Introduction to ROOT
What to teach ?

Lorenzo Moneta

ROOT
Data Analysis Framework

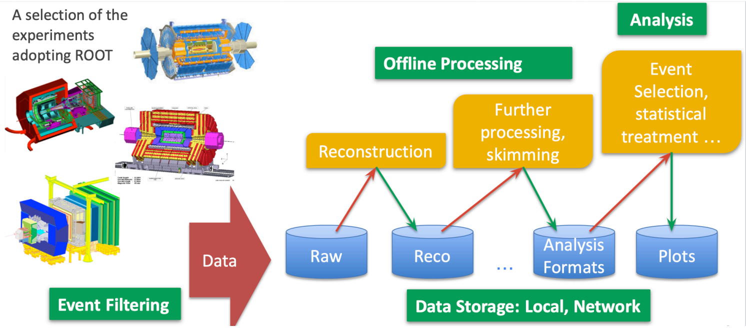
https://root.cern

Outline

- What should we teach as introduction to ROOT ?
  - Use as example the introduction course for CERN Summer students (link available here)
  - Additional material available for more in-depth courses. See in the ROOT Web site

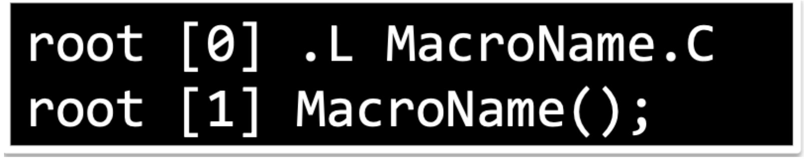
ROOT Introduction

- A quick tour of ROOT
  - What is ROOT ?
    - ROOT building blocks
  - What can you do with ROOT ?
  - ROOT Application domain
- Documentation resources
  - Web site : manuals, reference documentation, tutorial examples
- How to build ROOT
  - how to get a binary distribution


A selection of the experiments adopting ROOT
Event Filtering
Data
Analysis
Offline Processing
Reconstruction
Further processing, skimming
Event Selection, statistical treatment …
Raw
Reco
…
Analysis Formats
Plots
Data Storage: Local, Network

ROOT Prompt

- Introduction to ROOT interpreter
  - ROOT as a calculator
  - ROOT interpreter commands (.x , .!,…)
  - Work with interactive C++
    - e.g. include files, instantiate objects, etc…
- Working with ROOT macro
  - Names vs un-named macro
  - Load/Execute macro
  - Using ACLIC
  - Using root-config
- Jupyter notebooks and SWAN



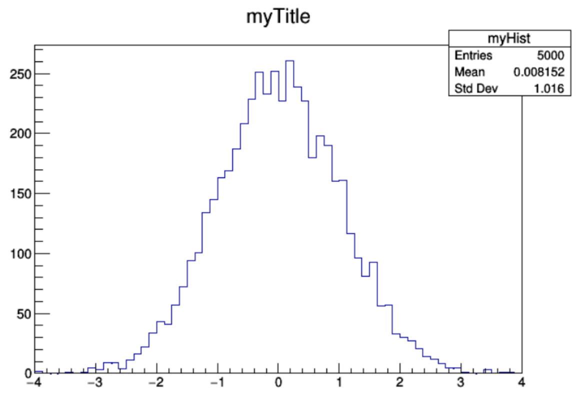
```
root [0] .L MacroName.C
root [1] MacroName();
```

ROOT Prompt: some comments

- Named vs Un-named macro:
    - some notion of ROOT object management is required
    - Example:
        - Histogram should be created with new in named macros
        - Objects in un-named macro don’t go out of scope
- Using ACLIC not useful anymore with Cling
    - only for using different compiler than Clang
    - root-config often better choice
- Start already using Python interface ?

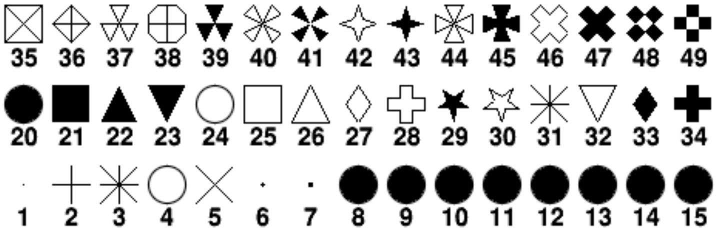
Histograms Graph and Functions

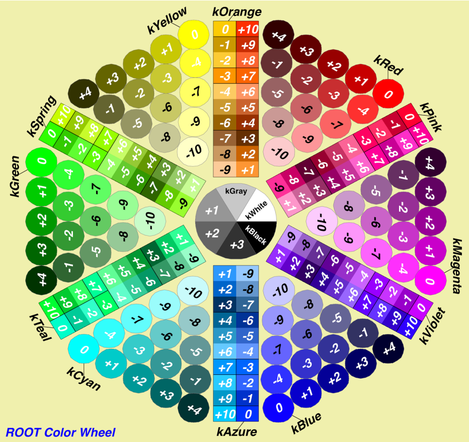
- What are histograms ?
- How to create histograms in ROOT
- Some on histogram management
  - use DrawCopy() / DrawClone()
- Functions: TF1 objects
  - function plotter
  - how to make function with parameters (for fitting)
- TGraph classes
  - Display data points with/without errors

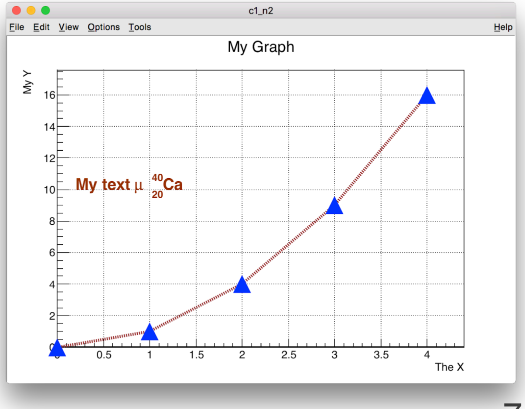


Introduction to Graphics

- Some basic tools for doing ROOT plots
  - how to make nice plots in ROOT
  - useful to point at examples and documentation of THistPainter class





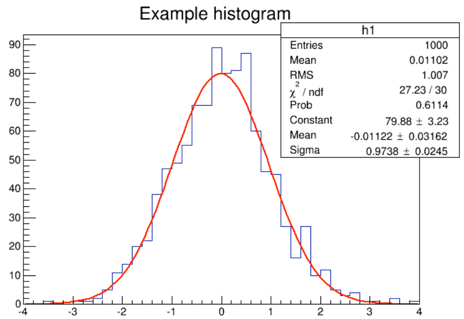


Notes on ROOT Histograms

- Use always TH1D as default type
  - Loss of precisions by using TH1F
  - Useful only in case of memory issues (multi-dim histograms)
- Useful to see usage from Python
  - Fill histogram from a numpy array using FillN
  - Create TGraph objects from numpy arrays
  - Create TF1 from a Python function
- TH1 vs TGraph
- Mention multi-dimensional histograms and profile
  - Projection operations

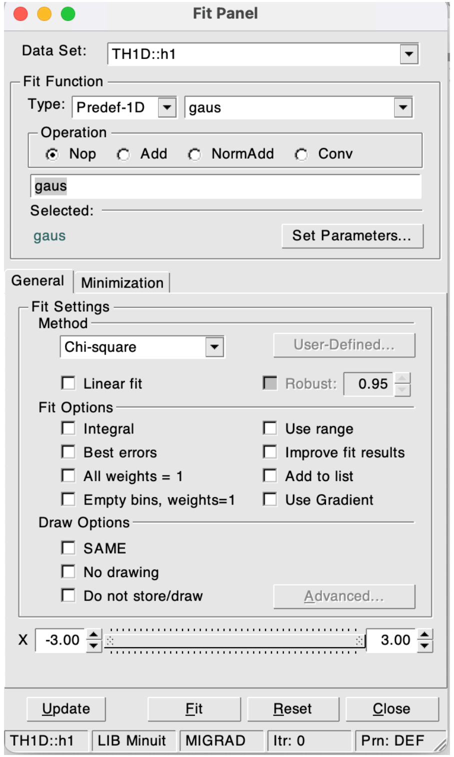
Introduction to Fitting in ROOT

- Basic fitting with ROOT
  - what is fitting ?
  - How to perform a fit to a histogram or graph
    - create a parametric function
    - fitting the ROOT object (fit options)
    - Examine the result



Notes on ROOT Fitting

- Good to show FitPanel
- important to provide links to documentations
  - Different ways to create TF1
  - fit options (in TH1::Fit and TGraph::Fit)
  - Minimization (how to make a fit converging)
  - FitResult class
- Link to statistics documentation:
  - Least square fit vs Likelihood fits
  - Parameter uncertainties and Minos erros
  - Goodness of fit
- Add links to more advanced fitting:
  - Function composition and convolutions
  - RooFit



Fit Panel

Data Set: TH1D::h1

Fit Function

Type: Predef-1D gaus

Operation: ● Nop ○ Add ○ NormAdd ○ Conv

gaus

Selected:

gaus

General | Minimization

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 -3.00 3.00

TH1D::h1 | LIB Minuit | MIGRAD | Itr: 0 | Prn: DEF

PyROOT

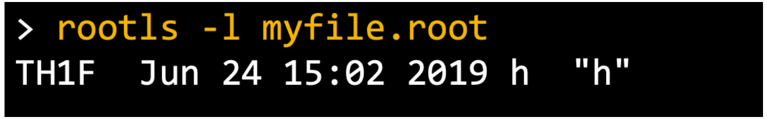
- Provide some examples on how to use ROOT classes in Python

```
import ROOT
h = ROOT.TH1D(‘h1’,’My Histogram’,100,-5,5)
h.FillRandom(‘gaus’)
h.Draw()
```

- Important to show some basic Pythonizations:
  - e.g. using numpy arrays for signatures using `double*` or `float*`
- Management of ROOT object in Python
- Provide links to Python examples and available Pytonizations

ROOT I/O and RDataFrame

- Read/Write Data
  - TFile documentation
  - How to write and then read histograms from a file
  - TBrowser
- Introduction to TTree
  - columnar data storage
  - not showing TTree API (e.g. TTree::Draw)
- Introduction to RDataFrame
  - Show basic functionality
  - Filter/Define and Actions (fill histograms)
  - Saving data (Snapshot)
  - Using callable functions



```
> rootls -l myfile.root
TH1F  Jun 24 15:02 2019 h  "h"
```

```
auto d2 = d.Filter("x > 0")
             .Define("z", "x*x
// make multiple histograms out of it
auto hz = d2.Histo1D("z");
auto hx = d2.Histo1D("x");
```

Notes on I/O

- Showing TTree API is too complex for an introduction course
  - Maybe only TTree::Draw could be mention
- Better to show RDataFrame
- Good to show available Pythonizations
  - easy conversion to Numpy and Panda

- Can be useful to show shortcuts available in the Cling interpreter for objects managed by the current directory

```
TFile::Open("hsimple.root");
hpx->Draw()
```

Summary

- Important to provide links to update documentation and tutorial examples
  - improved significantly last year the ROOT documentation
- Crucial having time to go through some code examples (e.g. using Jupyter notebooks) on the shown topics
- Often important to have both C++ and Python examples
  - give choice to students to work in one of the two