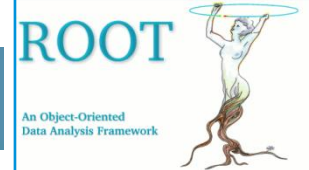


Προσαρμογές Δεδομένων

- Προσαρμογές δεδομένων ιστογράμματος από το Fit Panel.
- Προσαρμογές με χρήση της μεθόδου Fit.
 - Προσαρμογή σε Γκαουσιανή
- Προσαρμογές δεδομένων γραφήματος



Προσαρμογές δεδομένων ιστογράμματος από το Fit Panel



- Οι προσαρμογές δεδομένων είναι ένα πολύ σημαντικό κεφάλαιο στην ανάλυση των πειραματικών δεδομένων.
- Το ROOT διαθέτει έναν εξαιρετικό τρόπο προσαρμογής ιστογραμμάτων-γραφημάτων βασισμένο στην χρήση του προγράμματος Minuit.
- Το **Minuit** αναπτύχθηκε την δεκαετία του '70 στο CERN από τον Fred James αρχικά σε Fortran. Το πρόγραμμα υπολογίζει τα ελάχιστα μιας συνάρτησης (την οποία ορίζει ο χρήστης) σε σχέση με μία ή περισσότερες παραμέτρους.
- Σε απλές περιπτώσεις μπορούμε να κάνουμε εύκολα προσαρμογή των δεδομένων ενός ιστογράμματος (σε κάποια γνωστή κατανομή) χρησιμοποιώντας απλά το Fit Panel.

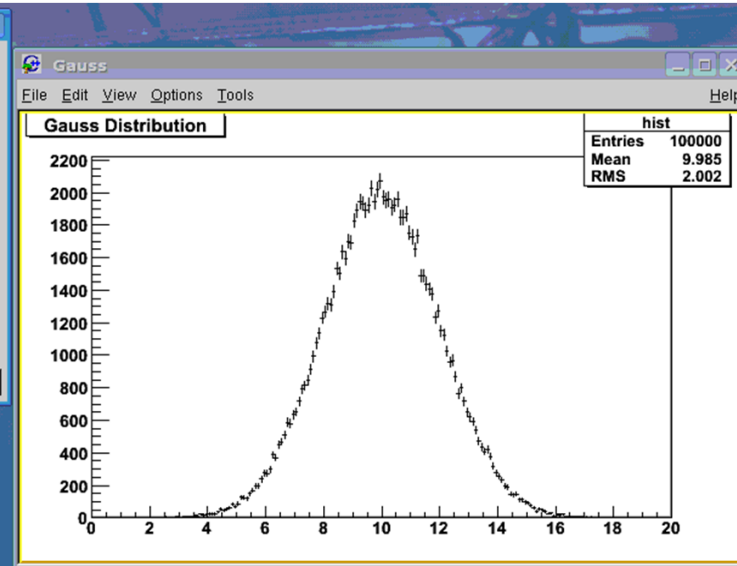
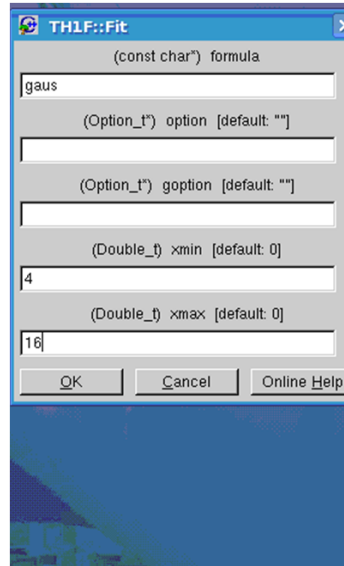
- Ας πάρουμε για παράδειγμα ένα ιστόγραμμα του οποίου τα δεδομένα ακολουθούν κατανομή Gauss. (Παράδειγμα στα αριστερά).
- Εκτελούμε το πρόγραμμα.
- Στη συνέχεια με το ποντίκι επιλέγουμε το ιστόγραμμα, κάνουμε δεξί κλικ και επιλέγουμε το Fit από την αναδυόμενη λίστα επιλογών.
- Συμπληρώνουμε την αναδυόμενη λίστα επιλογών όπως δίπλα και πατάμε ok.

```
{
  delete gRandom;
  gRandom = new TRandom(123456);

  TH1F *hist = new TH1F("hist", "Gauss Distribution", 200, 0, 20);

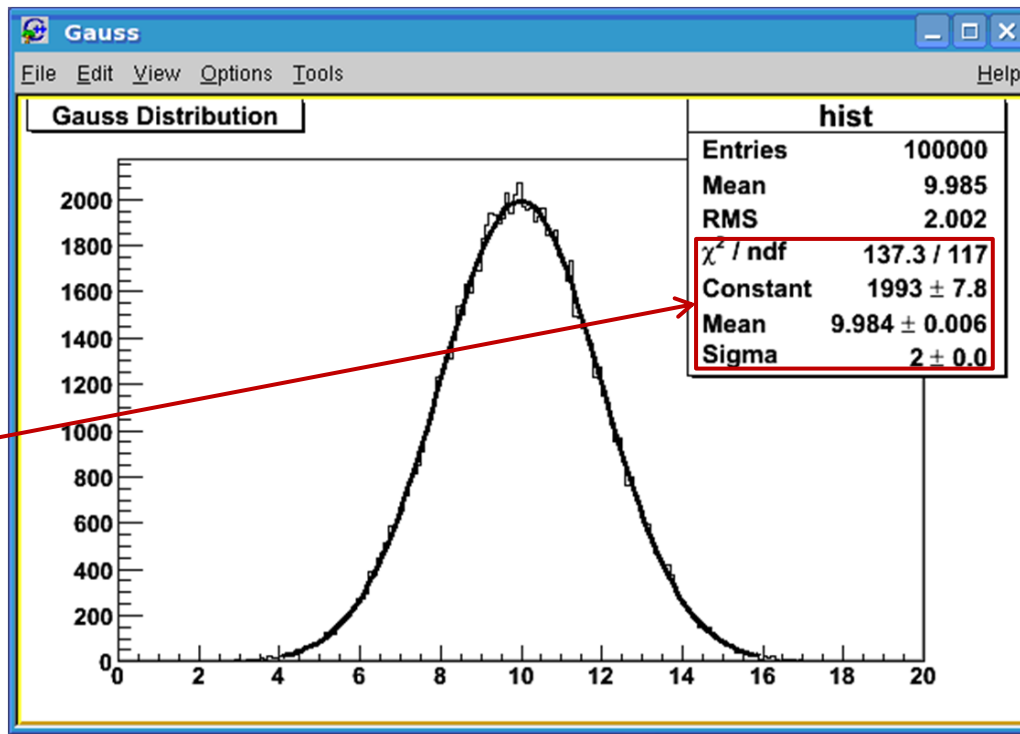
  for(int i=0; i<100000; ++i){
    double a=gRandom->Gaus(10, 2);
    hist->Fill(a);
  }

  cl=new TCanvas("c1", "Gauss", 600, 400);
  cl->cd();
  hist->Draw("e");
}
```



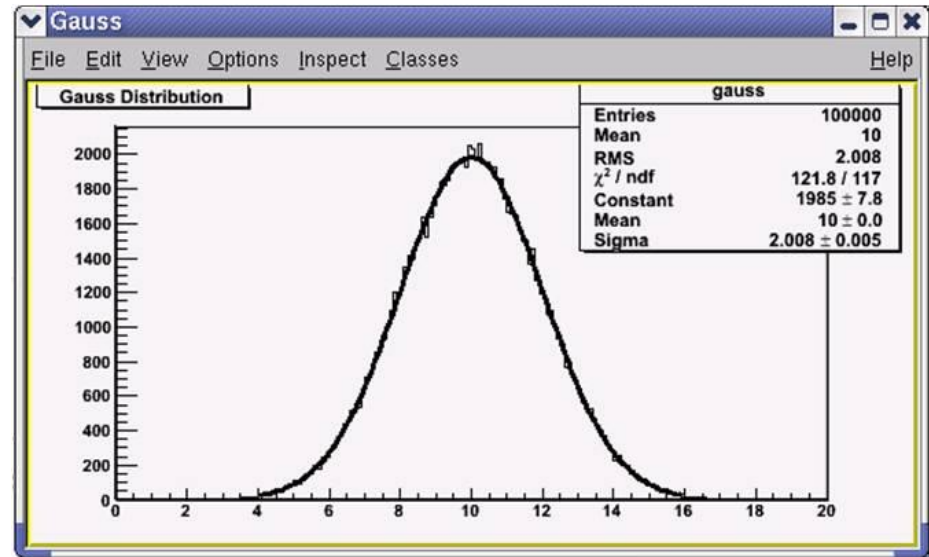
- Η προσαρμογή των δεδομένων σε κατανομή Gauss θα εμφανιστεί όπως δίπλα με τα αποτελέσματα της.

Τα αποτελέσματα της προσαρμογής



- Οι βασικές προκαθορισμένες συναρτήσεις είναι οι ακόλουθες
 - gaus: Συνάρτηση Gauss με 3 παραμέτρους (ύψος, μέσος και σίγμα)
 - expo: Εκθετική συνάρτηση με 2 παραμέτρους: $f(x) = \exp(p_0 + p_1 * x)$
 - polN: πολυώνυμο N^{ου} βαθμού: $f(x) = p_0 + p_1 * x + p_2 * x^2 + \dots$
 - landau: Συνάρτηση Landau με τον μέσο και το σίγμα.

- Στο παρακάτω παράδειγμα χρησιμοποιεί την μέθοδο **Fit** για να προσαρμόσει τα δεδομένα στη συνάρτηση **f1**.
- Η συνάρτηση **f1** ορίζεται ως η συνάρτηση **Gauss**.

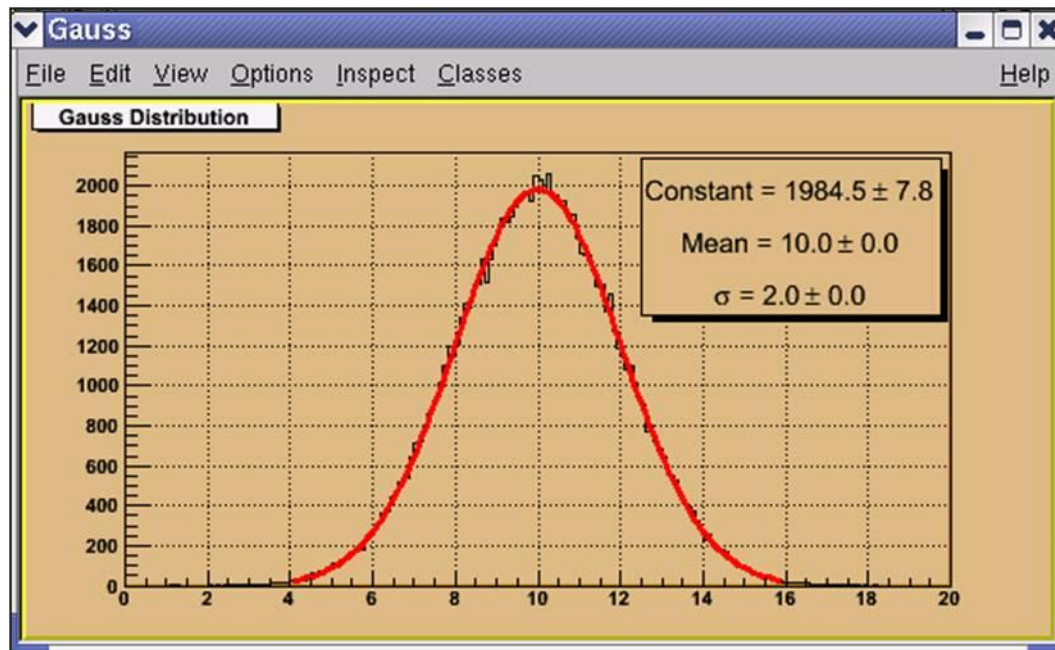


```
{
  delete gRandom;
  gRandom = new TRandom(123456); // Setting the seed
  TH1F *gauss = new TH1F("gauss", "Gauss Distribution", 200, 0, 20);
  for(int i=0; i<100000; ++i){
    double a=gRandom->Gaus(10, 2);
    gauss->Fill(a);
  }

  TF1 *f1 = new TF1("f1", "gaus", 0, 20); // Orismos tis synartisis f1
  gStyle->SetOptFit(111); // Ektypwsi toy apotelesmatos tis prosarmoghhs

  c1=new TCanvas("c1", "Gauss", 700, 400);
  c1->cd();
  gauss->Fit(f1, "", "", 4, 16); // Prosarmogh istogrammatos stin f1 apo to 4 ews to 16
}
```

- Στο παρακάτω παράδειγμα κάνουμε την ίδια προσαρμογή όπως στο προηγούμενο παράδειγμα ενώ για την εκτύπωση του αποτελέσματος χρησιμοποιούμε κάποιες από τις πολλές δυνατότητες του ROOT.



- Προσέξτε τον τρόπο παρουσίασης των αποτελεσμάτων της προσαρμογής.
- Ακολουθεί ο κώδικας.

```
{
  delete gRandom;
  gRandom = new TRandom(123456); // Setting the seed
  TH1F *gauss = new TH1F("gauss", "Gauss Distribution", 200, 0, 20);
  for(int i=0; i<100000; ++i){
    double a=gRandom->Gaus(10, 2);
    gauss->Fill(a);
  }

  TF1 *f1 = new TF1("f1", "gaus", 0, 20); // Orismos tis synartisis f1
  f1->SetLineColor(2);

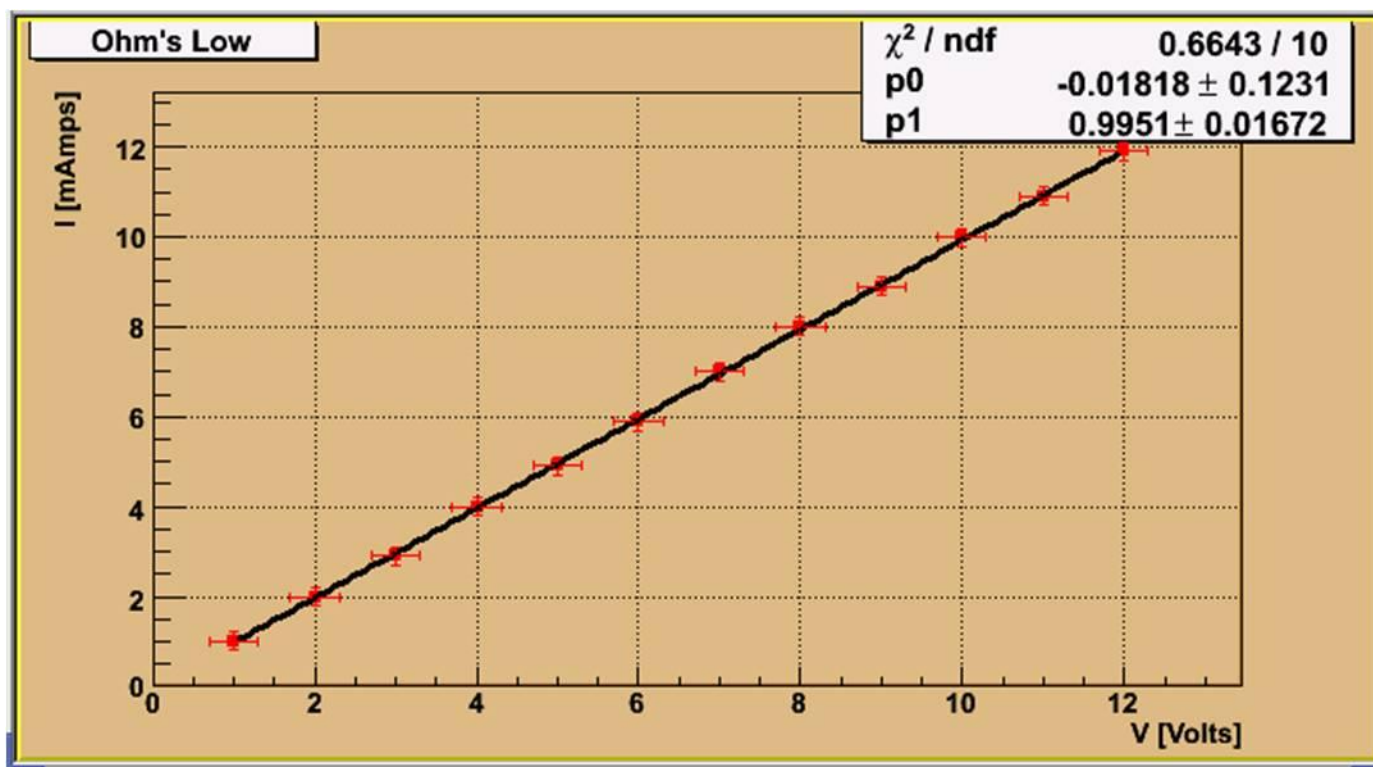
  gStyle->SetOptStat(0); // Kamia ektypsi twn statistikwn toy istogrammatos
  gStyle->SetOptFit(0); // Kamia ektypsi toy apotelesmatos tis prosarmoghs

  c1=new TCanvas("c1", "Gauss", 700, 400);
  c1->SetGrid();
  c1->SetFillColor(42);
  c1->cd();
  gauss->Fit(f1, "", "", 4, 16); // Prosarmogh istogrammatos stin f1 apo to 4 ews to 16

  double p0 = f1->GetParameter(0); //Diavasma tou apotelesmatwn tis prosarmogis
  double p1 = f1->GetParameter(1);
  double p2 = f1->GetParameter(2);
  double ep0 = f1->GetParError(0);
  double ep1 = f1->GetParError(1);
  double ep2 = f1->GetParError(2);

  char tit[256];
  txt = new TPaveText(0.6, 0.6, 0.89, 0.89, "NDC");
  sprintf(tit, "#color[1]{#font[42]{Constant = %.1f \\pm %.1f}}", p0, ep0);
  txt->AddText(tit);
  sprintf(tit, "#color[1]{#font[42]{Mean = %.1f \\pm %.1f}}", p1, ep1);
  txt->AddText(tit);
  sprintf(tit, "#color[1]{#font[42]{#sigma = %.1f \\pm %.1f}}", p2, ep2);
  txt->AddText(tit);
  txt->SetFillColor(42);
  txt->Draw();
}
```

- Η προσαρμογή των δεδομένων ενός γραφήματος σε μια συνάρτηση γίνεται με τον ίδιο ακριβώς τρόπο όπως και σε ένα ιστόγραμμα.
- Στο παράδειγμα που ακολουθεί προσαρμόζουμε τα δεδομένα τάση-ρεύμα σε ένα πολυώνυμο πρώτου βαθμού (Νόμος του Ohm).
- Το αποτέλεσμα της προσαρμογής φαίνεται παρακάτω και ο κώδικας στην επόμενη διαφάνεια.




```

{
// Dedomena Tash-Revma kai peiramatika sfalmata
double V[12]={1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0};
double I[12]={1.0, 2.0, 2.9, 4.0, 4.9, 5.9, 7.0, 8.0, 8.9, 10.0, 10.9, 11.9};

double sfalma_V[12]={0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3};
double sfalma_I[12]={0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2};

TGraphErrors *gr=new TGraphErrors(12,V,I,sfalma_V,sfalma_I); // Orismos toy grafimatos
TF1 *f1 = new TF1("f1", "pol1",0,15); // Orismos tis synartisis prosarmogis (polynomio lou vathmou)
gStyle->SetOptFit(111); // Ektypsi toy apotelesmatos tis prosarmogis

c1=new TCanvas("c1", "Grafima",1000,800);
c1->SetFillColor(42);
c1->SetGrid();

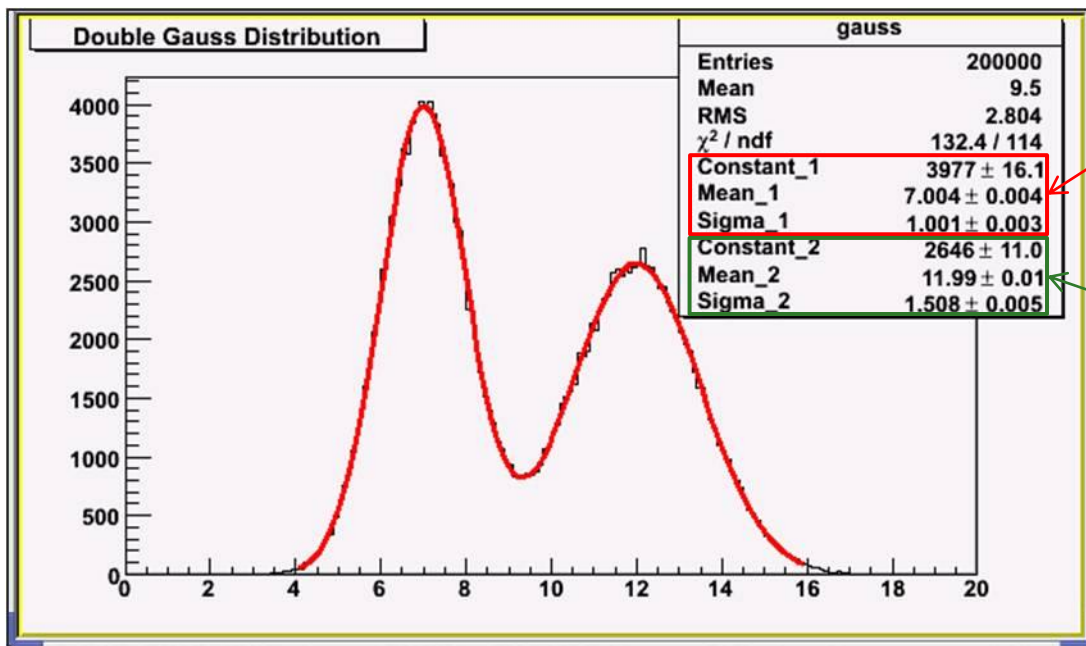
gr->SetTitle("Ohm's Law"); //Titlos grafimatos
gr->GetXaxis()->SetTitle("V [Volts]"); //Monades ston x axona
gr->GetYaxis()->SetTitle("I [mAmps]"); //Monades ston y axona

gr->SetMarkerStyle(21);
gr->SetMarkerSize(0.7);
gr->SetMarkerColor(2);
gr->SetLineColor(2);

gr->Draw("AP"); // Sxediasmos grafimatos
gr->Fit(f1, "", "", 1, 12); //Prosarmogi grafimatos
}

```

- Ας υποθέσουμε ότι έχουμε σε ένα ιστόγραμμα δεδομένα από δύο γκαουσιανές. Το πρόβλημα σε αυτή την περίπτωση είναι πολύπλοκο. Πρέπει να ορίσουμε δική μας εξωτερική συνάρτηση η οποία να περιγράφει ακριβώς το άθροισμα δύο γκαουσιανών και να την προσαρμόσουμε στα δεδομένα.
- Ο συνολικός αριθμός παραμέτρων της προσαρμογής είναι έξι (τρεις για κάθε γκαουσιανή). Οι παράμετροι της προσαρμογής πρέπει να λαμβάνουν αρχικές τιμές ώστε να κατευθύνουμε την προσαρμογή.
- Το αποτέλεσμα της προσαρμογής φαίνεται παρακάτω και ο κώδικας στην επόμενη διαφάνεια.



Αποτέλεσμα 1^{ης} Gaussian

Αποτέλεσμα 2^{ης} Gaussian

Άσκηση* : Προσαρμογή σε δύο Γκαουσιανές

```
double gauss(double *x, double *par){
    return par[0]*TMath::Gaus(x[0], par[1], par[2]);
}

double fitFunc(double *x, double *par){
    return gauss(x, par)+gauss(x, &par[3]);
}

void double_simple_gauss_fit(){
    delete gRandom;
    gRandom = new TRandom(123456); // Setting the seed

    TH1F *gauss = new TH1F("gauss", "Double Gauss Distribution", 200, 0, 20);

    for(int i=0; i<100000; ++i){
        double a=gRandom->Gaus(7, 1);
        gauss->Fill(a);
        double a=gRandom->Gaus(12, 1.5);
        gauss->Fill(a);
    }

    TF1 *f1 = new TF1("f1", fitFunc, 0, 20, 6);
    f1->SetLineColor(2);

    // Orismos oriwn kai onomatwn parametrwn prwtis gaussianis
    f1->SetParameter(0, 4000); f1->SetParName(0, "Constant_1");
    f1->SetParameter(1, 7); f1->SetParName(1, "Mean_1");
    f1->SetParameter(2, 1); f1->SetParName(2, "Sigma_1");

    // Orismos oriwn kai onomatwn parametrwn deyteris gaussianis
    f1->SetParameter(3, 2500); f1->SetParName(3, "Constant_2");
    f1->SetParameter(4, 12); f1->SetParName(4, "Mean_2");
    f1->SetParameter(5, 1.5); f1->SetParName(5, "Sigma_2");

    gStyle->SetOptFit(111);
    c1=new TCanvas("c1", "Gauss", 800, 500);
    c1->cd();
    gauss->Fit("f1", "", "", 4, 16);
}
```