ROOT Statistical Software

Lorenzo Moneta (CERN, PH-SFT) on behalf of the ROOT Math Work Package (R. Brun, A. Kreshuk, E. Offermann + many others contributors)
Outline

- ROOT Statistical classes
  - statistical functions
  - random numbers
  - data analysis classes and their visualization
  - fitting
  - confidence levels (limits settings)
  - smoothing
  - robust estimators
  - multi-variate methods
- Organization of Math and Statistical Libraries
- Plans and new developments
- Conclusions
Statistical Functions

- Function evaluations in \textit{TMath} namespace
- Mathematical libraries (\textit{MathCore}/\textit{MathMore})
  - major special functions
    - gamma, beta, errorf, bessel, hyperg., Legendre, elliptic int., etc.
  - statistical functions (with a coherent naming scheme)
    - probability density functions (pdf)
    - cumulative distributions (lower tail and upper tail)
    - inverse of cumulative distributions (quantiles)
      - Example for $\chi^2$ distribution:
        
        \begin{verbatim}
        chisquared_pdf
        chisquared_cdf, chisquared_cdf_c,
        chisquared_quantile, chisquare_quantile_c
        \end{verbatim}
  - provide all major statistical distributions
    - normal, lognormal, Landau, Cauchy, $\chi^2$, gamma, beta, F, t, poisson, binomial, etc.
Numerical Algorithms

- **MathMore**: C++ interface to GNU Scientific Library (GSL) algorithms and functions
- Numerical algorithms for 1D functions:
  - **Numerical Derivation**
    - central evaluation (5 points rule) and forward/backward
  - **Numerical Integration**
    - adaptive integration for finite and infinite intervals
  - **Root Finders**
    - bracketing and polishing algorithms using derivatives
  - **Minimization**
    - Golden section and Brent algorithm
  - **Interpolation**
    - linear, polynomial, cubic and Akima spline
  - **Chebyshev polynomials** (for function approximation)
- Complement the various algorithms existing in **TF1** class
Random Number Generators

- **TRandom3**: Mersenne-Twister generator
  - fast and good pseudo-random quality
  - very long period, $\sim 10^{6000}$, large state (624 words)

- **TRandom1**: RanLux generator
  - proven random quality, but slower

- **TRandom2**: TausWorthe generator from L’Ecuyer
  - fast generator based only on 3 words (period $\sim 10^{26}$)

- **TRandom**: linear congruential generator
  - maintain only for backward compatibility
  - bad quality although improved recently

- Generators can be seeded with an **UUID** (unique 128 bit number)
  - convenient when running parallel jobs on the Grid
Random Number Distributions

- Methods available in the class `TRandom` for sampling according to some standard distributions
  - improved algorithms for generating Gaussian and Poisson random numbers
- Approximate (but efficient) sampling for user functions via `TF1::GetRandom`
- Introduced interface to `UNU.RAN`
  - package for generating non uniform random numbers
    - from J. Leydold et al, Vienna TU.
  - various methods for generic 1D, multi-dim., discrete and empirical distributions (set of un-binned or binned data)
  - provides efficient and exact methods
Performances of Random Number

- Performances tests
  - lxplus, gcc 3.4
  - Intel 32 and 64 bits
- Uniform generation
- Gaussian
- Poisson number generation

<table>
<thead>
<tr>
<th>Random Number Uniform Generators</th>
<th>Intel32 (ns/call)</th>
<th>Intel64 (ns/call)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT (TRandom3)</td>
<td>22</td>
<td>9</td>
</tr>
<tr>
<td>TausWorthe (TRandom2)</td>
<td>17</td>
<td>6</td>
</tr>
<tr>
<td>RanLux (TRandom1)</td>
<td>120</td>
<td>98</td>
</tr>
<tr>
<td>LCG (TRandom)</td>
<td>14</td>
<td>5</td>
</tr>
</tbody>
</table>

**Random Gauss (ns/call)**

- Old TRandom::Gaus: 183
- New TRandom::Gaus: 42
- TRandom::Rannor: 130
- GSL Gaus: 54
- CLHEP Gaus: 90
- TF1 Gaus: 102
- UNURAN Gaus: 50

**Poisson Random Numbers**

- TRandom::Poisson
- Unuran Poisson
- Unuran Dari

Phystat LHC Workshop, CERN, June 27-29, 2007
Lorenzo Moneta, CERN/PH-SFT
Data Analysis Classes

- **TTree**
  - for sets of un-binned data and optimized for dealing with large data volumes
- **Histogram classes** (for binning data in 1, 2, 3 dimensions)
  - Profile histograms (1, 2, 3 dim.)
- **TGraph** classes:
  - **TGraph, TGraphErrors, TGraphAsymmErrors, TGraphBentErrors**
    - for sets of 2D (x,y) data
  - **TGraph2D, TGraph2DErrors**: 3D (x,y,z) data
  - provide various interpolation functions
    - splines, Delaunay triangulation for 2D
**QQ Plot**

- **TGraphQQ**
  - to draw quantiles of two data sets
  - to draw quantile of a data set vs a reference distribution
Visualization tools for Multi-Dimensional data

- Display of 3D histograms and functions (4D data) using OpenGL

- Developing tools for multi-dimensional data sets
  - spider (radar) plots
  - parallel coordinates
  - matrix of scatter plots
Fitting in ROOT

- Fit to ROOT data classes (Histograms, Trees, Graphs)
  - binned and un-binned fits
  - least square or likelihood fits
  - user defined model functions
  - possible to drive using a GUI
  - use linear and robust fits

- Interface exist (*TVirtualFitter*) for custom fits
  - user defined objective functions
  - various minimization methods
    - *Minuit, Fumili, Minuit2, Fumili2*

- **RooFit** for complex fitting and data modeling
- **TSpectrum** for peak finding and background subtraction
New Fitter GUI

- Developed a new Fit Graphics Interface for fitting the ROOT objects (TH1, TGraph etc...)

Lorentzian Peak on Quadratic Background
Classes for Specialized Fits

- **TBinomialEfficiencyFitter**: likelihood fit for efficiencies (data with binomial errors) obtained from division of two histograms.


- **TSplot**: extended maximum likelihood fit to signal and background with a tool (*SPlot*) to access the validity of the fit (unbias distribution of control variables).
Fitting Improvements

- Re-designing fitting and minimization classes
  - improve interfaces for easy of use
  - common entry for various fitting methods
  - better integration with other ROOT classes and packages \((\text{RooFit}, \text{TMVA}, \text{etc..})\)
  - easier to integrate (plug-in) new fitting and minimization methods
    - example: a user needs a minimizer from Nag C library
  - multi-thread support for parallel fits
Function Minimization

- New Object-Oriented version of Minuit (*Minuit2*)
  - Same basic functionality as in old version
    - *Migrad, Simplex, Minos* algorithms
  - Extended functionality:
    - single side parameter limits
    - added *Fumili* method for least square and likelihood fits
  - validated with extensive testing
    - same results and number of function calls to find minimum
  - interfaced in ROOT but can also be used standalone
- OO package for generic function minimization
  - easy to extend by inserting new minimization algorithms
  - plan to add eventually constrained minimization
Goodness of Fit

- Pearson Chi2 test for comparison of histograms.
  - new version (using algorithm from N. Gagunashvili)
    - weighted histograms comparisons
    - histogram with different scales
    - produce also normalized residuals
- Kolmogov-Smirnov test
  - for un-binned data
  - implemented a function in \textit{TMath}
Confidence Intervals

- Classes for confidence level estimation:
  - \textit{TFeldmanCousin}
    - FC confidence intervals for a Poisson process
      - without uncertainties in signal or background
  - \textit{TRolke}
    - profile likelihood for Poisson process
      - with uncertainty in background and/or signal
  - \textit{TLimit}
    - CL method used at LEP
      - apply to histograms of data and MC (signal + bkg)
      - can incorporate systematic uncertainties
      - semi-Bayesian method
Graphs Smoothing

- Cubic and Quintic splines via \textit{TSpline3,5} classes
- Smoothers of \((x,y)\) data via the class \textit{TGraphSmooth}
  - find regression function \(y(x)\)
  - algorithms from R
    - Kernel Smoother
    - Lowess Smoother
    - Super smoother (from Friedman)
- Plan to extend it for multi-dimensional data
  - for iso-surfaces \(z(x_1, \ldots, x_{n-1})\)
- Add smoothing for 1D un-bin data (kernel density estimator)
Robust Estimator

- Algorithms existing in R
- Robust least trimmed square fit (Linear Fit)
- remove outliers from fit

- \textit{TRobustEstimator} for multivariate analysis
- minimum covariance determinant estimator
MultiVariate Methods

- Neural networks via the `TMultiLayerPerceptron` class
  - can be used for classification or for regression analysis
- `TMultiDimFit` for function approximation
  - find parametrization of multidimensional data using polynomials (or Chebyshev or Legendre)
    - example: LHCb magnetic field map
- `TPrincipal` : principal component analysis
  - linear transformation of variables
- `TMVA` : toolkit for multivariate analysis
  - see next talk
Fast Fourier Transform

- Included in ROOT a common base class (*TVirtualFFT*)
  - add a functions to use it from *TH1* (*TH1::FFT*)
- Implemented an interface to the popular FFTW3 package (see www.fftw.org)
  - support for one and multi-dimensional transforms
  - support for complex and real transformations

- *TFFTClear* for complex input/complex output transforms
- *TFFTRadicalComplex* for real input/complex output
- *TFFTClearComplex* for complex input/real output
- *TFFTRadicalReal* for real input/output
Library Organization

- On-going re-organization of mathematical and statistical libraries
  - more modular libraries
    - libraries as MathCore will provide the basic functionality
  - reduce dependency between libraries
  - make easier the integration of contributed software
  - easier maintainability in the long term
- Review and revise some of existing algorithms
  - remove duplications and correct and improve them
- Better documentation (more examples and tutorials)
New Structure of ROOT
Math Libraries

- **Math Core**:
  - Functors & interfaces
  - Basic algorithms
  - Math functions

- **Math Core Extra**:
  - Physics Vectors
  - TComplex
  - TRandom
  - TMath

- **MathMore**:
  - Random Numbers
  - Extra algorithms
  - Extra Math functions
  - GSL

- **Statistical Libraries**:
  - New Stat Tools (Significance, Limit/CL etc.)
  - TMVA
  - MLP

- **Extra Libraries**:
  - Unuran
  - FFTW
  - Foam

- **Linear Algebra**:
  - TMatrix
  - SMatrix

- **Fitting and Minimization**:
  - New Fitter
  - RooFit
  - Minuit2 (OO Minuit)
  - Quadp
  - TFumili
  - TMinuit

- **Histogram Library**: TH1, TF1

- **Not yet released**: Y
- **Already existing**: Y

---

Phystat LHC Workshop, CERN, June 27-29, 2007  Lorenzo Moneta, CERN/PH-SFT 23
Aims for ROOT Math

- Concentrate on developing tools needed by experiments
  - in particular what is required for LHC data analysis
- Aim to have the tools and their implementations which are considered standard by our community
  - need input and feedback
    - often have a large variety of similar tools
    - or have various implementations of same tool
  - duplication can be good for easy comparison but can also create confusion for users
- We must decide on what is better to have in ROOT
  - need help from the statistical experts
Planned Developments

- New statistical tools for discovery (*K. Cranmer* and *W. Verkerke*)
  - for combination of results and able to incorporate systematics
  - will be based on *RooFit* classes (*RooPdf*)
- On-going developments also in *RooFit* and *TMVA*
- New visualization tools for multi-dimensional data sets
  - spider plots, parallel coordinates, etc..
- Cluster algorithms (from R)
- Loess smoothing for multi-dimensional data (locally weighted polynomial regressions)
- Improve goodness of fit tests
- Constrained minimization
- Requests from experiments
Conclusions

- Large collection of Math and Statistical tools already available in ROOT
  - working on improving them for better usability and for easier integrations of new tools
- Considerable efforts from external contributors in developing missing tools for LHC analysis
  - multivariate analysis
  - new statistical tools for discovery
- Important to ensure the correctness of math and statistical tools we are going to use
- Need continuously the feedback from users and experts
References and Documentation

- **ROOT** reference guide: [http://root.cern.ch/root/htmldoc/ClassIndex.html](http://root.cern.ch/root/htmldoc/ClassIndex.html)
- **MathCore** online doc: [http://www.cern.ch/mathlibs/sw/MathCore/html/index.html](http://www.cern.ch/mathlibs/sw/MathCore/html/index.html)
- **Minuit2** online doc: [http://www.cern.ch/mathlibs/sw/Minuit2/html/index.html](http://www.cern.ch/mathlibs/sw/Minuit2/html/index.html)
- **UNURAN** homepage: [http://statmath.wu-wien.ac.at/unuran/](http://statmath.wu-wien.ac.at/unuran/)
- **ROOT Talk Forum** (for support, requests and discussions)
- **ROOT Savannah** for reporting bugs