



Corso di Laboratorio di Meccanica e Termodinamica

Modulo ROOT

Lezione II

Silvia Arcelli

27 Marzo 2024

Istogrammi

Per “costruire” un istogramma (per es. 1-D):

Come puntatore:

```
TH1F *h1 = new TH1F("name","Title", nBins, xlow, xhigh);
```

Come istanza:

```
TH1F h1 = TH1F("name","Title", nBins, xlow, xhigh);
```

Per riempire ciascuna occorrenza della variabile x:

```
h1->Fill(x); (puntatore)
```

```
h1.Fill(x); (istanza)
```

NB: il metodo Fill ha anche un altro argomento, che è di default 1, ed è il “peso” di quell’ingresso:

Per esempio:

```
h1->Fill(x,8); //sono inserite simultaneamente 8 occorrenze  
              //identiche di x
```

Istogrammi

Per disegnare un istogramma su una finestra grafica:

```
h1->Draw() ; (puntatore)
```

```
h1.Draw() ; (istanza)
```

Disponibili varie opzioni e metodi per la grafica, ed es.:

```
h1->Draw("E") ; //visualizza gli errori
```

```
h1->SetMarkerStyle(22) ; //scelgo il tipo di simbolo (triangolo pieno)
```

```
h1->Draw("E,P") ; // con errori e simbolo sul punto
```

```
h1->Draw("HIST,SAME") ; // sovrappongo stesso istogramma  
con linea continua
```

Istogrammi-Opzioni Grafiche

`h1.SetMarkerStyle();`



`h1.SetFillColor();`



LineStyle

`h1->SetLineStyle();`



LineColor

`h1.SetLineColor();`

(Per molti di questi codici ci sono degli alias (enum), tipo `kBlue=4`, `kOpenCircle=24`, etc.)

Altri Metodi degli Istogrammi

Command	Parameters
<code>h1.GetMean()</code>	Mean
<code>h1.GetRMS()</code>	Root of Variance
<code>h1.GetMaximum();</code>	Maximum bin content
<code>h1.GetMaximumBin();</code>	location of maximum
<code>h1.GetBinCenter(int bin_number);</code>	Center of bin
<code>h1.GetBinContent(int bin_number);</code>	Content of bin

```
h1->GetBinError(ibin) //ritorna l'errore sul contenuto del bin
h1->SetBinContent(ibin,val) //assegna al contenuto del bin "ibin" il
valore "val"
h1->SetBinError(ibin,val) //assegna all'errore sul contenuto del bin
"ibin" il valore "val"
```

Per gli ingressi «fuori range»:

```
h1->GetBinContent(0) //ritorna il numero di UNDERFLOW
h1->GetBinContent(Nbin+1) //ritorna il numero di OVERFLOW
```

Altri Metodi degli Istogrammi

- **Ingressi:**

- `h->GetEntries()` // numero totale di conteggi
- `h->Integral(ibin1, ibin2)` ; // integrale nel range [ibin1, ibin2]
- `h->Integral()` ; // integrale totale, esclusi gli underflow e overflow
- `h->GetIntegral()` ; // array dei conteggi cumulativi

- **Incertezze su Media e RMS**

- `h->GetMeanError()` ;
- `H->GetRMSError()` ;

Per la RMS, comandi equivalenti sono:

`h->GetStdDev()` ;

`h->GetStdDevError()` ;

ROOT Global Variables

Incontrerete anche una serie di **puntatori globali** definiti automaticamente durante la sessione di ROOT, sempre disponibili. Alcuni esempi:

- **gROOT**: informazione globale relativa alla sessione corrente attraverso il quale si può accedere praticamente a qualunque oggetto creato durante la sessione di ROOT
- **gFile**: puntatore al root file corrente
- **gStyle**: puntatore alle funzionalità per gestire lo “stile” grafico
- **gRandom**: puntatore al generatore di numeri random (in maggior dettaglio nelle lezioni V e VI)

ROOT File

ROOT ha un'interfaccia di IO (Input/Output) propria, implementata nella classe **TFile**:

- Per **aprire** un file ROOT:

```
TFile *file = new TFile("example.root", "RECREATE") ;
```

- Per **scrivere un oggetto** di ROOT su un ROOT File, per esempio un istogramma indirizzato dal puntatore h:

```
h->Write() ;
```

- Per **chiudere** il file root:

```
file->Close() ;
```


Istogrammi-Esempio I:

Named macro (histo.C) che esegue il seguente compito:

- Riempire un istogramma monodimensionale di 100 bin da -5. a 5., con le **nev** estrazioni da una distribuzione gaussiana con $\mu = \text{mean}$ e $\sigma = \text{width}$ (nev, mean e width parametri di ingresso della macro, a cui attribuite rispettivamente un valore di default di 1.E5, 0., 1.)
- Disegnatelo in modo da visualizzare, oltre alla linea continua, anche le incertezze sugli ingressi dei bin.
- Scrivere l'istogramma risultante su un file root

Istogrammi-Esempio I:

In dettaglio:

- Editare un file histo.C
- Definire una funzione C++ : ad esempio, void histo(....)
- All'interno della funzione:
 - Creare istogramma
 - Riempire l'istogramma con nev estrazioni da una gaussiana (usare gRandom->Gaus(mean,width) per generare ogni singola occorrenza)
 - Disegnarlo con il metodo Draw()
 - Aprire un file root
 - Scrivere l'istogramma sul file root
 - Chiudere il file root
- Salvare histo.C ed eseguirlo (da linea di comando di root fare .L histo.C e poi histo())

histo.C versione 2:

Modificare la macro per stampare a schermo:

- Numero di Ingressi
- Media e RMS con relativi errori
- Il massimo contenuto e la sua posizione
- Underflow e Overflow

Presentazione: Aprite il file root su cui è stato scritto l'istogramma e manipolatelo attraverso l'interfaccia grafica:

- Fill di colore blu
 - Titoli degli assi
 - Tabella della statistica: numero di ingressi, media e RMS con i loro errori, Under(Over)Flows
 - In scala semilogaritmica, con griglia
- salvarlo in formato grafico, formato .C e .root

Generazione con $1E7$ estrazioni : considerazioni sul risultato (come scala l'incertezza su media e rms rispetto alla generazione con $1E5$ estrazioni? Gli Underflow e Overflow osservati sono consistenti con quanto ci si aspetta?)

Proprietà statistiche

Tabella Probabilità Gauss

n	$p = F(\mu + n\sigma) - F(\mu - n\sigma)$	i.e. $1 - p$	
1	0.682 689 492 137	0.317 310 507 863	
2	0.954 499 736 104	0.045 500 263 896	
3	0.997 300 203 937	0.002 699 796 063	
4	0.999 936 657 516	0.000 063 342 484	
5	0.999 999 426 697	0.000 000 573 303	
6	0.999 999 998 027	0.000 000 001 973	

Incertezza sulla media del campione:

$$\frac{\sigma}{\sqrt{N}}$$

Incertezza sulla deviazione standard del campione:

$$\frac{\sigma}{\sqrt{2N}}$$

Istogrammi-Esempio II:

Scrivere una macro che esegua il seguente compito:

- Rappresentazione dei Tempi di Caduta del Pendolo di Maxwell (una gaussiana con media 7.5 s e deviazione standard 2 s)
- Creare un istogramma monodimensionale
- lettura dati da file (maxwell.dat) di ogni singola “entry” e riempimento dell’istogramma
- Inserire label degli assi e alcuni “abbellimenti” **con comandi C++**
- Disegnare istogramma su una canvas
- Stampare a schermo numero di ingressi, media e RMS dell’istogramma, con relativi errori.
- Salvare la Canvas in formato .C, root e pdf

Su Virtuale: **maxwell.C,maxwell.dat**

Operazioni su Istogrammi

Operazioni (su istogrammi “omologhi”, stesso numero di bin e stesso range). Si può fare attraverso:

-Overload degli operatori somma, moltiplicazione etc.

```
TH1F h1;  
TH1F h2=3*h1  
TH1F h3=h1+h2;
```

N.B. Questa modalità funziona solo per le istanze e non per i puntatori!

-Oppure attraverso i metodi della classe (raccomandato):

```
h->Add(...) //somma  
h->Multiply(...) //moltiplicazione  
h->Divide(...) //divisione
```

Operazioni su Istogrammi

Per dividere e sommare/dividere due istogrammi utilizzate:

- **Il Metodo Add**, virtual Bool_t Add (const TH1 * h1, const TH1 *h2, Double_t c1=1, Double_t c2=1) // this= c1*h1+c2*h2 (per sottrarre due istogrammi mettere c2=-1)
- **Il Metodo Divide**, virtual Bool_t Divide (const TH1 *h1, const TH1 *h2, Double_t c1=1, Double_t c2=1, Option_t *option="") //this= c1*h1/c2*h2
- Bisogna prima creare un istogramma “extra” che contenga l’esito della somma e della differenza. Fatelo con il Copy Constructor:
- **Il Copy Constructor** di TH1F: TH1F::TH1F (const TH1F & h1)

Operazioni su Istogrammi

//creazione dei due istogrammi

```
TH1F *h1 = new TH1F("h1","Tempi di Caduta 1",8,-0.5,15.5);
```

```
TH1F *h2 = new TH1F("h2","Tempi di Caduta 2",8,-0.5,15.5);
```

//....// cicli di lettura dati e riempimento istogrammi

//sommiamo i due campioni in un unico istogramma

```
TH1F *hSum=new TH1F(*h1);
```

```
hSum->Add(h1,h2,1,1);
```

//Rapporto fra i due istogrammi:

```
TH1F *hRatio=new TH1F(*h1);
```

```
hRatio->Divide(h1,h2,1,1);
```


Operazioni su Istogrammi

- **Applicazione:**
 - Costruire due istogrammi/riempirli (**costruttore**)
 - Leggere da due file distinti (files di input su Virtuale, **maxwell1.dat,maxwell2.dat** ciascuno contenente 60 misure di tempo) e riempire i rispettivi istogrammi (**metodo Fill()**)
 - Farne la somma (combinare i due esperimenti), usando il **metodo Add(...)**
 - Farne il rapporto (dovrebbe fluttuare entro gli errori intorno a 1) (usare **metodo Divide(...)**)
 - Rappresentare i 4 istogrammi in una **TCanvas divisa in 4 parti**
 - Estrae Media e RMS della distribuzione somma con rispettive incertezze (metodi **GetMean()**, **GetMeanError()** etc..)