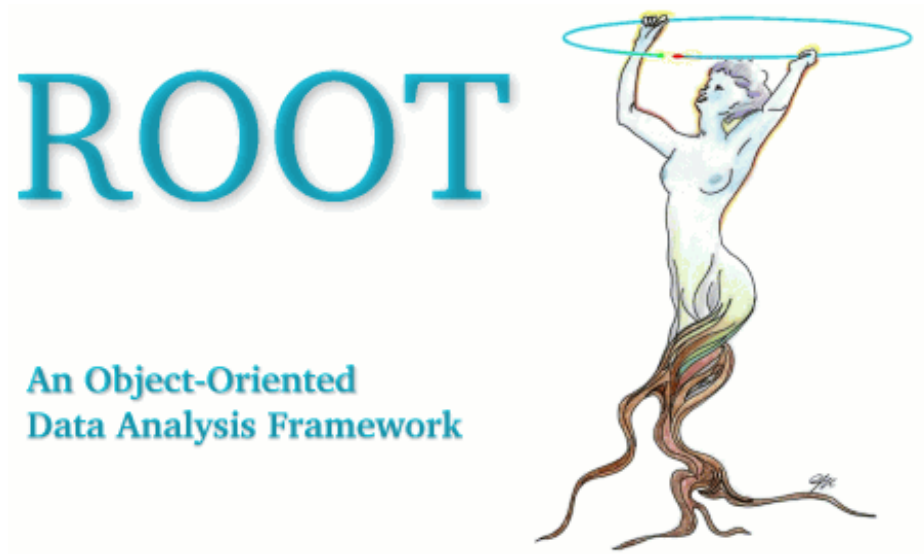


ROOT 3



<http://root.cern.ch/>

Συναρτήσεις (Functions)

Συνάρτηση είναι ένα TF1 αντικείμενο το οποίο ορίζεται σε ένα διάστημα (x_{\min} , x_{\max}).

Μια συνάρτηση μπορεί να είναι :

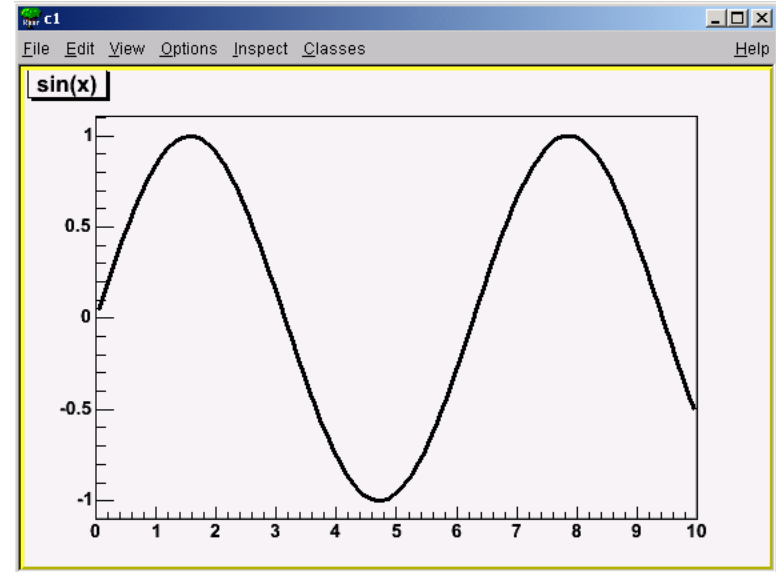
- Έκφραση με τη μεταβλητή x χωρίς παραμέτρους.
- Έκφραση με τη μεταβλητή x με παραμέτρους.
- Μια γενικά συνάρτηση C με παραμέτρους.

Συναρτήσεις (Functions)

Συναρτήσεις μιας διάστασης (1D)

```
fun1 = new TF1("fun1", "sin(x)", 0, 10)  
fun1->Draw()
```

Με το ποντίκι *DrawPanel*



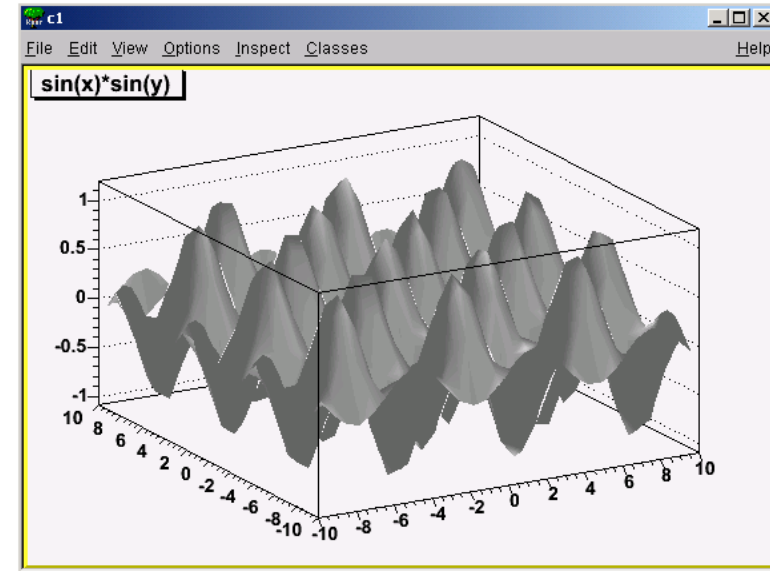
```
TF1 *f1 = new TF1("f1", "sin(x)/x", 0, 10);  
f1->Draw();  
TF1 *f2 = new TF1("f2", "f1 * 2", 0, 10);
```

Συναρτήσεις (Functions)

Συναρτήσεις δύο διαστάσεων (2D)
(π.χ. $\sin(x)*\sin(y)$)

```
fun2 = new TF2("fun2","sin(x)*sin(y)",-0.0,  
10.0,-10.0,10.0);  
fun2->Draw("surf4");
```

Στην Draw("OPTION") υποστηρίζονται οι γνωστές επιλογές:
"LEGO", "LEGO1" "LEGO2"
"SURF" "SURF1" "SURF2" "SURF3" "SURF4"



Συναρτήσεις (με παραμέτρους)

Συνάρτηση με παραμέτρους

Η εντολή δημιουργίας της συνάρτησης καλείται με πίνακα των παραμέτρων [0],[1],[2],.....

```
TF1 *f1 = new TF1 ("f1","[0] *x*sin( [1] *x)",-3,3);
```

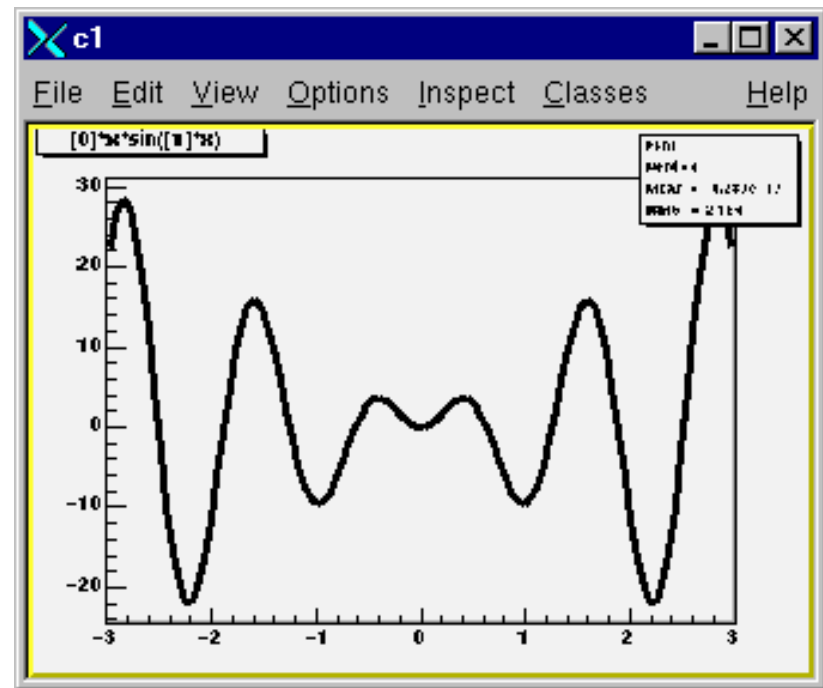
Οι τιμές των παραμέτρων δίνονται με την εντολή :

```
f1->SetParameter(0,10);
```

Σημαίνει [0]=0, [1]=10

```
f1->SetParameter(1,5);
```

```
f1->Draw();
```



Συναρτήσεις (με παραμέτρους)

Επιτρεπτές εκφράσεις

- $\sin(x)/x$
- $[0]*\sin(x) + [1]*\exp(-[2]*x)$
- $x + y**2$
- $x^2 + y^2$
- $[0]*\text{pow}([1],4)$
- $2*\text{pi}*\text{sqrt}(x/y)$
- $\text{gaus}(0)*\text{expo}(3) + \text{ypol3}(5)*x$

$\text{gaus}(0)$ παριστάνει τη συνάρτηση $[0]*\exp(-0.5*((x-[1])/[2])**2)$ και το (0) σημαίνει ότι η αρίθμηση των παραμέτρων αρχίζουν από το 0

- $\text{gausn}(0)*\text{expo}(3) + \text{ypol3}(5)*x$

$\text{gausn}(0)$ παριστάνει τη συνάρτηση $[0]*\exp(-0.5*((x[1])/[2])**2)/(\text{sqrt}(2*\text{pi})*[2])$ και το (0) σημαίνει ότι η αρίθμηση των παραμέτρων αρχίζουν από το 0

$\text{expo}(3)$ παριστάνει τη συνάρτηση $\exp([3]+[4]*x)$

$\text{pol3}(5)$ παριστάνει τη συνάρτηση $\text{par}[5]+\text{par}[6]*x+\text{par}[7]*x**2+\text{par}[8]*x**3$

Επίσης οι συναρτήσεις [TMath](#) μπορούν να είναι μέρος μιας έκφρασης

- $\text{TMath}::\text{Landau}(x)*\sin(x)$
- $\text{TMath}::\text{Erf}(x)$

Συναρτήσεις (με ορισμό)

Χρησιμοποιώντας μια “ορισμένη” (με ορισμό) συνάρτηση

```
Double_t MyFunction(Double_t *x, Double_t *par){  
    Float_t    xx=x[0];  
    Double_t    val=TMath::Abs(par[0]*sin(par[1]*xx)/xx);  
    return val;  
}
```

par[] είναι ο πίνακας με τις παραμέτρους

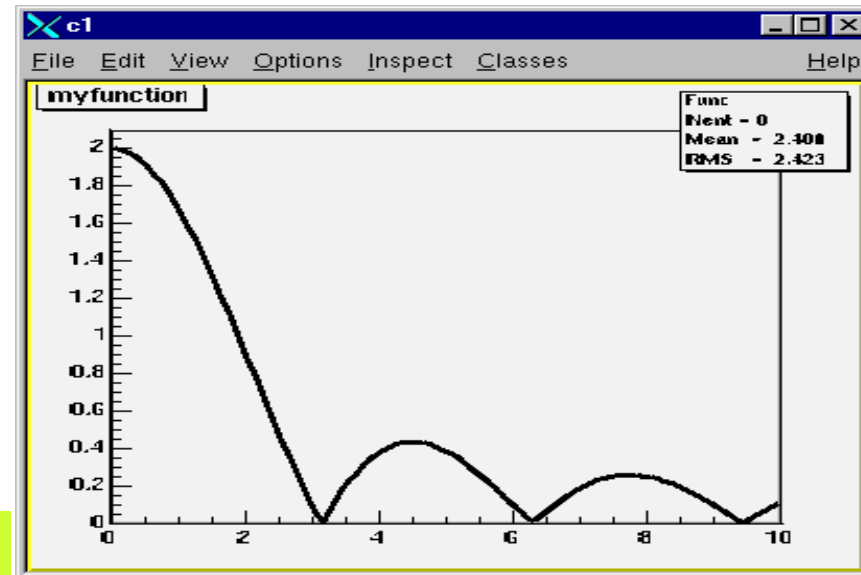
x[] είναι ο πίνακας του των τιμών του x

```
TF1 *f1 = new F1("f1",MyFunction,0,10,2);
```

2 είναι ο αριθμός των παραμέτρων

```
f1->SetParameters(2,1);
```

```
f1->Draw();
```



fun_par.c

Προσαρμογή (*Fitting*)

Προσαρμογή ενός **histogram** ή ενός **graph**

```
hx->Fit("gaus");           // fit histogram  
graph->Fit("pol5");        // fit graph
```

gaus, expo, poln ($n=1,2,3,\dots$) Συναρτήσεις προ-ορισμένες (Predefined)

Προσαρμογή (*Fitting*)

Προσαρμογή στο ROOT

Βασίζεται στην τάξη TMinuit.

Η τάξη TMinuit χρησιμοποιεί την συνάρτηση προσαρμογής
πολλαπλών παραμέτρων FCN.

Η τιμή της FCN εξαρτάτε γενικά από μια ή περισσότερες
μεταβλητές.

Προσαρμογή (Fitting)

Προσαρμογή με συνάρτηση ορισμένη από το χρήστη

Η συνάρτηση ορίζεται με ένα από τους γνωστούς τρόπους.

Αρχικές τιμές στις παραμέτρους

Προσαρμογή

Ορισμός μιας συνάρτησης "myfit" με 3 παραμέτρους στο διάστημα [2,3]

```
TF1 *myfit = new TF1("myfit", "[0]*sin(x)+[1]*exp(-[2]*x)", 0, 2, 3);
```

Μπορούν να δοθούν ονόματα στις παραμέτρους (προαιρετικά)

Και αρχικές τιμές των παραμέτρων (υποχρεωτικά):

```
myfit->SetParName(0, "c0");  
myfit->SetParName(1, "c1");  
myfit->SetParName(2, "slope");
```

```
myfit->SetParameter(0, 1);  
myfit->SetParameter(1, 0.05);  
myfit->SetParameter(2, 0.2);
```

Και προσαρμογή

```
hist->Fit("myfit");
```



Ισοδύναμα

```
myfit->SetParameters(1, 0.05, 0.2);
```

Προσαρμογή (Fitting)

```
//_____ fitexample.C _____  
Double_t fitf(Double_t *x, Double_t *par)  
{  
  Double_t arg = 0;  
  if (par[2]) arg = (x[0] - par[1])/par[2];  
  Double_t fitval = par[0]*TMath::Exp(-0.5*arg*arg);  
  return fitval;  
}  
void fitexample()  
{  
  TFile *f = new TFile("hsimple.root");  
  TH1F *hpx = (TH1F*)f->Get("hpx");  
  TF1 *func = new TF1("fit",fitf,-3,3,3);  
  func->SetParameters(500,hpx->GetMean(),  
  hpx->GetRMS());  
  func->SetParNames("Constant","Mean_value","Sigma");  
  hpx->Fit("fit");  
}
```

Προσαρμογή υποπεριοχής

Προσαρμογή μιας υπο-περιοχής

TF1::Fit() η προσαρμογή γίνεται στο διάστημα ορισμού του ιστογράμματος.

Ορίζοντας την επιλογή “r” σαν δεύτερη παράμετρο της Fit() μπορεί να περιοριστεί η περιοχή στην περιοχή ορισμού της συνάρτησης TF1.

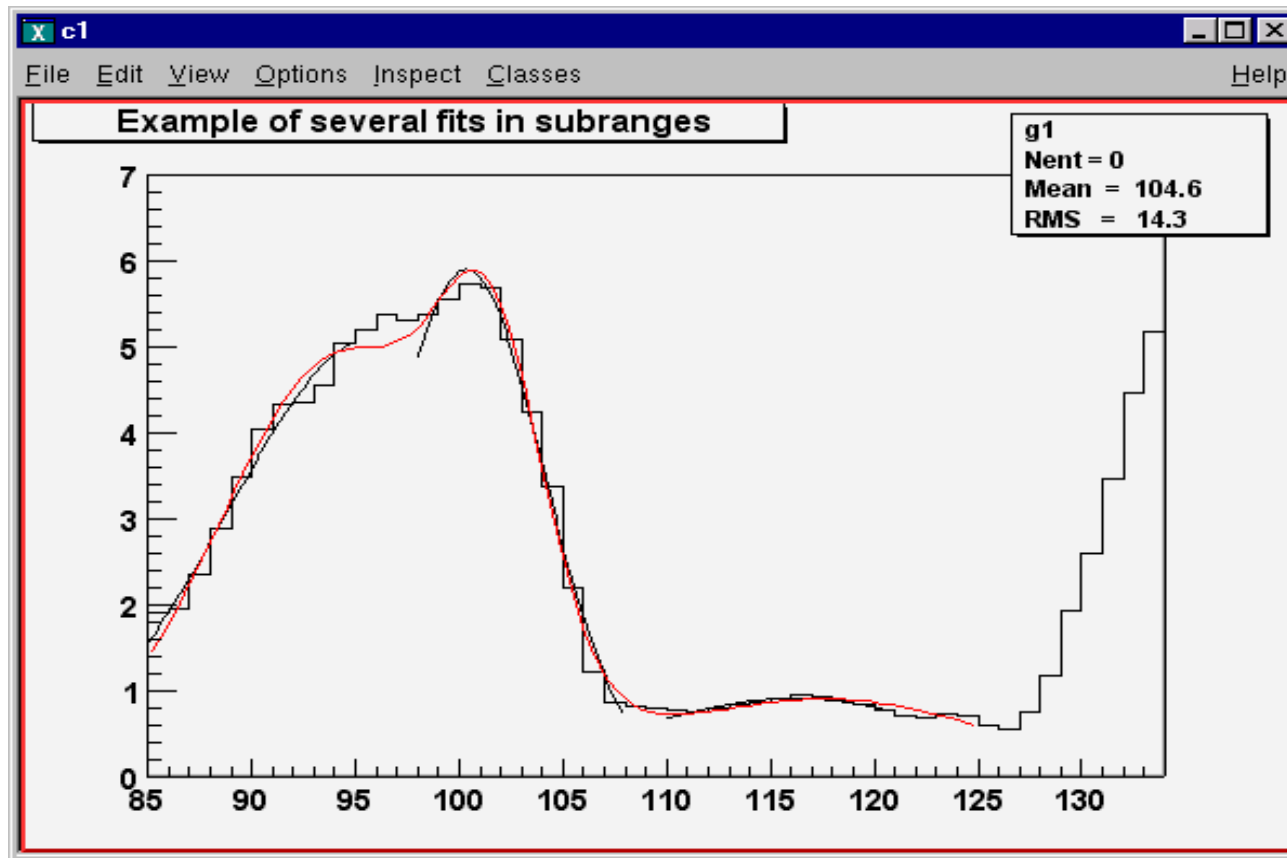
[fitrange.c](#)

Αντικαταστήστε την
`s1->Fit("gaus");`

```
TF1 *f1 = new TF1("f1", "gaus", 1, 3);  
histo->Fit("f1", "R");
```

[signbgnd.c](#)

Fitting Subranges



Fitting Sub-Ranges Example

\$ROOTSYS/tutorials/multifit.C

- Define the range in the TF1 constructor.

```
TF1 *g1 = new TF1("g1", "gaus", 85, 95);
```

- By default, TH1::Fit on the defined histogram range. Use "R" option in the Fit() method.

```
h->Fit("g1", "R");
```

Fitting Subranges++

Define gaussian functions

```
g1      = new TF1 ("m1", "gaus", 85, 95) ;
g2      = new TF1 ("m2", "gaus", 98, 108) ;
g3      = new TF1 ("m3", "gaus", 110, 121) ;
total = new
TF1 ("mstotal", "gaus (0) +gaus (3) +gaus (6) ", 85, 125) ;
```

From TFormula:

gaus(0) is a substitute for : $[0] \cdot \exp(-0.5 \cdot ((x-[1])/[2])^2)$

and (0) means start numbering parameters at 0

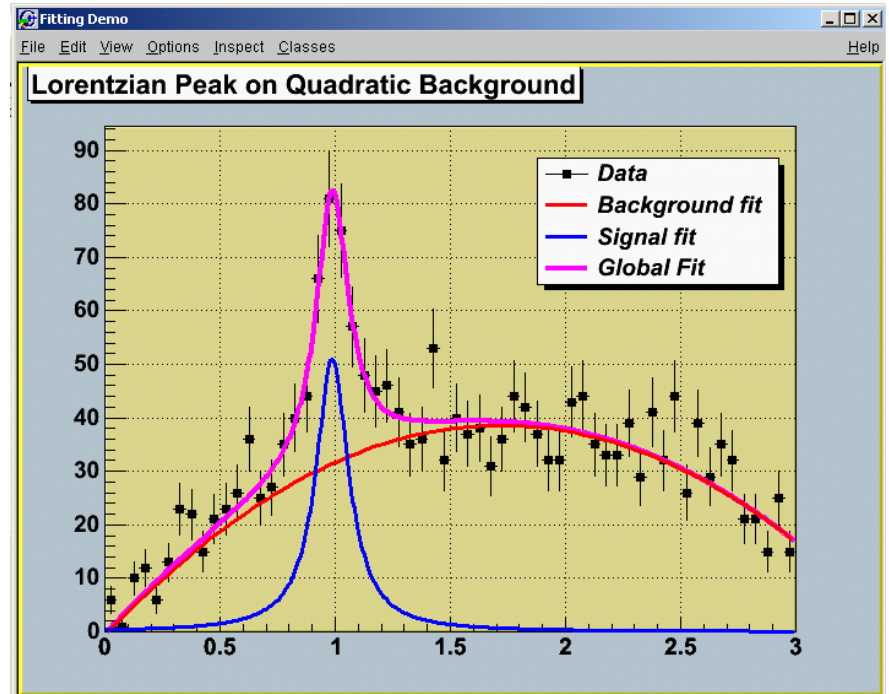
Fitting Sub-Ranges++

Fit each range and get the parameter for "total"

```
h->Fit(g1, "R");  
h->Fit(g2, "R+");  
h->Fit(g3, "R+");  
g1->GetParameters(&par[0]);  
g2->GetParameters(&par[3]);  
g3->GetParameters(&par[6]);  
total->SetParameters(par);  
h->Fit(total, "R+");
```


Fitting Demo

- FittingDemo.C



Signal and Background Demo

$$y(E) = a_1 + a_2 E + a_3 E^2 + A_p (\Gamma / 2 \pi) / ((E - \mu)^2 + (\Gamma/2)^2)$$

background

$$\text{par}[0] = a_1$$

$$\text{par}[1] = a_2$$

$$\text{par}[2] = a_3$$

lorenzianPeak

$$\text{par}[0] = A_p$$

$$\text{par}[1] = \Gamma$$

$$\text{par}[2] = \mu$$

$$\text{fitFunction} = \text{background}(x, \text{par}) + \text{lorenzianPeak}(x, \&\text{par}[3])$$

$$\text{par}[0] = a_1$$

$$\text{par}[1] = a_2$$

$$\text{par}[2] = a_3$$

$$\text{par}[3] = A_p$$

$$\text{par}[4] = \Gamma$$

$$\text{par}[5] = \mu$$

Random Generator

Η μεταβλητή **gRandom** είναι ένας δείκτης στη γεννήτρια τυχαίων αριθμών. Συνήθως δείχνει σε ένα αντικείμενο TRandom.

Θέτοντας το «σπόρο» 0 συνεπάγεται ότι ο σπόρος θα γεννηθεί από το χρόνο. Κάθε άλλη τιμή θα χρησιμοποιηθεί σαν σταθερά.

Βασικές τυχαίες κατανομές

Gaus(mean,sigma)
Rndm()
Landau(mean,sigma)
Poisson(mean)
Binomial(ntot,prob)
BreitWigner(mean, gamma)
Exp(tau)
Integer(max)
Rannor(a,b)
Uniform(min, max)

```
x= gRandom->Landau(2.5,0.8)
```