Goal of today's class

Introduction into data analysis package ROOT
(available free of charge on http://root.cern.ch):

- Start and quit program
- Create functions and histograms
- Graphical representation of functions and histograms
- Using macros
- C++ intermezzo
- Usage of random number generators

Detailed documentation available on
http://root.cern.ch/root/doc/RootDoc.html

What this class cannot do ...

ROOT uses C++ syntax. A comprehensive introduction into the C++ programming language is *not* possible within the framework of this class. We will restrict ourselves to the most important C++ knowledge which is needed to write simple ROOT macros.

More information will be provided if needed in the following classes ...

Let's go ...

Start ROOT:

```
[wiene@haco72:~] root
  ****************************************
  *                                      *
  *        W E L C O M E  to  R O O T     *
  *                                      *
  *   Version   4.00/08      28 July 2004  *
  *                                      *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch          *
  *                                      *
  ****************************************

FreeType Engine v2.1.3 used to render TrueType fonts.
Compiled for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.15.138, May 23 2004
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
```

Quit the program:

```
root [1] .q
[wiene@haco72:~]
```

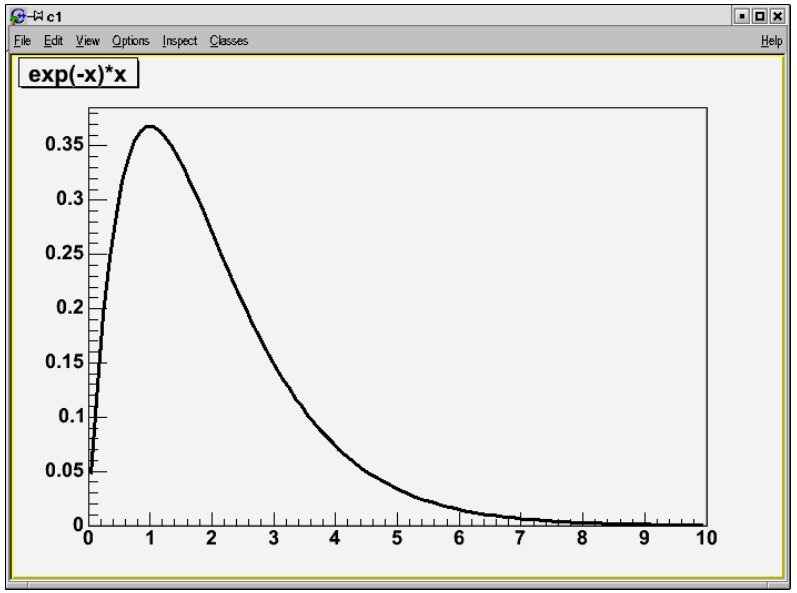
Defining and plotting functions

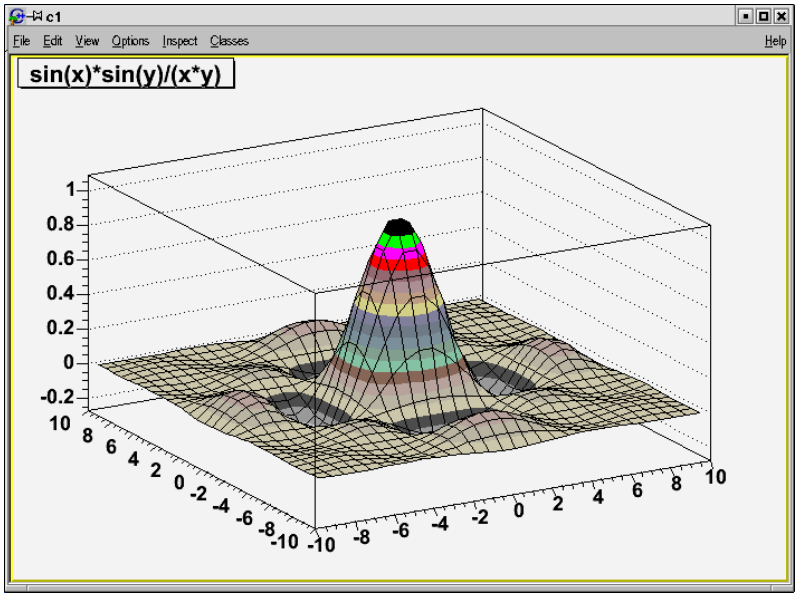
One-dimensional function:

```
root [0] TF1 f1("func1","exp(-x)*x",0,10)
```

arbitrary fct. name — formula — abscissa range

```
root [1] f1.Draw()
```





Two-dimensional function:

```
root [2] TF2 f2("func2","sin(x)*sin(y)/(x*y)",-10,10,-10,10)
root [3] f2.Draw("surf1")
```

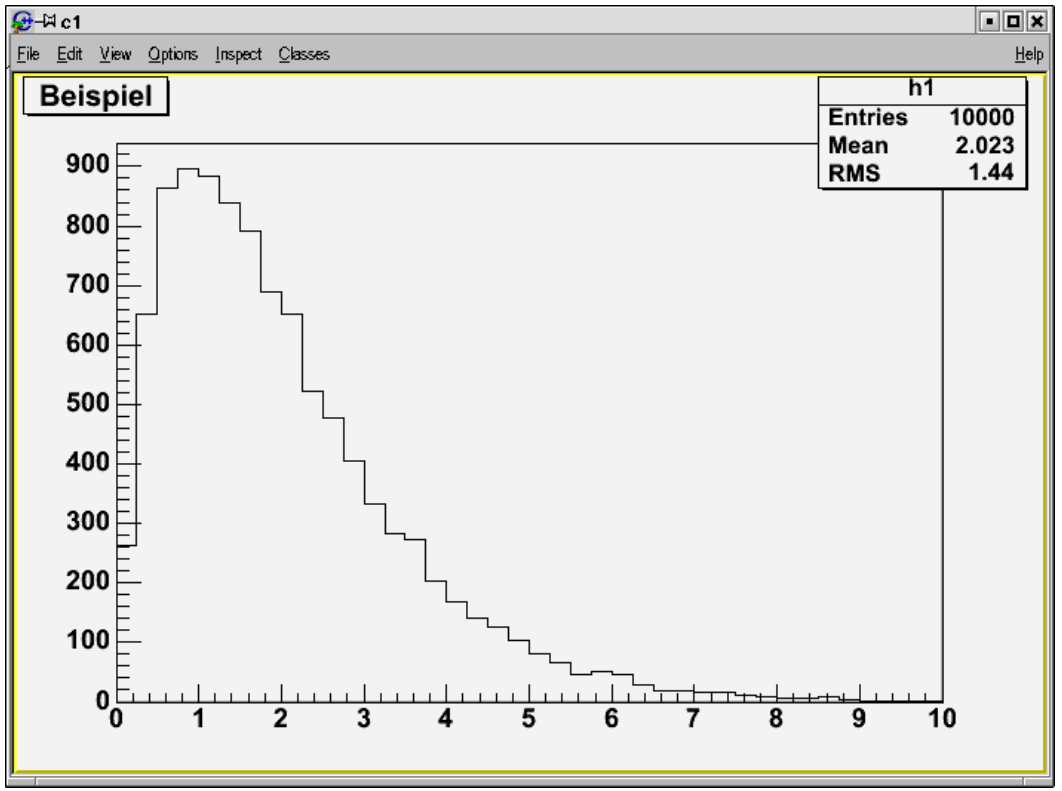
plotting option

Defining and plotting histograms

```
root [0] TH1F h1("h1","myexample",40,0,10)
```

arbitrary hist. name — hist. title — number of bins — Binning range

```
root [1] h1.Draw()
root [2] TF1 f1("func1","exp(-x)*x",0,10)
root [3] h1.FillRandom("func1", 10000)
root [4] h1.Draw()
```



Similar: Definition of two and three dimensional histograms

Macros

ROOT commands can be stored in an ASCII file from which ROOT can read them in and execute them:

```
{
    TH1F h1("h1",”example",40,0,10);
    TF1 f1("func1","exp(-x)*x",0,10);
    h1.FillRandom(“func1”, 10000);
    h1.Draw();
}
```

Note parentheses and semicolons

Execute macro in ROOT:

```
root [0] .x histo.C
```

name of file which contains commands

C++ intermezzo

Intermezzo: Output in C++

“Hello world” in ROOT:

```
root [0] cout << “Hello world” << endl;
```

Output command

Output text

Newline

Intermezzo: Arithmetic operations in C++

A C++ type must be assigned to each used variable.
The most important types are:

`int, float, double, char, unsigned int, ...`

The most important arithmetic operators in C++ are:

+ (addition), - (subtraction), * (multiplication), / (division), % (modulo),

++ (increment by 1), -- (subtract 1)

Examples:

```
int a = 1;
int b = a;
int c;
c = a + b;
c++;
cout << ”c = “ << c << endl;
```

Intermezzo: if-else if-else

Comparison operators in C++:

==, !=, <, <=, >, >=

Example:

```
{
    int a = 4;
    if (a == 0) {
        cout << “a is zero” << endl;
    }
    else if (a < 0) {
        cout << “a is negative” << endl;
    }
    else {
        cout << “a is positive” << endl;
    }
}
```

Intermezzo: loops

Often the same commands must be repeated many times. Loops simplify coding work:

```
{
    int sum = 0;
    for (int i=0; i<100; i++) {
        sum += i;
    }
}
```

short for: `sum = sum + i;`

Commands in red parenthesis are executed 100 times

Back to ROOT

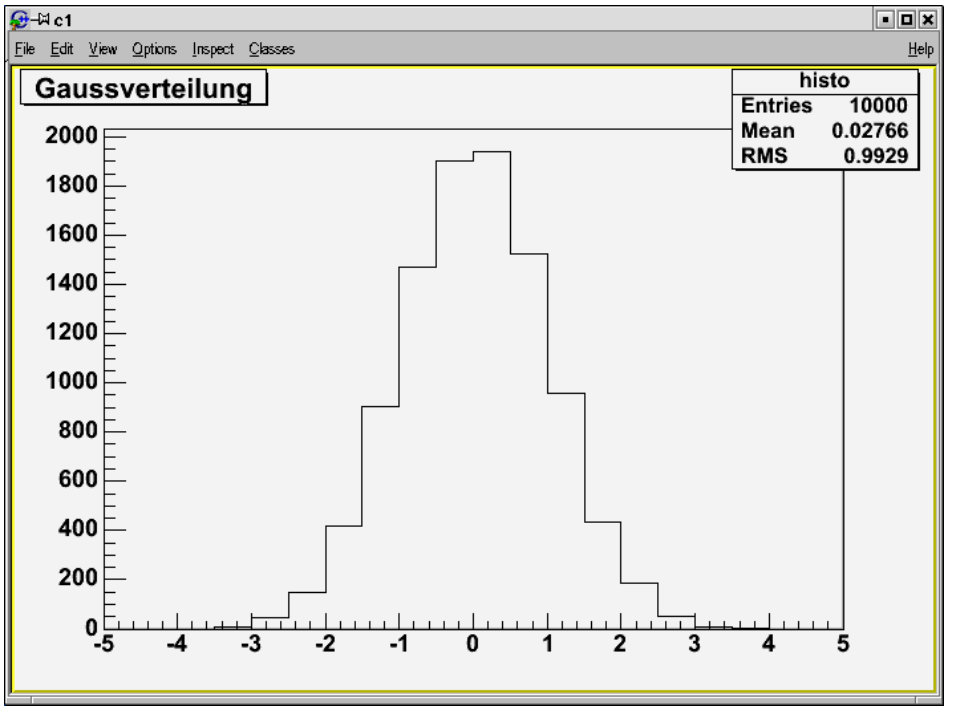
Using random number generators

ROOT provides various different random number generators. An example:

```
{
    TH1F histo("histo","Gauss",20,-5,5);
    for (int i=0; i<10000; i++) {
        histo.Fill(gRandom.Gaus(0,1),1);
    }
    histo.Draw();
}
```

Gauss with $\mu=0$ und $\sigma=1$

weight



Examples for different random number distributions:

```
gRandom.Rndm()
gRandom.Exp(tau)
gRandom.Uniform(min, max)
```

and many more

Exercise

Consider the random variable

$$y = \sum_{i=1}^{n} x_i^2$$

where x is distributed according to a Gaussian with mean 0 and variance 1. Carry out 10 000 toy experiments for a few values of n in the range from 1 to 10 and plot the distribution of y in form of a histogram. According to which probability density function is y distributed?