

Introduction to Programming and Computational Physics

Root : A brief tutorial

Root

ROOT

An Object-Oriented
Data Analysis Framework



Root is an Object Oriented Data Analysis Framework developed at CERN starting from 1994. It is the most common analysis tool used in particle physics. Root libraries are written in C++, the most common OO language used in physics and other sciences.

Root installation

Linux users: root is already installed in computer room, don't need any action.

If you plan to install it on your laptop visit

<https://root.cern.ch/>

and follow instructions

Documentation

Available on root website:

<http://root.cern.ch/> -> Documentation ->

- User's guide
- Reference guide
- Tutorials
- HowTo's

The content of these slides is anyway enough for the purposes of our course

Basic commands

To open a root session, just type `root`

```
% root
*****
*                                     *
*      W E L C O M E   t o   R O O T      *
*                                     *
*   Version    5.20/00           24 June 2007   *
*                                     *
*   You are welcome to visit our Web site   *
*      http://root.cern.ch                *
*                                     *
*****

ROOT 5.20/00 (trunk@24525, Jun 25 2008, 12:52:00 on linux)

CINT/ROOT C/C++ Interpreter version 5.16.29, June 08, 2008
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
```

To terminate a root session, type `.q`

```
root[] .q
```

To load and/or execute type `.L (filename)` or `.x (filename)`

```
root[] .L <filename>           //load [filename]
root[] .x <filename>           //load and execute [filename]
```

You don't need to compile your source code!

Histograms

We will use the class TH1F and TH2F (one- and two- dimensional histograms with floats) derived from TH1

<http://root.cern.ch/root/html404/TH1F.html>

Creators

```
TH1F *h1 = new TH1F("h1","h1 title",100, 0., 4.);  
TH2F *h2 = new TH2F("h2","h2 title",40,0.,4.,30,-3.,3.);
```

Add an entry:

```
h1 → Fill(x);  
h2 → Fill(x,y);
```

Set axis title:

```
h1 → GetXAxis() →SetTitle("X axis title");  
h1 → GetYAxis() →SetTitle("Y axis title");
```

Draw:

```
h1->Draw();    option  
h2->Draw( "lego" );
```

Documentation:

<http://root.cern.ch/download/doc/3Histograms.pdf>

Histograms

One example

```
void my_first_histo()
{
    TH1F *h1 = new TH1F("h1", "My first root histogram",
100, 0., 40.);
    FILE *fp = fopen("data.txt", "r");
    float data;
    while (!feof(fp))
    {
        fscanf(fp, "%f", &data);
        h1->Fill(data);
    }
    fclose(fp);
    h1->GetXaxis()->SetTitle("Value");
    h1->GetYaxis()->SetTitle("# of entries");
    h1->Draw();
}
```

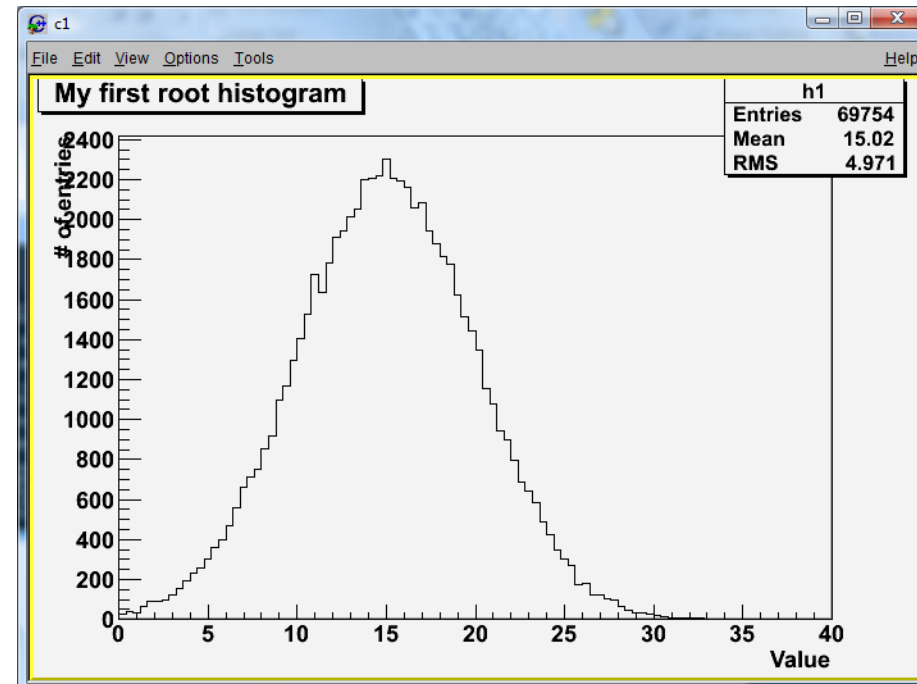
Histograms

```
ROOT session
mylib.c
mylib.h
my_first_histo.C
root [0] .q

C:\Users\ciro\Desktop\C\scripts>root
*****
*          W E L C O M E  t o  R O O T          *
*   Version   5.22/00  17 December 2008         *
*   You are welcome to visit our Web site       *
*   http://root.cern.ch                        *
*****

ROOT 5.22/00 <trunk@26997, Jan 05 2009, 10:56:00 on win32>

CINT/ROOT C/C++ Interpreter version 5.16.29, Jan 08, 2008
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] .x my_first_histo.C
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [1] _
```



Fitting histograms

```
void Fit(const *char fname, ...) TH1::Fit(...)
```

fname is the name of the fitted function. Predefined functions are:

gaus (gaussian function with 3 parameters)

$$f(x) = p_0 \cdot \exp(-0.5 \cdot ((x-p_1)/p_2)^2)$$

expo (an exponential with 2 parameters)

$$f(x) = \exp(p_0 + p_1 \cdot x)$$

polN: a polynomial of degree N:

$$f(x) = p_0 + p_1 \cdot x + p_2 \cdot x^2 + \dots$$

```
h1->Fit("gaus");
```

```
h1->Fit("pol1");
```

Fitting histograms

```
void my_first_histo()
{
    TH1F *h1 = new TH1F("h1", "My first root histogram",
100, 0., 40.);
    FILE *fp = fopen("data.txt", "r");
    float data;
    while (!feof(fp))
    {
        fscanf(fp, "%f", &data);
        h1->Fill(data);
    }
    fclose(fp);
    h1->GetXaxis()->SetTitle("Value");
    h1->GetYaxis()->SetTitle("# of entries");
    h1->Draw();
    h1->Fit("gaus");
}
```

Fitting histograms

```
ROOT session

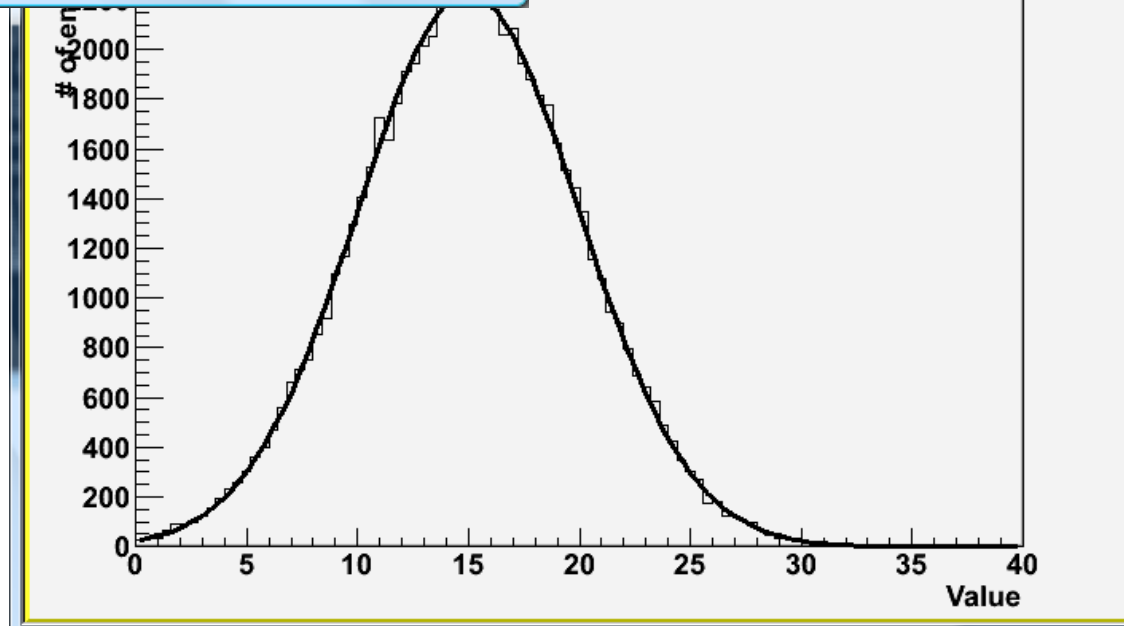
*      WELCOME to ROOT      *
*      Version  5.22/00  17 December 2008      *
*      You are welcome to visit our Web site      *
*      http://root.cern.ch      *
*****

ROOT 5.22/00 (trunk@26997, Jan 05 2009, 10:56:00 on win32)

CINT/ROOT C/C++ Interpreter version 5.16.29, Jan 08, 2008
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] .x my_first_histo.C
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
FCN=90.7775 FROM MIGRAD  STATUS=CONVERGED  65 CALLS  66 TOTAL
EDM=1.3658e-009  STRATEGY= 1  ERROR MATRIX ACCURATE

EXT  PARAMETER
NO.  NAME      VALUE      ERROR      STEP      FIRST
1  Constant  2.22474e+003  1.03718e+001  3.94528e-002  4.62806e-006
2  Mean      1.49938e+001  1.91248e-002  8.92586e-005  6.07439e-004
3  Sigma     4.99776e+000  1.37587e-002  3.49488e-006  6.68762e-002
root [1]
```

h1	
Entries	69754
Mean	15.02
RMS	4.971



Fit with a user defined function

```
TF1 *f1 = new TF1("f1", "sin(x)/x", 0, 10)
```

f1 is then $\sin(x)/x$ 0,10 is the range

```
TF1 *f1 = new TF1("f1", "[0]*x*sin([1]*x)", -3, 3);  
f1->SetParameter(0, 10); //setting initial value  
f1->SetParameter(1, 0.5);
```

**f1 here has two free parameters and SetParameter
Sets the initial value of each of them**

```
f1->Draw(); // to draw the function
```

```
h1->Fit(f1); // to fit the histogram h1 as f1
```

<http://root.cern.ch/download/doc/5FittingHistograms.pdf>

Example: drawing $\log(1+x^2)\sqrt{\cos(x)}$

```
void fun_draw()  
{  
    TF1 *f1 = new TF1("f1", "log(1+pow(x,2))*sqrt(cos(x))", -1, 1);  
    f1->Draw();  
}
```

