Exercises for the lecture „Moderne Methoden der Datenanalyse”

Prof. Dr. S. Hansmann-Menzemer, M. Schiller
Physikalisches Institut der Universität Heidelberg

Apr 27 2010

Some useful Root commands

1D Histograms

create histogram	`TH1D* h = new TH1D("name", "title", nbins, xmin, xmax);`
fill histogram	`h->Fill(x, weight);`
find bin number	`int ibin = h->FindBin(x);`
set bin content	`h->SetBinContent(ibin, content);`
set bin error	`h->SetBinError(ibin, err);`
get bin content	`double content = h->GetBinContent(ibin);`
draw histogram	`h->Draw();`
reset historgram	`h->Reset();`
get array with contents	`Double_t* arr = h->GetArray();`
set low end of y axis	`h->SetMinimum(ymin);`
set high end of y axis	`h->SetMaximum(ymax);`

2D Histograms

create histogram:

```
TH2D* h = new TH2D("name", "title", nbinsx, xmin, xmax, nbinsy, ymin, ymax
);
```

fill histogram: `h->Fill(x,y,weight);`

Sometimes, a `TProfile` histogram is more expressive than a 2D scatter plot:
create histogram:

```
TProfile* h = new TProfile("name", "title", nbinsx, xmin, xmax, ymin, ymax
);
```

Functions

create function	`TF1* f = new TF1("name", "[0]*sin([1]*x)/x+[2]", xmin, xmax);`
set parameters	`f->SetParameters(2., 0.24, 1e-3);`
fit to histogram	`h->Fit(f, "", "", xmin, xmax);`
get ith parameter	`Double_t pari = f->GetParameter(i);`
get error on ith parameter	`Double_t erri = f->GetParError(i);`
predefined function names:	`"gaus"`,`"expo"`, `"polN"` (N=0, 1, 2, . . .), `"landau"`

Random numbers

predefined distributions	"gaus($\mu = 0, \sigma = 1$)", "uniform(x1,x2)", "landau($\mu = 0, \sigma = 1$)", "poisson(μ)", "binomial(n,p)"
fill a histogram	`h->FillRandom("gaus", 1000);`
fill acc. to function	`h->FillRandom("fname", 1000);`

replace RNG:	`delete gRandom;`
	`gRandom = new TRandom3(0);`

gStyle

You can adapt the amount of information given in the statistics box of a histogram and the amount of information displayed along with fit parameters by using the following calls (vary the number of 1's or replace some of the by 0 to see the effects):

```
gStyle->SetOptFit(11...1);
gStyle->SetOptStat(11...1);
```

You can also set a plain style (no grey background in histograms etc.) by using:

```
gROOT->SetStyle("Plain");
```

Ntuples

create ntuple	`TNTtuple* nt = new TNtuple("name","title","x1:x2:x3");`
fill	`nt->Fill(x1, x2, x3);`
draw x_1	`nt->Draw("x1");`
project x_1 into histogram	`TH1* h = nt->Project("hname", "x1);`
draw with cuts	`nt->Draw("x1", "x2␣<␣3");`
draw scatter plot	`nt->Draw("x1:x2");`

Should you come over a `TTree`, please note that drawing and projecting works the same way, in fact, a `TNtuple` is a special kind of `TTree`.

Files

open file	`TFile* file = new TFile("myfile.root", "RECREATE");`
write histogram to file	`file->cd(); hist->Write();`
close file	`file->Close();`
read histogram from file	`TH1* hist = (TH1*)file->Get("hname");`
read ntuple from file	`TNtuple* nt = (TNtuple*)file->Get("ntname");`

Canvases

create canvas	`TCanvas* c = new TCanvas("name", "title", 500, 700);`
draw histogram into `c`	`c->cd(); h->Draw(); c->Update();`
divide canvas (2x3 tiles)	`c->Divide(2,3);`
draw into first tile	`c->cd(1); h->Draw(); c->Update();`
print canvas to file	`c->Print("myplot.eps");` or `c->Print("myplot.png");`