Computing and Statistical Data Analysis
Some slides about ROOT

Interactive data analysis program and class library.
Commands based on C++ (CINT interpreter).
Home page: root.cern.ch
Many tutorials, e.g. google for ROOT tutorial
See course web page for stand-alone C++ programs that use ROOT classes, e.g., for histograms.
Run the program

Installation and set up non-trivial. See your local expert.

To run, type \texttt{root}
Run the demo program

Try typing this stuff in yourself.

```
ROOT is based on CINT, a powerful C/C++ interpreter.
Blocks of lines can be entered within {...}.
Previous typed lines can be recalled.

Root > float x=5; float y=7;

Root > x*sqrt(y)

(double) 1.322875655552e+01

Root > for (int i=2;i<7;i++) printf("sqrt(%d) = %f",i,sqrt(i));

sqrt[3] = 1.732051
sqrt[4] = 2.000000
sqrt[5] = 2.236068
sqrt[6] = 2.449490

Root > TF1 f1("f1","sin(x)/x",0,10)

Root > f1.Draw()
```
Book, fill, display a histogram
Fancy stuff with histograms
Plot a graph with error bars
Function minimization for fitting
Working with ROOT

In this course we will write stand-alone C++ programs that use classes from the ROOT library.

Often the program will create histograms (or n-tuples), that we store in a root file (e.g. myHistogramFile.root) and we can use the ROOT executable program to analyze these.

To work with ROOT classes, e.g., to know what member functions are available, check the class definition on the web.

You can either find the documentation on root.cern.ch or google for the class name (ROOT classes always start with T, e.g., TFile, TRandom,...)
Googling for ROOT class name usually quickest

To open non-local files use the static TFile::Open() method, that will take care of opening the files using the correct remote file access plugin. ...

root.cern.ch/root/html/TFile.html - 195k - Cached - Similar pages

TFile
Array of pointers to TProcessIDs static Double_t fgBytesWrite Number of bytes written by all TFile objects static Double_t fgBytesRead Number of bytes read ...

root.cern.ch/root/html302/TFile.html - 58k - Cached - Similar pages

[ More results from root.cern.ch ]
Sample program `simpleMC.cc`

Below is a stand alone C++ program that uses ROOT classes for random numbers and histograms:

```cpp
#include <TH1D.h>
#include <TFile.h>
#include <TRandom3.h>

using namespace std;

int main(){

    // Open output file
    TFile* file = new TFile("simpleMC.root", "recreate");

    // Book histograms
    TH1D* h_Uni = new TH1D("h_Uni", "uniform random numbers", 100, 0, 1.0);
    TH1D* h_Exp = new TH1D("h_Exp", "exponential random numbers", 100, 0, 5.0);

    title # of bins bin limits
```
// Create a TRandom3 object to generate random numbers

int seed = 12345;
TRandom3* ran = new TRandom3(seed);

// Generate some random numbers and fill histograms

const int numValues = 10000;
const double xi = 1.0;                        // mean of exponential pdf

for (int i=0; i<numValues; ++i){
    double r = ran->Rndm();               // uniform in ]0,1]
    double x = - xi * log(r);
    h_Uni->Fill(r);
    h_Exp->Fill(x);
}

// Store all histograms in the output file and close up

file->Write();
file->Close();

return 0;
}
GNUmakefile to build simpleMC

PROGNAME = simpleMC
SOURCES = simpleMC.cc
INCLUDES =
OBJECTS = $(patsubst %.cc, %.o, $(SOURCES))
ROOTCFLAGS := $(shell root-config --cflags)
ROOTLIBS := $(shell root-config --libs)
ROOTGLIBS := $(shell root-config --glibs)
ROOTLIBS := $(shell root-config --nonew --libs)
CFLAGS += $(ROOTCFLAGS)
LIBS += $(ROOTLIBS)
# Not sure why Minuit isn't being included -- put in by hand
#
LIBS += -lMinuit
LDFLAGS = -O

$(PROGNAME): $(OBJECTS)
g++ -o $@ $(OBJECTS) $(LDFLAGS) $(LIBS)

%.o : %.cc $(INCLUDES)
g++ ${CFLAGS} -c -g -o $@ $<
Looking at output `simpleMC.root`

You type at root prompt (lines numbered [0], [1], etc.)

```
[Inappserv2]~/.WWW/stat/root/mc> root
*******************************
* Welcome to ROOT v5.14/00e *
*******************************
root [0] TFile* f = new TFile("simpleMC.root");
root [1] f->ls();
TFile** simpleMC.root
TFile* simpleMC.root
  KEY: TH1D h_Uni:1 uniform random numbers
  KEY: TH1D h_Exp:1 exponential random numbers
root [2] h_Uni->Draw();
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [3] 
```

ROOT creates canvas.

To save plot: File, Save as, etc.
Better plots (store commands in a macro file)

// To execute, type .X plotHist.C
{
  TFile* f = new TFile("simpleMC.root");
  f->ls();
  TH1D* h1 = (TH1D*)f->Get("h_Exp");
  h1->SetXTitle("x");
  h1->SetYTitle("f(x;#xi)";
  h1->Draw();
}
Carry on with ROOT

Root home page: root.cern.ch (manual, tutorials, etc.)

Class definitions: root.cern.ch/root/html/ClassIndex.html
E.g. for TH1D: root.cern.ch/root/html/TH1D.html
(tick box “show inherited”).

More ROOT lectures (links from course website):

Adrian Bevan (QMUL)
www.ph.qmul.ac.uk/~bevan/GCL/ROOT.pdf

Benno List (DESY)
www.desy.de/~blist/summerstudents/summer_lectures.2007cpp.html