

Root



Stefano Lacaprara

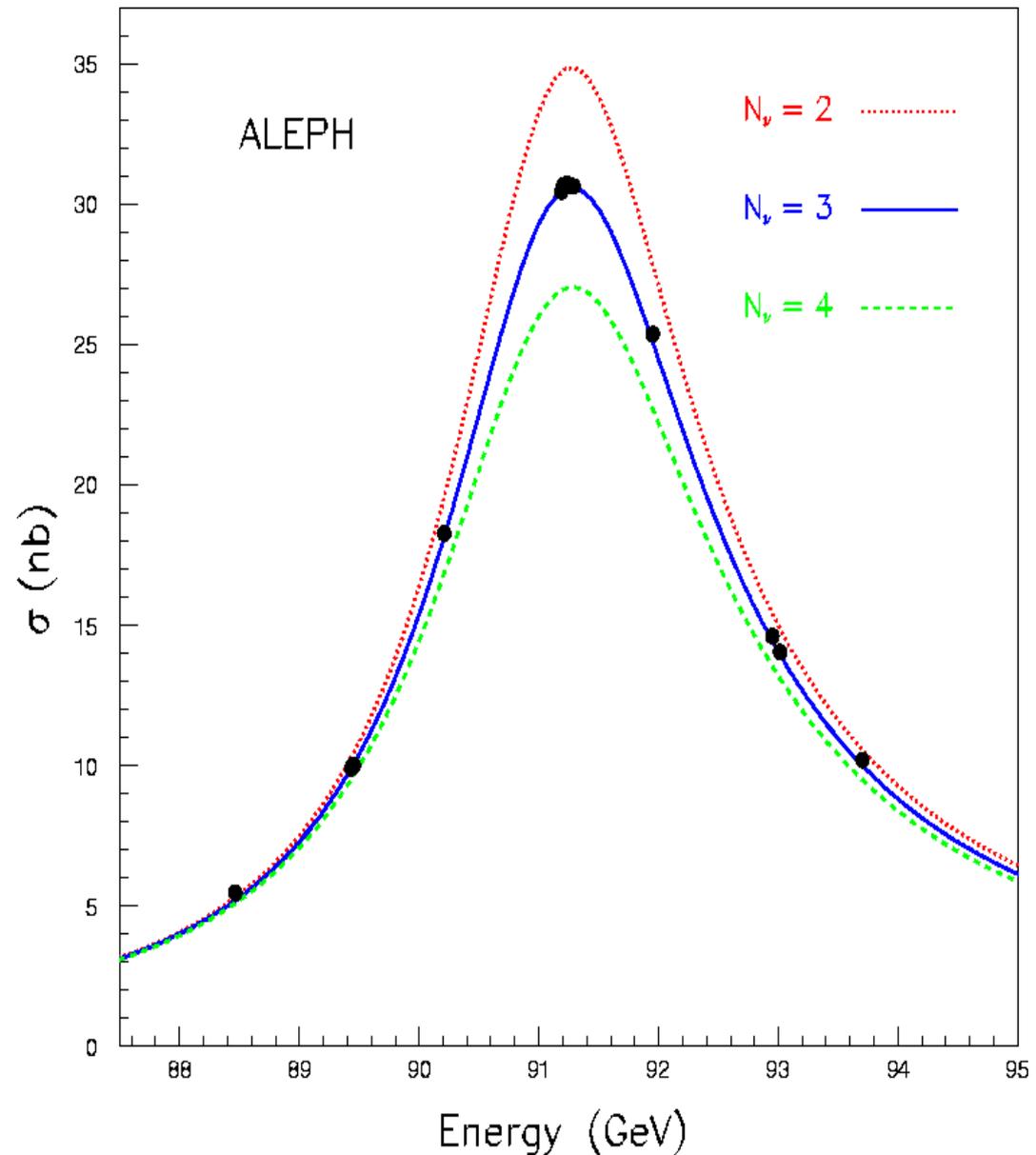
Analisi Dati

- Caso tipico:
 - grossa quantità di numeri di cui fare media, RMS ecc
 - visualizzare gli andamenti
- fare input-output da uno strumento
 - o da un file per noi
- per il momento ci limitiamo a questo caso semplice

Misure molto indirette (LEP)

... ancora più indiretto

- con i dati raccolti ed un modello teorico fare la simulazione per vedere cosa ci si aspetta dato un certo numero di famiglie di neutrini, e confrontare con i dati



Come fare?

- Esistono dei programmi commerciali per fare analisi dati
- Excel è uno di questi, ma non è particolarmente pensato per la fisica (avete mai provato a fare un istogramma con Excel?)
- certamente più per gli investimenti in borsa
- Costa!!!!

Soluzioni open source

- useremo queste, sono state prodotte prevalentemente dai fisici delle particelle a alte energie
 - quelli con più dati da analizzare
 - quelli con le collaborazioni più grandi
 -
 - ... quelli con più soldi nei decenni passati

ROOT

- Stato dell'arte al momento
- Sviluppato al CERN
- Usato per ... tutto, ormai anche dalle società finanziarie per simulazioni di borsa
- <http://root.cern.ch>
- manuale:
- <http://root.cern.ch/root/doc/RootDoc.html>
- download
- <http://root.cern.ch/root/Availability.html>



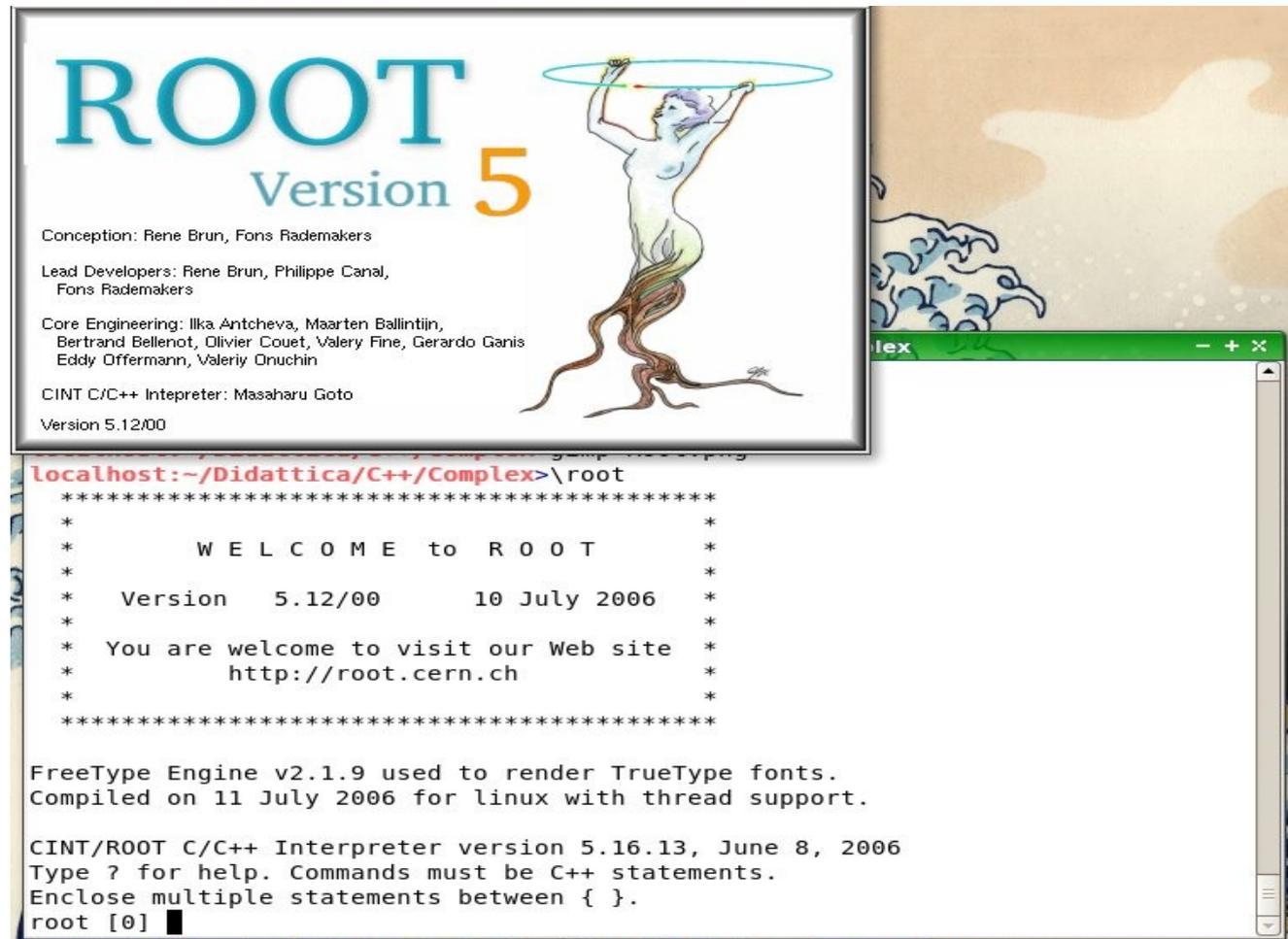
Come partire

- Se ve lo hanno installato (si!)

- root

- per uscire

- .q



Come installarlo

- Dalla web page di root (<http://root.cern.ch/>)
- ->Download Binaries
- ->Pro
- Selezionare il binario a seconda del proprio OS (c'e' anche per Win)
 - `cat /etc/redhat-release` (per OS release su RH)
 - `gcc -v` (per versione gcc)
- Scaricare il tar.gz
- Da qualche parte: ***tar xzvf root....tgz***
- *Leggere README/README e seguire le istruzioni*
- *Ci sono da settare alcune variabili di ambiente*

Ambiente (linux)

```
(bash) nel ~/.bashrc
```

```
export ROOTSYS=/<MyPathToRoot>
```

```
export PATH=$PATH:$ROOTSYS/bin
```

```
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
```

```
(csh/tcsh) in ~/.login
```

```
setenv ROOTSYS /root      (in csh or tcsh)
```

```
set path=($path $ROOTSYS/bin)      (in csh or tcsh)
```

```
setenv LD_LIBRARY_PATH $ROOTSYS/lib:$LD_LIBRARY_PATH
```

Cosa fa

- In generale, un programma di analisi dovrebbe offrire la possibilità di interazione con l'utente, che vuole dare comandi del tipo
 - crea un vettore di numeri
 - riempilo dal file pippo.dat
 - fammelo vedere come istogramma
 - fai un fit con una gaussiana
- deve quindi definire un linguaggio, più o meno come le machine unix che avete davanti a voi
 - ls, cd ecc ecc

Idea di base (furba)

- L'idea di root è
 - **perché definire un linguaggio per questo, usare direttamente il C++!**
- Capiamo cosa questo voglia dire
 - fino ad ora, voi avete scritto programmi in C++
 - poi avete compilato con g++
 - poi avete eseguito
- questo naturalmente non può funzionare in interattivo
 - è come se per fare ls doveste scrivere ls in un file, compilarlo ed eseguirlo

Interprete C++

- root contiene un interprete C++, cioè un sistema che si sostituisce al compilatore e esegue i comandi uno per uno
- quindi
 - scriviamo direttamente codice C++ nella shell di [root]
- **int a=4+5;**
- **cout <<"Ecco a " << a <<endl;**

...

Note...

- nota: se non metto il ; finale, funziona, ma mi dice che sta facendo (per il debug)...
- Cout stampa solo se alla fine mettete l'endl
- quindi
 - qualunque codice C++ funziona senza modifiche all'interno di root, anche se più lentamente (perche' e' interpretata, non compilata)
- Proviamo con i nostri numeri complessi

Estendere root con i complessi

- se `Complex.h` era il file che includevamo per usare i complessi, da riga di comando possiamo dire

carica in memoria `Complex.cc`

- **`.L Complex.cc`**

- e `root` interpreta il file e d'ora in poi ci permette di usare i complessi così come noi li abbiamo definiti
- Quindi
 - **`Complex c(3,6)`**
 - **`cout << c<<endl;`**

Oppure...

- possiamo direttamente utilizzare un programma, invece di scrivere i comandi ogni volta uno per uno
 - Unica limitazione: non viene eseguito **main**, ma una funzione dello stesso nome del file
 - vuole .C come suffisso del file ...
- **.x mandel.C**
- funziona, ma è lento, visto che deve interpretare riga per riga

Posso renderlo piu' veloce

- se faccio
 - **.x mandel.C++**
- il programma viene compilato in modo standard con g++ e eseguito
- quindi non perdo in velocità
 - Nota: in questo caso non si usa l'interprete C++, ma il compilatore, che e' piu' rigoroso per la sintassi!

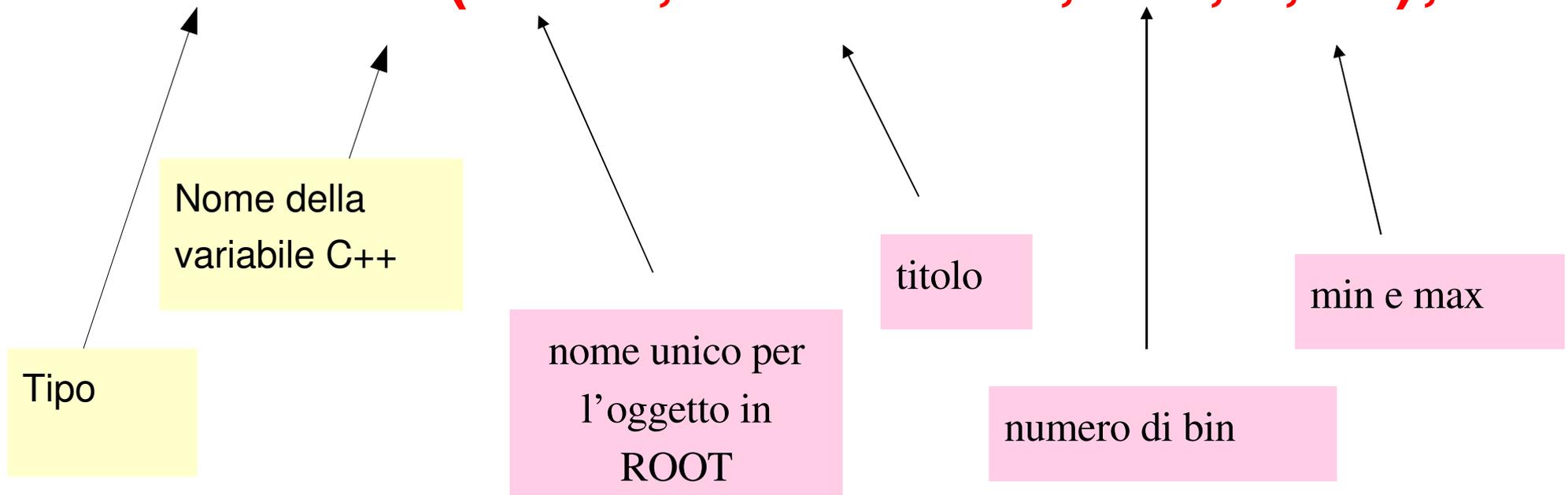
Naturalmente...

- se fossero queste le funzionalità sarebbe inutile, visto che sarebbe solo un altro modo per fare le stesse cose che abbiamo fatto fino ad ora.
- ROOT però è anche una estensione del C++, in particolare aggiunge buona parte delle caratteristiche necessarie per l'analisi dati
 - **visualizzazione**
 - **analisi statistica**
 - **input/output da sorgenti di ogni tipo**
- **Estensione: definisce classi che estendono in linguaggio C++**

Istogrammi

- in ROOT il semplice istogramma 1D si chiama **TH1F**
- E' una classe, con una interfaccia molto completa!
- Eg: costruttore:

■ **TH1F isto("isto", "Il mio isto", 100, 0, 10);**



Sapete cosa e' un istogramma?

- In pratica è una serie di secchi, ognuno dei quali può contenere un certo numero di palline
- in pratica, la sua visualizzazione dà un'idea della popolazione relativa dei vari secchi
- Esempio:

Appello del 01 dicembre 2003

Numero partecipanti: 68

Score minimo: 22%

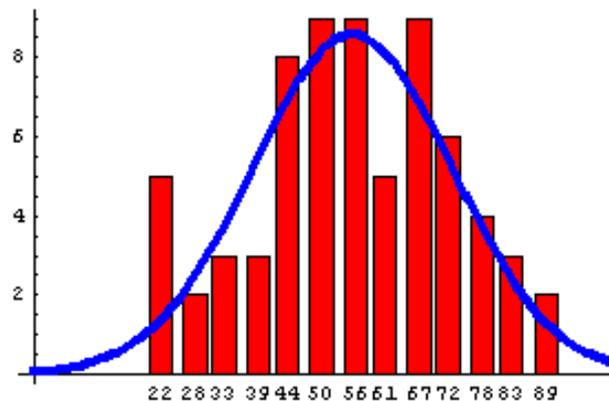
Score massimo: 89%

Score medio: 55.3971%

Score mediano: 56%

Sigma score: 17.4083%

Istogramma



Rispetto alla [tabella standard di corrispondenza](#) sono stati rimodulati con 18/30 i punteggi da 60% a 64% compresi (5 candidati su compresi (0 candidati su 64).

Operazioni

- Cosa voglio fare su di un istogramma?
- Quale sono le cose che sa fare un istogramma (interfaccia)
 - riempirlo  `isto.Fill(45);`
 - visualizzarlo  `isto.Draw();`
`isto.DrawCopy();`
 - calcolare quantità elementari  `isto.GetMean();`
`isto.GetRMS();`
`isto.GetEntries();`
- Facciamo una prova: creiamo un istogramma e riempiamolo con numeri casuali

Numeri casuali

- ROOT ha molto più di rand(), utilizzando l'oggetto `gRandom`
- È un puntatore!!!!
 - `gRandom->Rndm()` distribuzione flat [0,1)
 - `gRandom->Gaus(0,1)` distribuzione gaussiana di media 0 e sigma 1
 - `gRandom->Poisson(4)` distribuzione poissoniana con media 4
 - `gRandom->Binomial(4,.4)` distribuzione binomiale, con 4 lanci e probabilità 40%

Esercizio

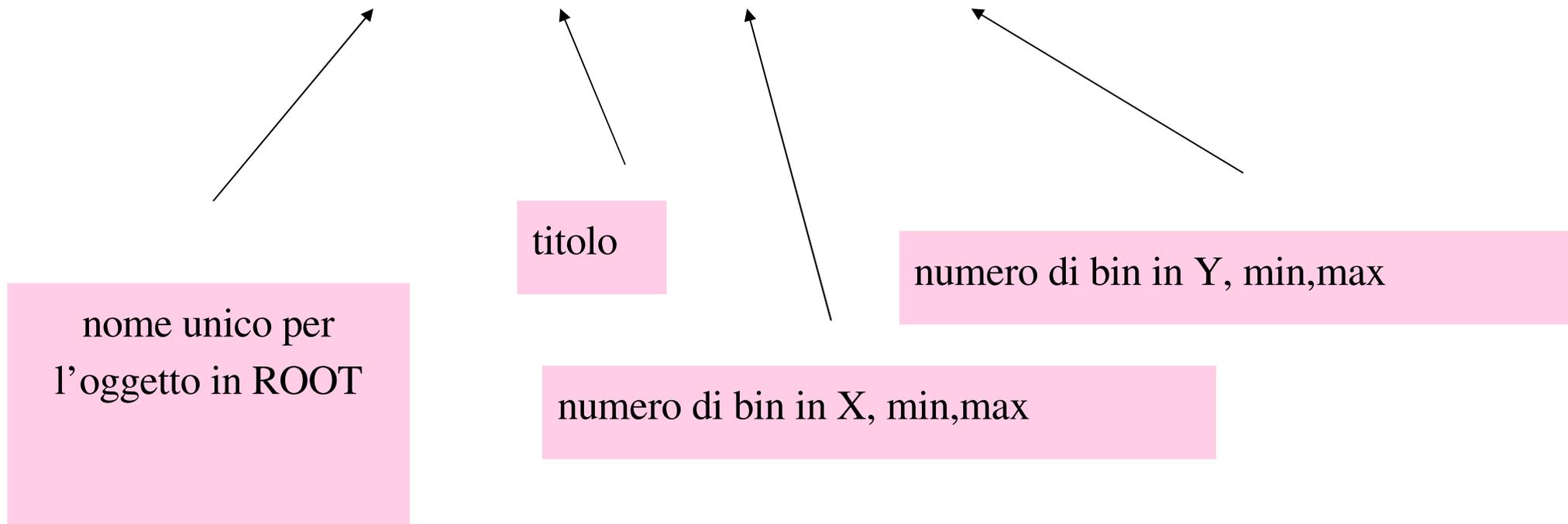
- Facciamo
 - Creiamo un istogramma
 - Lo riempiamo con numeri casuali secondo una certa distribuzione (eg gaussiana)
 - Lo visualizziamo
- Poi facciamo lo stesso per una distribuzione bi-dimensionale
 - Devo fare un istogramma 2d
- `TH2F h2d("pluto","isto 2d", 100,-2,2,100,-1.5,1.5);`
 - Genero x secondo gaus, y secondo Poisson (eg)
 - Riempio "pluto" con x,y
 - `h2d.Fill(x,y);`

Mandelbrot...

- finalmente.....
- invece di ritornare vero o falso per dire se un punto appartiene o meno a mandelbrot, facciamo una funzione che ritorna il numero di iterazioni per la divergenza (o il max se non diverge...)
 - `int mandel(Complex c) { ... }`
- Poi per ogni punto del piano (`c.real()`,`c.imm()`) mettiamo un colore che corrisponde al numero di iterazioni

Visualizziamo Mandelbrot

- Una griglia? Posso usare un istogramma 2D, che in pratica è una matrice
- Facciamo un istogramma 2d che rappresenta la nostra griglia di punti sul piano (diciamo 200*200) NB ci mettera' un po' ...
 - TH2F h2d("pluto", "isto 2d", 200, -2, 2, 200, -1.5, 1.5);



Riempio l'istogramma

- Per ogni bin creo un numero complesso
 - Complex $c(x,y)$;
- Riempio l'istogramma con $\text{mandel}(c)$ (e' in intero, num iterazioni)
 - $\text{h2d.Fill}(c.\text{real}(), c.\text{imm}(), \text{mandel}(c))$;
- Attenzione: Fill “aggiunge”, quindi devo essere sicuro di fare Fill solo una volta per ogni bin!
 - Faccio iterazione sui bin e calcolo $\text{mandel}(c)$ per ogni bin
- In alternativa uso h2d.SetBinContent (non aggiunge, bin)
 - $\text{h2d.SetBinContent}(3,6,22)$
- Decido che il terzo bin in X, il 6o

$$x = i * \text{step} + x_{\min}$$
$$\text{step} = \frac{(x_{\max} - x_{\min})}{N\text{bin}X}$$

Prego...

Mandelbrot

