Histograms

Anna Simon

October 16, 2015

Copying files

A disk space with all the examples, data files for future homeworks, etc. is at: /afs/crc.nd.edu/group/PHYS60070 It is a read only access, feel free to copy whatever you need to your home directory.
How to copy files from a command line:

- between locations on CRC computer:

```
cp /path/to/file /path/to/destination
```

- from CRC computer to your local machine

```
scp netID@regulus.crc.nd.edu:/afs/crc.nd.edu/group/PHYS60070/
    functions.C /path/to/destination
```

- if the destination is your current directory, the full path can be replaced by period "."

```
scp netID@regulus.crc.nd.edu:/afs/crc.nd.edu/group/PHYS60070/
    functions.C .
```

- Filezilla is a free client that can be used for file transfer (with GUI)

Viewing/editing files via command line

Open text files:

```
gedit filename.C & // ampersand will prevent the command from
    locking the prompt
```

Open a graphics file (png, jpg)

```
display filename.png //opens a simple viewer
gnome-open filename.png // opens file using default program (on CRC
    it's Gimp)
```

Open a pdf or eps file

```
gnome-open filename.pdf // opens with default program (Pdf Viewer)
```

NOTE: You can have as many terminals with active login to CRC as you want, in principle one could be running ROOT and another is open to have file access.

Histograms

From Wikipedia:

- A histogram is a graphical representation showing a visual impression of the distribution of data. It is an estimate of the probability distribution of a continuous variable and was first introduced by Karl Pearson.
- A histogram consists of tabular frequencies, shown as adjacent rectangles, erected over discrete intervals (bins), with an area equal to the frequency of the observations in the interval.
- The height of a rectangle is also equal to the frequency density of the interval, i.e., the frequency divided by the width of the interval. The total area of the histogram is equal to the number of data entries.

Histograms

ROOT histograms, 1- 2- and 3D, can store various type of data:

- TH1C : histograms with one byte per channel. Maximum bin content = 127
- TH1S : histograms with one short per channel. Maximum bin content = 32767
- TH1I : histograms with one int per channel. Maximum bin content = 2147483647
- TH1F : histograms with one float per channel. Maximum precision 7 digits
- TH1D : histograms with one double per channel. Maximum precision 14 digits

Creating a histogram

Allocate dynamic memory for the histogram:

```
TH1F *h1 = new TH1F("h1", "sample histogram", 30, 0, 15)
```

This will create a histogram storing data with 7-digit precision in 30 bins in the range from 0 to 15

To fill the histogram:

```
h1->Fill(10,4); // find a bin corresponding to value 10 and increment it by 4

h1->Fill(5); //// find a bin corresponding to value 5 and increment it by 1
```

And a loop through the whole histogram

```
for (int bin=0;bin<30;bin++){
    double x = h1->GetBinCenter(bin);
    double y = pow(x,2) - 3*x;
    h1->Fill(x,y);
}
```

Creating a histogram

- Using the fSum function from previous class create a histogram that will represent a "spectrum" containing two peaks on a linear background. (f1− >Eval(x) - calculate f1(x))
- Make sure the number of bins in your histograms is sufficient to show the spectrum
- Check various plotting options described at http://root.cern.ch/root/html/THistPainter.html
- Open the Fit Panel:

```
h1->FitPanel()
```

  - Fit a linear background between the peaks
  - Fit a Gaussian to the peak at x=15
  - Fit fSum to the peak at x=15
  - Fit fSum to the whole spectrum (you may need to set and fix some parameters to "help" the fitting procedure)

Random numbers (TRandom)

Random number generators are available within ROOT. Following distributions are available:
Exp(tau), Integer(imax), Gaus(mean,sigma), Rndm() - uniform distribution from 0 to 1, Uniform(x1) - uniform distribution from 0 to x1, Landau(mpv,sigma), Poisson(mean), Binomial(ntot,prob)

```
double r = gRandom->Gaus(0,1);
```

To get a random number generated from any function ($\forall x \; f(x) \geq 0$):

```
TF1 *f1 = new TF1("f1","abs(sin(x)/x)",0,10);
double r = f1->GetRandom();
```

```
//fill in a histogram with random numbers:
int nEvents = 1000;
for (int i=0;i<nEvents;i++)
{
    double r = f1->GetRandom();
    h1->Fill(r);
}
```

”Inspecting” a histogram

Retrieving histogram properties:

```
h1->GetEntries()

h1->GetMean()

h1->GetMeanError()

h1->GetRMS()

h1->GetMinimum()

h1->GetTitle()
```

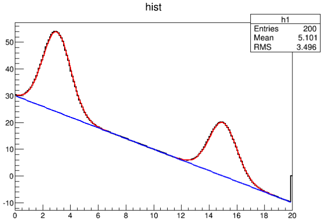
and many many other...

Setting histogram properties:

```
h1->SetTitle("new title")

h1->GetXaxis()->SetTitle("x-axis title")

h1->GetYaxis()->SetLabelSize(0.05) //in fractions of the canvas size
```

Fitting within a macro

```cpp
//initialize the parameters for the fit function
fSum->SetParameters(10,2,1,10,12,1,1,-2);
//fit the histogram
h1->Fit("fSum");

//copy the fit parameters to an array
double FitPar[8];
fSum->GetParameters(FitPar);
//set the background parameters to the fit result
fBgrd->SetParameters(&FitPar[6]);

//draw the background function on the histogram
fBgrd->SetLineColor(4);
fBgrd->Draw("same");
```



Fitting the background

```
//clone the original histogram (to make sure all of them have the same nBins and range)
TH1 *hSig = (TH1*) h1->Clone();
hSig->SetName("hSig");
hSig->SetLineColor(3);
TH1 *hbgrd = (TH1*) h1->Clone();
hbgrd->SetName("hbgrd");

//fit the background, 20 iterations
//store the result in hbgrd
hbgrd = h1->ShowBackground(20);

//subtract the background from hSig
hSig->Add(hbgrd,-1);

//draw all the histograms
h1->Draw();
hSig->Draw("same");
hbgrd->Draw("same");
```

Homework

- Check the examples fitExclude and multifit at http://root.cern.ch/root/html/tutorials/fit to learn how to fit ranges of a histogram and exclude ranges from a fit
- Using random number generators create a histogram that contains three Gaussians (at x=20, 50, 80, area=100, sigma=1) on an quadratic background ($-0.01x^2 + x + 9$). The total spectrum should have at least 2000 events and span the x-range from 0 to 100.
- Fit the background using ShowBackground(), a quadratic function with and without excluding the peaks.
- For each fitting procedure plot the original spectrum, the background and the spectrum after background subtraction.

NOTE: how to put multiple plots in a single canvas

```
//Create a canvas c1 with a size of 800x600 px
TCanvas *c1 = new TCanvas("c1","Homework plots",800,600);
//Divide the canvas into two rows and two columns
c1->Divide(2,2); //Divide(#columns,#rows)
//activate the first section of the canvas
c1->cd(1);
```