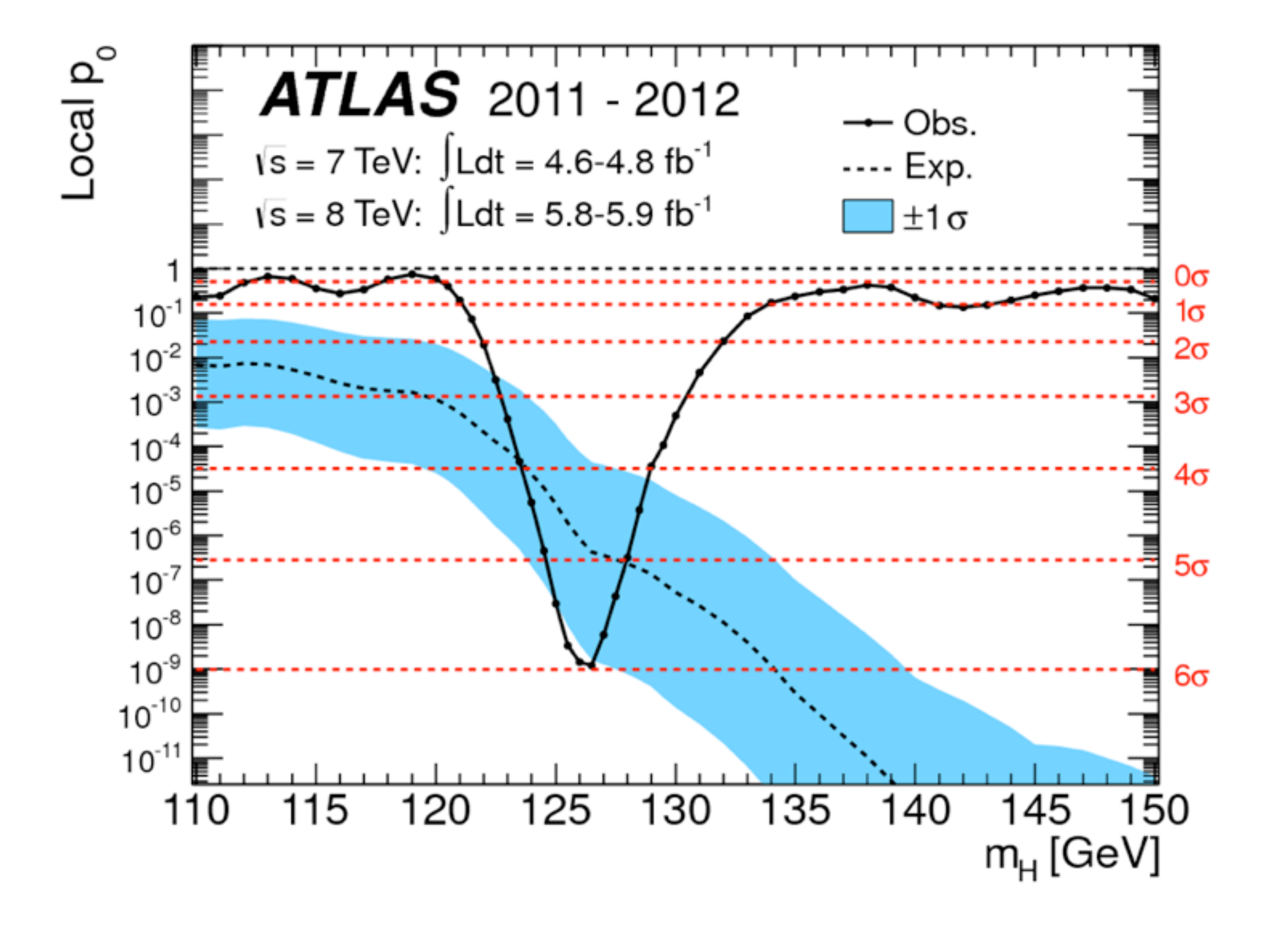
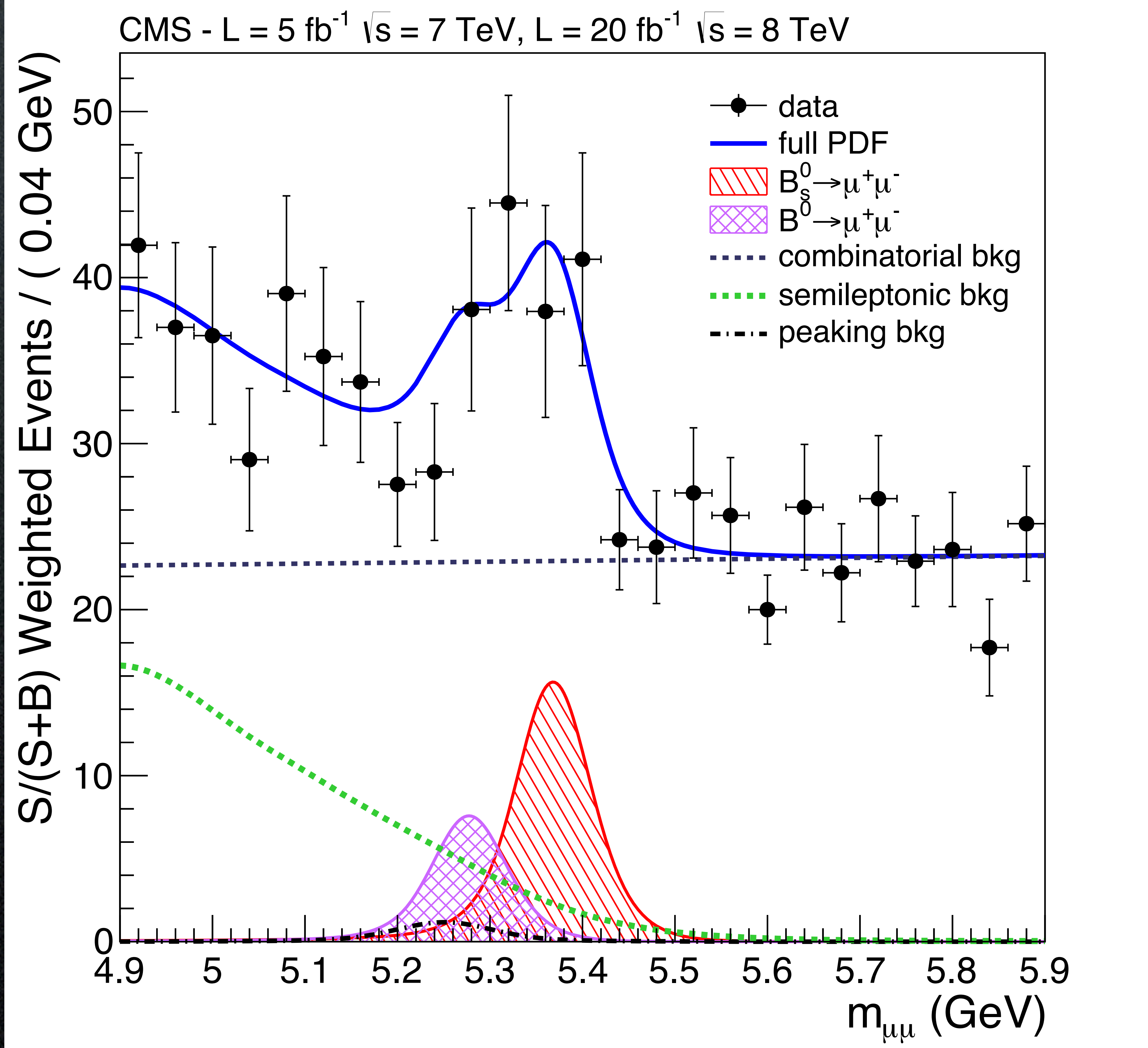
What is ROOT?
Why do we use it?

Answer:
ROOT does what physicists do:
It makes plots.

Heather M. Gray, CERN
Material from: Bill Seligman, Lorenzo Moneta



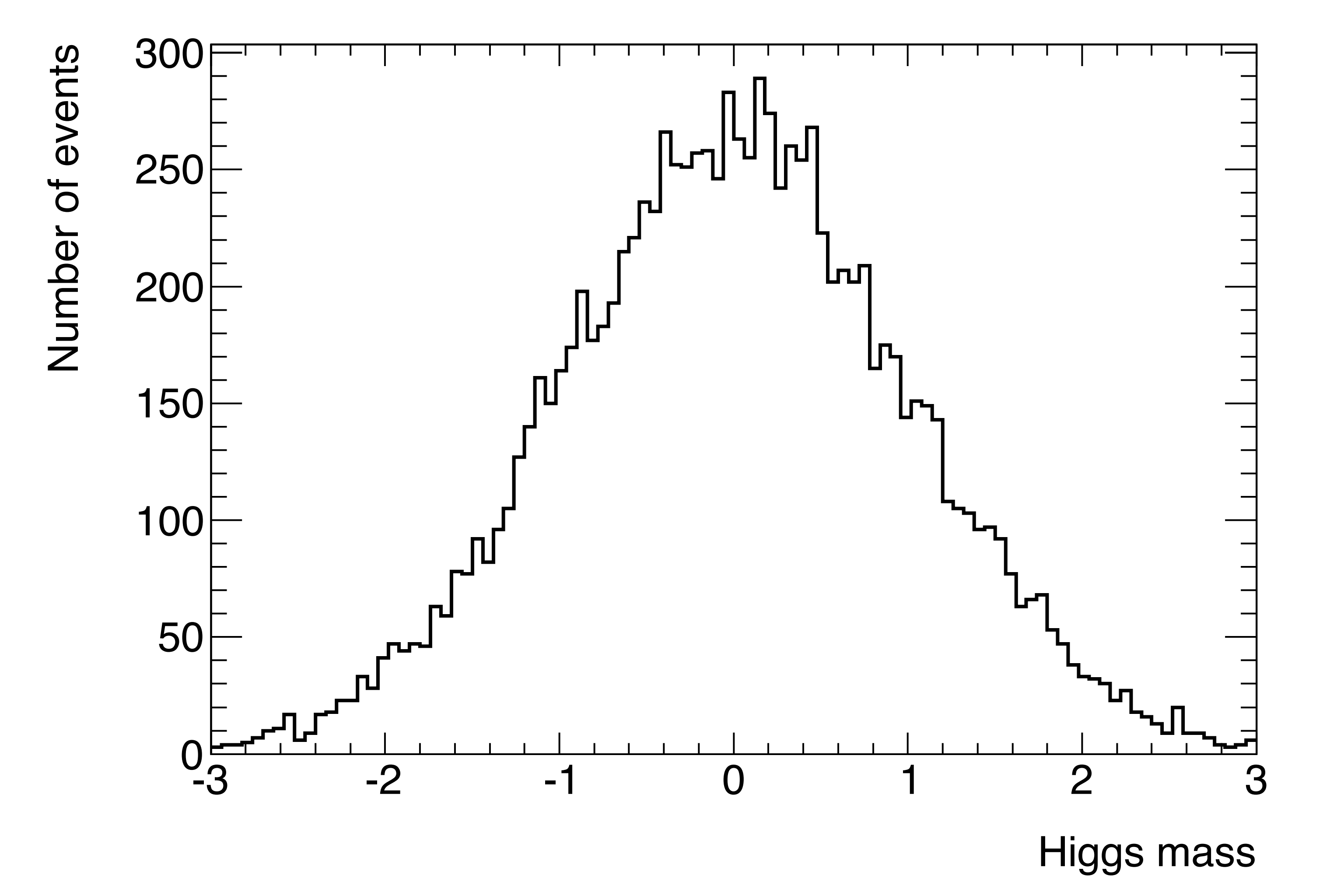
ATLAS 2011 - 2012
√s = 7 TeV: ∫Ldt = 4.6-4.8 fb-1
√s = 8 TeV: ∫Ldt = 5.8-5.9 fb-1
Obs.
Exp.
±1σ
Local p0
1
10-1
10-2
10-3
10-4
10-5
10-6
10-7
10-8
10-9
10-10
10-11
0σ
1σ
2σ
3σ
4σ
5σ
6σ
110
115
120
125
130
135
140
145
150
mH [GeV]



CMS - L = 5 fb-1 √s = 7 TeV, L = 20 fb-1 √s = 8 TeV
S/(S+B) Weighted Events / (0.04 GeV)
data
full PDF
B0s→μ+μ-
B0→μ+μ-
combinatorial bkg
semileptonic bkg
peaking bkg
0
10
20
30
40
50
4.9
5
5.1
5.2
5.3
5.4
5.5
5.6
5.7
5.8
5.9
mμμ (GeV)

- High-energy physics students are typically asked to
  - Take variables from an ntuple, perform some computations, make histograms and fit them
- So ... what is a histogram, what is an ntuple and how do we perform computations and fits?

Anatomy of a Histogram



Name or identifier

Title

x-axis title

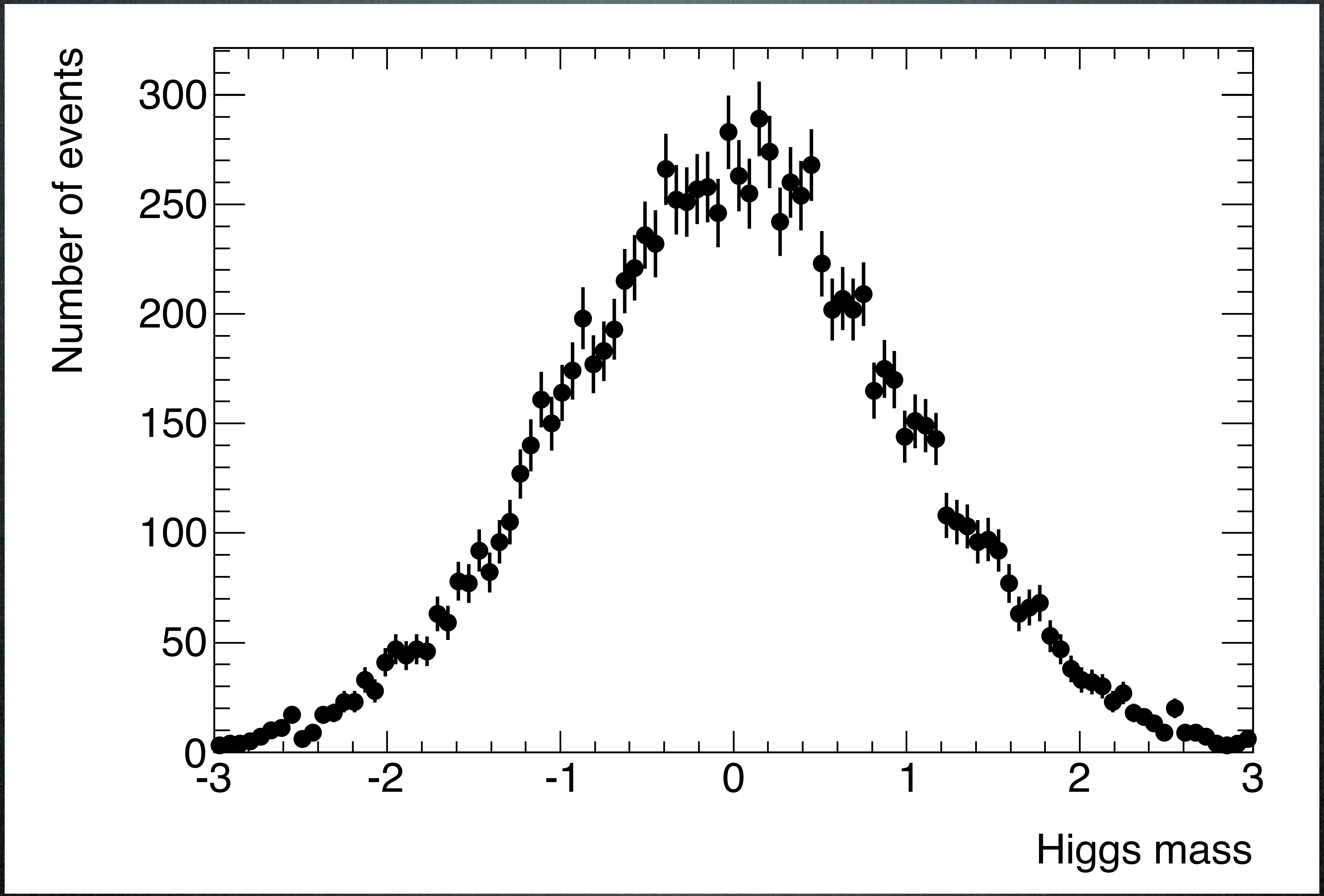
TH1F * h1 = new TH1F("h1", "Sample Histogram;Higgs mass;Number of events", 100, -3, 3)

y-axis title

number of bins

min

max



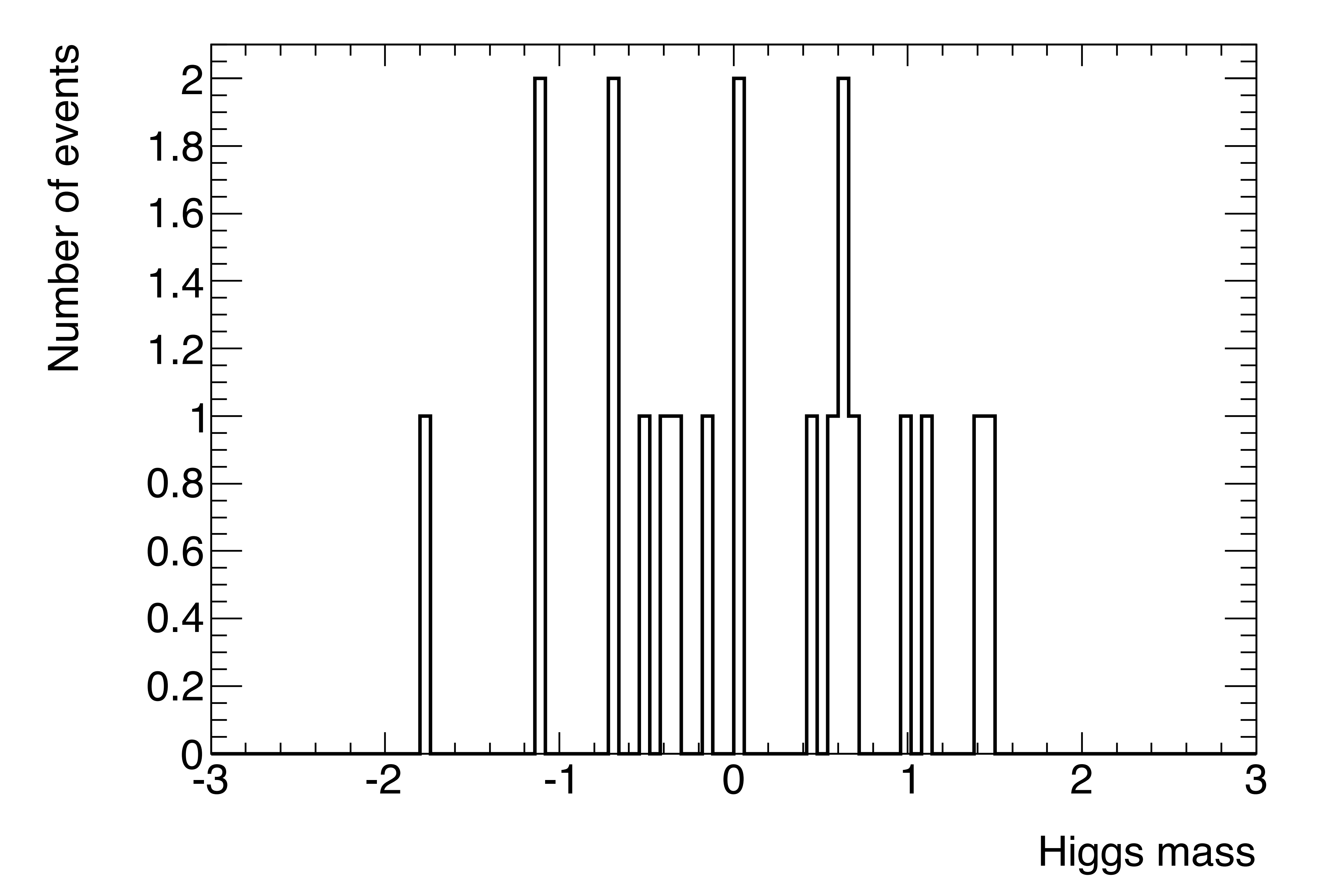
h1->FillRandom("gaus", 10000);

h1->Draw("PE") Don't forget the errors !

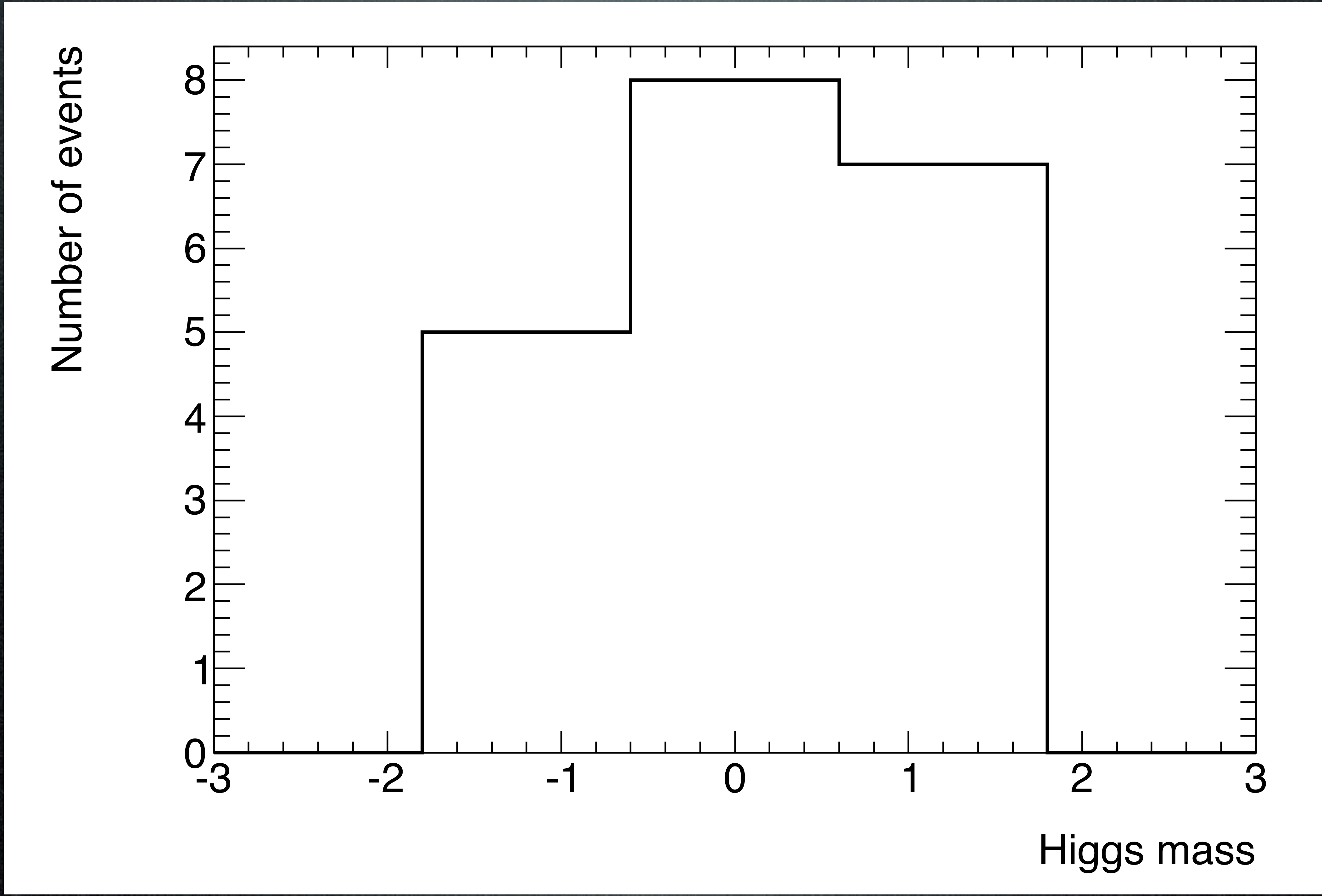
For simple histograms, the error in each bin is the square root of the number of events in that bin

There's an art to histogram design

Too many bins



Too few bins



Anatomy of an ntuple

An ntuple is an ordered list of numbers

A simple form of a ROOT tree

Branches

Entries

Row	Event	m	pt	eta	phi
0	0	90.12	3.96	-3.86	0.38
1	1	89.37	31.36	-1.08	-1.09
2	2	138.95	0.67	6.19	-3.02
3	3	87.69	3.66	2.89	-2.10
4	4	89.48	35.94	-2.59	0.52
5	5	91.92	2.65	-5.91	1.52

Bonus: Can you guess what particle is in this tree?

ROOT Tree

- Ntuples are ordered lists of numbers
- Trees are ordered lists of any collection of objects
- They can be used in very similar ways

Why ROOT ?

- It knows about ntuples and histograms
  - and 4-vectors and object persistency and detector geometry and Feynman diagrams
- It can handle large volumes of data
  - millions of physics events
  - files of gigabytes -> terabytes in size
- Multi-platform: Windows, MAC, many UNIX flavours
- It's free

Why ROOT ? But ...

- It’s open-source with a complicated design history
- User-interface issues and documentation are often neglected
- ROOT is not easy to use
- You have to know some C++ in order to used ROOT effectively
  - to perform computations

Hands On Tutorials

- Histograms
- Plots
- Trees
- Fitting

If we're moving too slowly for you, try working through the tutorials available here: https://twiki.cern.ch/twiki/bin/view/Main/RootIRMMTutorial2013

Step 1: Setting Up and Basic Commands

Logging in to the server

- We're going to run ROOT on the server
- So step 1, is to log into the server and set up your environment
- Username = the name of your computer
- Password = the one on the blackboard
- So ... type in a terminal (e.g.)
  - ssh -X -Y capetown@10.0.0.252

Usernames

- tripoli, malako, mogadiscio,luanda, bissau, conakry, freetown, praia, antananarivo, saotome, kinshasa, accra, djibouti, brazzaville, lome, rabat, kampala, libreville, bujumbura, lecaire, lilongwe, maseura, khartoum, portlouis, ndjamena, addisababa, asmara, bangui, bamako
- If your computer name is missing just let me know

http://root.cern.ch

Lots of useful documentation and tutorials

Starting ROOT

- To run ROOT, type
  - **root**
- in a terminal window
- First you'll see the white-and-blue ROOT window appear on your screen. It will disappear and a brief "Welcome to ROOT" display will be written on your command window


ROOT
Version 5
Conception: Rene Brun, Fons Rademakers
Lead Developers: Rene Brun, Philippe Canal, Fons Rademakers
Core Engineering: Bertrand Bellenot, Olivier Couet, Gerardo Ganis, Andrei Gheata, David Gonzalez, Jan Iwaszkiewicz, Lorenzo Moneta, Axel Naumann, Paul Russo, Matevz Tadel
Version 5.26/00

```
Heathers-MacBook-Pro% root
 *******************************************
 *                                         *
 *        W E L C O M E  to  R O O T       *
 *                                         *
 *   Version   5.34/05  14 February 2013   *
 *                                         *
 *  You are welcome to visit our Web site  *
 *          http://root.cern.ch            *
 *                                         *
 *******************************************

ROOT 5.34/05 (branches/v5-34-00-patches@48624, Feb 19 2013,
09:50:22 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 { }.

Applying ATLAS style settings...

root [0]
```

Basic Commands

- Type **.h** to see a list of ROOT commands
  - Probably more information than you can use right now. Try it and see.
- Most important command is the one to quit ROOT. To exit type **.q**. Do this now and then start ROOT again
- Sometimes ROOT will crash. Sometimes it will crash so badly that **.q** won't work. Try typing .qqq if .q doesn't work or **.qqqqq** or **.qqqqqqq**

ROOT as a calculator

- Try out these commands to use ROOT as a calculator
  - 1+1
  - 2 * (4 +2)/12.
  - sqrt(3)
  - 1 > 2
  - TMath::Pi()
  - TMath::Erf(2)

note the .

Pop Question:
What is the square root of
76440049 ?

More information:
http://root.cern.ch/root/html/TMath.html

Exercise 1

- Calculate the mass of a Z boson
- Provided data: kinematics of the two muons that it decays to

Row	Event	pt	eta	phi	m	charge
0	0	35.67	-0.16	-0.36	0.106	-1
1	0	32.89	-1.71	2.70	0.106	+1


Remember Lorentz invariance from special relativity ?

TLorentzVector

- The TMath class provides many useful functions that you can use when analysing data
  - http://root.cern.ch/root/html/TLorentzVector.html
- Step 1: Define a TLorentzVector for each muon
  - TLorentzVector v1, v2;
  - Useful function: SetPtEtaPhiM(pt, eta, phi, m)

Exercise 1 (cont)

- The TLorentzVector class provides operators to add, subtract or compare four-vectors:

- ```
  v3 = -v1;
  v1 = v2+v3;
  v1+= v3;
  v1 = v2 + v3;
  v1-= v3;
  ```

- Use this information to add together the two muons to get a TLorentzsVector for the Z-boson
  ```

Exercise 1 (cont)

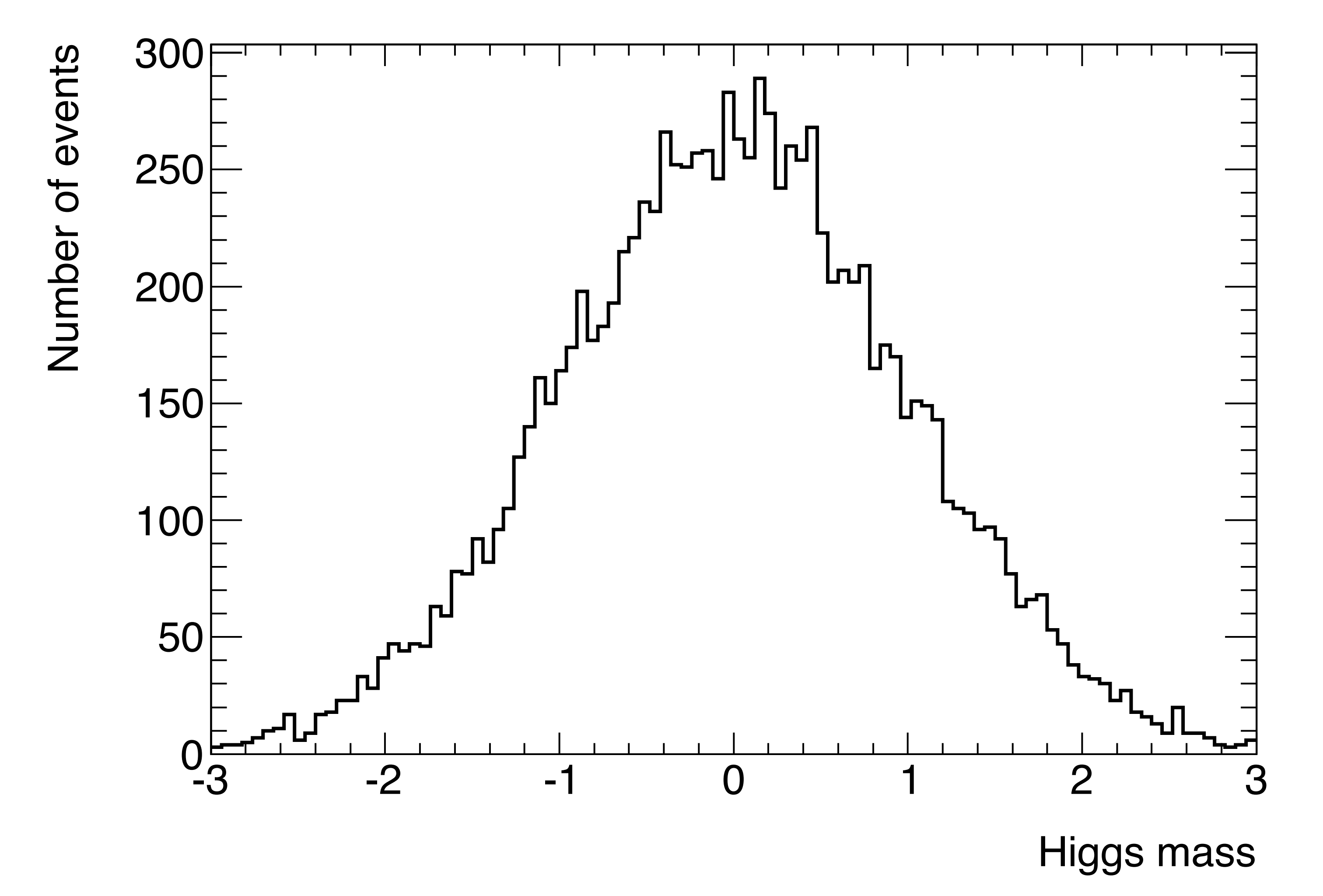
- Use the M() function to display its mass
- Bonus: Compare Pt(), Eta() and Phi() to slide 10

Bonus

- There are many more things that you can do with the TLorentzVector class
  - Boost, rotation, etc
- Have a look at the documentation !
- http://root.cern.ch/root/html/TLorentzVector.html

Step 2: Histograms

Reminder: Histograms



Name or identifier Title x-axis title

TH1F * h1 = new TH1F("h1", "Sample Histogram;Higgs mass;Number of events", 100, -3, 3)

y-axis title number of bins min max

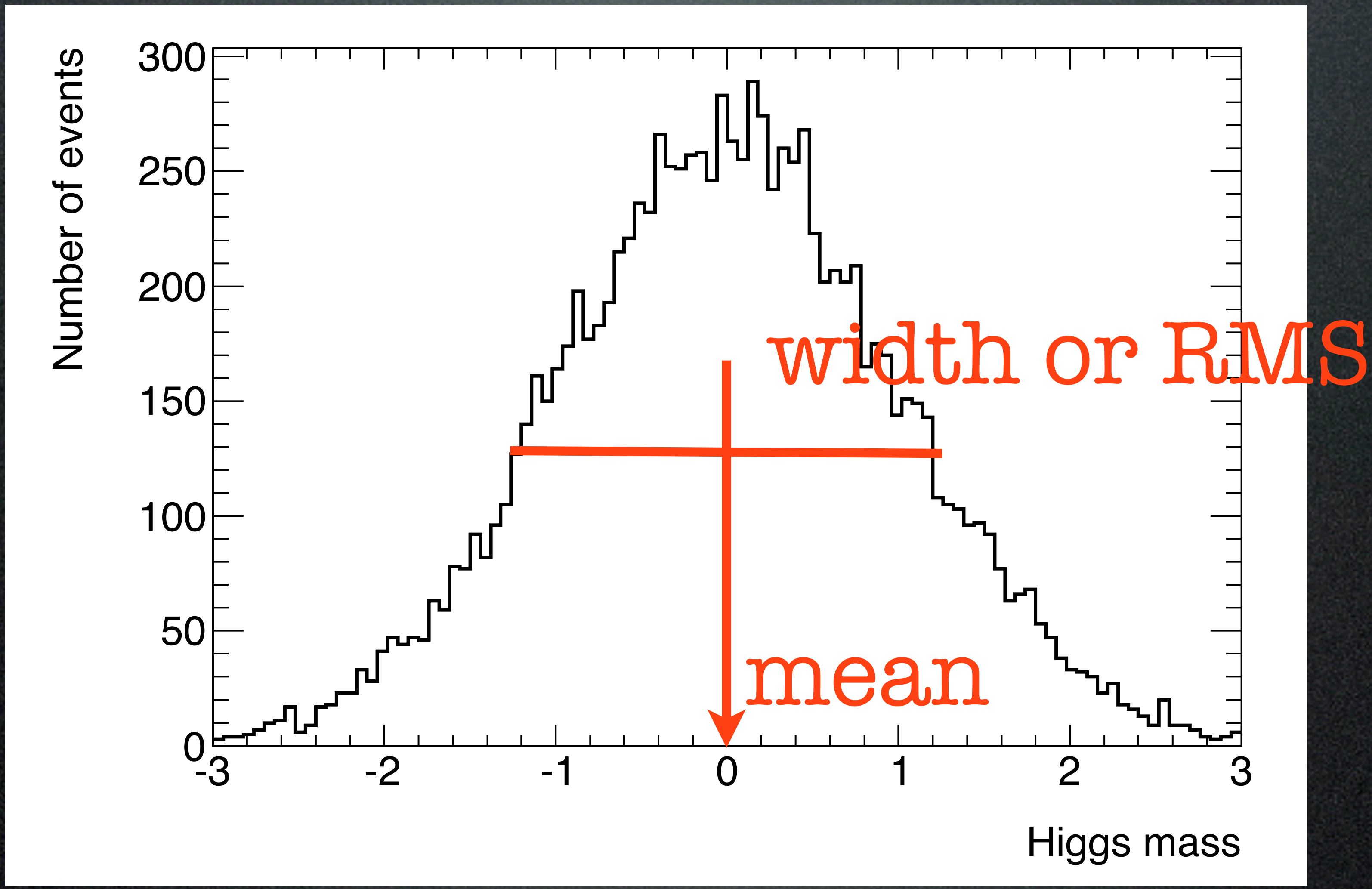
Creating a Histogram

- Creating a histogram
  - **TH1F * h1 = new TH1F("h1", "Sample Histogram;Higgs mass;Number of events", 100,-3,3)**
- Filling a histogram
  - **h1->Fill(2);** Fill the histogram with one observation
  - **h1->FillRandom("gaus", 10000);**

Randomly fill the histogram with a 1000 random numbers

Displaying Histograms

- **h1->Draw()**



Histogram Statistics

- Try out these commands on your histogram
- **`h1->GetEntries()`**
- **`h1->Integral()`**
- **`h1->GetMean()`**
- **`h1->GetMeanError()`**
- **`h1->GetRMS()`**
- **`h1->GetRMSError()`**

Bonus:
How does the error on the mean depend on how many random numbers you use ?

Drawing Options

- Try out the various different drawing options
  - Draw errors bars for each bin
    - **h1->Draw("E")**
  - Draw more than one histogram on the same canvas
    - **h2->Draw("same")**
  - Apply a logarithmic scale to one axis
    - **gPad->SetLogy()**
- More details: http://root.cern.ch/root/html/THistPainter.html

Exercise 2

- Declare and fill a histogram to display the mass of the Z boson

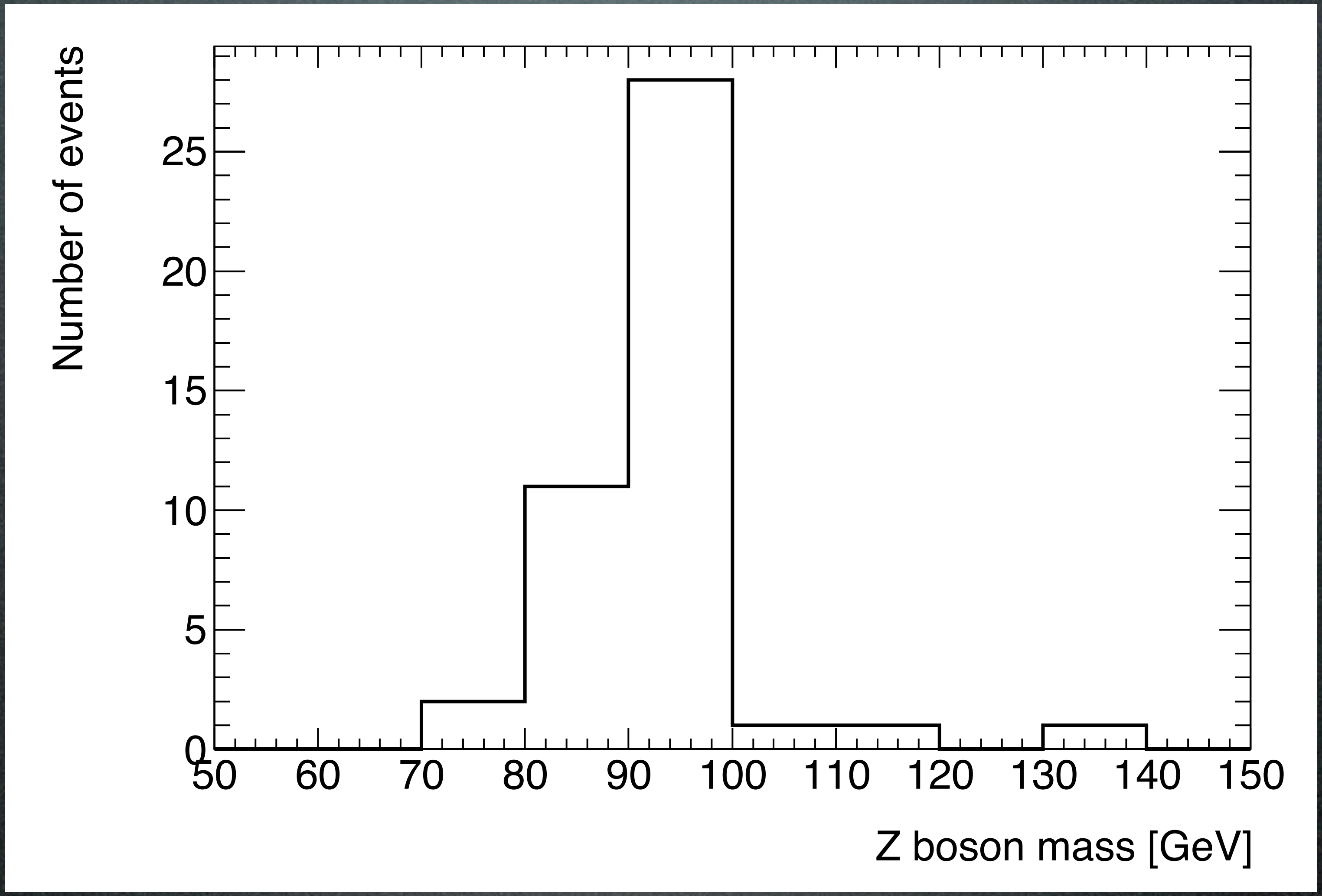
Event	m
0	90.12
1	89.37
2	138.95
3	87.69
4	89.48
5	91.92
6	91.86
7	91.16
8	88.84
9	87.24
10	105.20

Event	m
11	87.99
12	90.84
13	91.46
14	91.46
15	91.19
16	81.89
17	92.10
18	91.36
19	89.53
20	95.00
21	93.94

Event	m
22	91.13
23	91.04
24	98.67
25	89.84
26	90.33
27	97.44
28	74.41
29	89.28
30	93.36
31	110.24
32	91.46

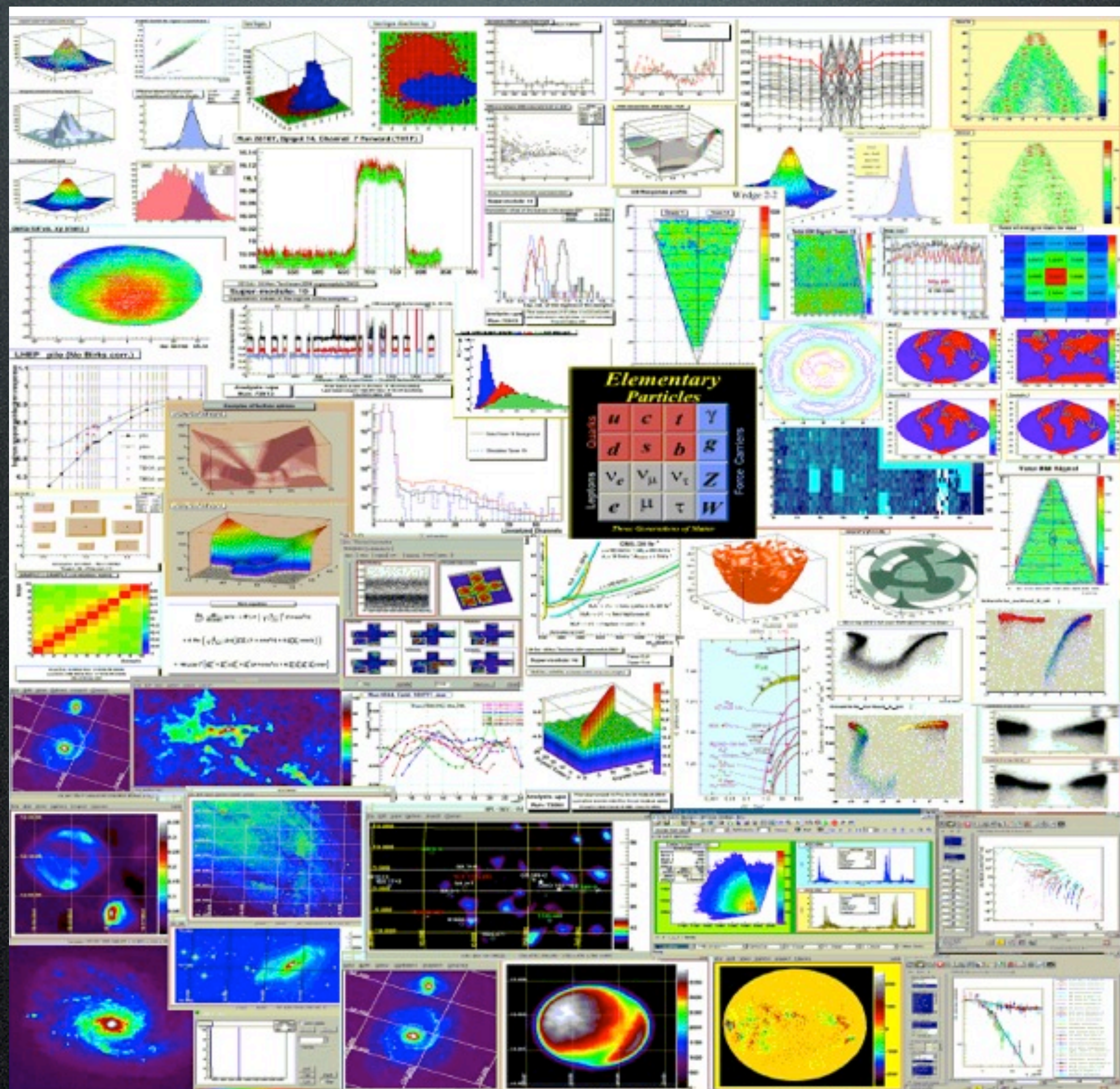
Event	m
11	90.45
12	91.91
13	89.85
14	93.49
15	91.21
16	91.76
17	90.40
18	79.10
19	93.64
20	90.82
21	93.30

Solution

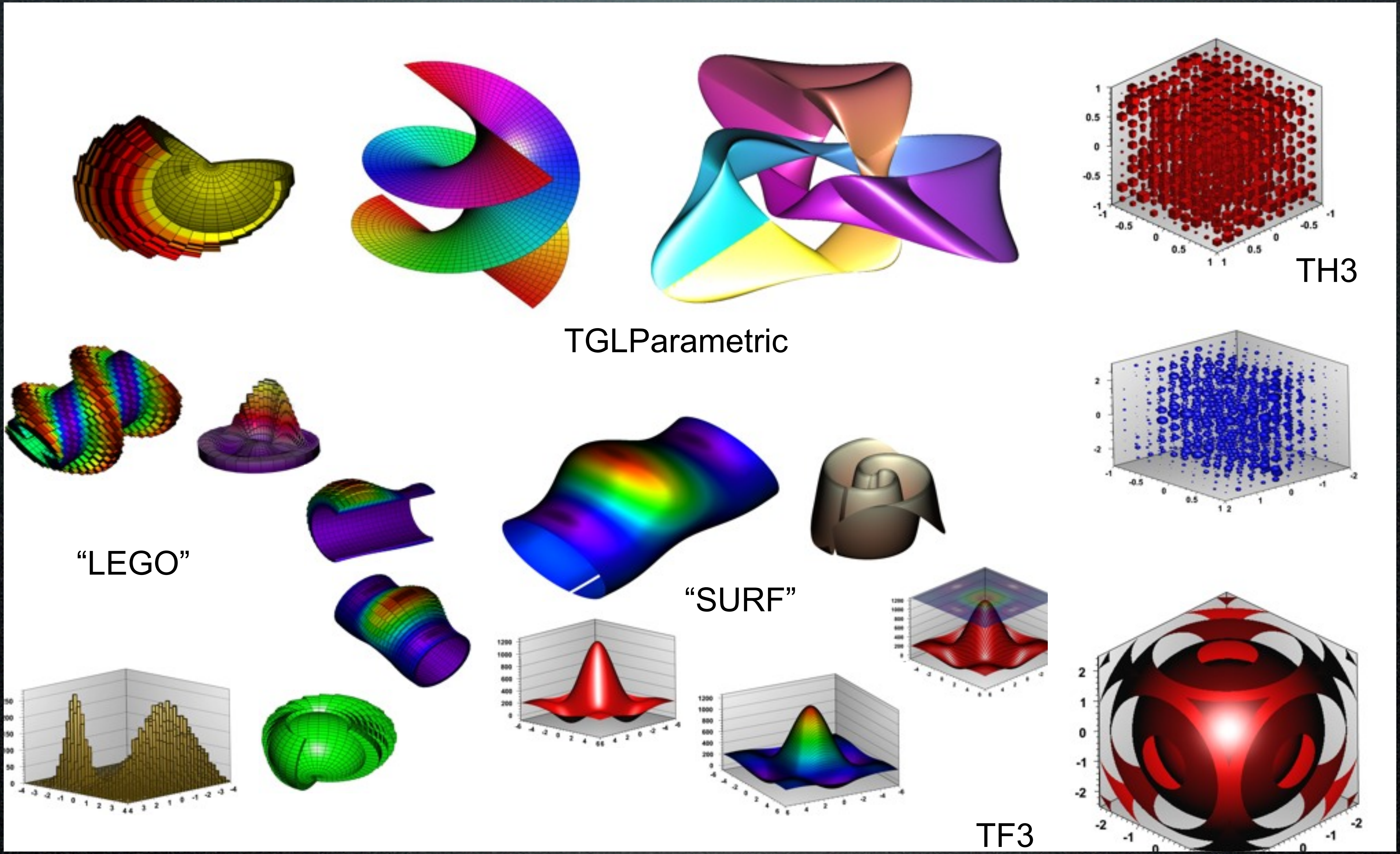


Bonus:
What is the mass ?
How many entries are in the histogram ?
Can you save your histogram as a pdf file ?

2D Graphics



3D Graphics



Some examples: http://root.cern.ch/root/html/tutorials/graphics/index.html

Wasn't the last exercise a lot of work ?
Entering all those numbers in by hand...
It turns out that there's a better way

Step 3: Trees

Reminder: Ntuples

An ntuple is an ordered list of numbers

A simple form of a ROOT tree

Branches

Entries

Row	Event	m	pt	eta	phi
0	0	90.12	3.96	-3.86	0.38
1	1	89.37	31.36	-1.08	-1.09
2	2	138.95	0.67	6.19	-3.02
3	3	87.69	3.66	2.89	-2.10
4	4	89.48	35.94	-2.59	0.52
5	5	91.92	2.65	-5.91	1.52

Trees

- Essentially tables of data stored within a ROOT while
- A very useful way to store data from a physics event
  - e.g. from the LHC
- Can also store almost any dataset that you're interested in

Reading a Tree

- Find the file mytree.root
- Start root
- Open the file:
  - TFile * f = new TFile("mytree.root")
- Start a new browser
  - TBrowser t

Reading a Tree 2

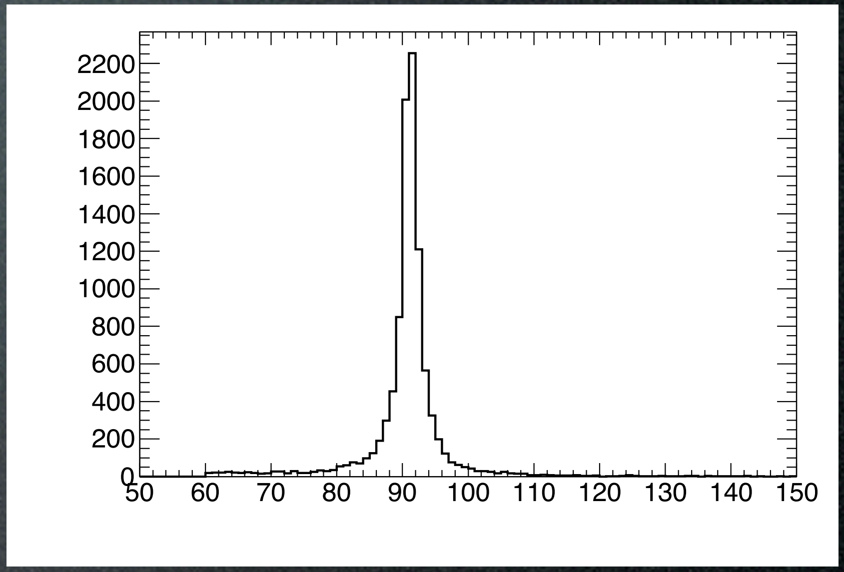
- Right click on the events icon select Scan, then click Ok
    - You'll see lots of numbers including the names of the variables in the tree
- If you hit enter, you'll see more numbers
- Hit q to finish the scan
- Left click on the events icon
    - You'll see a list of variables (branches)
    - Click on them one at a time to display them

Reading a Tree 3

- You can also access the information in the tree on the command line
- **events->Show(0)** //displays the first event
- **events->Scan("Z_mass")** //scan through all the different entries for the Z boson mass
- **events->Draw("Z_mass>>hZmass")** //Take all the entries for the Zboson mass and store them in a histogram

TTree Tricks

- **events->Draw("Z_mass>>hZmass(100, 50, 150)")** // save in a histogram
- **events->Draw("Z_mass>>hZmass(100, 50, 150)", "mu1_pt > 25 && TMath::Abs(mu1_eta) < 2.5")** // apply cuts to select events
- **events->Draw("mu1_pt:mu2_pt","", "colz")** //make a 24 histogram and apply formatting options



Draw() is very powerful !

Bonus: Find the correlation between mu1_pt and mu2_pt

Exercise 4

- Goal: Produce a few different histograms from mytree.root
- 3 histograms for Z boson mass, p_T and eta
  - Both muons have $p_T > 25$ GeV and $|eta| < 2.5$
- 2D histogram: Z boson p_T vs Z boson eta

Aside: Writing Code in ROOT

Running Code

- Macro: A short file (program) interpreted by ROOT
- Create a new file plotZboson.C (touch)
- Edit the file and include the following lines (use nano)

```
void plotZboson() {
  TFile * f = new TFile(“mytree.root”)
  TTree * events = (TTree *) f->Get(“events”);
  events->Draw(“Z_mass”);
}
```

- Save the file
- Execute in ROOT
  - **.x plotZboson.C()**

Which editor ?
Your choice ! A few popular options are nano, emacs and vi

Faster: Compiling Code

- Compile your code within ROOT
  - .x plotZboson.C+()
- The **+** tells ROOT that you want to compile your code
  - takes seconds and uses your system's compiler
- Much, much faster !
- But make sure you include the needed header files

Header Files

- These are pieces of code that tell ROOT how the various objects that you are using (TTree, TH1F, etc) are defined
- **#include “TTree.h”**
- **#include “TFile.h”**
- **#include “TH1F.h”**
- The error messages from ROOT will tell you which headers you missed

Using Trees in code

- Have a look at the provided macros
  - events.h and events.C
- They provide a snippet of code that shows you how to run over files and read TTrees
- Main function is called ‘Loop’
  - It loops over all events in the TTree and reads them in one by one

Using TTrees in code 2

- Run the macro as follows
- .L events.cxx++ // compiles the code
- events t; // defines an ‘events’ object
- t.Loop(); // loops over all the events and produces a print out

Exercise 4: Putting things together

- Step 1: Modify events.C so that it produces a histogram of the Z_mass
  - Run your code and check that you see the histogram (Did you declare the range correctly?)
- Step 2: Remember TLorentzVector ? Now use these to reconstruct the Z boson mass but using the information from the muons

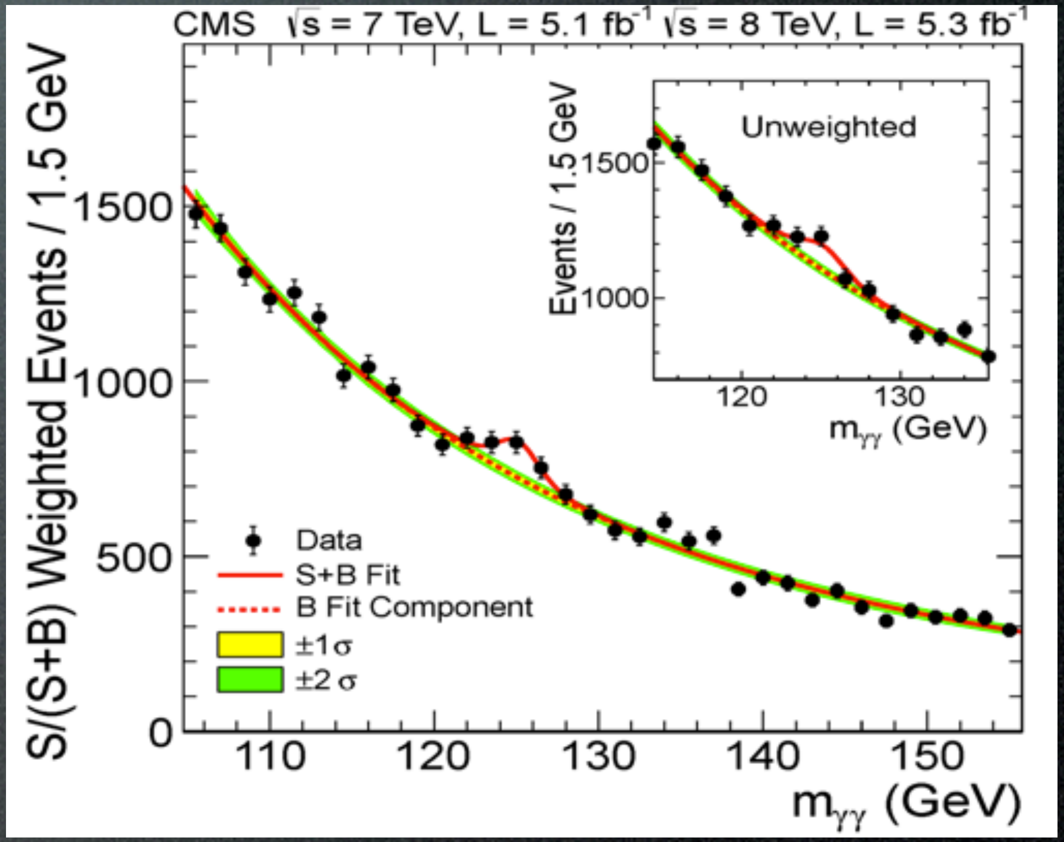
Exercise 4

- Overlay the two histograms for the mass of the Z boson on the same plot
    - Z_mass from the tree directly
    - Z_mass reconstructed from the two muons
- Hint: to draw a second histogram on the same canvas do `h2->Draw("same")`

Step 4: Fitting

What is fitting ?

- Use the observed data to try to extract the underlying distribution



- Example: Higgs-> γγ search from CMS
  - Fit the data for the expected number of events and the Higgs mass

How do we fit?

- A histogram is an estimate of an underlying distribution
- Use the histogram to estimate the parameters of the distribution
- Assume a relation between the x and y variables: $y = f(x \mid \theta)$
- $f(x \mid \theta)$ is the model
- y is the bin content of the histogram

Fitting

- Least square fit (χ^2)
    - Minimise square of deviation between data and prediction

$$\chi^2 = \sum_i \frac{(Y_i - f(X_i, \theta))^2}{\sigma_i^2}$$

Fitting II

- Maximum likelihood fit
  - Use the maximum of the likelihood (or the minimum of the log likelihood)

$$L(x|\theta) = \prod_i P(x_i|\theta)$$

- Obtain likelihood by assuming a Poisson distribution in every bin
  - Poisson= $(n_{obs} \mid n_{exp} = f(x_c|\theta))$
- Equivalent for high statistics, ML correct for low statistics

Fitting in ROOT

- Create a fit function representing the model
  - TF1 * fgaus = new TF1("fgaus", "gaus", -1, 1)
  - TH1F * h1 = new TH1F("h1, "data", 100, -5, 5)
  - h1->FillRandom("gaus", 10000)
  - h1->Fit("fgaus")

```
root [4] h1->Fit("fgaus")
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
 FCN=57.7629 FROM MIGRAD    STATUS=CONVERGED      60 CALLS          61 TOTAL
                     EDM=3.29508e-09    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                STEP         FIRST
  NO.   NAME      VALUE            ERROR       SIZE      DERIVATIVE
   1  Constant     3.97891e+02   4.93046e+00   1.49364e-02   3.65172e-06
   2  Mean         1.13884e-02   1.00492e-02   3.75730e-05   7.94991e-03
   3  Sigma        9.97455e-01   7.30906e-03   7.36429e-06   3.98607e-03
(class TFitResultPtr)140388148986080
```

Exercise 5

- Take the histogram that you have produced for the Z mass
- Fit it with a Gaussian function
- Extract the mass and width of the Z boson from the fit
  - Hint: **GetParameter(0)**
- How does the fitted mass compare to the mean of the histogram ?

That’s all folks!

Backup

Solution

Z $J = 1$

Charge = 0

Mass $m = 91.1876 \pm 0.0021$ GeV [d]

Full width $\Gamma = 2.4952 \pm 0.0023$ GeV

$\Gamma(\ell^+ \ell^-) = 83.984 \pm 0.086$ MeV [b]

$\Gamma(\text{invisible}) = 499.0 \pm 1.5$ MeV [e]

$\Gamma(\text{hadrons}) = 1744.4 \pm 2.0$ MeV

$\Gamma(\mu^+ \mu^-)/\Gamma(e^+ e^-) = 1.0009 \pm 0.0028$

$\Gamma(\tau^+ \tau^-)/\Gamma(e^+ e^-) = 1.0019 \pm 0.0032$ [f]

From:http://pdg.lbl.gov/2013/tables/rpp2013-sum-gauge-higgs-bosons.pdf

Lots of useful information about particles here!