

ROOT 1 & 2

Salim ÇERÇİ

03.02.2015

<http://root.cern.ch/drupal/content/users-guide#UG>

Grafik :(15 dak)

1-) Aşağıdaki grafik çizen programa a-f yönergelerine uyarak eklemeler yapınız.

```
void gerrors() {  
    TCanvas *c1 = new TCanvas("c1","A Simple Graph with error bars",200,10,700,500);  
    c1->SetFillColor(42);  
    c1->SetGrid(); c1->GetFrame()->SetFillColor(21);  
    c1->GetFrame()->SetBorderSize(12);  
    const Int_t n = 10;  
    Float_t x[n] = {-0.22, 0.05, 0.25, 0.35, 0.5, 0.61,0.7,0.85,0.89,0.95};  
    Float_t y[n] = {1,2.9,5.6,7.4,9,9.6,8.7,6.3,4.5,1};  
    Float_t ex[n] = {.05,.1,.07,.07,.04,.05,.06,.07,.08,.05};  
    Float_t ey[n] = {.8,.7,.6,.5,.4,.4,.5,.6,.7,.8};  
    TGraphErrors *gr = new TGraphErrors(n,x,y,ex,ey);  
    for (Int_t i=0; i<n; i++) { x[i] = i*0.1; y[i] = 10*sin(x[i]+0.2); }  
    TGraph *gr1 = new TGraph(n,x,y);  
    gr->Draw("ALP");  
    gr1->Draw("LP");  
    c1->Update();}
```

YARDIM : <http://root.cern.ch/download/doc/ROOTUsersGuideChapters/Graphs.pdf>

- a)** Grafiğin x ekseninin başlığı –X Eksenini--- olsun
- b)** Grafiğin y ekseninin başlığı – Y Eksenini--- olsun
- c)** Grafiğin Başlığı ----Deneme----- olsun
- d)** Grafiklerin Noktalarının Renklerini ve stillerini Seçiniz.
- e)** TLegend kullanarak noktalarınıza bir isim verin.

Histogram (30 dak)

§ Bazı Histogram Çeşitlerinin oluşturulması:

- TH1* h1 = new TH1I("h1", "h1 başlık", 100, 0.0, 4.0);
- TH2* h2 = new TH2F("h2", "h2 başlık", 40, 0.0, 2.0, 30, -1.5, 3.5);
- TH3* h3 = new TH3D("h3", "h3 başlık", 80, 0.0, 1.0, 100, -2.0, 2.0, 50, 0.0, 3.0);
- TH2* h = new TH2D(/* isim */ "h2", /* başlık */ "Histogram", /* X-boyutu */ 100, 0.0, 4.0, /* Y-boyutu */ 200, -3.0, 1.5);

§ Eğer farklı bin aralıklarına sahip histogram istiyorsam ?

- const Int_t NBINS = 5;
- Double_t aralık[NBINS + 1] = {0.0, 0.2, 0.3, 0.6, 0.8, 1.0};

// Bin 1 için karşılık gelen aralık [0.0, 0.2]

// Bin 2 için karşılık gelen aralık [0.2, 0.3] ... TH1* h = new TH1D("h1", "histo", NBINS, aralık);

§ Histogramın Doldurulması

h1->Fill(x); h2->Fill(x,y); h3->Fill(x,y,z);

§ Histogramın Çizdirilmesi : h1->Draw("istediğimiz şekilde");

İsteddiğimiz şekilde : HIST, LEGO, SURF1.....

```
#include "TCanvas.h"
#include "TStyle.h"
#include "TH1.h"
#include "TGaxis.h"
#include "TRandom.h"
void histogram() {
    TCanvas *c1 = new TCanvas("c1","c1",600,400);
    gStyle->SetOptStat(11111);
    TH1F *hist[1];
    hist[0] = new TH1F("h1","h1",100,10,50);
    Int_t i;
    for (i=0;i<10000;i++) hist[0]->Fill(gRandom>Gaus(30,5.5));
    h1[0]->Draw();
    c1->Update();
}
```

Yukarıdaki programı çalıştırdığınız zaman 10000 tane parçacığın enerji dağılımını göreceksiniz. Enerji dağılımının merkezinin 30 GeV ve standart sapması 5 GeV olduğunu göreceksiniz. Bu kodu geliştirmenizi istiyoruz.

- h1 histogramı gibi h2,h3,h4 ve h5 histogramlarını oluşturunuz.
- h2 histogramının sınırları 20 ile 60 GeV olsun. Bu histogramı merkezi 40 GeV de ve sigması 4.2 GeV olan Gauss dağılımı oluşturacak şekilde rastgele sayılar ile doldurunuz.
- H3 histogramının sınırları 30 ile 70 GeV olsun. Bu histogramı merkezi 50 GeV de ve sigması 3.4 GeV olan Gauss dağılımı oluşturacak şekilde rastgele sayılar ile doldurunuz.
- H4 histogramının sınırları 40 ile 80 GeV olsun. Bu histogramı merkezi 60 GeV de ve sigması 2.9 GeV olan Gauss dağılımı oluşturacak şekilde rastgele sayılar ile doldurunuz.
- H5 histogramının sınırları 50 ile 90 GeV olsun. Bu histogramı merkezi 70 GeV de ve sigması 2.6 GeV olan Gauss dağılımı oluşturacak şekilde rastgele sayılar ile doldurunuz.

Her bir histograma gauss fiti uygulayınız.

Her bir histograma ait fit fonksiyonunun Mean ve Sigma değerlerini çağırarak

`Float_t Mean[5] = {M1, M2, M3, M4, M5}; Float_t Sigma[5] = {S1,S2,S3,S4,S5};`
dizinlerini doldurunuz.

Bu dizinleri kullanarak

`TGraph *res = newTGraph(5,Mean,Sigma);` şeklinde bir grafik çizdiriniz

Yardım : <http://root.cern.ch/download/doc/ROOTUsersGuideChapters/Histograms.pdf>

Her bir histograma gauss fiti uygulayınız.

Her bir histograma ait fit fonksiyonunun Mean ve Sigma değerlerini çağırarak

Float_t Mean[5] = {M1, M2, M3, M4, M5}; Float_t Sigma[5] = {S1,S2,S3,S4,S5}; dizinlerini doldurunuz.

Bu dizinleri kullanarak

TGraph *res = new TGraph(5,Mean,Sigma); şeklinde bir grafik çizdiriniz

Çizdiğiniz grafiğe aşağıdaki fonksiyonu uydurarak a,b,c katsayılarını hesaplayınız

$$f(x) = \sqrt{\left(\frac{a}{x}\right)^2 + \left(\frac{b}{\sqrt{x}}\right)^2} + c$$

```
TF1 *resolution = new TF1("resolution","TMath::Sqrt((([0]*[0])/pow(x,2)+([1]*[1])/x+[2]*[2])",10.,100.);
```

```
resolution->SetParName(0,"a");
```

```
resolution->SetParName(1,"b");
```

```
resolution->SetParName(2,"c");
```

```
res->Fit(resolution);
```

Tree (45 dak)

```
void tree(){  
  
    TFile f("tree1.root","recreate");  
    TTree t1("t1","a simple Tree with simple variables");  
    Float_t px, py, pz;  
    Int_t ev;  
    t1.Branch("px",&px,"px/F");  
    t1.Branch("py",&py,"py/F");  
    t1.Branch("pz",&pz,"pz/F");  
    t1.Branch("ev",&ev,"ev/I");  
    // fill the tree  
    for (Int_t i=0; i<10000; i++) { gRandom->Rannor(px,py);  
        pz = px*px + py*py;  
        ev = i;  
        t1.Fill(); }  
    t1.Write();  
}
```

<https://root.cern.ch/download/doc/ROOTUsersGuideChapters/Trees.pdf>


```

void readtree(){

    TFile *f = new TFile("tree1.root");
    TTree *t1 = (TTree*)f->Get("t1");
    Float_t px, py, pz;
    Double_t random;
    Int_t ev;
    t1->SetBranchAddress("px",&px);
    t1->SetBranchAddress("py",&py);
    t1->SetBranchAddress("pz",&pz);
    t1->SetBranchAddress("ev",&ev);

    TH1F *hpx = new TH1F("hpx","px distribution",100,-3,3);
    TH2F *hpxpy = new TH2F("hpxpy","py vs px",30,-3,3,30,-3,3);
    Int_t nentries = (Int_t)t1->GetEntries();
    for (Int_t i=0; i<nentries; i++) {
        t1->GetEntry(i);
        hpx->Fill(px);
        hpxpy->Fill(px,py);
    }
    TCanvas *c1 = new TCanvas("c1","c1",600,400); gStyle->SetOptStat(11111);
    c1->Divide(1,2);
    c1->cd(1);
    hpx->Draw();
    c1->cd(2);
    hpxpy->Draw("surf2");
}

```