



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA





# Analisi Statistica dei dati nella Fisica Nucl. e Subnucl. [Laboratorio ]

**Gabriele Sirri**

Istituto Nazionale di Fisica Nucleare

**2015.04.16**

- Comunicazioni
-  Discussione (Esercizio 2)
- RooFit (segue...)
-  Home work (Esercizio 3)

# Comunicazioni

# Calendario

• Lunedì 23 febbraio 2015 14-16 M. Sioli

## MARZO

• Lunedì 2 marzo 2015 14-16 M. Sioli  
• Giovedì 5 marzo 2015 11-13 T. Chiarusi

• Lunedì 9 marzo 2015 14-16 M. Sioli  
• Giovedì 12 marzo 2015 11-13 M. Sioli

• Lunedì 16 marzo 2015 14-16 M. Sioli  
• Giovedì 19 marzo 2014 11-13 T. Chiarusi

• Lunedì 23 marzo 2015 14-16 M. Sioli  
• Giovedì 26 marzo 2015 11-13 M. Sioli  
• **Giovedì 26 marzo 2015 16-18 G. Sirri**

• Lunedì 30 marzo 2015 14-16 M. Sioli

## APRILE

• Mercol. 8 aprile 2015 10-13 M. Sioli/T.Chiar.  
• **Giovedì 9 aprile 2015 11-13 G. Sirri**

• Lunedì 13 aprile 2015 14-16 M. Sioli  
• Giovedì 16 aprile 2015 11-13 T. Chiarusi  
• **Giovedì 16 aprile 2015 16-18 G. Sirri**



• Lunedì 20 aprile 2015 14-16 M. Sioli  
• Giovedì 23 aprile 2015 11-13 T. Chiarusi  
• **Giovedì 23 aprile 2015 16-18 G. Sirri**

• **Giovedì 30 aprile 2015 11-13 G. Sirri**  
• **Giovedì 30 aprile 2015 16-18 G. Sirri**

## MAGGIO

• Lunedì 4 maggio 2015 14-16 M. Sioli  
• Giovedì 7 maggio 2015 11-13 T. Chiarusi

• Lunedì 11 maggio 2015 14-16 M. Sioli  
• **Giovedì 14 maggio 2015 11-13 G. Sirri**  
• **Giovedì 14 maggio 2015 16-18 G. Sirri**

• Lunedì 18 maggio 2015 14-16 M. Sioli  
• Giovedì 21 maggio 2015 11-13 T. Chiarusi  
• Lunedì 25 maggio 2015 14-16 M. Sioli

**Tutte le lezioni in Aula C, via Innerio**



# Esercitazioni (soluzioni)

	Lista Mail	ES. 1	ES. 2
1	sì	sì	sì
2	sì	sì	sì
3	sì	sì	sì
4	sì	sì	sì
5	sì	sì	sì
6	sì	sì	sì
7	sì	sì	sì
8		sì	
9	sì	sì	sì
10	sì	sì	sì
11	sì		
12		sì	sì
13	sì	sì	sì
14	sì	sì	sì
15	sì	sì	sì
16	sì	sì	sì

Le soluzioni di **Esercizio 1** sono pubblicate in AlmaCampus (seguite il link dal sito web)

accesso riservato agli iscritti a **gabriele.sirri2.ASD-2015**

## **Discussione (Esercizio 2)**



# RECAP... Esercizio 2

Scaricate da <http://www.unibo.it/docenti/gabriele.sirri2> : **roofit\_empty.C**

## [1] roofit\_ex1.C

Editate la macro e seguendo lo schema costruire una p.d.f. gaussiana con media 0, sigma 1. Modificate la sigma a 3. Visualizzate la p.d.f. . Generate un dataset unbinned di 10000 eventi. Eseguite un Fit con Maximum Likelihood. Visualizzate i risultati.

Utilizzate le informazioni in [Introduction to RooFit](ftp://root.cern.ch/root/doc/RooFit_Users_Manual_2.91-33.pdf) , nel manuale di roofit al paragrafo 2 ([ftp://root.cern.ch/root/doc/RooFit\\_Users\\_Manual\\_2.91-33.pdf](ftp://root.cern.ch/root/doc/RooFit_Users_Manual_2.91-33.pdf)) e in <http://root.cern.ch/drupal/content/roofit>).

## [2] roofit\_ex2.C

Si modifichi lo script e generare un dataset binned (bin width = 0.5) .

*The binning of the returned RooDataHist is controlled by the default binning associated with the observables generated. To set the number of bins in x to 200, do e.g. `x.setBins(200)` prior to the call to `generateBinned()`*



### [3] roofit\_ex3.C

Rinominate la p.d.f. gaussiana «sig» e aggiungete al modello un fondo esponenziale «bkg» espresso in funzione di un parametro tau,  $\exp(-x/\text{tau})$  .

Il valore iniziale di tau =10.

*Suggerimento: Si esprima  $-1./\text{tau}$  come RooFormulaVar*

Definite un parametro «fsig» rapporto segnale/fondo.

Costruite un modello composito nella forma

$$\text{model}(x) = \text{fsig} * \text{sig}(x) + (1 - \text{fsig}) * \text{bkg}(x)$$

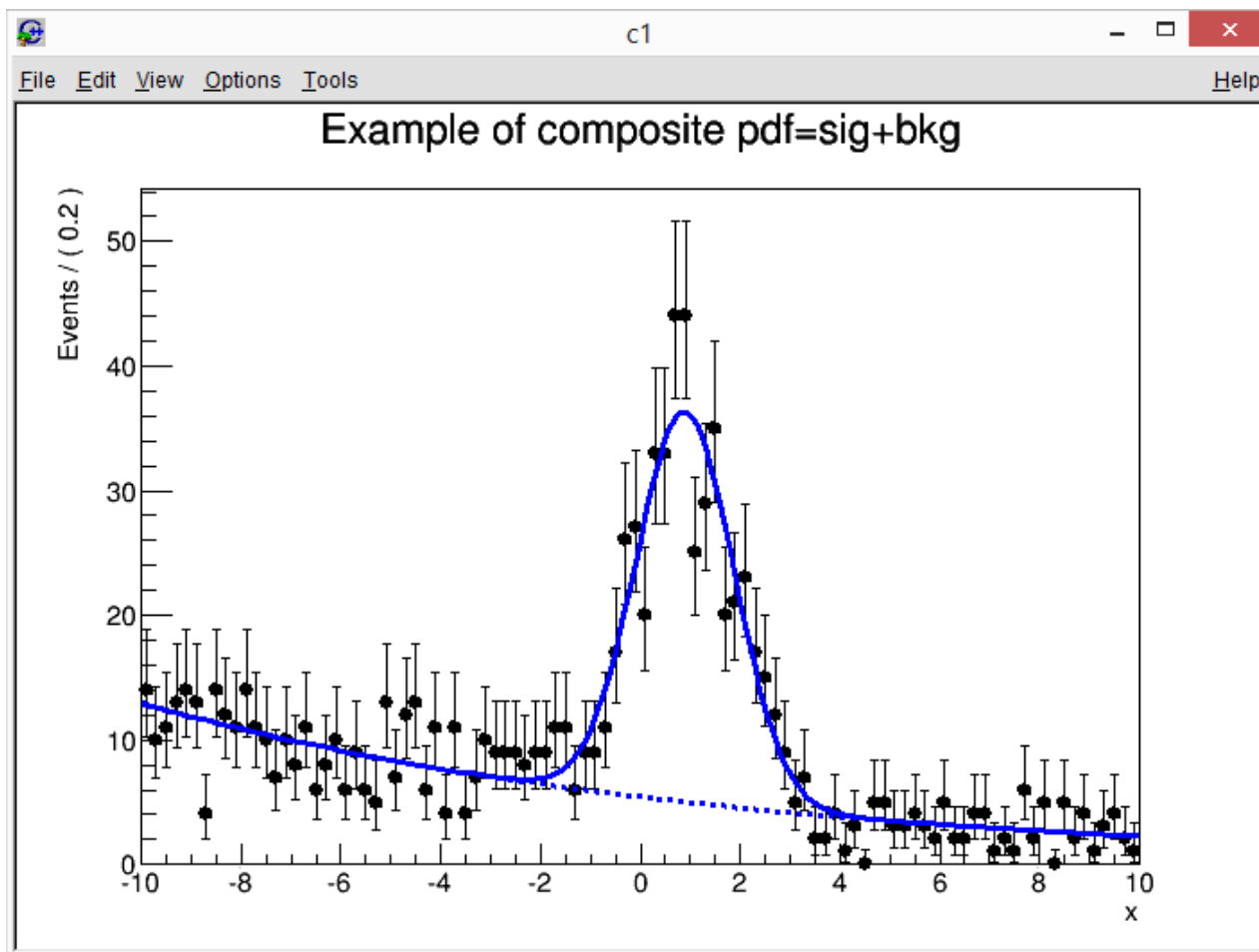
*Suggerimenti: usate la funzione RooAddPdf (paragrafo 3 del manuale)*





# Esercizio 2 – Soluzione Problema 3

[3] `roofit_ex3.C`

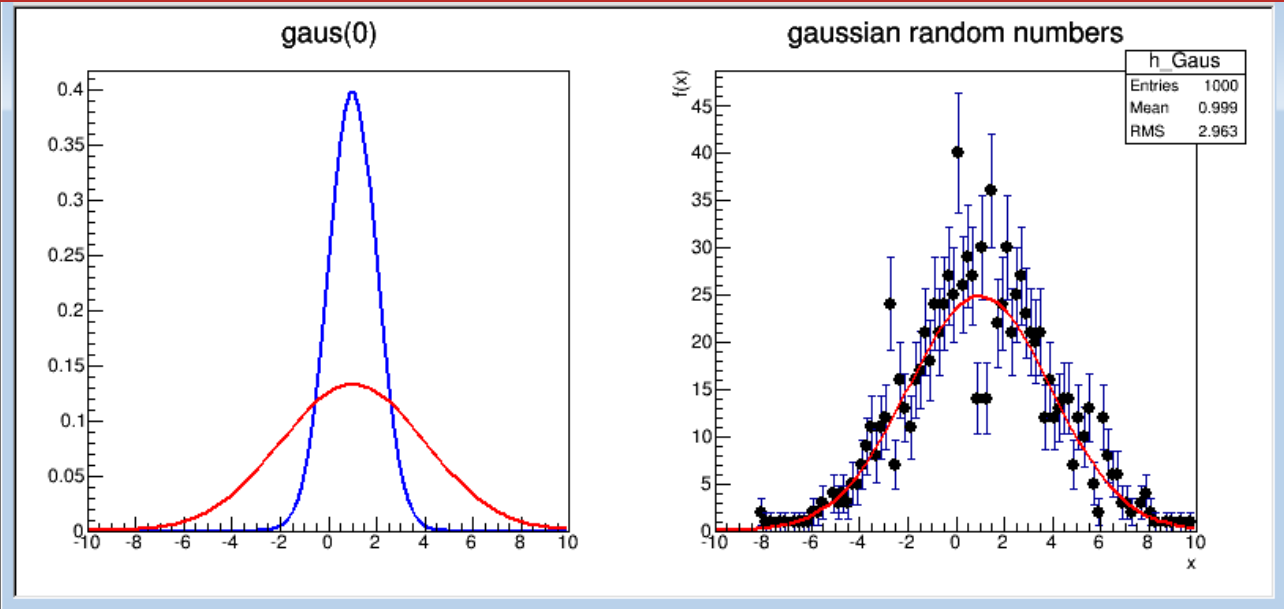


Guardare meglio `RooFormulaVar` e `RooAddPdf`

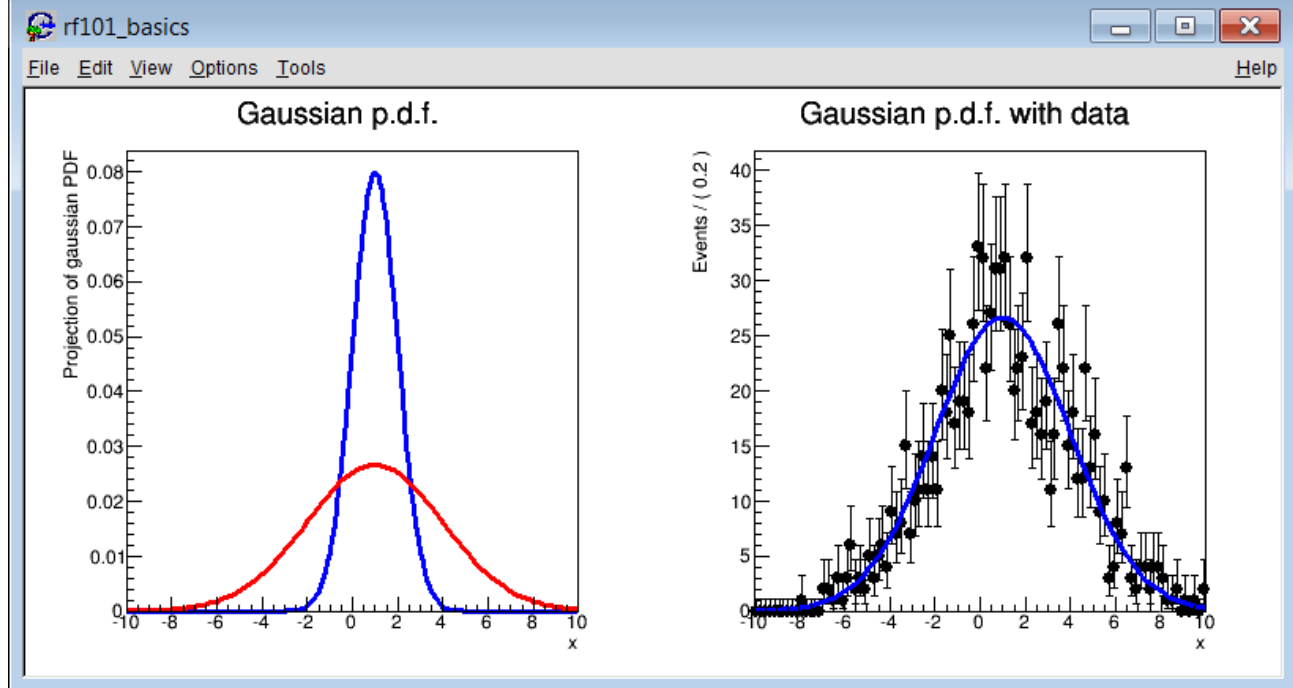


# Confronto con ROOT (1)

Questo era l'output dell'esercizio 4 della lezione-3, in cui non è stato utilizzato roofit.



roofit\_ex1





# Confronto con ROOT (2)

## tipo di dati fittati

**ROOT** il fit è stato fatto dopo aver istogrammato dati (**binned data**).

**ROOFIT** il fit è stato fatto su una distribuzione di dati simulati non binnata (**unbinned data**). Per alcune applicazioni, istogrammare i dati comporta una perdita di precisione.

## metodo di fit

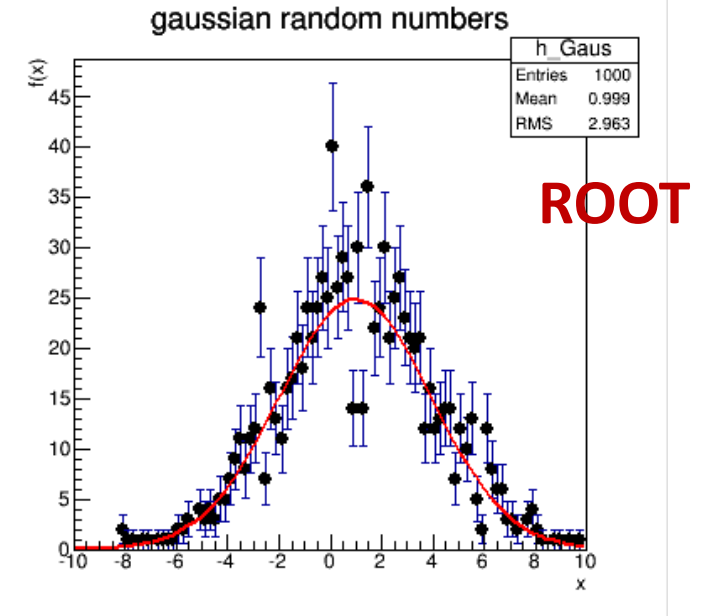
**ROOT** chi quadrato,  $\chi^2$

**ROOFIT** maximum likelihood,  $-\log \mathcal{L}$   
puo' essere fatto anche su dati non binnati.

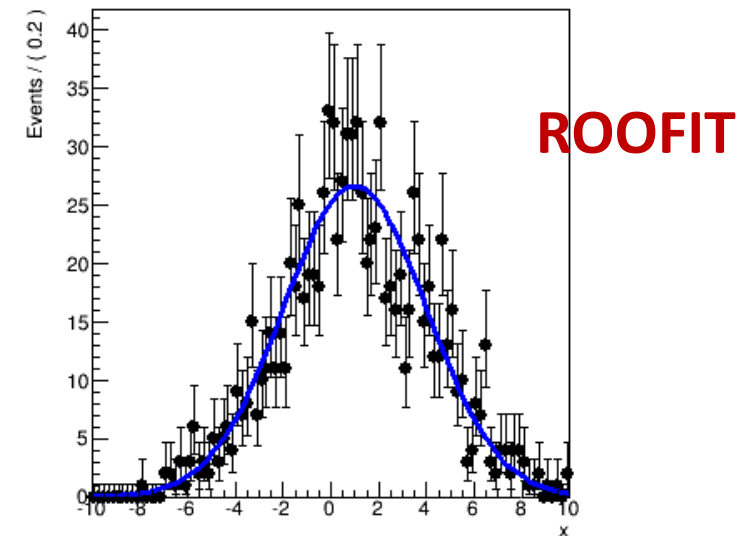
## barre di errore:

**ROOT** per default si assume l'approssimazione gaussiana (imprecisa quando N è piccolo).

**ROOFIT** sono correttamente calcolate per default utilizzando la statistica di Poisson (notate che per valori piccoli sono asimmetriche)



Gaussian p.d.f. with data



## ROOT

```
void recap_root()
{
// Book histograms
TH1D* h_Gaus =
    new TH1D("h_Gaus", "gaussian random numbers", 100, -10, 10);

// Create a TRandom3 object to generate random numbers
int seed = 12345;
TRandom3* ran = new TRandom3(seed);

// Generate some random numbers and fill histograms
const int numValues = 1000;

for (int i=0; i<numValues; i++){
    double r = ran->Gaus(1,3); // gaussian in mean = 1 sigma = 3
    h_Gaus->Fill(r);
}

c = new TCanvas("c1","c1",800,400) ;
c->Divide(2,1) ;

c->cd(1);
// define a gaussian p.d.f (mean 1 and sigma 1)
// gaus(0) is a substitute for [0]*exp(-0.5*((x-[1])/[2])**2)
// and (0) means start numbering parameters at 0.
TF1* g1 = new TF1( "gaus_1", "gaus(0)", -10, 10 );
double mean = 1 ;
double sigma = 1 ;
g1->SetParameter( 0, 1./( sigma * sqrt(2* 3.1415926 ) ) );
g1->SetParameter( 1, mean ) ; // set mean
g1->SetParameter( 2, sigma ) ; // set sigma
g1->SetLineColor( kBlue ) ;
g1->Draw() ;

// change sigma to 3
TF1* g2 = new TF1( "gaus_2", "gaus(0)", -10, 10 );
mean = 1 ;
sigma = 3 ;
g2->SetParameter( 0, 1./( sigma * sqrt(2* 3.1415926 ) ) );
g2->SetParameter( 1, mean ) ; // set mean
g2->SetParameter( 2, sigma ) ; // set sigma
g2->SetLineColor( kRed ) ;
g2->Draw("SAME") ;

c->cd(2);

h_Gaus->SetXTitle("x");
h_Gaus->SetYTitle("f(x)");
h_Gaus->SetMarkerStyle(20);
h_Gaus->Fit("gaus") ;
h_Gaus->Draw("E1");
}
```

## ROOFIT

```
void rooofit_ex1()
{
// Setup model
// =====
// Declare variables x,mean,sigma with associated name, title, initial
value and allowed range
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",1,0.1,10) ;
// Build gaussian p.d.f in terms of x,mean and sigma
RooGaussian gauss("gauss","gaussian PDF",x,mean,sigma) ;
// Construct plot frame in 'x'
RooPlot* xframe = x.frame(Title("Gaussian p.d.f.)) ;

// Plot model and change parameter values
// =====
// Plot gauss in frame (i.e. in x)
gauss.plotOn(xframe) ;
// Change the value of sigma to 3
sigma.setVal(3) ;
// Plot gauss in frame (i.e. in x) and draw frame on canvas
gauss.plotOn(xframe,LineColor(kRed)) ;

// Generate events
// =====
// Generate a dataset of 1000 events in x from gauss
RooDataSet* data = gauss.generate(x,1000) ;
// Make a second plot frame in x and draw both the
// data and the p.d.f in the frame
RooPlot* xframe2 = x.frame(Title("Gaussian p.d.f. with data")) ;
data->plotOn(xframe2) ;
gauss.plotOn(xframe2) ;

// Fit model to data
// =====
// Fit pdf to data
gauss.fitTo(*data) ;
// Print values of mean and sigma (that now reflect fitted values and
errors)
mean.Print() ;
sigma.Print() ;
// Draw all frames on a canvas
TCanvas* c = new TCanvas("rf101_basics","rf101_basics",800,400) ;
c->Divide(2) ;
c->cd(1) ;
gPad->SetLeftMargin(0.15) ;
xframe->GetYaxis()->SetTitleOffset(1.6) ;
xframe->Draw() ;
c->cd(2) ;
gPad->SetLeftMargin(0.15) ;
xframe2->GetYaxis()->SetTitleOffset(1.6) ;
xframe2->Draw() ;
}
```

# RooFIT (segue...)

# RooFit

[Introduction to RooFit](#) slides da 15(19) a 34

<http://hadron.physics.fsu.edu/~skpark/document/ROOT/roofit-intro-roostats-v12a.pdf>

# Home work (Esercizio 3)



# Esercizio 3 – TESTO (1)

## [1] roofit\_factory\_ex1.C

*Riprendiamo l'ex3.C della lezione scorsa e riformuliamolo utilizzando workspace e factory*

- make an exponential background + gaussian signal model using `w.factory("SUM::pdf( fs*sig, bkg)")`  
do first non-extended model (  $0 < fs < 1$  )  
the exponential background depends on the parameter tau as  $\exp(-x/\text{tau})$
- generate data (e.g. N=1000)
- set constant mean and sigma while fs and tau are floating  
`w.var("mean")->setConstant(true)`
- fit the data
- plot results  
(use `RooFit::Components()` to specify components to plot)

*Utilizzate le informazioni in [Introduction to RooFit](http://root.cern.ch/drupal/content/roofit), in <http://root.cern.ch/drupal/content/roofit> e nel manuale di roofit ([ftp://root.cern.ch/root/doc/RooFit\\_Users\\_Manual\\_2.91-33.pdf](ftp://root.cern.ch/root/doc/RooFit_Users_Manual_2.91-33.pdf))*





## [2] roofit\_factory\_ex2.C

*Modifichiamo l'esercizio precedente*

make extended model using  $N_s$  and  $N_b$  ( $N_s = f_s * N$ )

```
w.factory("SUM::pdf( N_s*gaus, N_b*expo)")
```

check difference in error in  $N_s$  obtained between extended and not-extended fit

save the macro making the extended model and also save the workspace in a file  
(we will be using it later in a RooStats exercise)