The Abdus Salam International Centre for Theoretical Physics

2057-10

First Workshop on Open Source and Internet Technology for Scientific Environment: with case studies from Environmental Monitoring

7 - 25 September 2009

Mathematical Software

Paul Bartholdi

Obeservatoire de Geneve
Chemin des Maillettes 51
CH-1290 Sauverny
Switzerland

Mathematical Software

Paul Bartholdi

OpenSource Workshop - 2009 – ICTP - Trieste

Goals

In these lectures will be presented many Open Source Softwares useful to control instrument and analyze resulting data, in particular:

- scientific libraries (mathematics, statistics)
- interactive programs (mathematics, statistics, graphics)

Then I will show some examples using interactive programs.

Choices

We have a set of open choices:

- develop programs using available libraries
  - in which language (Fortran, C, C++, Java)?
  - with which library (plenty available)?

  Slow development, little interactivity, very fast execution

- develop scripts for interactive program
  - which program (sm, gnuplot, octave, scilab, R)?

  Fast interactive development, slower execution

Criteria - what is the most important

- Speed in development
- Speed at execution
- Interaction with data bases
- production of good graphics - tables...

How we discovered the first extrasolar planet (1995)

- instrument and telescope fully computer controlled;
- network of real-time, non real-time and dB machines;
- data analyzed immediately;
- results visualized interactively in real time.

Our American competitors had larger telescopes, better instruments, and observed the planet one year before us, but kept unanalyzed raw data on magnetic tapes in a drawer for later processing.

What do we need?

- good understanding of the mathematics underneath;
- nature of the experiment;
- detail understanding of the instrument;
- underlying models and environment.

Open Source Software

Mathematical software started Open Source in the sixties (EISPACK is the best prototype, base for LAPACK, MATLAB , Octave, Scilab etc.)

- easily available for small or big applications
- developed and tested over many years by many users
- very robust, very efficient
- easy (user oriented, not commercial) to install

but

- lack of glamorous appearance
- Manuals not always very good

Our Responsability

Their is no free beer !

We are responsible with Open Source Software to

- share experience with others in using them
- take part in their development, improvement etc.

in a cooperative way.

Scientific Libraries - Introduction

The first large Open Source project was probably the EISPACK library, developed initially in England (James Wilkinson) and then at Argone National Laboratory (Jack Dongarra). Soon came LINPACK and MINPACK. The core of most modern Numerical Libraries still contains EISPACK, including commercial ones as MATLAB .

The goal was clear:

- use the best algorithms
- share development and testing
- make it widely available to receive feedback for improvements

Here are some of the best and most useful Open Source ones.

`netlib` web site

`http://www.netlib.org/`

This is by far the best site to start with.
It contains references to almost all Mathematical softwares available, including search commands.

This is also a free and cooperative work!

GAMS, the Guide to Available Mathematical Software

```
http://gams.nist.gov/HotGAMS/
http://math.nist.gov/
```

is a cross-index and virtual repository of mathematical and statistical software useful in science and engineering.

`Statlib` web site

```
http://lib.stat.cmu.edu/
```

This is by far the best site to start with for statistics.
It contains references to almost all Statistical softwares available, including search commands.

Linear Algebra

Linear Algebra is a big eater of CPU time. It is also an Art, a place where high efficiency can be attained and numerical difficulties best avoided.

The main libraries are:

- LAPACK (CLAPACK. PLAPACK etc.)
- LINPACK for linear equations
- BLAS for highly optimized, cpu dependant, kernel operations, can be used by EISPACK etc.

Numerical Recipes: The Book

Numerical Recipes, The Art of Scientific Computing
William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery
Cambridge University Press

is a good practical introduction to the main problems encountered in scientific programs.

Each subject has a short but complete introduction, followed by the code in either Fortran, C or C++ depending on the edition.

The choice of algorithms, their relative efficiency and overall quality have been questioned (see `http://en.wikipedia.org/wiki/Numerical_Recipes`)

Numerical Recipes: The Library

Exists for Fortran (77 and 90), Pascal, C and C++

- straight forward implementation, not always very carefully done
- useful when nothing else is available
- includes tests for each routine, with data and results

The source code, test data and results are available in computer readable form. They are not free neither "Open". (see `http://www.nr.com/`)

GNU Scientific Library

```
http://www.gnu.org/software/gsl/gsl.html
```

The GNU Scientific Library (GSL) is a numerical library for C and C++ programmers. It is free software under the GNU General Public License.

The library provides a wide range of mathematical routines such as random number generators, special functions and least-squares fitting. There are over 1000 functions in total with an extensive test suite.

Subject Areas Covered by GSL

- Complex Numbers
- Special Functions
- Permutations
- BLAS Support
- Eigensystems
- Quadrature
- Quasi-Random Sequences
- Statistics
- N-Tuples
- Simulated Annealing
- Interpolation
- Chebyshev Approximation
- Discrete Hankel Transforms
- Minimization
- Physical Constants
- Discrete Wavelet Transforms
- Roots of Polynomials
- Vectors and Matrices
- Sorting
- Linear Algebra
- Fast Fourier Transforms
- Random Numbers
- Random Distributions
- Histograms
- Monte Carlo Integration
- Differential Equations
- Numerical Differentiation
- Series Acceleration
- Root-Finding
- Least-Squares Fitting
- IEEE Floating-Point
- Basis splines

GSL Related Packages

ATLAS a portable self-optimising BLAS library with CBLAS interface

GLPK GNU Linear Programming Kit

FFTW Large-scale Fast Fourier Transforms

A very large set of extensions and applications exists for GSL, see the main gnu site for details.

Java Scientific Library

Many projects started in the late nineties, very few available now.

JLAPACK, using a translator from Fortran to Java.

Look at

```
http://math.nist.gov/jnt/
```

Most user prefer a non portable direct link to the original Libraries.

Python Scientific Library

Python seems to be almost ideal for interactive analysis:

- is fully interactive, yet execute very fast
- has access to most numerical and graphical libraries
- tools for integrating C/C++ and Fortran code
- tools for distributed/parallel processing

The best place to look at:

```
http://numpy.scipy.org/
http://www.scipy.org/
http://pygsl.sourceforge.net/
```

interface to the GNU Scientific library

Statistical Library

All scientific libraries contains routines for statistics.

For particular needs, look at:

```
http://lib.stat.cmu.edu/
```

which contains references to almost all statistical softwares available.

Notice that many statistical softwares are not free, neither "Open Source", but some of the best are.

The R Library

The (R) project stands alone as a library primarily oriented toward statistical problems. It is both a library and an interactive environment.

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

It was designed as a clone of (S), with some subtle differences.

see: `http://CRAN.R-project.org/`

Parallel Linear Algebra

```
http://www.cs.utexas.edu/ plapack/
```

PLAPACK is a library infrastructure for the parallel implementation of linear algebra algorithms and applications on distributed memory supercomputers such as the Intel Paragon, IBM SP2, Cray T3D/T3E, SGI PowerChallenge, and Convex Exemplar.

Fast Fourier Transform

`http://www.fftw.org/`

Fastest Fourier Transform in the West

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DCT/DST).

It uses a pre-run optimization phase.

A version for distributed parallel processing using mpi is also available.

(Bell) PORT Library

`http://www.bell-labs.com/project/PORT/`
PORT stands for Portable, Outstanding, Reliable, and Tested.

The PORT Mathematical Subroutine Library (third edition, 1997) is a collection of Fortran 77 routines that address many traditional areas of mathematical software, including approximation, ordinary and partial differential equations, linear algebra and eigensystems, optimization, quadrature, root finding, special functions, and Fourier transforms, but excluding statistical calculations.

Good user manual (pdf). Non-commercial use is free, but needs a license agreement.

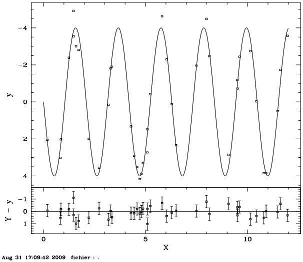
PORT is part of the set of Lucent libraries

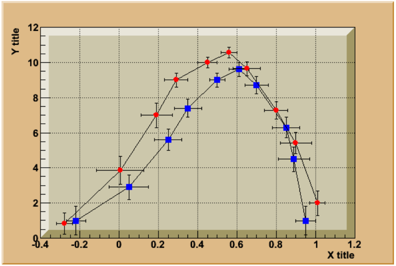
Lucent Library

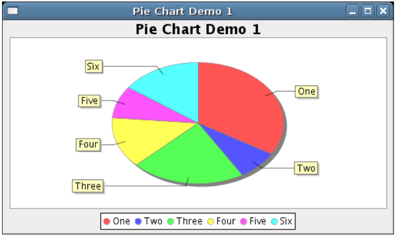
`http://www1.bell-labs.com/topic/swdist/`

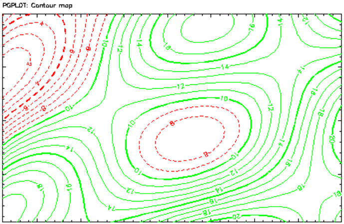
Lucent (earlier Bell Labs) provide a set of libraries, among them good scientific ones as PORT, BL-QMR, eigensolve, fptest, IQP. Non-commercial use is free, but needs a license agreement.

Graphics library









SuperMongo (SM)

SuperMongo (SM) is both a graphic library, with interfaces for Fortran, C and Python, and an interactive graphical data analysis program.

It runs under all Unix-like systems, as well as VMS and Windows, and interfaces with most graphical output, terminals, printers and files alike.

It is widely extensible through a macro language defined in `yacc` (Yet an other Compiler Compiler). The same is true as any new output device can be added in a very simple way.

It is copyrighted and not free in general. All the sources are Open Source.

pgplot

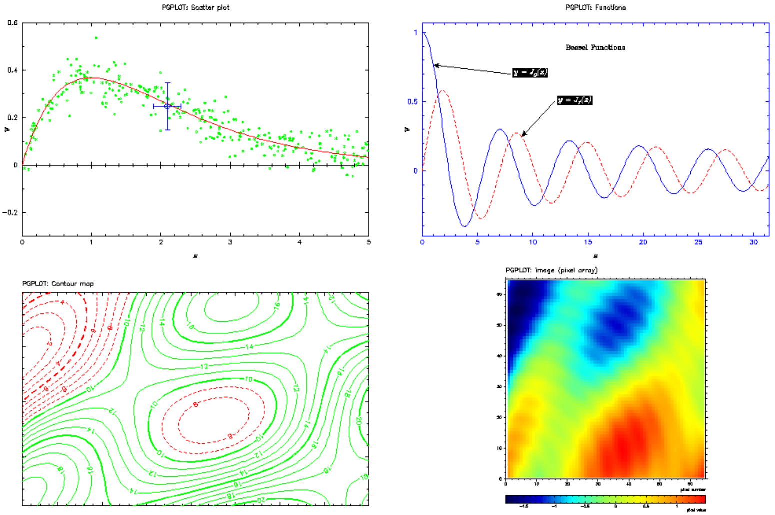
`http://www.astro.caltech.edu/ tjp/pgplot/`

The PGPLOT Library is a Fortran- or C-callable, device-independent graphics package for making simple scientific graphs. It is intended for making graphs of publication quality with minimum effort. For most applications, the program can be device-independent, and the output can be directed to the appropriate device at run time.

The PGPLOT library consists of two parts: a device-independent one and a set of ‘device handlers’ for output on various terminals and printers, or for PostScript and GIF files.

PGPLOT is not public-domain software. However, it is freely available for non-commercial use.

pgplot examples

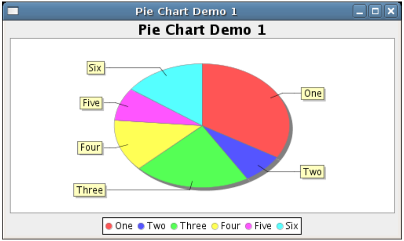


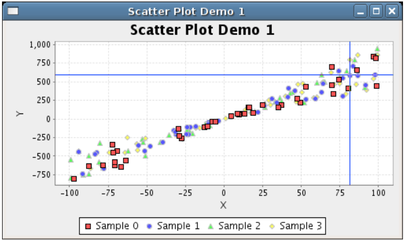
Jfreechart

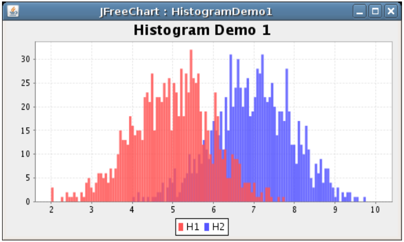
JFreeChart is a free 100 % Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

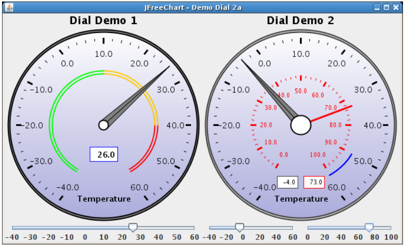
- a consistent and well-documented API, supporting a wide range of chart types;
- a flexible design that is easy to extend, and targets both server-side and client-side applications;
- support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
- JFreeChart is "open source" under of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

Jfreechart - Examples









Matplotlib (Python)

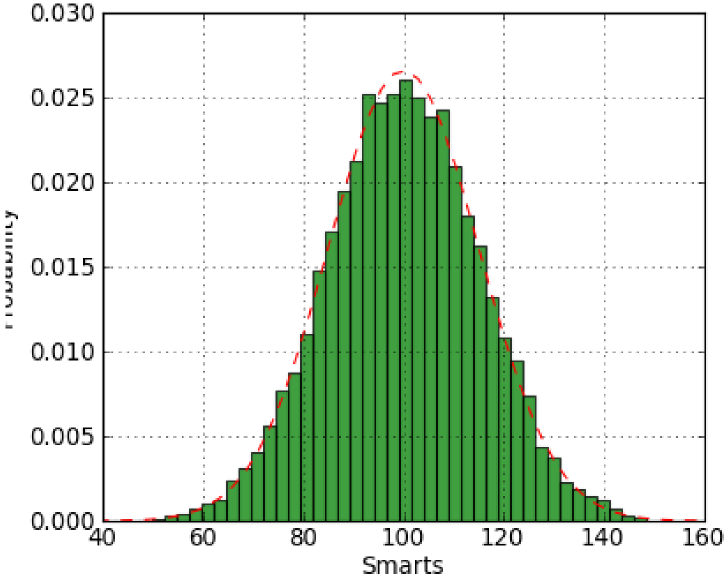
matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats. matplotlib can be used in python scripts, the python and ipython shell (ala matlab or mathematica), web application servers, and six graphical user interface toolkits.

matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code. For a sampling, see the screenshots, thumbnail gallery, and examples directory

Matplotlib - example

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

fig = plt.figure()
ax = fig.add_subplot(111)

n,bins,patches = ax.hist(x,50,normed=1,facecolor='green',alpha=0.75)

bincenters = 0.5*(bins[1:]+bins[:-1])
y = mlab.normpdf( bincenters, mu, sigma)
l = ax.plot(bincenters, y, 'r--', linewidth=1)

ax.set_xlabel('Smarts')
ax.set_ylabel('Probability')
ax.set_xlim(40, 160)
ax.set_ylim(0, 0.03)
ax.grid(True)

plt.show()
```

Matplotlib - example



Data Analysis - Introduction

To analyze data, we need:

- good visualisation tools,
- high interactivity,
- access to other/old data from data Base,
- good modeling facilities.

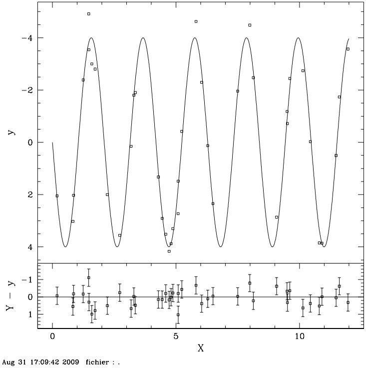
SuperMongo

It is widely extensible through a macro language defined in `yacc` (Yet an other Compiler Compiler). The same is true as any new output device can be added in a very simple way.

The compiled kernel contains the macro interpreter and all basic routines. It is extended with a set of predefined macros to which the user can add his own more specialized higher level ones.

SuperMongo example

```
# test Diagf
set x = 0,12,.01
set y = 4 * sin( 3 * x )
set X = 12 * random(40)
set Y = 4 * sin( 3 * X ) \
       + 0.5 * gaussdev(40)
set EB_Y = 0*X + 0.5
ptype 4 0
Winit 1 1
expand 1.2

Diagf x y X Y -y
```



Octave

Octave is essentially a clone of MATLAB . MATLAB started as an interactive interface to EISPACK. Then graphics, all kind of functions, including statistics, and a macro language were added.

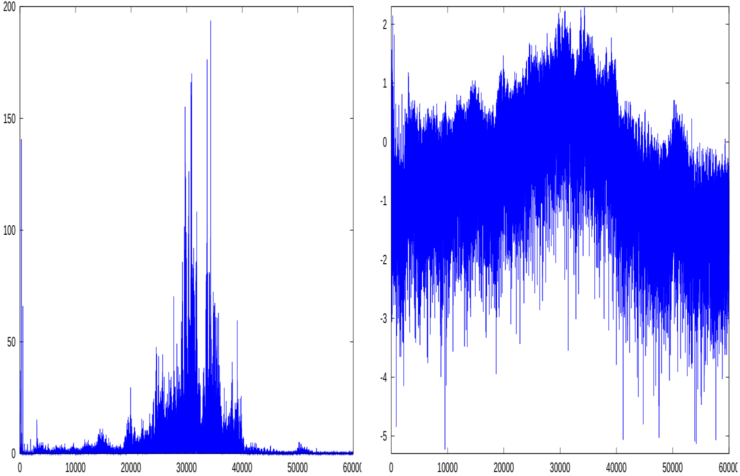
Le macro language of Octave has some slight differences to the MATLAB original. In most cases it is easy to move from one to the other.

The graphics are provided by gnuplot (see below) and are so very portable.

Octave - examples

```
z=wavread("MyVoice.wav")  # some sound file
zl=z(:,1)                 # left channel
zr=z(:,2)                 # right channel
zla=zl(1:262144,1)        # subpart (power of 2)
zra=zr(1:262144,1)        #
bl=fft(zla)               #
br=fft(zra)               # fft of each channel
s5=bl.*conj(bl)           # power spectrum
plot(s5)                  # linear plot
plot(log10(s5))           # logarithmic plot
size(s5)                  #
history -w SND1-2.log     # write history to a file
exit                      # ... and quit
```

Graphics produced



Gnuplot

Gnuplot is a portable command-line driven interactive data and function plotting utility for UNIX, IBM OS/2, MS Windows, DOS, Macintosh, etc.

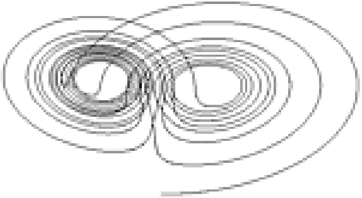
The software is copyrighted but freely distributed (i.e., you don't have to pay for it). It was originally intended as to allow scientists and students to visualize mathematical functions and data. It has grown to support many non-interactive uses, including web scripting and integration as a plotting engine for third-party applications like Octave and R.

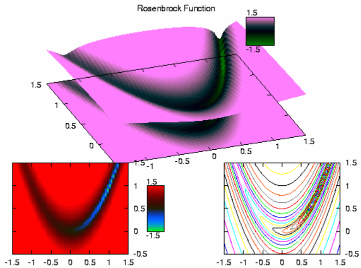
gnuplot 2

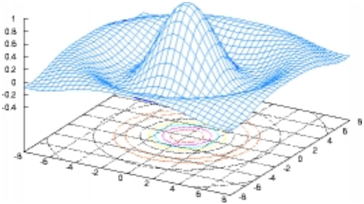
Gnuplot supports many types of plots in either 2D and 3D. It can draw using lines, points, boxes, contours, vector fields, surfaces, and various associated text.

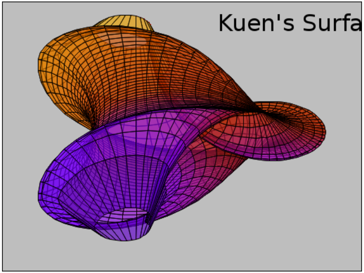
Gnuplot supports many different types of output: interactive terminals, direct output to pen plotters or modern printers, and output to many file formats (eps, fig, jpeg, LaTeX, metafont, pbm, pdf, png, postscript, svg, ...). Gnuplot is easily extensible to include new output modes. Recent additions include interactive terminals based on aquaterm (OSX) and wxWidgets (multiple platforms).

gnuplot examples









ROOT

Typically developed by scientists for scientists

ROOT is an object-oriented program and library developed by CERN. It was originally designed for particle physics data analysis and contains several features specific to this field, but it is also commonly used in other applications such as astronomy and data mining.

ROOT is both a library, all in C++, and a set of very powerful interactive program for data analysis. It also interfaces in both directions with Python and Ruby.

ROOT uses the GSL library (GPL) and is itself under LGPL.

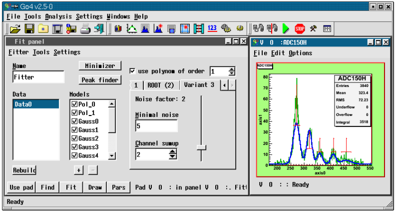
ROOT - example

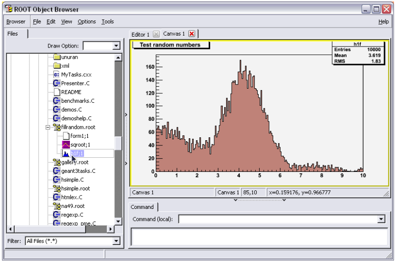
```
Root > myfit->SetParName(0,"c0");
Root > myfit->SetParName(1,"c1");
Root > myfit->SetParName(2,"slope");
Root > myfit->SetParameter(0, 1);
Root > myfit->SetParameter(1, 0.05);
Root > myfit->SetParameter(2, 0.2);
```

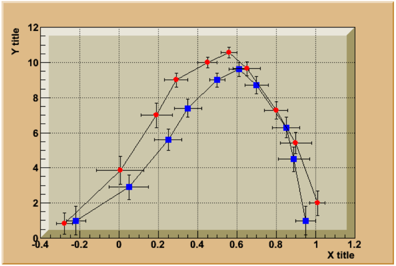
We are now ready to fit:

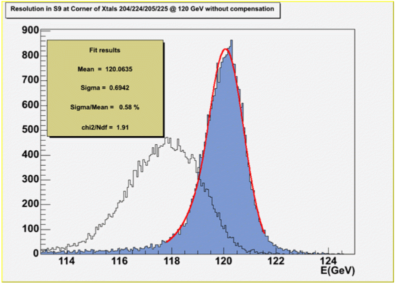
```
Root > hist->Fit("myfit");
```

ROOT - example









Sage Project

Sage is a software application which covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics and calculus. It has a wide range area of application, including Engineering, Science etc.

Initial goals of creating an "open source alternative to Magma, Maple, Mathematica, and MATLAB.

All written in Python (+Cython) and present a uniform interface to other packages like Octave, GAP etc.

Sage content

Algebra	GAP, Maxima, Singular
Algebraic Geometry	Singular
Arbitrary Precision Arithmetic	GMP, MPFR, MPFI, NTL
Arithmetic Geometry	PARI, NTL, mwrank, ecm
Calculus	Maxima, SymPy, GiNaC
Combinatorics	Symmetrica, Sage-Combinat
Linear Algebra	Linbox, IML
Graph Theory	NetworkX
Group Theory	GAP
Numerical computation	GSL, SciPy, NumPy, ATLAS
Other packages in Command line	IPython
Database	ZODB, Python Pickles, SQLite
Graphical Interface	Sage Notebook, jsmath
Graphics	Matplotlib, Tachyon3d, GD, Jmol
Interactive programming language	Python
Networking	Twisted

Statistics - Introduction

Gnuplot. SM, Octave provide random number generator (different distributions), linear and nonlinear fitting, as well as most nbasic statistics: mean, variance, correlation, t-test, Kolmogorov, etc.

For more advanced statistics, **R** is a better tool. It also provides graphics (using gnuplot) and algebra as Octave, SM or gnuplot.

R

R is an integrated suite for data manipulation, calculation and graphical display.

It has:

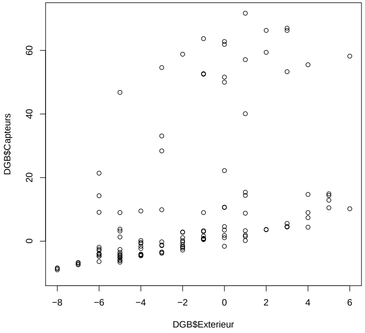
- effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hardcopy,
- a well developed, simple and effective programming language (called 'S') which includes conditionals, loops, user defined recursive functions and input/output facilities.

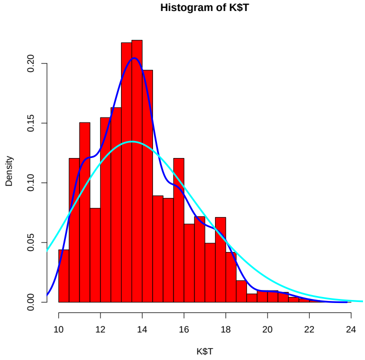
Indeed most of the system supplied functions are themselves written in the S language.

R examples

```
# read csv file
help(read.csv)
DGB <- read.csv("DGB.csv")
summary(DGB)
plot(DGB$Exterior, DGB$Captors)

K=scan("THP.txt",list(N=0,D="",TIME="",T=0,H=0,D=0),\
       skip=8)
summary(K)
summary(K$T)
hist(K$T, seq(10,24,.5), prob=TRUE)
lines(density(K$T, bw=2))
lines(density(K$T, bw=1))
lines(density(K$T, bw=.5))

q()
```

R - Examples Graphs





R - Numerical output

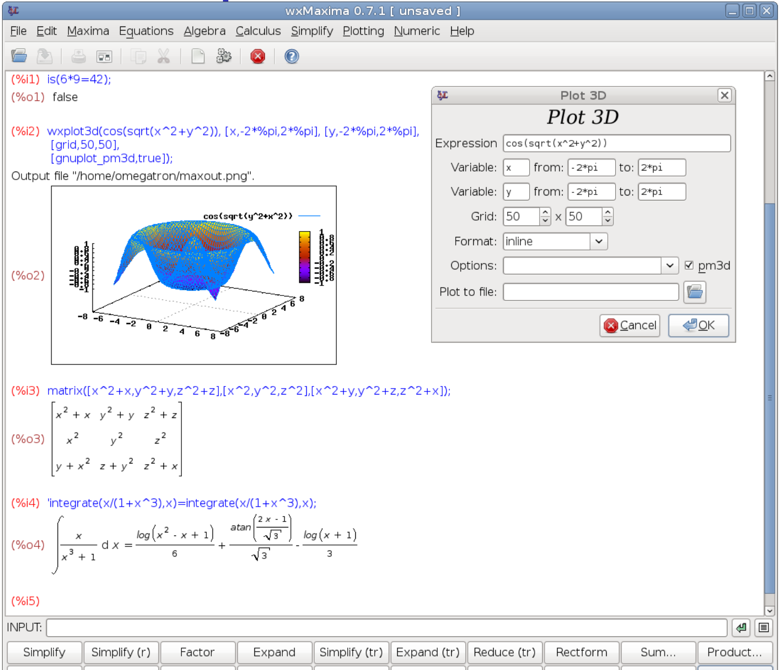
```
> summary(DGB)
  TCaptorKitchen    Exterior          Captors            TIR       
 Min.   :17.50   Min.   :-8.000   Min.   :-10.70   Min.   :20.50  
 1st Qu.:19.00   1st Qu.:-5.000   1st Qu.: -3.50   1st Qu.:44.60  
 Median :20.00   Median :-2.000   Median :  1.30   Median :51.00  
 Mean   :19.95   Mean   :-1.770   Mean   : 11.30   Mean   :49.70  
 3rd Qu.:20.77   3rd Qu.: 1.000   3rd Qu.: 13.05   3rd Qu.:55.85  
 Max.   :23.30   Max.   : 6.000   Max.   : 71.70   Max.   :66.00  
 NA's   :49.00   NA's   :49.000                                   

> summary(DGB$Exterior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  -8.00   -5.00   -2.00   -1.77    1.00    6.00   49.00 
```

Symbolic Algebra - Maxima

Symbolic Algebra

Maxima is a complete computer algebra system based on a 1982 version of Macsyma. It is written in Common Lisp and runs on all POSIX platforms such as Mac OS X, Unix, BSD, and Linux as well as under Microsoft Windows.

Maxima - example



$$\begin{bmatrix} x^2+x & y^2+y & z^2+z \\ x^2 & y^2 & z^2 \\ y+x^2 & z+y^2 & z^2+x \end{bmatrix}$$

$$\int \frac{x}{x^3+1}\,dx = \frac{\log\left(x^2-x+1\right)}{6} + \frac{\operatorname{atan}\left(\frac{2x-1}{\sqrt{3}}\right)}{\sqrt{3}} - \frac{\log(x+1)}{3}$$

Main Links

```
http://www.nr.com/
http://www.netlib.org/
http://lib.stat.cmu.edu/
http://www.gnu.org/software/gsl/gsl.html
http://math.nist.gov/jnt/
http://www.bell-labs.com/project/PORT/
http://www.gnuplot.info/
http://www.astro.caltech.edu/~tjp/pgplot/
http://CRAN.R-project.org/
http://www.jfree.org/jfreechart/
http://matplotlib.sourceforge.net/
http://www.sagemath.org/
http://maxima.sourceforge.net/
http://en.wikipedia.org/wiki/\
       List_of_open_source_software_packages
```