An introduction to RooStats

Grégory Schott

Karlsruhe Institute of Technology (KIT)

RooStats tutorials
CERN - November 26th 2009

Outline

- What is RooStats? It's a collaborative project between ATLAS, CMS and ROOT to provide a consolidated set of statistical tools
  - TWiki: https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome
- In this presentation: RooStats introduction: Motivation and general description of the project
  - This morning: RooFit presentation and tutorials
  - This afternoon: Presentation on the concrete implementation / usage of RooStats and tutorials
  - Tomorrow morning: Continued tutorials
- Hope most of you are familiar with the material in L. Lista introductory statistics lecture of last week:
  - See: http://indico.cern.ch/conferenceDisplay.py?confId=73545

Motivations

- Statistical interpretation of data in an analysis
  - useful to have a common, well tested package
- Combination of analyzes within/across experiments
- Be able to compare statistical methods
- Generalize and cleanup statistical tools in ROOT
- Want to agree on statistical conventions
  - avoid *apples-to-oranges* comparisons

Statistics usage

- Common purposes:
  - point estimation: determine the best estimate of a parameter
  - estimation of confidence/credible interval (multi-dimensional contours, in 1-D a 2-sided or just a lower or higher limit, ...)
  - hypothesis tests: evaluation of p-value for one or multiple hypotheses (significance)
  - goodness-of-fit: how well a model describes the data
- For these things and for others, RooStats can help you (there are ways to do GOF tests but no specific tools in RooStats yet)

Terminology

- Observable: quantities that are directly measured by an experiment (or their MC predictions) (eg. candidates mass, helicity angle, NNet output) – they form a dataset
- Model: probability density function (PDF) that describes one or multiples observables – parametric or non-parametric. PDF are normalized such that their integral over any observable is 1
- Parameter(s) of interest: parameters of the model that one wishes to estimate or constrain (eg. particle mass, cross-section)
- Nuisance parameters: parameters of the model that are uncertain but not “of interest” (systematics-associated normalization or shape parameters)

Features

- Rely on RooFit: provides a developed & flexible basis
- Extension to complex problems
  - Work on arbitrary data and model and can handle many observable, parameter of interest and nuisance parameters
- Combine at analysis level
  - Retain full information for treating correlations
- All statistical methods start from description of likelihood function (or PDF)

Likelihood analysis

- Simple likelihood: $L_i(n_i|r,s_i,b_i) = \frac{e^{-rs_i-b_i}}{n_i!}(rs_i+b_i)^{n_i}$
  - Can be extended to binned likelihood
- Multiple channels: $L(r) = \prod_i L_i(n_i|r,s_i,b_i)$
- With observables; extended, unbinned likelihood:

$$L(\vec{x}|r,s,b,\vec{\theta}_s,\vec{\theta}_b) = \frac{e^{-rs-b}}{n!}(rs+b)^n \prod_{j=1}^{n}(rsf_s(\vec{x}_j|\vec{\theta}_s)+bf_b(\vec{x}_j|\vec{\theta}_b))$$

  - *f_s*, *f_b* signal and background distribution from MC or control samples

RooFit PDFs

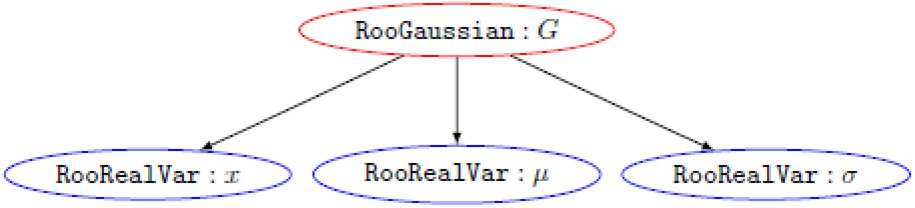
- Example of PDF definition in RooFit:

$G(x|\mu,\sigma)$

```
// define observables and parameters
RooRealVar x(“x”,”x”,100,200);
RooRealVar mu(“mu”,”#mu”,150);
RooRealVar sigma(“sigma”,”#sigma”,5,0,20);
// make a simple model
RooGaussian G(“G”,”gaussian”,x,mu,sigma);
G.graphVizTree(“GaussianModel.dot”);
```

```
// shortcut factory definition of the model
RooWorkspace w;
w.factory(Gaussian::G(x[100,200],mu[150],sigma[5,0,20]);
w.Print();
```

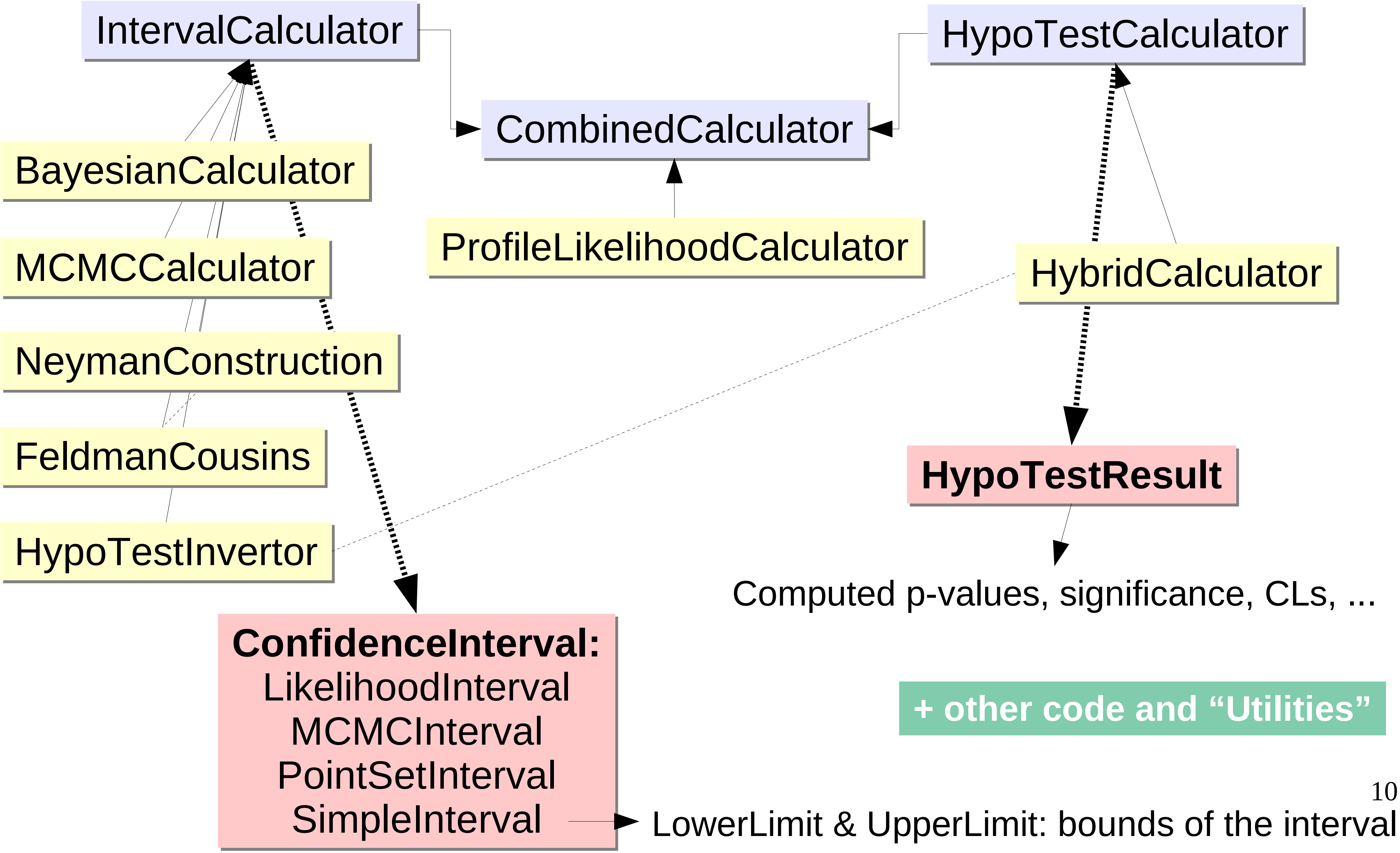
```
RooWorkspace()  contents
variables
        (mu,sigma,x)
p.d.f.s
        RooGaussian::G[ x=x mean=mu sigma=sigma ] = 1
```



(some elements adapted from R. Cousins – similar slide also presented recently by G. Cowan)

- Once the statistical problem is described, various methods can be easily applied and compared
  - Bayesian, Frequentist, Likelihood ratio, “CLs”, ...
- It is recommended / the community can ask the result be shown with one or another method and to study sampling properties
  - If methods agree → important check of robustness
  - If methods disagree → we learn something:
    - The results are answers to different questions
    - Bayesian methods can have poor frequentist properties
    - Frequentist methods can badly violate likelihood principle

Overview of classes in RooStats



IntervalCalculator

HypoTestCalculator

CombinedCalculator

BayesianCalculator

MCMCCalculator

NeymanConstruction

FeldmanCousins

HypoTestInvertor

ProfileLikelihoodCalculator

HybridCalculator

HypoTestResult

Computed p-values, significance, CLs, ...

ConfidenceInterval:
LikelihoodInterval
MCMCInterval
PointSetInterval
SimpleInterval

LowerLimit & UpperLimit: bounds of the interval

+ other code and “Utilities”

Calculator classes

- **ProfileLikelihoodCalculator**: interval estimation and hypothesis testing
- **BayesianCalculator**: adaptive numerical integration
- **MCMCCalculator**: Bayesian with Markov-Chain Monte Carlo
- **NeymanConstruction**: classical/frequentist interval calculator
- **FeldmanCousins**: Neyman construction with likelihood ratio ordering rule
- **HybridCalculator**: frequentist hypothesis testing with bayesian integration of nuisance parameters
- **HypoTestInvertor**: inversion of hypothesis tests into a confidence interval

Other RooStats classes and utilities

- **SPlot:** a technique used to produce a weighted plots of an observable distribution
- **ModelConfig:** holds all the elements about a model configuration
- **HLFactory:** wrapper around the RooFit factory to help in building RooFit PDFs
- **BernsteinCorrection**, utilities specific to number counting analyses, ...

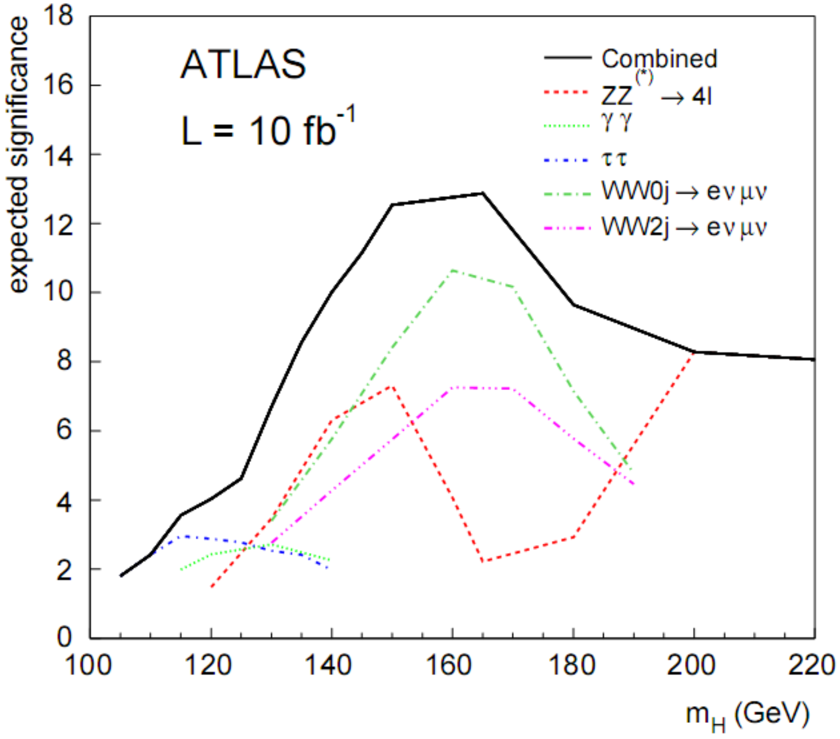
Workspace

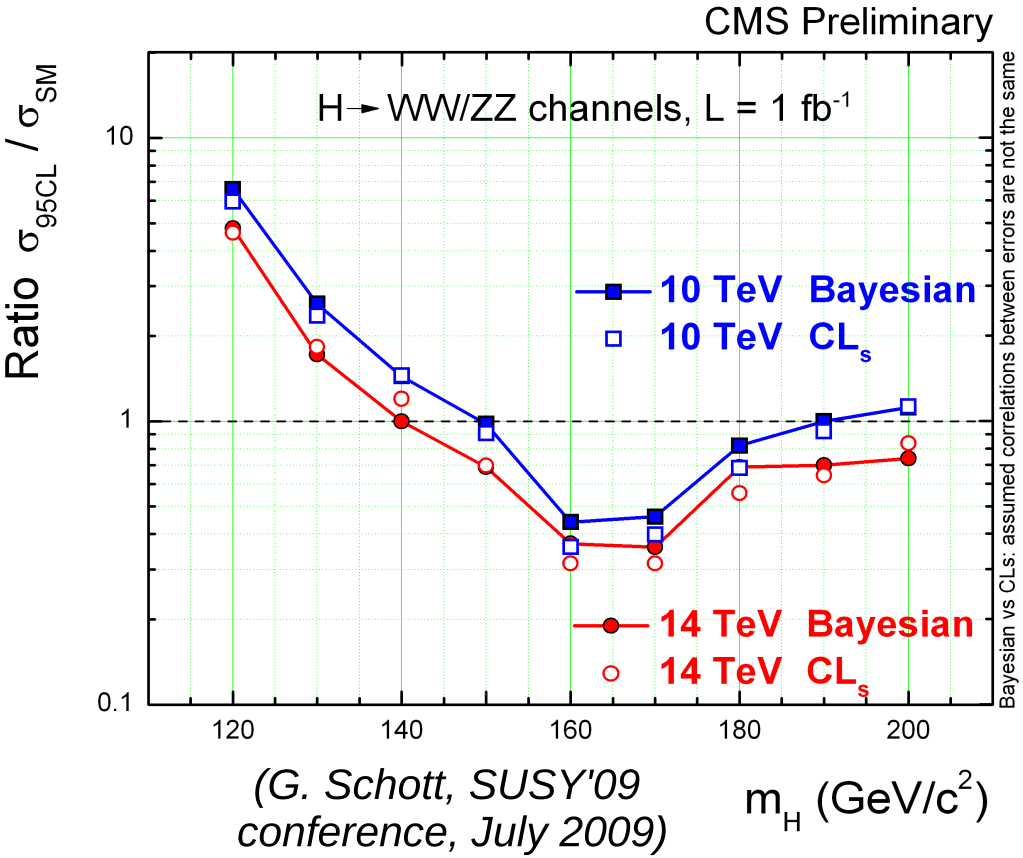
- Developed workspace class to facilitate combinations
- Workspaces contains any RooFit object, in particular:
  - Data (binned or unbinned)
  - PDF model
  - Uncertainty / shape of nuisance parameters
- Utilities to correlate objects or ease the description of the PDF model (Factory)
- Can be saved to file, easily shared and used in combination with Workspaces of other analyzes
- Allow to eventually distribute data and model in an electronic form once analysis has been published

Some Atlas/CMS Higgs projections

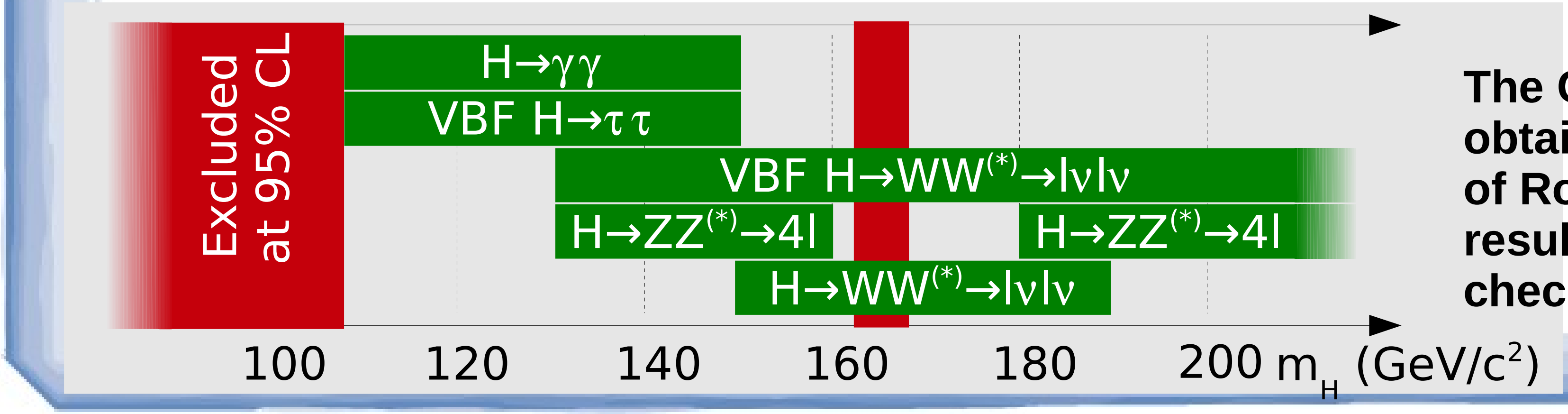
Median expected exclusion

(CSC report arXiv:0901.0512)





(G. Schott, SUSY'09 conference, July 2009)

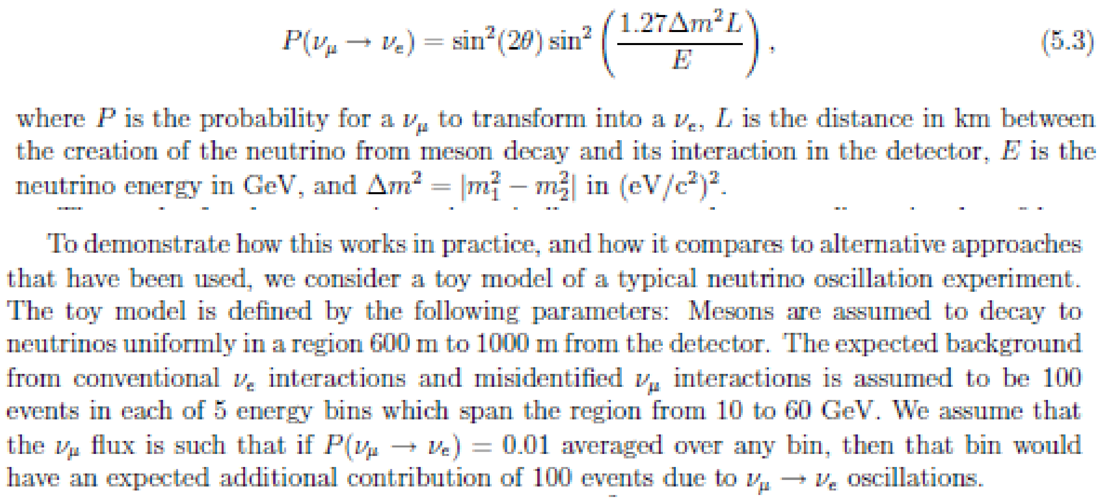


The CLs result was obtained with the code of RooStats, other results are being checked with RooStats

Neutrino oscillation example

Kyle coded up neutrino oscillation experiment based on description of in Feldman & Cousins's original paper.

Generate toy data at same true parameters and compare RooStats with results in paper

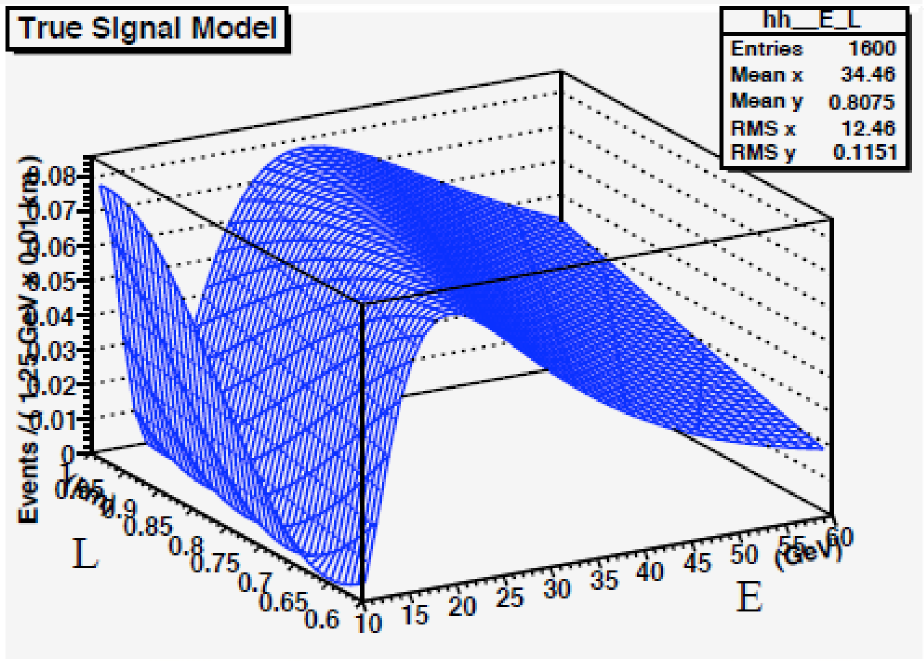


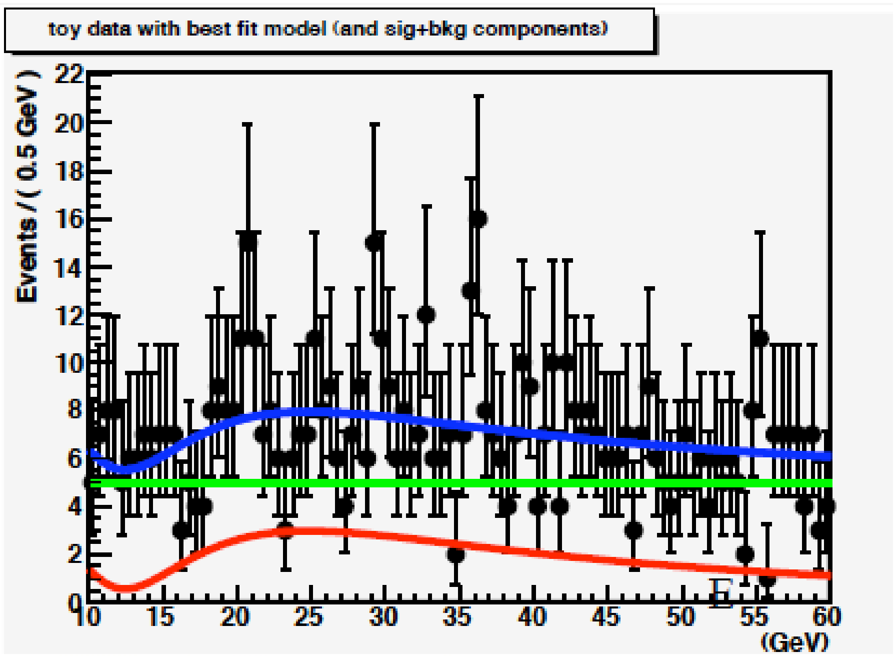
$$P(\nu_\mu \to \nu_e) = \sin^2(2\theta)\sin^2\left(\frac{1.27\Delta m^2 L}{E}\right), \qquad (5.3)$$

where P is the probability for a ν_μ to transform into a ν_e, L is the distance in km between the creation of the neutrino from meson decay and its interaction in the detector, E is the neutrino energy in GeV, and $\Delta m^2 = |m_1^2 - m_2^2|$ in $(\mathrm{eV}/c^2)^2$.

To demonstrate how this works in practice, and how it compares to alternative approaches that have been used, we consider a toy model of a typical neutrino oscillation experiment. The toy model is defined by the following parameters: Mesons are assumed to decay to neutrinos uniformly in a region 600 m to 1000 m from the detector. The expected background from conventional ν_e interactions and misidentified ν_μ interactions is assumed to be 100 events in each of 5 energy bins which span the region from 10 to 60 GeV. We assume that the ν_μ flux is such that if $P(\nu_\mu \to \nu_e) = 0.01$ averaged over any bin, then that bin would have an expected additional contribution of 100 events due to $\nu_\mu \to \nu_e$ oscillations.

http://root.cern.ch/root/html/tutorials/roostats/rs401d_FeldmanCousins.C.html





Summary

- Code in CMS and ATLAS combined and improved to form the RooStats project
- RooStats available from ROOT since December 2008 (new release yesterday 5.25.04)
  - Common implementation of methods
  - Speak common language for combination
  - Flexible enough to accommodate “all” cases
  - Most statistical classes one would need are there
- Some improvements needed:
  - Consolidation / speed / documentation / testing
  - Open project, new contributors are welcome

Documentation and user support

- Core developers: K. Cranmer (*Atlas*), L. Moneta (*ROOT*), G. Schott (*CMS*), W. Verkerke (*RooFit*)
- RooStats TWiki: https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome

- **Documentation:**
  - RooFit's user's guide: http://root.cern.ch/drupal/content/users-guide (*to be completed*)
  - RooStats manual (*in preparation*)
  - ROOT reference guide: http://root.cern.ch/root/html/ClassIndex.html
  - RooFit and RooStats tutorial macros: http://root.cern.ch/root/html/tutorials
  - RooFit interface to the Bayesian Analysis Toolkit (BAT): http://cern.ch/schott/public/BCRooInterface

- **RooStats user support:**
  - Request support via ROOT talk forum: http://root.cern.ch/phpBB2/viewforum.php?f=15 (questions on statistical concepts tolerated)
  - Submit bugs to ROOT Savannah: https://savannah.cern.ch/bugs/?func=additem&group=savroot
  - *In many cases, posting also a simple self-contained macro reproducing the problem helps a lot*

- **Contacts for statistical questions:**
  - ATLAS statistics forum: hn-atlas-physics-Statistics@cern.ch (Cowan, Gross *et al*)
    - TWiki: https://twiki.cern.ch/twiki/bin/view/AtlasProtected/StatisticsTools
  - CMS statistics committee: (Cousins, Demortier *et al*)
    - via hypernews: hn-cms-statistics@cern.ch or directly: cms-statistics-committee@cern.ch

Before we get started

- RooStats is distributed together with ROOT since version 5.22. In general, the latest version is strongly recommended (and mandatory for these tutorials: ROOT 5.25/04)
- Installation: http://root.cern.ch/drupal/content/development-version-52504
  - Locally using pre-built binaries
    - Compiled from source: `./configure -enable-roofit ; make`
  - On lxplus / with AFS (not recommended, WLAN saturation)

    ```
    /afs/cern.ch/sw/lcg/app/releases/ROOT/5.25.04/
    ```
- Usage:
  - In CINT: `using namespace RooFit; using namespace RooStats`
  - Strongly recommend you compile your macros:

    ```
    root[0] .L macro.C+
    root[1] macro()
    ```