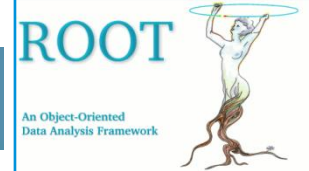


Ιστογράμματα - Γραφήματα

- Ιστογράμματα
- Παραδείγματα
 - Ιστόγραμμα 1D
 - Ιστόγραμμα 2D
 - Δεδομένα από αρχείο
 - Προσομοίωση ραδιενεργούς διάσπασης
- Γραφήματα
- Παραδείγματα
 - Γραφήματα με σφάλματα στα σημεία
 - Πολικό γράφημα



Ιστογράμματα



- Το ROOT υποστηρίζει ιστογράμματα μιας (1D), δύο (2D) και τριών (3D) διαστάσεων. Κάθε ιστογράμμο αποτελεί ένα αντικείμενο των κλάσεων
 - **TH1F, TH2F, TH3F** (τύπου float)
 - **TH1D, TH2D, TH3D** (τύπου double)
- Δημιουργία ιστογραμμάτων:

```
TH1F *h1 = new TH1F("h1", "h1 title", 100, 0, 4.4);  
TH2F *h2 = new TH2F("h2", "h2 title", 40, 0, 4, 30, -3, 3);
```
- Γέμισμα ιστογραμμάτων:

```
h1->Fill(x);  
h2->Fill(x,y);
```
- Σχεδίαση ιστογραμμάτων:

```
h1->Draw();  
h2->Draw();
```
- Η σχεδίαση των ιστογραμμάτων, γραφικών κτλ. γίνεται σε κατάλληλους καμβάδες αντικείμενα της κλάσης **TCanvas**. Για παράδειγμα:

```
c1 = new TCanvas("c1", "c1 title", 600, 400);  
c1->cd();  
h1->Draw();
```

Το διπλανό παράδειγμα:

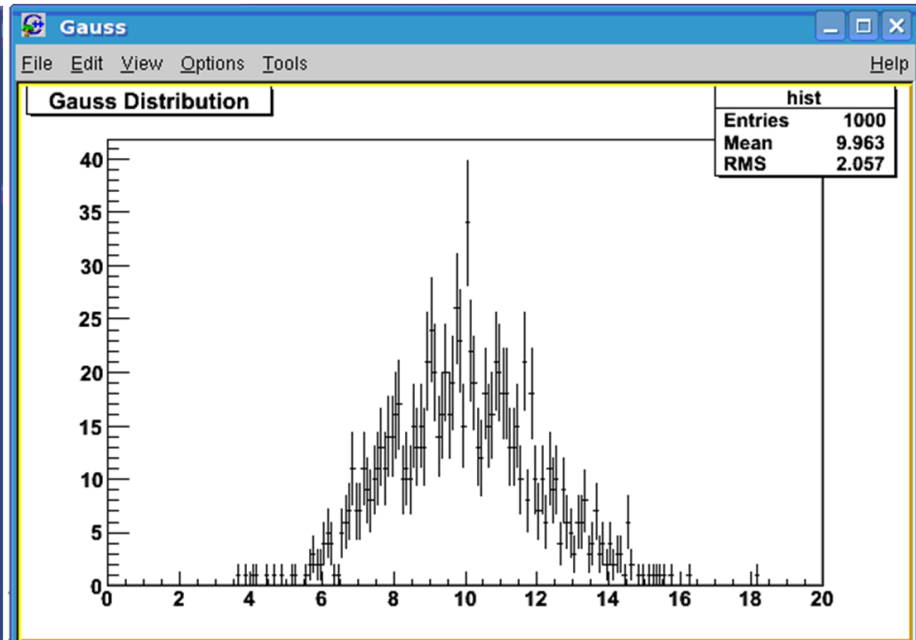
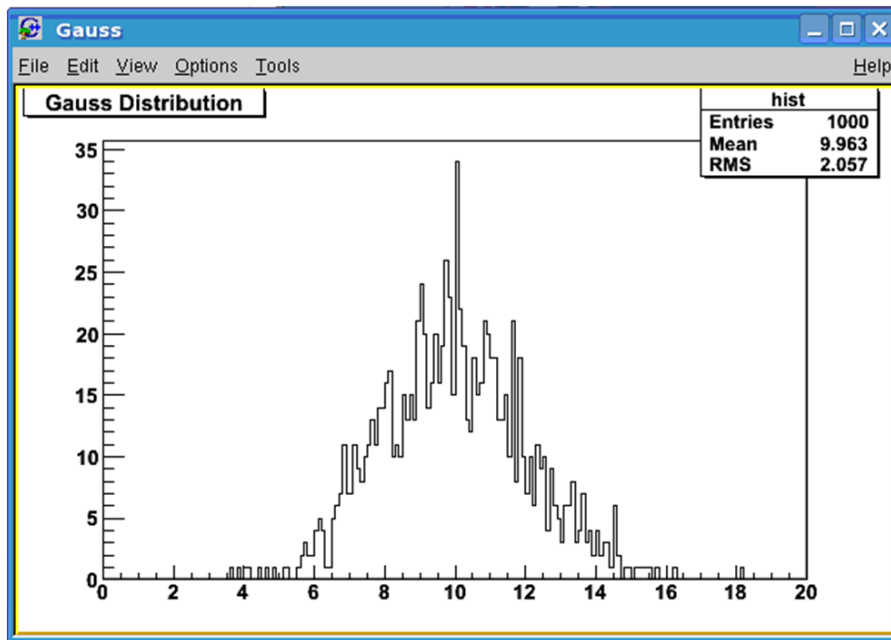
- Ορίζει το ιστόγραμμα "gauss"
- Γεμίζει με τυχαίους αριθμούς οι οποίοι ακολουθούν κατανομή Gauss
- Ορίζει τον καμβά "c1" διαστάσεων 600x400 pixels
- Σχεδιάζει το ιστόγραμμα "gauss"
- Το ιστόγραμμα δεξιά και κάτω σχεδιάζεται με : `gauss->Draw("e")`

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

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

  for(int i=0; i<1000; ++i){
    double a=gRandom->Gaus(10, 2);
    hist->Fill(a);
  }

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



Παράδειγμα 2: Ιστογράμματα 1D

Το διπλανό παράδειγμα:

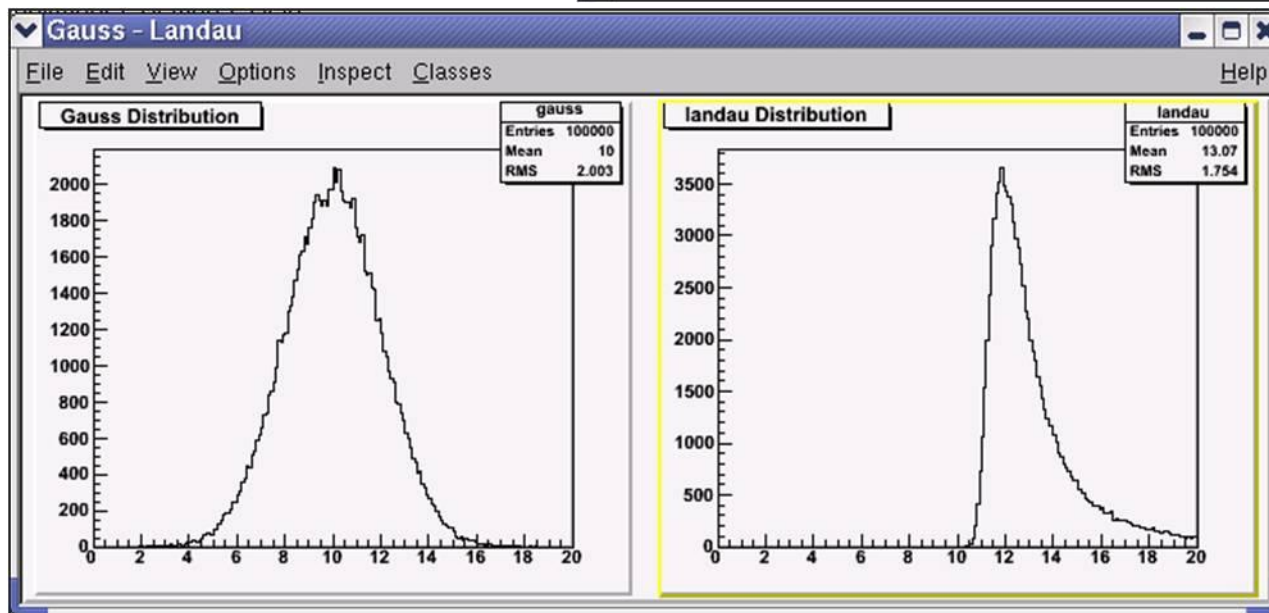
- Ορίζει τα ιστογράμματα "gauss" και "landau"
- Τα γεμίζει με τυχαίους αριθμούς.
- Ορίζει τον καμβά "c1" διαστάσεων 900x400 pixels και τον χωρίζει σε δύο μέρη
- Σχεδιάζει τα ιστογράμματα "gauss" και "landau"

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

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

  for(int i=0; i<100000; ++i){
    double a=gRandom->Gaus(10, 2);
    gauss->Fill(a);
    double b=gRandom->Landau(12, 0.5);
    landau->Fill(b);
  }

  c1=new TCanvas("c1", "Gauss - Landau", 900, 400);
  c1->Divide(2, 1);
  c1->cd(1);
  gauss->Draw();
  c1->cd(2);
  landau->Draw();
}
```



Το διπλανό παράδειγμα:

- Ορίζει τα ιστογράμματα "gauss" και "landau"
- Τα γεμίζει με τυχαίους αριθμούς.
- Ορίζει το κατάλληλο χρώμα στα δύο ιστογράμματα.
- Ορίζει τον καμβά "c1" διαστάσεων 600x400 pixels.
- Σχεδιάζει τα ιστογράμματα "gauss" και "landau" στον ίδιο καμβά.

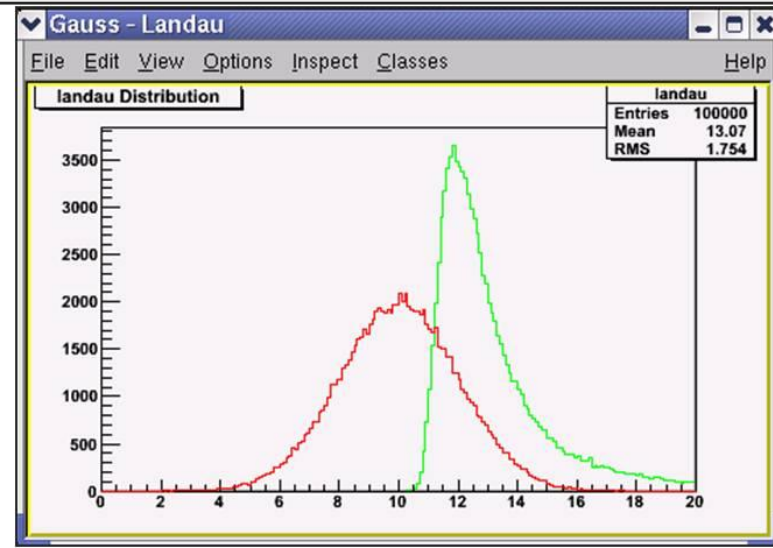
```
{
  delete gRandom;
  gRandom = new TRandom(123456); // Setting the seed

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

  for(int i=0; i<100000; ++i){
    double a=gRandom->Gaus(10, 2);
    gauss->Fill(a);
    double b=gRandom->Landau(12, 0.5);
    landau->Fill(b);
  }

  gauss->SetLineColor(2); // Kokkino xrwma
  landau->SetLineColor(3); // Prasino xrwma

  c1=new TCanvas("c1", "Gauss - Landau", 600, 400);
  c1->cd();
  landau->Draw();
  gauss->Draw("same");
}
```



Το διπλανό παράδειγμα:

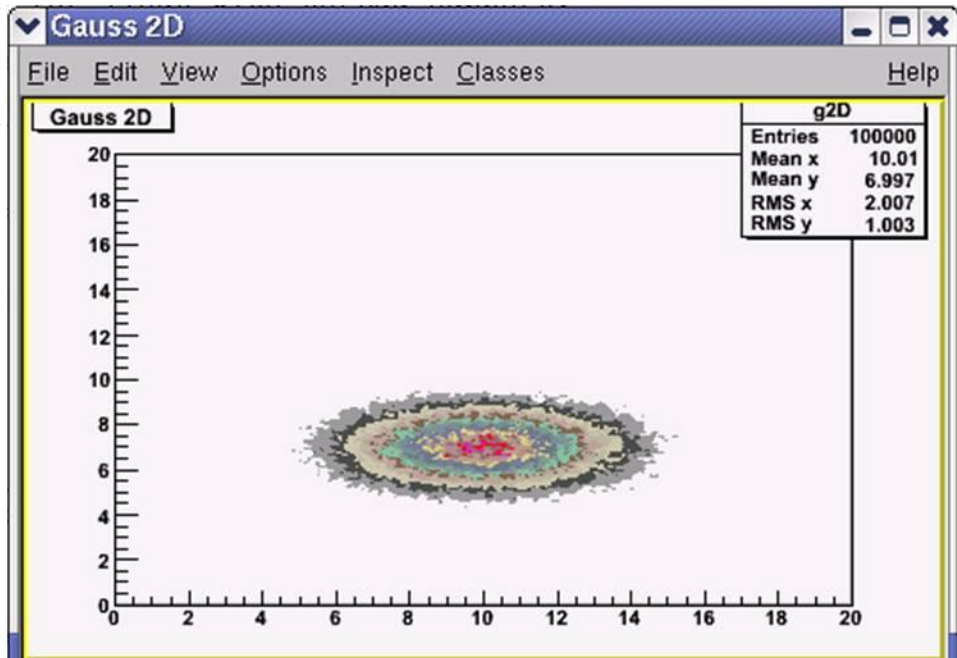
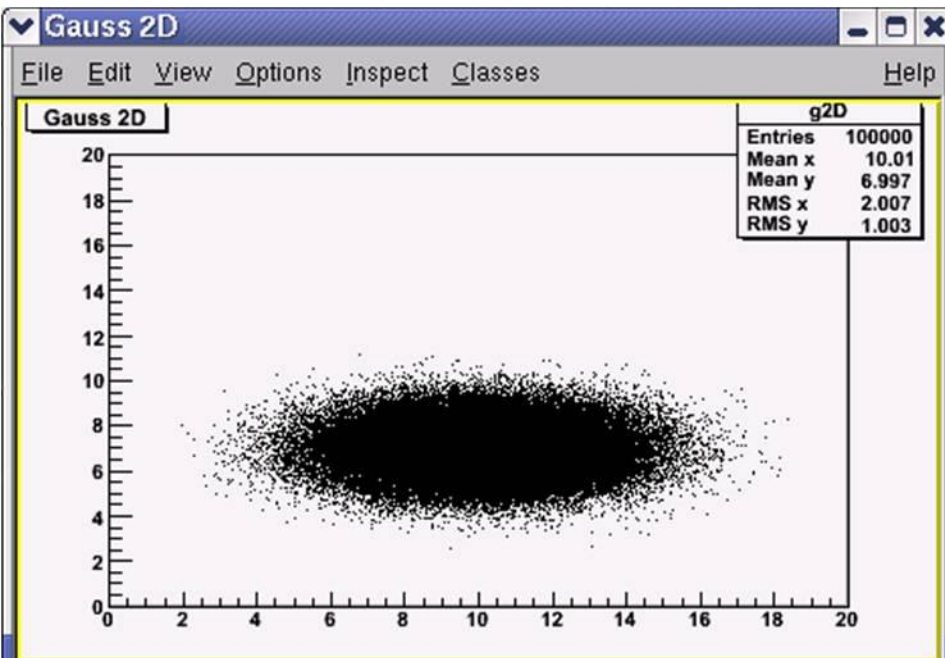
- Ορίζει το 2D ιστόγραμμα "g2D"
- Τα γεμίζει με τυχαίους αριθμούς.
- Ορίζει τον καμβά "c1" διαστάσεων 600x400 pixels.
- Σχεδιάζει το ιστογράμματα "g2D"
- Το ιστόγραμμα δεξιά και κάτω σχεδιάζεται με : `g2D->Draw("CONT")`

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

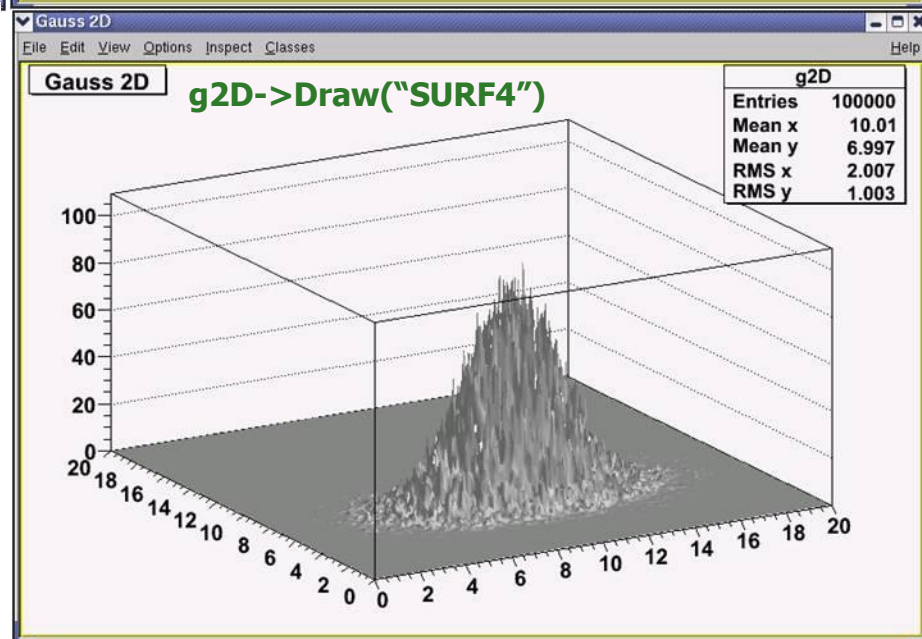
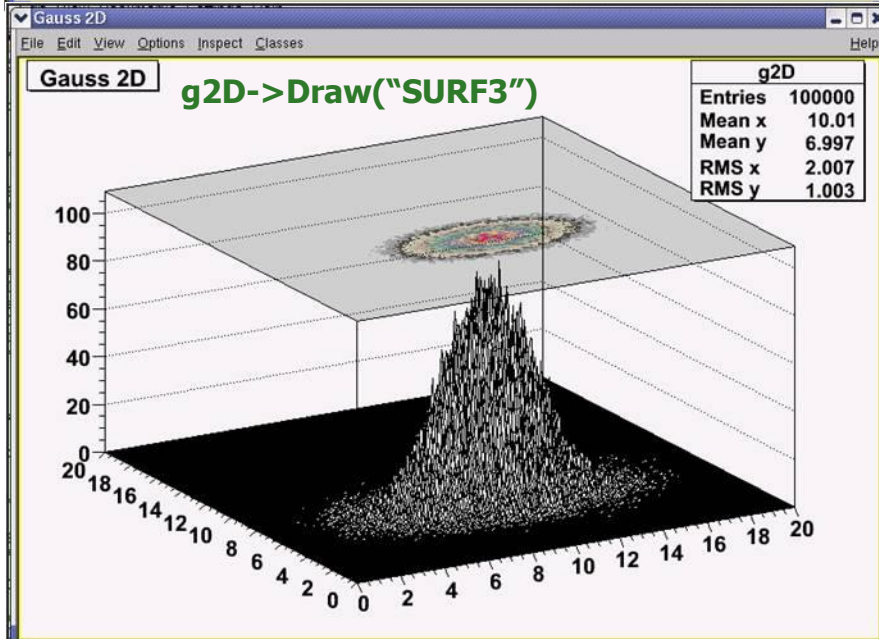
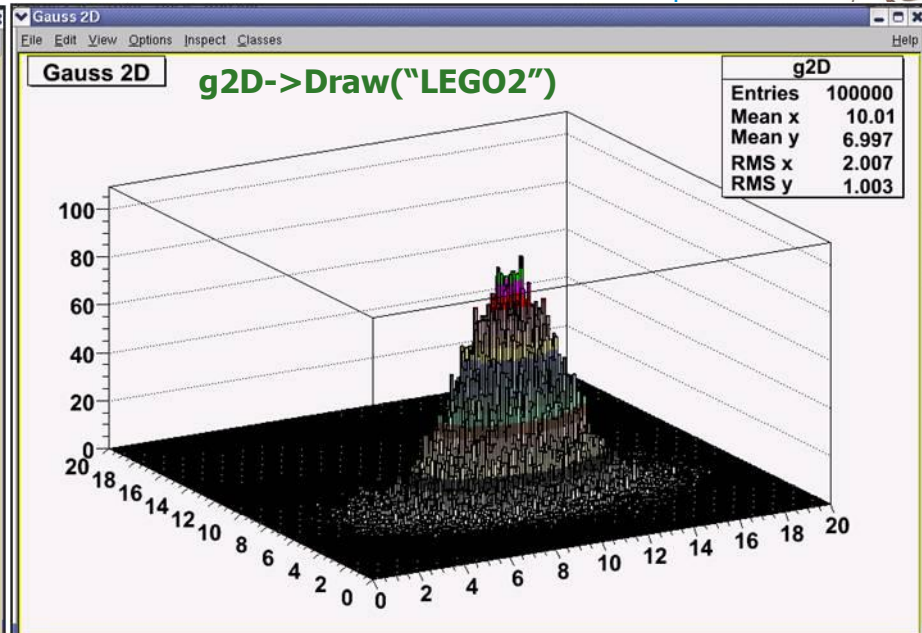
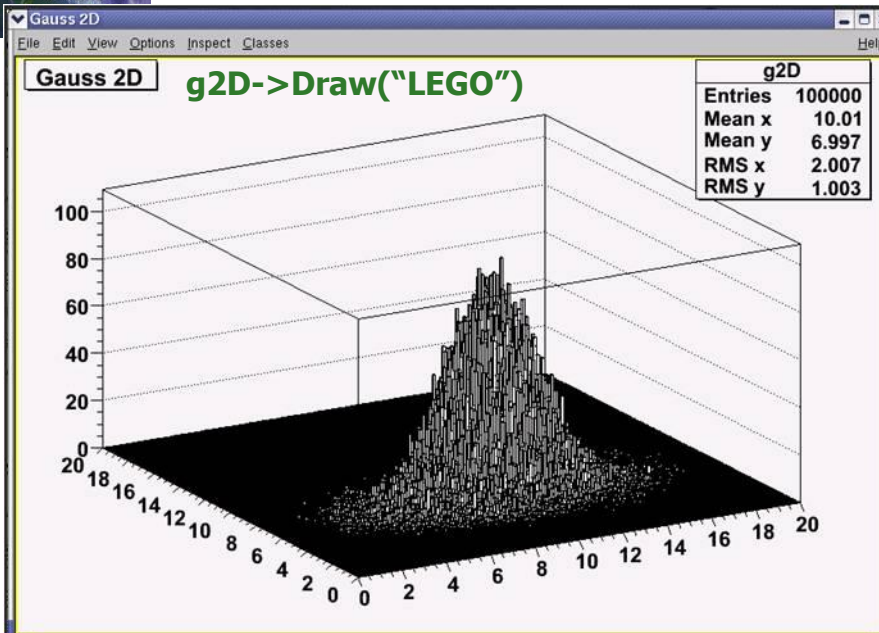
  TH2F *g2D = new TH2F("g2D", "Gauss 2D", 200, 0, 20, 200, 0, 20);

  for(int i=0; i<100000; ++i){
    double a=gRandom->Gaus(10, 2);
    double b=gRandom->Gaus(7, 1);
    g2D->Fill(a, b);
  }

  c1=new TCanvas("c1", "Gauss 2D", 600, 400);
  c1->cd();
  g2D->Draw();
}
```



Παράδειγμα 4: Ιστόγραμμα 2D



Το παρόν παράδειγμα:

- (Δεξιά επάνω) Δημιουργεί το αρχείο δεδομένων "my_data.txt" στο οποίο αποθηκεύει δεδομένα από 3 κατανομές Gauss
- (Αριστερά κάτω) Διαβάζει τα δεδομένα από το αρχείο, δημιουργεί το ιστόγραμμα "h1" και το γεμίζει με τα δεδομένα. Ορίζει τον καμβά "c1" διαστάσεων 600x400 pixels και σχεδιάζει το ιστόγραμμα "h1".

```
{
ofstream outfile("my_data.txt")

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

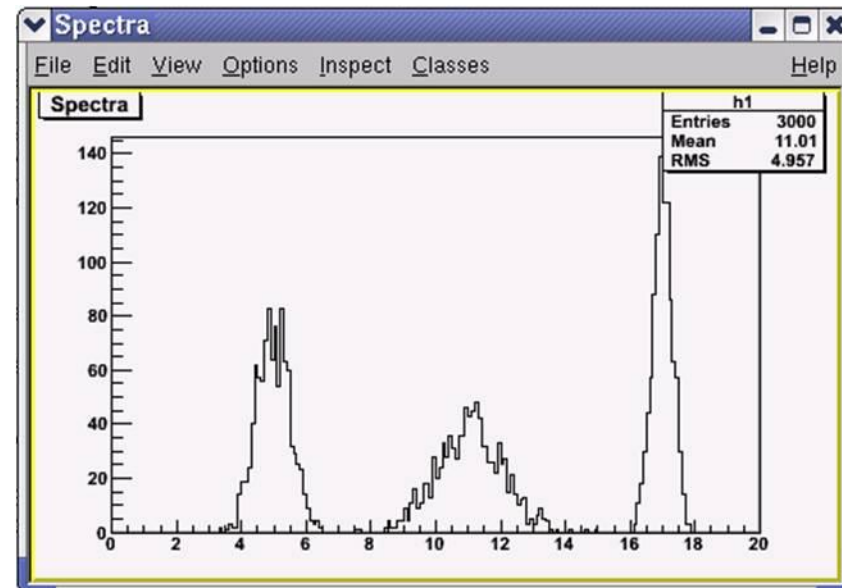
for(int i=0; i<1000; ++i){
double a=gRandom->Gaus(5,0.5);
outfile << a << "\n";
double b=gRandom->Gaus(11,1);
outfile << b << "\n";
double c=gRandom->Gaus(17,0.3);
outfile << c << "\n";
}
}
```

```
{
ifstream infile("my_data.txt"); // Input data file

TH1F *h1 = new TH1F("h1", "Spectra", 200, 0, 20);

for(int i=0; i<3000; ++i){
double a;
infile >> a;
h1->Fill(a);
}

c1=new TCanvas("c1", "Spectra", 600, 400);
c1->cd();
h1->Draw();
}
```



- Το ^{131}I είναι ένα ραδιενεργό στοιχείο με $\tau=11.57$ days. Ένας πειραματιστής καταμετρά ένα δείγμα ^{131}I επί 100 ημέρες και καταγράφει 100000 διασπάσεις. Στο τέλος κάνει το ιστόγραμμα του αριθμού των γεγονότων προς τον χρόνο από την αρχή του πειράματος. Να γράψετε ένα πρόγραμμα το οποίο να προσομοιώνει το παραπάνω πείραμα.

```

{
  delete gRandom;
  gRandom = new TRandom(123); // Setting the seed

  TH1F *decay = new TH1F("decay", "Radioactive decay ^{131}I", 100, 0, 100);

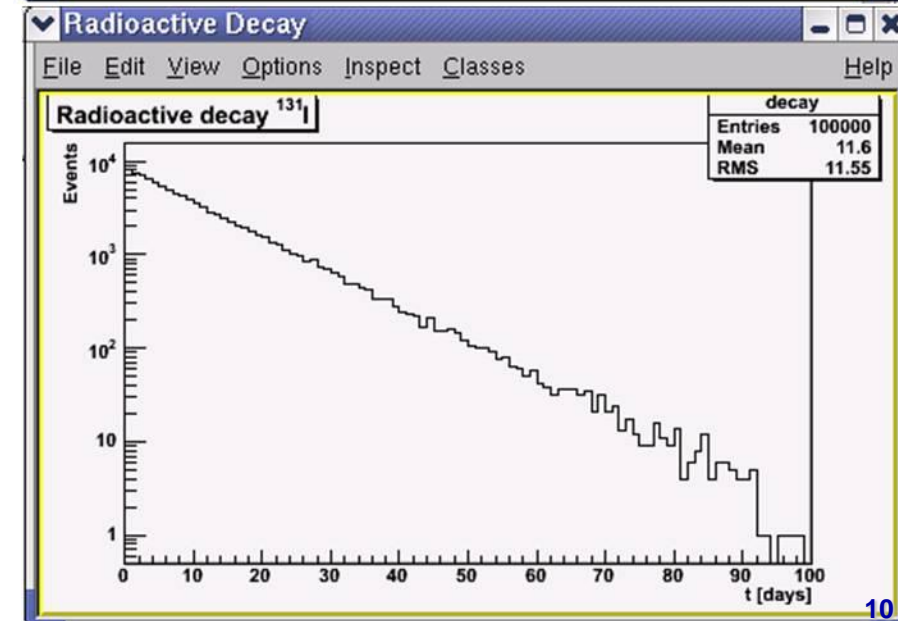
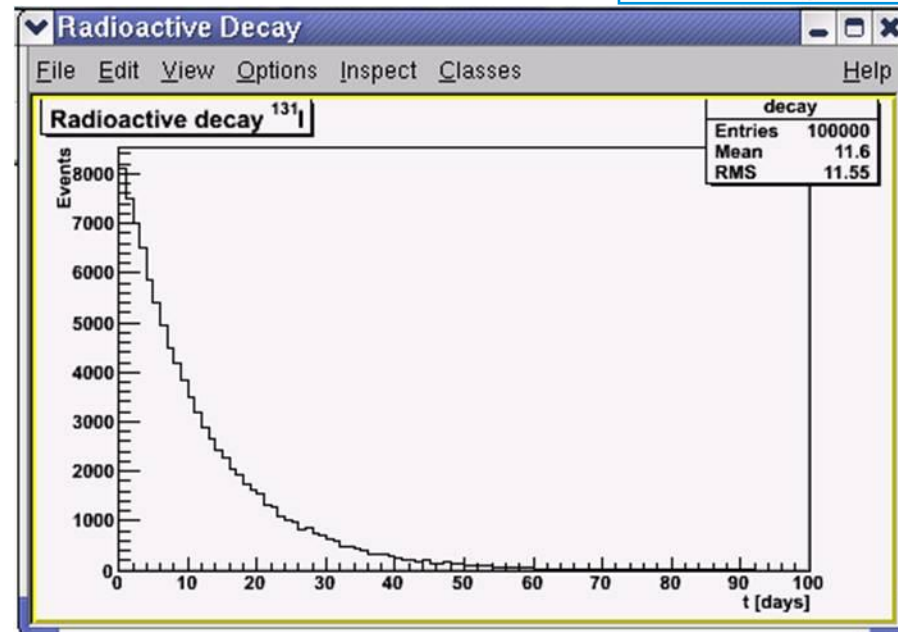
  double mesos_xronos=11.57;
  for(int i=0; i<100000; ++i){
    double a=gRandom->Exp(mesos_xronos);
    decay->Fill(a);
  }

  decay->SetXTitle("t [days]"); //Monades Axonas X
  decay->SetYTitle("Events"); //Monades Axonas Y

  c1=new TCanvas("c1", "Radioactive Decay", 600, 400);
  c1->cd();
  decay->Draw();
}

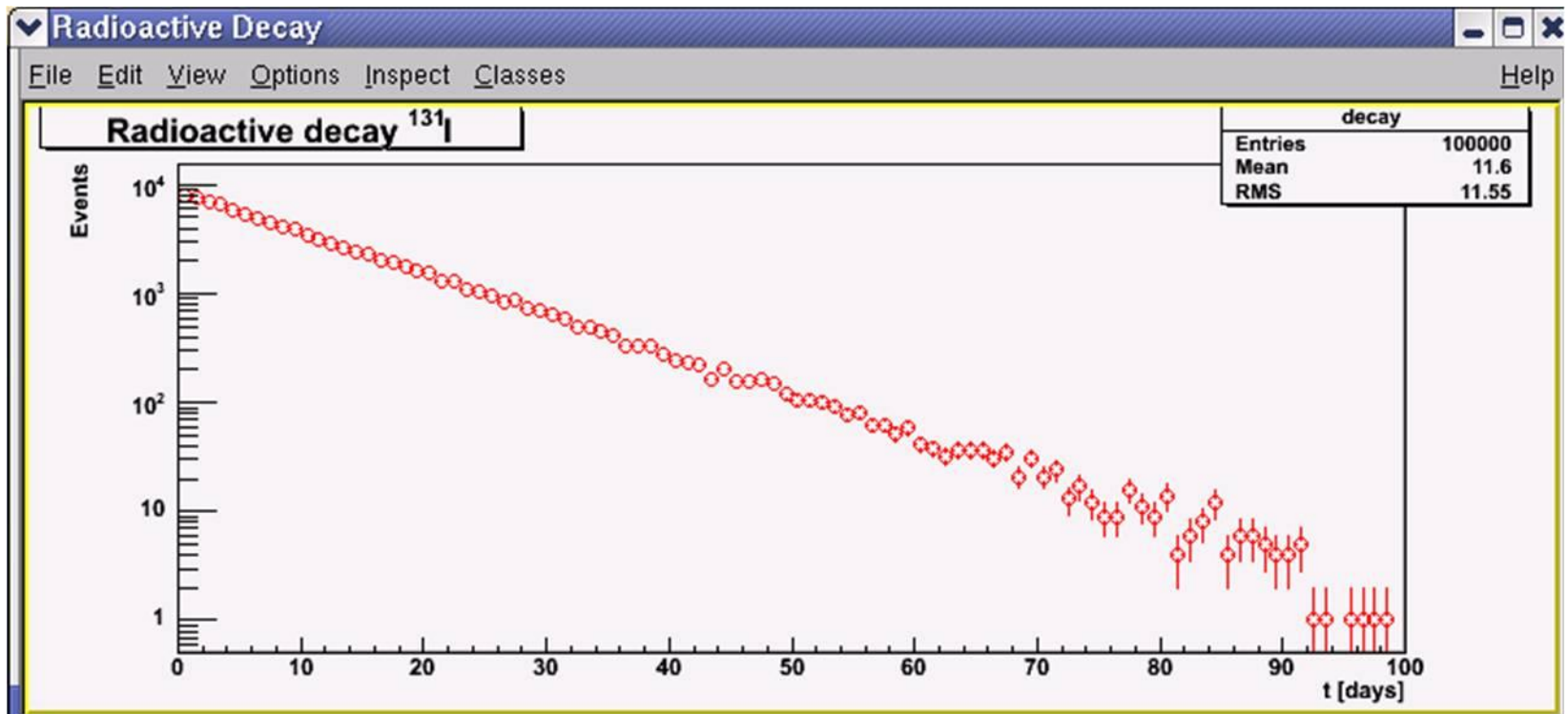
```

- Η προσομοίωση της ραδιενεργού διάσπασης γίνεται με τη δημιουργία 100000 τυχαίων αριθμών οι οποίοι ακολουθούν εκθετική κατανομή με μέσο χρόνο ζωής $\tau=11.57$ days
`gRandom->Exp(half_lifetime)`
- Το πρόγραμμα ορίζει το ιστόγραμμα "decay" και το γεμίζει με τους τυχαίους αριθμούς.
- Προσέξτε τον τρόπο με τον οποίο βάζουμε μονάδες στους άξονες. Ένα ιστόγραμμα φυσικής έχει ΠΑΝΤΟΤΕ μονάδες.
- Δεξιά επάνω είναι το ιστόγραμμα το οποίο οφείλει να μετρήσει και ο πειραματιστής.
- Δεξιά κάτω είναι το ίδιο ιστόγραμμα με λογαριθμικό τον κατακόρυφο άξονα. Αυτό γίνεται με την παρεμβολή πριν την σχεδίαση του ιστογράμματος της γραμμής:
`c1->SetLogy();`
- Στην επόμενη διαφάνεια παρουσιάζεται ένας διαφορετικός τρόπος παρουσίασης των σημείων ενός ιστογράμματος.



```
decay->SetMarkerStyle(24);
decay->SetMarkerSize(0.7);
decay->SetMarkerColor(2);
decay->SetLineColor(2);
```

```
c1=new TCanvas("c1", "Radioactive Decay", 600, 400);
c1->cd();
c1->SetLogy();
decay->Draw("e");
```



- Κάθε γράφημα αποτελεί αντικείμενο της κλάσης **TGraph**.
- Ένα γράφημα περιέχει n σημεία. Εάν υποθέσουμε ότι τα δεδομένα για τα σημεία εμπεριέχονται σε δύο πίνακες $x[n]$, $y[n]$ τότε μπορούμε να ορίσουμε το γράφημα ως:
`TGraph *gr = new TGraph(n,x,y);`
- Το ακόλουθο παράδειγμα δημιουργεί και σχεδιάζει ένα απλό γράφημα:

```

{
  int n=20; // Arithmos simeiwv grafimatos

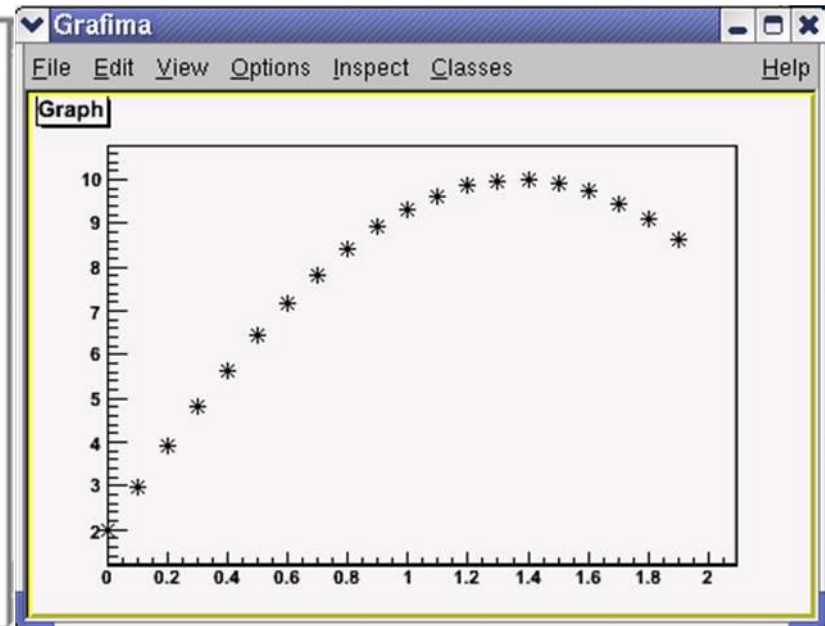
  double x[n], y[n]; // Pinakes dedomenwn

  for(int i=0; i<20; ++i){
    x[i]=i*0.1;
    y[i]=10*sin(x[i]+0.2);
  }

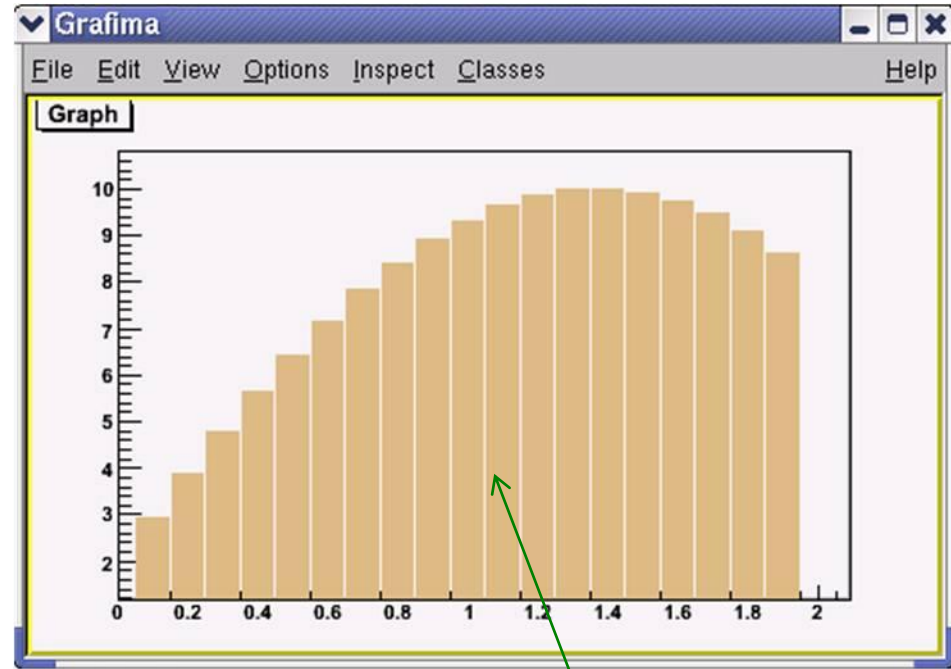
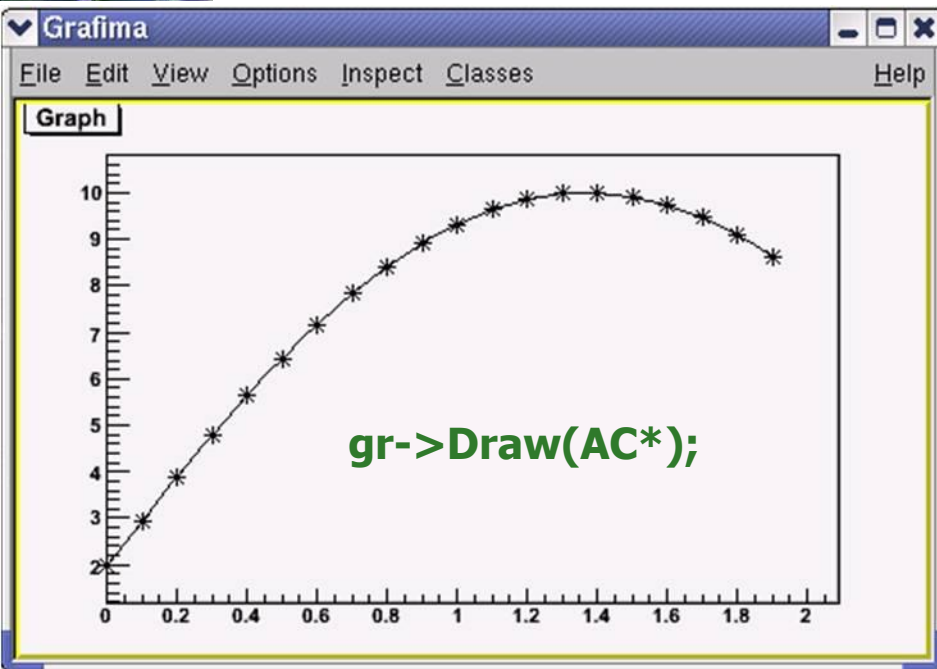
  TGraph *gr=new TGraph(n,x,y); // Orismos toy grafimatos

  TCanvas *c1=new TCanvas("c1", "Grafima", 700, 400);
  c1->cd();
  gr->Draw("A*"); // Sxediasmos grafimatos
}

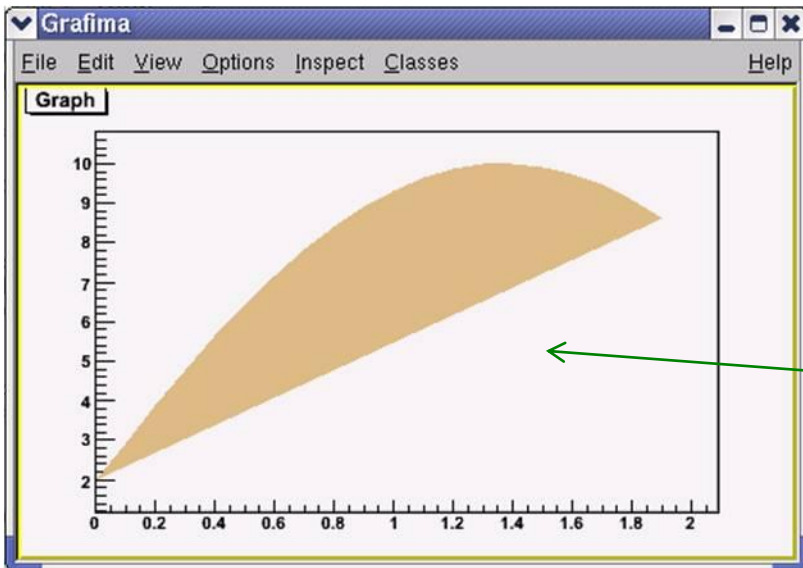
```



- Κατά τη σχεδίαση ενός γραφήματος μπορούμε να χρησιμοποιήσουμε τις ακόλουθες βασικές δυνατότητες:
 - "L" Μία απλή γραμμή ενώνει μεταξύ τους τα σημεία
 - "F" Γεμίζει την περιοχή κάτω από το γράφημα
 - "F2" Γεμίζει την περιοχή και σχεδιάζει μια πολυωνυμική γραμμή
 - "A" Τυπώνει τους δύο άξονες γύρω από το γράφημα
 - "C" Μία καμπύλη γραμμή ενώνει μεταξύ τους τα σημεία
 - "*" Ένα * τυπώνεται σε κάθε σημείο
 - "P" Ο τρέχων marker τυπώνεται σε κάθε σημείο
 - "B" Τυπώνει bar chart σε κάθε σημείο
- Ακολουθούν βασικά παραδείγματα εκτύπωσης γραφημάτων.



```
gr->SetFillColor(42);  
gr->Draw(AB);
```



```
gr->SetFillColor(42);  
gr->Draw(AF);
```

- Για να δημιουργήσουμε ένα γράφημα με σφάλματα στα σημεία αυτό πρέπει να είναι αντικείμενο της κλάσης **TGraphErrors**.

```
TGraphErrors *gr = new TGraphErrors(n,x,y,sx,sy);
```

- Όπου **n** ο αριθμός σημείων του γραφήματος, **x[n]**, **y[n]** οι πίνακες με τα δεδομένα και **sx[n]**, **sy[n]** οι πίνακες με τα σφάλματα.
- Το ακόλουθο παράδειγμα κάνει το γράφημα μιας σειράς μετρήσεων που ακολουθούν τον νόμο του Ohm.

```
{
// 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

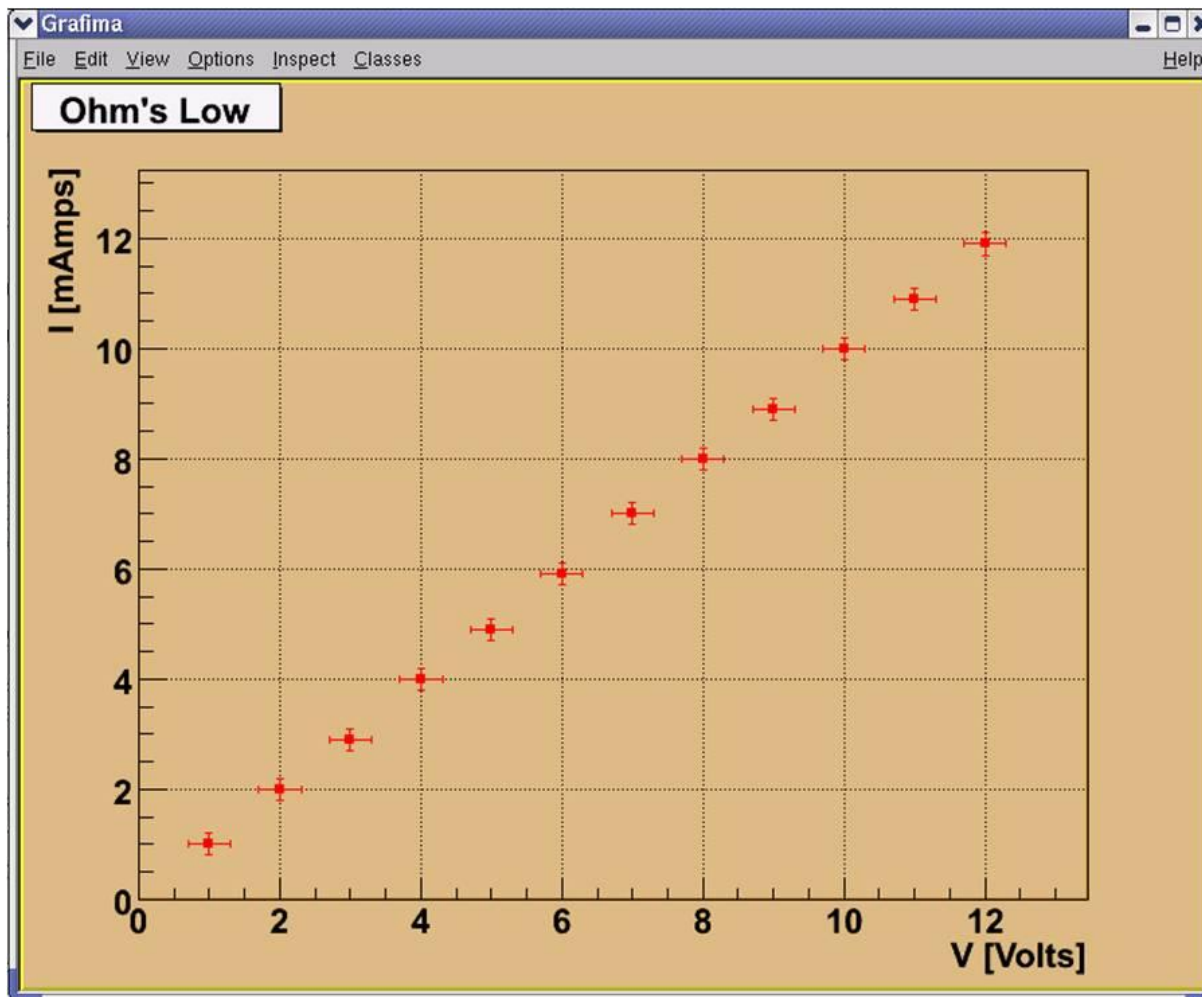
c1=new TCanvas("c1", "Grafima",1000,800);
c1->SetFillColor(42);
c1->SetGrid();

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
}
```

- Προσέξτε στον κώδικα τον τρόπο με το οποίο ορίζουμε τον τίτλο στο γράφημα, τις μονάδες στους άξονες και το στυλ, μέγεθος και χρώμα των σημείων.



- Για να δημιουργήσουμε ένα πολικό γράφημα αυτό πρέπει να είναι αντικείμενο της κλάσης **TGraphPolar**.

`TGraphPolar *gr = new TGraphPolar(n,x,y);`

γωνίες
ακτίνες

- Όπου n ο αριθμός σημείων του γραφήματος, $x[n]$, $y[n]$ οι πίνακες με τα δεδομένα.

```
{
double phimin=0;
double phimax=TMath::Pi();

double phi[1000];
double r[1000];

TF1 *fp1 = new TF1("fp1", "cos(x)", phimin, phimax) ;
for(int i=0; i<1000; ++i) {
    phi[i] = i*(phimax-phimin)/1000 + phimin; // γωνία
    r[i] = fp1->Eval(phi[i]); // ακτίνα
}

TCanvas *Cpol = new TCanvas("Cpol", "TGraphPolar Example", 600,600);

TGraphPolar *gr = new TGraphPolar(1000, phi, r);
gr->SetTitle("cos(#varphi) in polar coordinates: a circle!");
gr->SetLineColor(kRed);
gr->SetLineWidth(3);
gr->Draw ("AOL");

//-----
// Προσθέστε αυτές τις εντολές για σχεδίαση με κέντρο το 0
Cpol->Draw();
gr->SetMinRadial(0);
gr->SetMaxRadial(1); // <-- βάλτε τη σωστή τιμή του rho_max
gr->SetMinPolar(0);
gr->SetMaxPolar(2*TMath::Pi());
//-----
}
```

- Για να δημιουργήσουμε ένα πολικό γράφημα αυτό πρέπει να είναι αντικείμενο της κλάσης **TGraphPolar**.

```
TGraphPolar *gr = new TGraphPolar(n,x,y);
```



- Όπου n ο αριθμός σημείων του γραφήματος, $x[n]$, $y[n]$ οι πίνακες με τα δεδομένα.

