Exercises for the lecture „Moderne Methoden der Datenanalyse”

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

Apr 27 2010

Introduction — getting started

- The web page
  `http://www.physi.uni-heidelberg.de/~schiller/stat10`
  contains information, material and links for this course. You might want to set a bookmark to this page in your browser.
- It is assumed that you are familiar with the basics of Unix, emacs (or any other editor) and C++. Some links are provided on the web page to refresh this knowledge.
- It is recommended to create a subdirectory (with `mkdir`) in your home directory for this course.
- The programme we will use during this course is called ROOT. ROOT is started by typing `root` at the shell prompt. To quit root type `.q` (note the dot!) at the root prompt. A macro, e.g. with file name `macro.C`, can be executed in root by `.x macro.C`. To load this macro type `.L macro.C`. After a macro is loaded, functions defined in this macro can be called by simply typing the function name (with parentheses and arguments). Add a `+` after the file name, if you want to compile the macro, e.g. `.x macro.C+` or `.L macro.C+`. The programming language is C++.
- Feel free to ask questions and discuss problems and solutions with the tutor(s), your partner and other students.

Exercise 1: Root

- **Exercise 1.1**
  Open the web page
  `http://www.physi.uni-heidelberg.de/~schiller/stat10/ex01/root-tut.pdf`
  in a browser. You can find there a tutorial for root. Work through this tutorial step-by-step. Note also the summary of root commands in
  `http://www.physi.uni-heidelberg.de/~schiller/stat10/ex01/usefulROOT.pdf`.
  This can be helpful for you for the following exercises.

- **Exercise 1.2**
  Write a hello world macro, i.e. a macro that prints "Hello World!" on the screen. Write at the beginning:

```
// include header files needed for compilation
#include <iostream> // valid in each macro with text
    input/output
```

- **Exercise 1.3**
  Write a macro that takes two real numbers as arguments, prints whether the first or the second one is larger, and returns the absolute difference of the two numbers. Tip: Declare the function as double and return the absolute difference.

- **Exercise 1.4**
  Have a look at the macro `random.C` and run it. It creates a histogram, fills it with N Gaussian distributed random numbers (`gRandom->Gaus()`) with mean=0 and sigma=1 and draws the histogram. N is an argument of the macro.

- **Exercise 1.5**
  Change the macro from exercise 0.4 so that the histogram is written to a file. Write at the beginning of the function:

```
// Open a file with name "random.root"
// The option "RECREATE" causes the file to be
    overwritten if it already
// exists
TFile* file = TFile::Open("random.root", "RECREATE");
```

  and at the end:

```
// Write the histogram to the file and close the file.
hRandom->Write();
delete file;
```

  if `hRandom` is the name of the histogram.

- **Exercise 1.6**
  Add to the macro `fit.C` the missing parts so that it reads the histogram from the file created in exercise 0.4 and displays it.

- **Exercise 1.7**
  Add a fit of a Gaussian function to the histogram from Exercise 0.6.

- Make the plot nicer. Use filled boxes with error bars for the histogram and a red line with thickness 3 for the fitted function. Label the axes "x" and "Entries". Display only mean, rms, fit probability and fitted parameters with errors in the statistics box.

- Make a ps file of the plot created in the previous exercise.