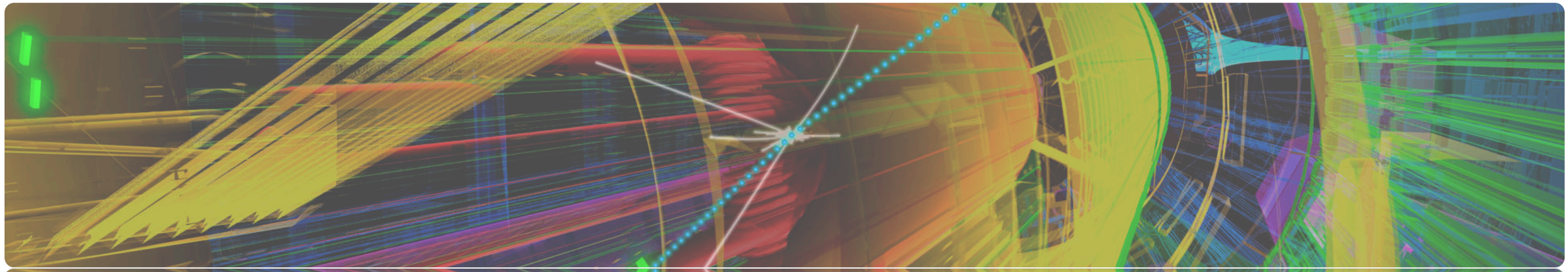
Statistics Tools for Particle Physics

An introduction...



Ioannis Nomidis

Aristotle University of Thessaloniki

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

Co- financed by Greece and the European Union

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

Introduction

- Statistics... what is it?

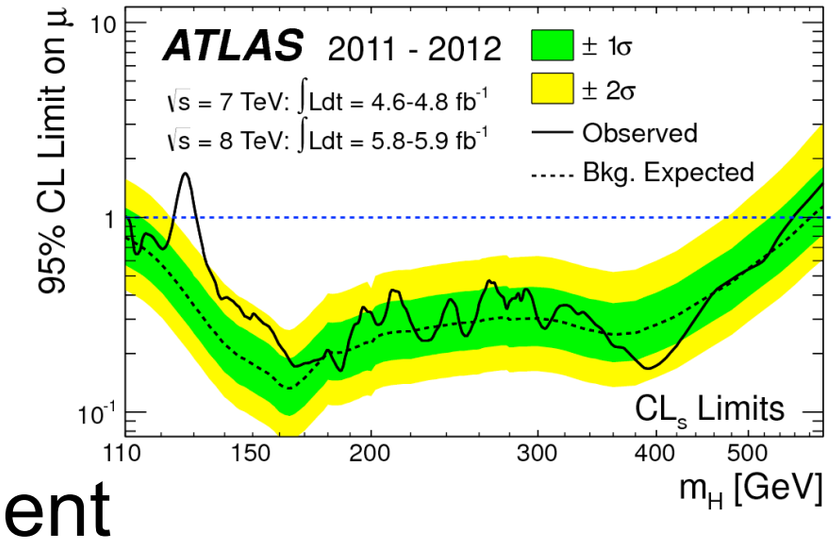
theory ⟶ **predictions**

data ⟶ **theory**

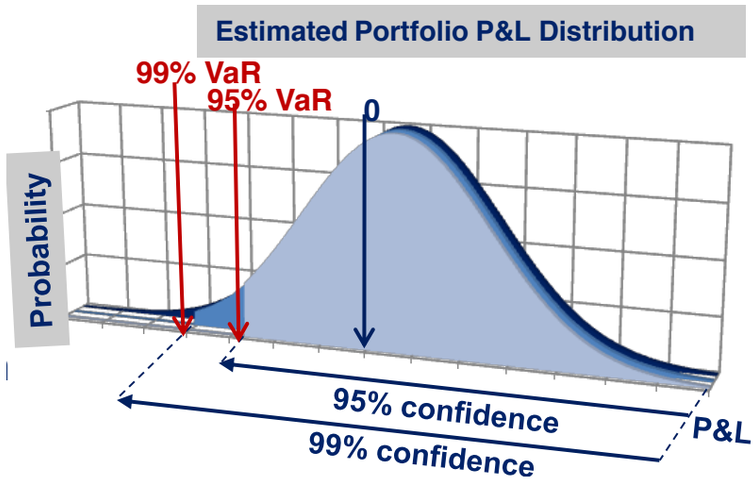
- For example, Statistics in Particle Physics involves:
  - looking at histograms, probability distributions
  - fits for parameter estimation
  - data unfolding, cross-section determination
  - setting limits in absence of signal

Introduction

Why become a Statistics expert



- Necessary for Particle Physics
  - See Higgs boson discovery
- Applications in finance and risk management
  - See job offers in consulting companies... Ph.D. in physics a common requirement!



In this lecture:

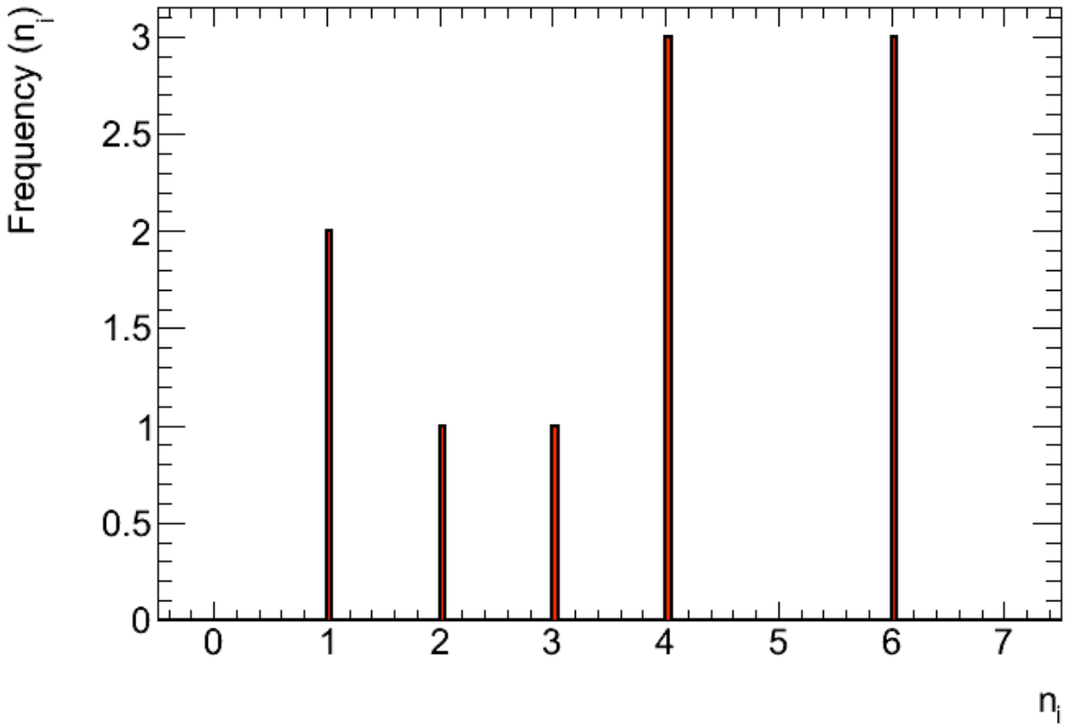
- Basics of Statistics
- Introduction to Roofit / Roostats
- Some code examples

Probability

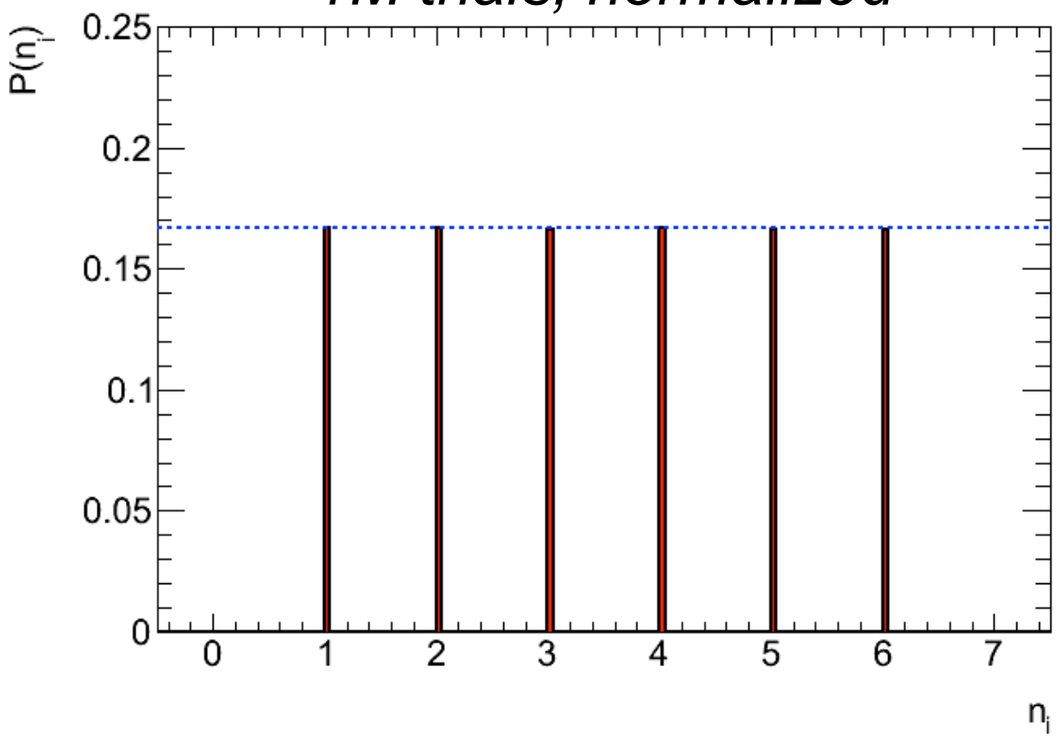
- **Random variable:** Its value cannot be predicted, instead the probability for obtaining a value can be expressed
- **Probability** (frequentistic): $$P_i = \lim_{N\to\infty} \frac{n_i}{N} \simeq \frac{n_i}{N} = P(n_i)$$

Rolling dices: a simple discrete-variable experiment

10 trials



1M trials, normalized



- The set of $P(n_1)$, $P(n_2)$, ... $P(n_i)$ defines a **probability distribution**

Probability density function

- **Probability density function (pdf):** defined for continuous variables

$f(x)$ *the pdf, not a probability*

$P(x \in [x_i, x_i + dx]) = f(x_i)dx$ *the probability*

$\int f(x)dx = 1$ *pdfs are always normalized to unity*

$f(x|\theta)$ *a parametrized pdf*

x *a set of random variables*

θ *a set of parameters*

- The random variable (x) can be independent observations or repetitions of a given experiment
- In HEP it usually is the number of events found in a given experiment, where *event* refers to the measurement of a given quantity (i.e. mass, cross-section)

Likelihood function

- The probability of obtaining a set of observations / measurements derives from the product of the pdf for each individual measurement (x_i)

$$P(x|\theta) = \prod_{i=1}^{N} f(x_i|\theta)$$

- For a given set of observations $x \rightarrow x^{\text{observed}}$, $P(x^{\text{observed}} \mid \theta)$ is not a pdf and is a function of θ only

- **Likelihood function**: $L(\theta) = P(x^{obs}|\theta)$

Binned and unbinned likelihood

- **Unbinned likelihood**:

$$L(\theta) = \prod_{i \equiv \text{event}}^{N} f(x_i|\theta)$$

➪ ***computationally slow for large N, but can use the full event information***

x_i *in this case is the result of the i-th measurement*

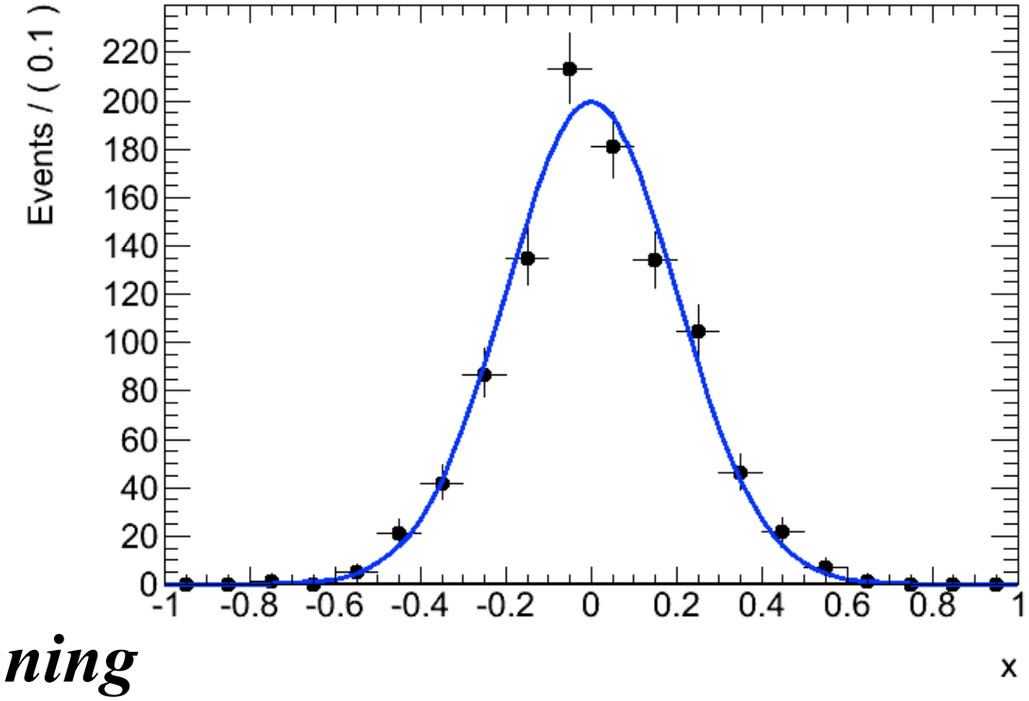
➪ ***ideal for small N and multi-dimensional fits***

- **Binned likelihood**:

$$L(\theta) = \prod_{i \equiv \text{bin}}^{N} f(n_i|\theta)$$

bin *is a range in variable* x*, say* $x_i + dx$

n_i *in this case is the number of events measured in range* $x_i + dx$



➪ ***much faster, but result depends on the binning***

➪ ***ideal for large N***

Likelihood vs χ^2 method

- Likelihood function can be easily interpreted and applied on event-by-event level. The χ^2 method cannot be applied on event level... (or at least it is not straight-forward)
- It can be shown that ML converges faster and with better efficiency. Unless there are special conditions (i.e. limitations in computing power), experts recommend to use ML method.
- ML does not have a problem when there are bins with zero or few events. The χ^2 method is wrong when bin has less than 4 events!

Terminology

- **observable:** quantity directly measured by an experiment and present in a data set.
- **model:** a pdf that describes/predicts the distribution of certain observables, normalized to unity.
- **parameter of interest:** the model parameter that we want to estimate / set limits on (eg. mass, x-section)
- **nuisance parameters:** the model parameters that are known with limited precision and are not 'interesting' in the above sense (eg. background normalization, shape parameters, systematic unc.)
- **fitting:** the set of statistical tests we run to estimate a parameter $\boldsymbol{\theta}$ given some data $\boldsymbol{x}$ and a model $\boldsymbol{P(x|\theta)}$
- **Maximum likelihood estimate:** the result of the ML fit

ROOT, RooFit, RooStats

- ROOT is the standard analysis package used in HEP
  - Can be downloaded from the site: http://root.cern.ch/drupal/
  - Available binaries and source code to compile in your system
- RooFit is a ROOT library providing a complete toolkit for data analysis
  - Comes with ROOT; if compiling ROOT locally, when running configure use flag: `--enable-roofit`
  - To use it, just add in your macros: `using namespace RooFit;`
  - Documentation: http://root.cern.ch/drupal/content/roofit
- RooStats is a set of statistical tools built on top of RooFit, distributed with ROOT
  - To use it:

```
using namespace RooFit;
using namespace RooStats;
```

- Apart from online documentation, plenty of helpful examples in ROOT: $ROOTSYS/tutorials/roofit, $ROOTSYS/tutorials/roostats

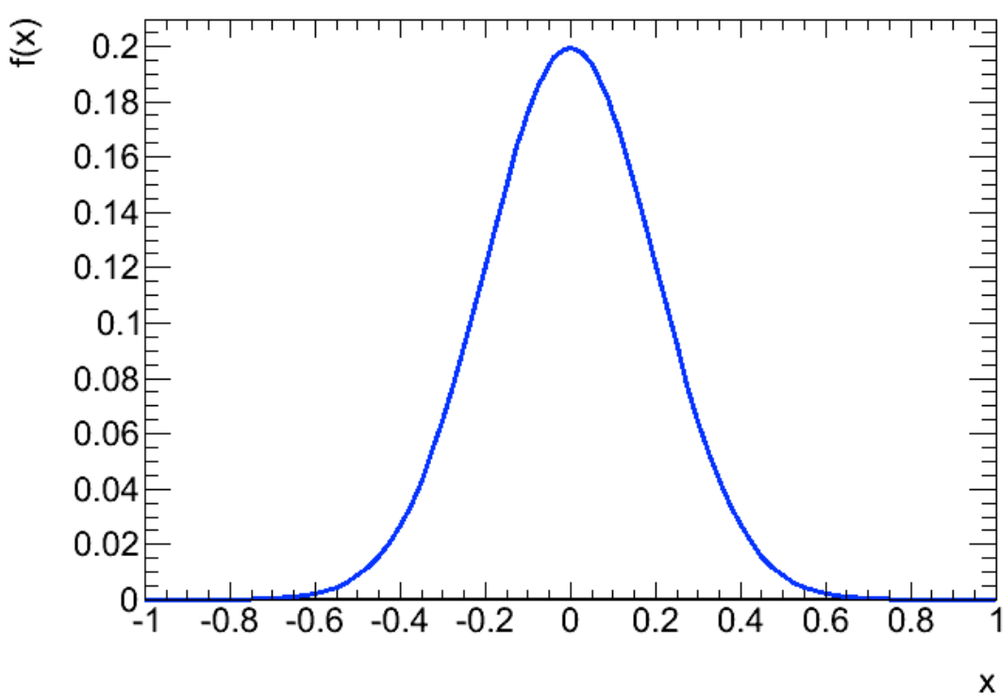
Simple pdf

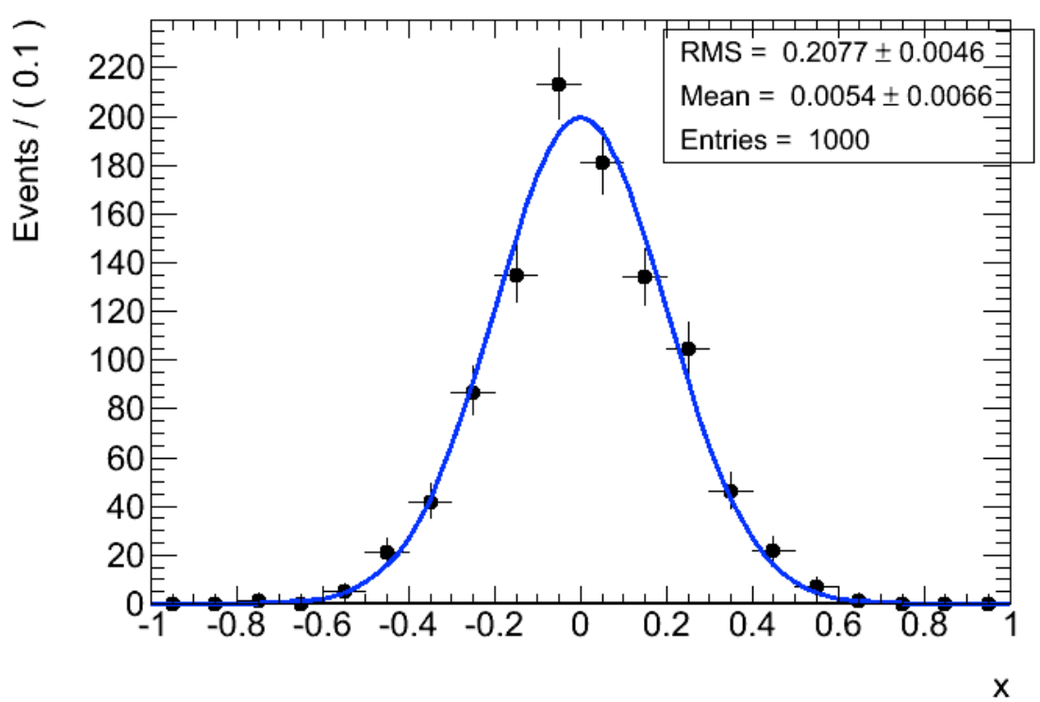
- Create a simple pdf, plot it and project with pseudo-data

```
using namespace RooFit;

  RooRealVar x("x","some random variable",-1,1);
  RooRealVar mu("mu","gaussian mean value",0,-1,1);
  RooRealVar sigma("sigma","gaussian width",0.2,0.1,0.3);

  RooGaussian myPdf("myPdf","gaussian pdf",x,mu,sigma);
```



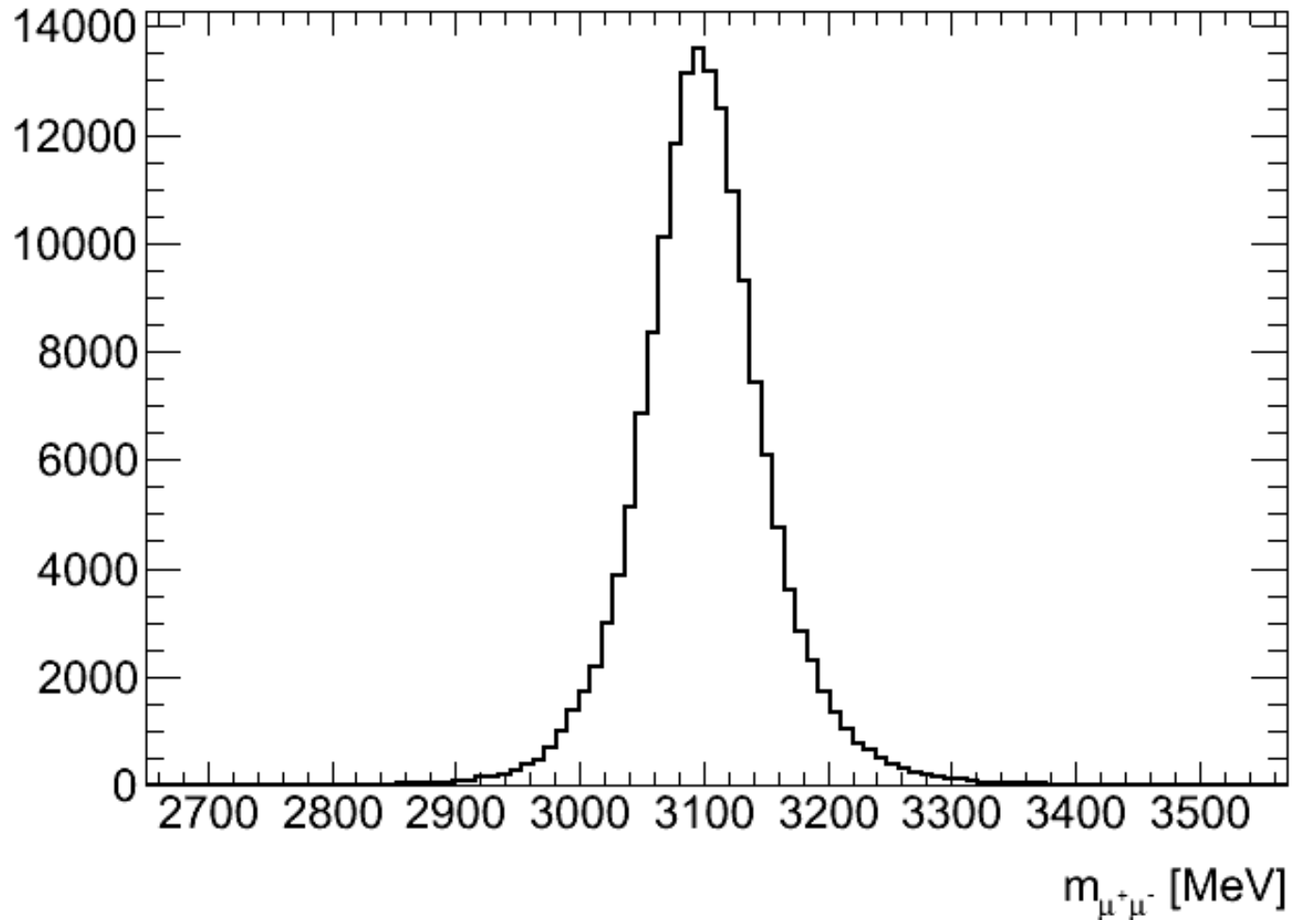


$$\mathcal{L} = \prod_i G(m_i|\mu, \sigma)$$

Adding pdfs

- Fit a simple mass distribution

Extract signal parameters (mean and sigma)



$$\mathcal{L} = \prod_i f_1 \cdot G(m_i|\mu, \sigma_1) + (1 - f_1) \cdot G(m_i|\mu, \sigma_2)$$

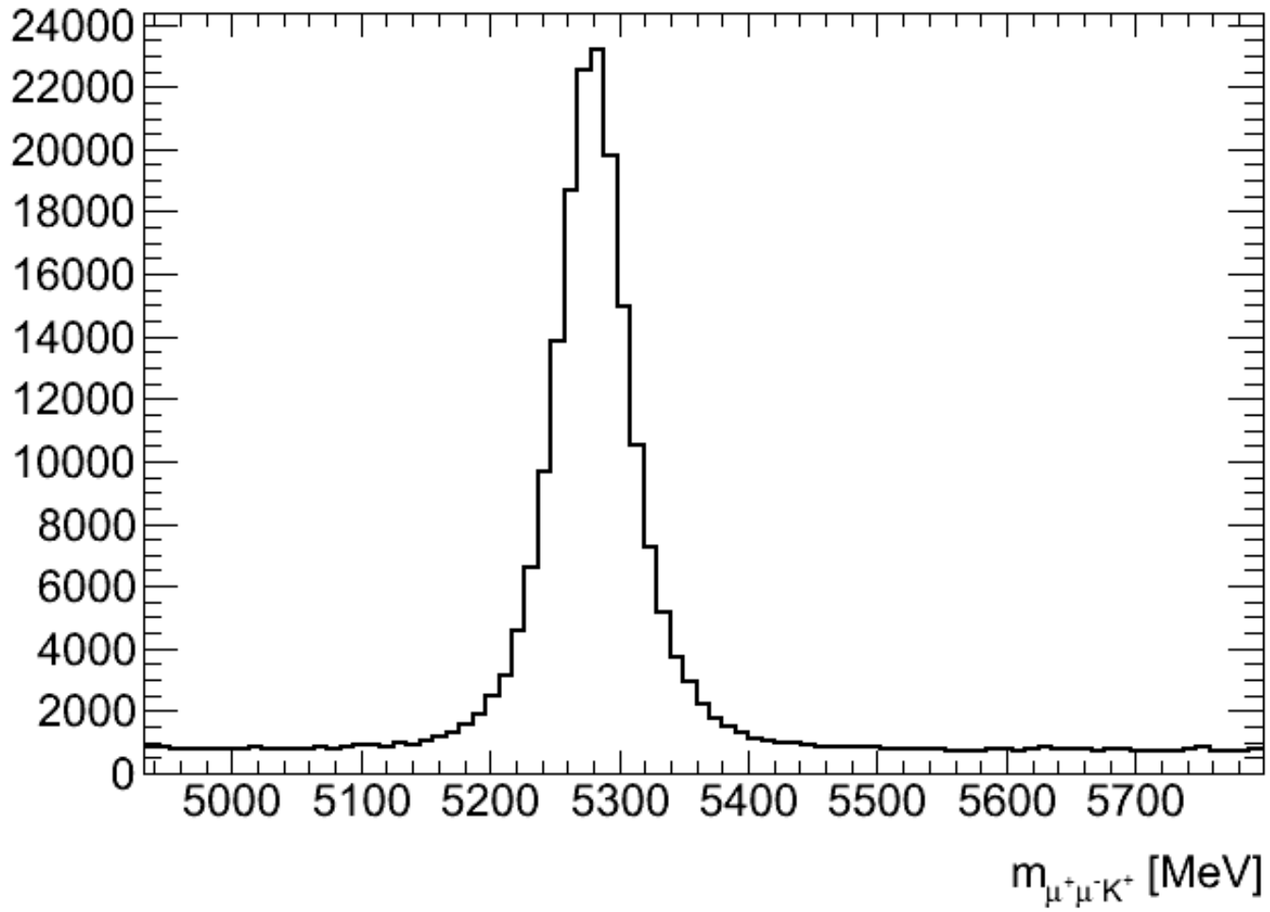
```
RooRealVar mean("mean","mean of gaussians",3097,3000,3200,"MeV");
RooRealVar sigma1("sigma1","width of gaussian 1",30,0,100,"MeV");
RooRealVar sigma2("sigma2","width of gaussian 2",80,50,200,"MeV");

RooGaussian gauss1("gauss1","Signal component 1",m,mean,sigma1) ;
RooGaussian gauss2("gauss2","Signal component 2",m,mean,sigma2) ;
RooRealVar a("a","",0.5,0,1);
RooAddPdf  signal_model("signal_model","a*G1+(1-a)*G2",RooArgList(gauss1,gauss2),a) ;
```

Signal+background pdf

- Fit a signal+background mass distribution

Extract the signal parameters and signal fraction



$$\mathcal{L} = \prod_i f_1 \cdot G(m_i|\mu, \sigma_1) + f_2 \cdot G(m_i|\mu, \sigma_2) + (1 - f_1 - f_2) F_{bg}(m_i, a)$$

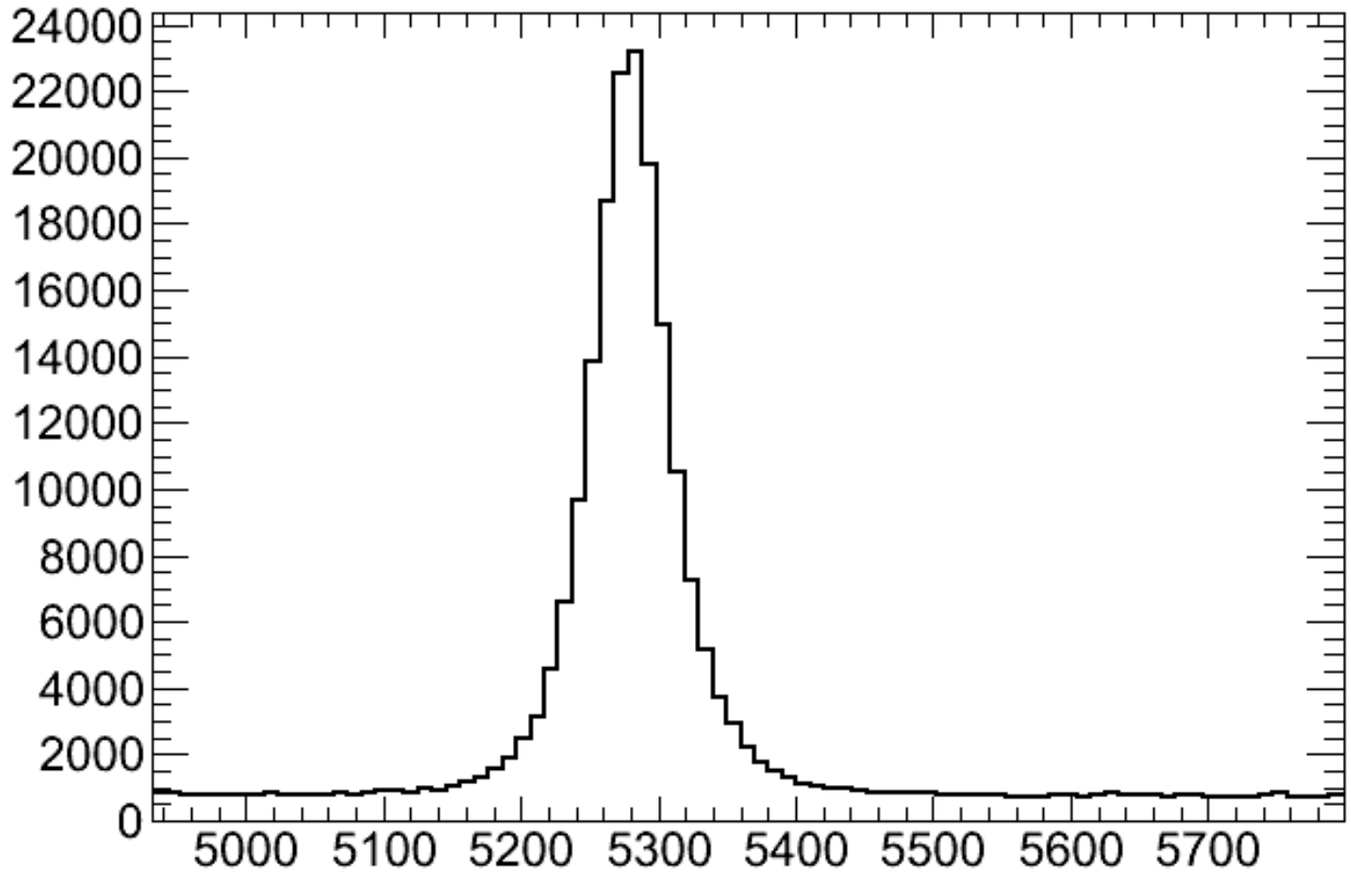
$$\mathcal{L} = \prod_i f_{sig} \cdot F_{sig}(m_i|\mu, \sigma_1, \sigma_2) + (1 - f_{sig}) F_{bg}(m_i, a)$$

```
RooRealVar fsig("fsig","",0.8,0,1);
RooAddPdf model("model","",RooArgList(signal_model,bkg_model),fsig);
```

Extended likelihood

- Extended likelihood

Usually, we don't need the signal fraction but the number of signal and background events



$$\mathcal{L} = \prod_i f_{sig} \cdot F_{sig}(m_i|\mu, \sigma_1, \sigma_2) + (1 - f_{sig}) F_{bg}(m_i, a)$$

$$\mathcal{L} = \prod_i \left[\frac{n_{sig}}{n_{sig} + n_{bg}} \cdot F_{sig}(m_i|\mu, \sigma_1, \sigma_2) + \frac{n_{bg}}{n_{sig} + n_{bg}} F_{bg}(m_i, a) \right] \cdot P(n_{sig} + n_{bg}, N)$$

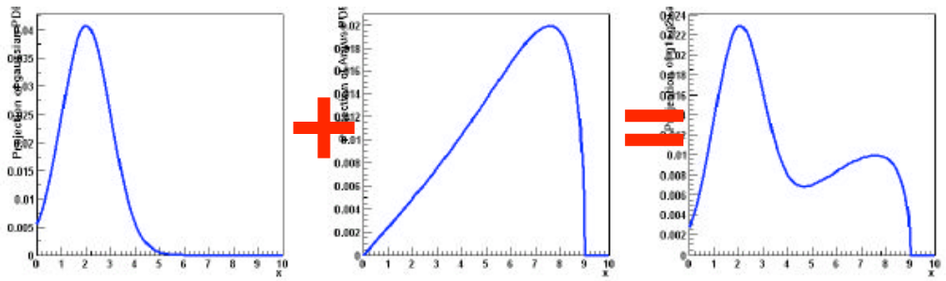
```
  RooRealVar nsig("nsig","",50000,0,1e6);
  RooRealVar nbkg("nbkg","",150000,0,1e6);
  RooAddPdf model("model","",RooArgList(signal_model,bkg_model),RooArgList
(nsig,nbkg));
```

Composite models

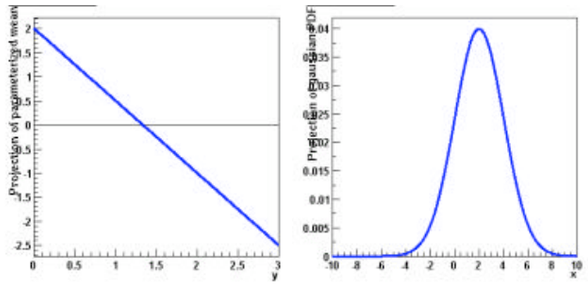
- Other possibilities for composite models

RooAddPdf

– Addition



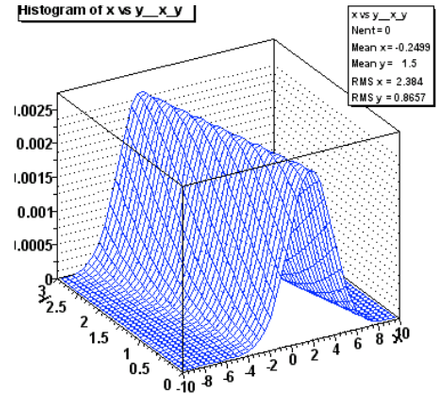
– Composition ('plug & play')



$m(y;a_0,a_1)$ $g(x;m,s)$

Possible in *any* PDF

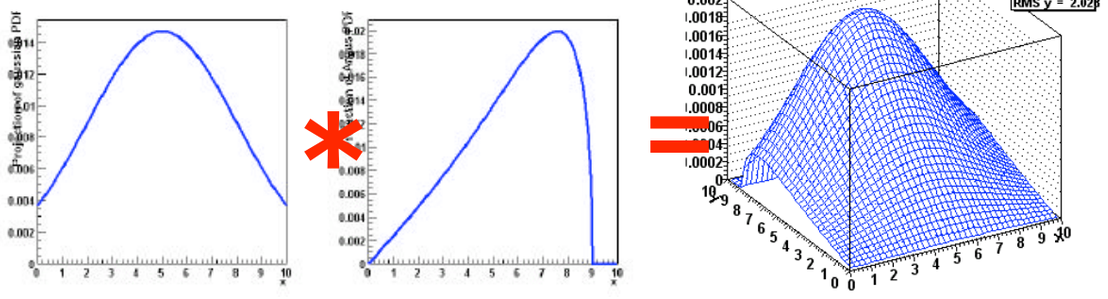
No explicit support in PDF code needed



g(x,y;a0,a1,s)

RooProdPdf

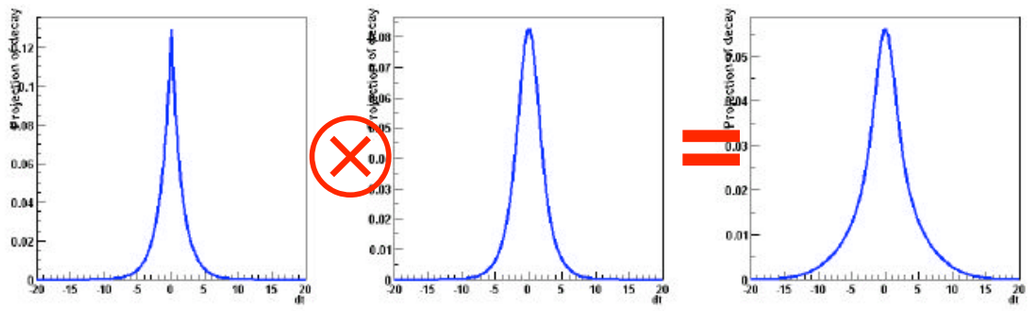
– Multiplication



Wouter Verkerke,

RooFFTConvPdf

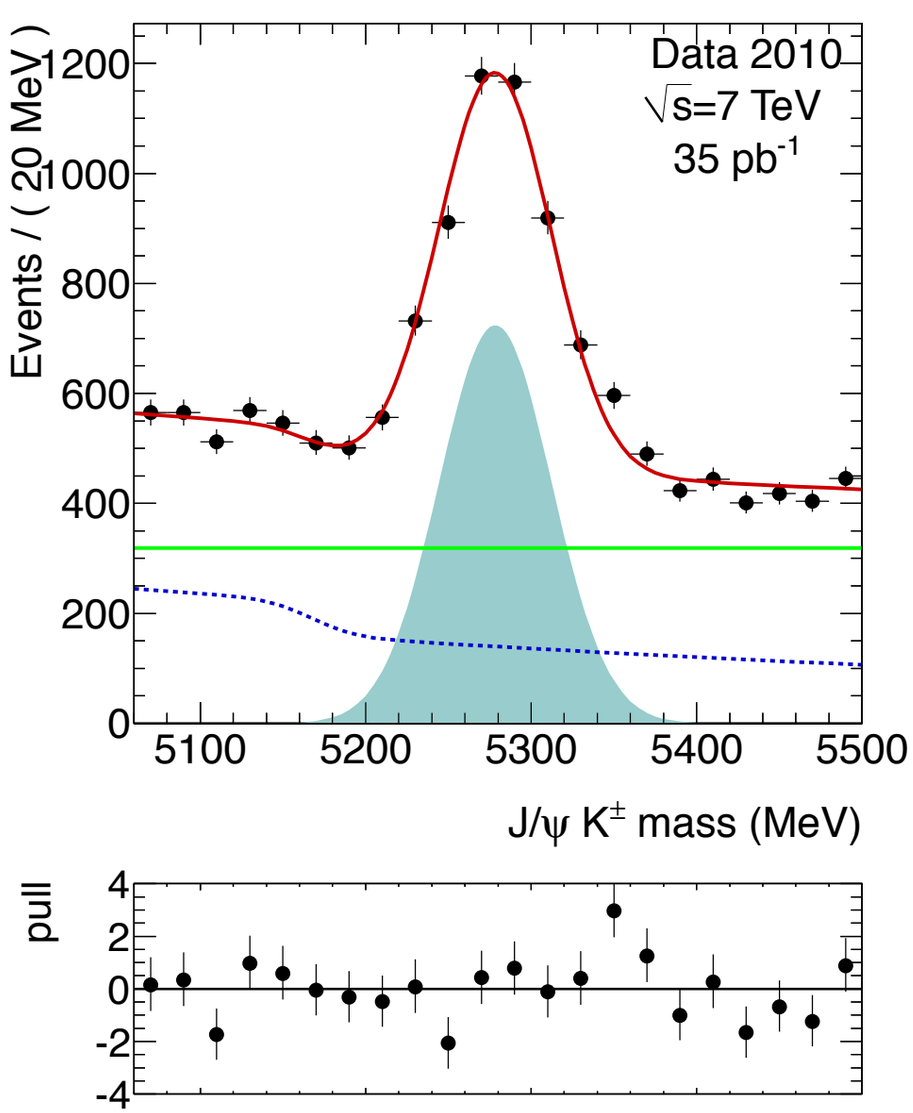
– Convolution

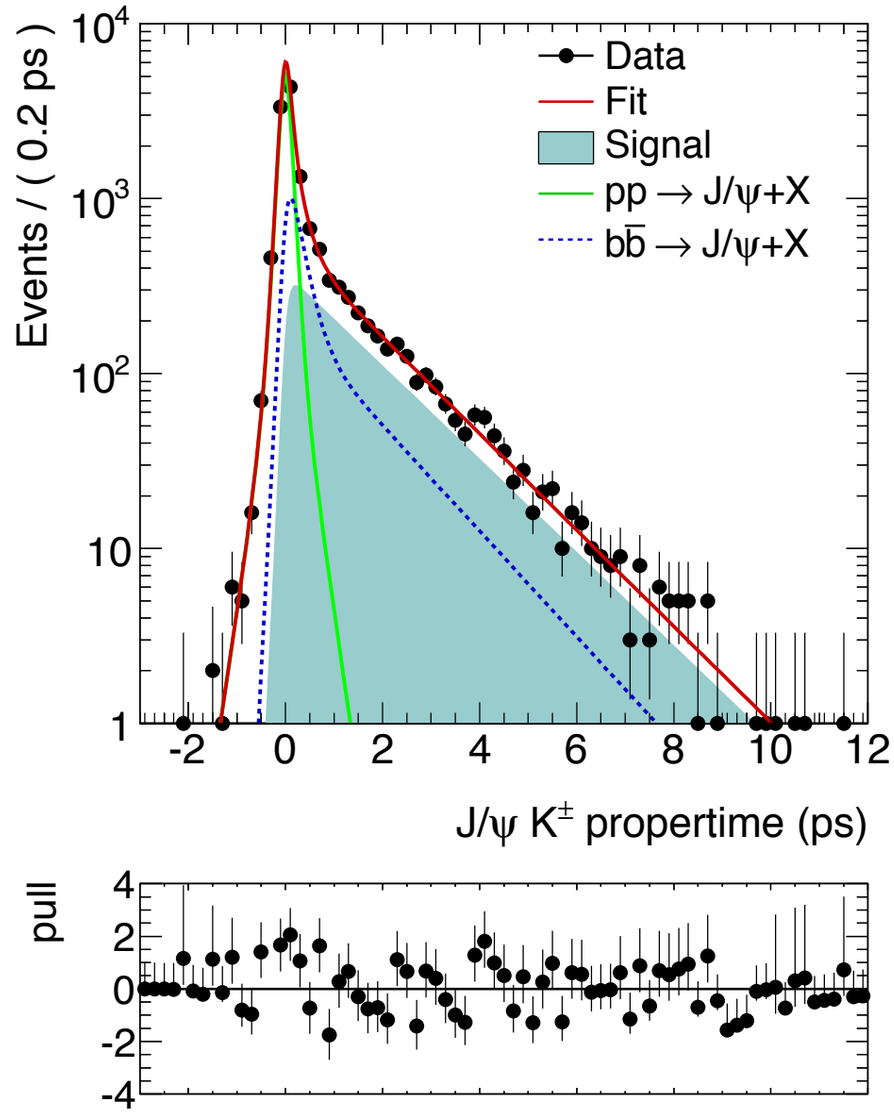


Wouter Verkerke, UCSB

Examples

- A two-dimensional fit on mass and lifetime



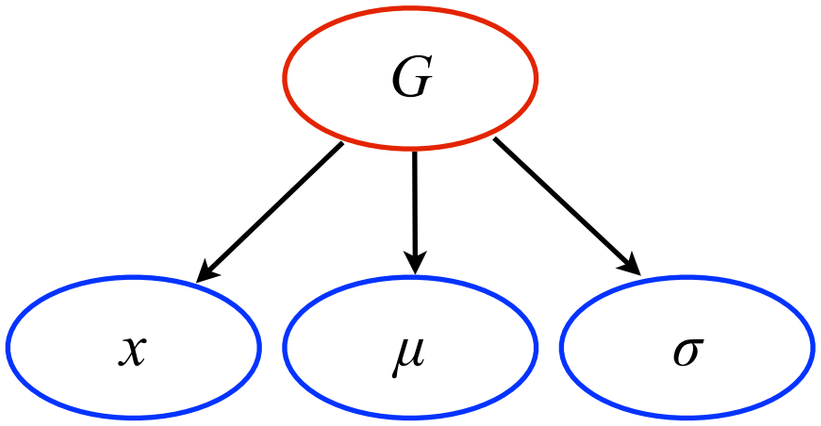


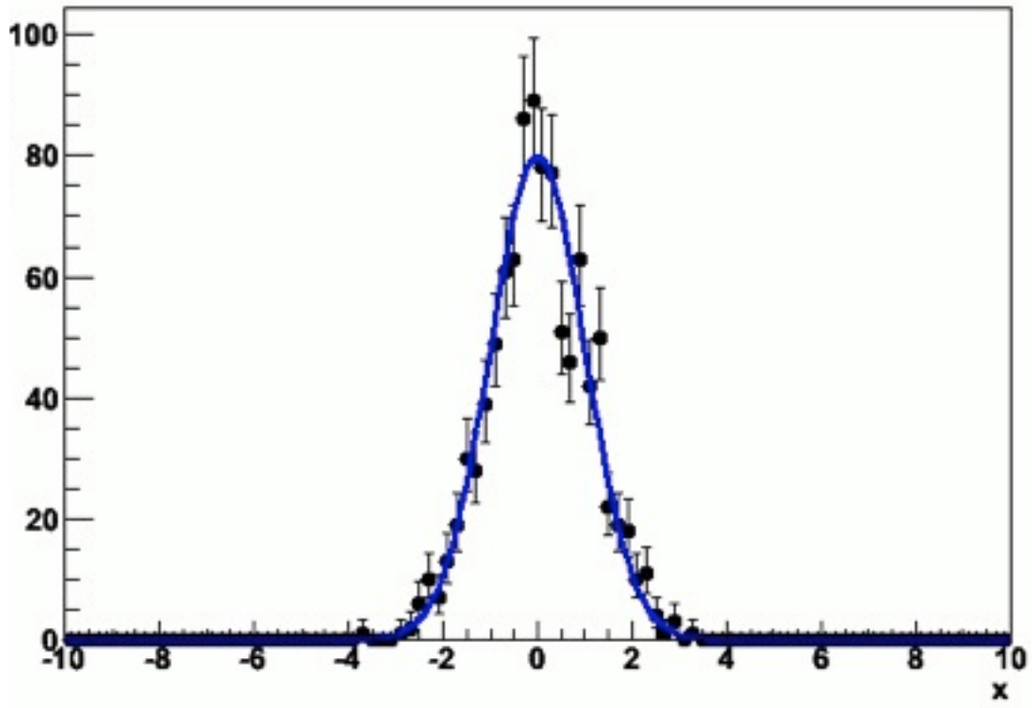
$$\mathscr{L} = \prod_{i=1}^{N} P\left(m_i, \tau_i | m_s, \tau_s\right) = \prod_{i=1}^{N} \left[f_s \cdot S_m \cdot S_\tau + \left(1 - f_s\right) \cdot B_m \cdot B_\tau\right]$$

RooStats

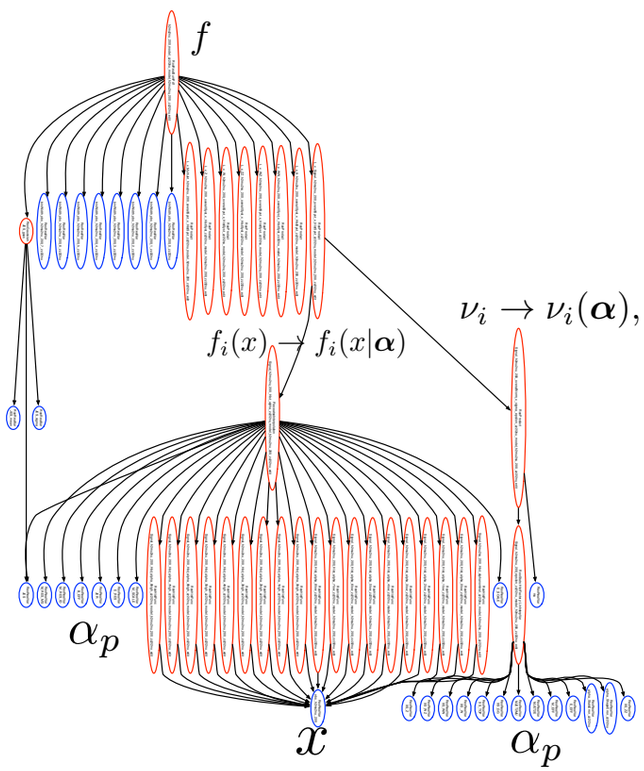
- Tools for higher-level stat. analysis

A simple model

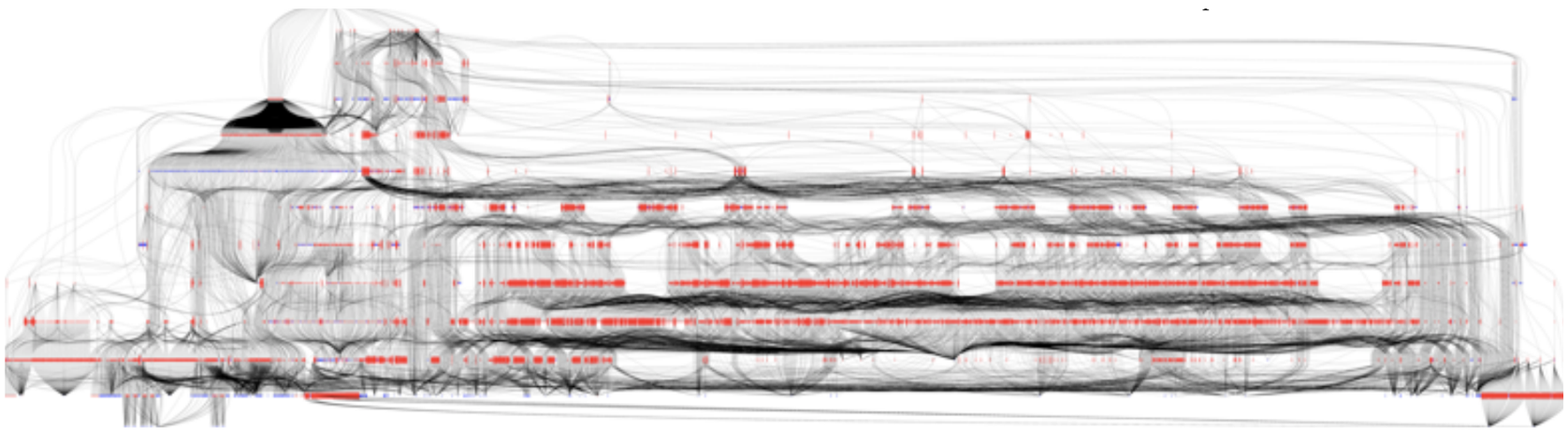




A simple realistic model



The model used in the combination of the Higgs results from various categories in various decay channels



Signal+background fit in RooStats

- Simple mass fit model with RooStats formalism

```
  using namespace RooFit;
  using namespace RooStats;

  RooWorkspace* wspace = new RooWorkspace("wspace");
  RooRealVar *m = wspace->factory("m[5000,5700]");

...

  wspace->factory("Gaussian:gauss1(m,mean[5278,5275,5285],sigma1[20,0,100])");
  wspace->factory("Gaussian:gauss2(m,mean,sigma2[80,40,200])");
  wspace->factory("SUM:signal_model(a[0.3,0,1]*gauss1,gauss2)");
  wspace->factory("Polynomial:bkg_model(m,b[0.0005,-0.001,0.001])");
  RooAbsPdf *model = wspace->factory(
"SUM:model(nsig[50000,0,500000]*signal_model,nbkg[10000,0,1000000]*bkg_model)");
```

- Based on the concept of RooWorkspace: the technology to record and share arbitrarily complex models in ".root" format ('*digital publishing*')
- Combining models from various workspaces one can perform combinations across channels, experiments, etc.

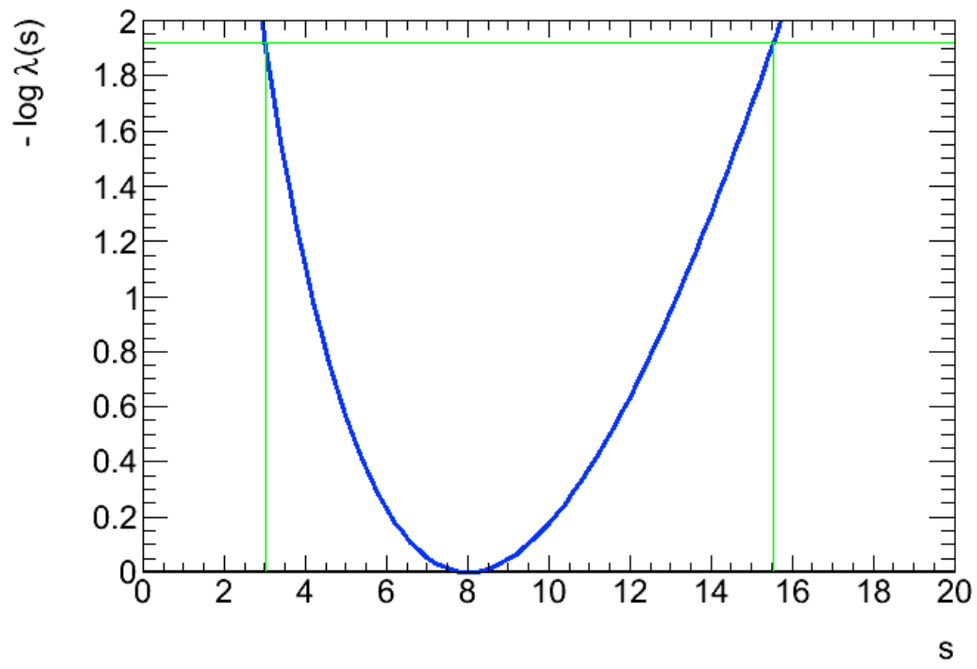
Confidence interval

- Confidence interval for simple counting experiment

*Observed 10 events, expect **exactly 2** background events.*

95% confidence interval for signal?

$$P(N_{obs}|s+b)$$




```
  wspace->factory("obs[10]");
  wspace->factory("Poisson::countingModel(obs,sum(s[8,0,20],b[2]))");

...

  ModelConfig modelConfig(new RooWorkspace());
  modelConfig.SetWorkspace(*wspace);
  modelConfig.SetPdf(*wspace->pdf("countingModel"));
  modelConfig.SetParametersOfInterest(*wspace->var("s"));

  ProfileLikelihoodCalculator plc(*data, modelConfig);
  plc.SetConfidenceLevel( 0.95 );
  LikelihoodInterval* plInt = plc.GetInterval();
```

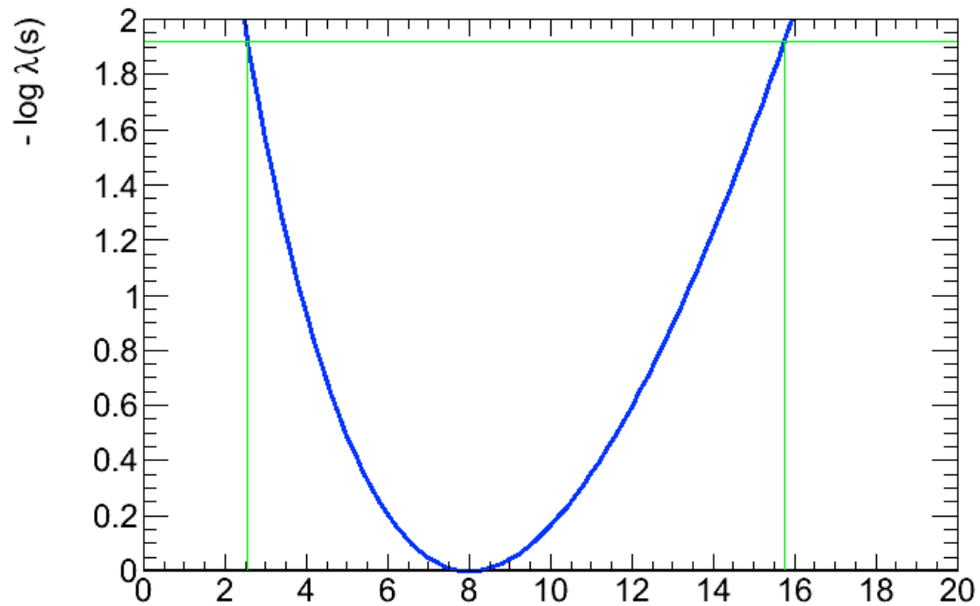
Including systematics

- Confidence interval for simple counting experiment

*Observed 10 events, expect **2 ± 1** background events.*

95% confidence interval for signal?

$$P(N_{obs}|s+b)\cdot\pi(b)$$




```
  wspace->factory("obs[10]");
  wspace->factory("Poisson::countingModel(obs, sum(s[8,0,20],b[2,1,3]))");
  wspace->factory("Gaussian::bkgConstraint(b,meanb[2],sigmab[1])");
  wspace->factory("PROD::modelWithConstraints(countingModel,bkgConstraint)");

...

  ModelConfig modelConfig(new RooWorkspace());
  modelConfig.SetWorkspace(*wspace);
  modelConfig.SetPdf(*wspace->pdf("modelWithConstraints"));
  modelConfig.SetParametersOfInterest(*wspace->var("s"));

  ProfileLikelihoodCalculator plc(*data, modelConfig);
  plc.SetConfidenceLevel( 0.95 );
  LikelihoodInterval* plInt = plc.GetInterval();
```

Profile likelihood

- Cross-section estimation with profile likelihood

We want to measure the cross-section given the number of observed signal

But the observed signal is a function of the cross-section!

$$\sigma = \frac{s}{L \cdot \varepsilon}$$

$$s = \sigma \cdot L \cdot \varepsilon$$

$$\mathcal{L}(\sigma) = P(N_{obs}|s+b) \cdot G(b, \delta b) \cdot G(L, \delta L) \cdot G(\varepsilon, \delta\varepsilon)$$