Extreme
Energy
Events
Science inside Schools

ROOT @ EEE

Lezione 3

Realizzazione di istogrammi

RUN COORDINATION MEETING – 19 Febbraio 2020

Lezioni precedenti:

- Lettura e scrittura da file testo

https://agenda.centrofermi.it/event/162/contributions/1367/attachments/690/1026/ROOT_lesson1.pdf

- Realizzazione di grafici per punti

https://agenda.centrofermi.it/event/164/contributions/1370/attachments/695/1033/ROOT_lesson2.pdf

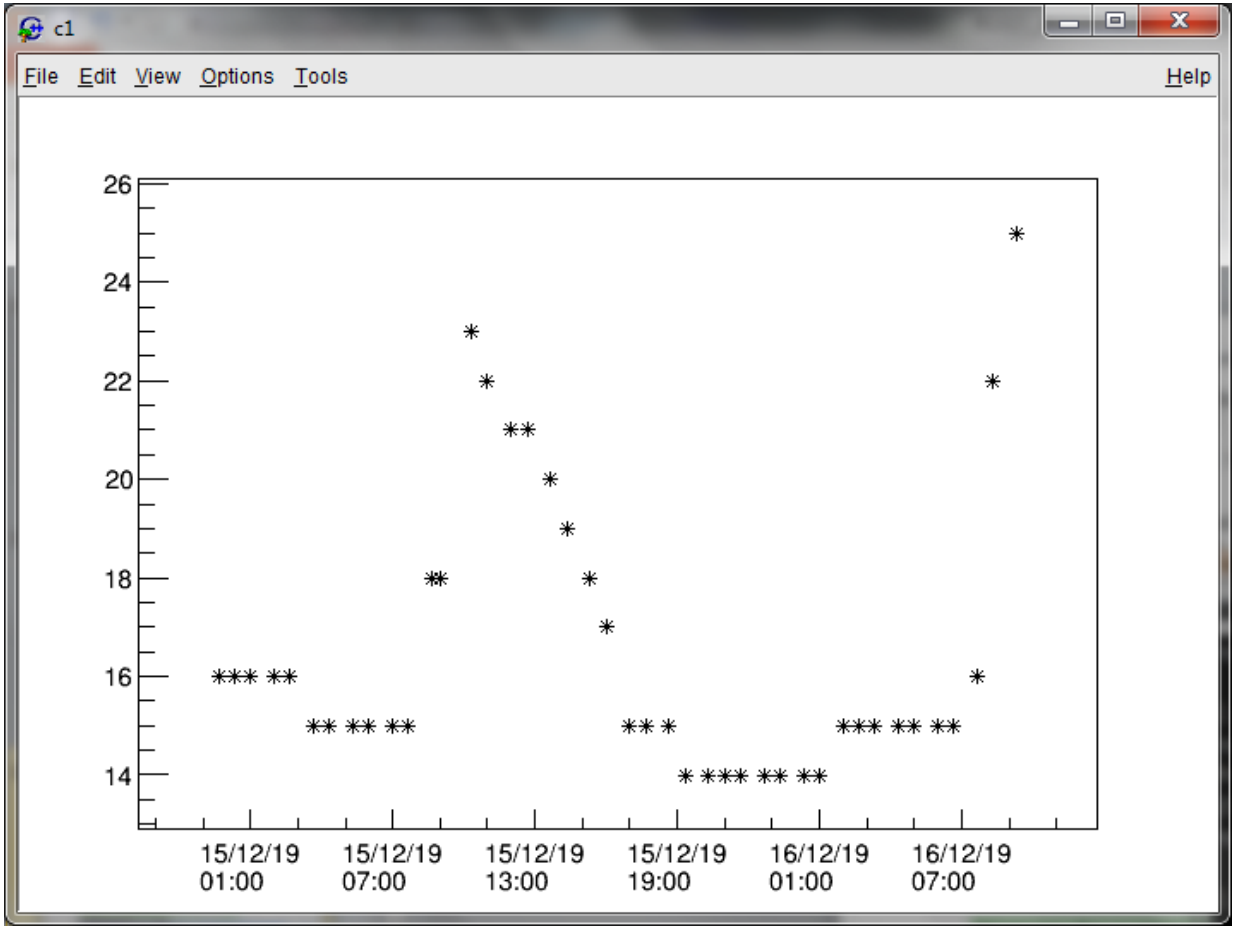
Lezione 1: Lettura e scrittura da file testo

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        double Tin_media = 0;
        int nrighe = 0;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                Tin_media = Tin_media + Tin;
                nrighe++;
                }
        Tin_media = Tin_media / (double) nrighe;
        cout << Tin_media << endl;
        ofstream fileout("D:\\EEE\\ROOT lessons\\out.txt");
        fileout << Tin_media ;


}
```

Lezione 2: Realizzazione di grafici per punti

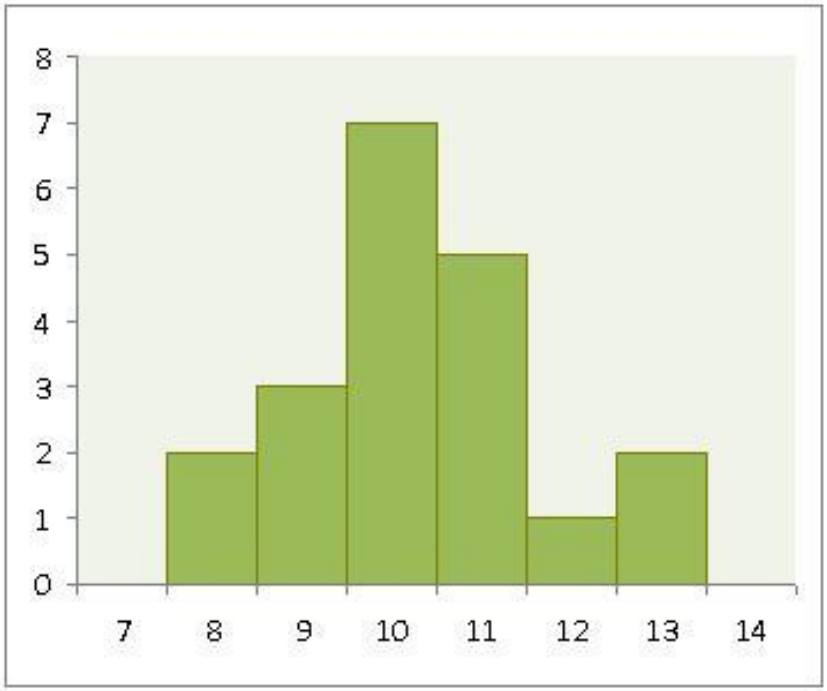
```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;
        TGraph *g = new TGraph();

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        double Tin_media = 0;
        int nrighe = 0;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                Tin_media = Tin_media + Tin;
                g-> SetPoint(nrighe, secondi, Tout);
                nrighe++;
                }
        Tin_media = Tin_media / (double) nrighe;
        cout << Tin_media << endl;
        ofstream fileout("D:\\EEE\\ROOT lessons\\out.txt");
        fileout << Tin_media ;

        TCanvas *c = new TCanvas();
        g->Draw("AP*");
        g->GetXaxis()->SetTimeDisplay(1);
        TDatime da(2007,01,01,00,00,00);
        gStyle->SetTimeOffset(da.Convert());
        g->GetXaxis()->SetTimeFormat("#splitline{%d\/%m\/%y}{%H:%M}");
        g->GetXaxis()->SetNdivisions(10,5,0,kTRUE);
        g->GetXaxis()->SetLabelOffset(0.030);
}
```

Lezione 2: Realizzazione di grafici per punti



ISTOGRAMMA

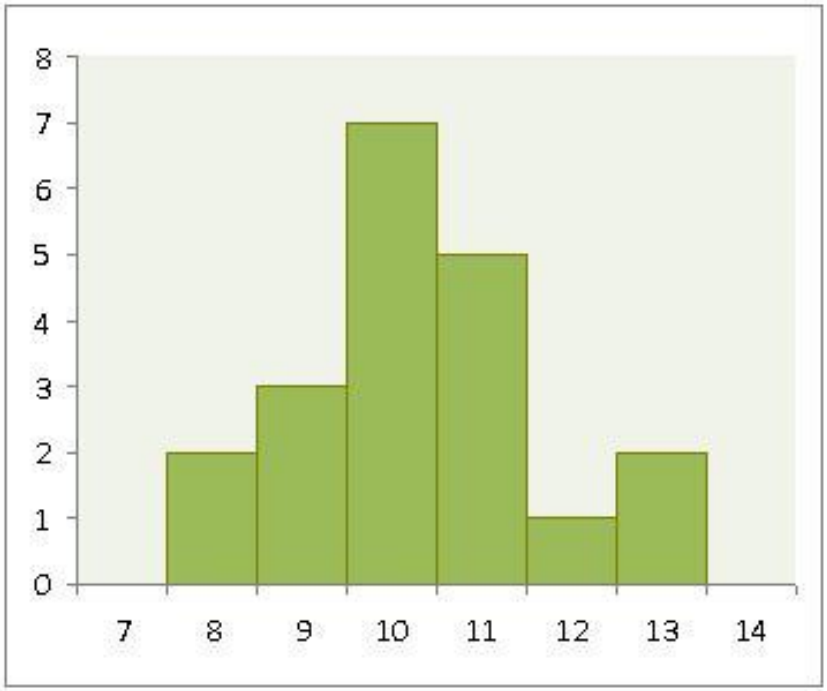
Un istogramma è un grafico che mostra la frequenza , o il numero di volte, che un determinato elemento appare in un intervallo specifico.



ISTOGRAMMA

Un istogramma è un grafico che mostra la frequenza , o il numero di volte, che un determinato elemento appare in un intervallo specifico.

Per definire un istogramma è necessario scegliere:

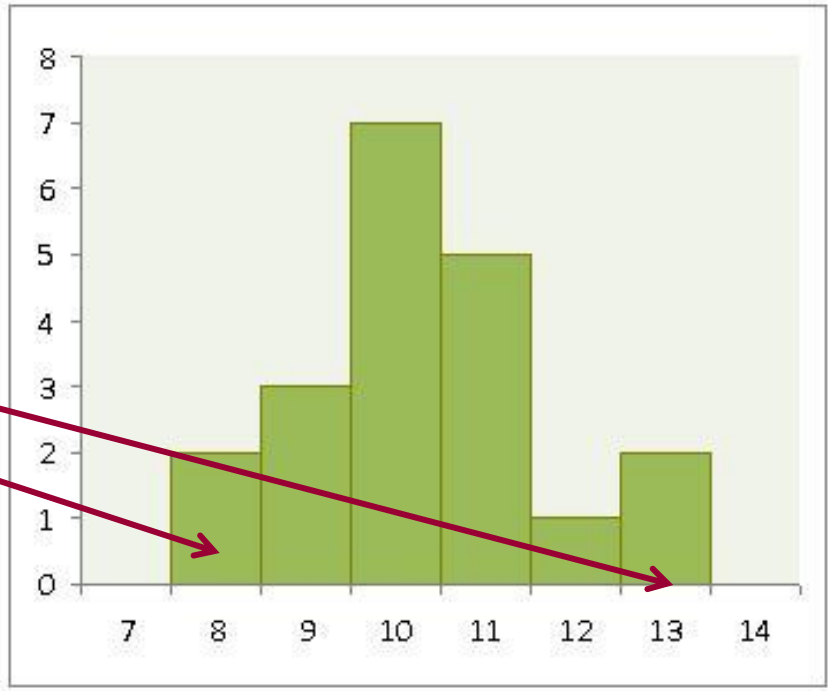


ISTOGRAMMA

Un istogramma è un grafico che mostra la frequenza , o il numero di volte, che un determinato elemento appare in un intervallo specifico.

Per definire un istogramma è necessario scegliere:

- *Range* [min; max]

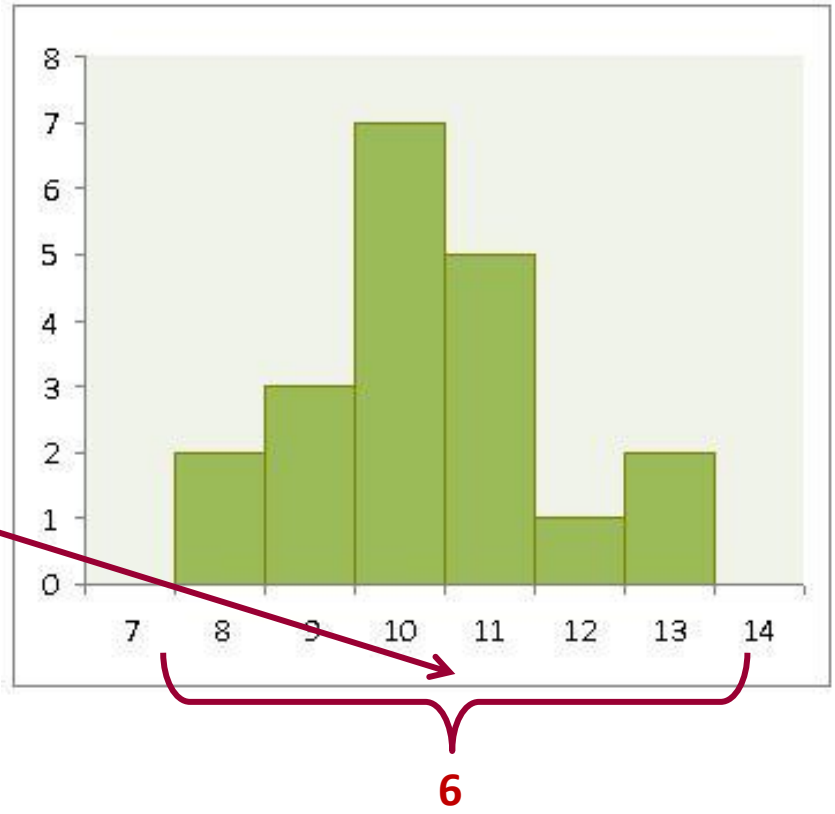


ISTOGRAMMA

Un istogramma è un grafico che mostra la frequenza , o il numero di volte, che un determinato elemento appare in un intervallo specifico.

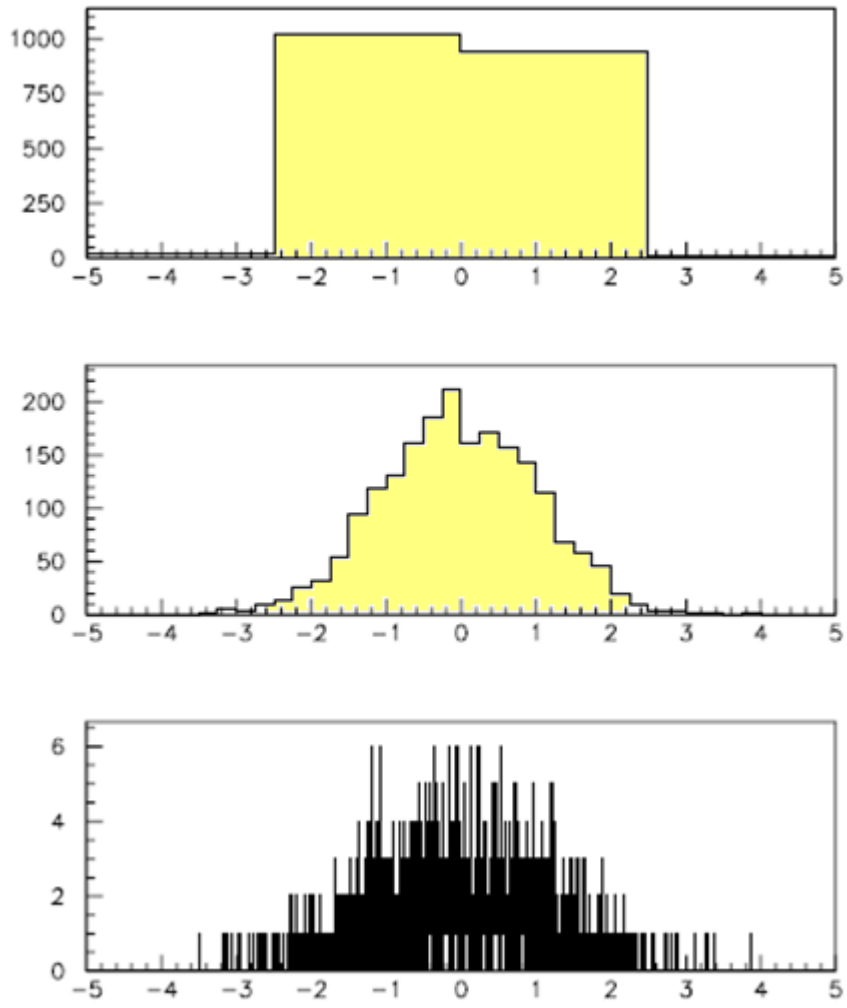
Per definire un istogramma è necessario scegliere:

- *Range* [min; max]
- Numero di intervalli (*bin*)



ISTOGRAMMA

La scelta del *range* e del numero di *bin* è di fondamentale importanza per poter avere una rappresentazione grafica efficace

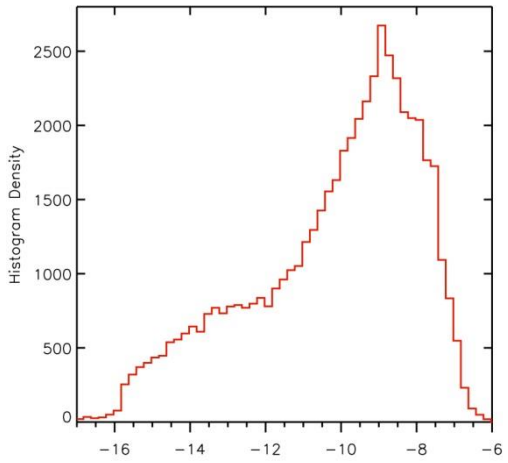


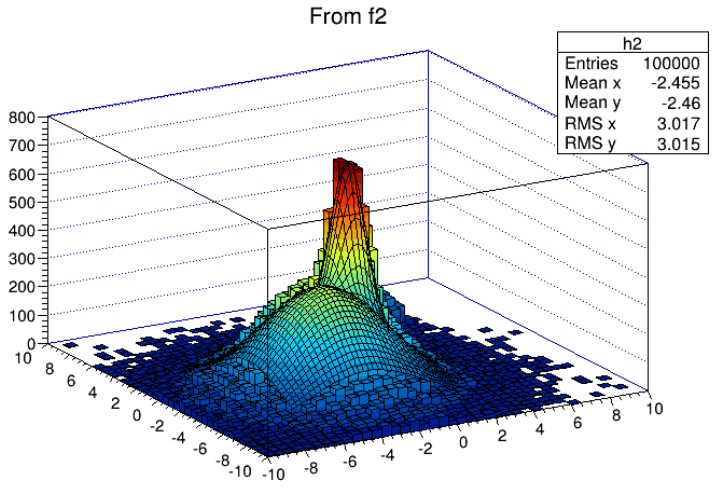
ROOT permette di gestire vari tipi di istogrammi:

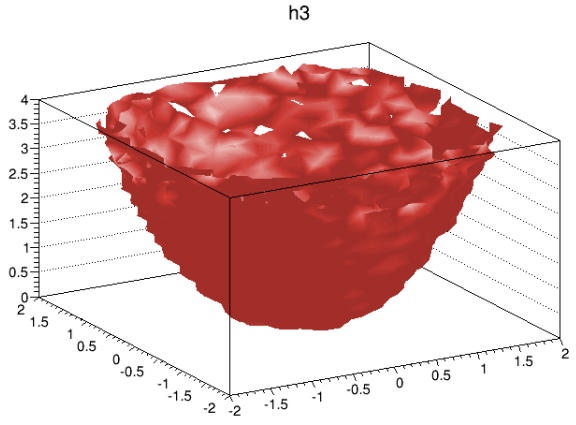
- 1-D
- 2-D
- 3-D

} short, integer, float, double

Esempio: TH1F *h = new TH1F("h", "il mio primo histo", 1000, 0, 100)







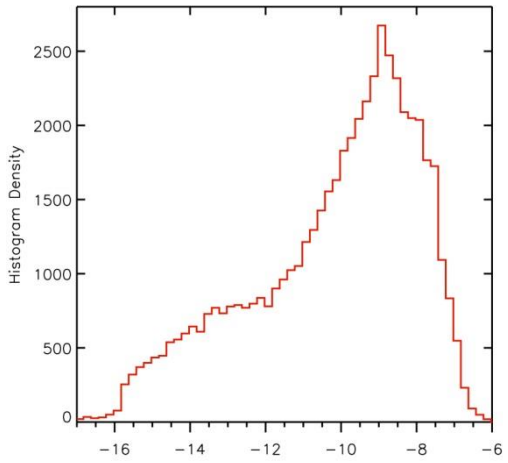
ROOT permette di gestire vari tipi di istogrammi:

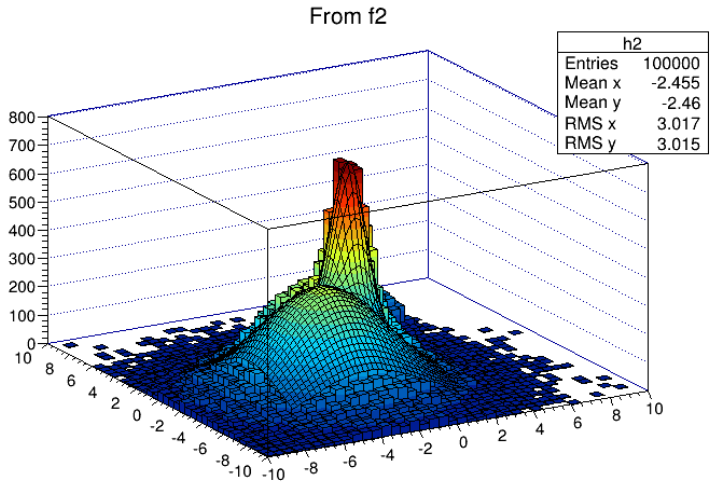
- 1-D
- 2-D
- 3-D

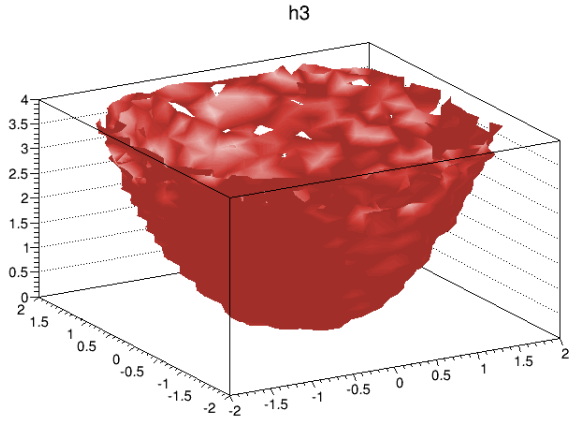
} short, integer, float, double

Nome

Esempio: TH1F *h = new TH1F("h", "il mio primo histo", 1000, 0, 100)







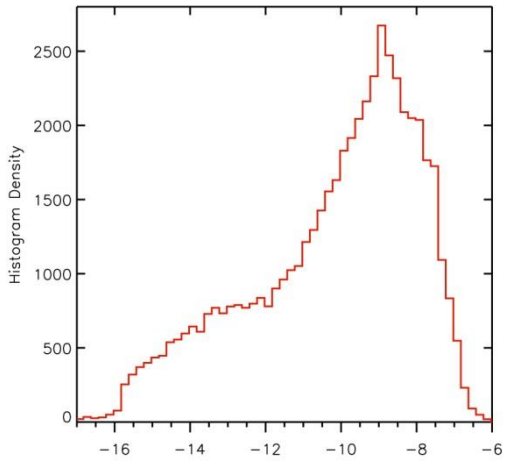
ROOT permette di gestire vari tipi di istogrammi:

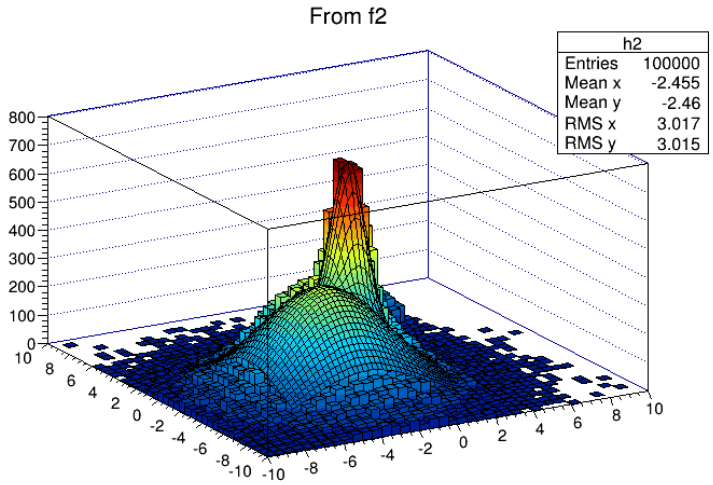
- 1-D
- 2-D
- 3-D

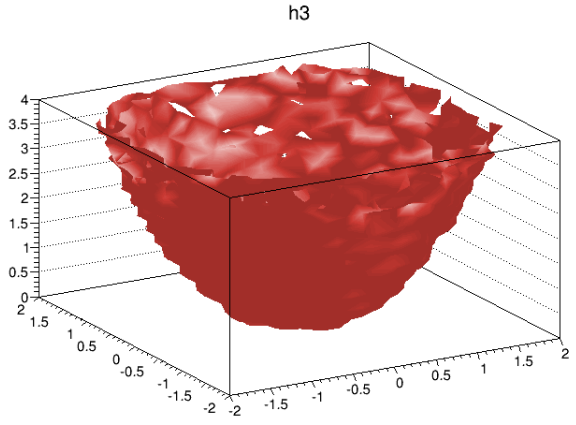
} short, integer, float, double

Nome Titolo

Esempio: TH1F *h = new TH1F("h", "il mio primo histo", 1000, 0, 100)







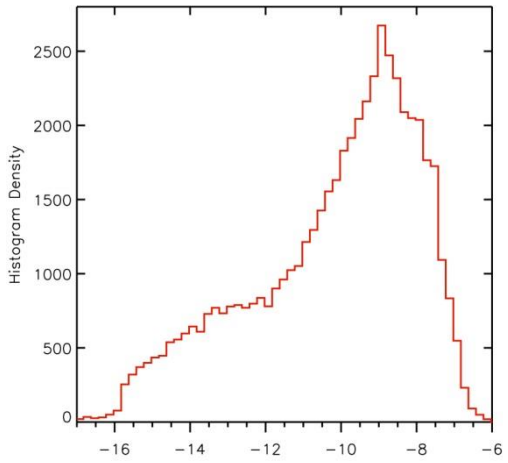
ROOT permette di gestire vari tipi di istogrammi:

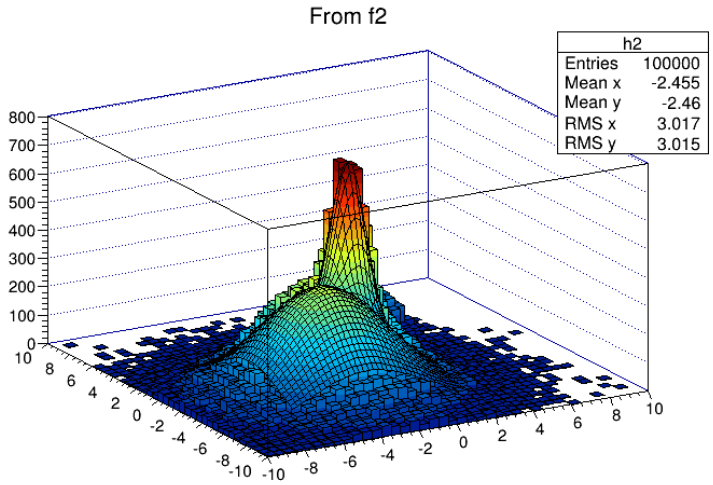
- 1-D
- 2-D
- 3-D

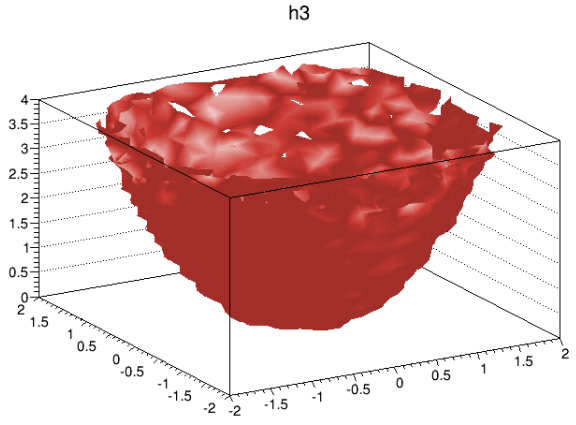
} short, integer, float, double

Nome Titolo Nbins

Esempio: TH1F *h = new TH1F("h", "il mio primo histo", 1000, 0, 100)







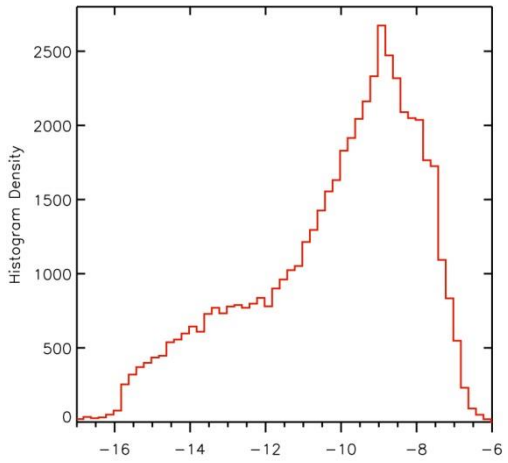
ROOT permette di gestire vari tipi di istogrammi:

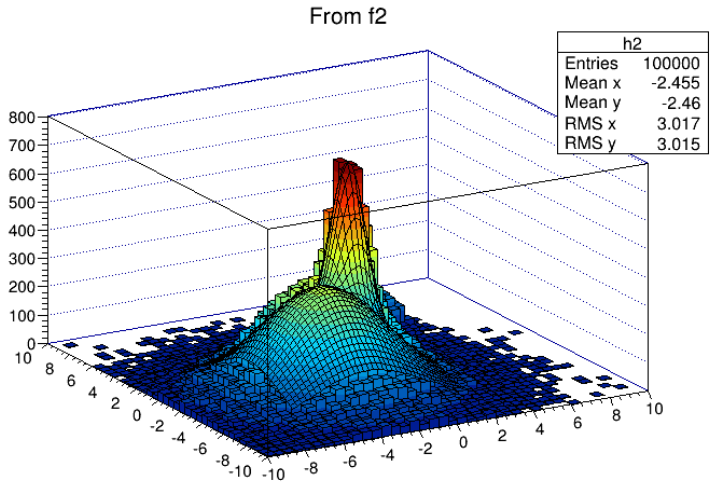
- 1-D
- 2-D
- 3-D

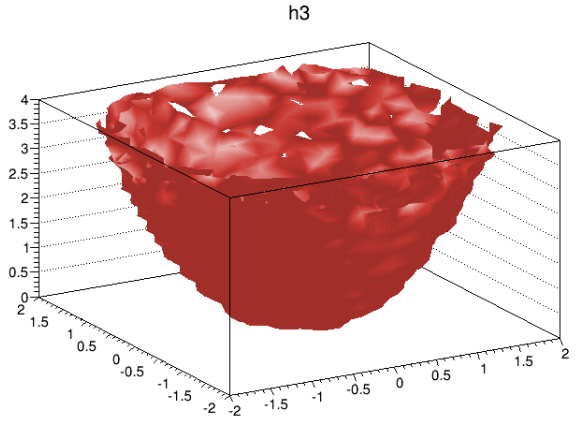
} short, integer, float, double

Nome Titolo Nbins Min

Esempio: TH1F *h = new TH1F("h", "il mio primo histo", 1000, 0, 100)







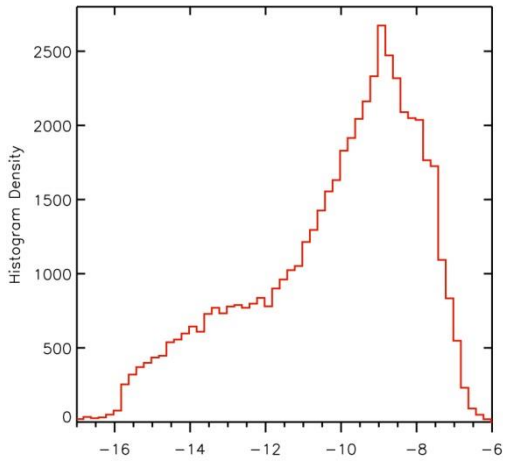
ROOT permette di gestire vari tipi di istogrammi:

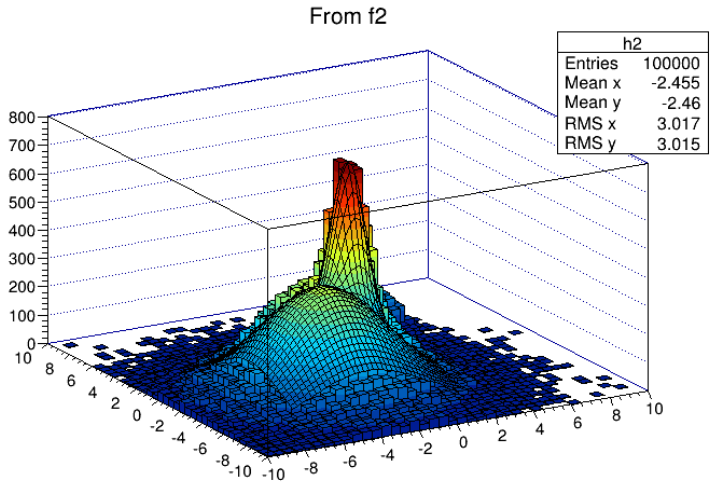
- 1-D
- 2-D
- 3-D

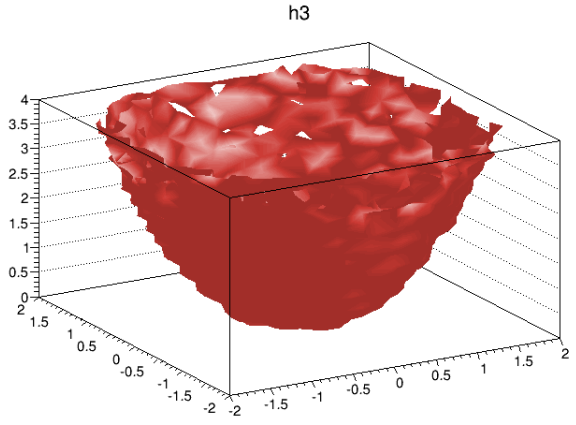
} short, integer, float, double

Nome Titolo Nbins Min Max

Esempio: TH1F *h = new TH1F("h", "il mio primo histo", 1000, 0, 100)







1-D histograms:

TH1C : histograms with one byte per channel. Maximum bin content = 127
TH1S : histograms with one short per channel. Maximum bin content = 32767
TH1I : histograms with one int per channel. Maximum bin content = 2147483647
TH1F : histograms with one float per channel. Maximum precision 7 digits
TH1D : histograms with one double per channel. Maximum precision 14 digits

2-D histograms:

TH2C : histograms with one byte per channel. Maximum bin content = 127
TH2S : histograms with one short per channel. Maximum bin content = 32767
TH2I : histograms with one int per channel. Maximum bin content = 2147483647
TH2F : histograms with one float per channel. Maximum precision 7 digits
TH2D : histograms with one double per channel. Maximum precision 14 digits

3-D histograms:

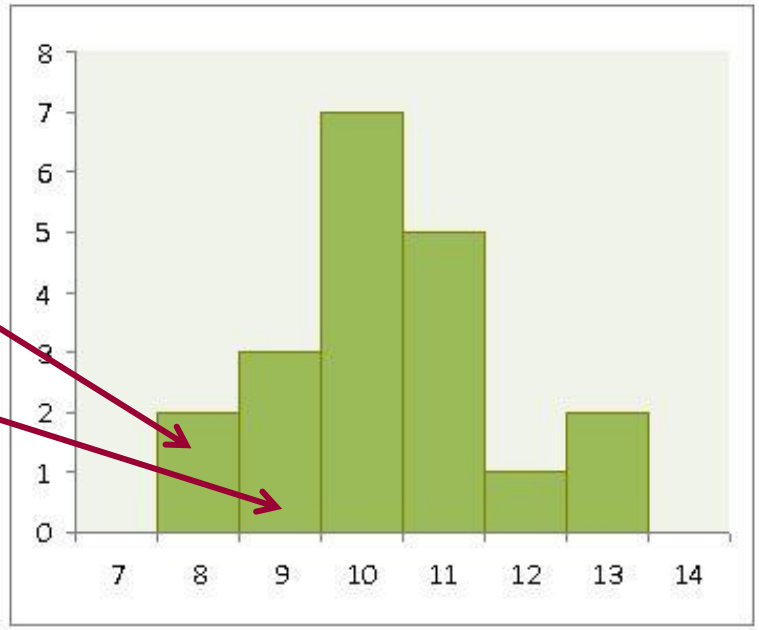
TH3C : histograms with one byte per channel. Maximum bin content = 127
TH3S : histograms with one short per channel. Maximum bin content = 32767
TH3I : histograms with one int per channel. Maximum bin content = 2147483647
TH3F : histograms with one float per channel. Maximum precision 7 digits
TH3D : histograms with one double per channel. Maximum precision 14 digits

Metodi comunemente usati per riempire un istogramma:

Fill

- si utilizza quando si dispone dell'elenco dei valori con cui riempire l'istogramma
- da ripetere tante volte quanti sono i valori da inserire nell'histo

```
h1->Fill(8)
h1->Fill(8)
h1->Fill(9)
h1->Fill(9)
h1->Fill(9)
...
```



Metodi comunemente usati per riempire un istogramma:

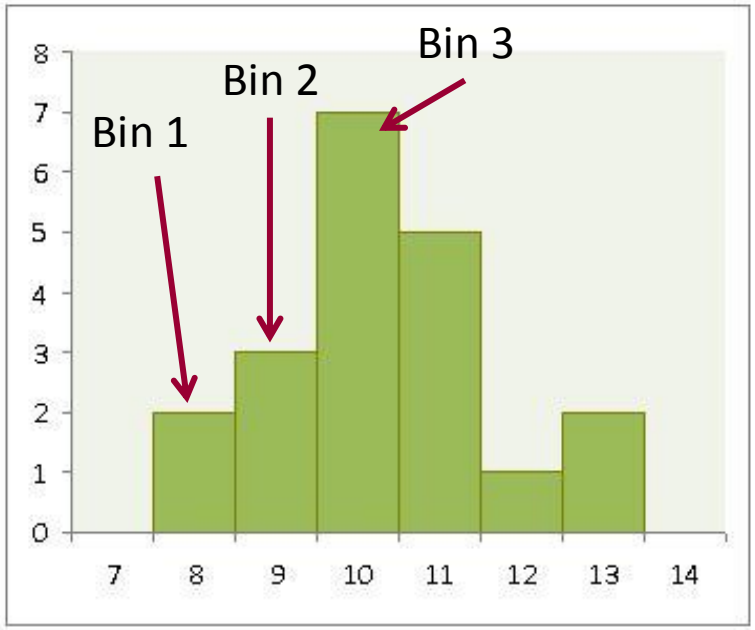
SetBinContent

- Si utilizza quando si conosce il numero di volte in cui si presentano i valori con cui riempire l'istogramma

```
h1->SetBinContent(1, 2)
h1->SetBinContent(2, 3)
...
```

Bin

Frequenza o meglio numero di volte in cui si presenta il valore corrispondente al bin indicato



ESEMPIO

Realizziamo un istogramma della temperatura esterna (estratta dal file "cvs weather")

- Definiremo:

  TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30)

- E lo riempiremo con l'istruzione

  hTout->Fill(Tout)

  all'interno del ciclo di lettura

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

Percorso file dati

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

Percorso file dati

Verifica esistenza del file

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

Percorso file dati

Verifica esistenza del file

Dichiarazione variabili utilizzate per dati in lettura

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

Percorso file dati

Verifica esistenza del file

Dichiarazione variabili utilizzate per dati in lettura

Dichiarazione histo

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

Percorso file dati

Verifica esistenza del file

Dichiarazione variabili utilizzate per dati in lettura

Dichiarazione histo

Lettura prima riga

```
{
        // inizio della macro

        ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
        if(filedati.fail())
                {
                cout << "Il file non esiste, verifica il percorso!" << endl;
                break;
                }
        string titolo1, titolo2, titolo3, titolo4, titolo5;
        double secondi, Tin, Tout, pressione, run;

        TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

        filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

        while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
                {
                hTout->Fill(Tout);
                }

        TCanvas *c = new TCanvas();
        hTout->Draw();
}
```

Percorso file dati

Verifica esistenza del file

Dichiarazione variabili utilizzate per dati in lettura

Dichiarazione histo

Lettura prima riga

Ciclo di lettura e riempimento histo

```
{
    // inizio della macro

    ifstream filedati("D:\\EEE\\ROOT lessons\\weather_trending.csv");
    if(filedati.fail())
            {
            cout << "Il file non esiste, verifica il percorso!" << endl;
            break;
            }
    string titolo1, titolo2, titolo3, titolo4, titolo5;
    double secondi, Tin, Tout, pressione, run;

    TH1D *hTout = new TH1D("hTout", "histo Tout", 15, 0, 30);

    filedati >> titolo1 >> titolo2 >> titolo3 >> titolo4 >> titolo5;

    while(filedati >> secondi >> Tin >>  Tout >> pressione >> run)
            {
            hTout->Fill(Tout);
            }

    TCanvas *c = new TCanvas();
    hTout->Draw();
}
```

Percorso file dati

Verifica esistenza del file

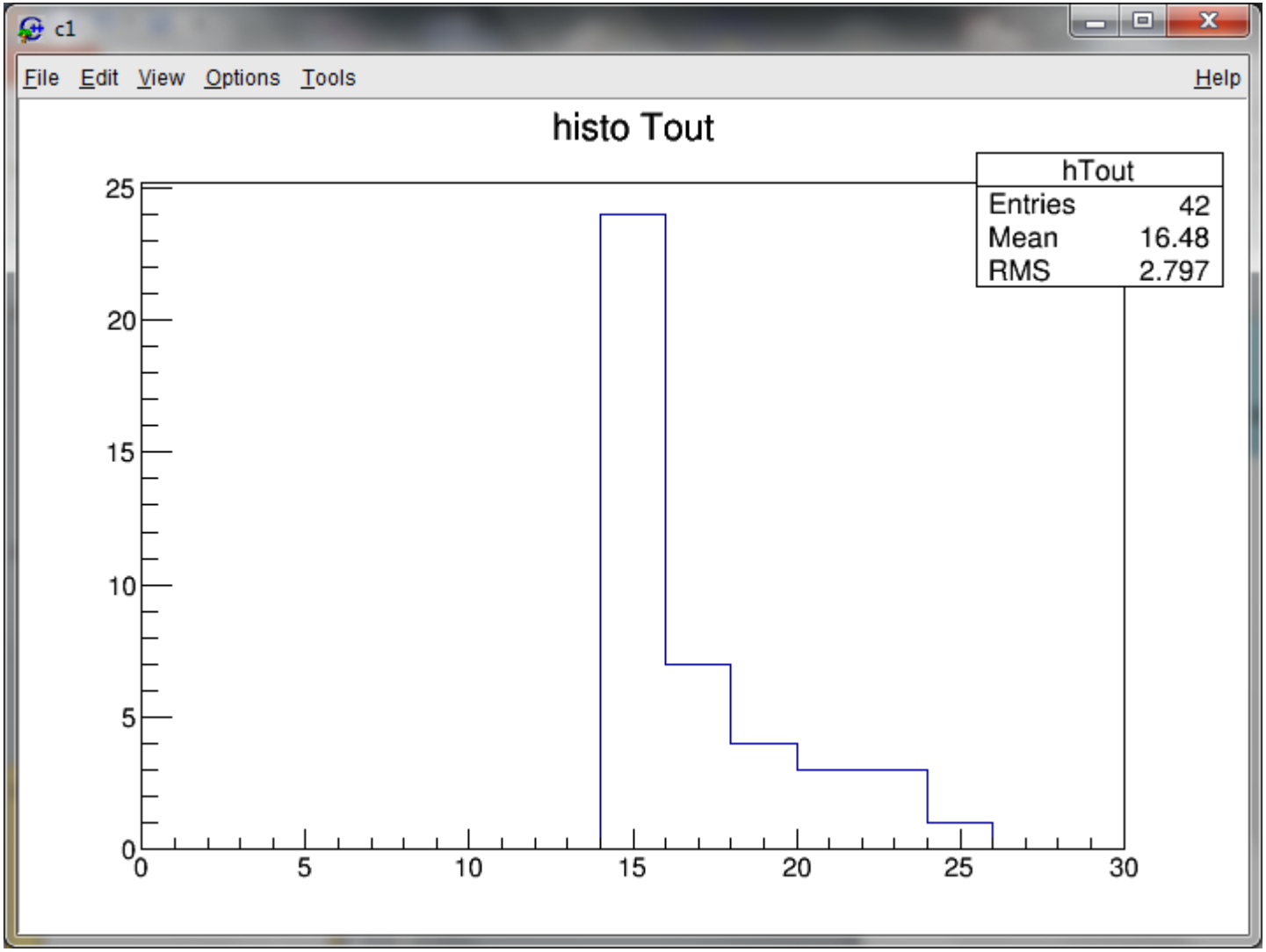
Dichiarazione variabili utilizzate per dati in lettura

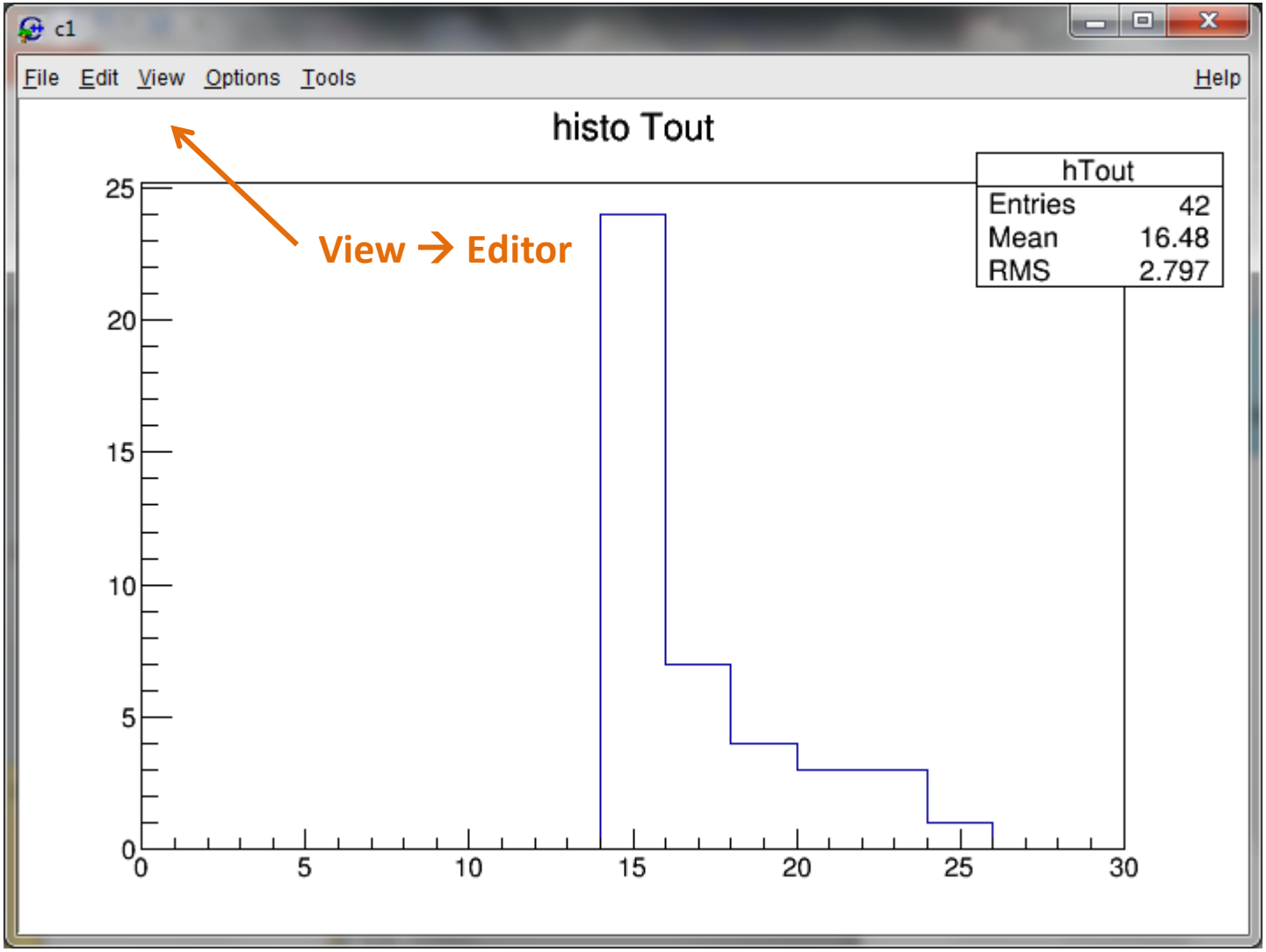
Dichiarazione histo

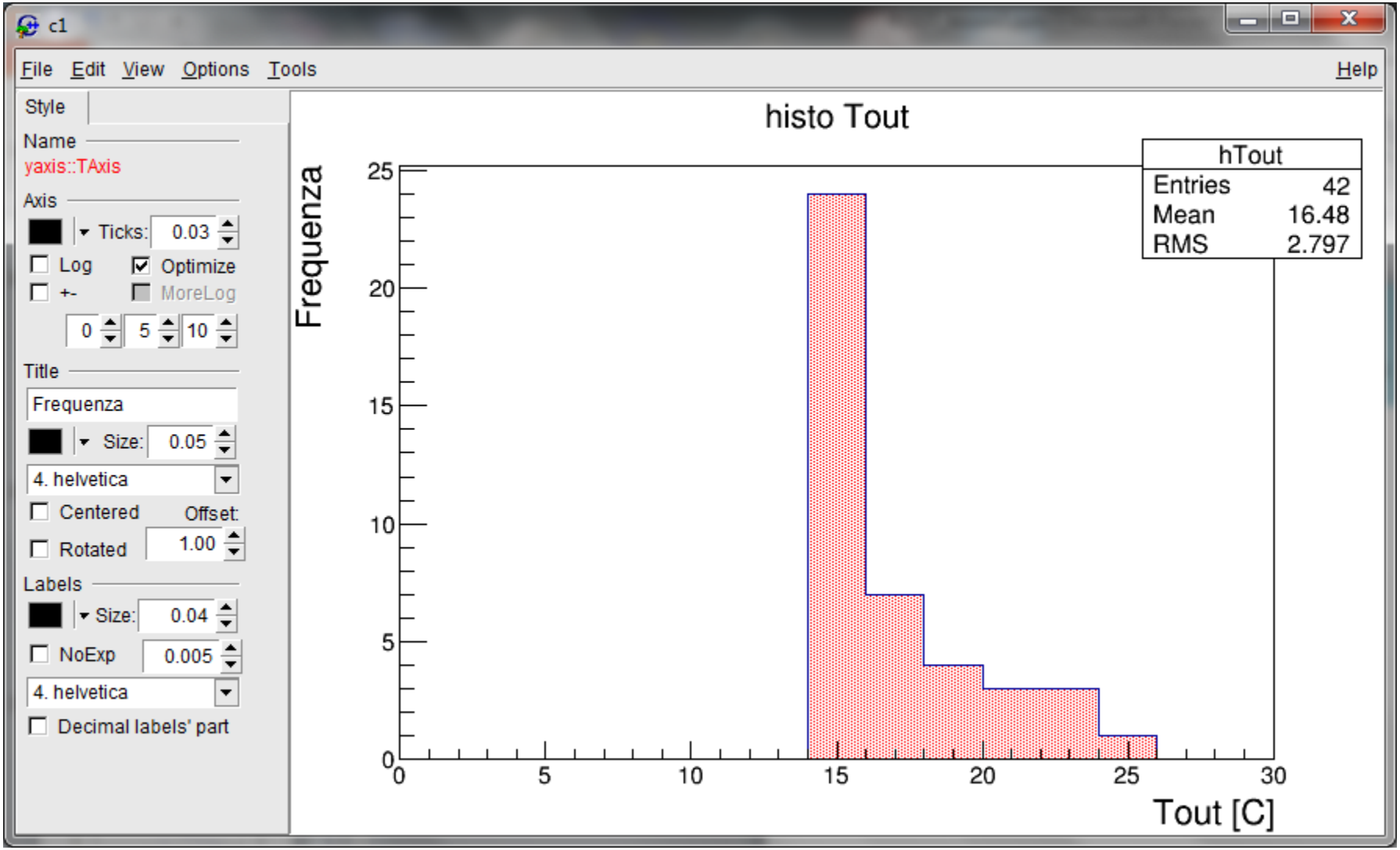
Lettura prima riga

Ciclo di lettura e riempimento histo

Finestra grafica e disegno dell'histo

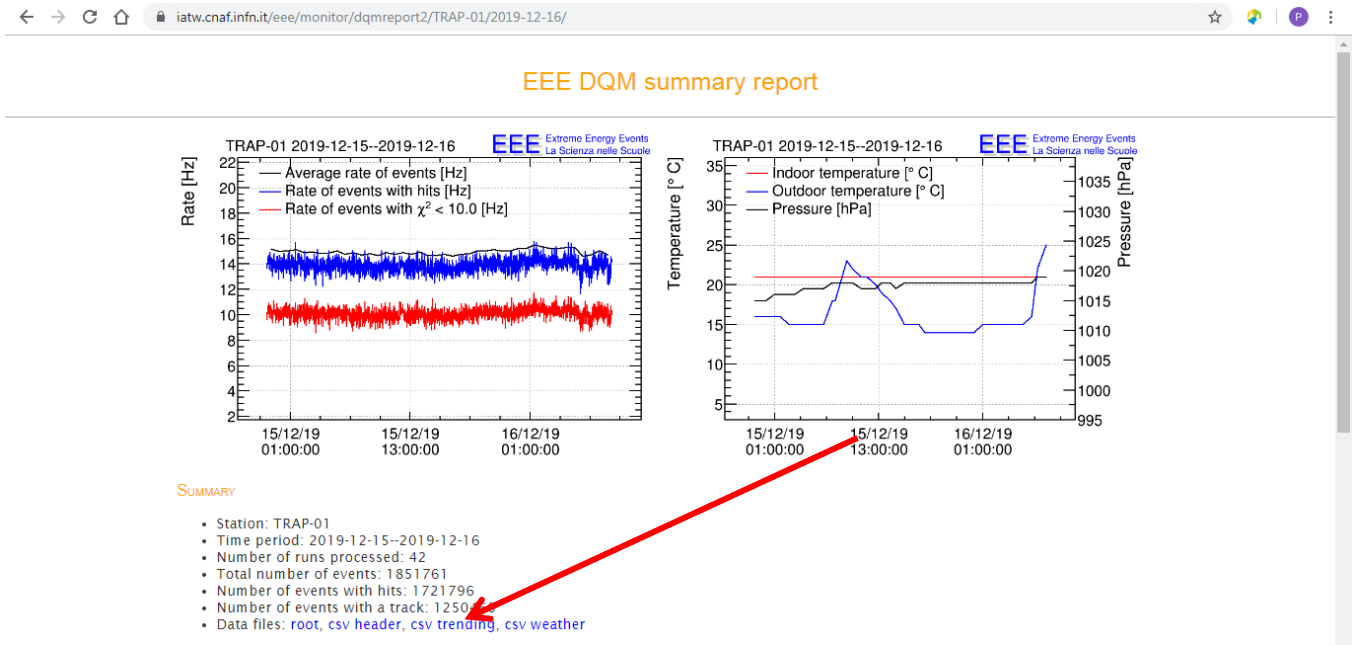

c1
File Edit View Options Tools Help
histo Tout
hTout
Entries 42
Mean 16.48
RMS 2.797
25
20
15
10
5
0
0 5 10 15 20 25 30


c1
File Edit View Options Tools Help
histo Tout
hTout
Entries 42
Mean 16.48
RMS 2.797
View → Editor
25
20
15
10
5
0
0 5 10 15 20 25 30


c1
File Edit View Options Tools Help
Style
Name
yaxis::TAxis
Axis
Ticks: 0.03
Log
Optimize
+-
MoreLog
0 5 10
Title
Frequenza
Size: 0.05
4. helvetica
Centered
Offset:
Rotated
1.00
Labels
Size: 0.04
NoExp
0.005
4. helvetica
Decimal labels' part
histo Tout
hTout
Entries 42
Mean 16.48
RMS 2.797
Frequenza
Tout [C]
0
5
10
15
20
25
30

PROPOSTA DI ESERCIZIO/ANALISI

1. Scrivere una macro per leggere i dati presenti nel file csv trending (22 variabili!!!!)



PROPOSTA DI ESERCIZIO/ANALISI

1. Scrivere una macro per leggere i dati presenti nel file csv trending (22 variabili!!!!)
2. Creare e disegnare un istogramma di una o più variabili

SALVATE LA MACRO IN UN FILE (all'interno della cartella macros) ED ESEGUITELA DAL TERMINALE DI ROOT (con il comando .x)

PER CHIARIMENTI E DUBBI RIVOLGERSI AL PROPRIO RESPOSABILE LOCALE

RUN COORDINATION MEETING – 19 Febbraio 2020