

Solutions To ROOT Exercises

Session A

Question A1:

1. Once you have run the tutorial `hsum.C`, select the histogram "s2" with the right button.
2. Select the "FitPanel"
3. In the FitPanel, select "Landau" and "Same Picture", and use the slider to select the sub-range for s2.

Question A2:

1. Double click on `hsimple.root` to see the contents of the file
2. Double click on `ntuple` to see the contents of the ntuple
3. Double click on `pz` to see a histogram of `pz`.

These are the lines needed to do the same thing from the command line. This assumes the `hsimple.root` is not yet opened.

```
root [] TFile f("hsimple.root");  
root [] ntuple->Draw("pz");
```

Question A3:

1. Drag the `pz` box to the X box.
2. Drag the `px` box to the Y box.
3. Right click on the white label in the `Gopt` box and select `SetLabel`
4. Type "`prof`"
5. Click on the Draw button.

These are the lines needed to do the same thing from the command line. This assumes the `hsimple.root` is not yet opened.

```
root [] TFile f("hsimple.root");
root [] ntuple->Draw("pz:px","", "prof");
```

Question A4:

To add the fit:

1. Right click on the graph and select the Fit Panel from the context menu.
2. Select "pol2" and "same picture"
3. Click on Fit.

These are the lines needed to do the same thing from the command line. This assumes the `hsimple.root` is not yet opened.

```
root [] TFile f("hsimple.root");
root [] ntuple->Draw("pz:px","", "prof");
root [] htemp->Fit("pol2");
```

Question A5:

1. In the Tree Viewer, drag the `py` box to the X box.
2. Drag the `pz` box to the Ybox
3. Set the label in the weight box to: `px*px + py*py < 20`
4. Look up the draw options and set the label in the `Gopt` box.

Question A6:

The first line finds the global list of contours using the naming service of `gROOT`. It cast it to a pointer to an array of `TObjects`. Each contour can have multiple disjoint poly lines. The second line gets the list of poly lines for the first contour. The third statement gets the first poly line in the list of poly lines for the contour and casts them to a graph.

```
TObjArray *contours =
    (TObjArray*)gROOT->GetListOfSpecials()->FindObject( "contours");
TList *lcontour1 = (TList*)contours->At(0);
TGraph *gc1 = (TGraph*)lcontour1->First();
```

Question A7:

This loop fills the poly-marker using an infinite loop generating a flat distribution, but keeping only the points inside the cut.

```
while(1) {
    Double_t x = -4 + 8*gRandom->Rndm();
    Double_t y = -4 + 8*gRandom->Rndm();
    if (cutg->IsInside(x,y)) {
        pm->SetPoint(np,x,y);
        np++;
        if (np == npmax) break;
    }
}
```

Question A8:

```
root [] TFile f("hsimple.root");
root [] ntuple.Draw(">>elist", "pz > 10");
root [] TEventList *elist = (TEventList*)f->Get("elist");
root [] elist->Print("all"); //show the list of events
root [] elist->GetN(); //show the number of entries 148
```

Question A9:

To set the event list and draw px:

```
root [] ntuple->SetEventList(elist);
root [] ntuple->Draw("px");
```

The script:

```
{
  TFile f("hsimple.root");
  ntuple.Draw(">>elist", "pz > 10");
  TEventList *elist = (TEventList*)f->Get("elist");
  elist->Print("all"); //show the list of events
  elist->GetN(); //show the number of entries 148
  ntuple->SetEventList(elist);
  ntuple->Draw("px");
}
```

Question A10:

```
root [] htemp->GetRMS(); //shows 2.407
```

Question A11:

Run the tutorial h1draw.C, click on the top right pad of the canvas containing the Lego plot. This will select this pad as being the current pad. You can rotate the Lego using the left button. To find the current viewing angle theta, you can type the following command:

```
root [] gPad->GetTheta();
```

see next page for A12.

Question A12:

An example of script `hrandom1.C` is the following:

```
//----- hrandom.C
#include "TStopwatch.h"
#include "TRandom2.h"
#include "TRandom3.h"
#include "TH1.h"

void hrandom()
{
    // example of a script computing the CPU time to fill an histogram
    // with 3 random number generators.

    const Int_t nfills = 10000000;
    TStopwatch timer;

    // create an histogram and evaluate the time to fill nfills time
    TH1F h("h","h",100,0,1);
    Int_t i;
    timer.Start();
    for (i=0;i<nfills;i++) h.Fill(0.5);
    Double_t fillTime = timer.CpuTime();
    printf("Time for Fill      = %f seconds\n",fillTime);

    //using TRandom
    timer.Start();
    TRandom r1;
    for (i=0;i<nfills;i++) h.Fill(r1.Rndm());
    printf("Time for TRandom   = %f seconds\n",timer.CpuTime()-fillTime);

    //using TRandom2
    timer.Start();
    TRandom2 r2;
    for (i=0;i<nfills;i++) h.Fill(r2.Rndm());
    printf("Time for TRandom2 = %f seconds\n",timer.CpuTime()-fillTime);

    //using TRandom3
    timer.Start();
    TRandom3 r3;
    for (i=0;i<nfills;i++) h.Fill(r3.Rndm());
    printf("Time for TRandom3 = %f seconds\n",timer.CpuTime()-fillTime);
}
```

The output of the first session should be something like:

```
root [0] .x hrandom1.C
Time for Fill      = 17.750000 seconds
Time for TRandom   = 6.050000 seconds
Time for TRandom2 = 8.370000 seconds
Time for TRandom3 = 5.290000 seconds
```

The output of the second session should be something like:

```
root [0] .x hrandom1.C++  
Creating shared library /export/apps/staff/brun/root/./hrandom.so  
Note: operator new() masked 1c  
Note: operator delete() masked 1c  
Time for Fill      = 3.410000 seconds  
Time for TRandom   = 0.660000 seconds  
Time for TRandom2  = 4.220000 seconds  
Time for TRandom3  = 1.250000 seconds
```

Question A13:

An example of script `hrandom2.C` is the following:

```
//----- hrandom2.C
#include "TStopwatch.h"
#include "TRandom2.h"
#include "TRandom3.h"
#include "TH1.h"

void hrandom2()
{
    // example of a script computing the CPU time to fill an histogram
    // with 3 random number generators.

    const Int_t nfills = 10000000;
    TStopwatch timer;

    // create an histogram and evaluate the time to fill nfills time
    // change h to be a pointer and create it on the heap
    // change all operators of "h." to "h->"

    TH1F * h = new TH1F("h","h",100,0,1);
    Int_t i;
    timer.Start();
    for (i=0;i<nfills;i++) h->Fill(0.5);
    Double_t fillTime = timer.CpuTime();
    printf("Time for Fill      = %f seconds\n",fillTime);

    //using TRandom
    timer.Start();
    TRandom r1;
    for (i=0;i<nfills;i++) h->Fill(r1.Rndm());
    printf("Time for TRandom  = %f seconds\n",timer.CpuTime() - fillTime);

    //using TRandom2
    timer.Start();
    TRandom2 r2;
    for (i=0;i<nfills;i++) h->Fill(r2.Rndm());
    printf("Time for TRandom2 = %f seconds\n",timer.CpuTime()-fillTime);

    //using TRandom3
    timer.Start();
    TRandom3 r3;
    for (i=0;i<nfills;i++) h->Fill(r3.Rndm());
    printf("Time for TRandom3 = %f seconds\n",timer.CpuTime()-fillTime);

    // add a draw command
    h->Draw();
}
```

Session B

Question B1:

You can run the following session:

```
root [] TFile f("c1.root");
root [] c1->Draw();
root [] TH1F *hdmd = (TH1F*)c1->GetPrimitive("hdmd");
root [] TF1 *f5 = hdmd->GetFunction("f5");
root [] double hint = hdmd->Integral();
root [] double fint = f5->Integral(0.139,0.170);
root [] double ratio = fint/hint
      1.00072237393612740e-03
```


