CERN School of Computing 2009

17 August – 28 August 2009 in Göttingen, Germany

Data Analysis with ROOT

A. Heikkinen and I. Puljak

Electronic version of this document can be found at the CSC web portal →CSC Live →Handouts Update (http://csc.web.cern.ch/CSC/2009/This_year_school/Scientific_Programme/Programme_update/handouts_update.htm)

ROOT documentation is available at:
- Users guide (http://root.cern.ch/drupal/content/users-guide)
- Reference guide (http://root.cern.ch/drupal/content/reference-guide)
- Tutorials (http://root.cern.ch/root/html/tutorials/)

Note: Root v5.24.00 used for all exercises

Exercise 1

A. Visualization of data with ROOT

A1. 2D histograms drawing options (basic)

- *Code:* `$ROOTSYS/tutorial/hist/draw2dopt.C`
- *Tasks:*
    - Run the program by executing `.X draw2dopt.C`
        - Suggestions: Create new directory in your home directory and copy code to the new directory before running the code
            - `mkdir data_analysis`
            - `cd data_analysis`
            - `mkdir ex1`
            - `cd ex1`
            - `cp ROOTSYS/tutorial/hist/draw2dopt.C .`
    - Analyze the program
    - Change some drawing options
      (look at http://root.cern.ch/root/html/THistPainter.html for drawing options)
        - *Suggestions:*
            - try with LEGO2 and SURF options
            - suppress front and back box on LEGO and SURF options
            - add x label on the top side of the plot
            - add y axis on the right side of the plot
    - *Optional:* Find a problem with this macro and suggest a solution
        - *Hint:* there is a problem with the Landau distribution

A2. Animating projections (advanced)

- *Code:* **`$ROOTSYS/tutorial/hist/DynamicSlice.C`**
- *Tasks:*
  - Run the program by executing **`.X DynamicSlice.C`**
  - Analyze the program
  - Display values of errors on the 'mean' and the 'RMS' using the `gStyle->SetOptStat()` command and understand the difference between values of the 'RMS' and the 'error on the mean' (Lecture2::Slides28-31)
    - *Suggestion*: try the same for fitting values of the mean and sigma
  - Animate y projections too

B1. Generating random numbers

B1. 1D and 2D examples (basic)

- *Code:* **`$ROOTSYS/tutorial/hist/fillrandom.C`**
- *Tasks:*
  - Run the program by executing **`.X fillrandom.C`**
  - Analyze the program
  - Generate histograms with 100 times less and 100 times more random numbers and display all histograms in 2×2 canvas
  - Understand weighted and unweighted events (Lecture1::Slide45)
    - ***Unweighted:*** Generate N (=100000) random numbers from the TF1 function '`sqroot`', using the function TF1::GetRandom(). The numbers obtained (corresponding to events in the language of the MC event generation) are 'unweighted', i.e. all have the same weights = 1, but they are generated according to the specified function. Fill new histograms with obtained numbers (same binning as the histogram '`h1f`').
    - ***Weighted:*** Generate N uniformly distributed random numbers in the '`sqroot`' domain (here from 0 to 10), using Root random number generators (f.g. `TRandom3`). Then calculate the weight for each number (event) as the value of the function '`sqroot`' at that number. Fill three histograms: one with generated numbers only, one with weights and one with numbers weighted by their weights.
    - Draw all four histograms from the two previous steps in the 2×2 canvas and understand them.
  - *Note:* 2D example was used in exercise **A1**

B2. Multidimensional example (advanced)

- Code: **`$ROOTSYS/tutorial/foam/foam_kanwa.C`**
- *Tasks:*
  - Run the program by executing **`.X foam_kanwa.C`**
  - Analyze the program
    (Look at http://root.cern.ch/root/html/TFoam.html for the TFoam class description)
  - Extend the example to four dimensions
- *Optional:* Run and study **`tutorial/math/testrandom.C`**
  - *Note*: be patient, it takes some time to run

Exercise 2

A. Statistical distributions in ROOT

A1. PDFs and CDFs (basic)

- *Code:* `$ROOTSYS/tutorial/math/mathcoreStatFunc.C`
- *Tasks:*
  - Run the program by executing `.X mathcoreStatFunc.C`
  - Analyze the program
  - Change PDFs in the example to 1D functions
  - Add more functions in another canvas (Student, Beta, Breit-Wigner and Log-Normal)
  - Reproduce "*All roads lead to Rome*" at Lecture2::Slide20 (binomial, Poisson and Chi2 only)

A2. Central limit theorem (advanced)

- *Tasks:*
  - Write a macro demonstrating the ***Central limit theorem*** (Lecture2::Slide13)
    - *Hint:* take for example the uniform and/or the exponential distributions (Lecture2::Slide21)

B. Fitting with ROOT

B1. Understanding errors with Minuit (basic/advanced)

- *Code:* `$ROOTSYS/tutorial/fit/FittingDemo.C`
- *Tasks:*
  - Run the program by executing `.X FittingDemo.C`
  - Analyze the program
    - *Hint*: fitting options are available at http://root.cern.ch/root/html/TH1.html#TH1:Fit
  - Understand errors (Lecture2::Slide35)
    - Calculate errors with MINOS and compare them with default errors
    - *Note*:
      - by default errors are calculated by HESSE
      - be aware that the 'maximum likelihood' minimizes '-logL'
  - Compare fitting methods
    - Change to the likelihood method and compare with chi2 (default one)
    - Compare MINOS and HESSE errors here too
  - Display fitting information by using `gStyle->SetOptFit(1111)`
    - *Hint*: comment the line `histo->SetStats(0);`
    - Calculate and display `TMath:Prob(chi2,ndf)` and compare with the `Prob` value in the fitting display box (see Lecture2::Slide56)
  - Explore the fit panel: right click on the histogram → select "Fit Panel"
  - Optional (advanced): Understanding n-sigma countours
    - Run and analyze `$ROOTSYS/tutorial/fit/fitcont.C`
      - Estimate correlations coefficients from the plots
      - Find the right correlation coefficients from the fit (include the option "V" in the fit)

B2. Minuit2 (advanced)

- *Code:* **`$ROOTSYS/tutorial/fit/minuit2GausFit.C`**
- *Tasks:*
  - Run the program by executing **`.X minuit2GausFit.C`**
    - *Note*: be patient, it takes some times to run (~ 3 min)
  - Analyze the program
  - Redo the example **B1** with Minuit2
    - *Hint*: Include the statement (at the beginning of the macro) `TVirtualFitter::SetDefaultFitter("Minuit2");`

C. Introduction to RooFit (advanced)

The RooFit library provides a toolkit for modeling the expected distribution of events in a physics analysis. In case you don't have previous experience with the RootFit, before starting with these exercises we suggest to go through the Chapter 2 of the RooFit manual (ftp://root.cern.ch/root/doc/RooFit_Users_Manual_2.91-33.pdf).

C1. Modeling signal and background with RooFit (advanced)

- *Code:* **`$ROOTSYS/tutorial/roofit/rf101_basics.C`**
- *Tasks:*
  - Understand basics of RooFit (fitting, plotting, toy data generation)
    - Run and analyze **`rf101_basics.C`**

C2. Fitting with RooFit (advanced)

- *Code:* **`$ROOTSYS/tutorial/roofit/`**
- *Tasks:*
  - Understand minimization, error analysis, contour plots, saving results
    - Run and analyze **`rf601_intminuit.C`**
  - Understand profile likelihood
    - Run and analyze **`rf605_profilell.C`**
  - Understand Chi2 fit to unbinned dataset with y (and optionally x errors)
    - Run and analyze **`rf609_xychi2fit.C`**

C3. Monte Carlo integration (advanced)

- *Code:* **`$ROOTSYS/tutorial/roofit/rf315_projectpdf.C`**
- *Tasks:*
  - Understand marginalization of multi-dimensional PDFs through the Monte Carlo integration
    - Run and analyze **`rf315_projectpdf.C`**