.C最后一行加入：

```
txtN(0.2,0.95,h2);
```



newTH1F

```
TH1F * newTH1F(Char_t name[]="h1",Double_t binw=0.01, Double_t LowBin=0.0, Double_t HighBin=3.0,Boo
_t MevTitle = kTRUE,Int_t iMode=-1){
  Int_t nbin = TMath::Nint( (HighBin - LowBin)/binw );
  HighBin = binw*nbin + LowBin;

  TH1F *h = new TH1F(name,"",nbin,LowBin,HighBin);
  if(MevTitle) h->GetYaxis()->SetTitle(Form("Events / (%.0fMeV/c^{2})",h->GetBinWidth(1)*1000));
  h->SetMinimum(0.0);
  h->GetYaxis()->SetTitleOffset(1.1);
  if(iMode>=0 && iMode<14){
    Int_t iMarker[] = {20,21,24,25,28,29,30,27,3,  5,2, 26,22,23};
    Int_t iColor[]  = { 2, 4, 6, 9, 1,50,40,31,41,35,44,38,47,12};
    h ->SetMarkerStyle(iMarker[iMode]);
    h ->SetMarkerColor(iColor[iMode]);
    h ->SetLineColor(iColor[iMode]);
  }

  return h;
}
```

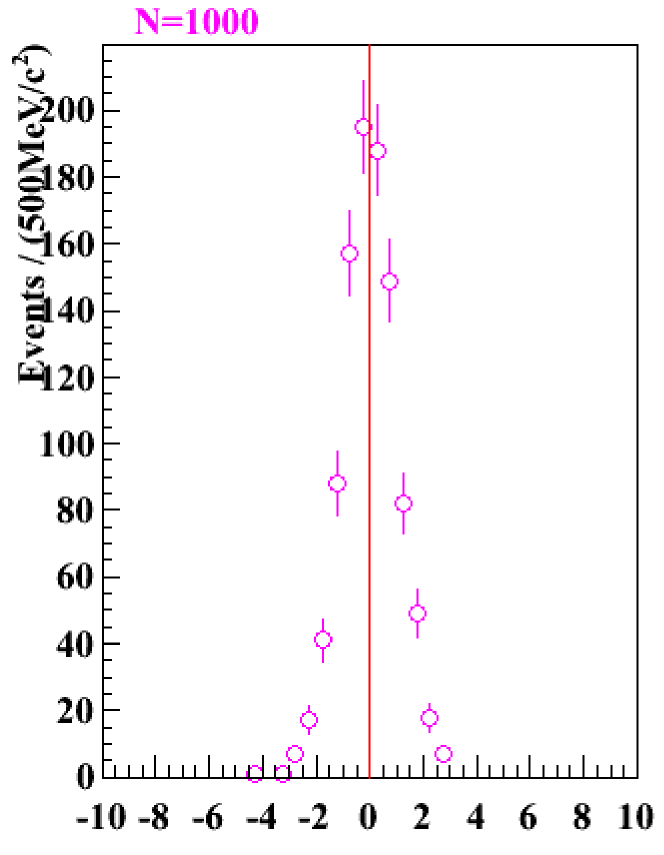
```
void testStyle2(){
  TH1F *h1 = newTH1F("h1",0.5,-10,10,kTRUE,1);
  SetSgStyle();
  TH1F *h2 = newTH1F("h2",0.5,-10,10,kTRUE,2);
  h1->FillRandom("gaus",1000);
  h2->FillRandom("gaus",1000);
  TCanvas *c1 = new TCanvas("c1","");
  c1->Divide(2,1);
  c1->cd(1);
  h1->Draw("EP");
  c1->cd(2);
  h2->Draw("EP");
  txtN(0.2,0.95,h2);
}
```

siguang@pku.edu.cn

LineX1

```
void LineX1(Double_t atX,Int_t iColor=kRed,Int_t iStyle=1,Double_t iWidth=1){
  gPad->Modified();
  gPad->Update();
  TLine *l1 = new TLine(atX,gPad->GetUymin(),atX,gPad->GetUymax());
  l1->SetLineColor(iColor);
  l1->SetLineStyle(iStyle);
  l1->SetLineWidth(iWidth);
  l1->Draw();
}
```

```
void testStyle2(){
  TH1F *h1 = newTH1F("h1",0.5,-10,10,kTRUE,1);
  SetSgStyle();
  TH1F *h2 = newTH1F("h2",0.5,-10,10,kTRUE,2);
  h1->FillRandom("gaus",1000);
  h2->FillRandom("gaus",1000);
  TCanvas *c1 = new TCanvas("c1","");
  c1->Divide(2,1);
  c1->cd(1);
  h1->Draw("EP");
  c1->cd(2);
  h2->Draw("EP");
  txtN(0.2,0.95,h2);
  LineX1(0.0);
}
```



避免多次引用保护

```
#ifndef USEFUL_H
#define USEFUL_H
#include <TStyle.h>

// Set the general style options
void SetSgStyle(){
  // No Canvas Border
  gStyle->SetCanvasBorderMode(0);
  gStyle->SetCanvasBorderSize(0);
  // White BG
  gStyle->SetCanvasColor(10);
  // Format for axes
  gStyle->SetLabelFont(22,"xyz");
  gStyle->SetLabelSize(0.06,"xyz");
```

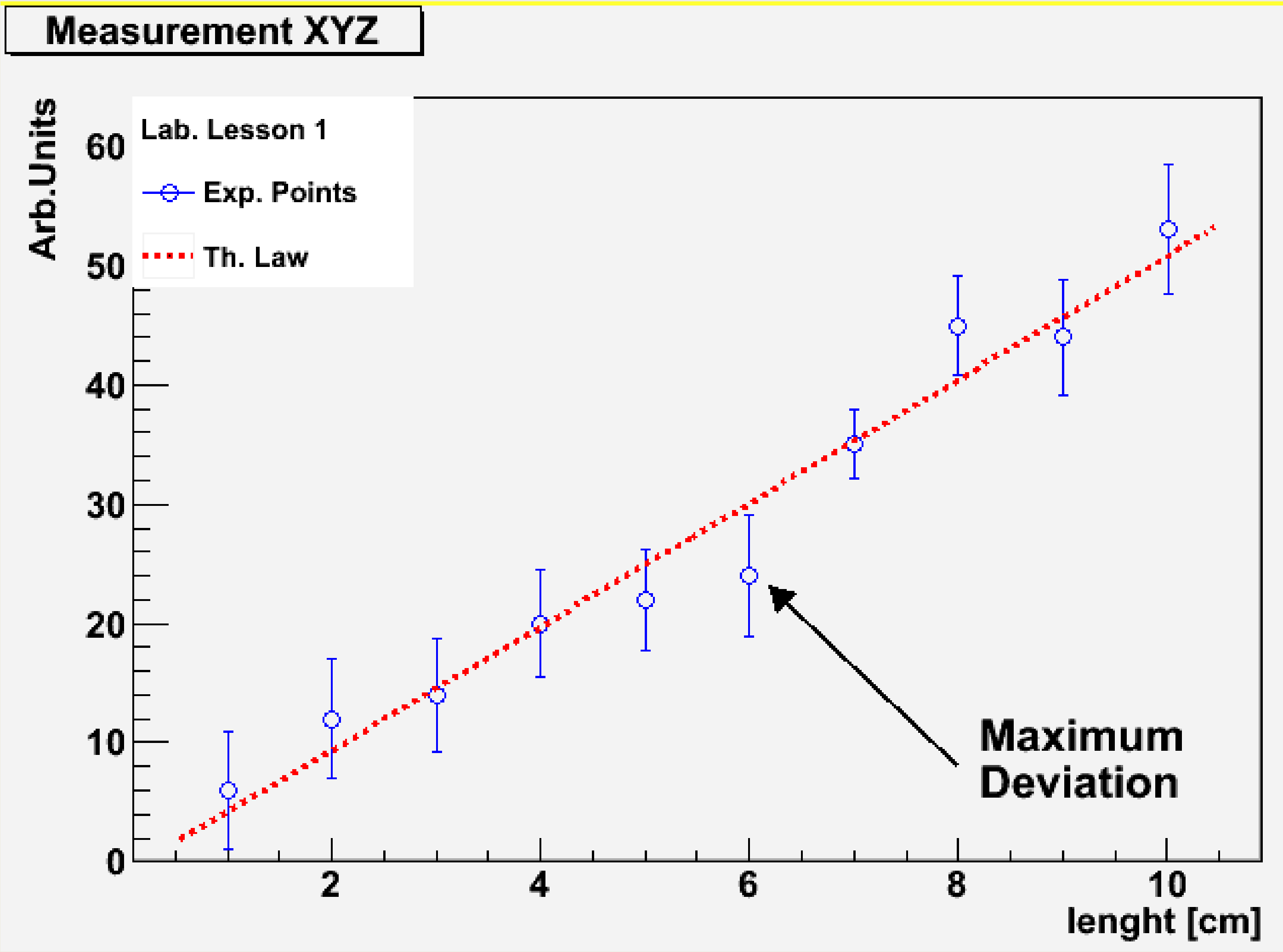
…

…

…

```
#endif
```

marco1.C


Measurement XYZ
Arb.Units
Lab. Lesson 1
Exp. Points
Th. Law
60
50
40
30
20
10
0
2
4
6
8
10
lenght [cm]
Maximum
Deviation

marco1.C

```
// Builds a graph with errors, displays it and saves it as
// image. First, include some header files (within CINT,
// these will be ignored).

#include "TCanvas.h"
#include "TROOT.h"
#include "TGraphErrors.h"
#include "TF1.h"
#include "TLegend.h"
#include "TArrow.h"
#include "TLatex.h"

void macro1(){
    // The values and the errors on the Y axis
    const int n_points=10;
    double x_vals[n_points]=
            {1,2,3,4,5,6,7,8,9,10};
    double y_vals[n_points]=
            {6,12,14,20,22,24,35,45,44,53};
    double y_errs[n_points]=
            {5,5,4.7,4.5,4.2,5.1,2.9,4.1,4.8,5.43};

    // Instance of the graph
    TGraphErrors graph(n_points,x_vals,y_vals,NULL,y_errs);
    graph.SetTitle("Measurement XYZ;lenght [cm];Arb.Units");

    // Make the plot estetically better
    graph.SetMarkerStyle(kOpenCircle);
    graph.SetMarkerColor(kBlue);
    graph.SetLineColor(kBlue);

    // The canvas on which we'll draw the graph
    TCanvas* mycanvas = new TCanvas();

    // Draw the graph !
    graph.DrawClone("APE");
```

marco1.C [continue]

```
    // Define a linear function
    TF1 f("Linear law","[0]+x*[1]",.5,10.5);
    // Let's make the funcion line nicer
    f.SetLineColor(kRed); f.SetLineStyle(2);
    // Fit it to the graph and draw it
    graph.Fit(&f);
    f.DrawClone("Same");

    // Build and Draw a legend
    TLegend leg(.1,.7,.3,.9,"Lab. Lesson 1");
    leg.SetFillColor(0);
    graph.SetFillColor(0);
    leg.AddEntry(&graph,"Exp. Points");
    leg.AddEntry(&f,"Th. Law");
    leg.DrawClone("Same");

    // Draw an arrow on the canvas
    TArrow arrow(8,8,6.2,23,0.02,"|>");
    arrow.SetLineWidth(2);
    arrow.DrawClone();

    // Add some text to the plot
    TLatex text(8.2,7.5,"#splitline{Maximum}{Deviation}");
    text.DrawClone();

    mycanvas->Print("graph_with_law.pdf");
}

#ifndef __CINT__
int main(){
    macro1();
    }
#endif
```

编译（Compilation）

用ACLiC (The Automatic Compiler of Libraries for CINT--C/C++ interpreter）进行编译

```
root [0] .L macro1.C++
root [1] macro1()
```

Compile a Macro with the Compiler

- **g++ -o macro1.exe macro1.C `root-config --cflags --libs`**
- **./macro1.exe**

- evince **graph_with_law.pdf &** **查看pdf文件，**是Ubuntu和GNOME默认的文档阅读工具。

root-config --cflags –libs的作用

```
wsg@debian:~/Utilities/plots$ root-config --cflags --libs
-pthread -m64 -I/home/wsg/work/root/534/include
-L/home/wsg/work/root/534/lib -lCore -lCint -lRIO -lNet -lHist
-lGraf -lGraf3d -lGpad -lTree -lRint -lPostscript -lMatrix
-lPhysics -lMathCore -lThread -pthread -lm -ldl -rdynamic
```

ExampleMarco.C

```
/*
 * Note that this file can be either used as a compiled program
    or as a ROOT macro.
 * If it is used as a compiled program, additional include statements
   and the definition of the main program have to be made. This is
   not needed if the code is executed at the ROOT prompt.
                                                                   */

//#ifndef __CINT__      // These include-statements are needed if the program is
#include "TFile.h"     // run as a "stand-alone application", i.e. if it is not
#include "TH1F.h"      // called from an interactive ROOT session.
#include "TCanvas.h"
#include "TMath.h"
// eventually, load some C libraries
#include <math.h>

void ExampleMacro();

//_____________________________________________________________________________
int main()
{
  ExampleMacro();
  return 0;
}
//#endif
```

```
//______________________________________________________________________________

/*
   * From here on, the code can also be used as a macro
   * Note though, that CINT may report errors where there are none
     in C++. E.g. this happens here where CINT says that f1 is
     out of scope ...

     ==>>  put your code here
      (remember to update the name of you Macro in the
       lines above if you intend to comile the code)
                                                                  */
void ExampleMacro() {
  // Create a histogram, fill it with random gaussian numbers
  TH1F *h = new TH1F ("h", "example histogram", 100, -5.,5.);
  h->FillRandom("gaus",1000);

  // draw the histogram
  h->DrawClone();

/* - Create a new ROOT file for output
   - Note that this file may contain any kind of ROOT objects, histograms,
     pictures, graphics objects etc.
   - the new file is now becoming the current directory */
  TFile *f1 = new TFile("ExampleMacro.root","RECREATE","ExampleMacro");

  // write Histogram to current directory (i.e. the file just opened)
  h->Write();

  // Close the file.
  //   (You may inspect your histogram in the file using the TBrowser class)
  f1->Close();
}
```

将TH1F类写入root文件中

Compile&Run ExampleMarco.C [continue]

```
>g++ -o ExampleMacro.exe ExampleMacro.C `root-config --cflags --libs`
> ./ExampleMacro.exe
```

ExampleMacro_GUI.C

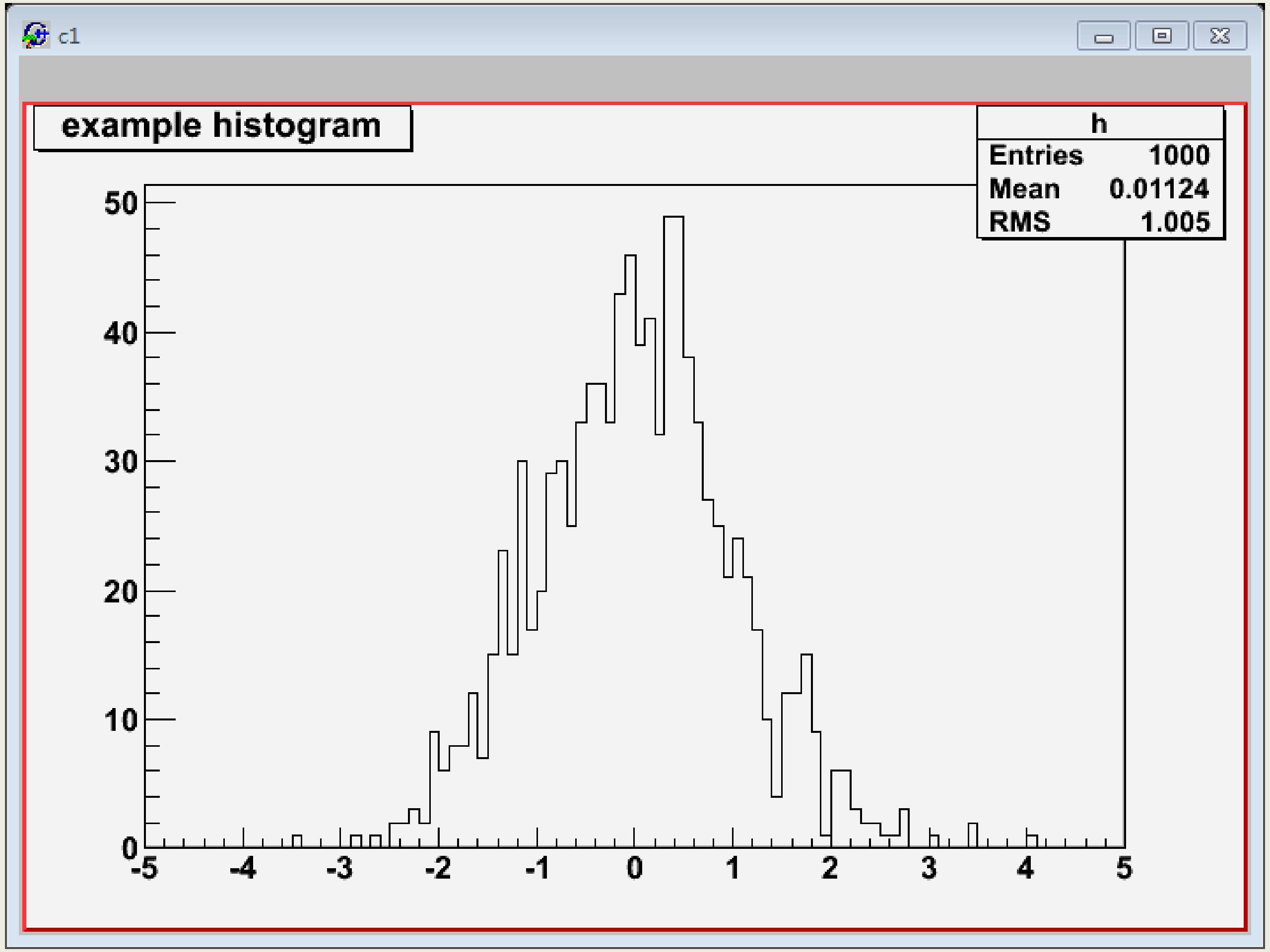
```
/*
 This piece of code demonstrates how a root macro is used as a standalone
    application with full acces the grapical user interface (GUI) of ROOT   */

// include ALL header files needed
#ifndef __CINT__
#include "TROOT.h"
#include "TApplication.h"
#include "TBrowser.h"
#include "TFile.h"
#include "TH1F.h"
#include "TCanvas.h"
#include "TMath.h"
#endif
// eventually, include some additoinal C or C++ libraries
#include <math.h>

//     ==>>  put the code of your macro here
void ExampleMacro_GUI() {
  // Create a histogram, fill it with random gaussian numbers
  TH1F *h = new TH1F ("h", "example histogram", 100, -5.,5.);
  h->FillRandom("gaus",1000);

  // draw the histogram
  h->DrawClone();
```

ExampleMacro_GUI.C [continue]

```
/* - Create a new ROOT file for output
   - Note that this file may contain any kind of ROOT objects, histograms,
     pictures, graphics objects etc.
   - the new file is now becoming the current directory */
  TFile *f1 = new TFile("ExampleMacro.root","RECREATE","ExampleMacro");

  // write Histogram to current directory (i.e. the file just opened)
  h->Write();

  // Close the file.
  //    (You may inspect your histogram in the file using the TBrowser class)
  f1->Close();
}

// the "dressing" code for a stand-alone ROOT application starts here
#ifndef __CINT__
void StandaloneApplication(int argc, char** argv) {
  // ==>> here the ROOT macro is called
  ExampleMacro_GUI();
}

// This is the standard main of C++ starting a ROOT application
int main(int argc, char** argv) {
   TApplication app("Root Application", &argc, argv);
   StandaloneApplication(app.Argc(), app.Argv());
   app.Run();
   return 0;
}
#endif
```

Compile&Run ExampleMarco_GUI.C [continue]

```
>g++ -o ExampleMacro_GUI.exe ExampleMacro_GUI.C `root-config --cflags --libs`
> ./ExampleMacro_GUI.exe
```



进一步了解TGraphErrors

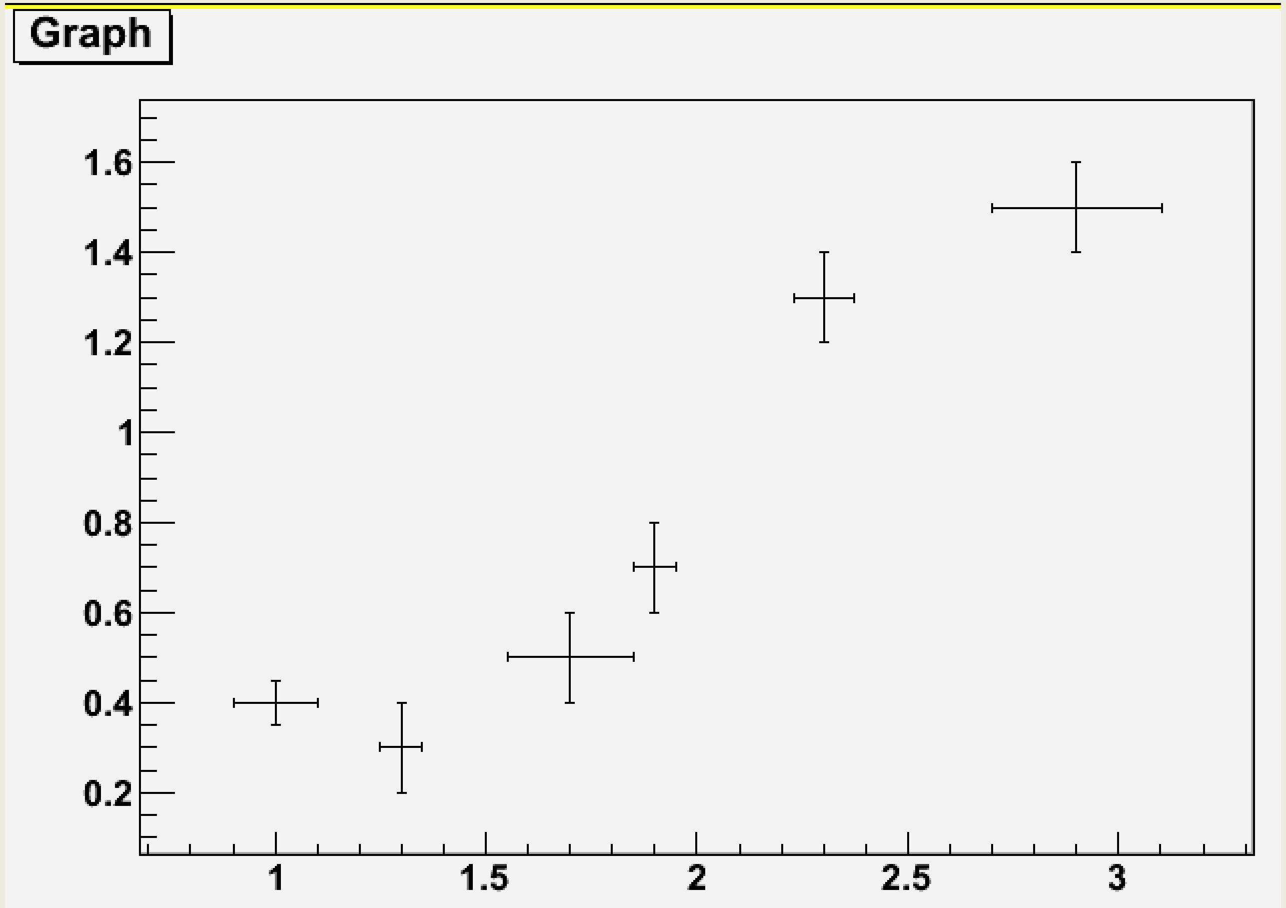
从文本文件中直接读取数据画图：

```
TGraphErrors(const char *filename,
const char *format="%lg %lg %lg %lg", Option_t *option="");
```

The format string can be:

- "\%lg \%lg" read only 2 first columns into X,Y
- "\%lg \%lg \%lg" read only 3 first columns into X,Y and EY
- "\%lg \%lg \%lg \%lg" read only 4 first columns into X,Y,EX,EY

TGraphErrors类



打开html
看声明函数

```
root [0] TGraphErrors *gr=new TGraphErrors("ExampleData.txt")
root [1] gr->Draw("AP")
```

ExampleData.txt

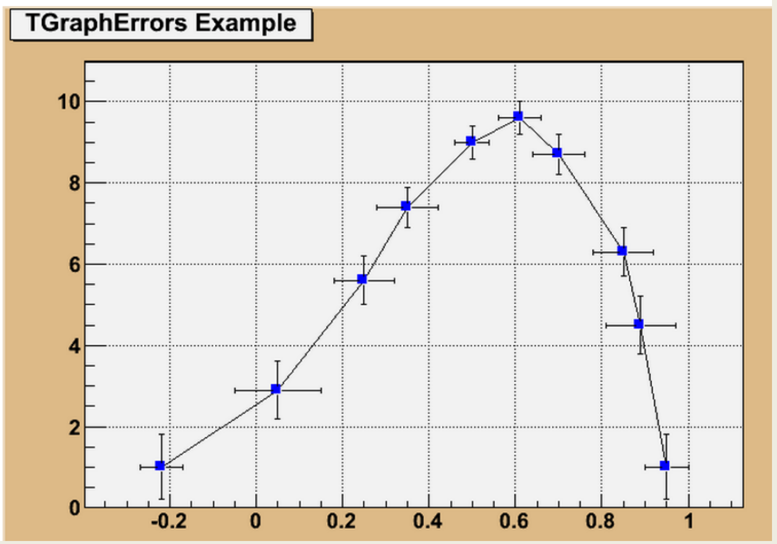
```
[] cat ExampleData.txt
# fake data to demonstrate the use of TGraphErrors

# x y ex ey
  1. 0.4 0.1 0.05
  1.3 0.3 0.05 0.1
  1.7 0.5 0.15 0.1
  1.9 0.7 0.05 0.1
  2.3 1.3 0.07 0.1
  2.9 1.5 0.2 0.1
```

```
TGraphErrors(const char* filename, const char* format = "%lg %lg %lg %lg", Option_t* option = "")
```

TGraphErrors

TGraphErrors (Int_t n, const Float_t* x, const Float_t* y, const Float_t* ex = 0, const Float_t* ey = 0)
TGraphErrors (Int_t n, const Double_t* x, const Double_t* y, const Double_t* ex = 0, const Double_t* ey = 0)

```
{
   c1 = new TCanvas("c1","A Simple Graph with error bars",200,10,700,500);
   c1->SetFillColor(42);
   c1->SetGrid();
   c1->GetFrame()->SetFillColor(21);
   c1->GetFrame()->SetBorderSize(12);
   Int_t n = 10;
   Double_t x[n]  = {-0.22, 0.05, 0.25, 0.35, 0.5, 0.61,0.7,0.85,0.89,0.95};
   Double_t y[n]  = {1,2.9,5.6,7.4,9,9.6,8.7,6.3,4.5,1};
   Double_t ex[n] = {.05,.1,.07,.07,.04,.05,.06,.07,.08,.05};
   Double_t ey[n] = {.8,.7,.6,.5,.4,.4,.5,.6,.7,.8};
   gr = new TGraphErrors(n,x,y,ex,ey);
   gr->SetTitle("TGraphErrors Example");
   gr->SetMarkerColor(4);
   gr->SetMarkerStyle(21);
   gr->Draw("ALP");
   return c1;
}
```



macro2.C

macro2_input_expected.txt

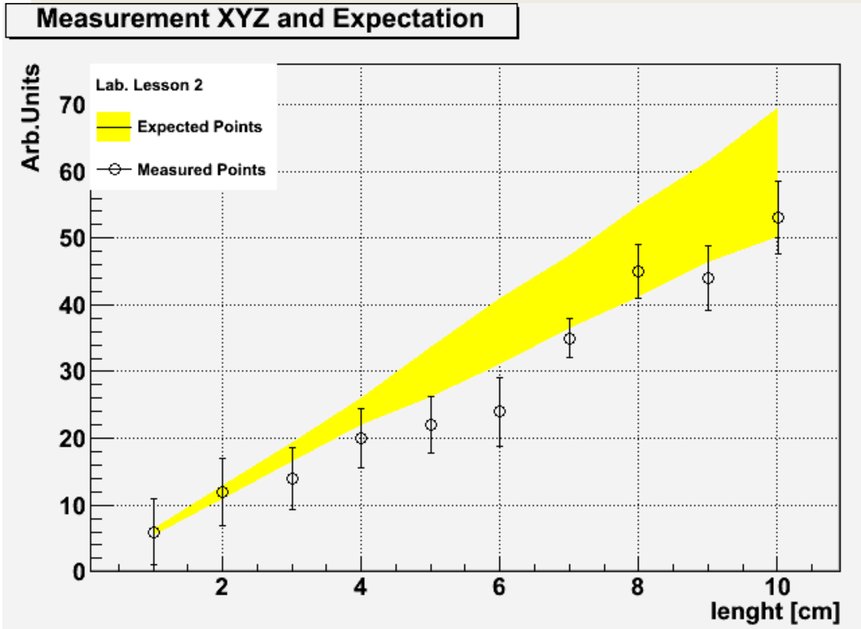
```
# Expected points from theory predictions

1   6   0.5
2   12  1.
3   18  1.5
4   24  2.0
5   30  3.7
6   36  4.9
7   42  5.4
8   48  6.8
9   54  7.5
10  60  9.7
```

macro2_input.txt

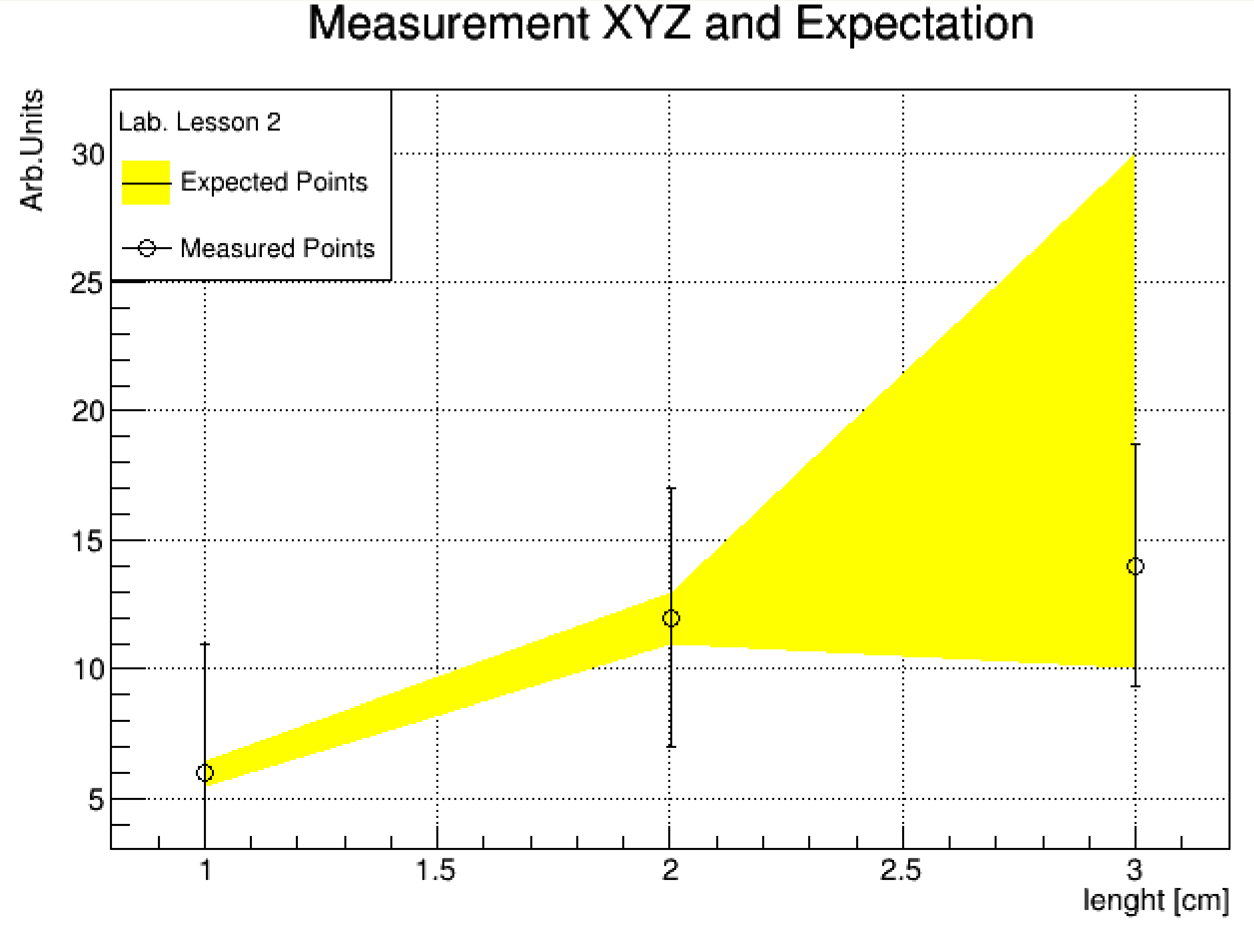
```
# Experiment 2 Physics Lab

1   6   5
2   12  5
3   14  4.7
4   20  4.5
5   22  4.2
6   24  5.1
7   35  2.9
8   45  4.1
9   44  4.8
10  53  5.43
```

marco2.C

```
// Reads the points from a file and produces a simple graph.
int macro2(){
    TCanvas* c=new TCanvas();
    c->SetGrid();

    TGraphErrors graph_expected("./macro2_input_expected.txt",
                                "%lg %lg %lg");
    graph_expected.SetTitle(
       "Measurement XYZ and Expectation;
        lenght [cm];
        Arb.Units");
    graph_expected.SetFillColor(kYellow);
    graph_expected.DrawClone("E3AL"); // E3 draws the band

    TGraphErrors graph("./macro2_input.txt","%lg %lg %lg");
    graph.SetMarkerStyle(kCircle);
    graph.SetFillColor(0);
    graph.DrawClone("PESame");

    // Draw the Legend
    TLegend leg(.1,.7,.3,.9,"Lab. Lesson 2");
    leg.SetFillColor(0);
    leg.AddEntry(&graph_expected,"Expected Points");
    leg.AddEntry(&graph,"Measured Points");
    leg.DrawClone("Same");

    graph.Print();
}
```



改变输入文件查看影响




```
# Expected points from theory predictions
#Yellow band

1   6   0.5
2   12  1.
3   20  10
```

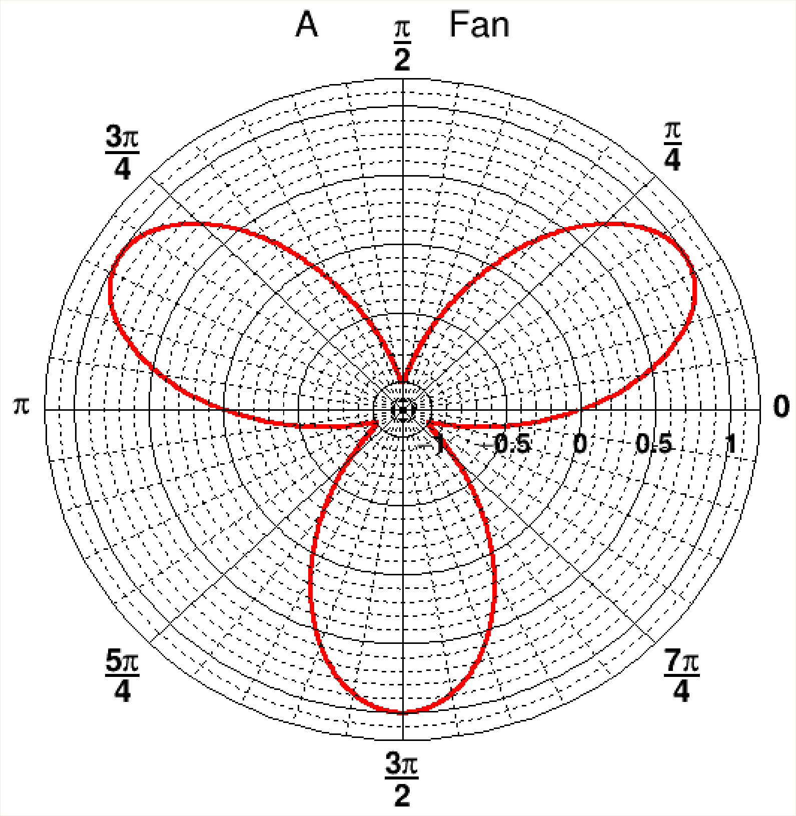
结论：

1、黄带的宽度由Error控制（第3列）

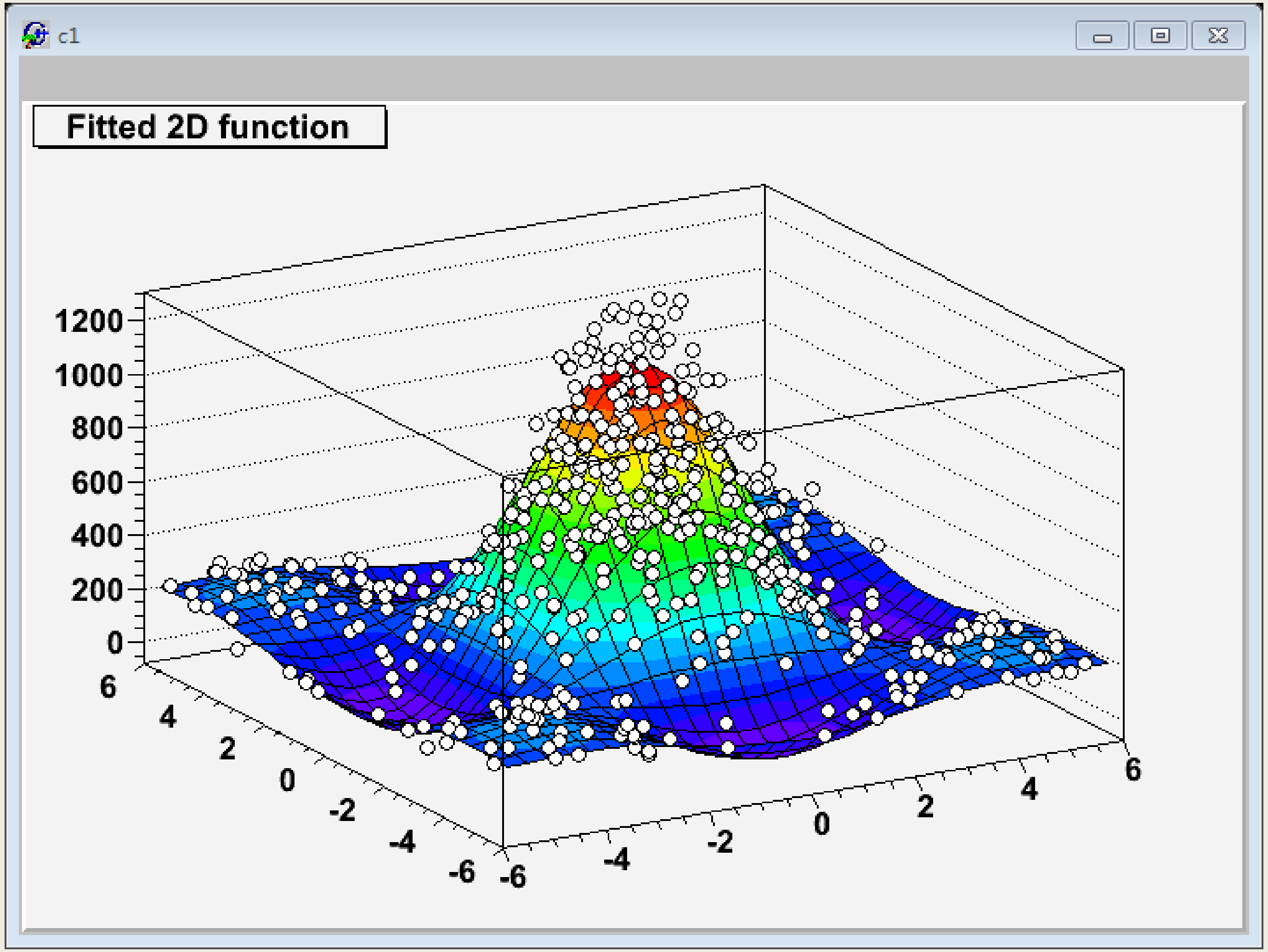
2、X轴、Y轴的区间由第一次画图的数据控制,其中DrawClone(“E3AL”)的选项中“A”启动计算坐标轴区间

极化图（Polar Graph）

```
// Builds a polar graph in a square Canvas.

void macro3(){
    TCanvas* c = new TCanvas("myCanvas","myCanvas",600,600);
    double Theta_min=0;
    double Theta_max=TMath::Pi()*6;
    const int npoints=1000;
    Double_t r[npoints];
    Double_t theta[npoints];
    for (Int_t ipt = 0; ipt < npoints; ipt++) {
        theta[ipt] = ipt*(Theta_max-Theta_min)/npoints+Theta_min;
        r[ipt] = TMath::Sin(theta[ipt]);
    }
    TGraphPolar grP1 (npoints,theta,r);
    grP1.SetTitle("A             Fan");
    grP1.SetLineWidth(3);
    grP1.SetLineColor(2);
    grP1.DrawClone("AL");
}
```



2D Graphs

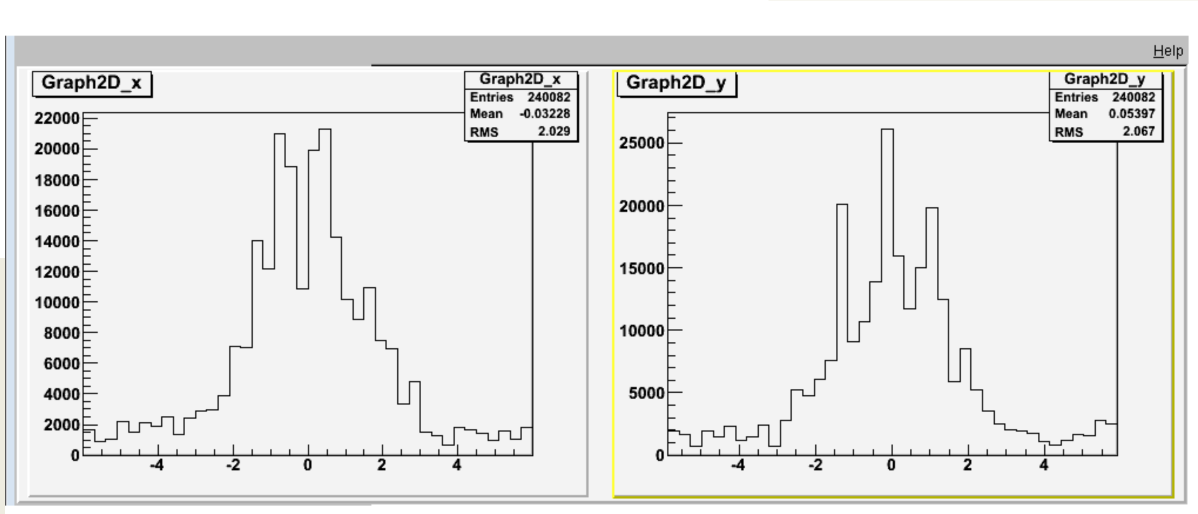

c1
Fitted 2D function
1200
1000
800
600
400
200
0
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

macro4.C

```
// Create, Draw and fit a TGraph2DErrors
void macro4(){
   gStyle->SetPalette(1);
   const double e = 0.3;
   const int nd = 500;

   TRandom3 my_random_generator;
   TF2 *f2 = new TF2("f2",
                    "1000*(([0]*sin(x)/x)*([1]*sin(y)/y))+200",
                    -6,6,-6,6);
   f2->SetParameters(1,1);
   TGraph2DErrors *dte = new TGraph2DErrors(nd);
   // Fill the 2D graph
   double rnd, x, y, z, ex, ey, ez;
   for (Int_t i=0; i<nd; i++) {
      f2->GetRandom2(x,y);
      // A random number in [-e,e]
      rnd = my_random_generator.Uniform(-e,e);
      z = f2->Eval(x,y)*(1+rnd);
      dte->SetPoint(i,x,y,z);
      ex = 0.05*my_random_generator.Uniform();
      ey = 0.05*my_random_generator.Uniform();
      ez = TMath::Abs(z*rnd);
      dte->SetPointError(i,ex,ey,ez);
   }
   // Fit function to generated data
   f2->SetParameters(0.7,1.5);  // set initial values for fit
   f2->SetTitle("Fitted 2D function");
   dte->Fit(f2);
```

macro4.C [continue]

```
   // Plot the result
   TCanvas *c1 = new TCanvas();
   f2->Draw("Surf1");
   dte->Draw("P0 Same");
   // Make the x and y projections
   TCanvas* c_p= new TCanvas("ProjCan",
                             "The Projections",1000,400);
   c_p->Divide(2,1);
   c_p->cd(1);
   dte->Project("x")->Draw();
   c_p->cd(2);
   dte->Project("y")->Draw();
}
```

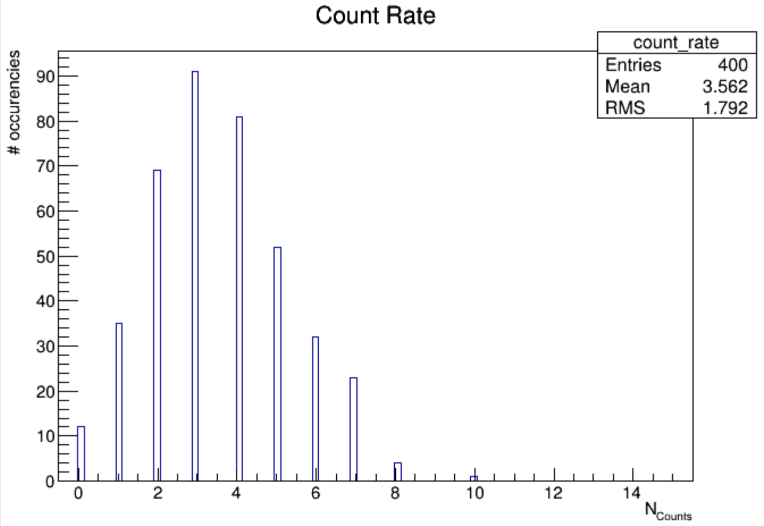


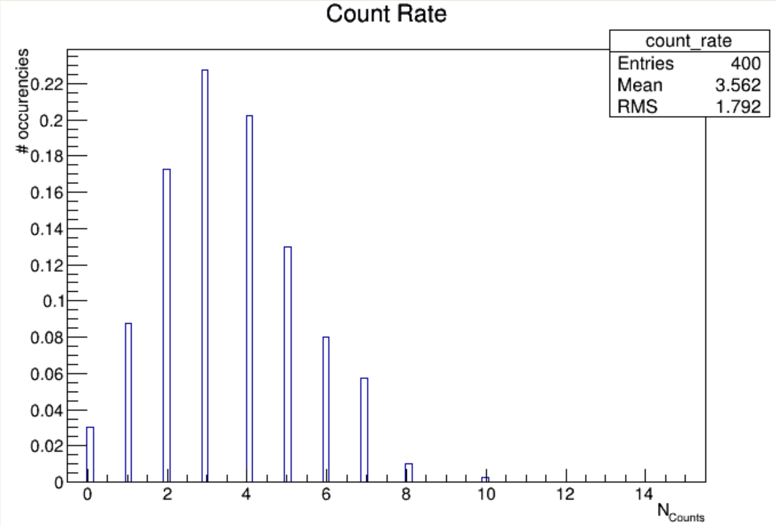
直方图的操作

王思广
北京大学物理学院

直方图归一

```
// Create, Fill and draw an Histogram which reproduces the
// counts of a scaler linked to a Geiger counter.

void macro5(){
    TH1F* cnt_r_h=new TH1F("count_rate",
                "Count Rate;N_{Counts};# occurencies",
                100, // Number of Bins
                -0.5, // Lower X Boundary
                15.5); // Upper X Boundary

    const float mean_count=3.6;
    TRandom3 rndgen;
    // simulate the measurements
    for (int imeas=0;imeas<400;imeas++)
        cnt_r_h->Fill(rndgen.Poisson(mean_count));

    TCanvas* c= new TCanvas();
    cnt_r_h->Draw();

    TCanvas* c_norm= new TCanvas();
    cnt_r_h->DrawNormalized();

    // Print summary
    cout << "Moments of Distribution:\n"
         << " - Mean = " << cnt_r_h->GetMean() << " +- "
                         << cnt_r_h->GetMeanError() << "\n"
         << " - RMS = " << cnt_r_h->GetRMS() << " +- "
                        << cnt_r_h->GetRMSError() << "\n"
         << " - Skewness = " << cnt_r_h->GetSkewness() << "\n"
         << " - Kurtosis = " << cnt_r_h->GetKurtosis() << "\n";
}
```



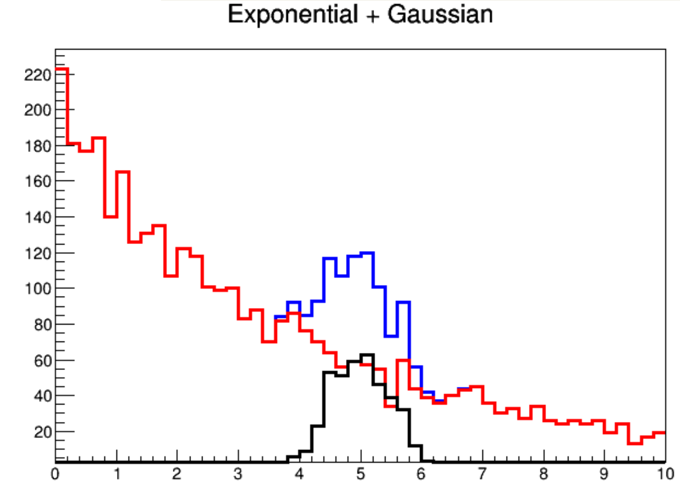


直方图的加和除

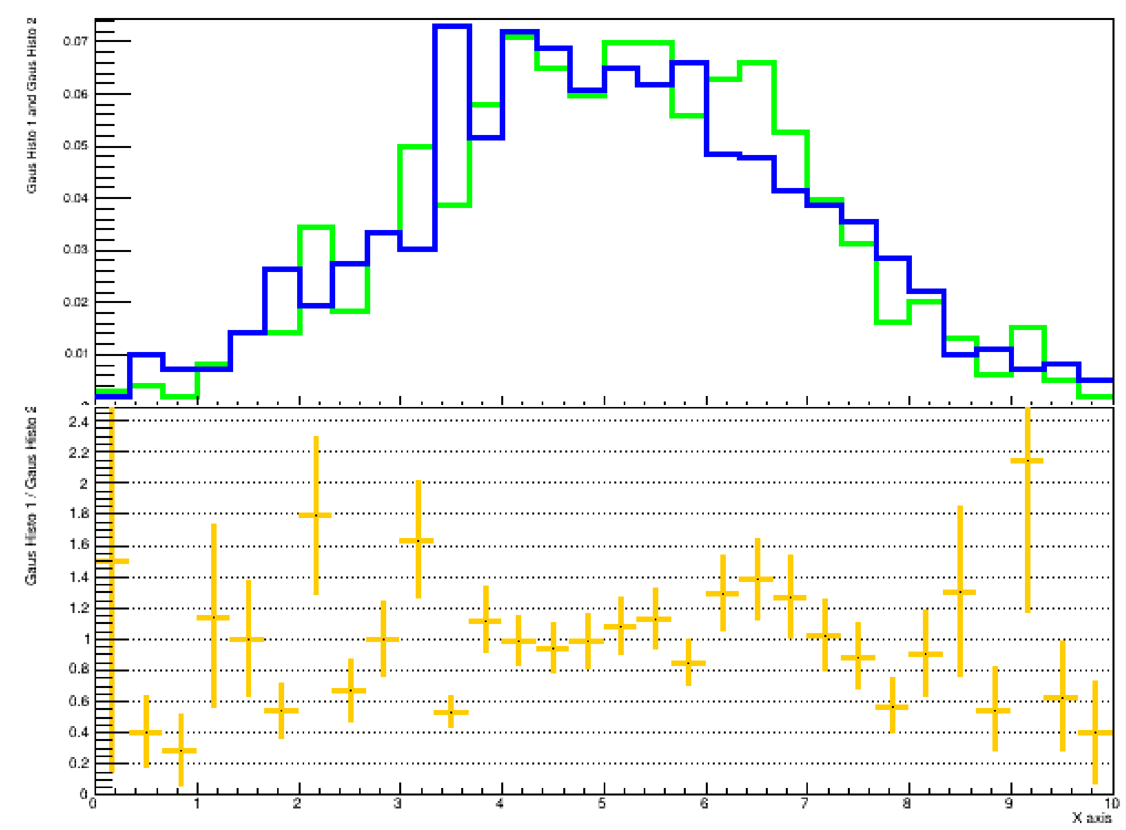
```
// Divide and add 1D Histograms

void format_h(TH1F* h, int linecolor){
  h->SetLineWidth(3);
  h->SetLineColor(linecolor);
}

void macro6(){

  TH1F* sig_h=new TH1F("sig_h","Signal Histo",50,0,10);
  TH1F* gaus_h1=new TH1F("gaus_h1","Gauss Histo 1",30,0,10);
  TH1F* gaus_h2=new TH1F("gaus_h2","Gauss Histo 2",30,0,10);
  TH1F* bkg_h=new TH1F("bkg_h","Exponential Histo",50,0,10);

  // simulate the measurements
  TRandom3 rndgen;
  for (int imeas=0;imeas<4000;imeas++){
    bkg_h->Fill(rndgen.Exp(4));
    if (imeas%4==0) gaus_h1->Fill(rndgen.Gaus(5,2));
    if (imeas%4==0) gaus_h2->Fill(rndgen.Gaus(5,2));
    if (imeas%10==0)sig_h->Fill(rndgen.Gaus(5,.5));
  }
```

```
// Format Histograms
TH1F* histos[4]={sig_h,bkg_h,gaus_h1,gaus_h2};
for (int i=0;i<4;++i){
  histos[i]->Sumw2(); // *Very* Important
  format_h(histos[i],i+1);
}

// Sum
TH1F* sum_h= new TH1F(*bkg_h);
sum_h->Add(sig_h,1.);
sum_h->SetTitle("Exponential + Gaussian");
format_h(sum_h,kBlue);
```

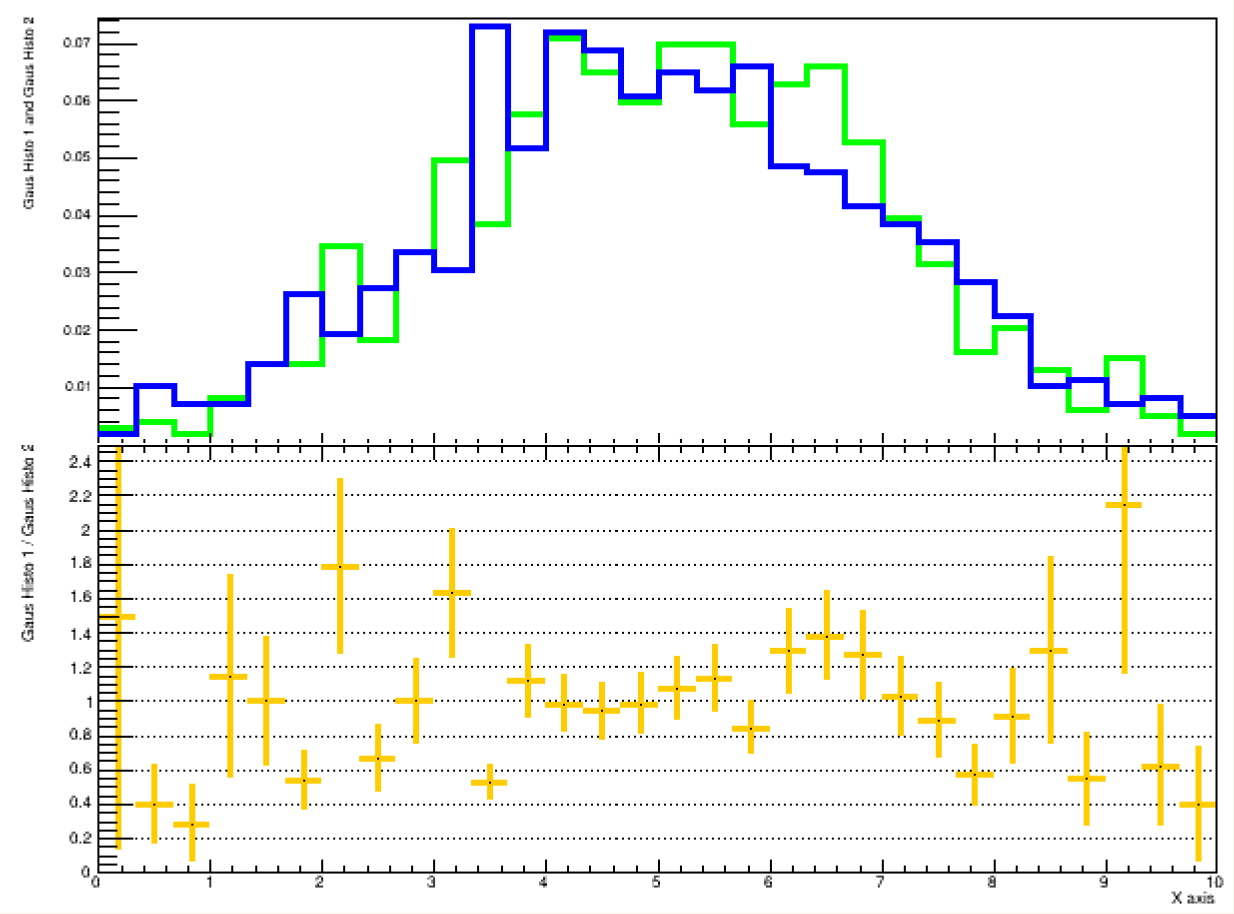
```
TCanvas* c_sum= new TCanvas();
sum_h->Draw("hist");
bkg_h->Draw("SameHist");
sig_h->Draw("SameHist");

// Divide
TH1F* dividend=new TH1F(*gaus_h1);
dividend->Divide(gaus_h2);

// Graphical Maquillage
dividend->SetTitle(";X axis;Gaus Histo 1 / Gaus Histo 2");
format_h(dividend,kOrange);
gaus_h1->SetTitle(";;Gaus Histo 1 and Gaus Histo 2");
gStyle->SetOptStat(0);
```



```
  TCanvas* c_divide= new TCanvas();
  c_divide->Divide(1,2,0,0);
  c_divide->cd(1);
  c_divide->GetPad(1)->SetRightMargin(.01);
  gaus_h1->DrawNormalized("Hist");
  gaus_h2->DrawNormalized("HistSame");

  c_divide->cd(2);
  dividend->GetYaxis()->SetRangeUser(0,2.49);
  c_divide->GetPad(2)->SetGridy();
  c_divide->GetPad(2)->SetRightMargin(.01);
  dividend->Draw();
}
```



```
  TCanvas* c_divide= new TCanvas();
  c_divide->Divide(1,2,0,0);
  c_divide->cd(1);
  c_divide->GetPad(1)->SetRightMargin(.01);
  gaus_h1->SetMinimum(0.001); //added for removing half 0
  gaus_h1->DrawNormalized("Hist");
  gaus_h2->DrawNormalized("HistSame");
  c_divide->cd(2);
  dividend->GetYaxis()->SetRangeUser(0,2.49);
  c_divide->GetPad(2)->SetGridy();
  c_divide->GetPad(2)->SetRightMargin(.01);
  gPad->SetTickx(1); //for the X-axis tick
  dividend->Draw();
}
```

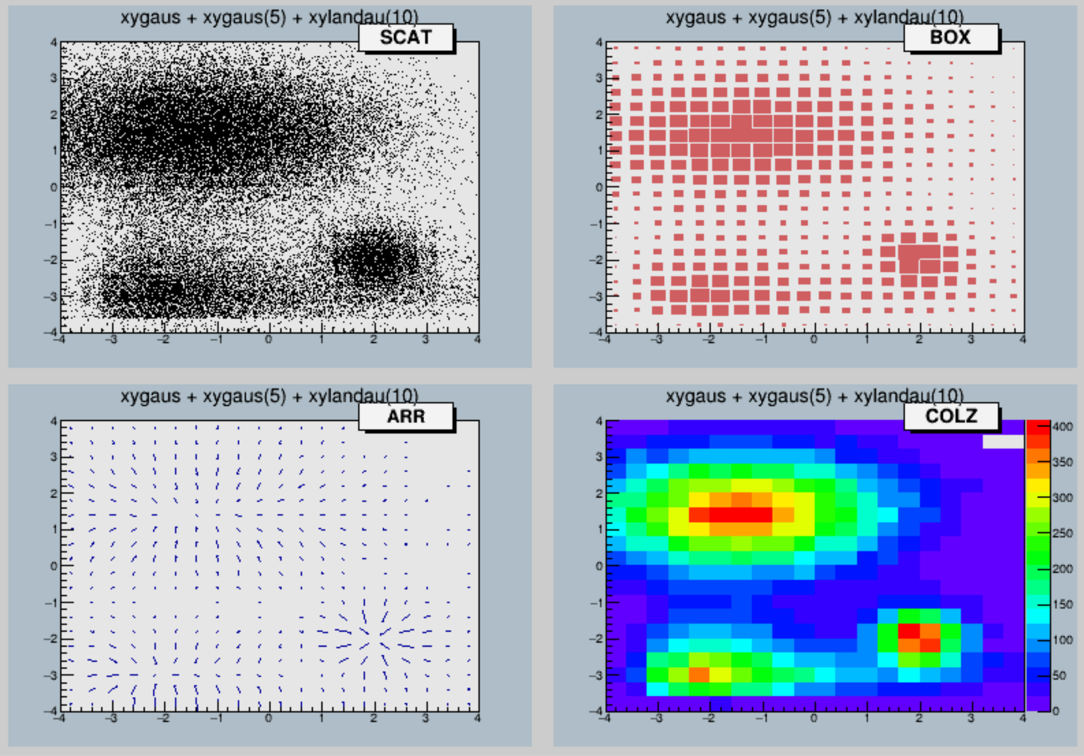


2D直方图画图选项($ROOTSYS/tutorials/hist/draw2dopt.C)

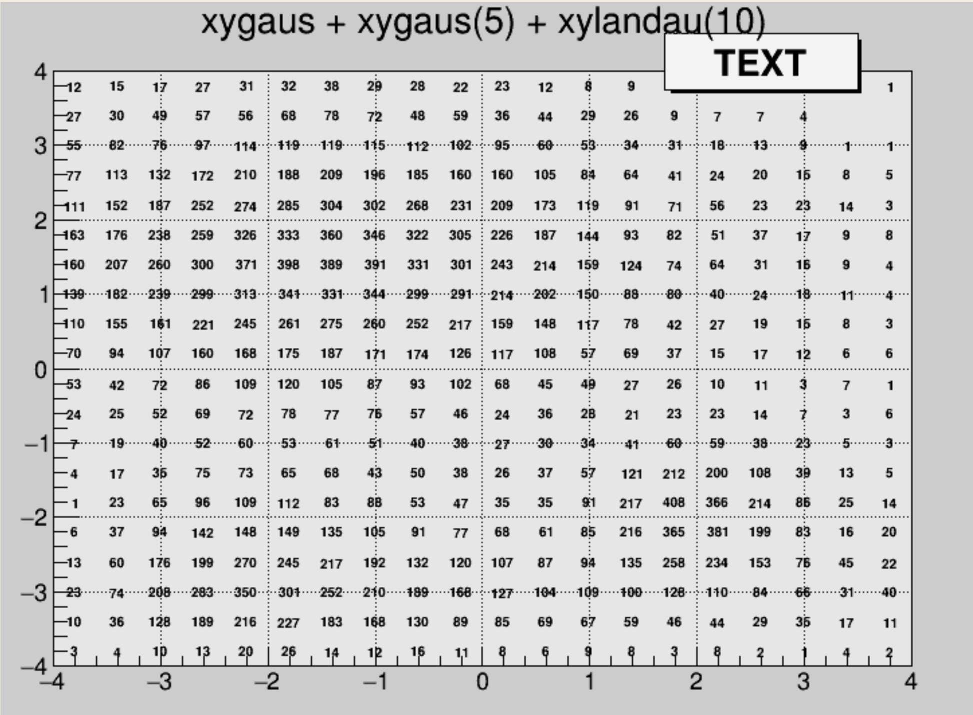
```
{
   // display the various 2-d drawing options
   //Author: Rene Brun

   gROOT->Reset();
   gStyle->SetOptStat(0);
   gStyle->SetPalette(1);
   gStyle->SetCanvasColor(33);
   gStyle->SetFrameFillColor(18);
   TF2 *f2 = new TF2("f2","xygaus + xygaus(5) + xylandau(10)",-4,4,-4,4);
   Double_t params[] = {130,-1.4,1.8,1.5,1, 150,2,0.5,-2,0.5, 3600,-2,0.7,-3,0.3};
   f2.SetParameters(params);
   TH2F h2("h2","xygaus + xygaus(5) + xylandau(10)",20,-4,4,20,-4,4);
   h2.SetFillColor(46);
   h2.FillRandom("f2",40000);
   TPaveLabel pl;

   //basic 2-d options
   Float_t x1=0.67, y1=0.875, x2=0.85, y2=0.95;
   Int_t cancolor = 17;
   TCanvas c2h("c2h","2-d options",10,10,800,600);
   c2h.Divide(2,2);
   c2h.SetFillColor(cancolor);
```

siguang@pku.edu.cn

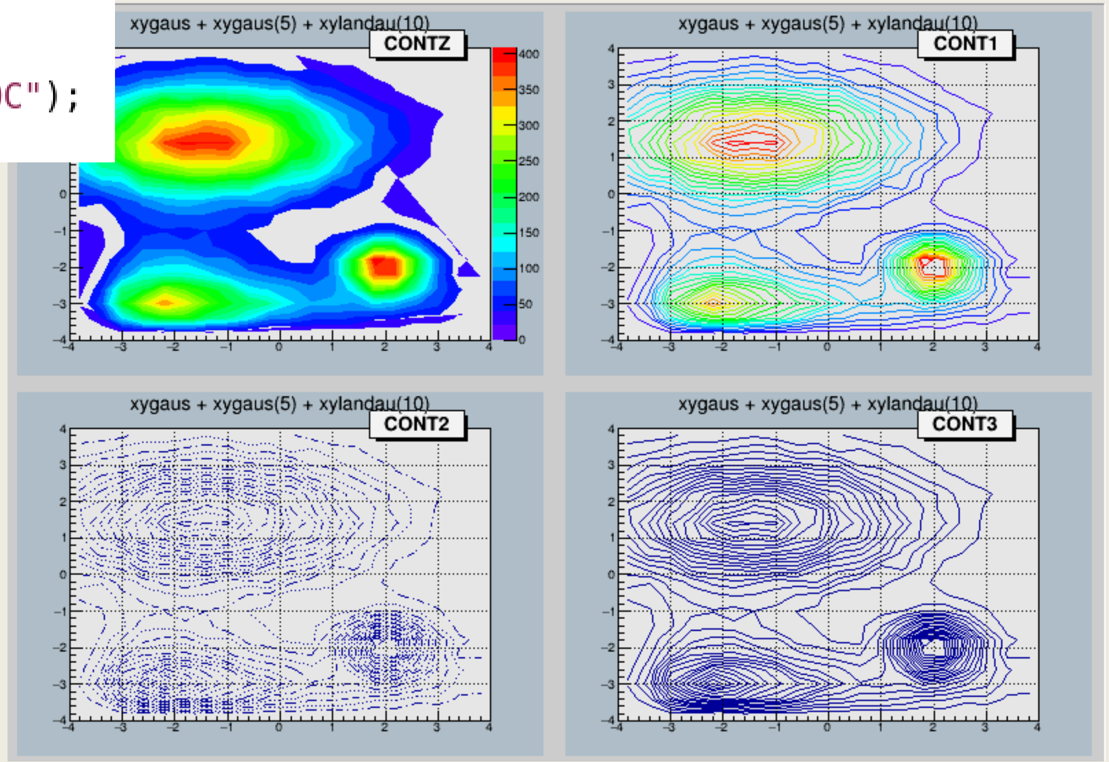
```
c2h.cd(1);
h2.Draw();         pl.DrawPaveLabel(x1,y1,x2,y2,"SCAT","brNDC");
c2h.cd(2);
h2.Draw("box");   pl.DrawPaveLabel(x1,y1,x2,y2,"BOX","brNDC");
c2h.cd(3);
h2.Draw("arr");   pl.DrawPaveLabel(x1,y1,x2,y2,"ARR","brNDC");
c2h.cd(4);
h2.Draw("colz"); pl.DrawPaveLabel(x1,y1,x2,y2,"COLZ","brNDC");
c2h.Update();
```



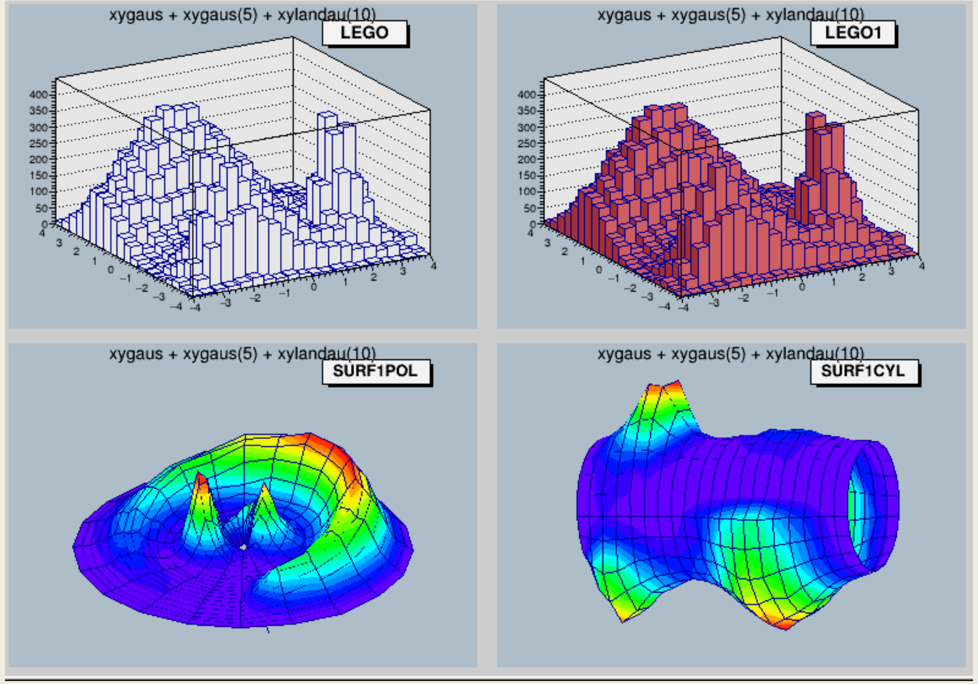
```
//text option
TCanvas ctext("ctext","text option",50,50,800,600);
gPad->SetGrid();
ctext.SetFillColor(cancolor);
ctext->SetGrid();
h2.Draw("text"); pl.DrawPaveLabel(x1,y1,x2,y2,"TEXT","brNDC");
ctext.Update();
```



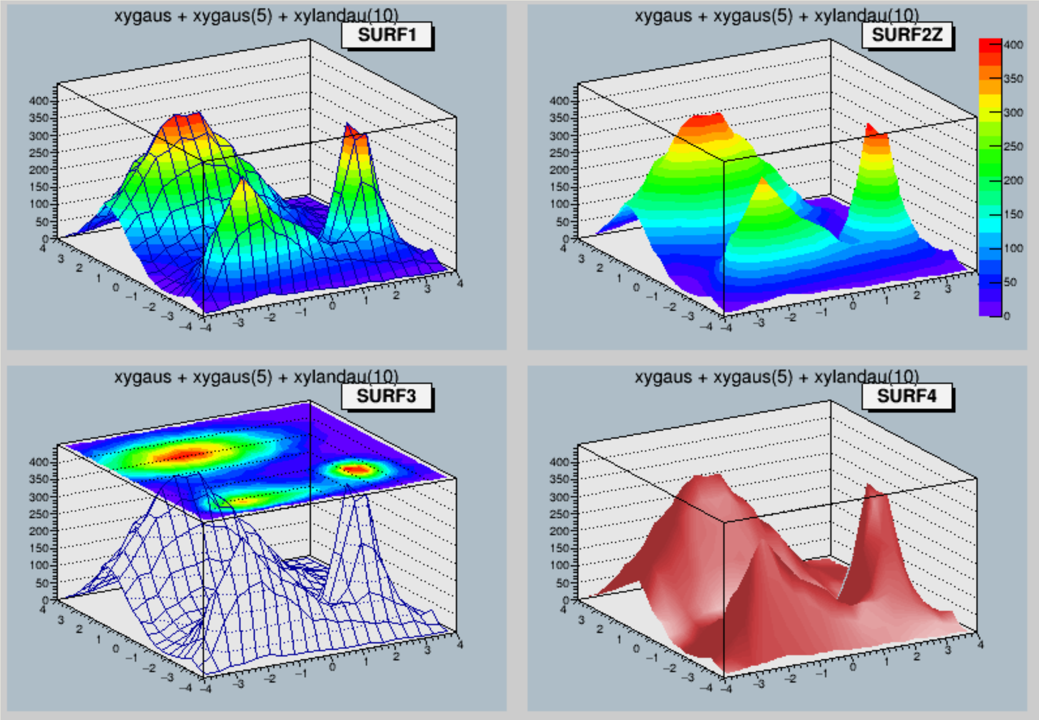
```
//contour options
TCanvas cont("contours","contours",100,100,800,600);
cont.Divide(2,2);
gPad->SetGrid();
cont.SetFillColor(cancolor);
cont.cd(1);
h2.Draw("contz"); pl.DrawPaveLabel(x1,y1,x2,y2,"CONTZ","brNDC");
cont.cd(2);
gPad->SetGrid();
h2.Draw("cont1"); pl.DrawPaveLabel(x1,y1,x2,y2,"CONT1","brNDC");
cont.cd(3);
gPad->SetGrid();
h2.Draw("cont2"); pl.DrawPaveLabel(x1,y1,x2,y2,"CONT2","brNDC");
cont.cd(4);
gPad->SetGrid();
h2.Draw("cont3"); pl.DrawPaveLabel(x1,y1,x2,y2,"CONT3","brNDC");
cont.Update();
```



```
//lego options
TCanvas lego("lego","lego options",150,150,800,600);
lego.Divide(2,2);
lego.SetFillColor(cancolor);
lego.cd(1);
h2.Draw("lego");     pl.DrawPaveLabel(x1,y1,x2,y2,"LEGO","brNDC");
lego.cd(2);
h2.Draw("lego1");    pl.DrawPaveLabel(x1,y1,x2,y2,"LEGO1","brNDC");
lego.cd(3);
gPad->SetTheta(61); gPad->SetPhi(-82);
h2.Draw("surf1pol"); pl.DrawPaveLabel(x1,y1,x2+0.05,y2,"SURF1POL","brNDC");
lego.cd(4);
gPad->SetTheta(21); gPad->SetPhi(-90);
h2.Draw("surf1cyl"); pl.DrawPaveLabel(x1,y1,x2+0.05,y2,"SURF1CYL","brNDC");
lego.Update();
```

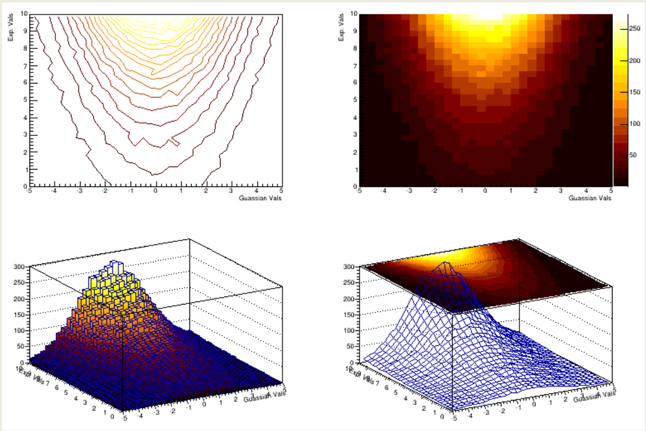


```
//surface options
TCanvas surf("surfopt","surface options",200,200,800,600);
surf.Divide(2,2);
surf.SetFillColor(cancolor);
surf.cd(1);
h2.Draw("surf1");   pl.DrawPaveLabel(x1,y1,x2,y2,"SURF1","brNDC");
surf.cd(2);
h2.Draw("surf2z");  pl.DrawPaveLabel(x1,y1,x2,y2,"SURF2Z","brNDC");
surf.cd(3);
h2.Draw("surf3");   pl.DrawPaveLabel(x1,y1,x2,y2,"SURF3","brNDC");
surf.cd(4);
h2.Draw("surf4");   pl.DrawPaveLabel(x1,y1,x2,y2,"SURF4","brNDC");
surf.Update();
```

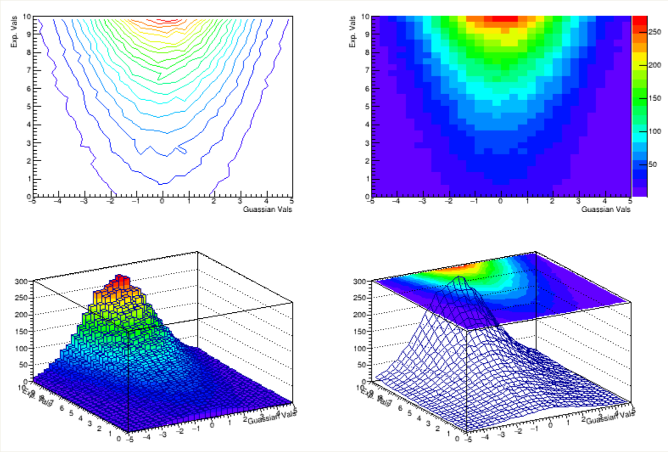


二维直方图

```
// Draw a Bidimensional Histogram in many ways
// together with its profiles and projections

void macro7(){
    gStyle->SetPalette(53);
    gStyle->SetOptStat(0);
    gStyle->SetOptTitle(0);

    TH2F bidi_h("bidi_h","2D Histo;Guassian Vals;Exp. Vals",
                30,-5,5,  // X axis
                30,0,10); // Y axis

    TRandom3 rgen;
    for (int i=0;i<500000;i++)
        bidi_h.Fill(rgen.Gaus(0,2),10-rgen.Exp(4),.1);

    TCanvas* c=new TCanvas("Canvas","Canvas",800,800);
    c->Divide(2,2);
    c->cd(1);bidi_h.DrawClone("Cont1");
    c->cd(2);bidi_h.DrawClone("Colz");
    c->cd(3);bidi_h.DrawClone("lego2");
    c->cd(4);bidi_h.DrawClone("surf3");

    // Profiles and Projections
    TCanvas* c2=new TCanvas("Canvas2","Canvas2",800,800);
    c2->Divide(2,2);
    c2->cd(1);bidi_h.ProjectionX()->DrawClone();
    c2->cd(2);bidi_h.ProjectionY()->DrawClone();
    c2->cd(3);bidi_h.ProfileX()->DrawClone();
    c2->cd(4);bidi_h.ProfileY()->DrawClone();
}
```

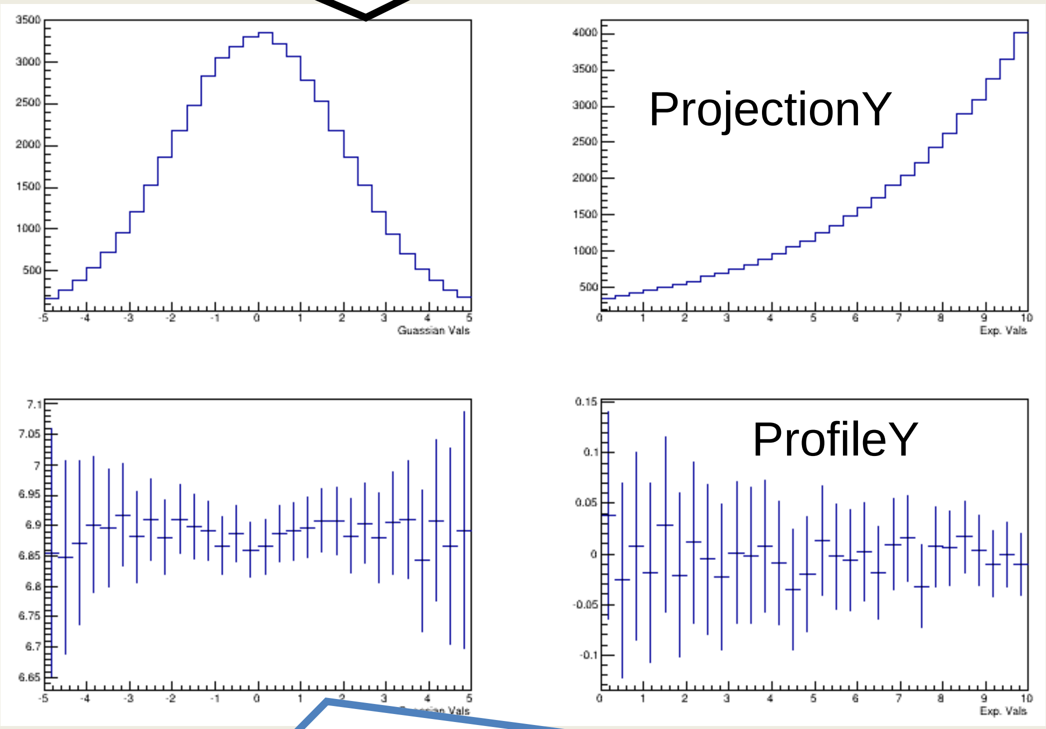


SetPalette(53)



SetPalette(1)

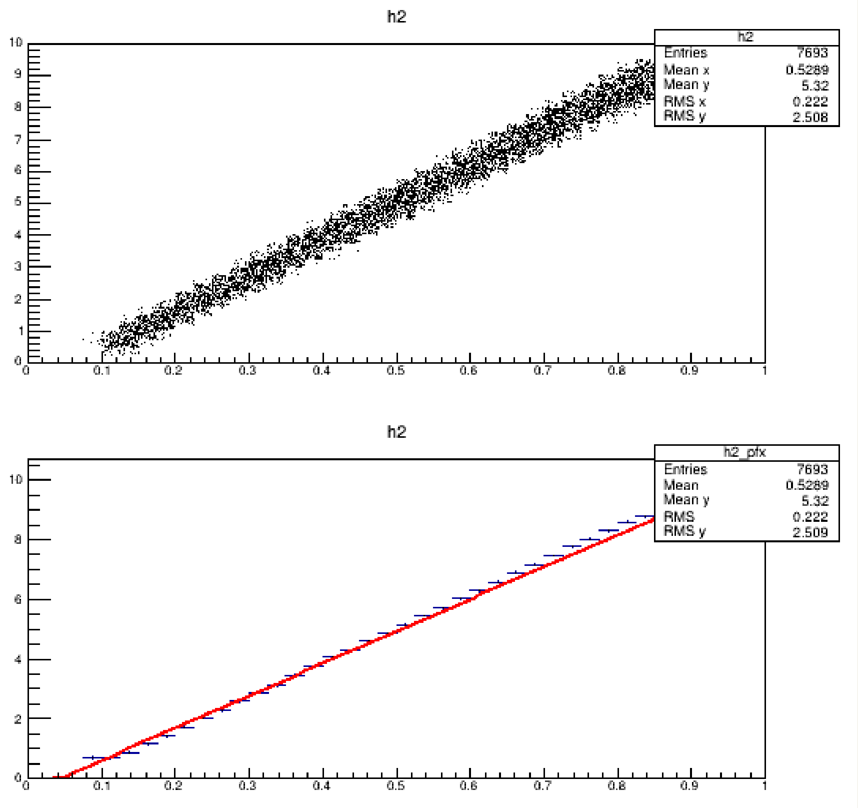
ProjectionX
相当于同Xbin不同Ybin的bin中的计数Z相加



ProfileX
相当于同Xbin不同Ybin的以bin的计数Z为权重将Ybin的中心Y值相加

$ROOTSYS/tutorials/fit/fit2d.C

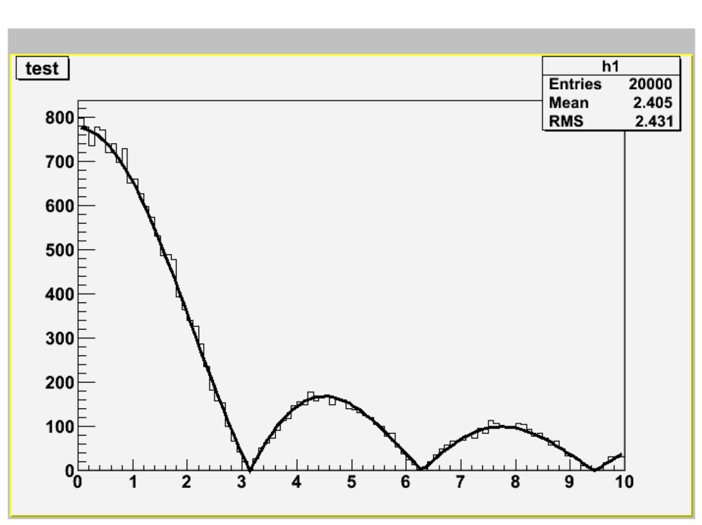
```
void fit2d()
{
   //example illustrating how to fit a 2-d histogram of type y=f(x)
   //Author: Rene Brun

   // generate a 2-d histogram using a TCutG
   const Int_t n = 6;
   Float_t x[n] = {0.092,0.83,0.94,0.81,0.12,0.1};
   Float_t y[n] = {0.71,9.4,9,8,0.3,0.71};
   TCutG *cut = new TCutG("cut",n,x,y);
   TH2F *h2 = new TH2F("h2","h2",40,0,1,40,0,10);
   Float_t u,v;
   for (Int_t i=0;i<100000;i++) {
      u = gRandom->Rndm();
      v = 10*gRandom->Rndm();
      if (cut->IsInside(u,v)) h2->Fill(u,v);
   }
   TCanvas *c1 = new TCanvas("c1","show profile",600,900);
   c1->Divide(1,2);
   c1->cd(1);
   h2->Draw();
   c1->cd(2);

   //use a TProfile to convert the 2-d to 1-d problem
   TProfile *prof = h2->ProfileX();
   prof->Fit("pol1");
}
```



自定义函数进行拟合的例1

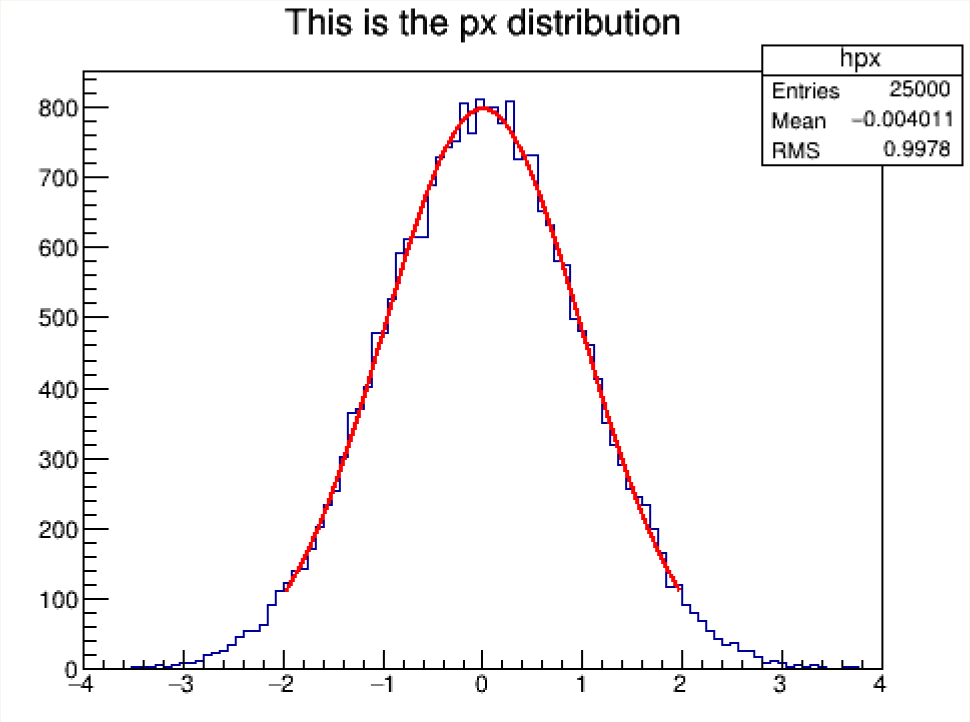
```
#include <TF1.h>
#include <TH1F.h>
#include <TROOT.h>

 // Macro myfunc.C
   Double_t myfunction(Double_t *x, Double_t *par)
   {
      Float_t xx =x[0];
      Double_t f = TMath::Abs(par[0]*sin(par[1]*xx)/xx);
      return f;
   }
   void myfunc()
   {
      TF1 *f1 = new TF1("myfunc",myfunction,0,10,2);
      f1->SetParameters(2,1);
      f1->SetParNames("constant","coefficient");
      f1->Draw();
   }
   void myfit()
   {
      TH1F *h1=new TH1F("h1","test",100,0,10);
      h1->FillRandom("myfunc",20000);
      TF1 *f1=(TF1 *)gROOT->GetFunction("myfunc");
      f1->SetParameters(800,1);
      h1->Fit("myfunc");
   }
```

root [].L Cmyfit.C
root [] myfunc()
root [] myfit()



自定义函数进行拟合的例2

$ROOTSYS/tutorials/fit/myfit.C



myfit.C

```
// Get in memory an histogram from a root file and fit a user defined function.
// Note that a user defined function must always be defined
// as in this example:
//  - first parameter: array of variables (in this example only 1-dimension)
//  - second parameter: array of parameters
// Note also that in case of user defined functions, one must set
// an initial value for each parameter.
//Author: Rene Brun

Double_t fitf(Double_t *x, Double_t *par)
{
   Double_t arg = 0;
   if (par[2] != 0) arg = (x[0] - par[1])/par[2];

   Double_t fitval = par[0]*TMath::Exp(-0.5*arg*arg);
   return fitval;
}
void myfit()
{
   TFile* hsimple = TFile::Open("../hsimple.root");
   if (!hsimple) return;
```

先运行hsimple.C文件生成root数据文件

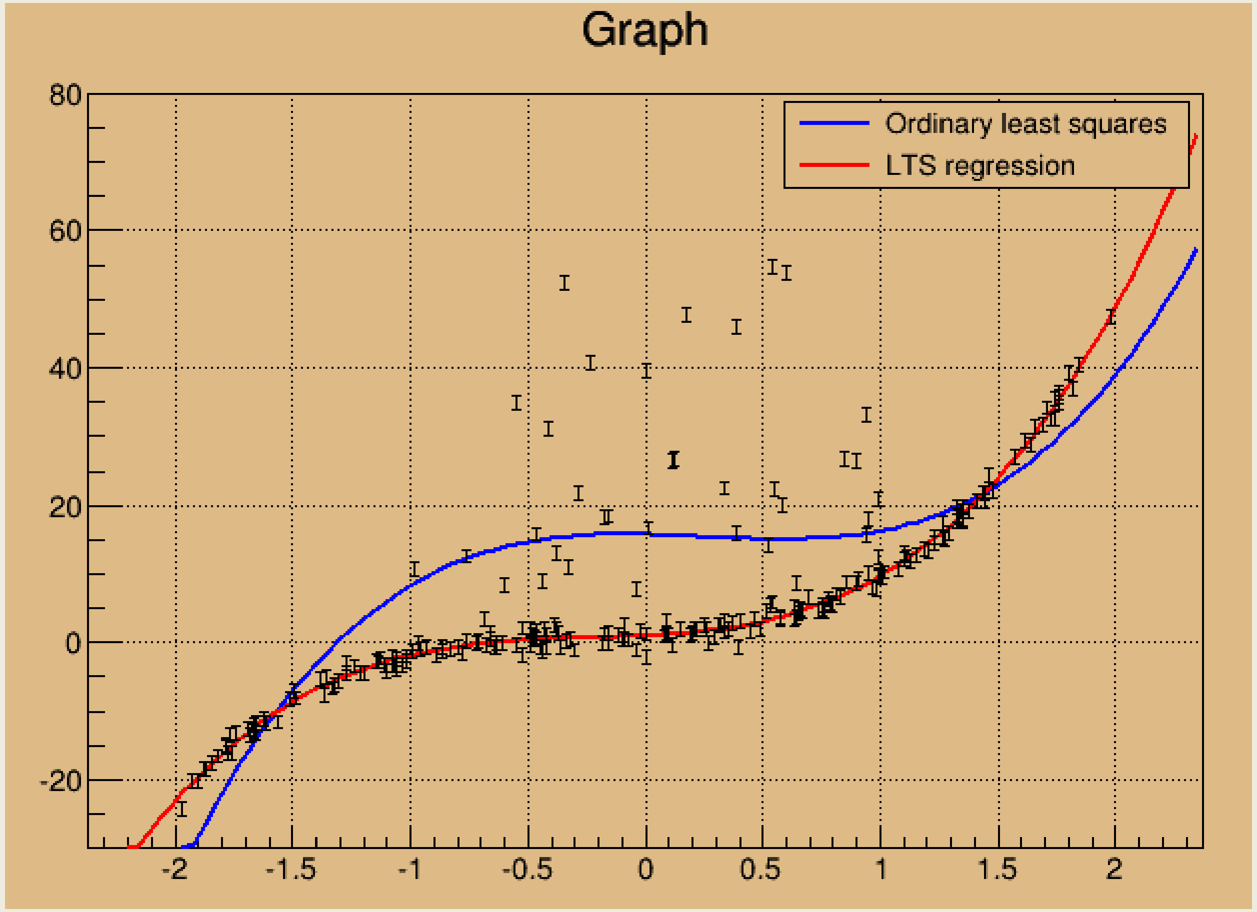
myfit.C [continue]

```
   TCanvas *c1 = new TCanvas("c1","the fit canvas",500,400);

   TH1F *hpx = (TH1F*)hsimple->Get("hpx");

// Creates a Root function based on function fitf above
   TF1 *func = new TF1("fitf",fitf,-2,2,3);

// Sets initial values and parameter names
   func->SetParameters(100,0,1);
   func->SetParNames("Constant","Mean_value","Sigma");

// Fit histogram in range defined by function
   hpx->Fit(func,"r");

// Gets integral of function between fit limits
   printf("Integral of function = %g\n",func->Integral(-2,2));
}
```

限定区间范围

r 限定区间范围拟合

$ROOTSYS/tutorials/fit/fitLinearRobust.C



$ROOTSYS/tutorials/fit/fitLinearRobust.C

```
#include "TRandom.h"
#include "TGraphErrors.h"
#include "TF1.h"
#include "TCanvas.h"
#include "TLegend.h"

void fitLinearRobust()
{
//This tutorial shows how the least trimmed squares regression,
//included in the TLinearFitter class, can be used for fitting
//in cases when the data contains outliers.
//Here the fitting is done via the TGraph::Fit function with option "rob":
//If you want to use the linear fitter directly for computing
//the robust fitting coefficients, just use the TLinearFitter::EvalRobust
//function instead of TLinearFitter::Eval
//Author: Anna Kreshuk

   //First generate a dataset, where 20% of points are spoiled by large
   //errors
   Int_t npoints = 250;
   Int_t fraction = Int_t(0.8*npoints);
   Double_t *x = new Double_t[npoints];
   Double_t *y = new Double_t[npoints];
   Double_t *e = new Double_t[npoints];
  TRandom r;
  Int_t i;
  for (i=0; i<fraction; i++){
     //the good part of the sample
     x[i]=r.Uniform(-2, 2);
     e[i]=1;
     y[i]=1 + 2*x[i] + 3*x[i]*x[i] + 4*x[i]*x[i]*x[i] + e[i]*r.Gaus();
  }
  for (i=fraction; i<npoints; i++){
     //the bad part of the sample
     x[i]=r.Uniform(-1, 1);
     e[i]=1;
     y[i] = 1 + 2*x[i] + 3*x[i]*x[i] + 4*x[i]*x[i]*x[i] + r.Landau(10, 5);
  }
```

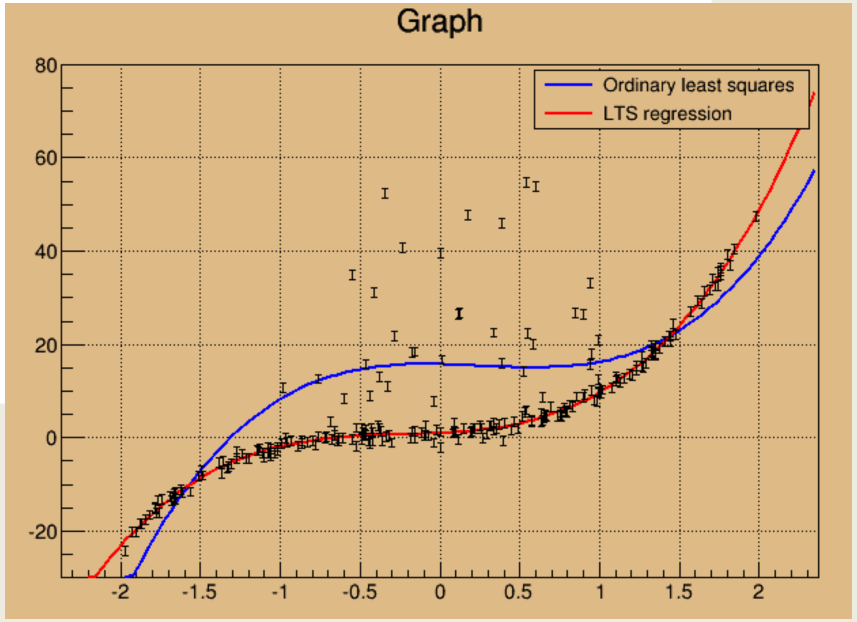
数据准备

$ROOTSYS/tutorials/fit/fitLinearRobust.C

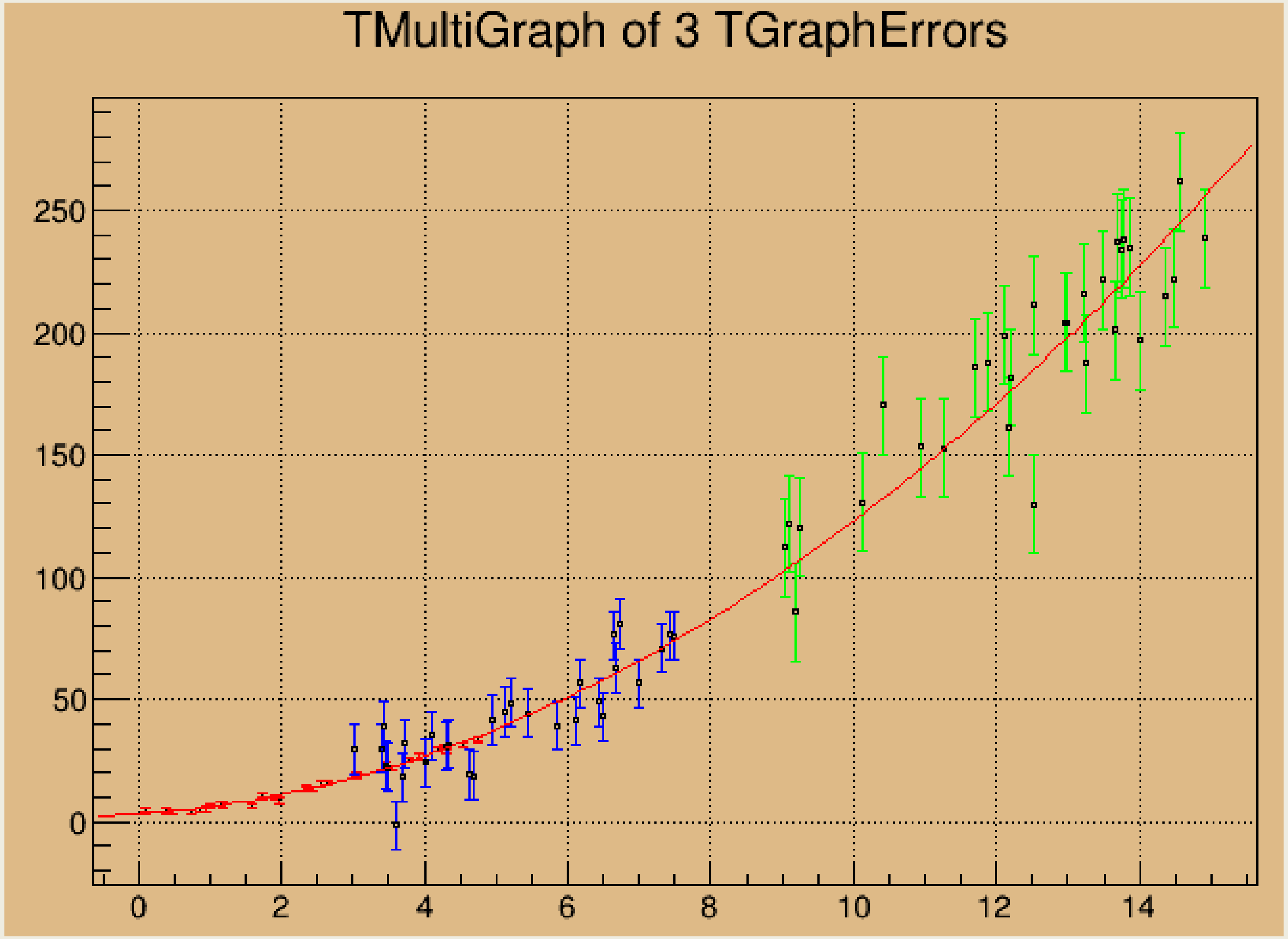
```
TGraphErrors *grr = new TGraphErrors(npoints, x, y, 0, e);
grr->SetMinimum(-30);
grr->SetMaximum(80);
TF1 *ffit1 = new TF1("ffit1", "pol3", -5, 5);
TF1 *ffit2 = new TF1("ffit2", "pol3", -5, 5);
ffit1->SetLineColor(kBlue);
ffit2->SetLineColor(kRed);
TCanvas *myc = new TCanvas("myc", "Linear and robust linear fitting");
myc->SetFillColor(42);
myc->SetGrid();
grr->Draw("ap");
//first, let's try to see the results of ordinary least-squares fit:
printf("Ordinary least squares:\n");
grr->Fit(ffit1);
//the fitted function doesn't really follow the pattern of the data
//and the coefficients are far from the real ones
```

$ROOTSYS/tutorials/fit/fitLinearRobust.C

```
printf("Resistant Least trimmed squares fit:\n");
//Now let's try the resistant regression
//The option "rob=0.75" means that we want to use robust fitting and
//we know that at least 75% of data is good points (at least 50% of points
//should be good to use this algorithm). If you don't specify any number
//and just use "rob" for the option, default value of (npoints+nparameters+1)/2
//will be taken
grr->Fit(ffit2, "+rob=0.75");
//
TLegend *leg = new TLegend(0.6, 0.8, 0.89, 0.89);
leg->AddEntry(ffit1, "Ordinary least squares", "l");
leg->AddEntry(ffit2, "LTS regression", "l");
leg->SetFillColor(42);
leg->Draw();

delete [] x;
delete [] y;
delete [] e;
```



$ROOTSYS/tutorials/fit/fitMultiGraph.C



$ROOTSYS/tutorials/fit/fitMultiGraph.C

```
void fitMultiGraph()
{
   //fitting a parabola to a multigraph of 3 partly overlapping graphs
   //with different errors
   //Author: Anna Kreshuk

   Int_t n = 30;
   Double_t *x1 = new Double_t[n];
   Double_t *x2 = new Double_t[n];
   Double_t *x3 = new Double_t[n];
   Double_t *y1 = new Double_t[n];
   Double_t *y2 = new Double_t[n];
   Double_t *y3 = new Double_t[n];
   Double_t *e1 = new Double_t[n];
   Double_t *e2 = new Double_t[n];
   Double_t *e3 = new Double_t[n];
```

$ROOTSYS/tutorials/fit/fitMultiGraph.C

```
//generate the data for the graphs
TRandom r;
Int_t i;
for (i=0; i<n; i++) {
   x1[i] = r.Uniform(0.1, 5);
   x2[i] = r.Uniform(3, 8);
   x3[i] = r.Uniform(9, 15);
   y1[i] = 3 + 2*x1[i] + x1[i]*x1[i] + r.Gaus();
   y2[i] = 3 + 2*x2[i] + x2[i]*x2[i] + r.Gaus()*10;
   e1[i] = 1;
   e2[i] = 10;
   e3[i] = 20;
   y3[i] = 3 + 2*x3[i] + x3[i]*x3[i] + r.Gaus()*20;
}

//create the graphs and set their drawing options
TGraphErrors *gr1 = new TGraphErrors(n, x1, y1, 0, e1);
TGraphErrors *gr2 = new TGraphErrors(n, x2, y2, 0, e2);
TGraphErrors *gr3 = new TGraphErrors(n, x3, y3, 0, e3);
gr1->SetLineColor(kRed);
gr2->SetLineColor(kBlue);
gr2->SetMarkerStyle(24);
gr2->SetMarkerSize(0.3);
gr3->SetLineColor(kGreen);
gr3->SetMarkerStyle(24);
gr3->SetMarkerSize(0.3);
```

$ROOTSYS/tutorials/fit/fitMultiGraph.C

```
//add the graphs to the multigraph
TMultiGraph *mg=new TMultiGraph("mg",
    "TMultiGraph of 3 TGraphErrors");
mg->Add(gr1);
mg->Add(gr2);
mg->Add(gr3);
```

$ROOTSYS/tutorials/fit/fitMultiGraph.C

```
   TCanvas *myc = new TCanvas("myc",
      "Fitting a MultiGraph of 3 TGraphErrors");
   myc->SetFillColor(42);
   myc->SetGrid();

   mg->Draw("ap");

   //fit
   mg->Fit("pol2", "F");
   //mg->Fit("pol2");

   //access to the fit function
   TF1 *fpol = mg->GetFunction("pol2");
   fpol->SetLineWidth(1);

}
```

“F” If fitting a polN, switch to minuit fitter；
By default, polN functions are fitted by the linear fitter

Fit Options

```
void TH1::Fit(const char *fname, Option_t *option, Option_t *goption, Axis_t xxmin, Axis_t xxmax)
```

*option:The second parameter is the fitting option. Here is the list of fitting options

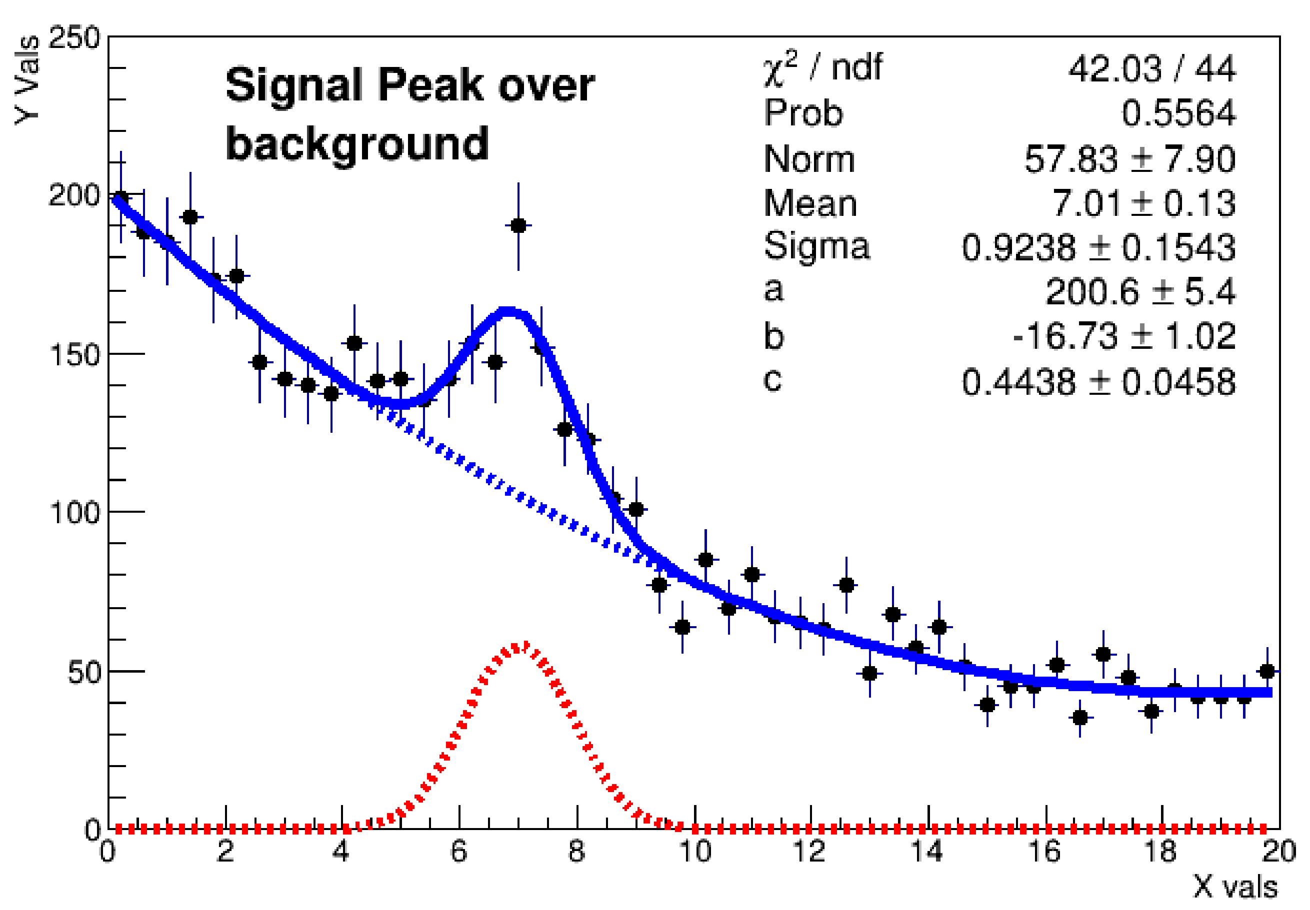
- “W” Set all weights to 1 for non empty bins; ignore error bars
- “WW” Set all weights to 1 including empty bins; ignore error bars
- “I” Use integral of function in bin instead of value at bin center
- “L” Use log likelihood method (default is chi-square method)
- “U” Use a user specified fitting algorithm
- “Q” Quiet mode (minimum printing)
- “V” Verbose mode (default is between Q and V)
- “E” Perform better errors estimation using the Minos technique
- “M” Improve fit results

- “R” Use the range specified in the function range
- “N” Do not store the graphics function, do not draw
- “0” Do not plot the result of the fit. By default the fitted function is drawn unless the option “N” above is specified.
- “+” Add this new fitted function to the list of fitted functions (by default, the previous function is deleted and only the last one is kept)
- “B”Use this option when you want to fix one or more parameters and the fitting function is like polN, expo,landau, gaus.
- “LL”An improved Log Likelihood fit in case of very low statistics and when bin contents are not integers. Do not use this option if bin contents are large (greater than 100).
- “C”In case of linear fitting, don’t calculate the chisquare (saves time).
- “F”If fitting a polN, switch to Minuit fitter (by default, polN functions are fitted by the linear fitter).

- *goption:The third parameter is the graphics option that is the same as in the **TH1**::Draw (see Draw Options).
- xxmin, xxmax:The fourth and fifth parameters specify the range over which to apply the fit.

Peak and Background Fit

Output of macro8.C （from A Root Guide For Beginners）



siguang@pku.edu.cn

macro8.C

```
void format_line(TAttLine* line,int col,int sty){
    line->SetLineWidth(5); line->SetLineColor(col);
    line->SetLineStyle(sty);}

double the_gausppar(double* vars, double* pars){
    return pars[0]*TMath::Gaus(vars[0],pars[1],pars[2])+
        pars[3]+pars[4]*vars[0]+pars[5]*vars[0]*vars[0];}

int macro8(){
    gStyle->SetOptTitle(0); gStyle->SetOptStat(0);
    gStyle->SetOptFit(1111); gStyle->SetStatBorderSize(0);
    gStyle->SetStatX(.89); gStyle->SetStatY(.89);

    TF1 parabola("parabola","[0]+[1]*x+[2]*x**2",0,20);
    format_line(&parabola,kBlue,2);

    TF1 gaussian("gaussian","[0]*TMath::Gaus(x,[1],[2])",0,20);
    format_line(&gaussian,kRed,2);
```

macro8.C [continue]

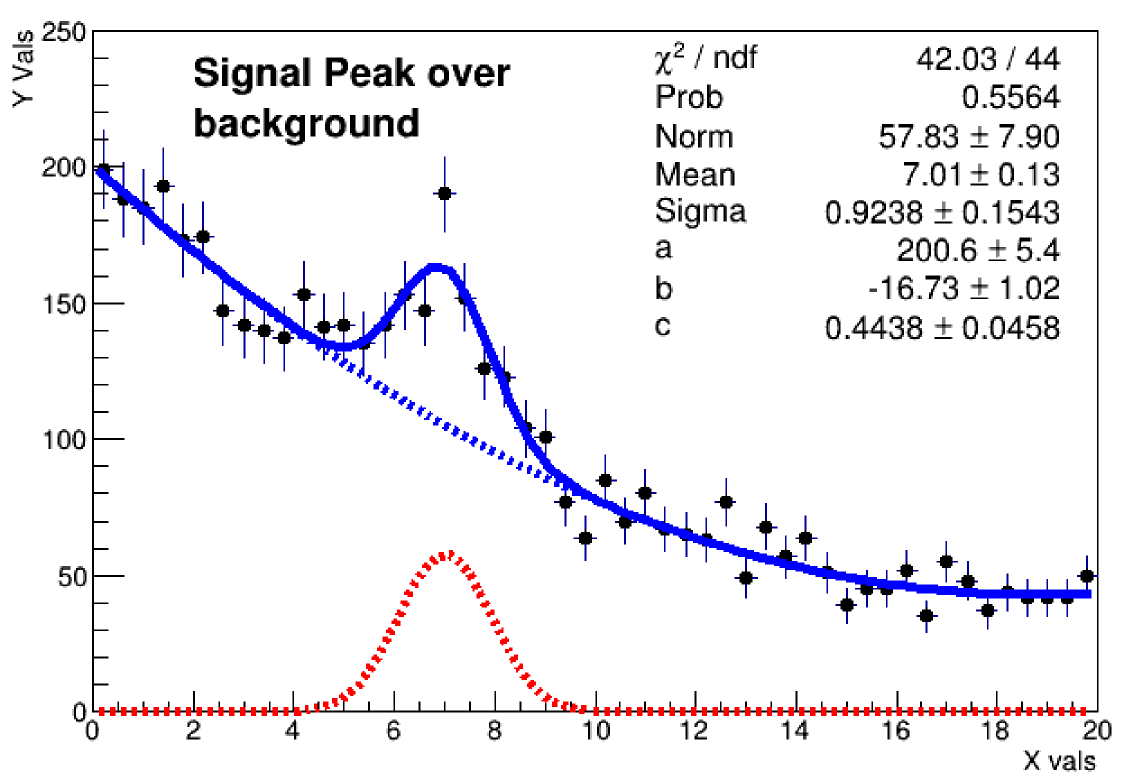
```
TF1 gausppar("gausppar",the_gausppar,-0,20,6);
double a=15; double b=-1.2; double c=.03;
double norm=4; double mean=7; double sigma=1;
gausppar.SetParameters(norm,mean,sigma,a,b,c);
gausppar.SetParNames("Norm","Mean","Sigma","a","b","c");
format_line(&gausppar,kBlue,1);

TH1F histo("histo","Signal plus background;X vals;Y Vals",
           50,0,20);
histo.SetMarkerStyle(8);

// Fake the data
for (int i=1;i<=5000;++i) histo.Fill(gausppar.GetRandom());

// Reset the parameters before the fit and set
// by eye a peak at 6 with an area of more or less 50
gausppar.SetParameter(0,50);
gausppar.SetParameter(1,6);
```

macro8.C [continue]

```
int npar=gausppar.GetNpar();
for (int ipar=2;ipar<npar;++ipar)
    gausppar.SetParameter(ipar,1);

// perform fit ...
TFitResultPtr frp = histo.Fit(&gausppar, "S");

// ... and retrieve fit results
frp->Print(); // print fit results
// get covariance Matrix an print it
TMatrixDSym covMatrix (frp->GetCovarianceMatrix());
covMatrix.Print();

// Set the values of the gaussian and parabola
for (int ipar=0;ipar<3;ipar++){
    gaussian.SetParameter(ipar,
                          gausppar.GetParameter(ipar));
    parabola.SetParameter(ipar,
                          gausppar.GetParameter(ipar+3));}

histo.GetYaxis()->SetRangeUser(0,250);
histo.DrawClone("PE");
parabola.DrawClone("Same"); gaussian.DrawClone("Same");
```

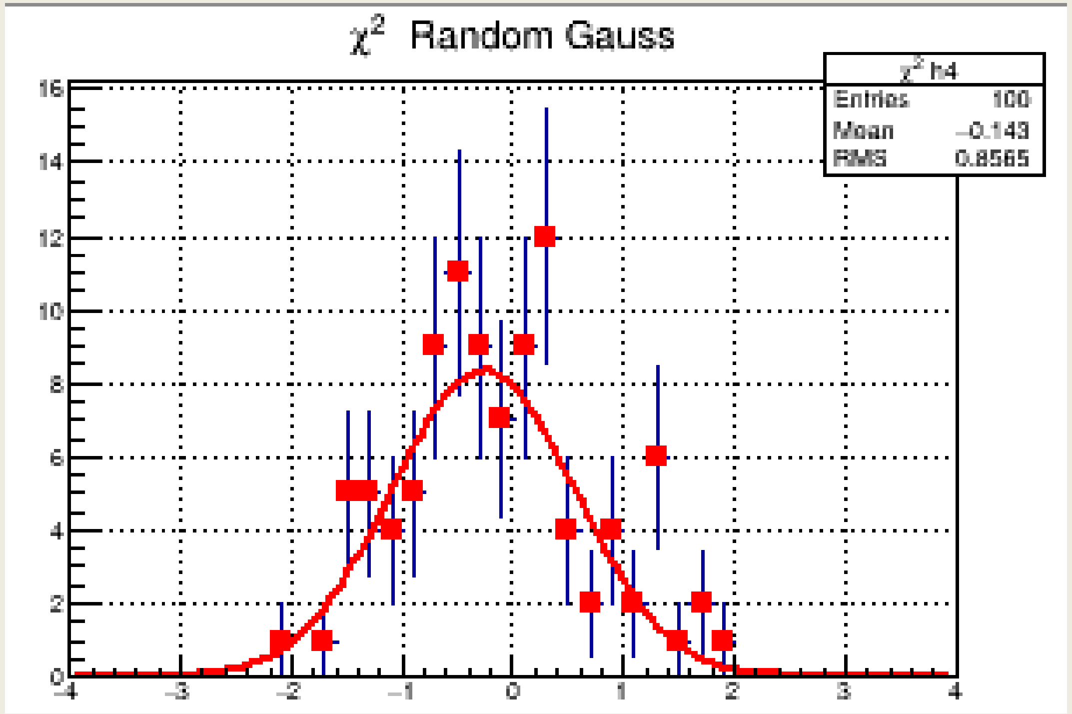
macro8.C [continue]

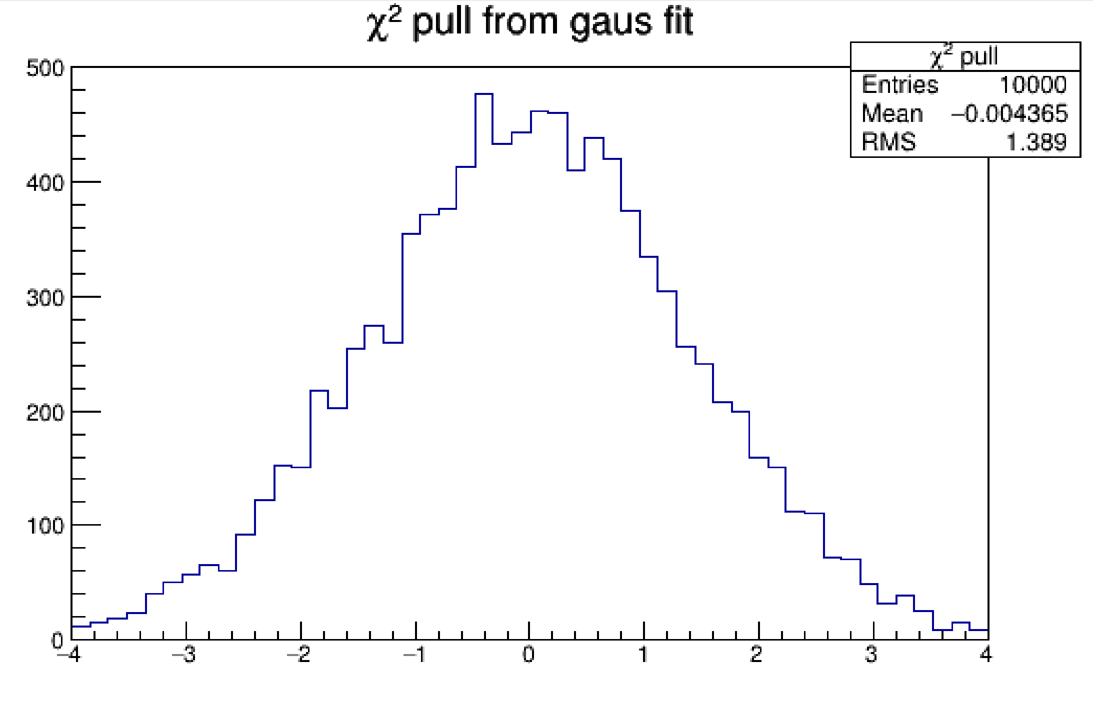
```
    TLatex latex(2,220,
                 "#splitline{Signal Peak over}{background}");
    latex.DrawClone("Same");
}
```




```
6x6 matrix is as follows

     |      0    |      1    |      2    |      3    |      4    |
----------------------------------------------------------------------
   0 |       62.4     0.07897    -0.5958     -1.991     -0.3167
   1 |    0.07897     0.01783  -0.004534     0.1189    -0.01733
   2 |    -0.5958   -0.004534    0.02382     -0.156   -0.008363
   3 |     -1.991      0.1189     -0.156      28.93     -4.619
   4 |    -0.3167    -0.01733  -0.008363     -4.619       1.04
   5 |    0.02557   0.0005859    0.00102     0.1698    -0.04511


     |      5    |
----------------------------------------------------------------------
   0 |    0.02557
   1 |  0.0005859
   2 |    0.00102
   3 |     0.1698
   4 |   -0.04511
   5 |   0.002099
```

χ^2及Likelihood拟合的Pull测试





按高斯函数产生100个数据进行
填充并拟合，给出Mean及Sigma值

拟合所得的Mean*Sqrt(100）/Sigma
检查分布的中心值是否接近于0而RMS值接近于1

siguang@pku.edu.cn

macro9.C （程序来自A ROOT Guide For Beginners）

```
// Toy Monte Carlo example.
// Check pull distribution to compare chi2 and binned
// log-likelihood methods.

pull( int n_toys = 10000,
      int n_tot_entries = 100,
      int nbins = 40,
      bool do_chi2=true ){

  TString method_prefix("Log-Likelihood ");
  if (do_chi2)
    method_prefix="#chi^{2} ";

  // Create histo
  TH1F* h4 = new TH1F(method_prefix+"h4",
                      method_prefix+" Random Gauss",
                      nbins,-4,4);
  h4->SetMarkerStyle(21);
  h4->SetMarkerSize(0.8);
  h4->SetMarkerColor(kRed);

  // Histogram for sigma and pull
  TH1F* sigma = new TH1F(method_prefix+"sigma",
                         method_prefix+"sigma from gaus fit",
                         50,0.5,1.5);
  TH1F* pull = new TH1F(method_prefix+"pull",
                        method_prefix+"pull from gaus fit",
                        50,-4.,4.);

  // Make nice canvases
  TCanvas* c0 = new TCanvas(method_prefix+"Gauss",
                            method_prefix+"Gauss",0,0,320,240);
  c0->SetGrid();
```

siguang@pku.edu.cn

macro9.C [continue]

```
// Make nice canvases
TCanvas* c1 = new TCanvas(method_prefix+"Result",
                          method_prefix+"Sigma-Distribution",
                          0,300,600,400);
c0->cd();

float sig, mean;
for (int i=0; i<n_toys; i++){
  // Reset histo contents
  h4->Reset();
  // Fill histo
  for ( int j = 0; j<n_tot_entries; j++ )
    h4->Fill(gRandom->Gaus());
  // perform fit
  if (do_chi2) h4->Fit("gaus","q"); // Chi2 fit
  else h4->Fit("gaus","lq"); // Likelihood fit
  // some control output on the way
  if (!(i%100)){
    h4->Draw("ep");
    c0->Update();}

  // Get sigma from fit
  TF1 *fit = h4->GetFunction("gaus");
  sig = fit->GetParameter(2);
  mean= fit->GetParameter(1);
  sigma->Fill(sig);
  pull->Fill(mean/sig * sqrt(n_tot_entries));
} // end of toy MC loop
```

Pull的定义：

$$P_{Pull} = \frac{P_{fit} - P_{true}}{\sigma_{Pfit}}$$

P_{fit}：是拟合给出的值
P_{true}：参数的真实值
σ_{Pfit}：是拟合给出的P的误差
正常情况下Ppull的分布以0为中心的σ为1的高斯分布

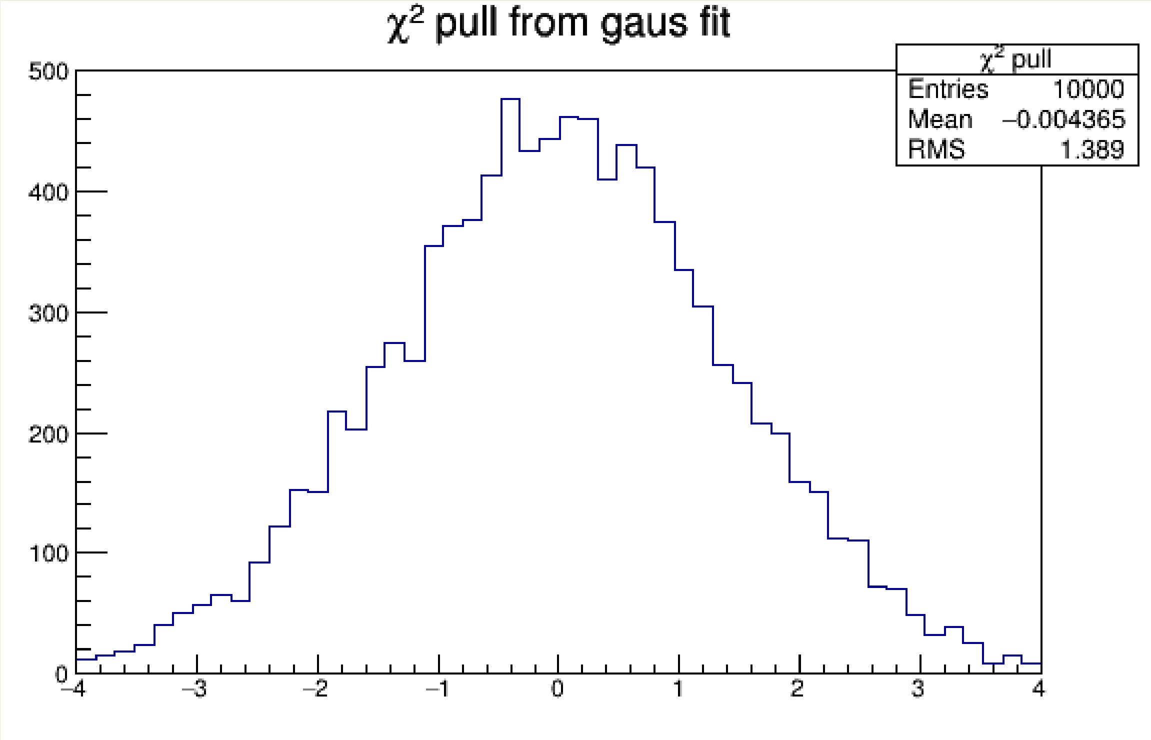
macro9.C [continue]

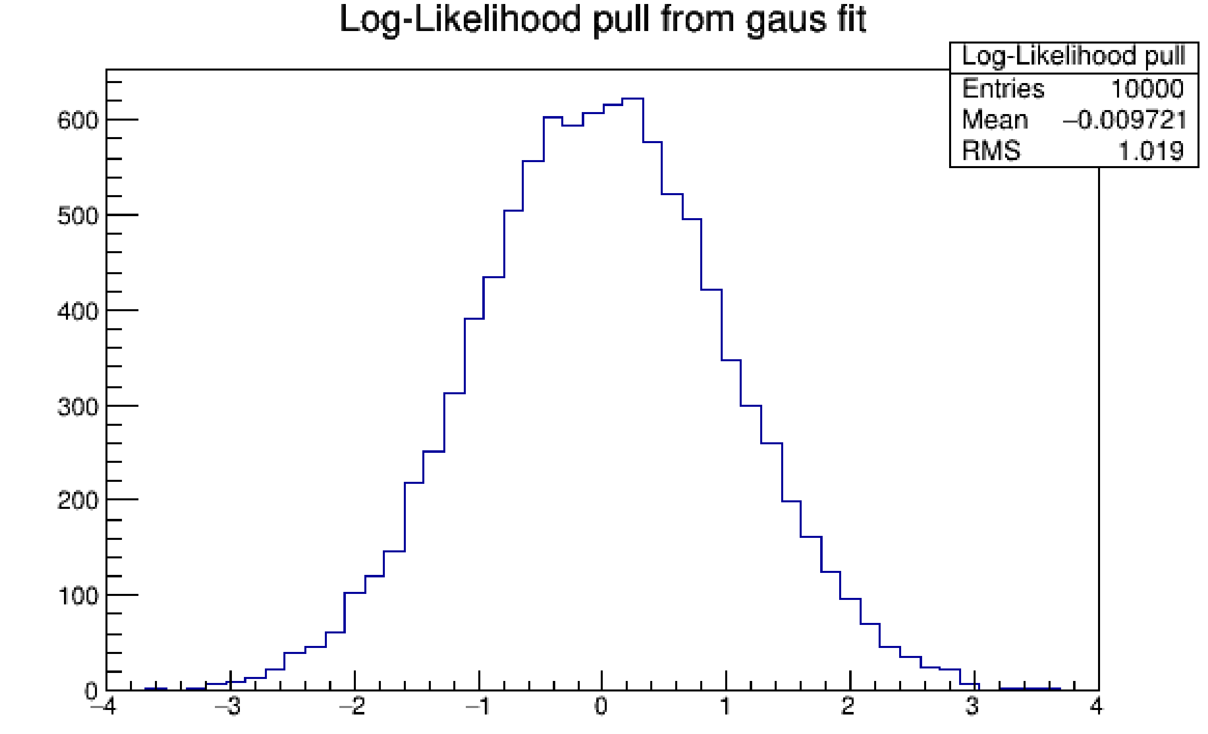
```
  // print result
  c1->cd();
  pull->Draw();
}

void macro9(){
  int n_toys=10000;
  int n_tot_entries=100;
  int n_bins=40;
  cout << "Performing Pull Experiment with chi2 \n";
  pull(n_toys,n_tot_entries,n_bins,true);
  cout << "Performing Pull Experiment with Log Likelihood\n";
  pull(n_toys,n_tot_entries,n_bins,false);
}
```

试画出sigma的分布样式

输出结果：

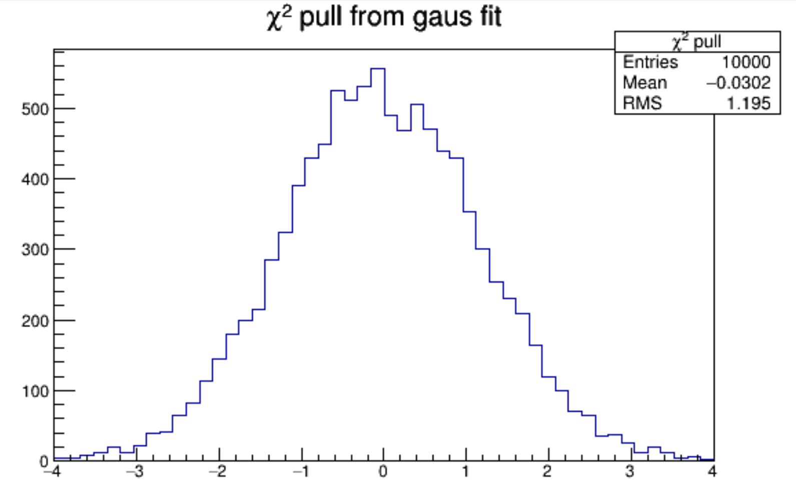
按高斯函数产生100个数据进行填充并拟合的结果：

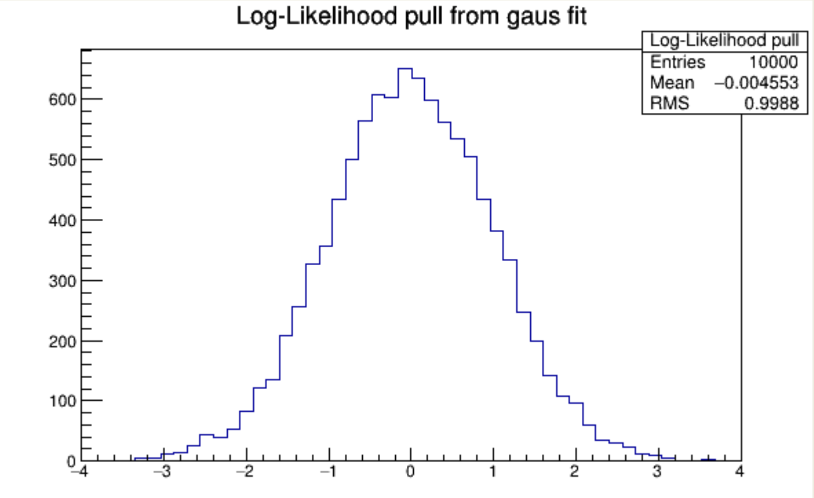




输出结果：

按高斯函数产生500个数据进行填充并拟合的结果：



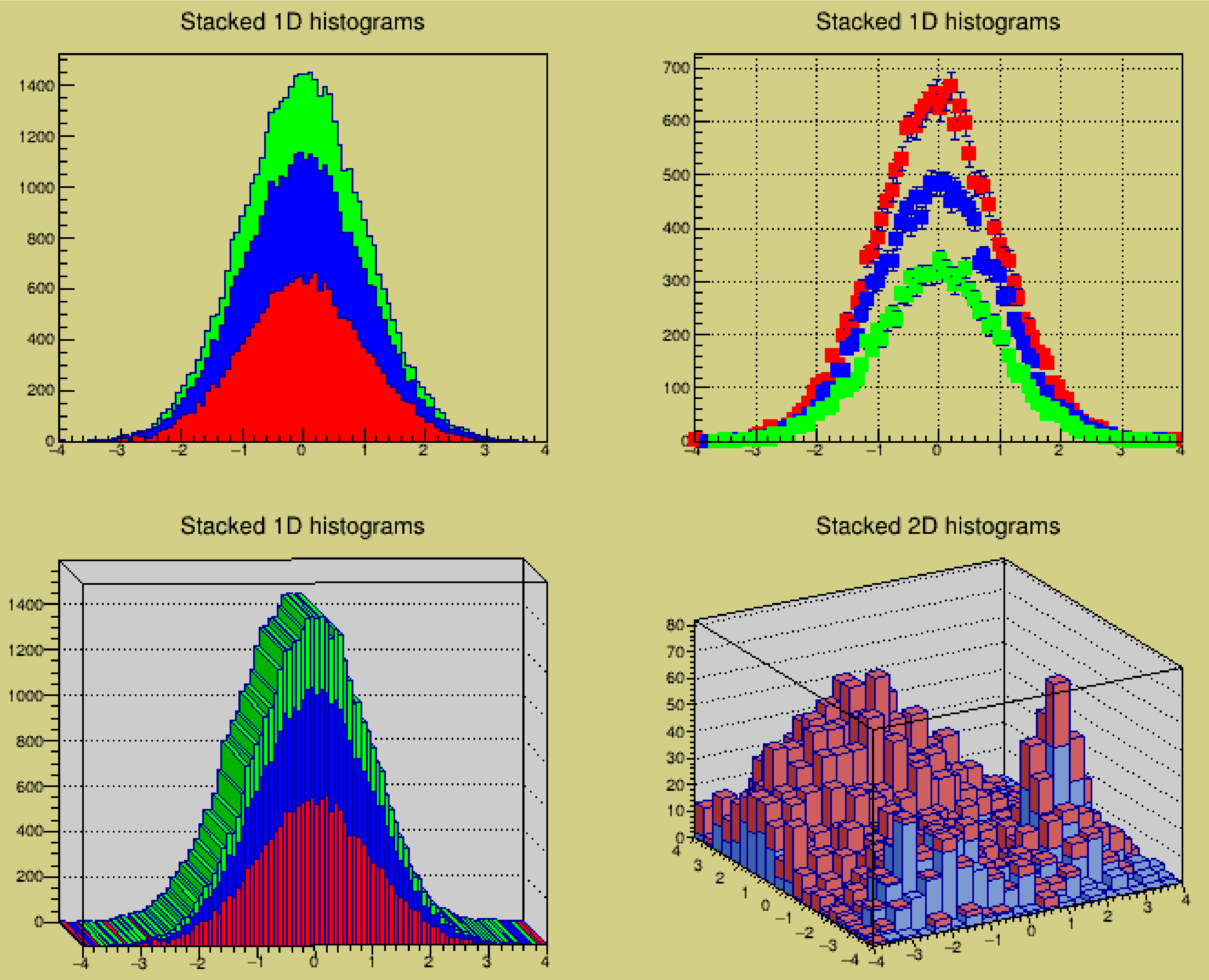


每个谱产生的数据	Mean		RMS	
	χ^2	Log-Likelihood	χ^2	Log-Likelihood
50	0.0058	0.0071	1.254	1.023
100	-0.0044	-0.0098	1.389	1.019
500	-0.0302	-0.0046	1.195	0.9988
1000	-0.0145	-0.0053	1.128	1.004
10000	-0.0237	0.0054	1.041	0.9938

测试显示：Log-Likelihood的Mean值更接近0而RMS值更接近1

siguang@pku.edu.cn

$ROOTSYS/tutorials/hist/hstack.C



siguang@pku.edu.cn

```
TCanvas *hstack() {
// Example of stacked histograms: class THStack
//
//  Author: Rene Brun

   THStack *hs = new THStack("hs","Stacked 1D histograms");
   //create three 1-d histograms
   TH1F *h1st = new TH1F("h1st","test hstack",100,-4,4);
   h1st->FillRandom("gaus",20000);
   h1st->SetFillColor(kRed);
   h1st->SetMarkerStyle(21);
   h1st->SetMarkerColor(kRed);
   hs->Add(h1st);
   TH1F *h2st = new TH1F("h2st","test hstack",100,-4,4);
   h2st->FillRandom("gaus",15000);
   h2st->SetFillColor(kBlue);
   h2st->SetMarkerStyle(21);
   h2st->SetMarkerColor(kBlue);
   hs->Add(h2st);
```

```
TH1F *h3st = new TH1F("h3st","test hstack",100,-4,4);
h3st->FillRandom("gaus",10000);
h3st->SetFillColor(kGreen);
h3st->SetMarkerStyle(21);
h3st->SetMarkerColor(kGreen);
hs->Add(h3st);

TCanvas *cst = new TCanvas("cst","stacked hists",10,10,700,700);
cst->SetFillColor(41);
cst->Divide(2,2);
// in top left pad, draw the stack with defaults
cst->cd(1);
hs->Draw();
// in top right pad, draw the stack in non-stack mode
// and errors option
cst->cd(2);
gPad->SetGrid();
hs->Draw("nostack,e1p");
```

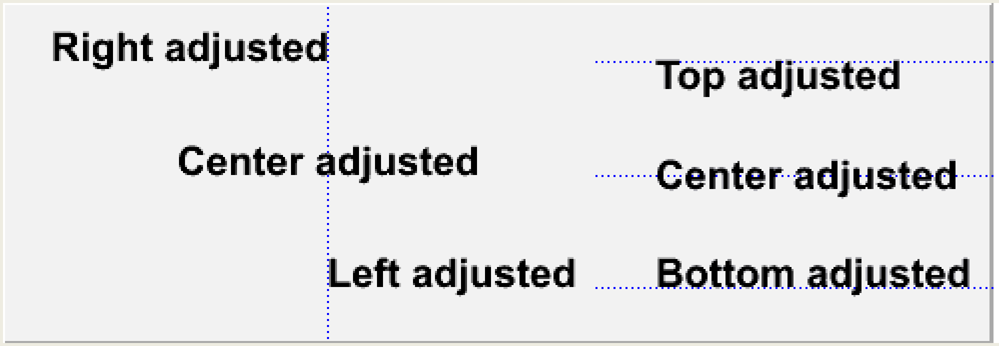
```
//in bottom left, draw in stack mode with "lego1" option
cst->cd(3);
gPad->SetFrameFillColor(17);
gPad->SetTheta(3.77);
gPad->SetPhi(2.9);
hs->Draw("lego1");
```

```
cst->cd(4);
//create two 2-D histograms and draw them in stack mode
gPad->SetFrameFillColor(17);
THStack *a = new THStack("a","Stacked 2D histograms");
TF2 *f1 = new TF2("f1",
   "xygaus + xygaus(5) + xylandau(10)",-4,4,-4,4);
Double_t params[] = {130,-1.4,1.8,1.5,1, 150,2,0.5,-2,0.5,
   3600,-2,0.7,-3,0.3};
f1->SetParameters(params);
TH2F *h2sta = new TH2F("h2sta","h2sta",20,-4,4,20,-4,4);
h2sta->SetFillColor(38);
h2sta->FillRandom("f1",4000);
TF2 *f2 = new TF2("f2","xygaus + xygaus(5)",-4,4,-4,4);
Double_t params[] = {100,-1.4,1.9,1.1,2, 80,2,0.7,-2,0.5};
f2->SetParameters(params);
TH2F *h2stb = new TH2F("h2stb","h2stb",20,-4,4,20,-4,4);
h2stb->SetFillColor(46);
h2stb->FillRandom("f2",3000);
a->Add(h2sta);
a->Add(h2stb);
a->Draw();
return cst;
```

文字属性控制

王思广
北京大学物理学院
siguang@pku.edu.cn

对齐控制




```cpp
// Horizontal alignment.
TText *th1 = new TText(0.33,0.165,"Left adjusted");
th1->SetTextAlign(11); th1->SetTextSize(0.12);
th1->Draw();

TText *th2 = new TText(0.33,0.493,"Center adjusted");
th2->SetTextAlign(21); th2->SetTextSize(0.12);
th2->Draw();

TText *th3 = new TText(0.33,0.823,"Right adjusted");
th3->SetTextAlign(31); th3->SetTextSize(0.12);
th3->Draw();
```

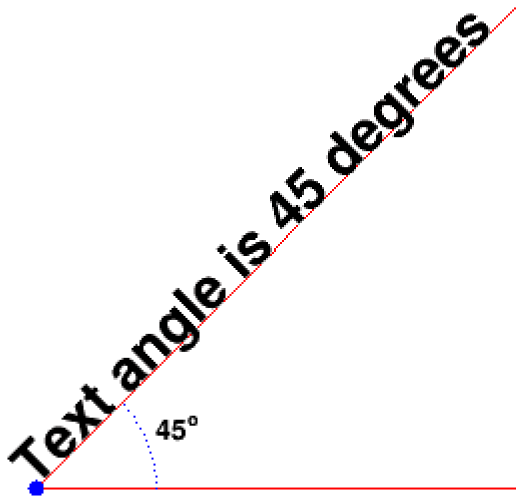
水平：1（左对齐）；2（中间对齐）；3（右对齐）

```cpp
// Vertical alignment.
TText *tv1 = new TText(0.66,0.165,"Bottom adjusted");
tv1->SetTextAlign(11); tv1->SetTextSize(0.12);
tv1->Draw();

TText *tv2 = new TText(0.66,0.493,"Center adjusted");
tv2->SetTextAlign(12); tv2->SetTextSize(0.12);
tv2->Draw();

TText *tv3 = new TText(0.66,0.823,"Top adjusted");
tv3->SetTextAlign(13); tv3->SetTextSize(0.12);
tv3->Draw();
```

垂直：1（底对齐）；2（中间对齐）；3（上对齐）

角度控制

```
void Angle(){
   TCanvas *Ta = new TCanvas("Ta","Text angle",0,0,300,326);
   Ta->Range(0,0,1,1);
   TLine *l = new TLine();
   l->SetLineColor(kRed);
   l->DrawLine(0.1,0.1,0.9,0.1);
   l->DrawLine(0.1,0.1,0.9,0.9);
   TMarker *m = new TMarker();
   m->SetMarkerStyle(20);
   m->SetMarkerColor(kBlue);
   m->DrawMarker(0.1,0.1);
   TArc *a = new TArc();
   a->SetFillStyle(0);
   a->SetLineColor(kBlue); a->SetLineStyle(3);
   a->DrawArc(0.1, 0.1, 0.2, 0.,45.,"only");
   TText *tt = new TText(0.1,0.1,"Text angle is 45 degrees");
   tt->SetTextAlign(11); tt->SetTextSize(0.1);
   tt->SetTextAngle(45);
   tt->Draw();
   TLatex *t1 = new TLatex(0.3,0.18,"45^{o}");
   t1->Draw();
```



参数：0.1,0.1为圆点
0.2为半径
0.， 45.为角度
“only”只画弧，缺省还包括两端到圆心的连线

字体种类


```
TCanvas * fonts(){
   TCanvas *Tf = new TCanvas("Tf", "Tf",0,0,500,700);
   Tf->Range(0,0,1,1);
   Tf->SetBorderSize(2);
   Tf->SetFrameFillColor(0);

   double y = 0.95;
   for (int f = 12; f<=142; f+=10) {
      if (f<142) drawtext(0.02,y, f,"ABCDFGF abcdefgh 0123456789 @#$");
      else drawtext(0.02,y, f,"ABCD efgh 01234 @#$");
      y- = 0.07;
   }
   return Tf;
}
void drawtext(double x, double y, int f, char *s)
{
   TLatex *t = new TLatex(x,y,Form("#font[41]{%d :} %s",f,s));
   t->SetTextFont(f);
   t->SetTextAlign(12);
   t->Draw();
}
```

建议：根据个人爱好，直接用SetTextFont选用上面的编号即可！个位数字可取0,1,2，数字越大越精确但速度会慢。非个位代表字体编号。

字体种类

152 : *ABXΔ εφγη 01234 ≅#∃*

162 : ABCD efgh 01234 @#$

172 : *ABCD efgh 01234 @#$*

182 : **ABCD efgh 01234 @#$**

192 : ***ABCD efgh 01234 @#$***

202 :

212 :

222 :

232 :

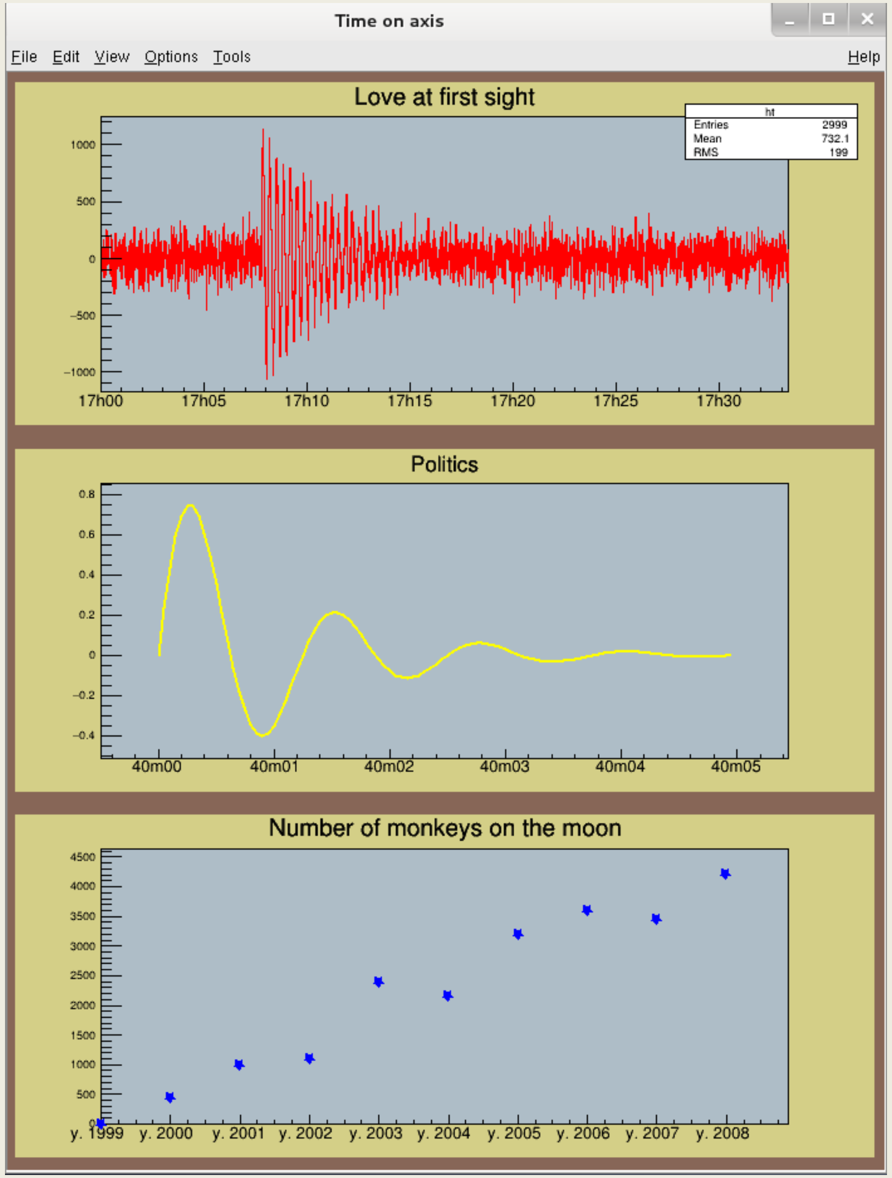
242 :

最大字体编号：19

时间坐标轴的显示

王思广
北京大学物理学院
siguang@pku.edu.cn

$ROOTSYS/tutorials/graphs/timeonaxis.C

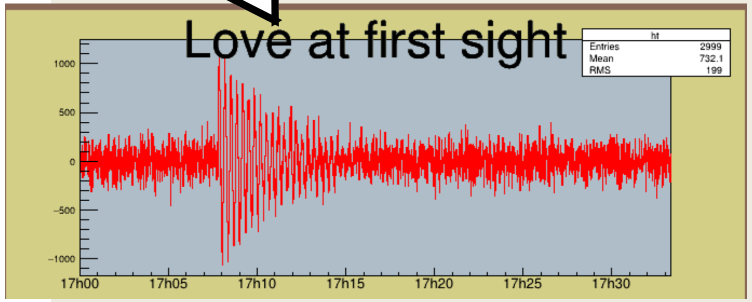


```
#include <time.h>

void timeonaxis()
{
// This macro illustrates the use of the time mode on the axis
// with different time intervals and time formats. It's result can
// be seen begin_html <a href="gif/timeonaxis.gif">here</a> end_html
// Through all this script, the time is expressed in UTC. some
// information about this format (and others like GPS) may be found at
// begin_html <a href="http://tycho.usno.navy.mil/systime.html">http://tycho.usno.navy.mil/systime.html</a> end_html
//  or
// begin_html <a href="http://www.topology.org/sci/time.html">http://www.topology.org/sci/time.html</a> end_html
//
// The start time is : almost NOW (the time at which the script is executed)
// actualy, the nearest preceding hour beginning.
// The time is in general expressed in UTC time with the C time() function
// This will obviously most of the time not be the time displayed on your watch
// since it is universal time. See the C time functions for converting this time
// into more useful structures.
//Author: Damir Buskulic

   time_t script_time;
   script_time = time(0);
   script_time = 3600*(int)(script_time/3600);

// The time offset is the one that will be used by all graphs.
// If one changes it, it will be changed even on the graphs already defined
   gStyle->SetTimeOffset(script_time);
```

```
   ct = new TCanvas("ct","Time on axis",10,10,700,900);
   ct->Divide(1,3);
   ct->SetFillColor(28);

   int i;

//======= Build a signal : noisy damped sine ======
//        Time interval : 30 minutes

   gStyle->SetTitleH(0.08);
   float noise;
   ht = new TH1F("ht","Love at first sight",3000,0.,2000.);
   for (i=1;i<3000;i++) {
      noise = gRandom->Gaus(0,120);
      if (i>700) {
         noise += 1000*sin((i-700)*6.28/30)*exp((double)(700-i)/300);
      }
      ht->SetBinContent(i,noise);
   }
   ct->cd(1);
   ct_1->SetFillColor(41);
   ct_1->SetFrameFillColor(33);
   ht->SetLineColor(2);
   ht->GetXaxis()->SetLabelSize(0.05);
   ht->Draw();
```

区间
2000

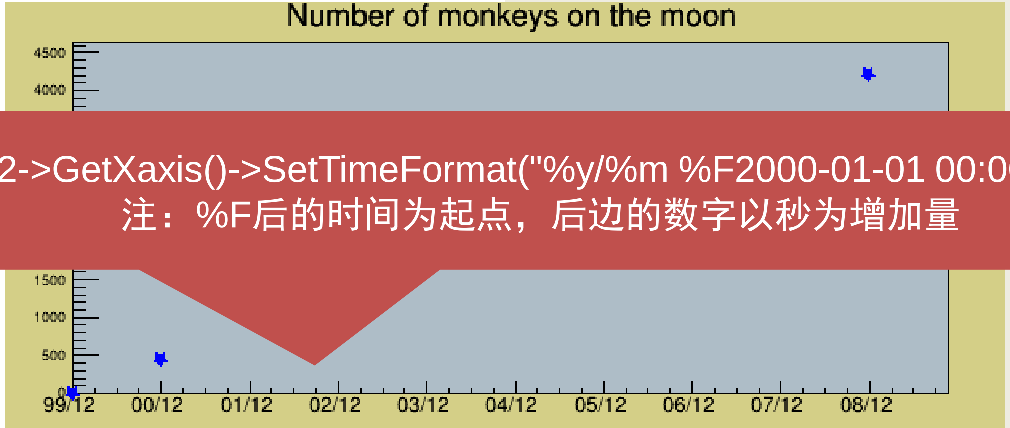
gStyle->SetTitleH(0.2)的结果



```
// Sets time on the X axis
// The time used is the one set as time offset added to the value
// of the axis. This is converted into day/month/year hour:min:sec and
// a reasonable tick interval value is chosen.
   ht->GetXaxis()->SetTimeDisplay(1);

//======= Build a simple graph beginning at a different time ======
//        Time interval : 5 seconds

   float x[100], t[100];
   for (i=0;i<100;i++) {
      x[i] = sin(i*4*3.1415926/50)*exp(-(double)i/20);
      t[i] = 6000+(double)i/20;
   }
   gt = new TGraph(100,t,x);
   gt->SetTitle("Politics");
   ct->cd(2);
   ct_2->SetFillColor(41);
   ct_2->SetFrameFillColor(33);
   gt->SetFillColor(19);
   gt->SetLineColor(5);
   gt->SetLineWidth(2);
   gt->Draw("AL");
   gt->GetXaxis()->SetLabelSize(0.05);
// Sets time on the X axis
   gt->GetXaxis()->SetTimeDisplay(1);
   gPad->Modified();
```

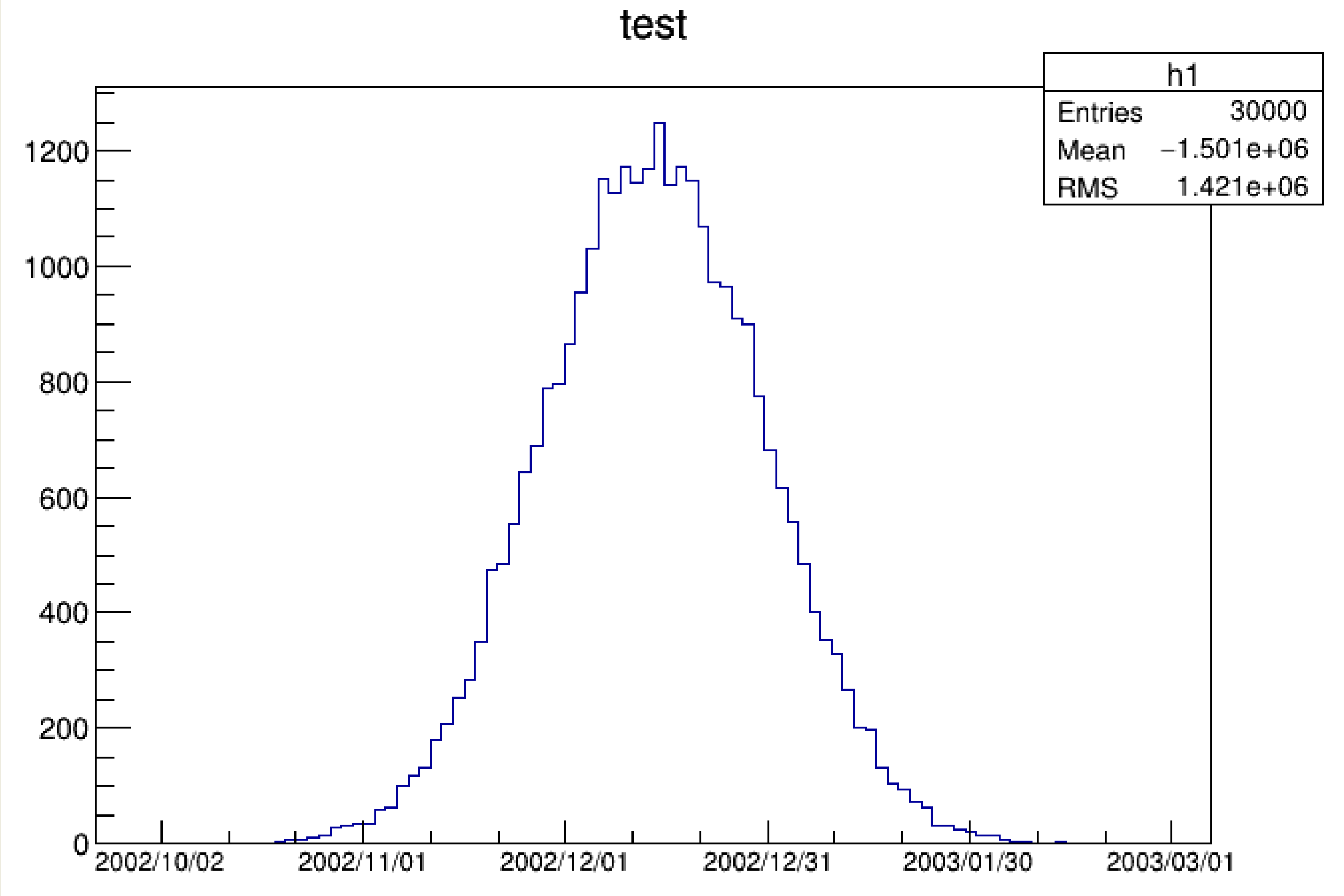
```
//======= Build a second simple graph for a very long time interval =======
//         Time interval : a few years

   float x2[10], t2[10];
   for (i=0;i<10;i++) {
      x2[i] = gRandom->Gaus(500,100)*i;
      t2[i] = i*365*86400;
   }
   gt2 = new TGraph(10,t2,x2);
   gt2->SetTitle("Number of monkeys on the moon");
   ct->cd(3);
   ct_3->SetFillColor(41);
   ct_3->SetFrameFillColor(33);
   gt2->SetFillColor(19);
   gt2->SetMarkerColor(4);
   gt2->SetMarkerStyle(29);
   gt2->SetMarkerSize(1.3);
   gt2->Draw("AP");
   gt2->GetXaxis()->SetLabelSize(0.05);
// Sets time on the X axis
   gt2->GetXaxis()->SetTimeDisplay(1);
```

```
// One can choose a different time format than the one chosen by default
// The time format is the same as the one of the C strftime() function
// It's a string containing the following formats :
//    for date :
//      %a abbreviated weekday name
//      %b abbreviated month name
//      %d day of the month (01-31)
//      %m month (01-12)
//      %y year without century
//      %Y year with century
//
//    for time :
//      %H hour (24-hour clock)
//      %I hour (12-hour clock)
//      %p local equivalent of AM or PM
//      %M minute (00-59)
//      %S seconds (00-61)
//      %% %
// The other characters are output as is.

   gt2->GetXaxis()->SetTimeFormat("y. %Y %F2000-01-01 00:00:00");
   gPad->Modified();

}
```



gt2->GetXaxis()->SetTimeFormat("%y/%m %F2000-01-01 00:00:00");
注：%F后的时间为起点，后边的数字以秒为增加量

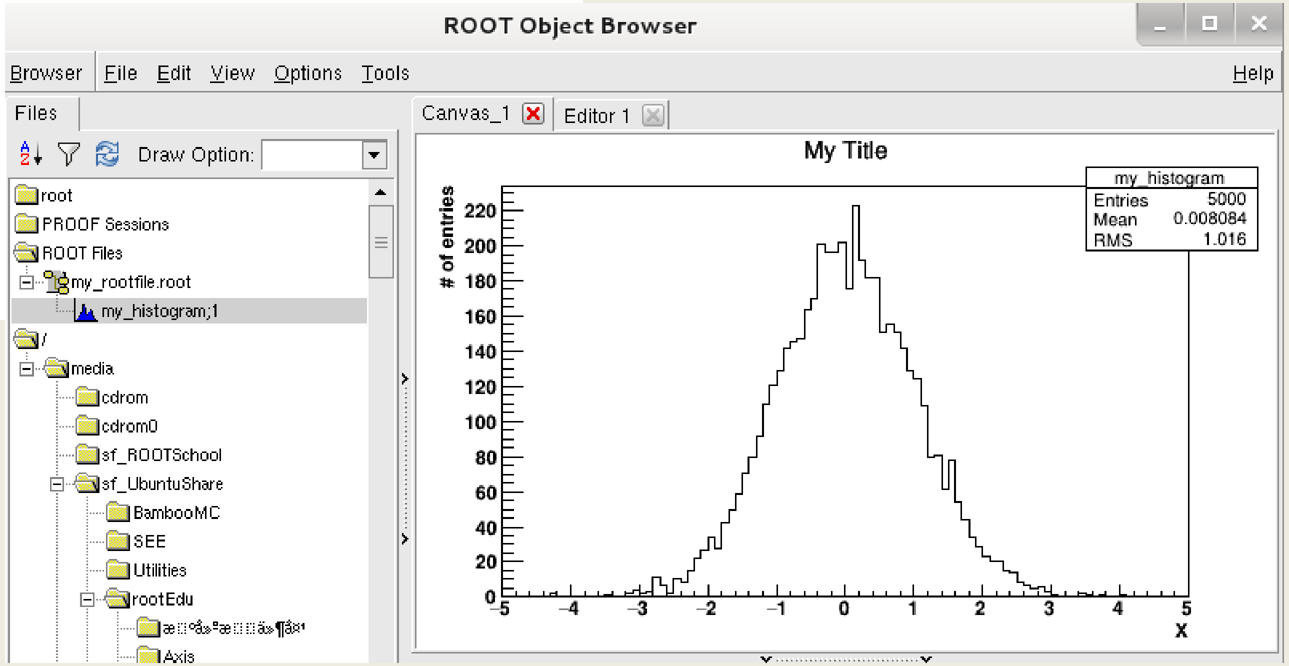
$ROOTSYS/tutorials/graphs/timeonaxis2.C



```
void timeonaxis2() {
   // Define the time offset as 2003, January 1st
   //Author: Olivier Couet

   TDatime T0(2003,01,01,00,00,00);
   int X0 = T0.Convert();
   gStyle->SetTimeOffset(X0);

   // Define the lowest histogram limit as 2002, September 23rd
   TDatime T1(2002,09,23,00,00,00);
   int X1 = T1.Convert()-X0;

   // Define the highest histogram limit as 2003, March 7th
   TDatime T2(2003,03,07,00,00,00);
   int X2 = T2.Convert(1)-X0;

   TH1F * h1 = new TH1F("h1","test",100,X1,X2);

   TRandom r;
   for (Int_t i=0;i<30000;i++) {
      Double_t noise = r.Gaus(0.5*(X1+X2),0.1*(X2-X1));
      h1->Fill(noise);
   }

   h1->GetXaxis()->SetTimeDisplay(1);
   h1->GetXaxis()->SetLabelSize(0.03);
   h1->GetXaxis()->SetTimeFormat("%Y\/%m\/%d");
   h1->Draw();
}
```

将TH1F保存到ROOT文件中

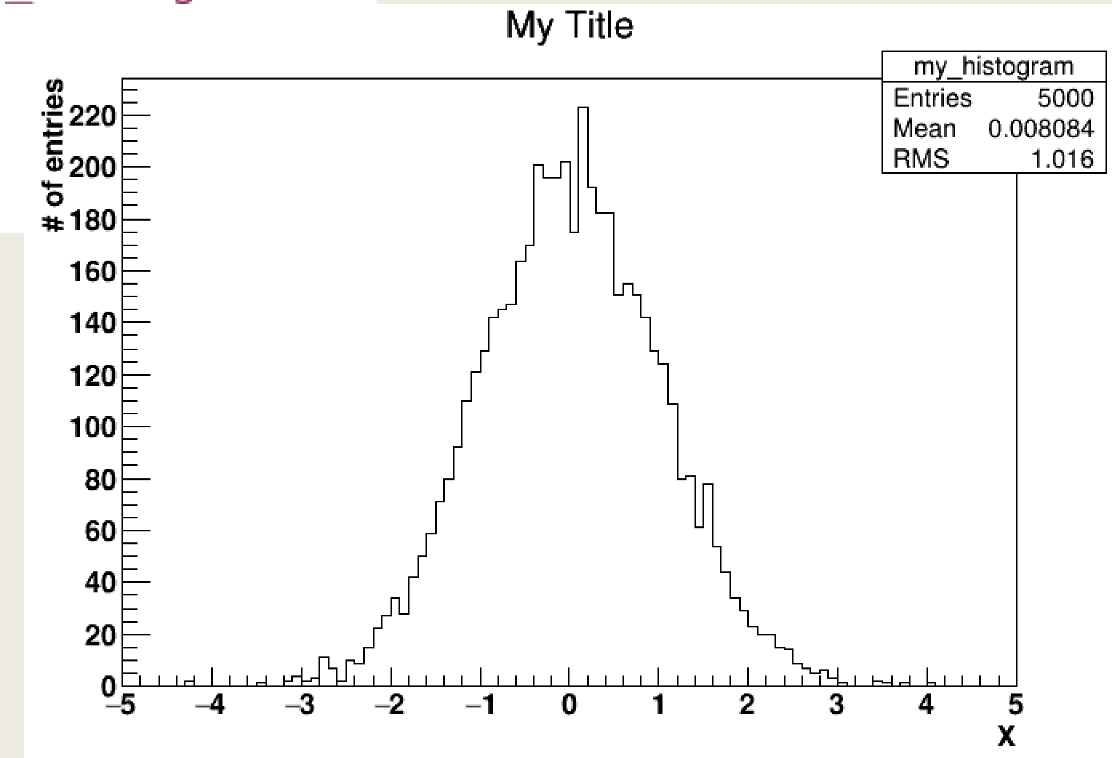
```
void write_to_file(){

    // Instance of our histogram
    TH1F h("my_histogram","My Title;X;# of entries",100,-5,5);

    // Let's fill it randomly
    h.FillRandom("gaus");

    // Let's open a TFile
    TFile out_file("my_rootfile.root","RECREATE");

    // Write the histogram in the file
    h.Write();

    // Close the file
    out_file.Close();
}
```



其它的继承于TObject类的保持方式相同

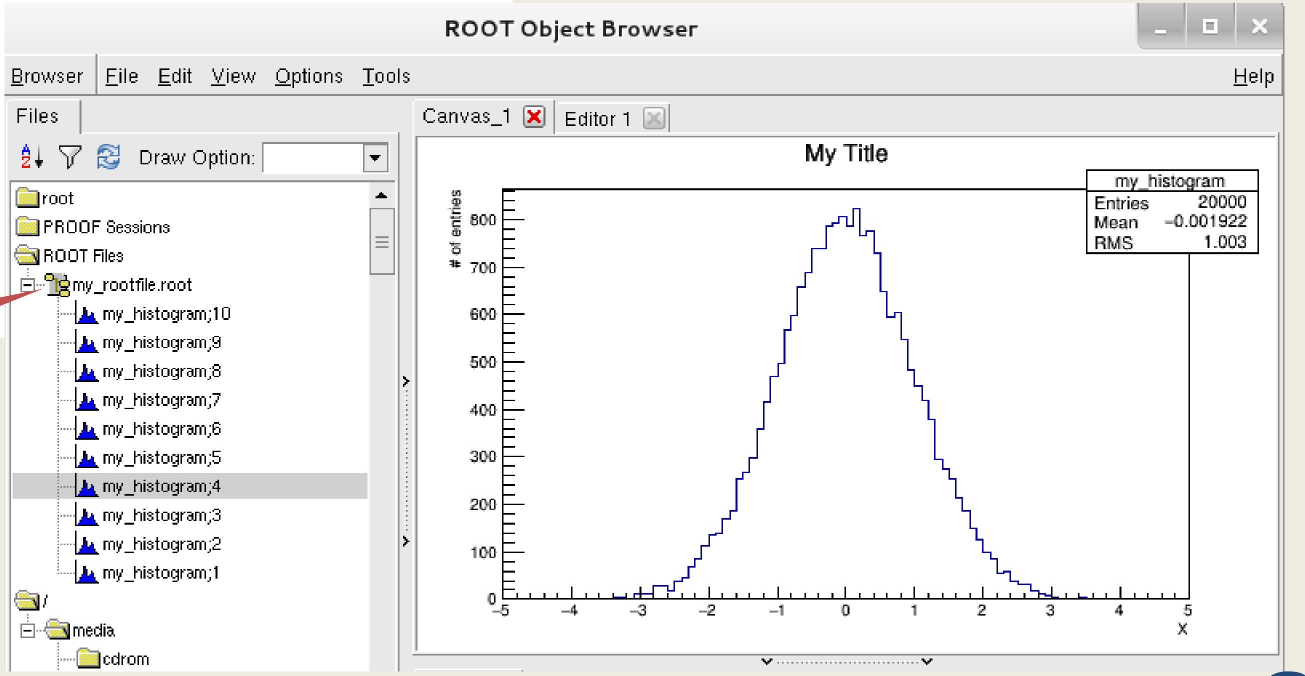
从ROOT文件中读出

```
void read_from_file(){

    // Let's open the TFile
    TFile* in_file= new TFile("my_rootfile.root");

    // Get the Histogram out
    TH1F* h = (TH1F*) in_file->Get("my_histogram");

    // Draw it
    h->Draw();
}
```



多次将TH1F写入文件

```
void write_to_file(){

  // Instance of our histogram
  TH1F h("my_histogram","My Title;X;# of entries",100,-5,5);

  // Let's open a TFile
  TFile out_file("my_rootfile.root","RECREATE");
  for(Int_t i=0; i<10; i++){
    // Let's fill it randomly
    h.FillRandom("gaus",5000);
    // Write the histogram in the file
    h.Write();
  }
  // Close the file
  out_file.Close();
}
```



查看内容：
$root my_rootfile.root
[]new TBrowser
双击展开

读取特定的第3个写入的TH1F

```
void read_from_file(){

    // Let's open the TFile
    TFile* in_file= new TFile("my_rootfile.root");

    // Get the Histogram out
    TH1F* h = (TH1F*) in_file->Get("my_histogram;3");

    // Draw it
    h->Draw();
}
```