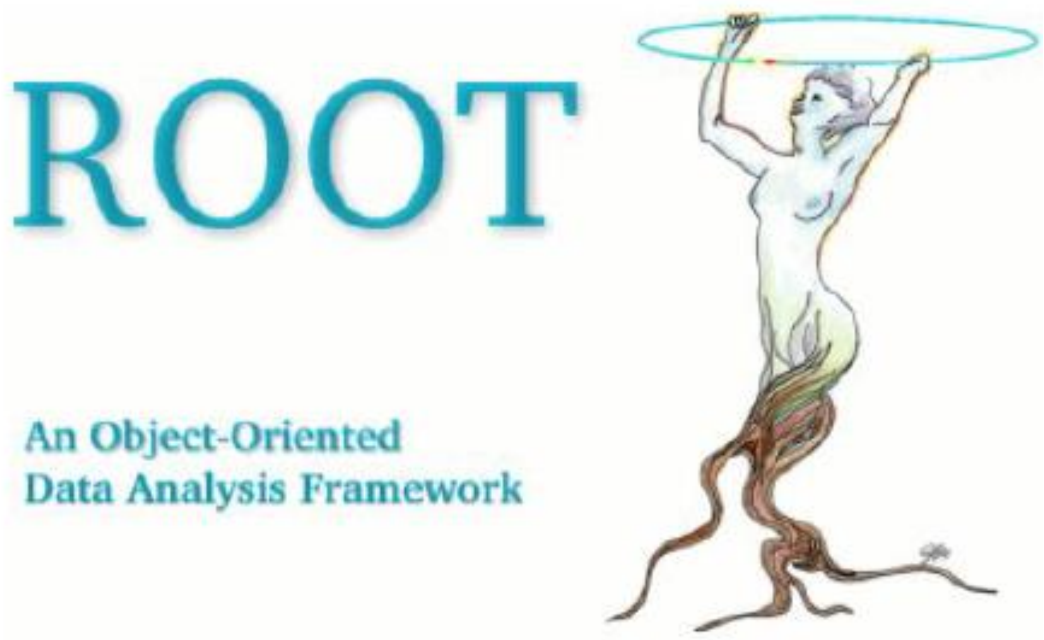
An Introduction to ROOT



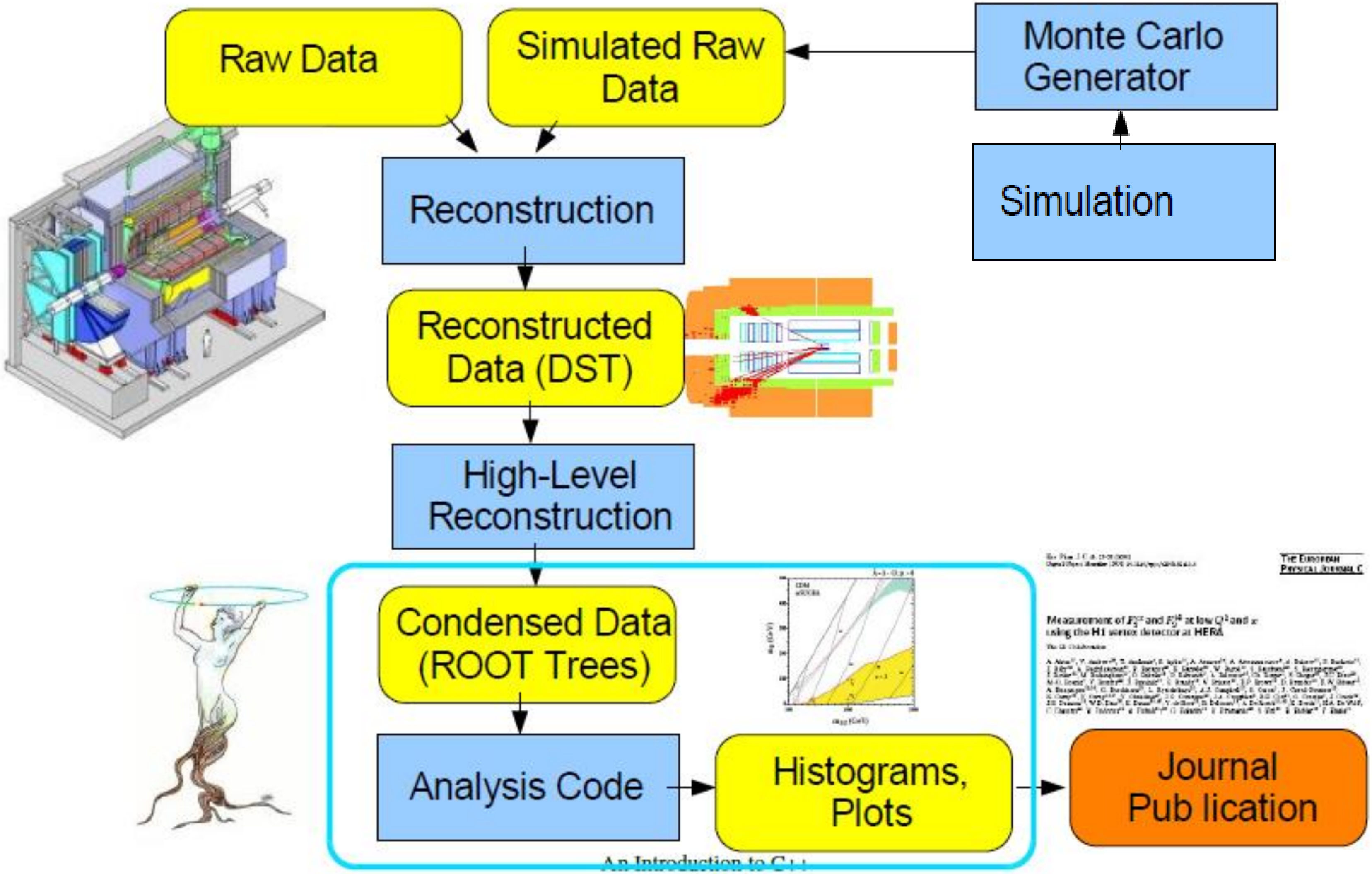
Introduction

ROOT is a Package for Data Analysis

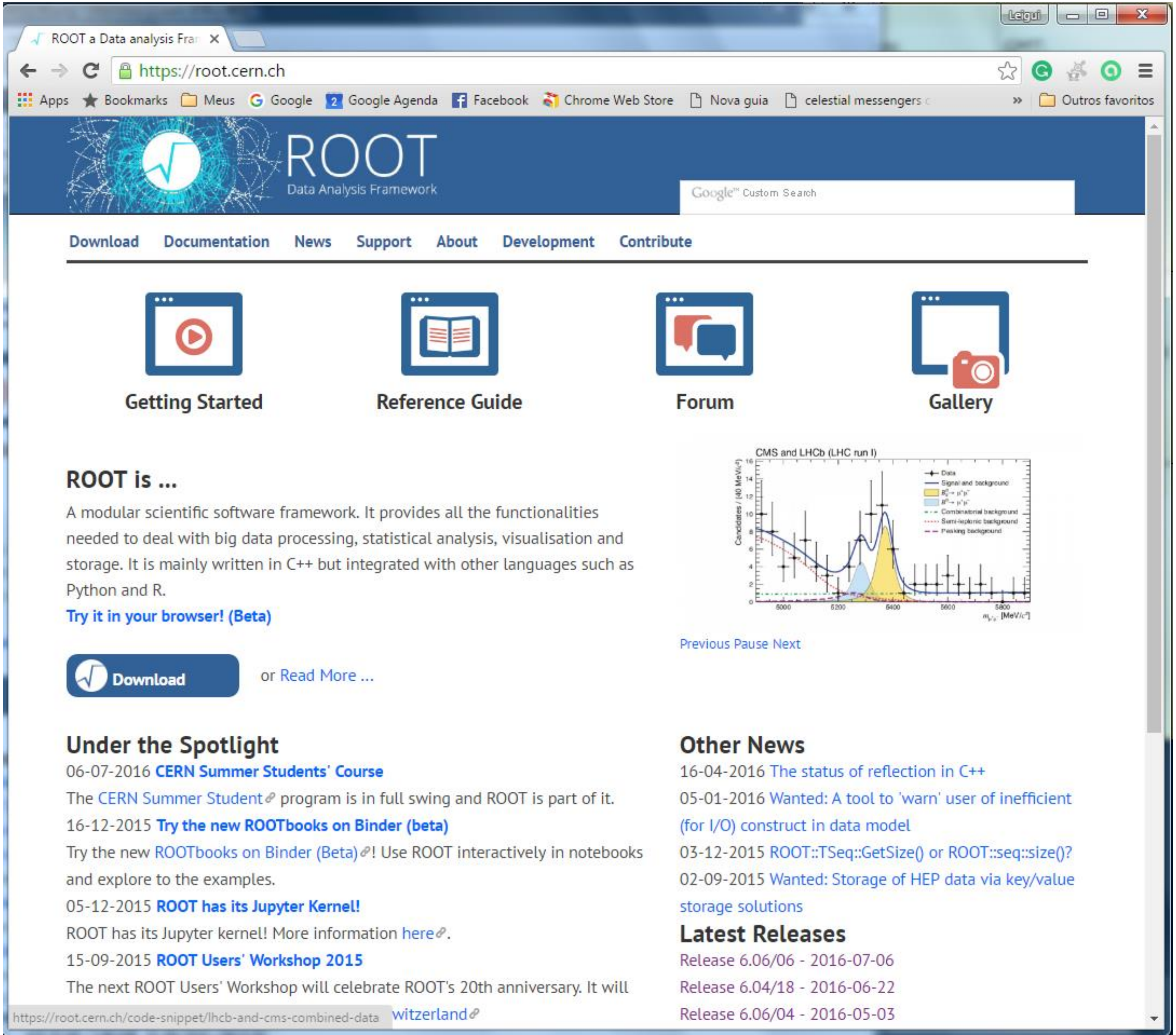
ROOT Provides:

- Several C++ Libraries
  - To store data in histograms
  - To store data in n-tuples, called “ROOT Trees”
  - To visualize histograms and n-tuples
  - To perform fits
- An Interactive Environment
  - To run C++ programs interactively
  - To visualize data
  - To perform fits

The Analysis Chain in High Energy Physics


Raw Data
Simulated Raw Data
Monte Carlo Generator
Simulation
Reconstruction
Reconstructed Data (DST)
High-Level Reconstruction
Condensed Data (ROOT Trees)
Analysis Code
Histograms, Plots
Journal Pub lication
THE EUROPEAN PHYSICAL JOURNAL C
Measurement of $F_2^{c\bar{c}}$ and $F_2^{b\bar{b}}$ at low Q^2 and x using the H1 vertex detector at HERA

An Introduction to C++



ROOT is ...

A modular scientific software framework. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualisation and storage. It is mainly written in C++ but integrated with other languages such as Python and R.

Try it in your browser! (Beta)

or Read More ...

Under the Spotlight

06-07-2016 **CERN Summer Students' Course**

The CERN Summer Student program is in full swing and ROOT is part of it.

16-12-2015 **Try the new ROOTbooks on Binder (beta)**

Try the new ROOTbooks on Binder (Beta)! Use ROOT interactively in notebooks and explore to the examples.

05-12-2015 **ROOT has its Jupyter Kernel!**

ROOT has its Jupyter kernel! More information here.

15-09-2015 **ROOT Users' Workshop 2015**

The next ROOT Users' Workshop will celebrate ROOT's 20th anniversary. It will

witzerland

Other News

16-04-2016 The status of reflection in C++

05-01-2016 Wanted: A tool to 'warn' user of inefficient (for I/O) construct in data model

03-12-2015 ROOT::TSeq::GetSize() or ROOT::seq::size()?

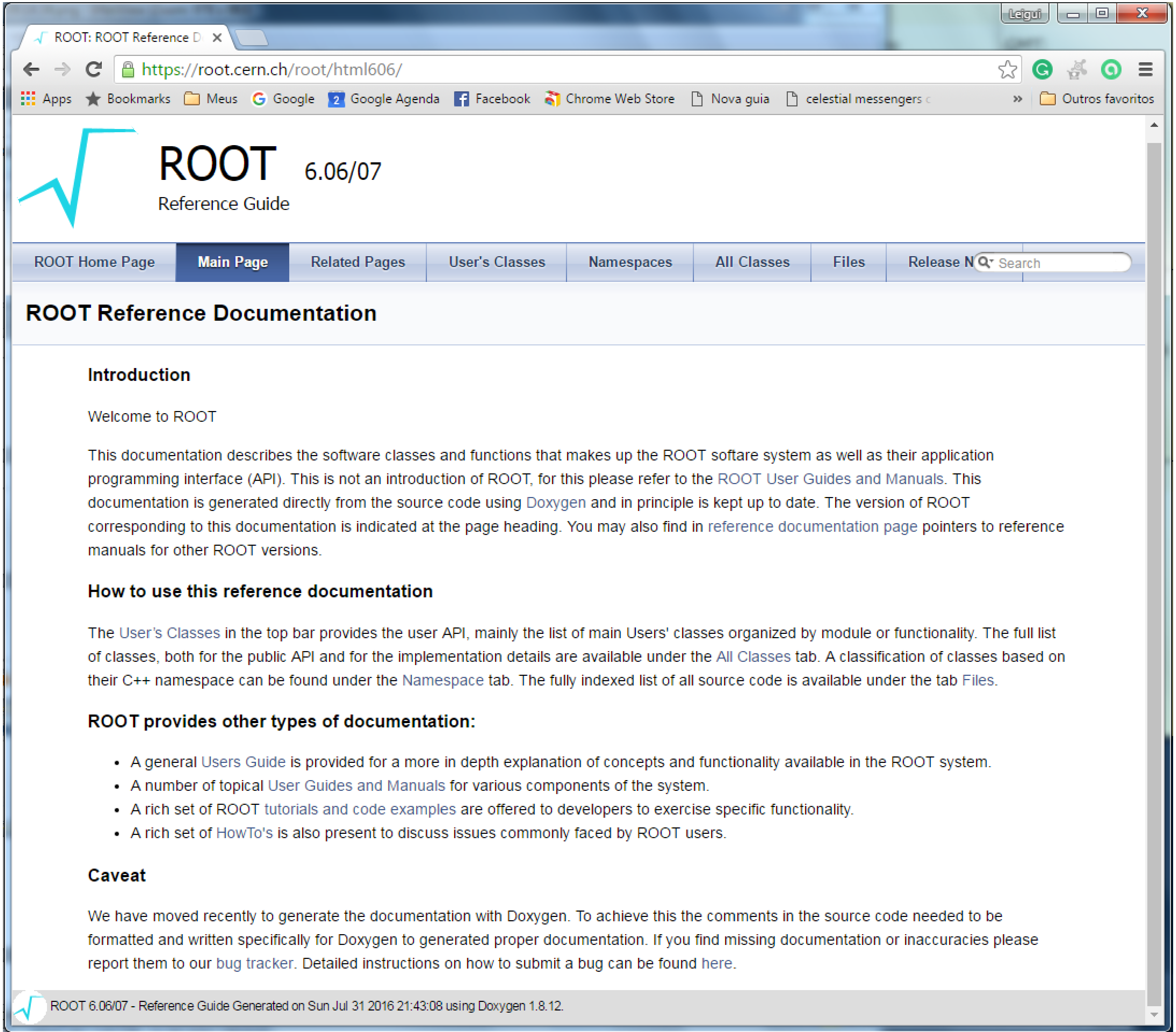
02-09-2015 Wanted: Storage of HEP data via key/value storage solutions

Latest Releases

Release 6.06/06 - 2016-07-06

Release 6.04/18 - 2016-06-22

Release 6.06/04 - 2016-05-03



ROOT 6.06/07

Reference Guide

ROOT Reference Documentation

Introduction

Welcome to ROOT

This documentation describes the software classes and functions that makes up the ROOT softare system as well as their application programming interface (API). This is not an introduction of ROOT, for this please refer to the ROOT User Guides and Manuals. This documentation is generated directly from the source code using Doxygen and in principle is kept up to date. The version of ROOT corresponding to this documentation is indicated at the page heading. You may also find in reference documentation page pointers to reference manuals for other ROOT versions.

How to use this reference documentation

The User's Classes in the top bar provides the user API, mainly the list of main Users' classes organized by module or functionality. The full list of classes, both for the public API and for the implementation details are available under the All Classes tab. A classification of classes based on their C++ namespace can be found under the Namespace tab. The fully indexed list of all source code is available under the tab Files.

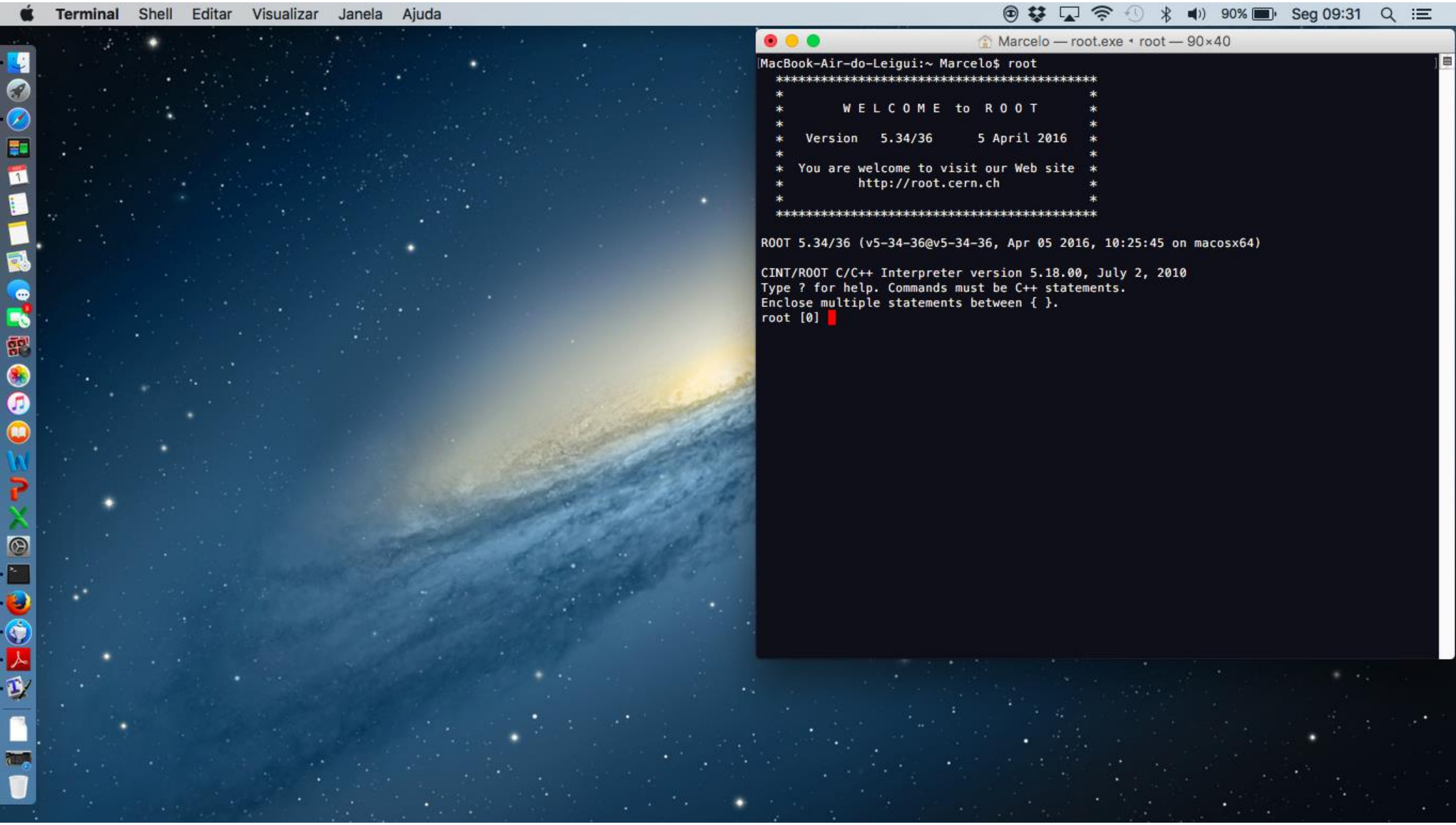
ROOT provides other types of documentation:

- A general Users Guide is provided for a more in depth explanation of concepts and functionality available in the ROOT system.
- A number of topical User Guides and Manuals for various components of the system.
- A rich set of ROOT tutorials and code examples are offered to developers to exercise specific functionality.
- A rich set of HowTo's is also present to discuss issues commonly faced by ROOT users.

Caveat

We have moved recently to generate the documentation with Doxygen. To achieve this the comments in the source code needed to be formatted and written specifically for Doxygen to generated proper documentation. If you find missing documentation or inaccuracies please report them to our bug tracker. Detailed instructions on how to submit a bug can be found here.

ROOT 6.06/07 - Reference Guide Generated on Sun Jul 31 2016 21:43:08 using Doxygen 1.8.12.



```
MacBook-Air-do-Leigui:~ Marcelo$ root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] 
```

Chapter 2

ROOT Basics

Now that you have installed ROOT, what's this interactive shell thing you're running ? It's like this: ROOT leads a double life. It has an interpreter for macros (Cint (Goto 2005)) that you can run from the command line or run like applications. But it is also an interactive shell that can evaluate arbitrary statements and expressions. This is extremely useful for debugging, quick hacking and testing. Let us first have a look at some very simple examples.

2.1 ROOT as calculator

You can even use the ROOT interactive shell in lieu of a calculator! Launch the ROOT interactive shell with the command

```
 > root
```

on your Linux box. The prompt should appear shortly:

```
 root [1]
```

and let's dive in with the steps shown here:

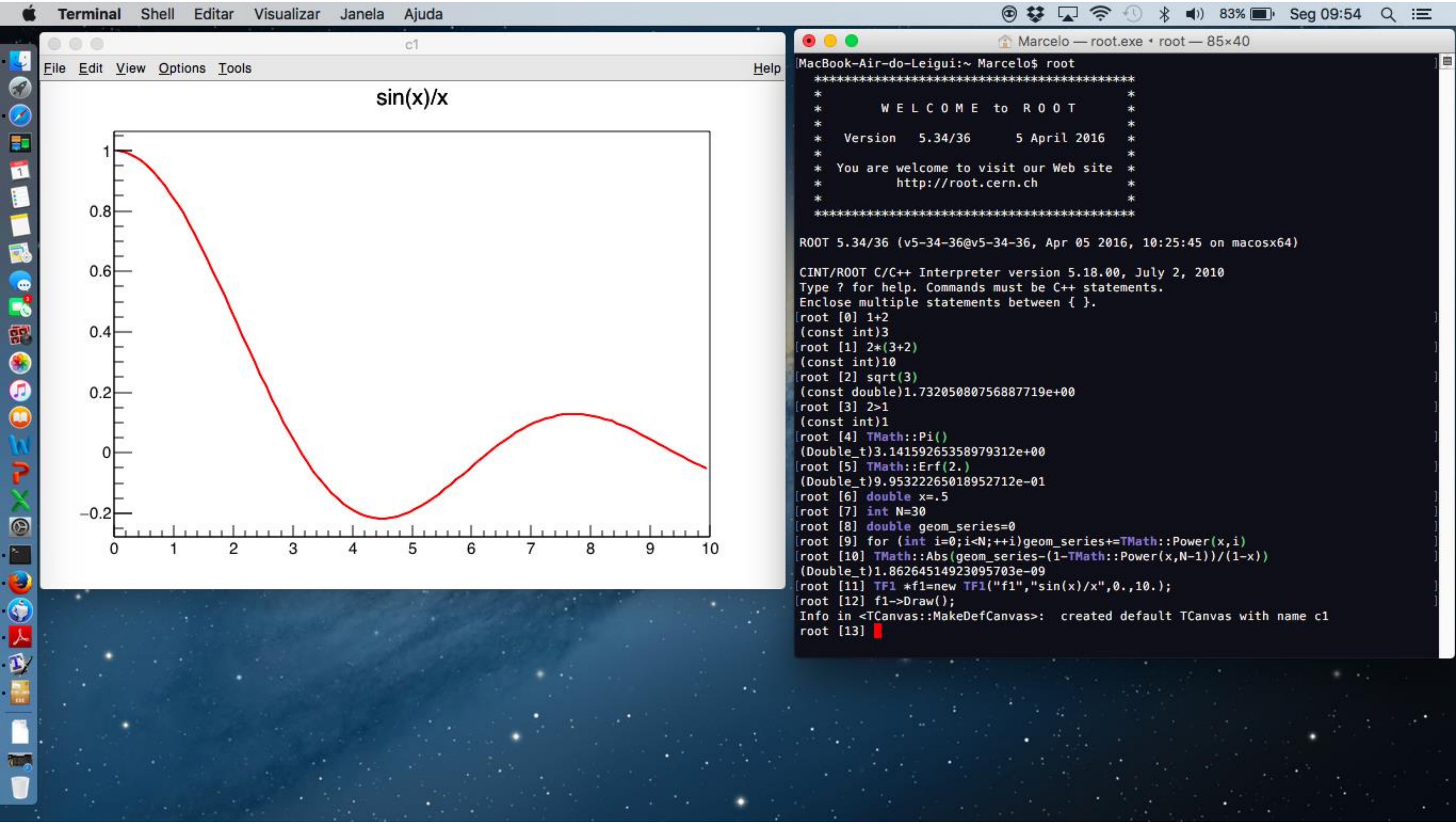
```
root [0] 1+1
(const int)2
root [1] 2*(4+2)/12.
(const double)1.00000000000000000e+00
root [2] sqrt(3)
(const double)1.73205080756887719e+00
root [3] 1 > 2
(const int)0
root [4] TMath::Pi()
(Double_t)3.14159265358979312e+00
root [5] TMath::Erf(.2)
(Double_t)2.22702589210478447e-01
```

Not bad. You can see that ROOT offers you the possibility not only to type in C++ statements, but also advanced mathematical functions, which live in the TMath namespace.

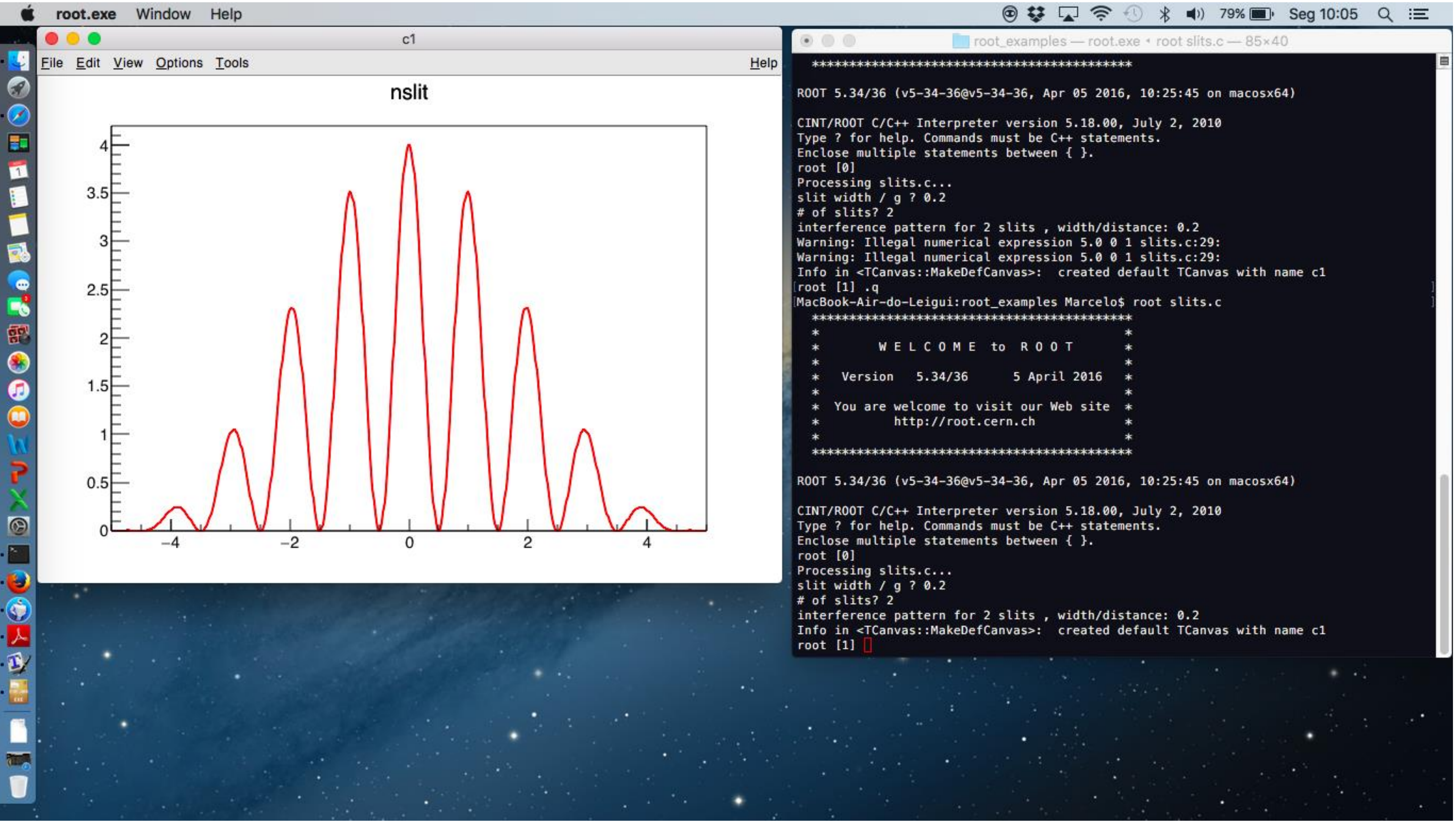
Now let's do something more elaborated. A numerical example with the well known geometrical series:

```
root [6] double x=.5
root [7] int N=30
root [8] double geom_series=0
root [9] for (int i=0;i<N;++i)geom_series+=TMath::Power(x,i)
root [10] TMath::Abs(geom_series - (1-TMath::Power(x,N-1))/(1-x))
(Double_t)1.86264514923095703e-09
```

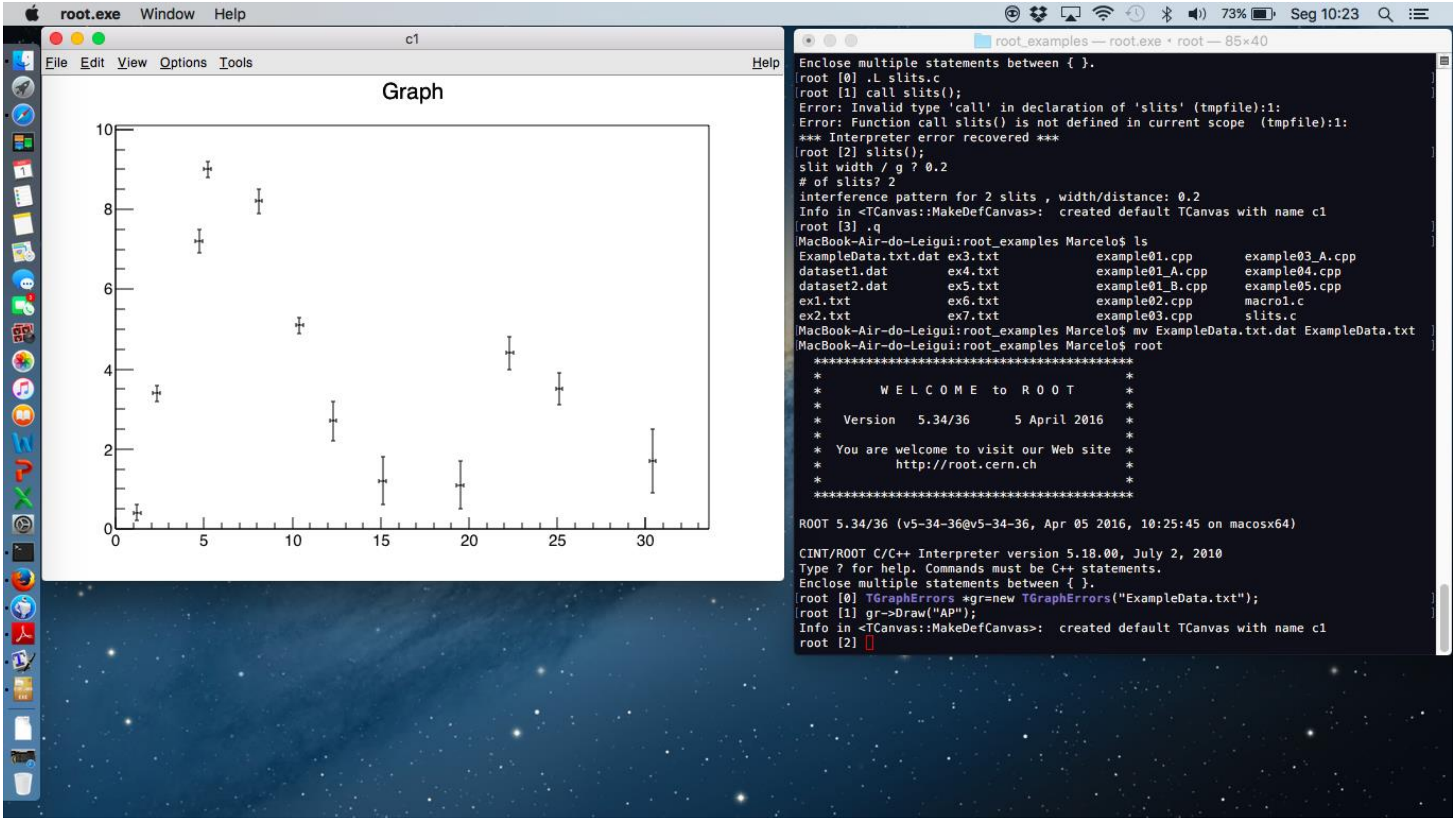
Here we made a step forward. We even declared variables and used a *for* control structure. Note that there are some subtle differences between Cint and the standard C++ language. You do not need the ";" at the end of line in interactive mode – try the difference e.g. using the command at line root [6].



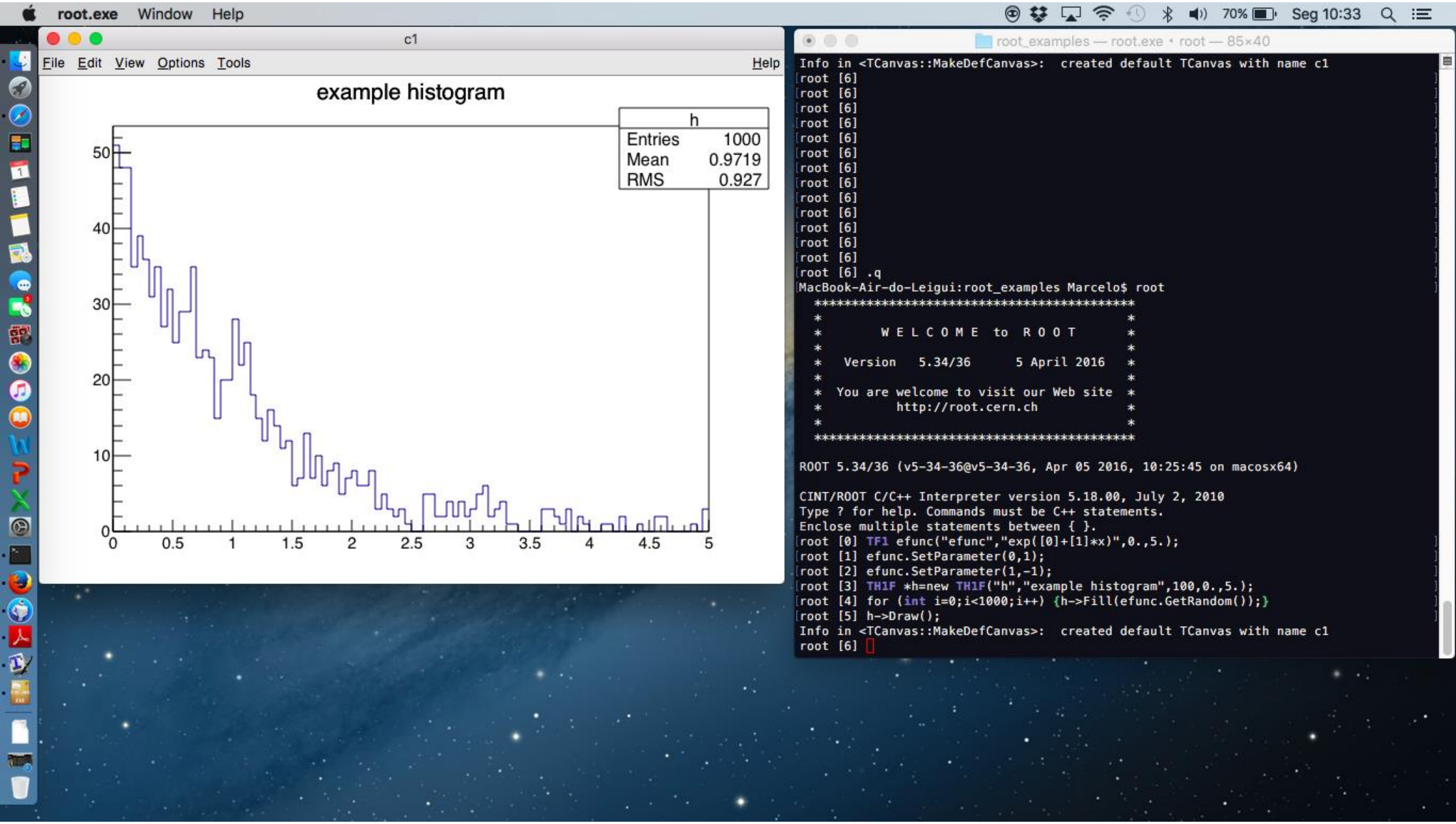
```
[MacBook-Air-do-Leigui:~ Marcelo$ root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
[root [0] 1+2
(const int)3
[root [1] 2*(3+2)
(const int)10
[root [2] sqrt(3)
(const double)1.73205080756887719e+00
[root [3] 2>1
(const int)1
[root [4] TMath::Pi()
(Double_t)3.14159265358979312e+00
[root [5] TMath::Erf(2.)
(Double_t)9.95322265018952712e-01
[root [6] double x=.5
[root [7] int N=30
[root [8] double geom_series=0
[root [9] for (int i=0;i<N;++i)geom_series+=TMath::Power(x,i)
[root [10] TMath::Abs(geom_series-(1-TMath::Power(x,N-1))/(1-x))
(Double_t)1.86264514923095703e-09
[root [11] TF1 *f1=new TF1("f1","sin(x)/x",0.,10.);
[root [12] f1->Draw();
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [13] 
```



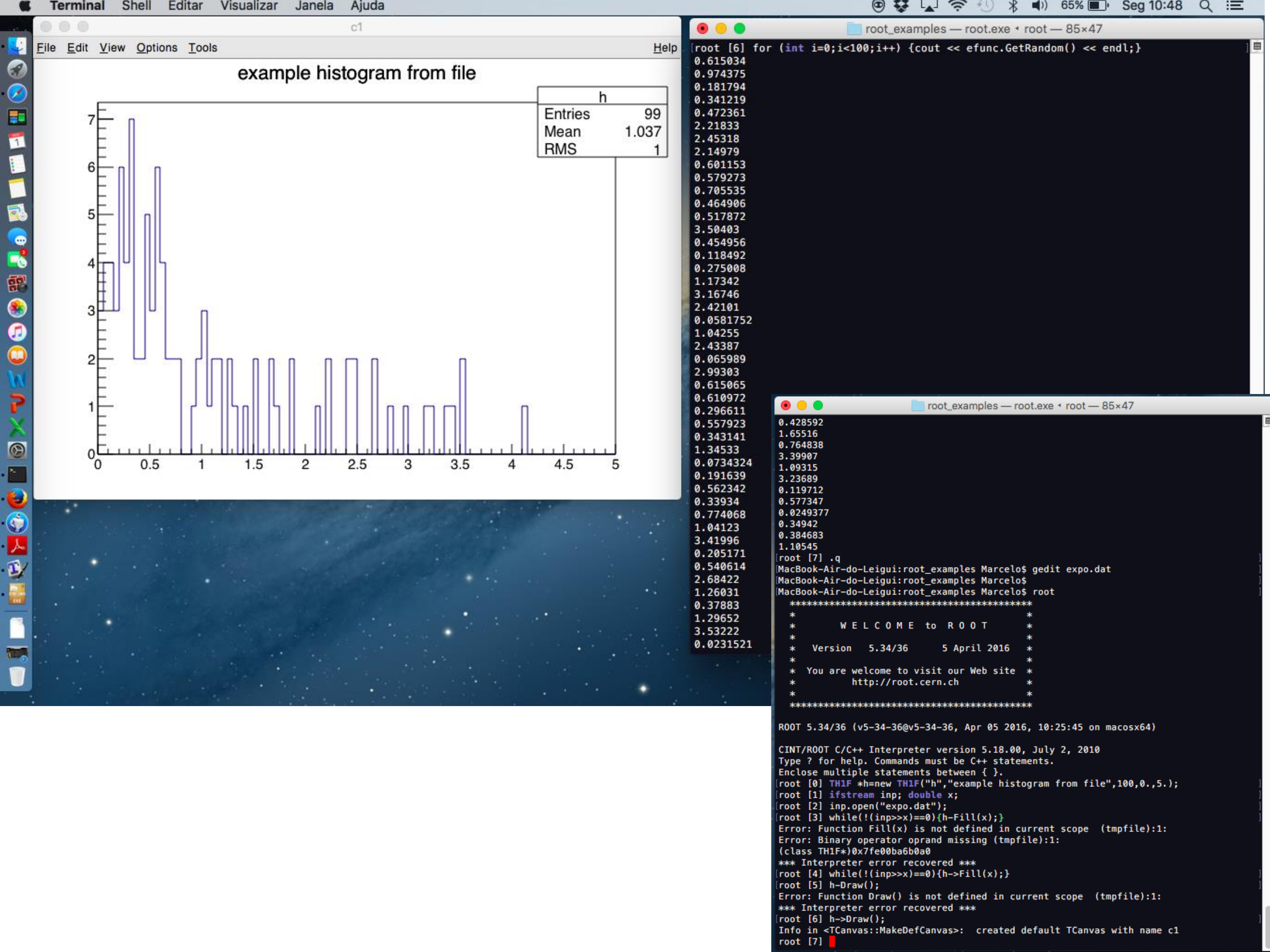
```
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing slits.c...
slit width / g ? 0.2
# of slits? 2
interference pattern for 2 slits , width/distance: 0.2
Warning: Illegal numerical expression 5.0 0 1 slits.c:29:
Warning: Illegal numerical expression 5.0 0 1 slits.c:29:
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [1] .q
MacBook-Air-do-Leigui:root_examples Marcelo$ root slits.c
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing slits.c...
slit width / g ? 0.2
# of slits? 2
interference pattern for 2 slits , width/distance: 0.2
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [1]
```



```
Enclose multiple statements between { }.
root [0] .L slits.c
root [1] call slits();
Error: Invalid type 'call' in declaration of 'slits' (tmpfile):1:
Error: Function call slits() is not defined in current scope  (tmpfile):1:
*** Interpreter error recovered ***
root [2] slits();
slit width / g ? 0.2
# of slits? 2
interference pattern for 2 slits , width/distance: 0.2
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [3] .q
MacBook-Air-do-Leigui:root_examples Marcelo$ ls
ExampleData.txt.dat ex3.txt             example01.cpp       example03_A.cpp
dataset1.dat        ex4.txt             example01_A.cpp     example04.cpp
dataset2.dat        ex5.txt             example01_B.cpp     example05.cpp
ex1.txt             ex6.txt             example02.cpp       macro1.c
ex2.txt             ex7.txt             example03.cpp       slits.c
MacBook-Air-do-Leigui:root_examples Marcelo$ mv ExampleData.txt.dat ExampleData.txt
MacBook-Air-do-Leigui:root_examples Marcelo$ root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] TGraphErrors *gr=new TGraphErrors("ExampleData.txt");
root [1] gr->Draw("AP");
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [2] 
```

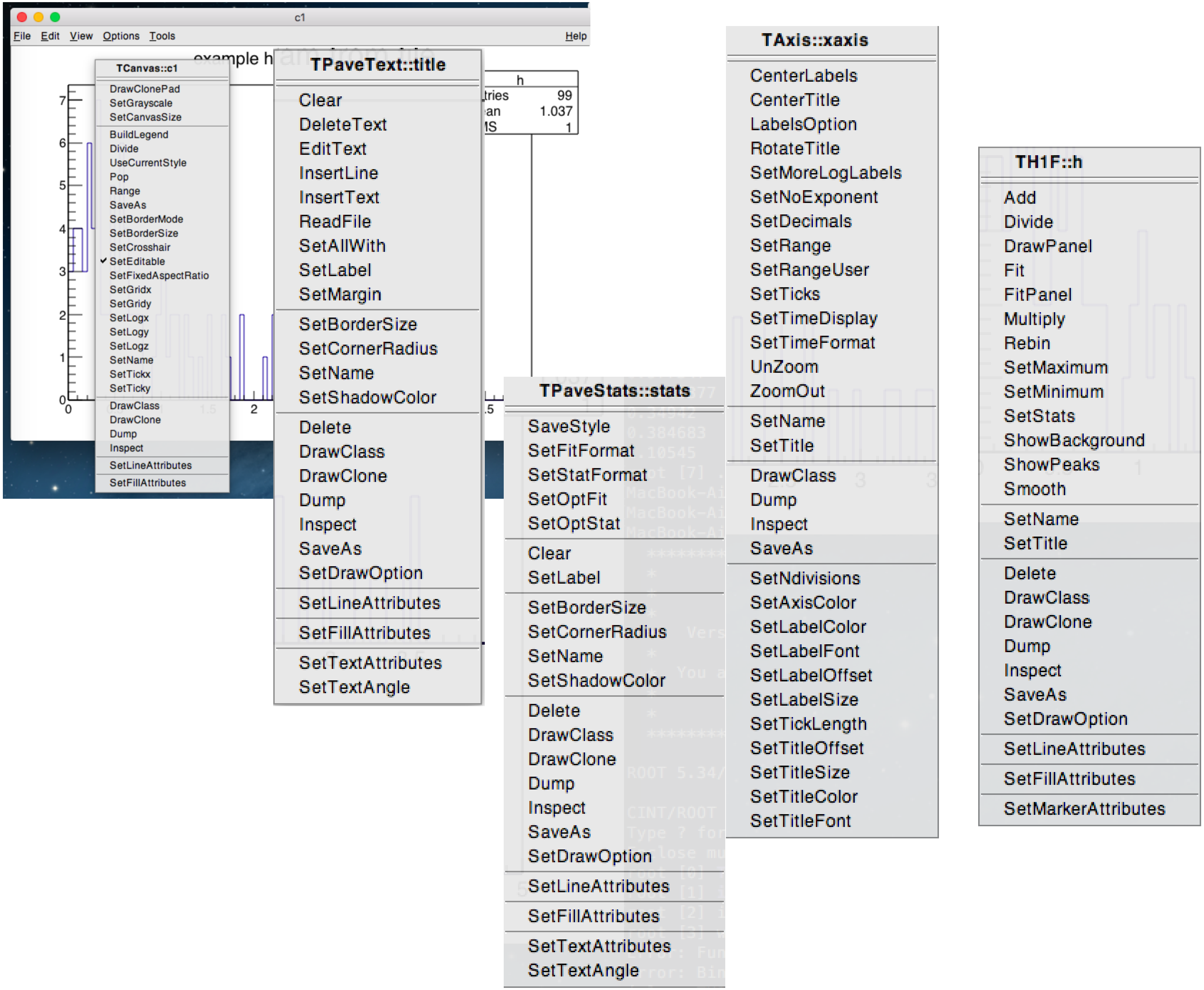


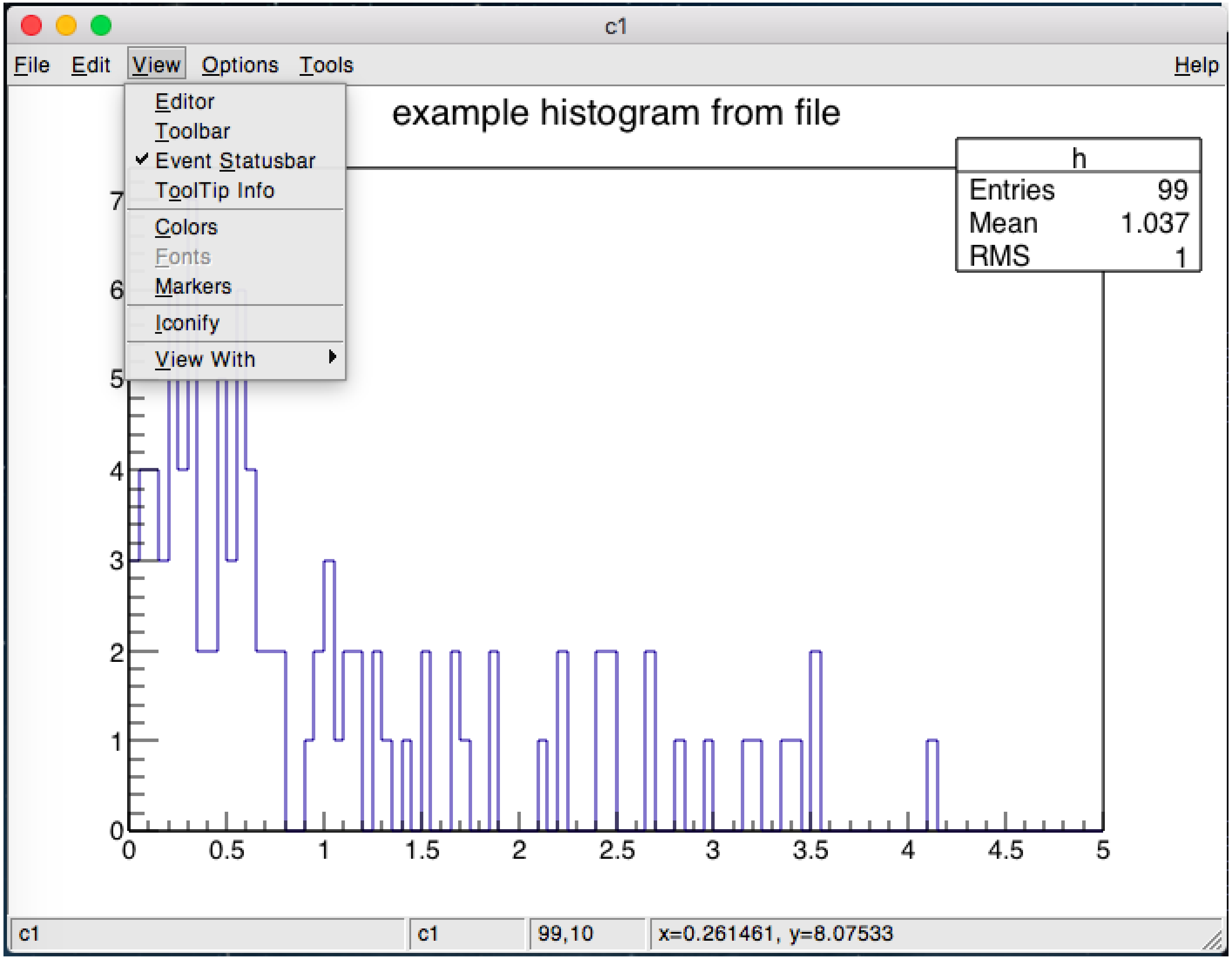
```
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6]
root [6] .q
MacBook-Air-do-Leigui:root_examples Marcelo$ root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
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();
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [6]
```


Terminal Shell Editar Visualizar Janela Ajuda
65% Seg 10:48
c1
File Edit View Options Tools Help
example histogram from file
h
Entries 99
Mean 1.037
RMS 1
0
1
2
3
4
5
6
7
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
root_examples — root.exe ◂ root — 85×47
root_examples — root.exe ◂ root — 85×47

```
root [6] for (int i=0;i<100;i++) {cout << efunc.GetRandom() << endl;}
0.615034
0.974375
0.181794
0.341219
0.472361
2.21833
2.45318
2.14979
0.601153
0.579273
0.705535
0.464906
0.517872
3.50403
0.454956
0.118492
0.275008
1.17342
3.16746
2.42101
0.0581752
1.04255
2.43387
0.065989
2.99303
0.615065
0.610972
0.296611
0.557923
0.343141
1.34533
0.0734324
0.191639
0.562342
0.33934
0.774068
1.04123
3.41996
0.205171
0.540614
2.68422
1.26031
0.37883
1.29652
3.53222
0.0231521
```

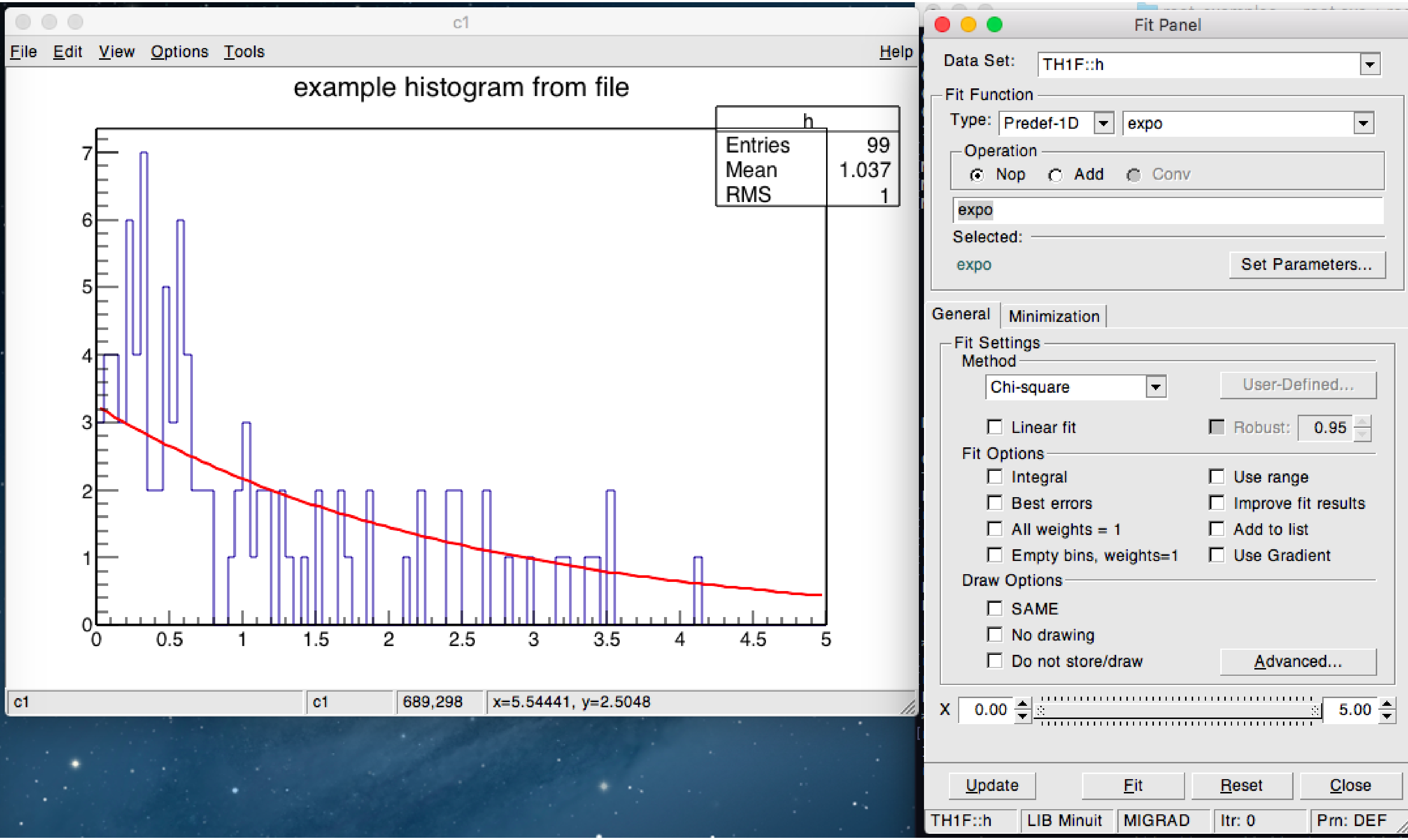
```
0.428592
1.65516
0.764838
3.39907
1.09315
3.23689
0.119712
0.577347
0.0249377
0.34942
0.384683
1.10545
root [7] .q
MacBook-Air-do-Leigui:root_examples Marcelo$ gedit expo.dat
MacBook-Air-do-Leigui:root_examples Marcelo$
MacBook-Air-do-Leigui:root_examples Marcelo$ root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] TH1F *h=new TH1F("h","example histogram from file",100,0.,5.);
root [1] ifstream inp; double x;
root [2] inp.open("expo.dat");
root [3] while(!(inp>>x)==0){h-Fill(x);}
Error: Function Fill(x) is not defined in current scope  (tmpfile):1:
Error: Binary operator oprand missing (tmpfile):1:
(class TH1F*)0x7fe00ba6b0a0
*** Interpreter error recovered ***
root [4] while(!(inp>>x)==0){h->Fill(x);}
root [5] h-Draw();
Error: Function Draw() is not defined in current scope  (tmpfile):1:
*** Interpreter error recovered ***
root [6] h->Draw();
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
root [7]
```


c1
File Edit View Options Tools Help
example h
h
tries 99
ean 1.037
IS 1
TCanvas::c1
DrawClonePad
SetGrayscale
SetCanvasSize
BuildLegend
Divide
UseCurrentStyle
Pop
Range
SaveAs
SetBorderMode
SetBorderSize
SetCrosshair
✓ SetEditable
SetFixedAspectRatio
SetGridx
SetGridy
SetLogx
SetLogy
SetLogz
SetName
SetTickx
SetTicky
DrawClass
DrawClone
Dump
Inspect
SetLineAttributes
SetFillAttributes
TPaveText::title
Clear
DeleteText
EditText
InsertLine
InsertText
ReadFile
SetAllWith
SetLabel
SetMargin
SetBorderSize
SetCornerRadius
SetName
SetShadowColor
Delete
DrawClass
DrawClone
Dump
Inspect
SaveAs
SetDrawOption
SetLineAttributes
SetFillAttributes
SetTextAttributes
SetTextAngle
TPaveStats::stats
SaveStyle
SetFitFormat
SetStatFormat
SetOptFit
SetOptStat
Clear
SetLabel
SetBorderSize
SetCornerRadius
SetName
SetShadowColor
Delete
DrawClass
DrawClone
Dump
Inspect
SaveAs
SetDrawOption
SetLineAttributes
SetFillAttributes
SetTextAttributes
SetTextAngle
TAxis::xaxis
CenterLabels
CenterTitle
LabelsOption
RotateTitle
SetMoreLogLabels
SetNoExponent
SetDecimals
SetRange
SetRangeUser
SetTicks
SetTimeDisplay
SetTimeFormat
UnZoom
ZoomOut
SetName
SetTitle
DrawClass
Dump
Inspect
SaveAs
SetNdivisions
SetAxisColor
SetLabelColor
SetLabelFont
SetLabelOffset
SetLabelSize
SetTickLength
SetTitleOffset
SetTitleSize
SetTitleColor
SetTitleFont
TH1F::h
Add
Divide
DrawPanel
Fit
FitPanel
Multiply
Rebin
SetMaximum
SetMinimum
SetStats
ShowBackground
ShowPeaks
Smooth
SetName
SetTitle
Delete
DrawClass
DrawClone
Dump
Inspect
SaveAs
SetDrawOption
SetLineAttributes
SetFillAttributes
SetMarkerAttributes



h	
Entries	99
Mean	1.037
RMS	1

```
c1    c1    99,10    x=0.261461, y=8.07533
```



h	
Entries	99
Mean	1.037
RMS	1

c1 | c1 | 689,298 | x=5.54441, y=2.5048

Data Set: TH1F::h

Fit Function

Type: Predef-1D expo

Operation

● Nop ○ Add ○ Conv

expo

Selected:

expo

Fit Settings

Method

Chi-square

□ Linear fit ▨ Robust: 0.95

Fit Options

□ Integral □ Use range

□ Best errors □ Improve fit results

□ All weights = 1 □ Add to list

□ Empty bins, weights=1 □ Use Gradient

Draw Options

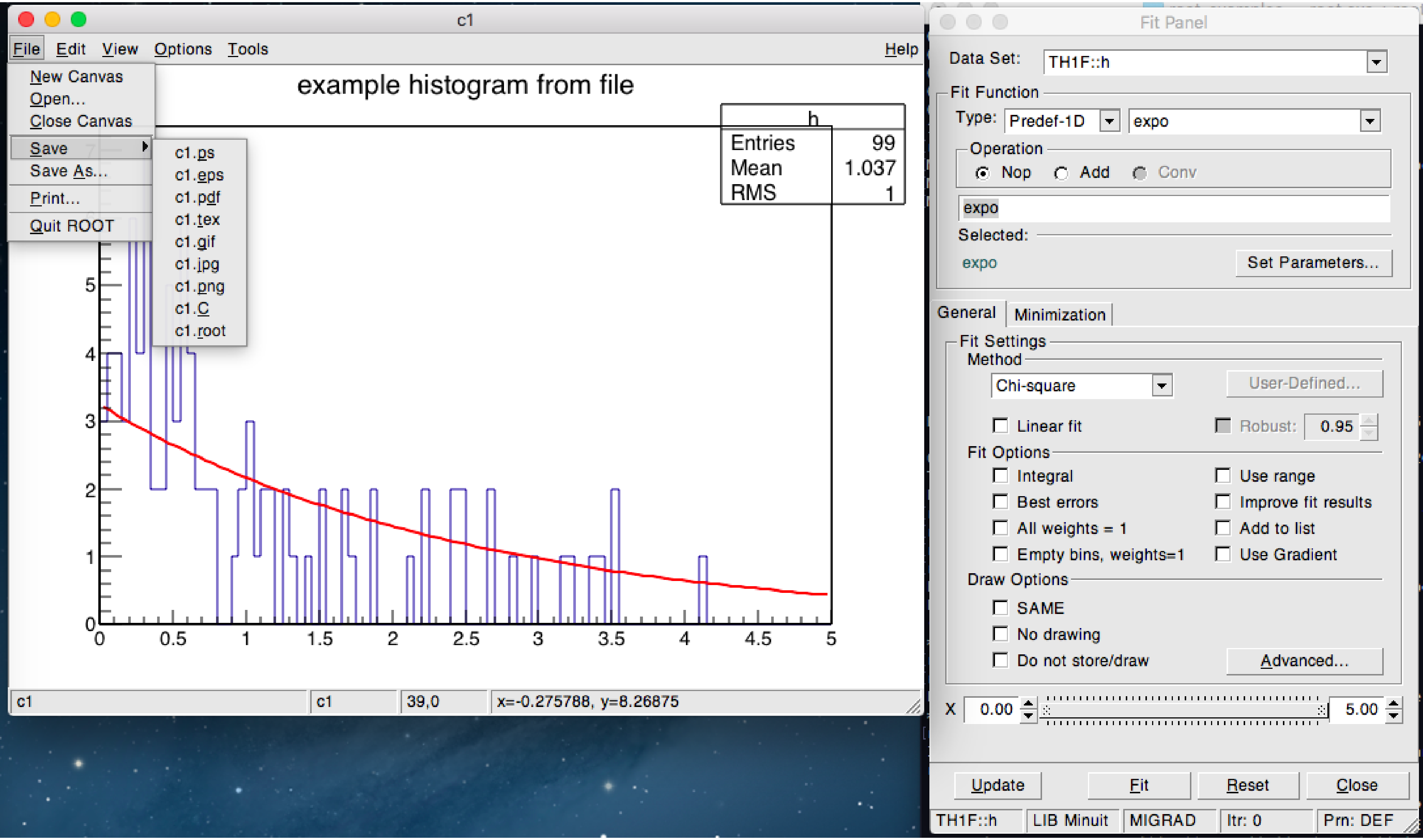
□ SAME

□ No drawing

□ Do not store/draw

X 0.00 5.00

TH1F::h | LIB Minuit | MIGRAD | Itr: 0 | Prn: DEF



c1 | c1 | 39,0 | x=-0.275788, y=8.26875

Data Set: TH1F::h

Fit Function

Type: Predef-1D expo

Operation

● Nop ○ Add ○ Conv

expo

Selected:

expo

Fit Settings

Method

Chi-square

□ Linear fit □ Robust: 0.95

Fit Options

□ Integral □ Use range

□ Best errors □ Improve fit results

□ All weights = 1 □ Add to list

□ Empty bins, weights=1 □ Use Gradient

Draw Options

□ SAME

□ No drawing

□ Do not store/draw

X 0.00 5.00

TH1F::h | LIB Minuit | MIGRAD | Itr: 0 | Prn: DEF

Chapter 3

ROOT Macros

You know how other books go on and on about programming fundamentals and finally work up to building a complete, working program ? Let's skip all that. In this guide, we will describe macros executed by the ROOT C++ interpreter CINT.

It is relatively easy to compile a macro, either as a pre-compiled library to load into ROOT, or as a stand-alone application, by adding some include statements for header file or some "dressing code" to any macro.

3.1 General Remarks on ROOT macros

If you have a number of lines which you were able to execute at the ROOT prompt, they can be turned into a ROOT macro by giving them a name which corresponds to the file name without extension. The general structure for a macro stored in file `MacroName.C` is

```
void MacroName() {
        <          ...
          your lines of C++ code
                   ...             >
}
```

The macro is executed by typing

```
 > root MacroName.C
```

at the system prompt, or it can be loaded into a ROOT session and then be executed by typing

```
root [0].L MacroName.C
root [1] MacroName();
```

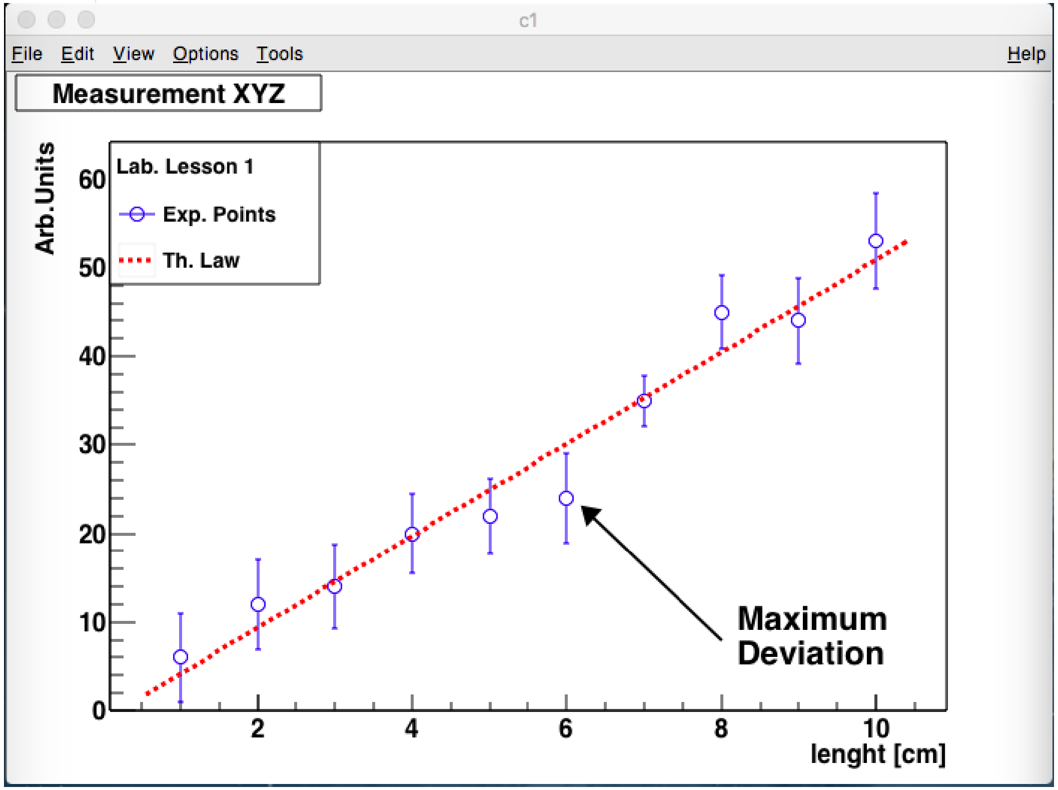
at the ROOT prompt. Note that more than one macro can be loaded this way, as each macro has a unique name in the ROOT name space. A small set of options can help making your plot nicer.

```
gROOT->SetStyle("Plain");   // set plain TStyle
gStyle->SetOptStat(111111); // draw statistics on plots,
                            // (0) for no output
gStyle->SetOptFit(1111);    // draw fit results on plot,
                            // (0) for no ouput
gStyle->SetPalette(53);     // set color map
gStyle->SetOptTitle(0);     // suppress title box
   ...
```

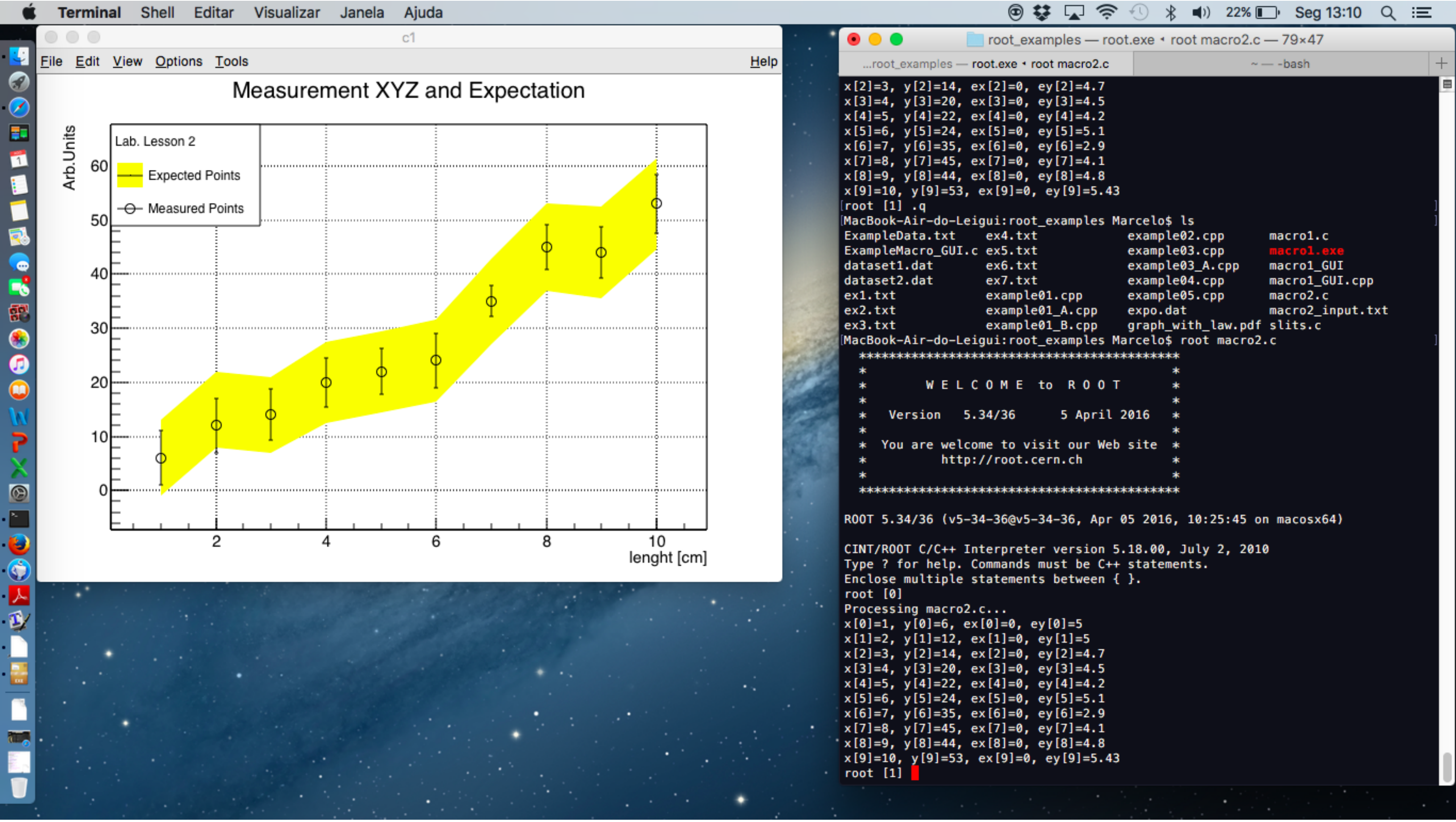
Next, you should create a canvas for graphical output, with size, subdivisions and format suitable to your needs, see documentation of class `TCanvas`:

```
/***** 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
gROOT->SetStyle("Plain");
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");

// Define a linear function
TF1 f("Linear law","[0]+x*[1]",.5,10.5);
// Let's make the function 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
```

macro1.c

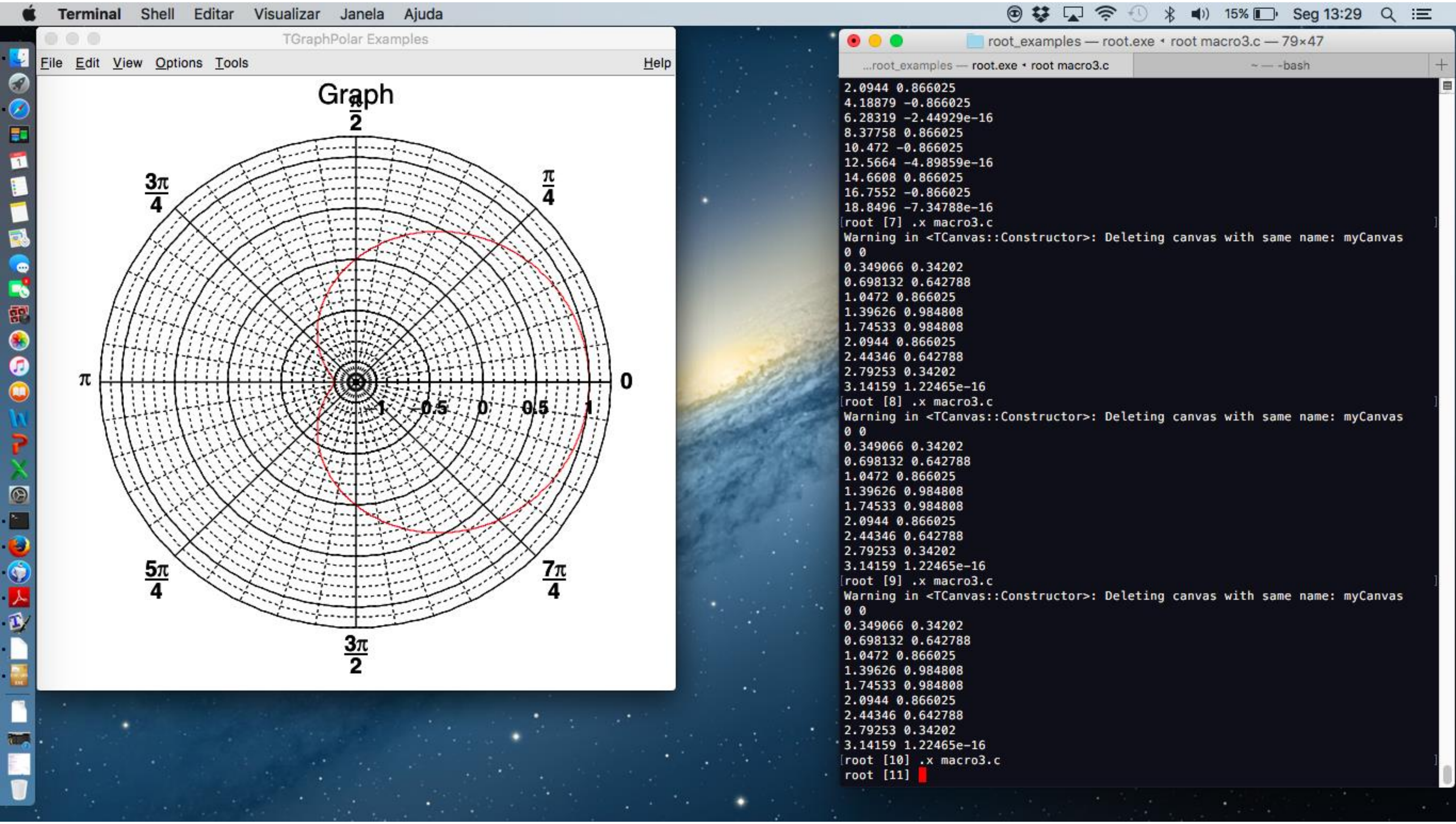


macro2.c



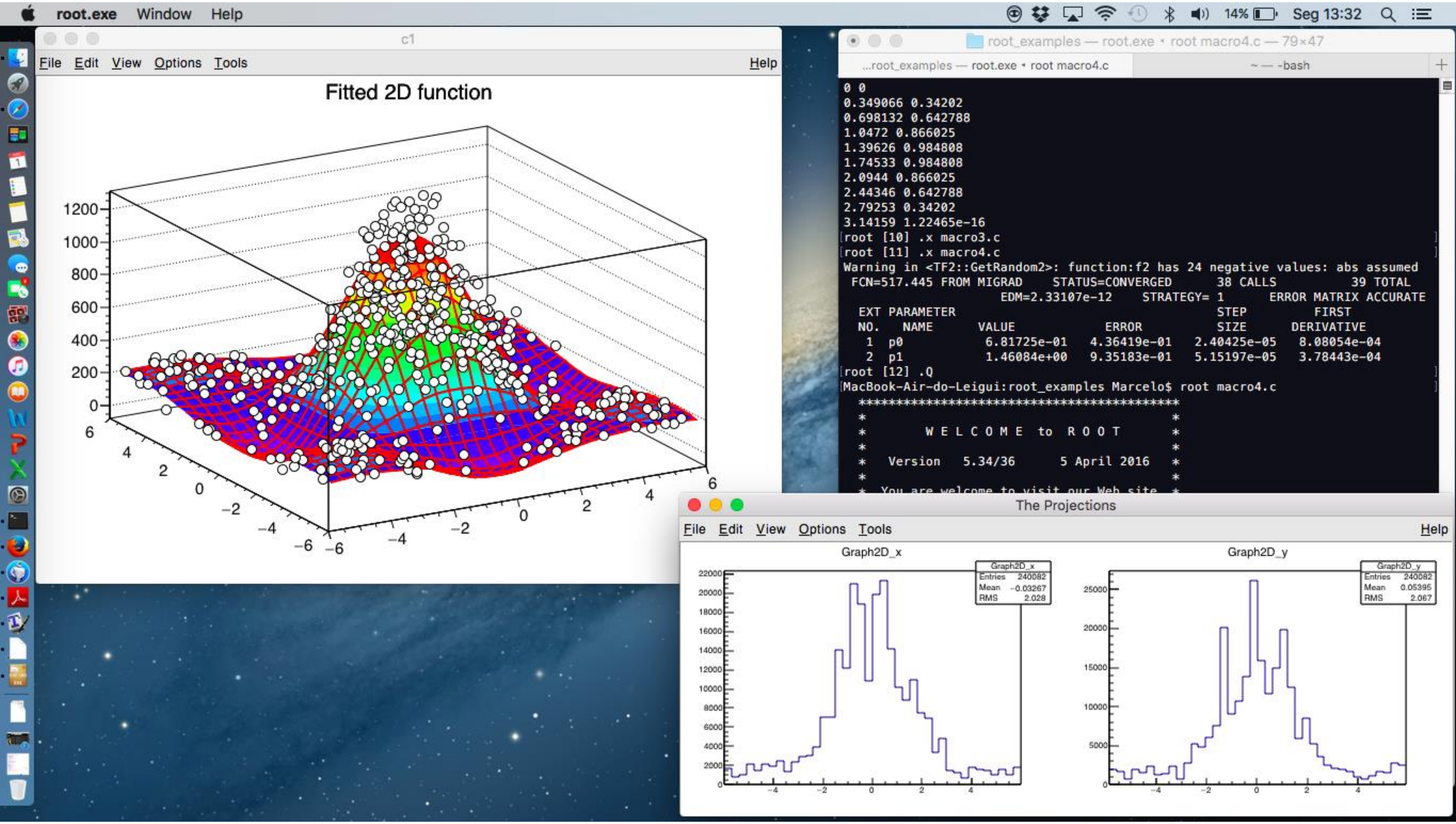
```
x[2]=3, y[2]=14, ex[2]=0, ey[2]=4.7
x[3]=4, y[3]=20, ex[3]=0, ey[3]=4.5
x[4]=5, y[4]=22, ex[4]=0, ey[4]=4.2
x[5]=6, y[5]=24, ex[5]=0, ey[5]=5.1
x[6]=7, y[6]=35, ex[6]=0, ey[6]=2.9
x[7]=8, y[7]=45, ex[7]=0, ey[7]=4.1
x[8]=9, y[8]=44, ex[8]=0, ey[8]=4.8
x[9]=10, y[9]=53, ex[9]=0, ey[9]=5.43
root [1] .q
MacBook-Air-do-Leigui:root_examples Marcelo$ ls
ExampleData.txt     ex4.txt             example02.cpp       macro1.c
ExampleMacro_GUI.c  ex5.txt             example03.cpp       macro1.exe
dataset1.dat        ex6.txt             example03_A.cpp     macro1_GUI
dataset2.dat        ex7.txt             example04.cpp       macro1_GUI.cpp
ex1.txt             example01.cpp       example05.cpp       macro2.c
ex2.txt             example01_A.cpp     expo.dat            macro2_input.txt
ex3.txt             example01_B.cpp     graph_with_law.pdf  slits.c
MacBook-Air-do-Leigui:root_examples Marcelo$ root macro2.c
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing macro2.c...
x[0]=1, y[0]=6, ex[0]=0, ey[0]=5
x[1]=2, y[1]=12, ex[1]=0, ey[1]=5
x[2]=3, y[2]=14, ex[2]=0, ey[2]=4.7
x[3]=4, y[3]=20, ex[3]=0, ey[3]=4.5
x[4]=5, y[4]=22, ex[4]=0, ey[4]=4.2
x[5]=6, y[5]=24, ex[5]=0, ey[5]=5.1
x[6]=7, y[6]=35, ex[6]=0, ey[6]=2.9
x[7]=8, y[7]=45, ex[7]=0, ey[7]=4.1
x[8]=9, y[8]=44, ex[8]=0, ey[8]=4.8
x[9]=10, y[9]=53, ex[9]=0, ey[9]=5.43
root [1]
```

macro3.c



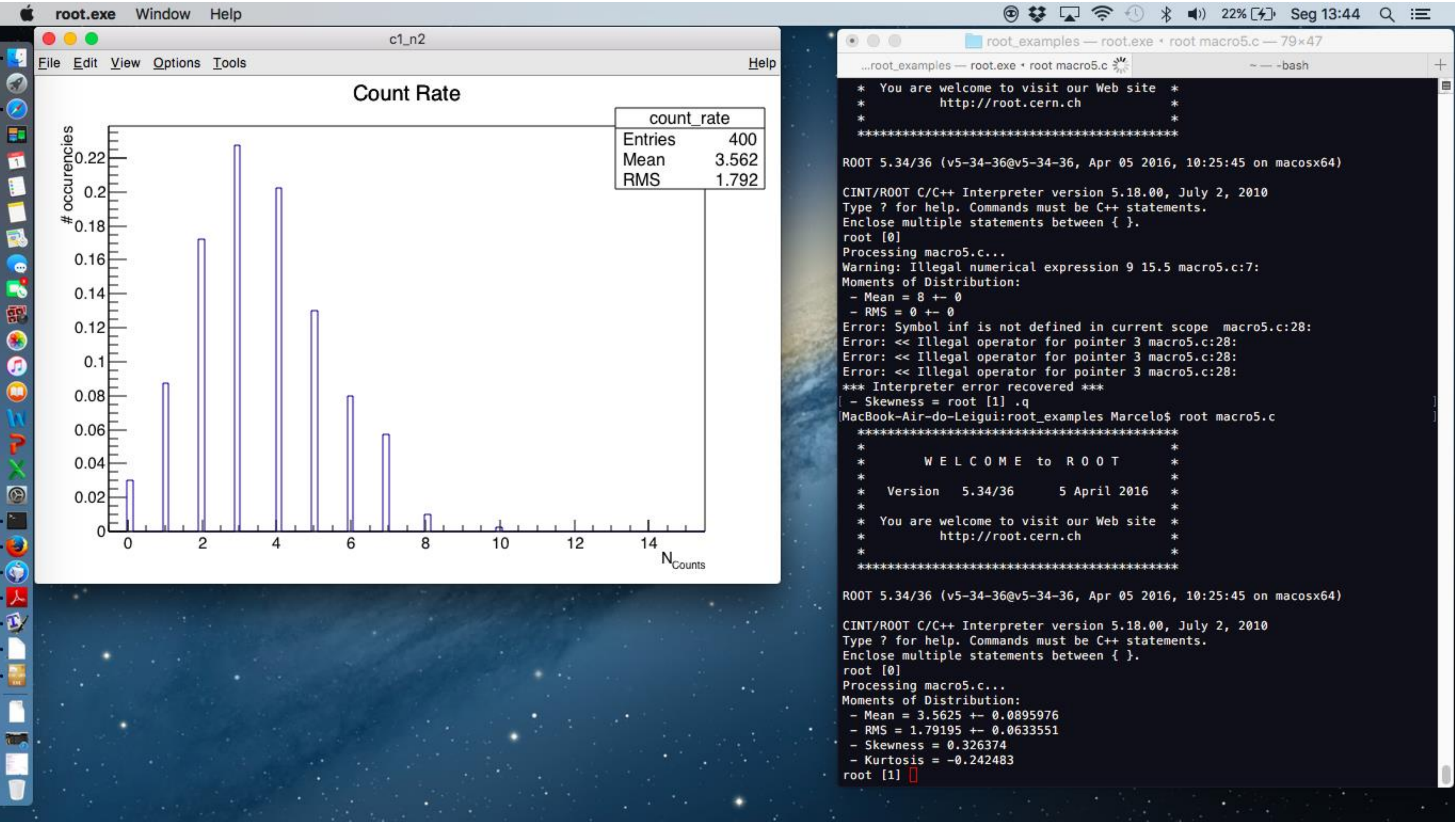
```
2.0944 0.866025
4.18879 -0.866025
6.28319 -2.44929e-16
8.37758 0.866025
10.472 -0.866025
12.5664 -4.89859e-16
14.6608 0.866025
16.7552 -0.866025
18.8496 -7.34788e-16
root [7] .x macro3.c
Warning in <TCanvas::Constructor>: Deleting canvas with same name: myCanvas
0 0
0.349066 0.34202
0.698132 0.642788
1.0472 0.866025
1.39626 0.984808
1.74533 0.984808
2.0944 0.866025
2.44346 0.642788
2.79253 0.34202
3.14159 1.22465e-16
root [8] .x macro3.c
Warning in <TCanvas::Constructor>: Deleting canvas with same name: myCanvas
0 0
0.349066 0.34202
0.698132 0.642788
1.0472 0.866025
1.39626 0.984808
1.74533 0.984808
2.0944 0.866025
2.44346 0.642788
2.79253 0.34202
3.14159 1.22465e-16
root [9] .x macro3.c
Warning in <TCanvas::Constructor>: Deleting canvas with same name: myCanvas
0 0
0.349066 0.34202
0.698132 0.642788
1.0472 0.866025
1.39626 0.984808
1.74533 0.984808
2.0944 0.866025
2.44346 0.642788
2.79253 0.34202
3.14159 1.22465e-16
root [10] .x macro3.c
root [11] 
```

macro4.c



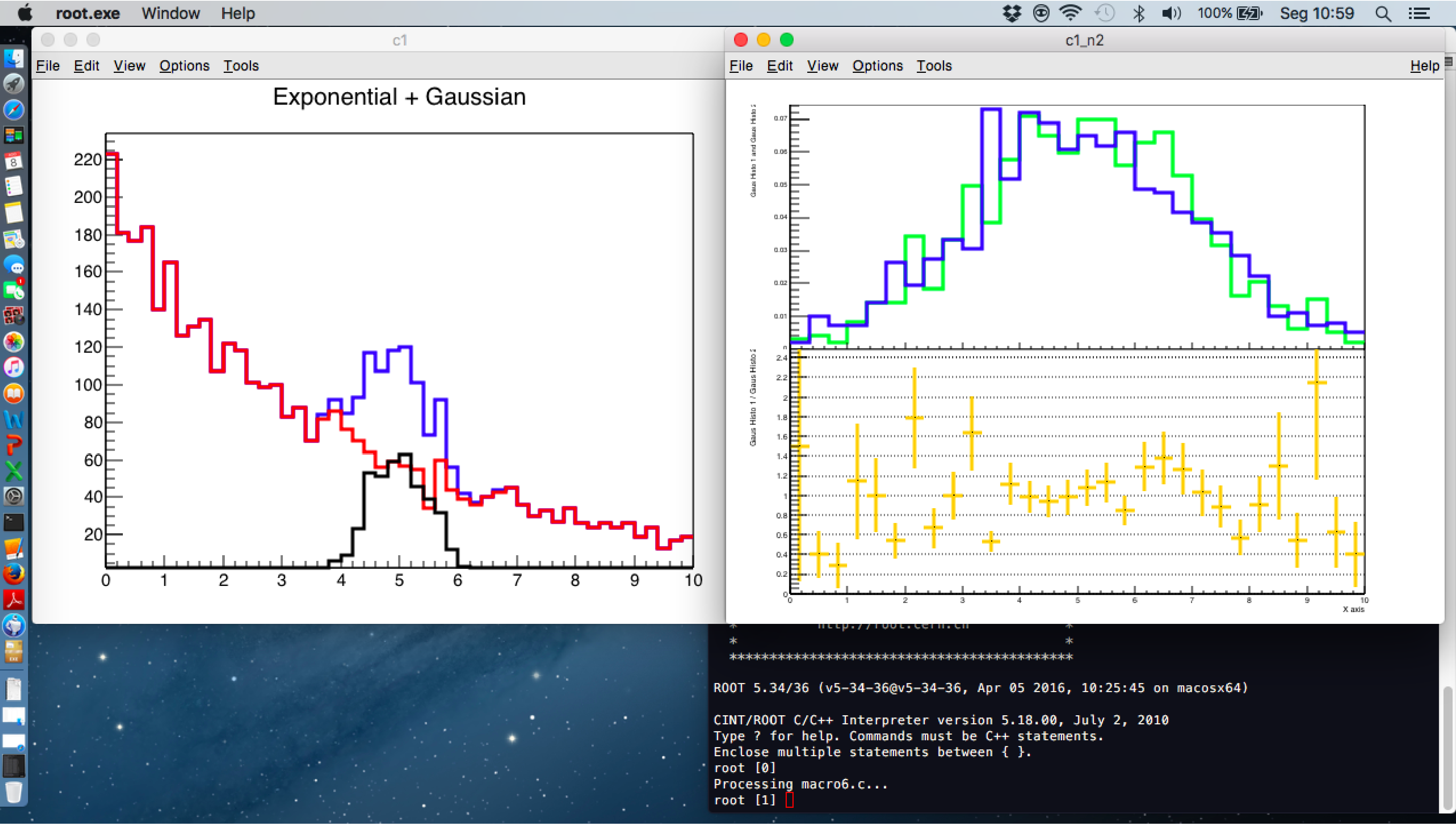
```
0 0
0.349066 0.34202
0.698132 0.642788
1.0472 0.866025
1.39626 0.984808
1.74533 0.984808
2.0944 0.866025
2.44346 0.642788
2.79253 0.34202
3.14159 1.22465e-16
root [10] .x macro3.c
root [11] .x macro4.c
Warning in <TF2::GetRandom2>: function:f2 has 24 negative values: abs assumed
 FCN=517.445 FROM MIGRAD    STATUS=CONVERGED      38 CALLS          39 TOTAL
                     EDM=2.33107e-12    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  p0           6.81725e-01   4.36419e-01   2.40425e-05   8.08054e-04
   2  p1           1.46084e+00   9.35183e-01   5.15197e-05   3.78443e-04
root [12] .Q
MacBook-Air-do-Leigui:root_examples Marcelo$ root macro4.c
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
```

macro5.c

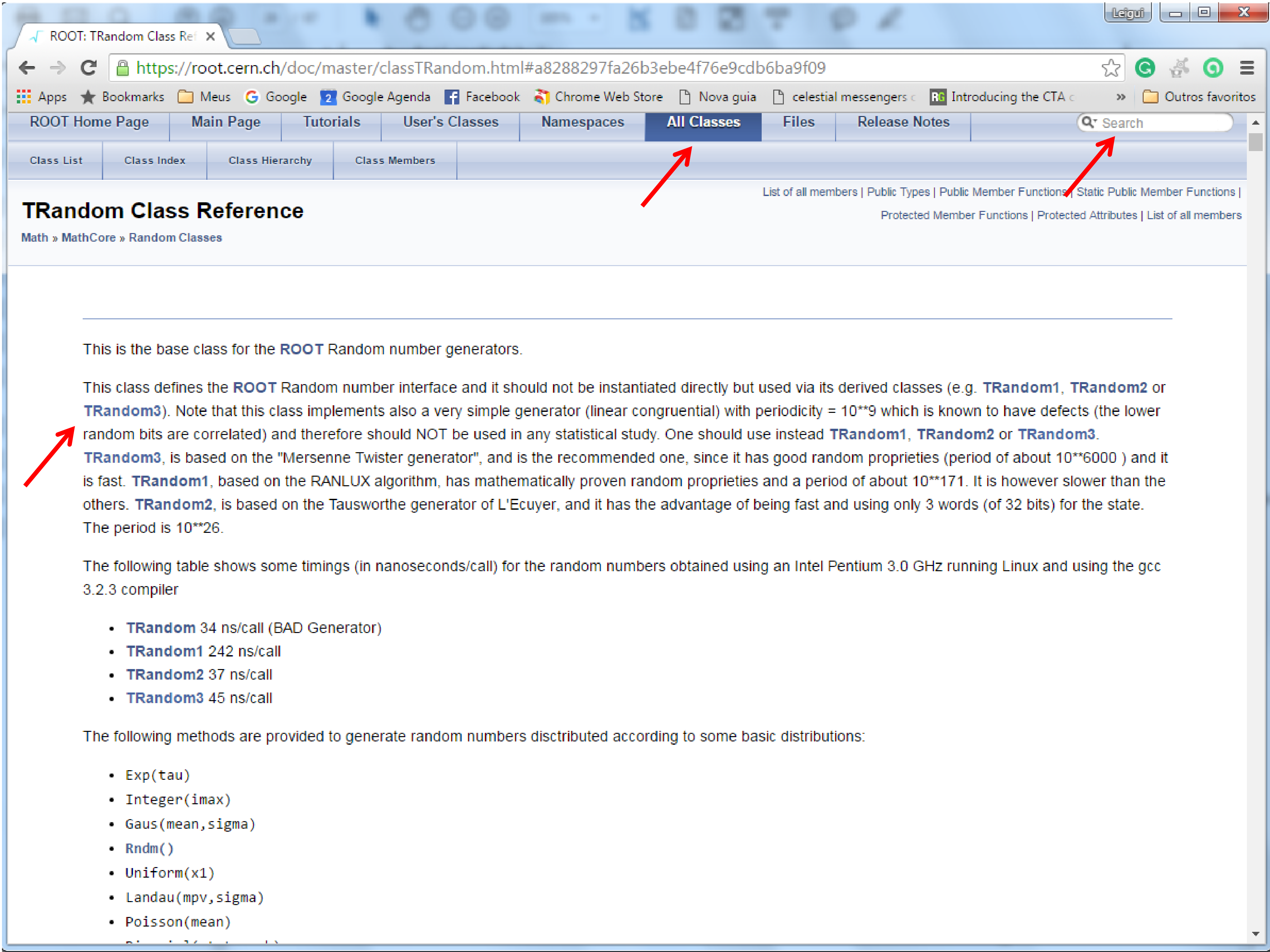


```
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing macro5.c...
Warning: Illegal numerical expression 9 15.5 macro5.c:7:
Moments of Distribution:
 - Mean = 8 +- 0
 - RMS = 0 +- 0
Error: Symbol inf is not defined in current scope  macro5.c:28:
Error: << Illegal operator for pointer 3 macro5.c:28:
Error: << Illegal operator for pointer 3 macro5.c:28:
Error: << Illegal operator for pointer 3 macro5.c:28:
*** Interpreter error recovered ***
 - Skewness = root [1] .q
MacBook-Air-do-Leigui:root_examples Marcelo$ root macro5.c
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/36      5 April 2016   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing macro5.c...
Moments of Distribution:
 - Mean = 3.5625 +- 0.0895976
 - RMS = 1.79195 +- 0.0633551
 - Skewness = 0.326374
 - Kurtosis = -0.242483
root [1]
```

macro6.c



```
  *         http://root.cern.ch            *
  *                                        *
  ******************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing macro6.c...
root [1]
```



TRandom Class Reference

Math » MathCore » Random Classes

This is the base class for the ROOT Random number generators.

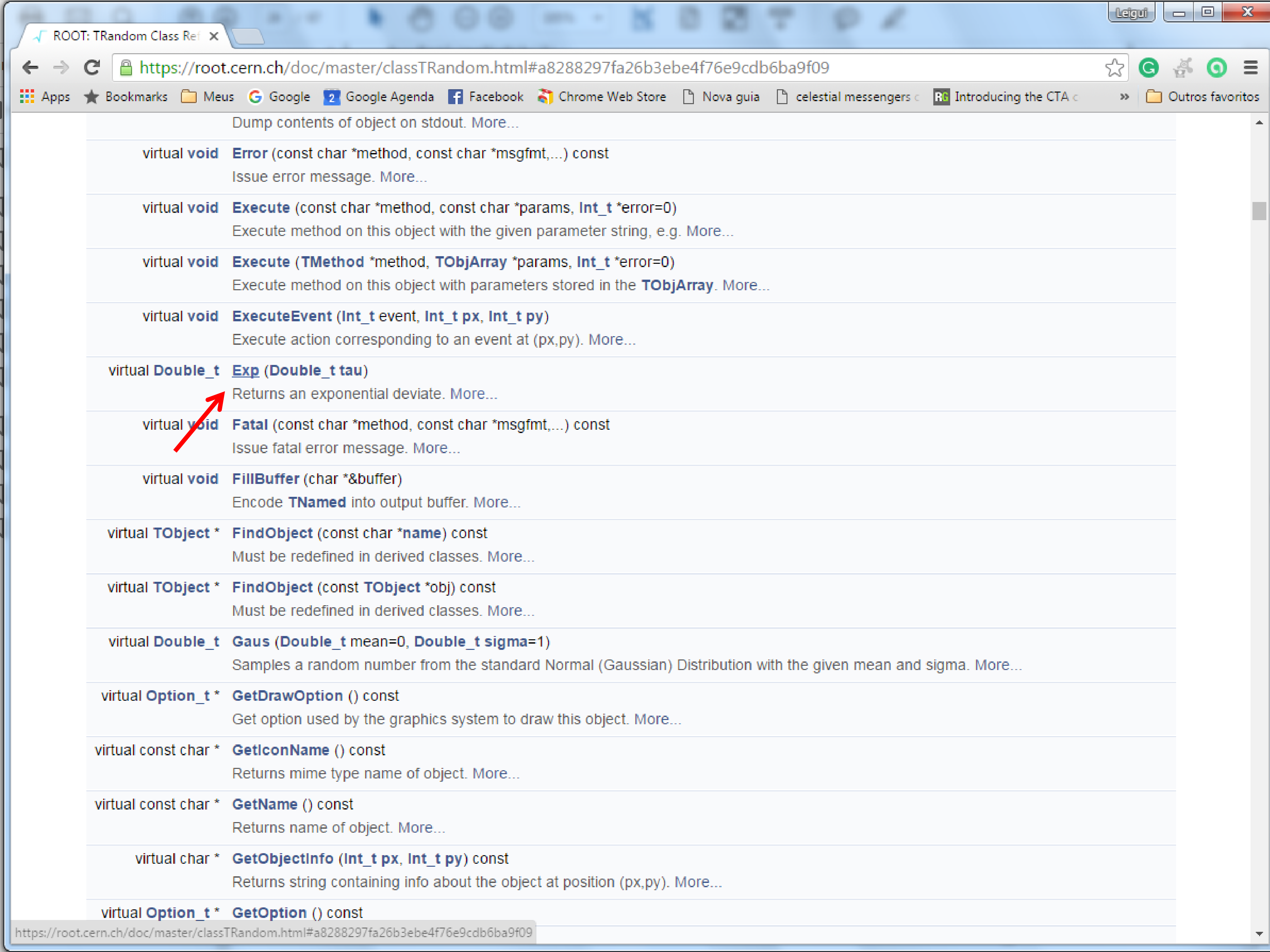
This class defines the ROOT Random number interface and it should not be instantiated directly but used via its derived classes (e.g. TRandom1, TRandom2 or TRandom3). Note that this class implements also a very simple generator (linear congruential) with periodicity = 10**9 which is known to have defects (the lower random bits are correlated) and therefore should NOT be used in any statistical study. One should use instead TRandom1, TRandom2 or TRandom3. TRandom3, is based on the "Mersenne Twister generator", and is the recommended one, since it has good random proprieties (period of about 10**6000) and it is fast. TRandom1, based on the RANLUX algorithm, has mathematically proven random proprieties and a period of about 10**171. It is however slower than the others. TRandom2, is based on the Tausworthe generator of L'Ecuyer, and it has the advantage of being fast and using only 3 words (of 32 bits) for the state. The period is 10**26.

The following table shows some timings (in nanoseconds/call) for the random numbers obtained using an Intel Pentium 3.0 GHz running Linux and using the gcc 3.2.3 compiler

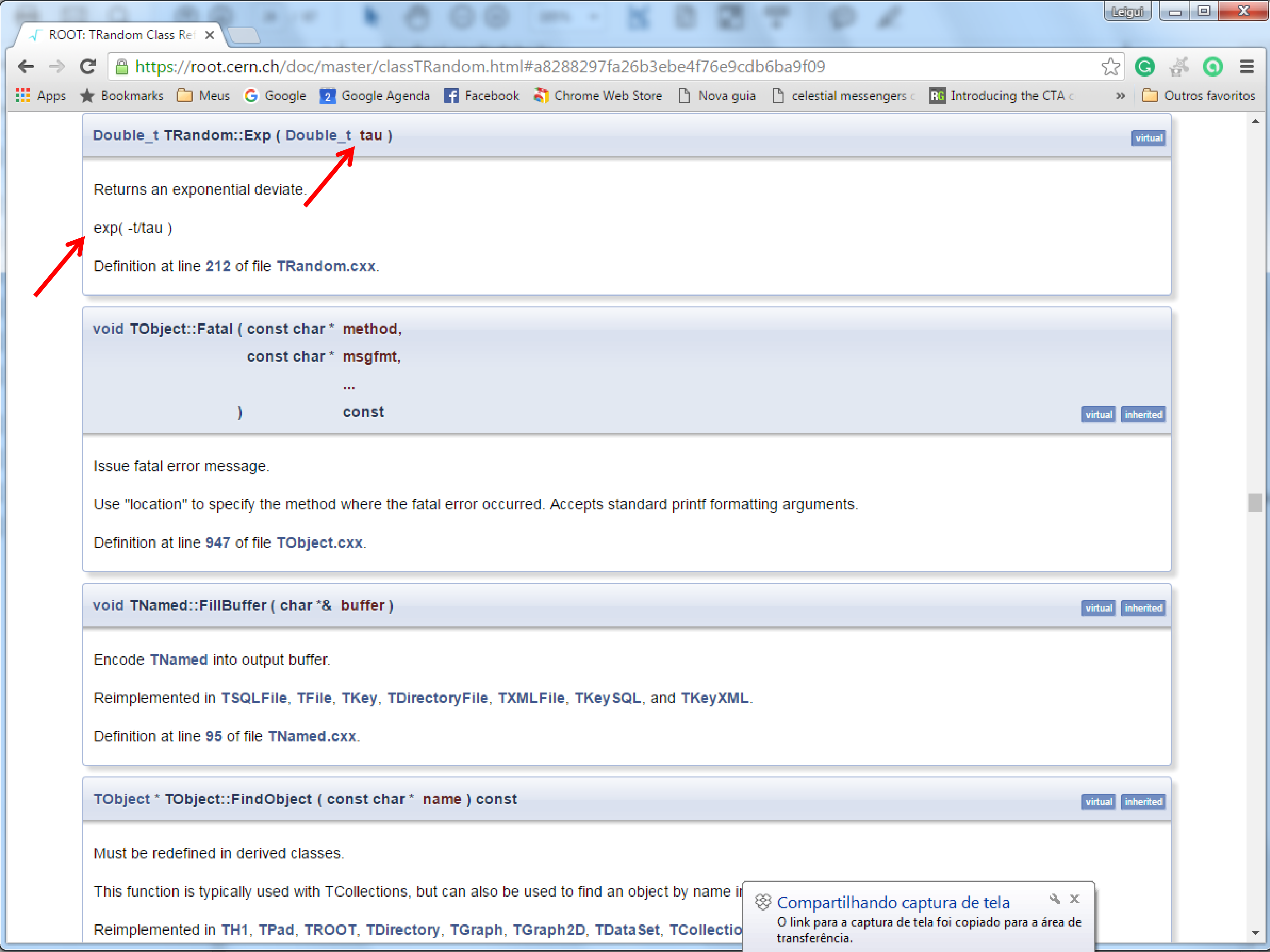
- TRandom 34 ns/call (BAD Generator)
- TRandom1 242 ns/call
- TRandom2 37 ns/call
- TRandom3 45 ns/call

The following methods are provided to generate random numbers disctributed according to some basic distributions:

- `Exp(tau)`
- `Integer(imax)`
- `Gaus(mean,sigma)`
- `Rndm()`
- `Uniform(x1)`
- `Landau(mpv,sigma)`
- `Poisson(mean)`


ROOT: TRandom Class Re
https://root.cern.ch/doc/master/classTRandom.html#a8288297fa26b3ebe4f76e9cdb6ba9f09
Leigui
Apps
Bookmarks
Meus
Google
Google Agenda
Facebook
Chrome Web Store
Nova guia
celestial messengers c
Introducing the CTA c
»
Outros favoritos
https://root.cern.ch/doc/master/classTRandom.html#a8288297fa26b3ebe4f76e9cdb6ba9f09

	Dump contents of object on stdout. More...
virtual void	**Error** (const char *method, const char *msgfmt,...) const Issue error message. More...
virtual void	**Execute** (const char *method, const char *params, **Int_t** *error=0) Execute method on this object with the given parameter string, e.g. More...
virtual void	**Execute** (**TMethod** *method, **TObjArray** *params, **Int_t** *error=0) Execute method on this object with parameters stored in the **TObjArray**. More...
virtual void	**ExecuteEvent** (**Int_t** event, **Int_t px**, **Int_t py**) Execute action corresponding to an event at (px,py). More...
virtual **Double_t**	**Exp** (**Double_t tau**) Returns an exponential deviate. More...
virtual void	**Fatal** (const char *method, const char *msgfmt,...) const Issue fatal error message. More...
virtual void	**FillBuffer** (char *&buffer) Encode **TNamed** into output buffer. More...
virtual **TObject** *	**FindObject** (const char ***name**) const Must be redefined in derived classes. More...
virtual **TObject** *	**FindObject** (const **TObject** *obj) const Must be redefined in derived classes. More...
virtual **Double_t**	**Gaus** (**Double_t** mean=0, **Double_t sigma**=1) Samples a random number from the standard Normal (Gaussian) Distribution with the given mean and sigma. More...
virtual **Option_t** *	**GetDrawOption** () const Get option used by the graphics system to draw this object. More...
virtual const char *	**GetIconName** () const Returns mime type name of object. More...
virtual const char *	**GetName** () const Returns name of object. More...
virtual char *	**GetObjectInfo** (**Int_t px**, **Int_t py**) const Returns string containing info about the object at position (px,py). More...
virtual **Option_t** *	**GetOption** () const



Double_t TRandom::Exp (Double_t tau) virtual

Returns an exponential deviate.

exp(-t/tau)

Definition at line 212 of file TRandom.cxx.

void TObject::Fatal (const char * method,
const char * msgfmt,
...
) const virtual inherited

Issue fatal error message.

Use "location" to specify the method where the fatal error occurred. Accepts standard printf formatting arguments.

Definition at line 947 of file TObject.cxx.

void TNamed::FillBuffer (char *& buffer) virtual inherited

Encode TNamed into output buffer.

Reimplemented in TSQLFile, TFile, TKey, TDirectoryFile, TXMLFile, TKeySQL, and TKeyXML.

Definition at line 95 of file TNamed.cxx.

TObject * TObject::FindObject (const char * name) const virtual inherited

Must be redefined in derived classes.

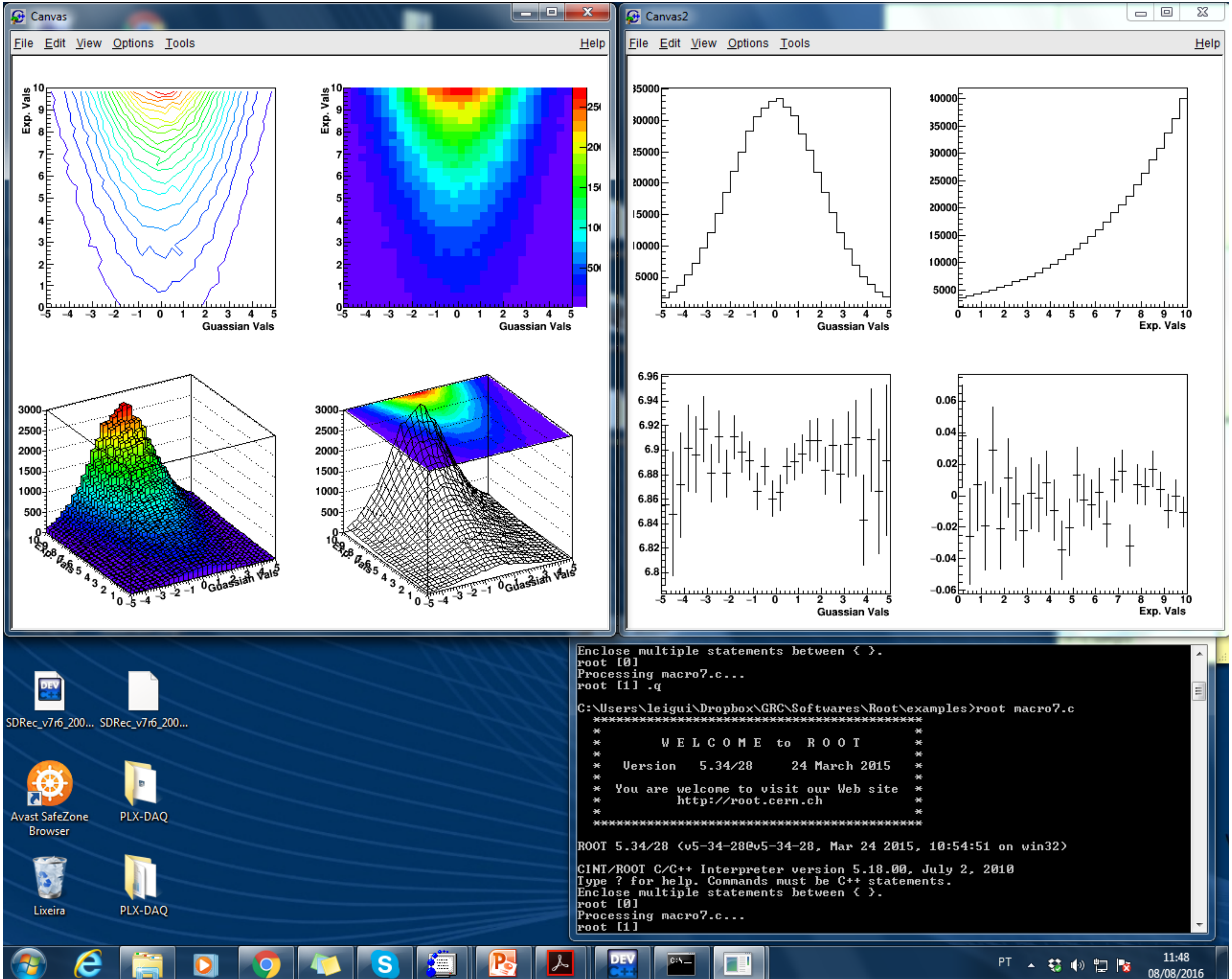
This function is typically used with TCollections, but can also be used to find an object by name i

Reimplemented in TH1, TPad, TROOT, TDirectory, TGraph, TGraph2D, TDataSet, TCollectio

Compartilhando captura de tela

O link para a captura de tela foi copiado para a área de transferência.

macro7.c



```
Enclose multiple statements between { }.
root [0]
Processing macro7.c...
root [1] .q

C:\Users\leigui\Dropbox\GRC\Softwares\Root\examples>root macro7.c
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.34/28      24 March 2015  *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

ROOT 5.34/28 (v5-34-28@v5-34-28, Mar 24 2015, 10:54:51 on win32)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
Processing macro7.c...
root [1]
```

CHAPTER 6

FILE I/O

6.1. Storing ROOT Objects

ROOT offers the possibility to write the instances of all the classes inheriting from the class `TObject` (basically all classes in ROOT) on disk, into what is referred to as *ROOT-file*, a file created by the `TFile` class. One says that the object is made “persistent” by storing it on disk. When reading the file back, the object can be restored to memory.

We can explore this functionality with histograms and two simple macros.

```cpp
void write_to_file(){

    // Istance 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();
}
```

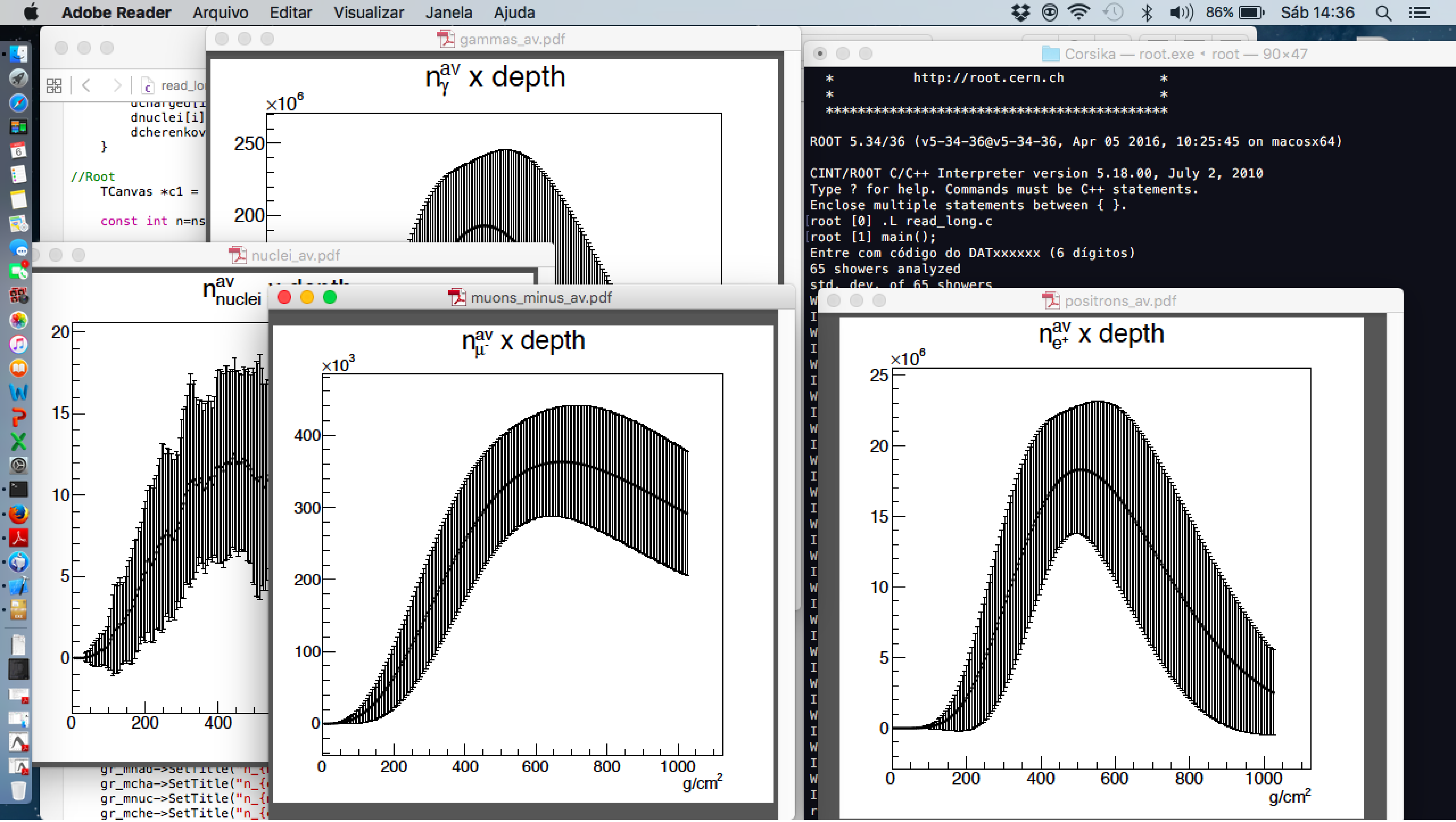
```
>>>  root my_rootfile.root
root [0]
Attaching file my_rootfile.root as _file0...
root [1] _file0.ls()
TFile**         my_rootfile.root
 TFile*         my_rootfile.root
  KEY: TH1F     my_histogram;1  My Title
root [2] my_histogram.Draw()
```

Alternatively, you can use a simple macro to carry out the job:

```
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->GetObjectChecked("my_histogram","TH1F");

    // Draw it
    h->DrawClone();

}
```

file: read_from_file.cxx

Macro read_long.c



```
            dnuclei[i]
            dcherenkov
        }

    //Root
        TCanvas *c1 =

        const int n=ns
```

```
        gr_mcha->SetTitle("n_{
        gr_mnuc->SetTitle("n_{
        gr_mche->SetTitle("n_{
```

```
  *          http://root.cern.ch                *
  *                                             *
  ***********************************************

ROOT 5.34/36 (v5-34-36@v5-34-36, Apr 05 2016, 10:25:45 on macosx64)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] .L read_long.c
root [1] main();
Entre com código do DATxxxxxx (6 dígitos)
65 showers analyzed
std. dev. of 65 showers
```