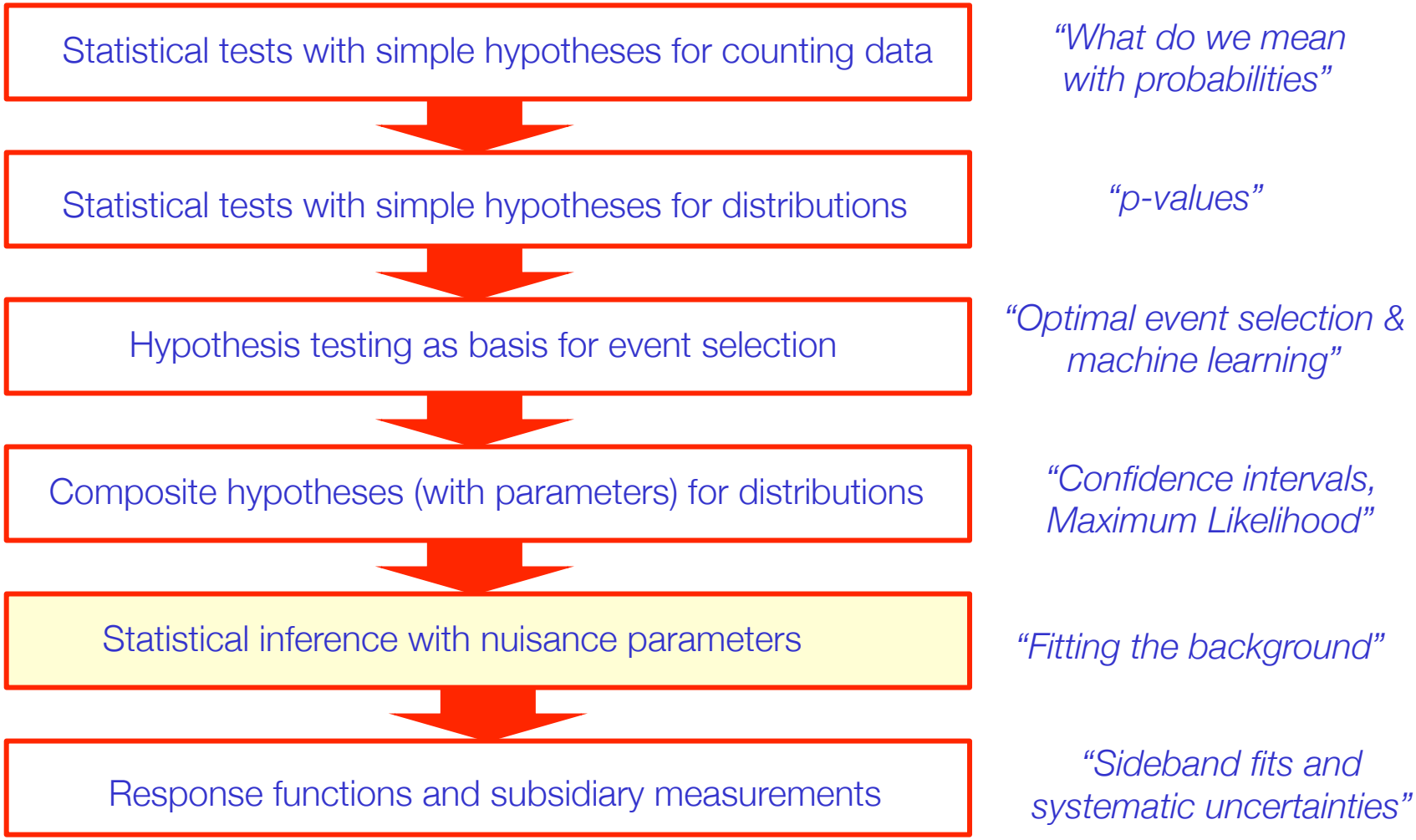
Next subject…

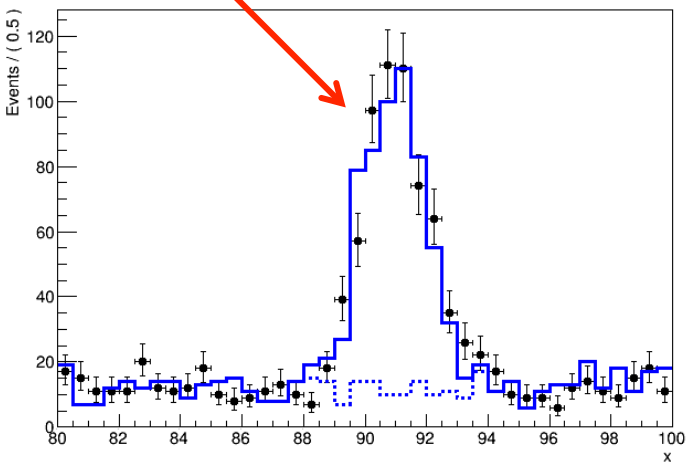
- Start with basics, gradually build up to complexity of

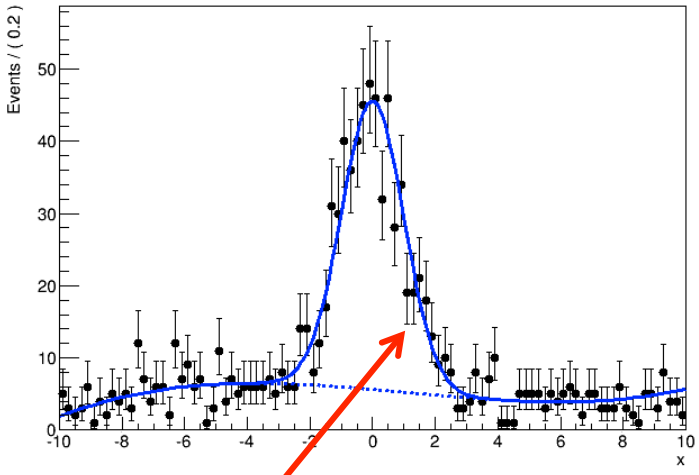


Adding parameters to the model

- We can describe uncertainties in our model by adding new parameters of which the value is uncertain

$$L(\vec{N} \mid \mu) = \prod_{bins} Poisson(N_i \mid \mu \cdot \tilde{s}_i + \tilde{b}_i)$$





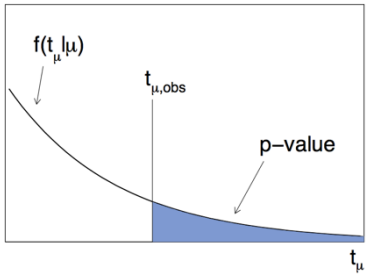
$$L(x \mid f, m, \sigma, a_0, a_1, a_2) = fG(x, m, \sigma) + (1 - f)Poly(x, a_0, a_1, a_2)$$

- These additional model parameters are not 'of interest', but we need them to model uncertainties → 'Nuisance parameters'

Treatment of NPs in hypothesis testing and conf. intervals

- We've covered frequentist hypothesis testing and interval calculation using likelihood ratios based on a likelihood with a single parameter (of interest) L(μ)
  - Result is p-value on hypothesis with given μ value, or
  - Result is a confidence interval $[\mu_-,\mu_+]$ with values of μ for which p-value is at or above a certain level (the confidence level)
- How do you do this with a likelihood L(μ,θ) where θ is a nuisance parameter?
  - With a test statistics q_μ, we calculate p-value for hypothesis θ as

$$p_\mu = \int_{q_{\mu,obs}}^{\infty} f(q_\mu \mid \mu,\theta)\,dq_\mu$$



- But what values of θ do we use for $f(q_\mu|\mu,\theta)$?
  Fundamentally, we want to reject μ only if p<α for all θ
  → Exact confidence interval

The profile likelihood ratio

- With this prescription we can construct the profile likelihood ratio as test statistic

Likelihood for given μ

Maximum Likelihood for given μ

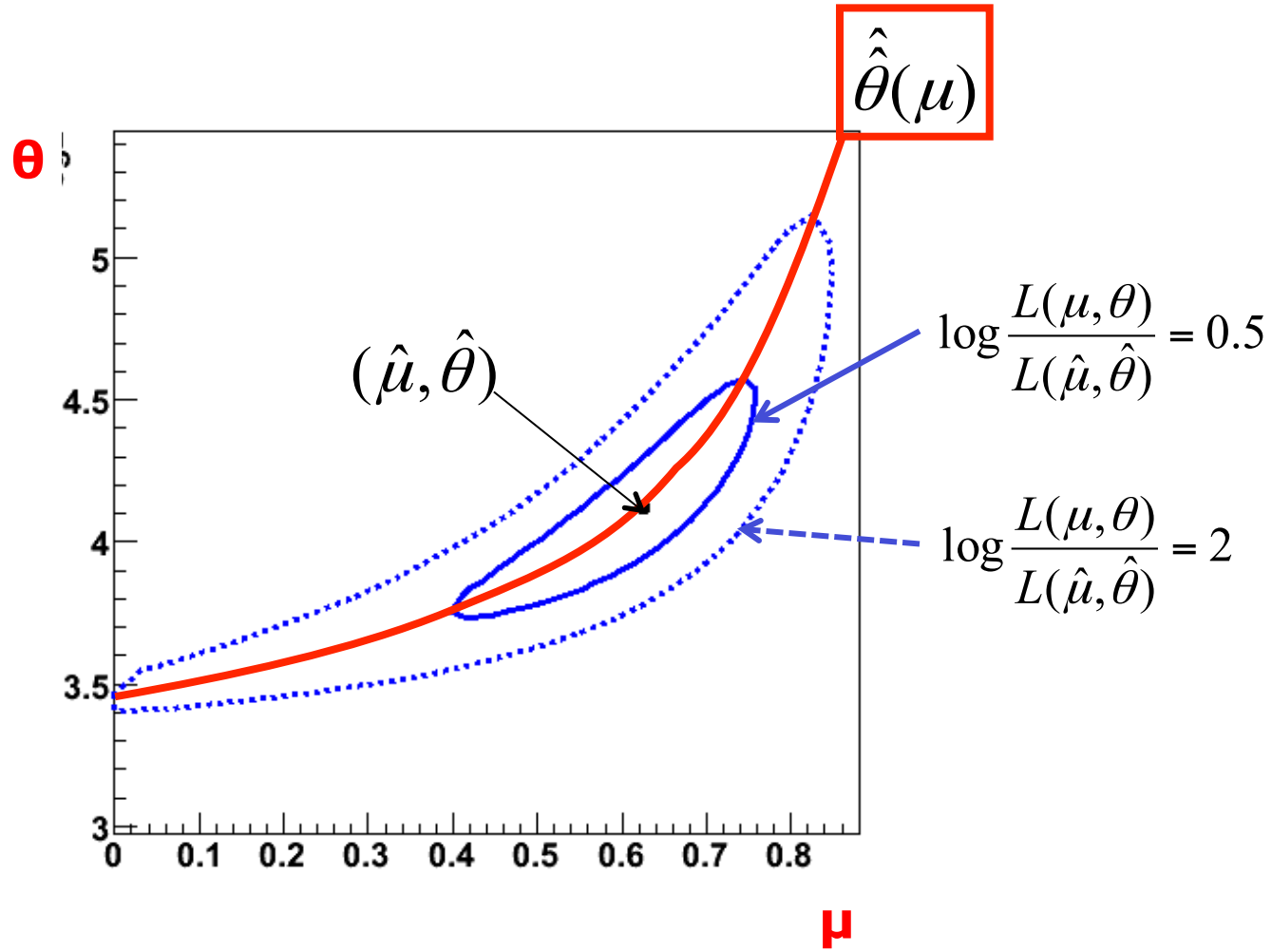
$$\lambda(\mu) = \frac{L(\mu)}{L(\hat{\mu})} \Rightarrow \lambda(\mu) = \frac{L(\mu, \hat{\hat{\theta}}(\mu))}{L(\hat{\mu}, \hat{\theta})}$$

Maximum Likelihood

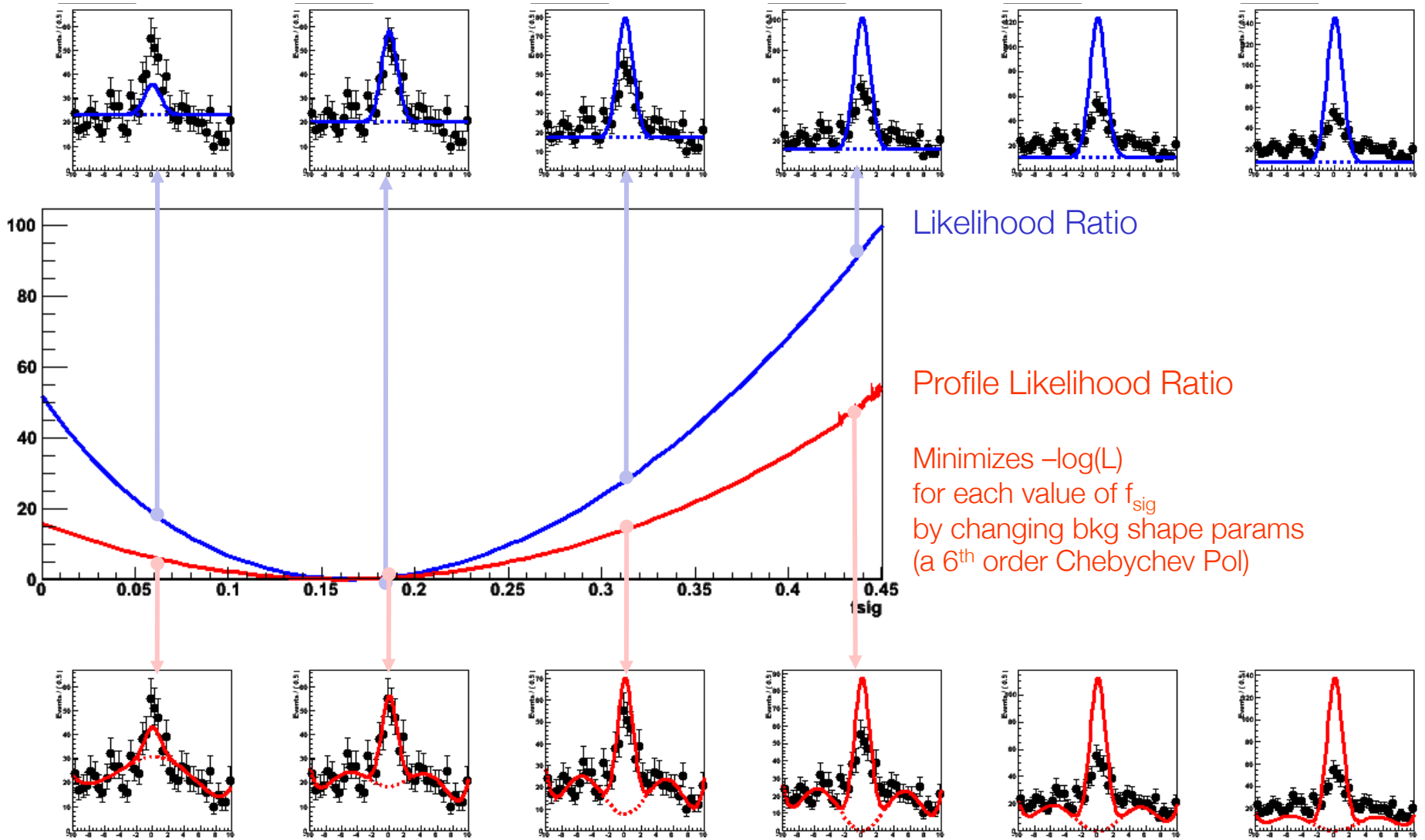
Maximum Likelihood

- NB: value profile likelihood ratio does *not* depend on θ

Profiling illustration with one nuisance parameter



Profile scan of a Gaussian plus Polynomial probability model

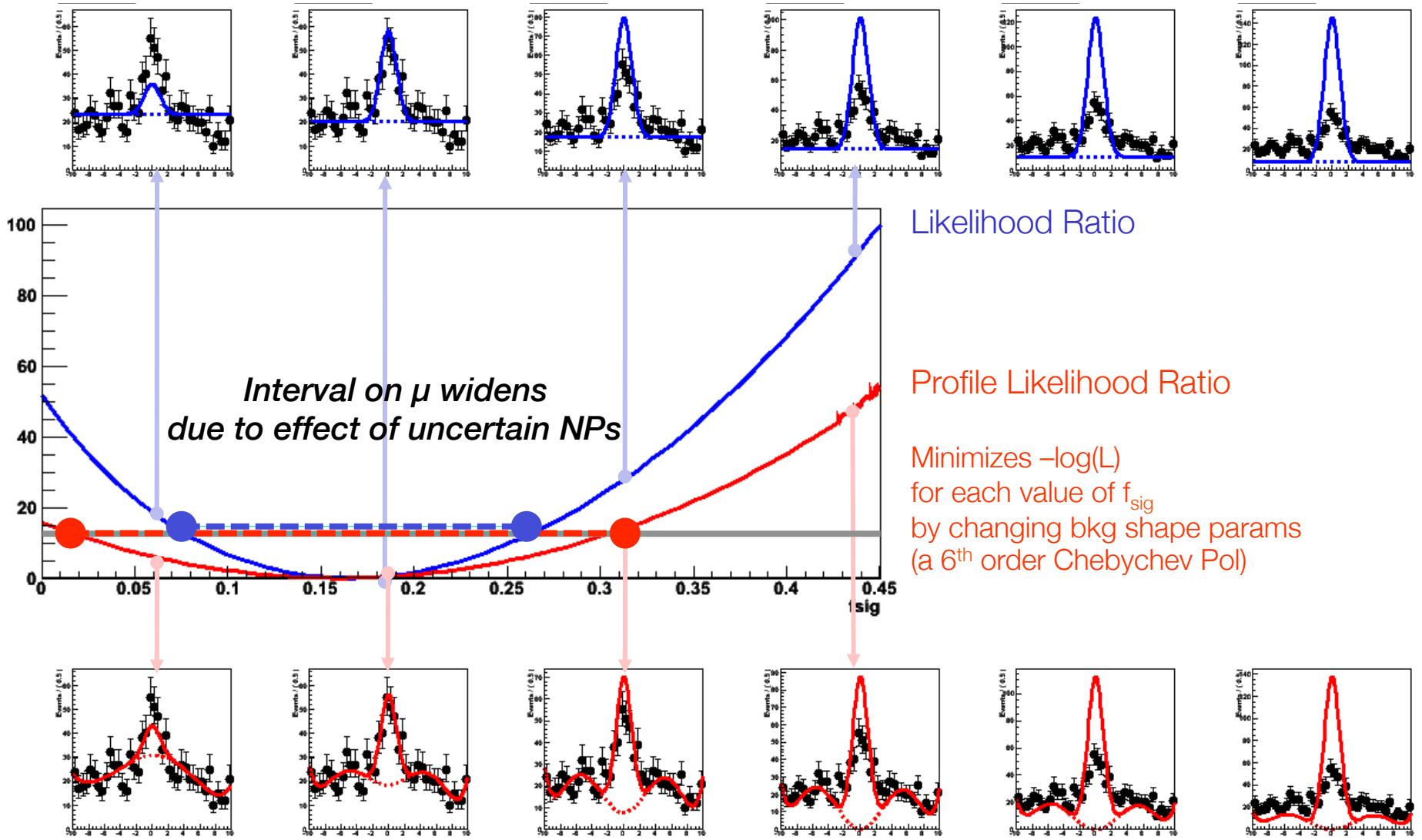


Likelihood Ratio

Profile Likelihood Ratio

Minimizes –log(L)
for each value of f_{sig}
by changing bkg shape params
(a 6th order Chebychev Pol)

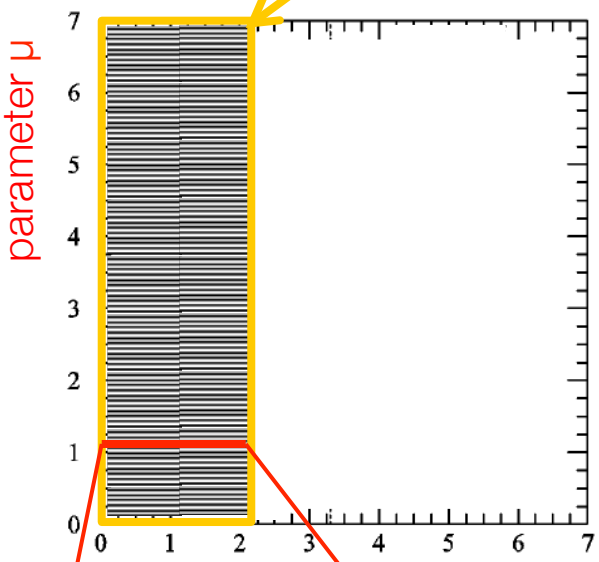
Profile scan of a Gaussian plus Polynomial probability model

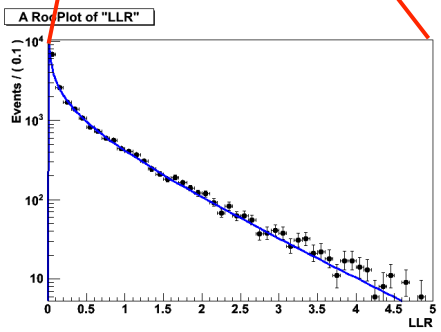


PLR Confidence interval vs MINOS

$t_\mu(x,\mu)$

Confidence belt now range in PLR

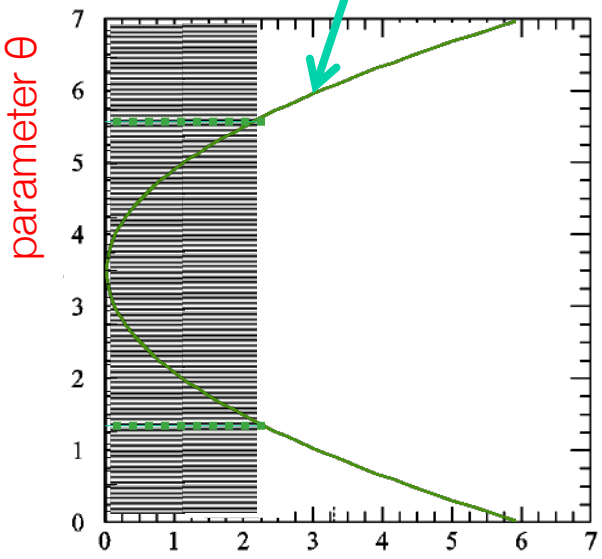




Asymptotically, distribution is identical for all μ

Measurement = $t_\mu(x_{obs},\mu)$ is now a function of μ

$t_\mu(x,\mu)$

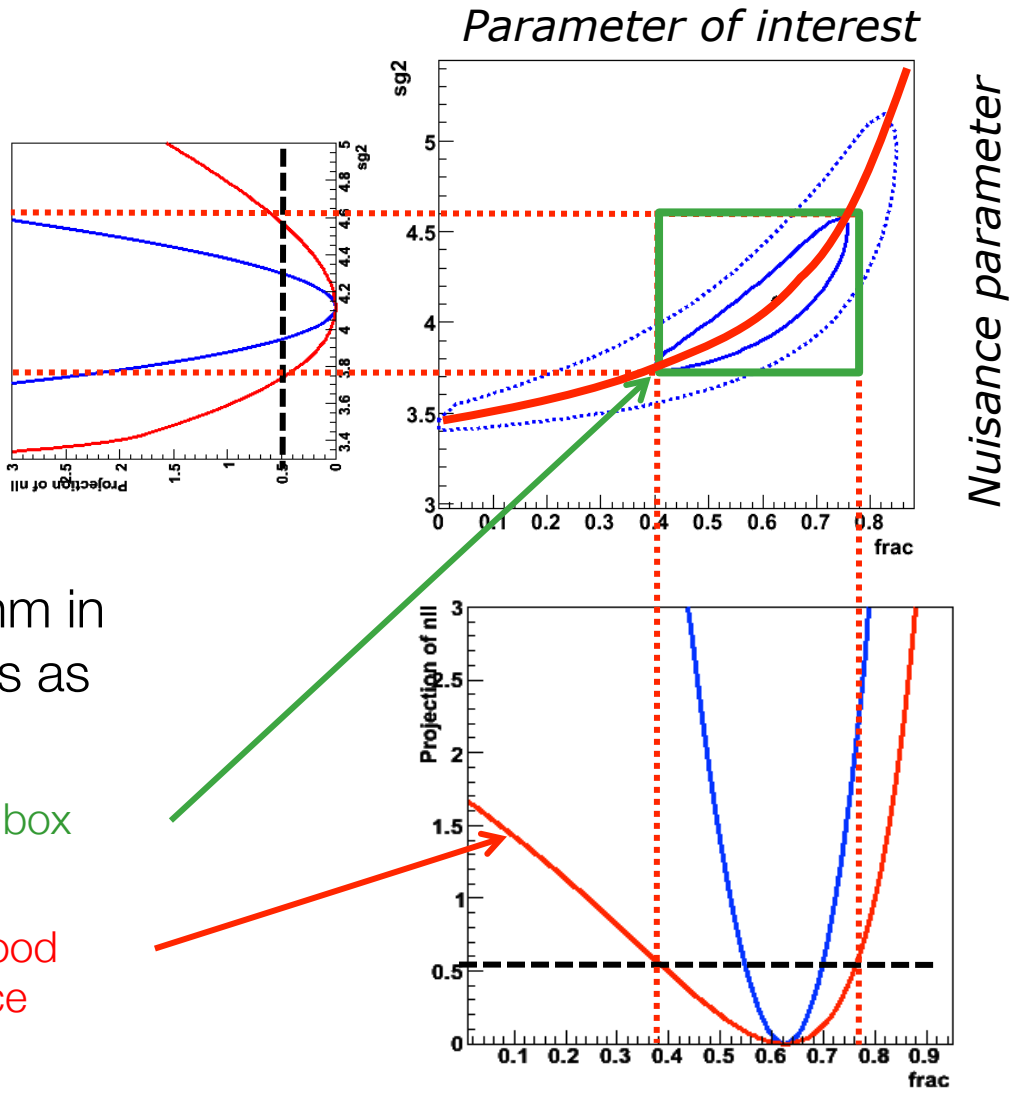


NB: asymptotically, distribution is also independent of true values of θ

$$f(t_\mu;\Lambda) = \frac{1}{2\sqrt{t_\mu}}\frac{1}{\sqrt{2\pi}}\left[\exp\left(-\frac{1}{2}\left(\sqrt{t_\mu}+\sqrt{\Lambda}\right)^2\right)+\exp\left(-\frac{1}{2}\left(\sqrt{t_\mu}-\sqrt{\Lambda}\right)^2\right)\right]$$

$$\Lambda = \frac{(\mu-\mu')^2}{\sigma^2}\,.$$

Link between MINOS errors and profile likelihood



- Note that MINOS algorithm in MINUIT gives same errors as Profile Likelihood Ratio
  - MINOS errors is bounding box around λ(s) contour
  - Profile Likelihood = Likelihood minimized w.r.t. all nuisance parameters

NB: Similar to graphical interpretation of variance estimators, but those always assume an elliptical contour from a perfectly parabolic likelihood

Summary on NPs in confidence intervals

- Exact confidence intervals are difficult with nuisance parameters
  - Interval should cover for any value of nuisance parameters
  - Technically difficult and significant over-coverage common
- LHC solution Profile Likelihood ratio → Guaranteed coverage at *measured* values of nuisance parameters only
  - Technically replace likelihood ratio with profile likelihood ratio
  - Computationally more intensive (need to minimize likelihood w.r.t all nuisance parameters for each evaluation of the test statistic), but still very tractable
- Asymptotically confidence intervals constructed with profile likelihood ratio test statistics correspond to (MINOS) likelihood ratio intervals
  - As distribution of profile likelihood becomes asymptotically independent of θ, coverage for all values of θ restored

Dealing with nuisance parameters in Bayesian intervals

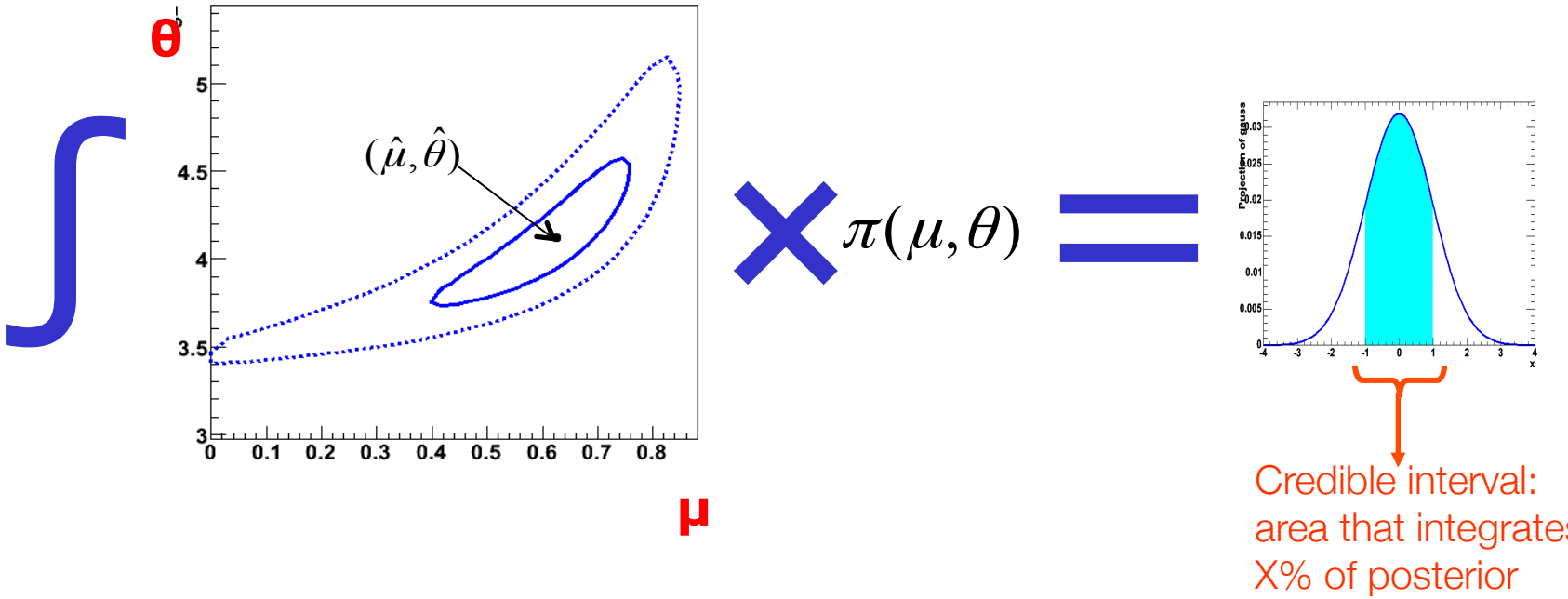
- Elimination of nuisance parameters in Bayesian interval: Integrate over the full subspace of all nuisance parameters;

$$P(\mu \mid x) \propto L(x \mid \mu) \cdot \pi(\mu)$$

$$\Downarrow$$

$$P(\mu \mid x) \propto \int \left(L(x \mid \mu, \vec{\theta}) \pi(\mu) \pi(\vec{\theta}) \right) d\vec{\theta}$$

- You are left with posterior pdf for μ



Computational aspects of dealing with nuisance parameters

- Dealing with many nuisance parameters is computationally intensive in both Bayesian and (LHC) Frequentist approach
- Profile Likelihood approach
  - Computational challenge = ***Minimization*** *of likelihood w.r.t. all nuisance parameters for every point in the profile likelihood curve*
  - Minimization can be a difficult problem,
    e.g. if there are strong correlations, or multiple minima
- Bayesian approach
  - Computational challenge = ***Integration*** *of posterior density of all nuisance parameters*
  - Requires sampling of very potentially very large space.
  - Markov Chain MC and importance sampling techniques can help, but still very CPU consuming

Other procedures that have been tried*

- Hybrid Frequentist-Bayesian approach ('Cousins-Highland / Z_N')
  - Integrate likelihood over nuisance parameters

$$L_m(\mu) = \int \left(L(\mu, \vec{\theta}) \pi(\vec{\theta}) \right) d\vec{\theta}$$

  - Then treat integrated L_m as test statistic → obtain p-value from its distribution
  - In practice integral is performed using MC integration, so often described as a 'sampling method'

$$L_m(\mu) = \frac{1}{N} \sum_{MC} L(\mu, \vec{\theta}_i) \pi(\vec{\theta}_i)$$

  - Method has been shown to have bad coverage
- Ad-hoc sampling methods of various types.
  - Usually amount to either MC integration or fancy error propagation

    Note that sampling the conditional estimator $\hat{\hat{\mu}}\big|_{\theta}$ over sample of θ values obtained from π(θ) is just glorified error propagation!

* But are known to have problems

How much do answers differ between methods?

A Prototype Problem

BROOKHAVEN NATIONAL LABORATORY

What is significance Z of an observation $x = 178$ events in a signal like region, if my expected background $b = 100$ with a 10% uncertainty?

- if you use the ATLAS TDR formula $Z_{5'}=5.5$
- if you use Cousins-Highland $Z_N=5.0$

The question seems simple enough, but it is not actually well-posed

- what do I mean by 10% background uncertainty?

Typically, we consider an auxiliary measurement y used to estimate background (Type I systematic)

- eg: a sideband counting experiment where backgrou… in sideband is a factor τ bigger than in signal region

$$L_P(x, y|\mu, b) = Pois(x|\mu + b) \cdot Pois(y|\tau b).$$

Kyle Cranmer (BNL) PhyStat 2007, CERN, June 26, 2007

These slide discuss
a 'prototype' likelihood
that statisticians like:

$\text{Poisson}(N_{sig}|s+b) \cdot \text{Poisson}(N_{ctl}|\tau \cdot b)$

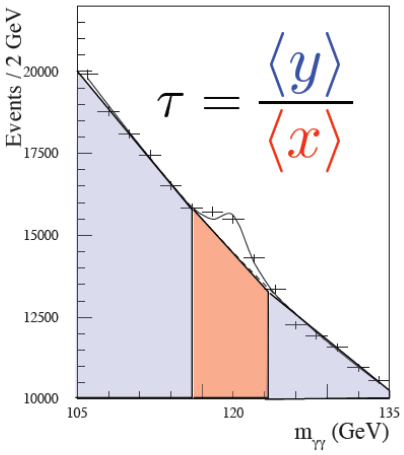
NB: This is one of the very few
problems with nuisance parameters
with can be exactly calculation

Example Sideband Measurement

BROOKHAVEN NATIONAL LABORATORY

Sideband measurement used to extrapolate / interpolate the background rate in signal-like region

For now ignore uncertainty in extrapolation.



$$L_P(x, y|\mu, b) = Pois(x|\mu + b) \cdot Pois(y|\tau b).$$

Kyle Cranmer (BNL) PhyStat 2007, CERN, June 26, 2007 14

Recent comparisons results from PhyStat 2007

Comparison of Methods for Prototype Problem

BROOKHAVEN NATIONAL LABORATORY

In my contribution to PhyStat2005, I considered this problem and compared the coverage for several methods

- See Linnemann's PhyStat03 paper

Major results:

- Cousins-Highland result (Z_N) badly under-covers (only 4.2σ)!
  - rate of Type I error is 110 times higher than stated!
  - much less luminosity required

- Profile Likelihood Ratio (MINUIT/ MINOS) works great out to 5σ!

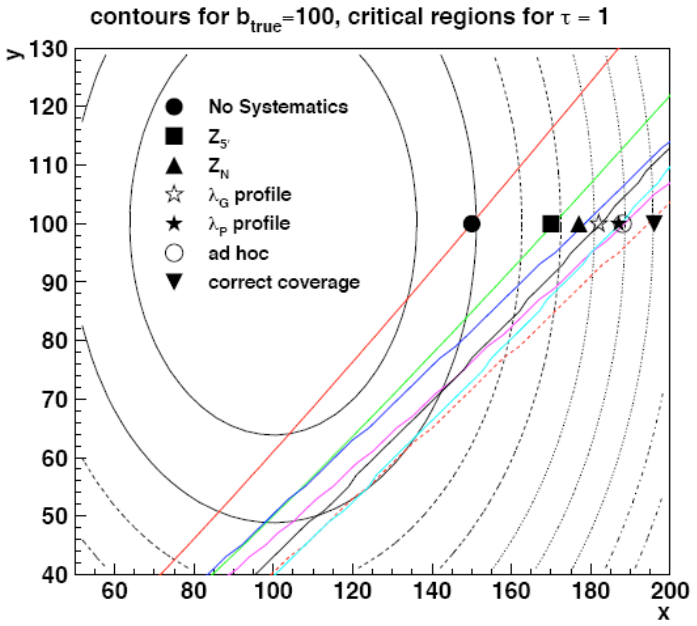


Figure 7. A comparison of the various methods critical boundary $x_{crit}(y)$ (see text). The concentric ovals represent contours of L_G from Eq. 15.

Method	L_G ($Z\sigma$)	L_P ($Z\sigma$)	$x_{crit}(y = 100)$
No Syst	3.0	3.1	150
$Z_{5'}$	4.1	4.1	171
Z_N (Sec. 4.1)	4.2	4.2	178
ad hoc	4.6	4.7	188
$Z_\Gamma = Z_{Bi}$	4.9	5.0	185
profile λ_P	5.0	5.0	185
profile λ_G	4.7	4.7	~182

Exact solution

Summary of statistical treatment of nuisance parameters

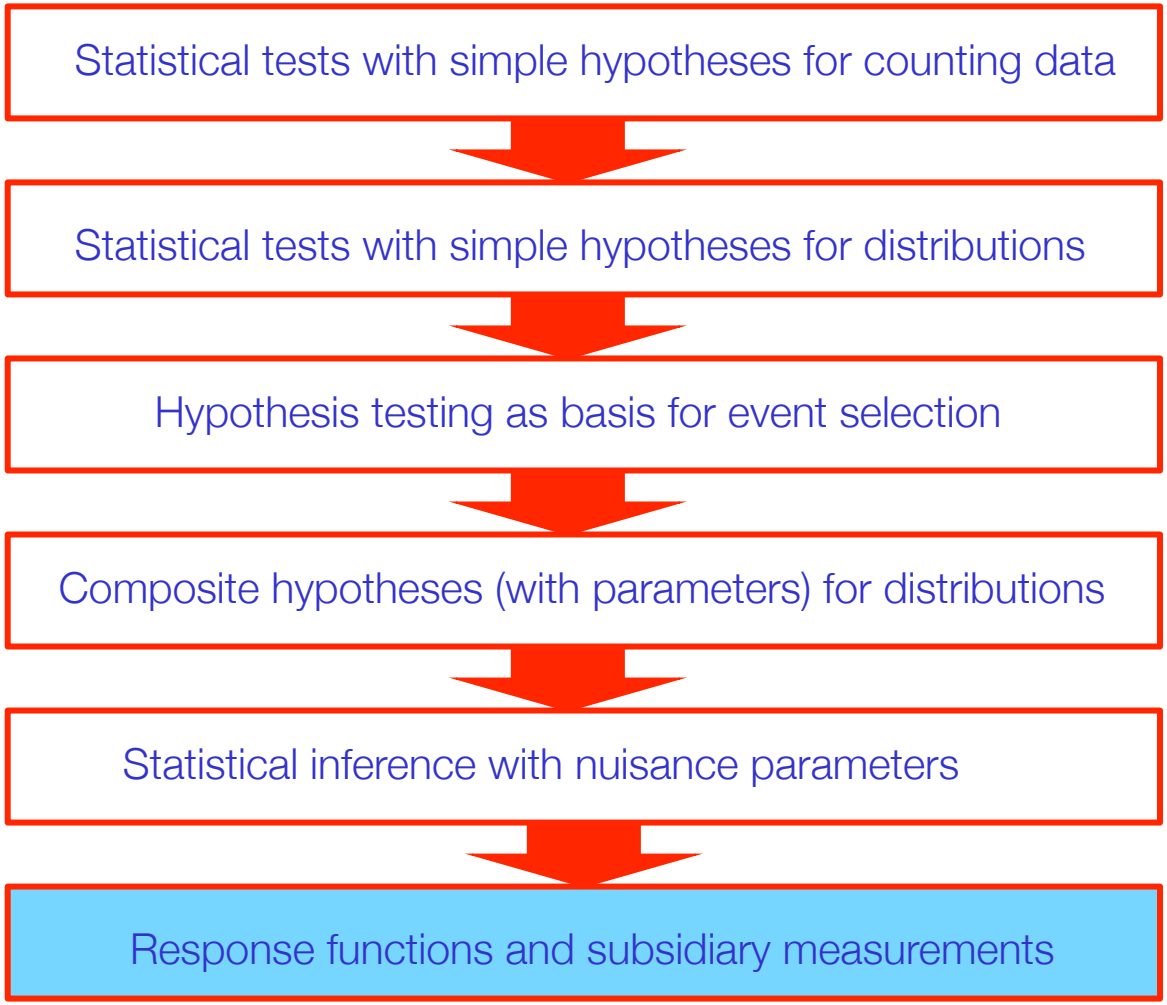
- Each statistical method has an associated technique to propagate the effect of uncertain NPs on the estimate of the POI
  - Parameter estimation → Joint unconditional estimation
  - Variance estimation → Replace d^2L/dp^2 with Hessian matrix
  - Hypothesis tests & confidence intervals → Use profile likelihood ratio
  - Bayesian credible intervals → Integration ('Marginalization')

- Be sure to use the right procedure with the right method
  - Anytime you integrate a Likelihood you are a Bayesian
  - If you are minimizing the likelihood you are usually a Frequentist
  - If you sample something chances are you performing either a (Bayesian) Monte Carlo integral, or are doing glorified error propagation

- Answers can differ substantially between methods!
  - This is not always a problem, but can also be a consequence of a difference in the problem statement

Overview

- Start with basics, gradually build up to complexity of



"What do we mean with probabilities"

"p-values"

"Optimal event selection & machine learning"

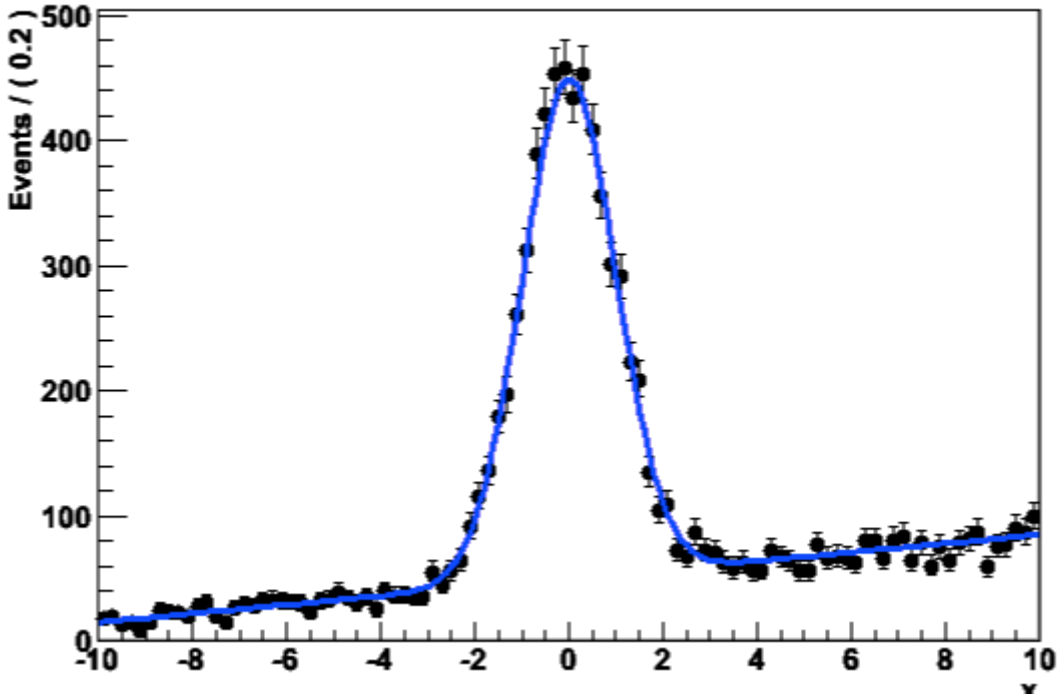
"Confidence intervals, Maximum Likelihood"

"Fitting the background"

"Profile Likelihood fits"

RooFit – Focus: coding likelihood functions

- Focus on one practical aspect of many data analysis in HEP: How do you formulate your likelihood functions in ROOT
  - For ‘simple’ problems (gauss, polynomial) this is easy



  - But if you want to do unbinned ML fits, use non-trivial functions, or work with multidimensional functions you quickly find that you need some tools to help you

RooFit core design philosophy

- Mathematical objects are represented as C++ objects

Mathematical concept		RooFit class
variable	x	`RooRealVar`
function	$f(x)$	`RooAbsReal`
PDF	$f(x)$	`RooAbsPdf`
space point	$\vec{x}$	`RooArgSet`
integral	$\int_{x_{\min}}^{x_{\max}} f(x)dx$	`RooRealIntegral`
list of space points		`RooAbsData`

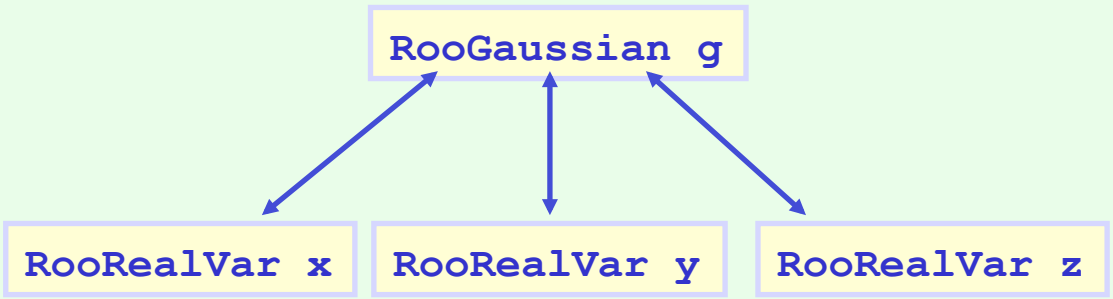
RooFit core design philosophy - Workspace

- Instead of **'double Likelihood(double paramVec[])'**, a flexible modular structure of 'programmed' functions

Math

$$\text{Gauss}(x,\mu,\sigma)$$

RooFit diagram



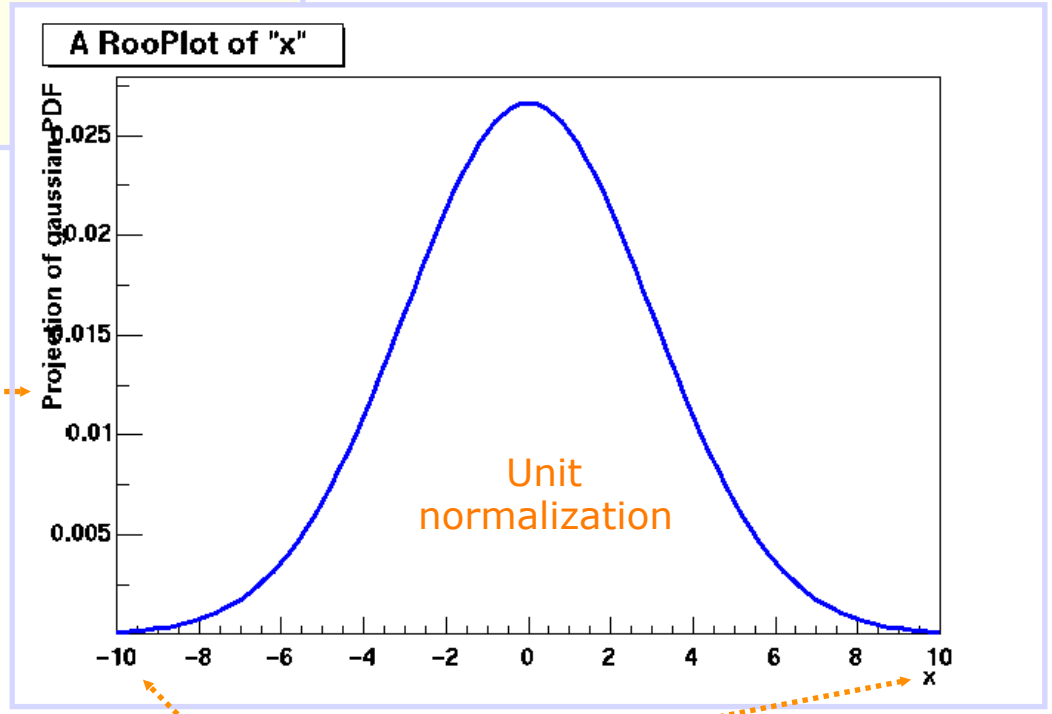
RooFit code

```
RooRealVar x("x","x",-10,10) ;
RooRealVar m("m","y",0,-10,10) ;
RooRealVar s("s","z",3,0.1,10) ;
RooGaussian g("g","g",x,m,s) ;
```

Basics – Creating and plotting a Gaussian p.d.f

Setup gaussian PDF and plot

```
// Create an empty plot frame
RooPlot* xframe = w::x.frame() ;

// Plot model on frame
model.plotOn(xframe) ;

// Draw frame on canvas
xframe->Draw() ;
```



Axis label from `gauss` title

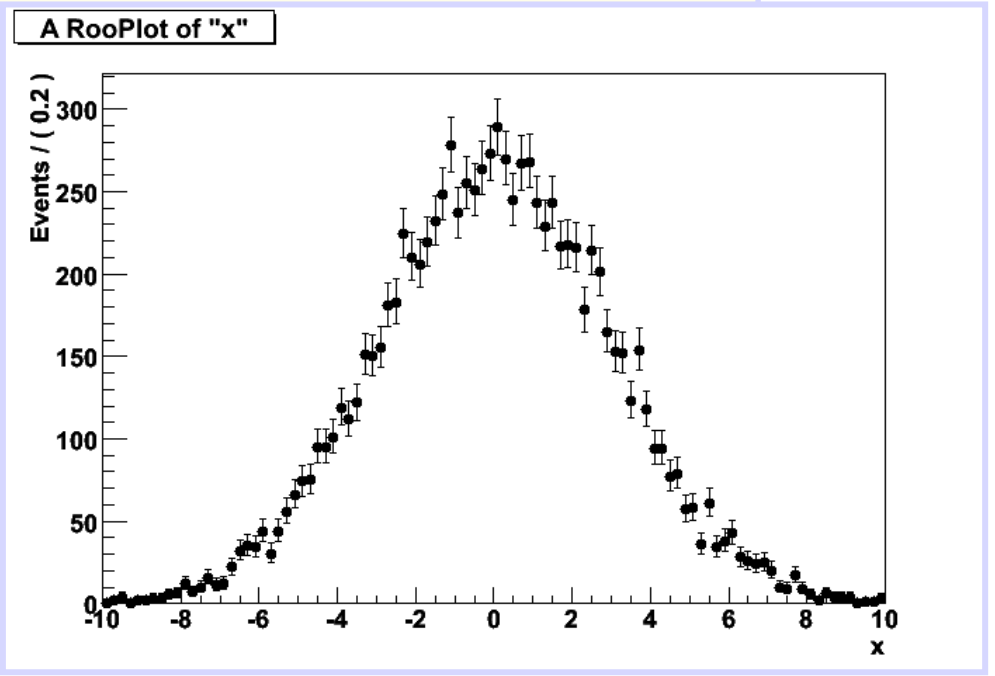
A `RooPlot` is an empty frame capable of holding anything plotted versus it variable

Plot range taken from limits of `x`

Basics – Generating toy MC events

Generate 10000 events from Gaussian p.d.f and show distribution

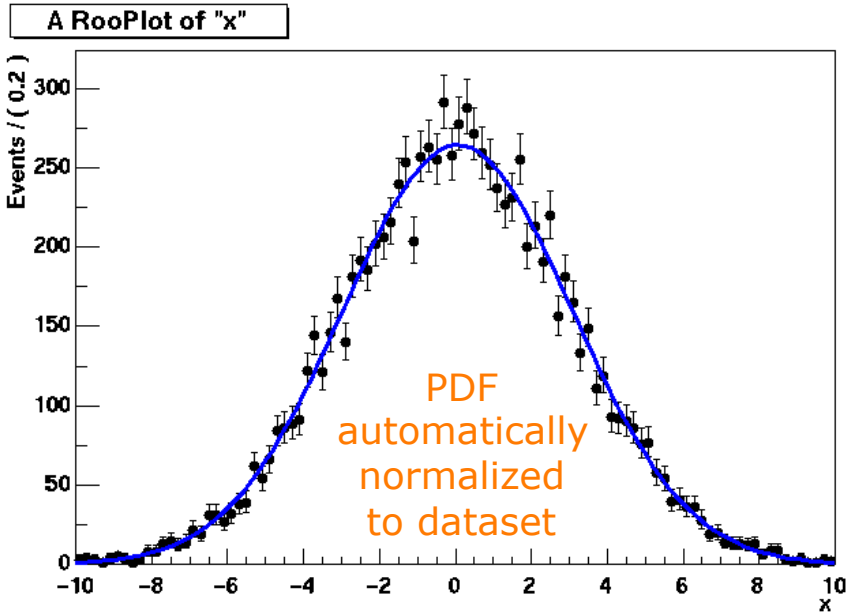
```
// Generate an unbinned toy MC set
RooDataSet* data = w::gauss.generate(w::x,10000) ;

// Generate an binned toy MC set
RooDataHist* data = w::gauss.generateBinned(w::x,10000) ;

// Plot PDF
RooPlot* xframe = w::x.frame()
data->plotOn(xframe) ;
xframe->Draw() ;
```



Can generate both binned and unbinned datasets

Basics – ML fit of p.d.f to *unbinned* data

```
// ML fit of gauss to data
w::gauss.fitTo(*data) ;
(MINUIT printout omitted)

// Parameters if gauss now
// reflect fitted values
w::mean.Print()
RooRealVar::mean = 0.0172335 +/- 0.0299542
w::sigma.Print()
RooRealVar::sigma = 2.98094  +/- 0.0217306

// Plot fitted PDF and toy data overlaid
RooPlot* xframe = w::x.frame() ;
data->plotOn(xframe) ;
w::gauss.plotOn(xframe) ;
```



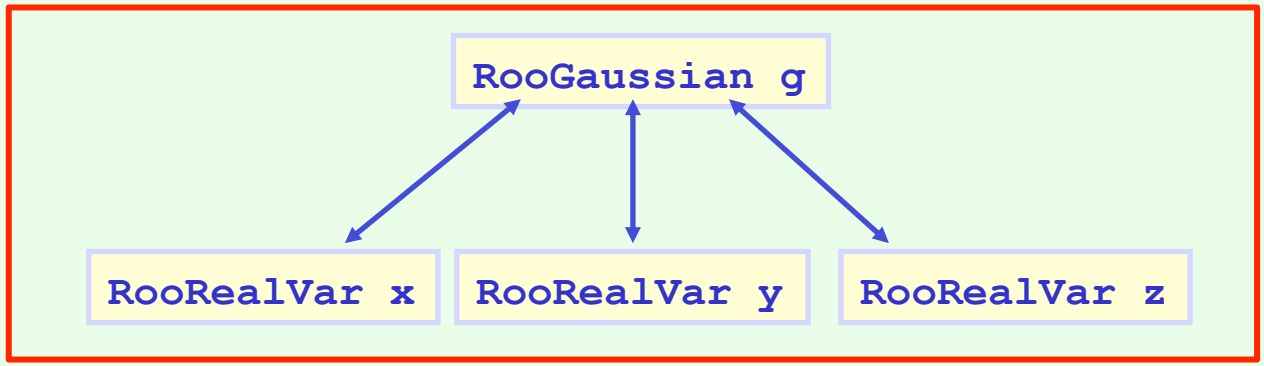
RooFit core design philosophy - Workspace

- The workspace serves a container class for all objects created

Math

$$\text{Gauss}(x,\mu,\sigma)$$

RooFit diagram

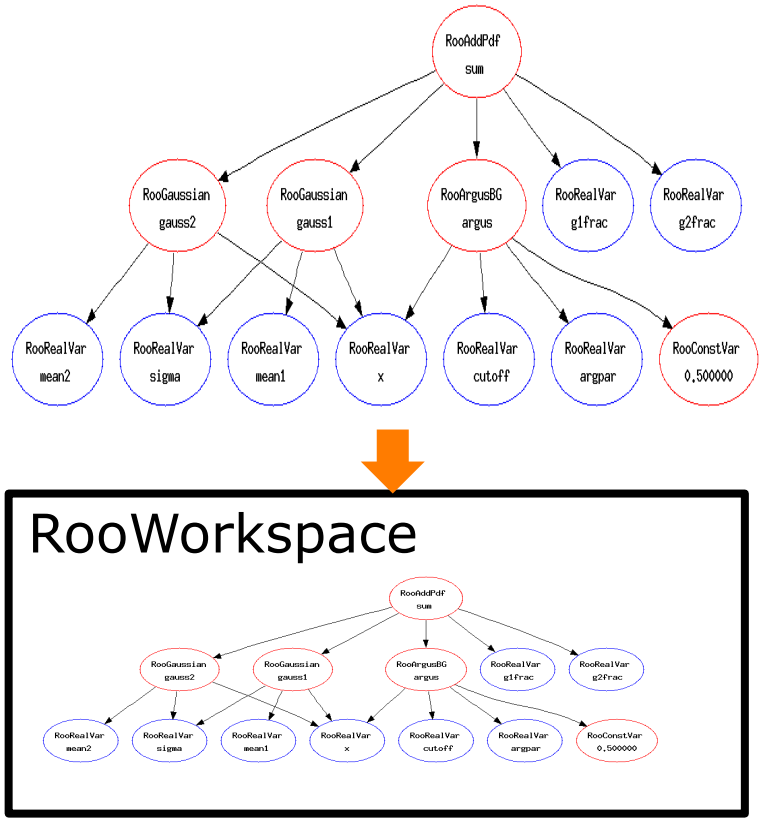


RooFit code

```
RooRealVar x("x","x",-10,10) ;
RooRealVar m("m","y",0,-10,10) ;
RooRealVar s("s","z",3,0.1,10) ;
RooGaussian g("g","g",x,m,s) ;
RooWorkspace w("w") ;
w.import(g) ;
```

The workspace

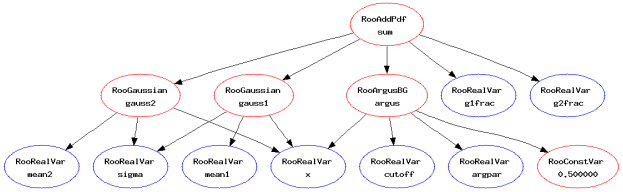
- The workspace concept has revolutionized the way people share and combine analysis
  - Completely factorizes process of building and using likelihood functions
  - You can give somebody an analytical likelihood of a (potentially very complex) physics analysis in a way to the easy-to-use, provides introspection, and is easy to modify.

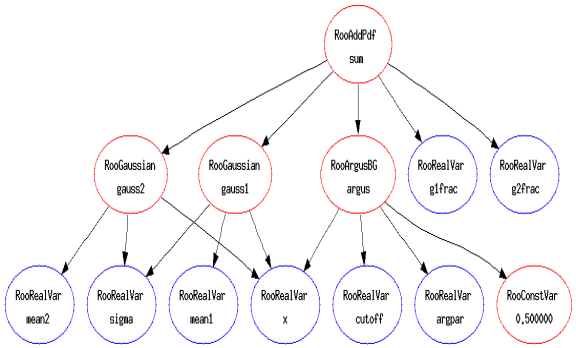



```
RooWorkspace w("w") ;
w.import(sum) ;
w.writeToFile("model.root") ;
```

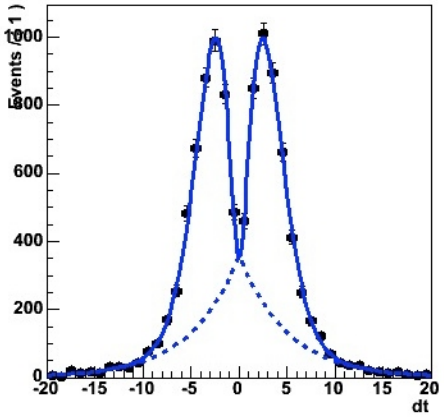
Using a workspace

RooWorkspace



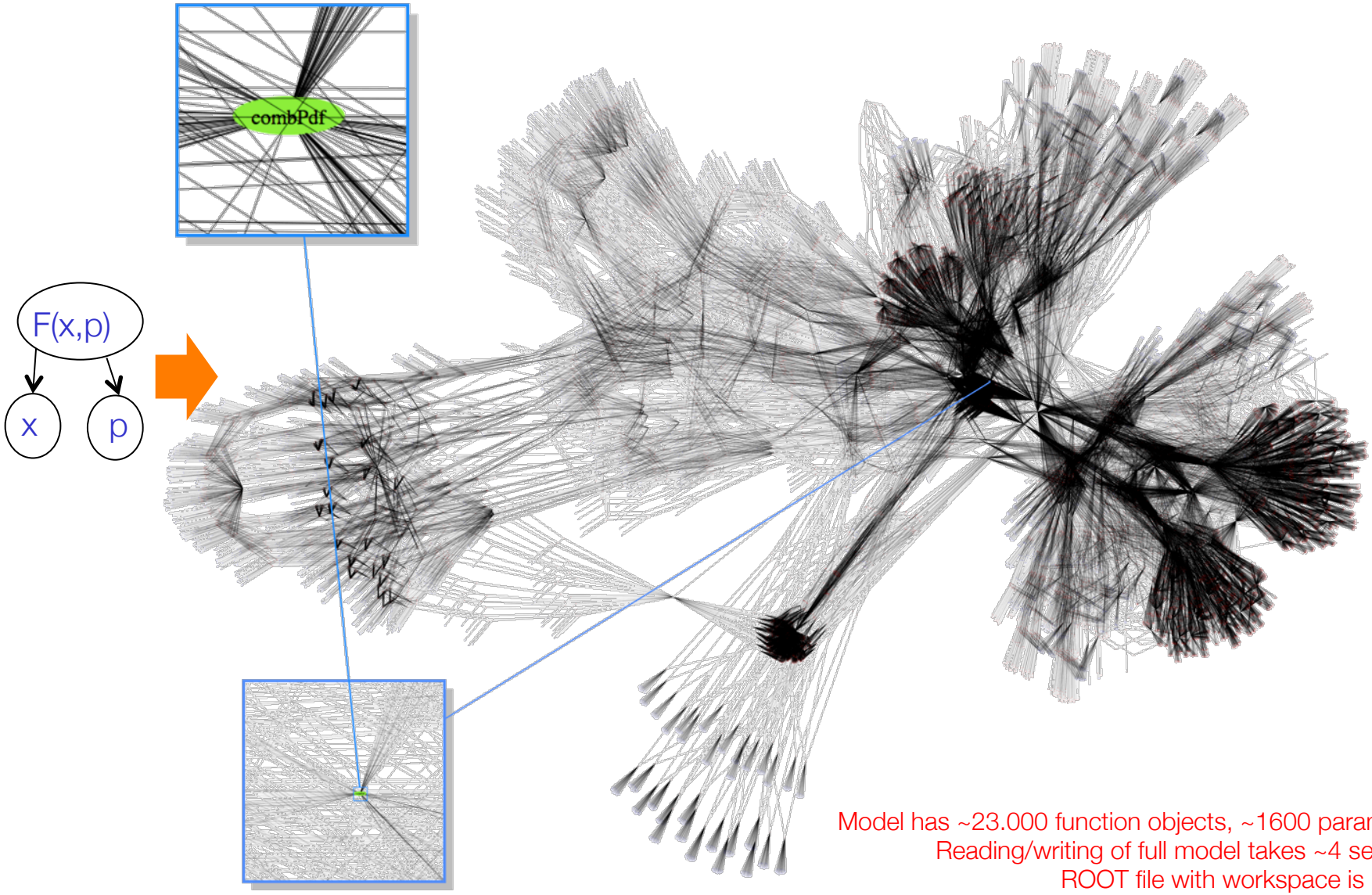



```
// Resurrect model and data
TFile f("model.root") ;
RooWorkspace* w = f.Get("w") ;
RooAbsPdf* model = w->pdf("sum") ;
RooAbsData* data = w->data("xxx") ;

// Use model and data
model->fitTo(*data) ;

RooPlot* frame =
          w->var("dt")->frame() ;
data->plotOn(frame) ;
model->plotOn(frame) ;
```



Workspace persistence of *really* complex models works too!

Atlas Higgs combination model (23.000 functions, 1600 parameters)



Model has ~23.000 function objects, ~1600 parameters
Reading/writing of full model takes ~4 seconds
ROOT file with workspace is ~6 Mb

Factory and Workspace

- *One C++ object per math symbol* provides ultimate level of control over each objects functionality, but results in lengthy user code for even simple macros
- Solution: add factory that auto-generates objects from a math-like language. Accessed through factory() method of workspace
- Example: reduce construction of Gaussian pdf and its parameters from 4 to 1 line of code

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean",5) ;
RooRealVar sigma("sigma","sigma",3)  ;
RooGaussian f("f","f",x,mean,sigma) ;
w.import(f) ;
```

```
w.factory("Gaussian::f(x[-10,10],mean[5],sigma[3])") ;
```

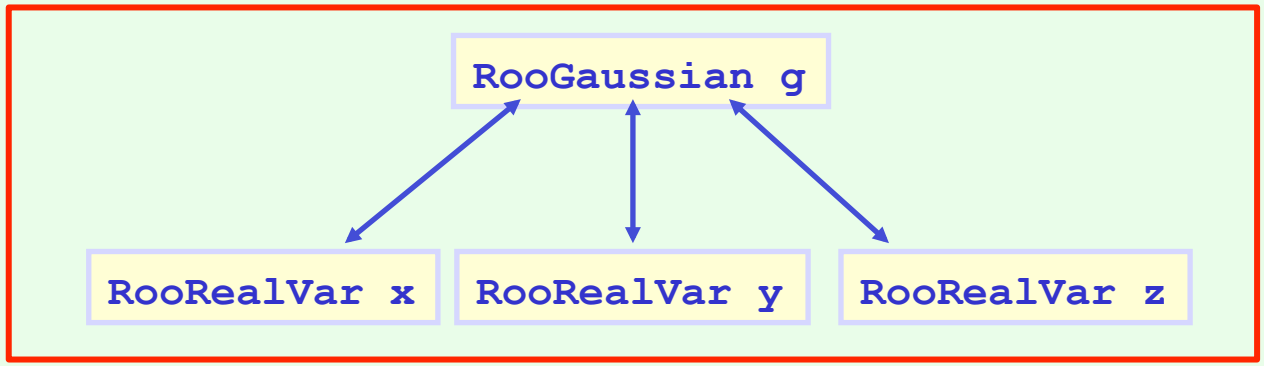
RooFit core design philosophy - Workspace

- The workspace serves a container class for all objects created

Math

$$\text{Gauss}(x,\mu,\sigma)$$

RooFit diagram



RooFit code

```
RooRealVar x("x","x",-10,10) ;
RooRealVar m("m","y",0,-10,10) ;
RooRealVar s("s","z",3,0.1,10) ;
RooGaussian g("g","g",x,m,s) ;
RooWorkspace w("w") ;
w.import(g) ;
```

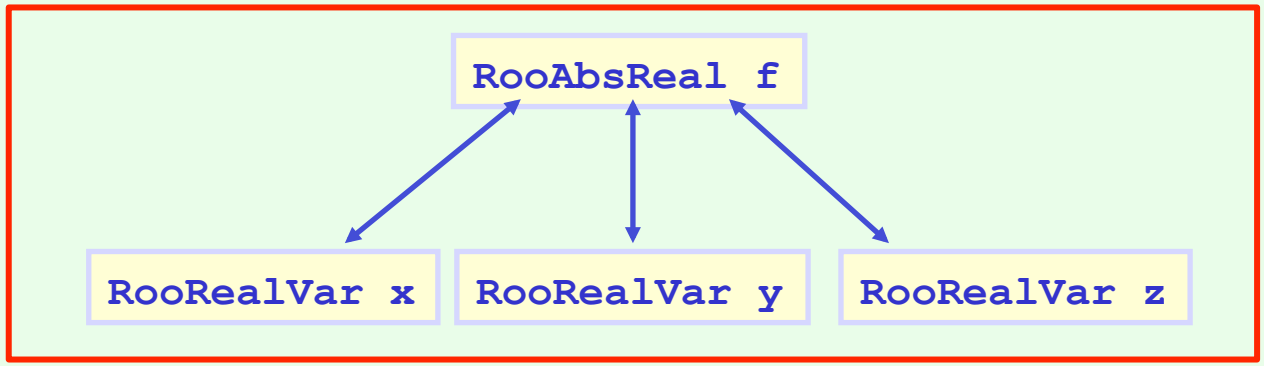
Populating a workspace the easy way – "the factory"

- The factory allows to fill a workspace with pdfs and variables using a simplified scripting language

Math

$$\text{Gauss}(x,\mu,\sigma)$$

RooFit diagram

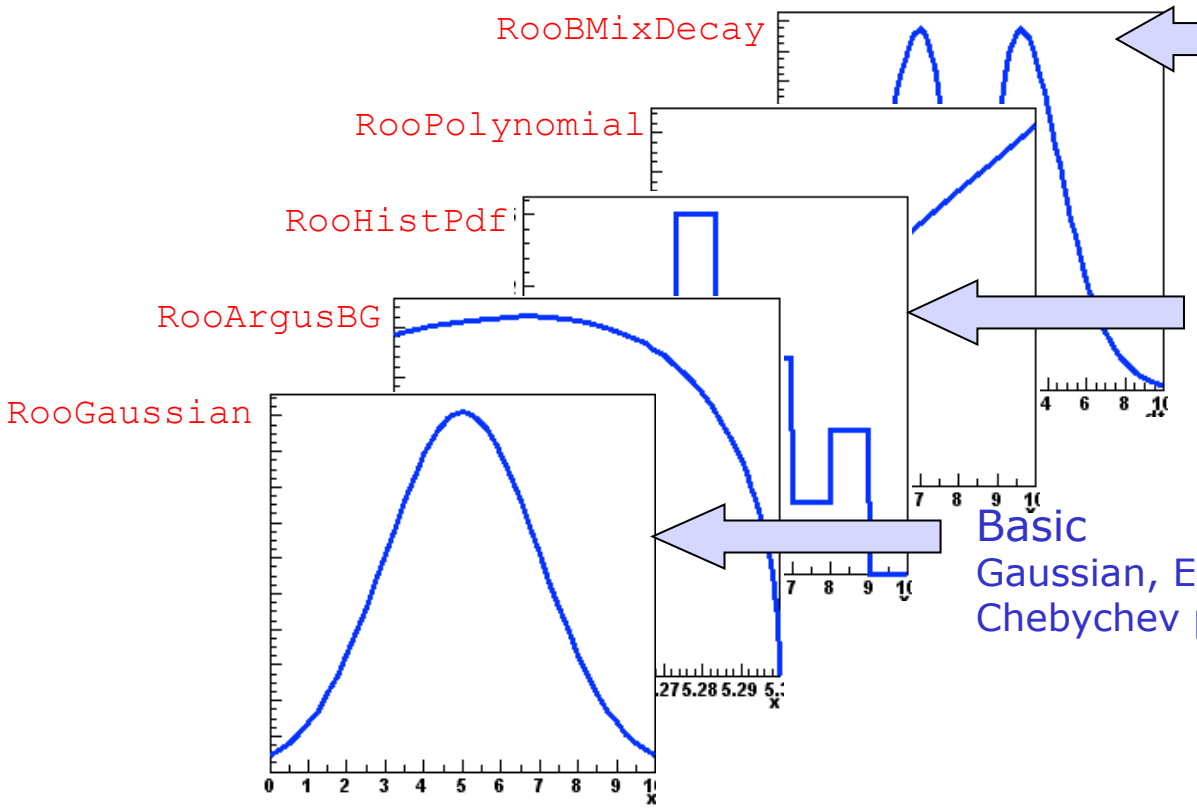


RooFit code

```
RooWorkspace w("w") ;
w.factory("RooGaussian::g(x[-10,10],m[-10,10],z[3,0.1,10])");
```

Model building – (Re)using standard components

- RooFit provides a collection of compiled standard PDF classes



Physics inspired
ARGUS,Crystal Ball,
Breit-Wigner, Voigtian,
B/D-Decay,….

Non-parametric
Histogram, KEYS

Basic
Gaussian, Exponential, Polynomial,…
Chebychev polynomial

Easy to extend the library: each p.d.f. is a separate C++ class

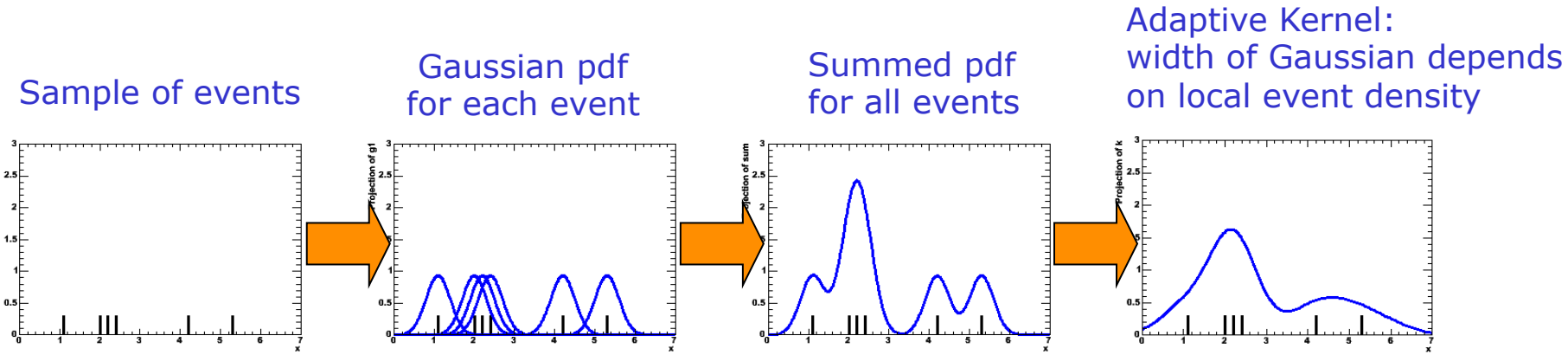
Model building – (Re)using standard components

- List of most frequently used pdfs and their factory spec

Gaussian	`Gaussian::g(x,mean,sigma)`
Breit-Wigner	`BreitWigner::bw(x,mean,gamma)`
Landau	`Landau::l(x,mean,sigma)`
Exponential	`Exponental::e(x,alpha)`
Polynomial	`Polynomial::p(x,{a0,a1,a2})`
Chebychev	`Chebychev::p(x,{a0,a1,a2})`
Kernel Estimation	`KeysPdf::k(x,dataSet)`
Poisson	`Poisson::p(x,mu)`
Voigtian (=BW⊗G)	`Voigtian::v(x,mean,gamma,sigma)`

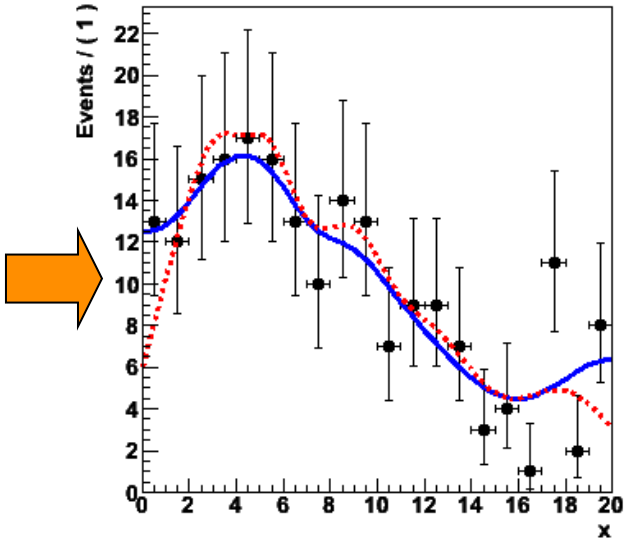
The power of pdf as building blocks – Advanced algorithms

- Example: a 'kernel estimation probability model'
  - Construct smooth pdf from unbinned data, using kernel estimation technique



- Example

```
w.import(myData,Rename("myData")) ;
w.factory("KeysPdf::k(x,myData)") ;
```



- Also available for n-D data

The power of pdf as building blocks – adaptability

- RooFit pdf classes do not require their parameter arguments to be variables, one can plug in functions as well
- Allows trivial customization, extension of probability models

class RooGaussian

$$Gauss(x \mid \mu, \sigma)$$

also class RooGaussian!

$$Gauss(x \mid \mu \cdot (1+2\alpha), \sigma)$$

Introduce a response function for a systematic uncertainty

```
// Original Gaussian
w.factory("Gaussian::g1(x[80,100],m[91,80,100],s[1])")


// Gaussian with response model in mean
w.factory("expr::m_response("m*(1+2alpha)",m,alpha[-5,5])") ;
w.factory("Gaussian::g1(x,m_response,s[1])")
```

NB: "expr" operates builds an intepreted function expression on the fly

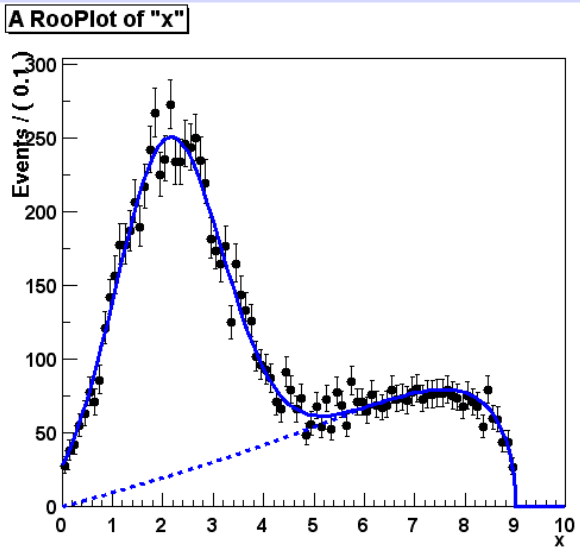
The power of building blocks – operator expressions

- Create a SUM expression to represent a sum of probability models

```
w.factory("Gaussian::gauss1(x[0,10],mean1[2],sigma[1]") ;
w.factory("Gaussian::gauss2(x,mean2[3],sigma)") ;
w.factory("ArgusBG::argus(x,k[-1],9.0)") ;

w.factory("SUM::sum(g1frac[0.5]*gauss1, g2frac[0.1]*gauss2, argus)")
```

- In composite model visualization components can be accessed by name

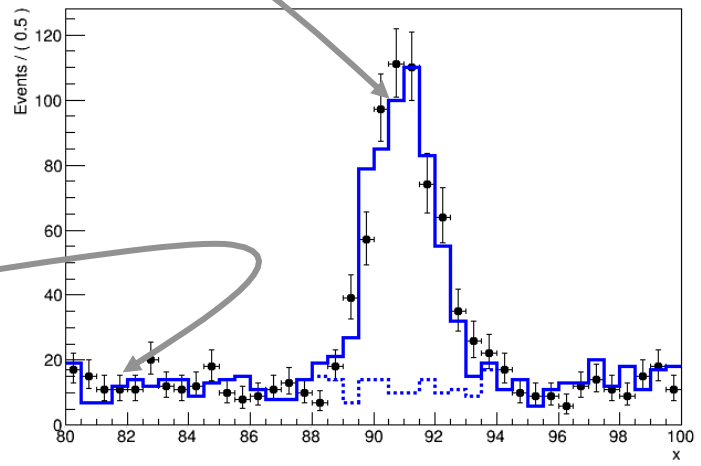
```
// Plot only argus components
w::sum.plotOn(frame,Components("argus"),
                    LineStyle(kDashed)) ;
```



The imperfect experiment

- In realistic measurements many effect that we don't control exactly influence measurements of parameter of interest
- How do you model these uncertainties in the likelihood?

$$L(\vec{N} \mid \mu) = \prod_{bins} Poisson(N_i \mid \mu \cdot \tilde{s}_i + \tilde{b}_i)$$

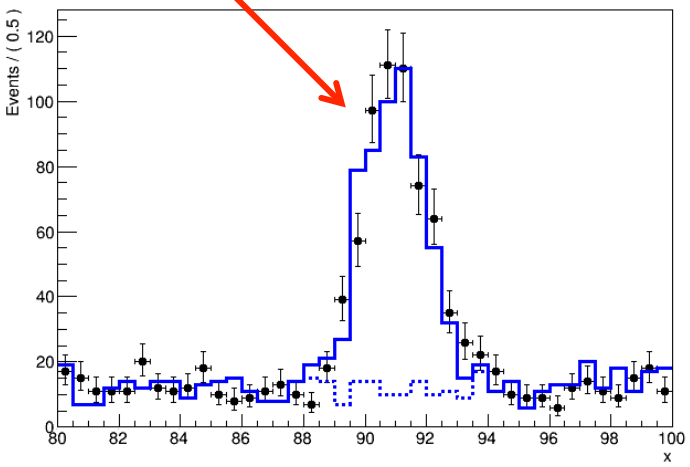


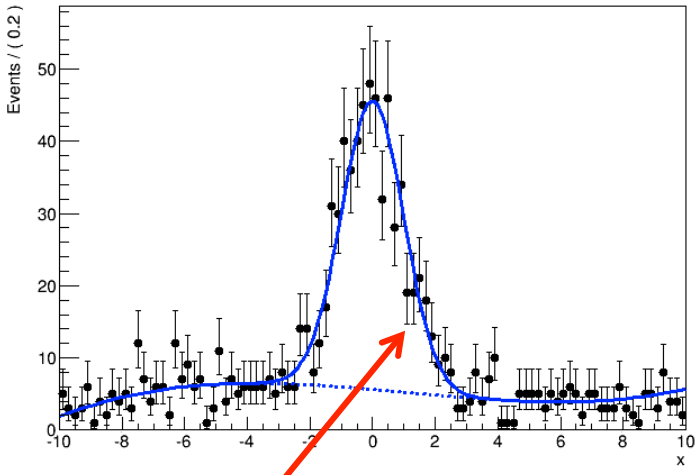
Signal and background predictions are affected by (systematic) uncertainties

Adding parameters to the model

- But parametric form of detector and theory systematic uncertainties is often, at first sight, elusive

$$L(\vec{N} \mid \mu) = \prod_{bins} Poisson(N_i \mid \mu \cdot \tilde{s}_i + \tilde{b}_i)$$



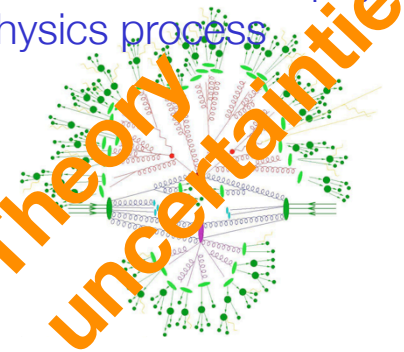


$$L(x \mid f, m, \sigma, a_0, a_1, a_2) = fG(x, m, \sigma) + (1 - f)Poly(x, a_0, a_1, a_2)$$

- Ad-hoc parameterizations (like above) do not necessarily capture all uncertain degrees of freedom, provide no meaningful insight in effect of known systematic uncertainties on the analysis.

The simulation workflow and origin of uncertainties

Simulation of 'soft physics' physics process



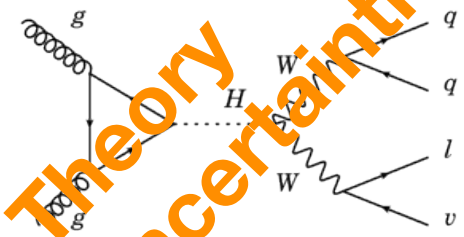
Theory uncertainties

Simulation of ATLAS detector

Detector modeling uncertainties

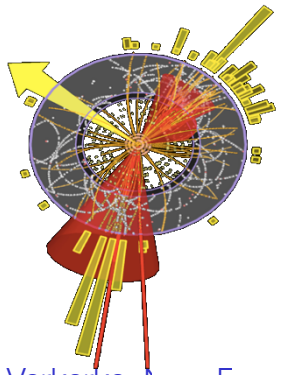
LHC data

Simulation of high-energy physics process

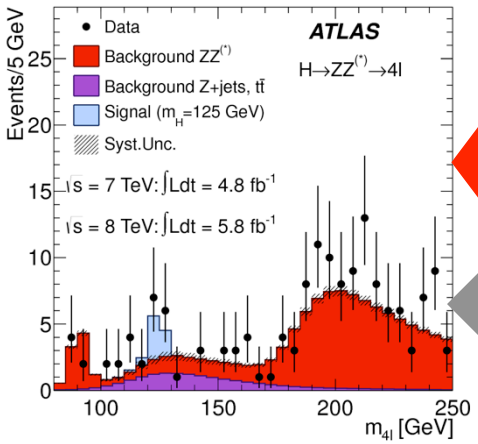


Theory uncertainties

Reconstruction of ATLAS detector



Analysis Event selection



Typical systematic uncertainties in HEP

- Detector-simulation related
  - "The Jet Energy scale uncertainty is 5%"
  - "The b-tagging efficiency uncertainty is 20% for jets with p_T<40"
- Physics/Theory related
  - The top cross-section uncertainty is 8%
  - "Vary the factorization scale by a factor 0.5 and 2.0 and consider the difference the systematic uncertainty"
  - "Evaluate the effect of using Herwig and Pythia and consider the difference the systematic uncertainty"
- MC simulation statistical uncertainty
  - Effect of (bin-by-bin) statistical uncertainties in MC samples

What can you do with *systematic* uncertainties

- As most of the typical systematic prescriptions have no immediately apparent parametric formulation in your likelihood, common approach is 'vary setting, rerun analysis, observe the difference'
- This common 'naïve' approach to assess effect of systematic uncertainties amounts to simple error propagation
- Error propagation procedure in a nutshell
  - Make nominal measurement (using your favorite statistical inference procedure)
  - Change setting in detector simulation or theory (e.g. shift Jet Calibration scale by '1 sigma' up and down) Redo measurement procedure for each shift
  - Consider propagated effect of shifted setting the systematic uncertainty

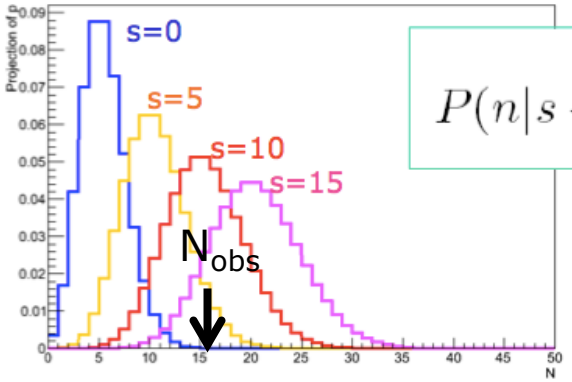
$$\mu = \underbrace{\mu_{nom} \pm \sigma_{stat}}_{\text{From statistical analysis}} \pm \underbrace{(\mu_{syst}^{up} - \mu_{syst}^{down}) / 2}_{\text{Systematic uncertainty from error propagation}} \pm \ldots$$

Pros and cons of the 'naïve' approach

- **Pros**
  - It's easy to do
  - It results in a seemingly easy-to-interpret table of systematics
- **Cons**
  - Uncorrelated source of systematic uncertainty can have correlated effect on measurement → Completely ignored
  - Magnitude of stated systematic uncertainty may be incompatible with measurement result → Completely ignored
  - You lost the connection with fundamental statistical techniques (i.e. evaluation of systematic uncertainties is completely detached from statistical procedure used to estimate physics quantity of interest) → No prescription to make confidence intervals, Bayesian posteriors etc in this way
  - No calibrated probabilistic statements possible (95% C.L.)
- 'Profiling' → Incorporate a description of systematic uncertainties in the likelihood function that is used in statistical procedures

Everything starts with the likelihood

- **All** fundamental statistical procedures are based on the likelihood function as 'description of the measurement'



$$P(n|s+b) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$

NB: b is a constant in this example

Definition: the Likelihood is P(observed data|theory)

e.g. L(15|s=0)

e.g. L(15|s=10)

Frequentist statistics → **Confidence interval on s**

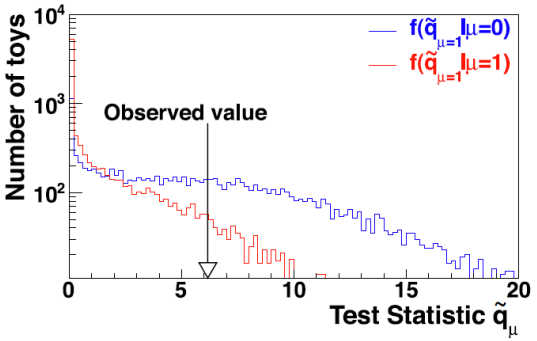
Bayesian statistics → **Posterior on s**

Maximum Likelihood → **s = x ± y**

Everything starts with the likelihood

Frequentist statistics

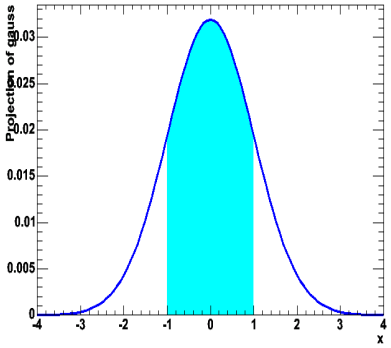
$$\lambda_\mu(\vec{N}_{obs}) = \frac{L(\vec{N} \mid \mu)}{L(\vec{N} \mid \hat{\mu})}$$



Confidence interval or p-value

Bayesian statistics

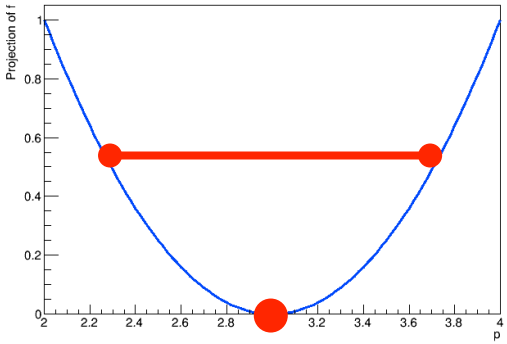
$$P(\mu) \propto L(x \mid \mu) \cdot \pi(\mu)$$



Posterior on s or Bayes factor

Maximum Likelihood

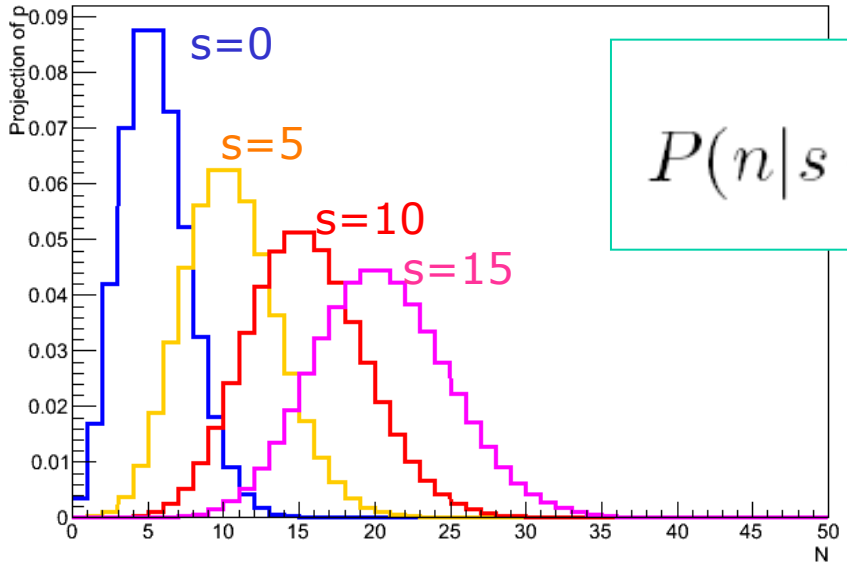
$$\left.\frac{d \ln L(\vec{p})}{d\vec{p}}\right|_{p_i = \hat{p}_i} = 0$$



s = x ± y

Introducing uncertainties – a non-systematic example

- The original model (with fixed *b*)



$$P(n|s+b) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$

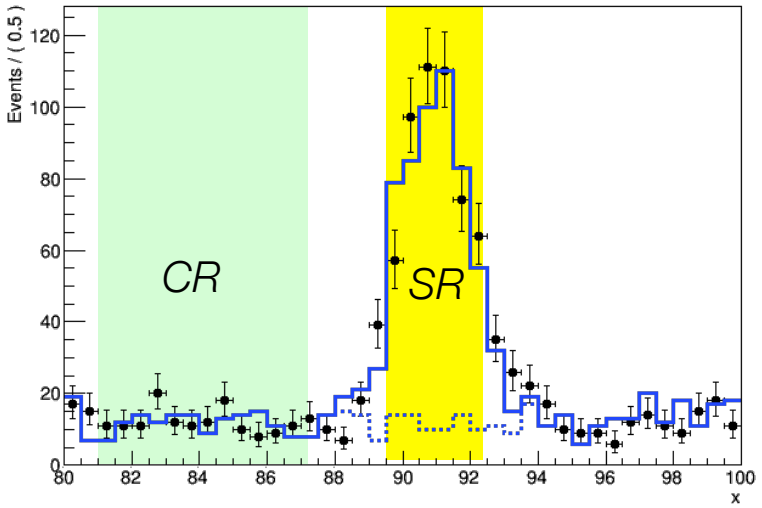
- Now consider *b* to be uncertain

  L(N|s) → L(N|s,b)

- The experimental data contains insufficient to constrain both *s* and *b* → Need to add an additional measurement to constrain *b*

The sideband measurement

- Suppose your data in reality looks like this ➔



Can estimate level of background in the 'signal region' from event count in a 'control region' elsewhere in phase space

$$L_{SR}(s,b) = Poisson(N_{SR} \mid s+b)$$

NB: Define parameter 'b' to represents the amount of bkg is the SR.

$$L_{CR}(b) = Poisson(N_{CR} \mid \tilde{\tau} \cdot b)$$

Scale factor τ accounts for difference in size between SR and CR

"Background uncertainty constrained from the data"

- Full likelihood of the measurement ('simultaneous fit')

$$L_{full}(s,b) = Poisson(N_{SR} \mid s+b) \cdot Poisson(N_{CR} \mid \tilde{\tau} \cdot b)$$

Generalizing the concept of the sideband measurement

- Background uncertainty from sideband clearly clearly not a 'systematic uncertainty'

$$L_{full}(s,b) = Poisson(N_{SR} \mid s+b) \cdot Poisson(N_{CR} \mid \tilde{\tau} \cdot b)$$

- Now consider scenario where *b* is not measured from a sideband, but is taken from MC simulation **with an 8% cross-section 'systematic' uncertainty**

'Measured background rate by MC simulation'

$$L_{full}(s,b) = Poisson(N_{SR} \mid s+b) \cdot Gauss(\tilde{b} \mid b, 0.08)$$

'**Subsidiary measurement**' of background rate

  - *We can model this in the same way, because the cross-section uncertainty is also (ultimately) the result of a measurement*

Generalize: 'sideband' → 'subsidiary measurement'

What is a systematic uncertainty?

- Concept & definitions of ‘systematic uncertainties’ originates from physics, not from fundamental statistical methodology.
  - E.g. Glen Cowans (excellent) 198pp book “statistical data analysis” does not discuss systematic uncertainties at all

- A common definition is
  - “Systematic uncertainties are all uncertainties that are not directly due to the statistics of the data”

- But the notion of ‘the data’ is a key source of ambiguity:
  - does it include control measurements?
  - does it include measurements that were used to perform basic (energy scale) calibrations?

Typical systematic uncertainties in HEP

- **Detector-simulation related**
  - "The Jet Energy scale uncertainty is 5%"
  - "The b-tagging efficiency uncertainty is 20% for jets with p_T<40"

Subsidiary measurement is an actual measurement → conceptually similar to a 'sideband' fit

- **Physics/Theory related**
  - The top cross-section uncertainty is 8%
  - "Vary the factorization scale by a factor 0.5 and 2.0 and consider the difference the systematic uncertainty"
  - "Evaluate the effect of using Herwig and Pythia and consider the difference the systematic uncertainty"

Subsidiary measurement unclear, but origin of prescription may well be another measurement (if yes, like sideband, if no, what is source of info?)

- **MC simulation statistical uncertainty**
  - Effect of (bin-by-bin) statistical uncertainties in MC samples

Subsidiary measurement is a Poisson counting experiment (but now in MC events), otherwise conceptually identical to a 'sideband fit'

Typical systematic uncertainties in HEP

- Detector-simulation related
  - "The Jet Energy scale uncertainty is 5%"
  - "The b-tagging efficiency uncertainty is 20%
- P
- MC simulation statistical uncertainty
  - Effect of (bin-by-bin) statistical uncertainties in MC samples

Subsidiary measurement is an actual measurement → conceptually to

Almost all systematic uncertainties are similar in nature to 'sidebands' measurements of some form or shape

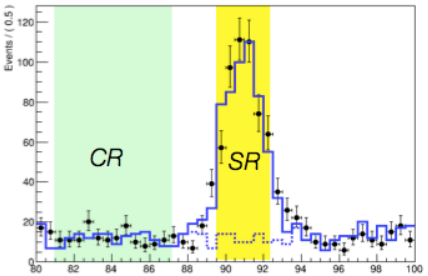
→ Can always model systematics like sidebands in the Likelihood

And even when the are not the (in)direct result of some measurement (certainty theory uncertainties) we can still model them in that form

Subsidiary measurement is a Poisson counting experiment (but now in MC events), otherwise conceptually identical to a 'sideband fit'

Modeling a detector calibration uncertainty

$$L_{full}(s,b) = Poisson(N_{SR} \mid s+b) \cdot Gauss(\tilde{b} \mid b, 0.08)$$



- **Now consider a detector uncertainty**, e.g. jet energy scale calibration, which can affect the analysis acceptance in a non-trivial way (unlike the cross-section example)

Signal rate (our parameter of interest)

Nominal calibration

Assumed calibration

$$L(N,\tilde{\alpha} \mid s,\alpha) = Poisson(N \mid s + \tilde{b}(\alpha/\tilde{\alpha}) \cdot 2)) \cdot Gauss(\tilde{\alpha} \mid \alpha, \sigma_{\alpha})$$

Observed event count

Nominal background expectation from MC (a constant), obtained with a=a˜

Response function for JES uncertainty (a 1% JES change results in a 2% acceptance change)

"Subsidiary measurement" Encodes 'external knowledge' on JES calibration

Uncertainty on nominal calibration **(here 5%)**

Modeling a detector calibration uncertainty

- Simplify expression by renormalizing "subsidiary measurement"

$$Gauss(\tilde{\alpha} \mid \alpha, \sigma_\alpha)$$

Signal rate (our parameter of interest)

$$L(N \mid s, \alpha) = Poisson(N \mid s + \tilde{b}(1 + 0.1\alpha)) \cdot Gauss(0 \mid \alpha, 1)$$

Observed event count

Nominal background expectation from MC (a constant)

Response function for normalized JES parameter
[a unit change in α – a 5% JES change – still results in a 10% acceptance change]

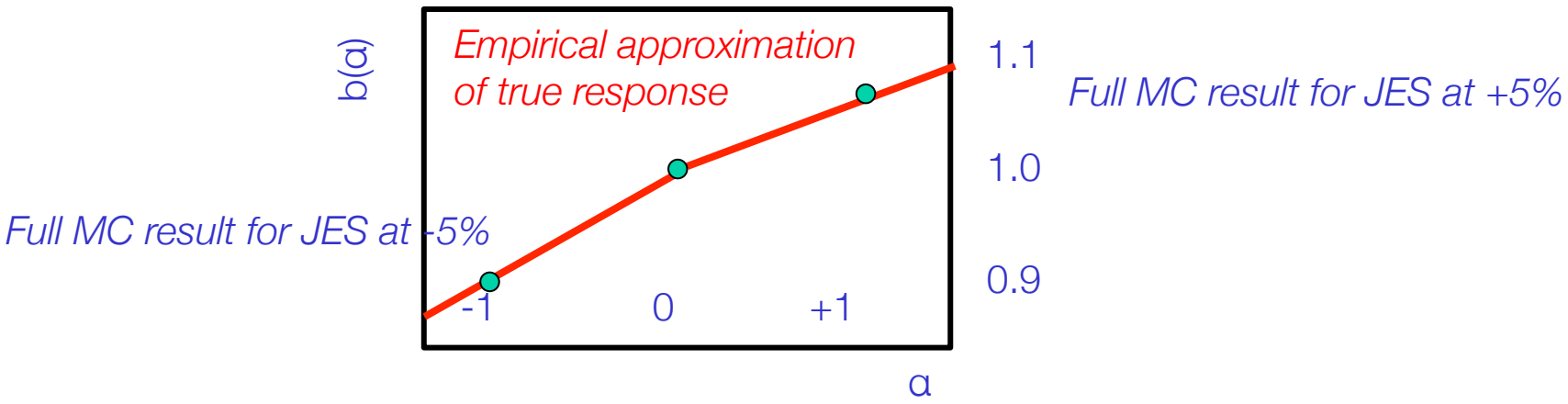
"Normalized subsidiary measurement"

The scale of parameter α is now chosen such that values ±1 corresponds to the nominal uncertainty (in this example 5%)

The response function as empirical model of full simulation

$$L(N,0 \mid s,\alpha) = Poisson(N \mid s + b(\alpha)) \cdot Gauss(0 \mid \alpha,1)$$

- Note that the response function is generally not linear, but can in principle *always be determined by your full simulation chain*
  - But you cannot run your full simulation chain for any arbitrary 'systematic uncertainty variation' → Too much time consuming
  - Typically, run full MC chain for nominal and ±1σ variation of systematic uncertainty, and approximate response for other values of NP with interpolation
  - For example run at nominal JES and with JES shifted up and down by ±5%



What is a systematic uncertainty?

- It is an uncertainty in the Likelihood of your physics measurement that is characterized deterministically, up to a set of parameters, of which the true value is unknown.
- A fully specified systematic uncertainty defines
  - 1: A set of one or more parameters
    of which the true value is unknown,
  - 2: A response model that describes the effect of those parameters on the measurement
    (sampled from full simulation, and interpolation)
  - 3: A subsidiary measurement of the parameters
    that constrains the values the parameters can take
    (implies a specific distribution: Gaussian *(default, CLT)*,
    Poisson *(low-stats counting)*, or otherwise)

Names and conventions – 'profiling' & 'constraints'

- The full likelihood function of the form

$$L(N,0 \mid s,\alpha) = Poisson(N \mid s + b(\alpha)) \cdot Gauss(0 \mid \alpha,1)$$

  is usually referred to by physicists as a 'profile likelihood', and systematics are said to be 'profiled' when incorporated this way
  - Note: statisticians use the word profiling for something else
- Physicists often refer to the subsidiary measurement as a 'constraint term'
  - This is correct in the sense that it constrains the parameter α, but this labeling commonly lead to mistaken statements (e.g. that it is a pdf for α)
  - But it is *not* a pdf in the NP

~~$Gauss(\alpha \mid 0,1)$~~ $Gauss(0 \mid \alpha,1)$

Names and conventions

- The ‘subsidiary measurement’ as simplified form of the ‘full calibration measurement’ also illustrates another important point
  - The full likelihood is simply a *joint likelihood of a physics measurement and a calibration measurement* where both terms are treated on equal footing in the statistical procedure
  - In a perfect world, not bound by technical modelling constraints you would use this likelihood

$$L(N,\vec{y}\,|\,s,\alpha) = Poisson(N\,|\,s+b(1+0.1\alpha))\cdot L_{JES}(\vec{y}\,|\,\alpha,\vec{\theta})$$

    where L_{JES} is the full calibration measurement as performed by the Jet calibration group, based on a dataset y, and which may have other parameters θ specific to the calibration measurement.

- Since we are bound by technical constrains, we substitute L_{JES} with simplified (Gaussian) form, but the statistical treatment and interpretation remains the same

MC statistical uncertainties as systematic uncertainty

- Another example of modeling a systematic uncertainty: MC statistical uncertainty
- Follow same procedure again as before:
  - Define response function (this is trivial for MC statistics: it is the luminosity ratio of the MC sample and the data sample)
  - Define distribution for the 'subsidiary measurement' – This is a Poisson distribution – since MC simulation is also a Poisson process
  - Construct full likelihood ('profile likelihood')

$$L(N, N_{MC} \mid s, b) = Poisson(N \mid s + b) \cdot Poisson(N_{MC} \mid \tau \cdot b)$$

Constant factor τ = *L*(MC)/*L*(data)

- Note uncanny similarity to full likelihood of a sideband measurement!

$$L(N, N_{ctl} \mid s, b) = Poisson(N \mid s + b) \cdot Poisson(N_{ctl} \mid \tau \cdot b)$$

Modeling multiple systematic uncertainties

- Introduction of multiple systematic uncertainties presents no special issues
- Example JES uncertainty plus generator ISR uncertainty

$$L(N,0 \mid s,\alpha_{JES},\alpha_{ISR}) = P(N \mid s + b(1 + 0.1\alpha_{JES} + 0.05\alpha_{ISR})) \cdot G(0 \mid \alpha_{JES},1) \cdot G(0 \mid \alpha_{ISR},1)$$

Joint response function for both systematics

One subsidiary measurement for each source of uncertainty

- A brief note on correlations
  - Word "correlations" often used sloppily – proper way is to think of correlations of parameter estimators. Likelihood defines parameters α_{JES}, α_{ISR}. The (ML) estimates of these are denoted $\hat{\alpha}_{JES}, \hat{\alpha}_{ISR}$
  - The ML estimators of $\hat{\alpha}_{JES}, \hat{\alpha}_{ISR}$ using the Likelihood of the subsidiary measurements are uncorrelated (since the product factorize in this example)
  - The ML estimators of $\hat{\alpha}_{JES}, \hat{\alpha}_{ISR}$ using the full Likelihood may be correlated. This is due to physics modeling effects encoded in the joint response function

Modeling systematic uncertainties in multiple channels

- Systematic effects that affect multiple measurements should be modeled coherently.
  - Example – Likelihood of two Poisson counting measurements

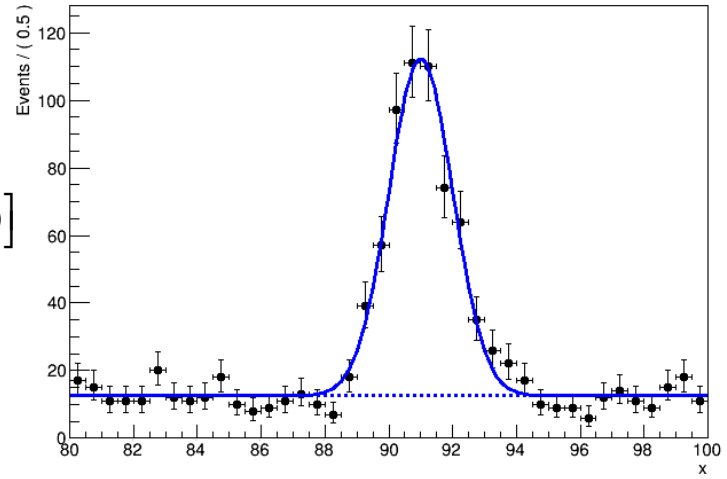
$$L(N_A, N_B \mid s, \alpha_{JES}) = P(N_A \mid s \cdot f_A + b_A \underbrace{(1 + 0.1\alpha_{JES})}_{\text{JES response function for channel A}}) \cdot P(N_B \mid s \cdot f_B + b_B \underbrace{(1 - 0.3\alpha_{JES})}_{\text{JES response function for channel B}}) \cdot \underbrace{G(0 \mid \alpha_{JES}, 1)}_{\text{JES subsidiary measurement}} \cdot$$

  - Effect of changing JES parameter α_{JES} coherently affects both measurement.
  - Magnitude and sign effect does not need to be same, this is dictated by the physics of the measurement

Introducing response functions for shape uncertainties

- Modeling of systematic uncertainties in Likelihoods describing distributions follows the same procedure as for counting models
  - Example: Likelihood modeling distribution in a di-lepton invariant mass. POI is the signal strength μ

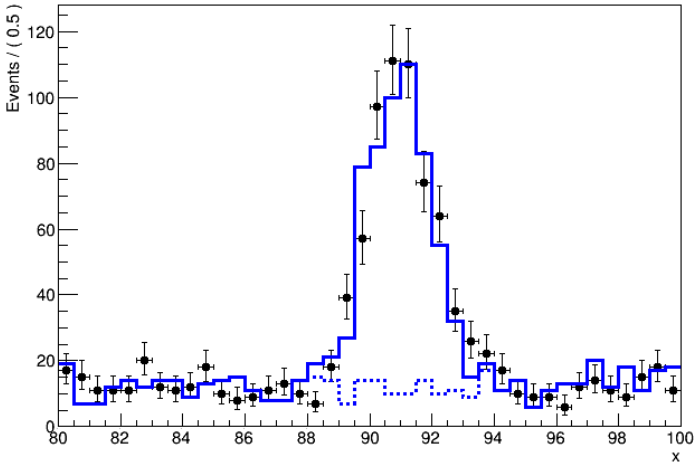
$$L(\vec{m}_{ll} \mid \mu) = \prod_i \left[\mu \cdot \text{Gauss}(m_{ll}^{(i)}, 91, 1) + (1-\mu) \cdot \text{Uniform}(m_{ll}^{(i)}) \right]$$



- Consider a lepton energy scale systematic uncertainty that affects this measurement
  - The LES has been measured with a 1% precision
  - The effect of LES on m_{ll} has been determined to a 2% shift for 1% LES change

$$L(\vec{m}_{ll} \mid \mu, \alpha_{LES}) = \prod_i \left[\mu \cdot \text{Gauss}(m_{ll}^{(i)}, 91 \cdot \underbrace{(1+2\alpha_{LES}}_{\text{Response function}}, 1) + (1-\mu) \cdot \text{Uniform}(m_{ll}^{(i)}) \right] \cdot \underbrace{Gauss(0 \mid \alpha_{LES}, 1)}_{\text{Subsidiary measurement}}$$

Response modeling for distributions



- For a change in the **rate**, response modeling of histogram-shaped distribution is straightforward: **simply scale entire distribution**

$$L(\vec{N} \mid \mu) = \prod_i Poisson(N_i \mid \mu \tilde{s}_i + \tilde{b}_i)$$

$$L(\vec{N} \mid \mu, \alpha) = \prod_i Poisson(N_i \mid \mu \tilde{s}_i \cdot (1 + 3.75\alpha) + \tilde{b}_i) \cdot Gauss(0 \mid \alpha, 1)$$

Response function for signal rate

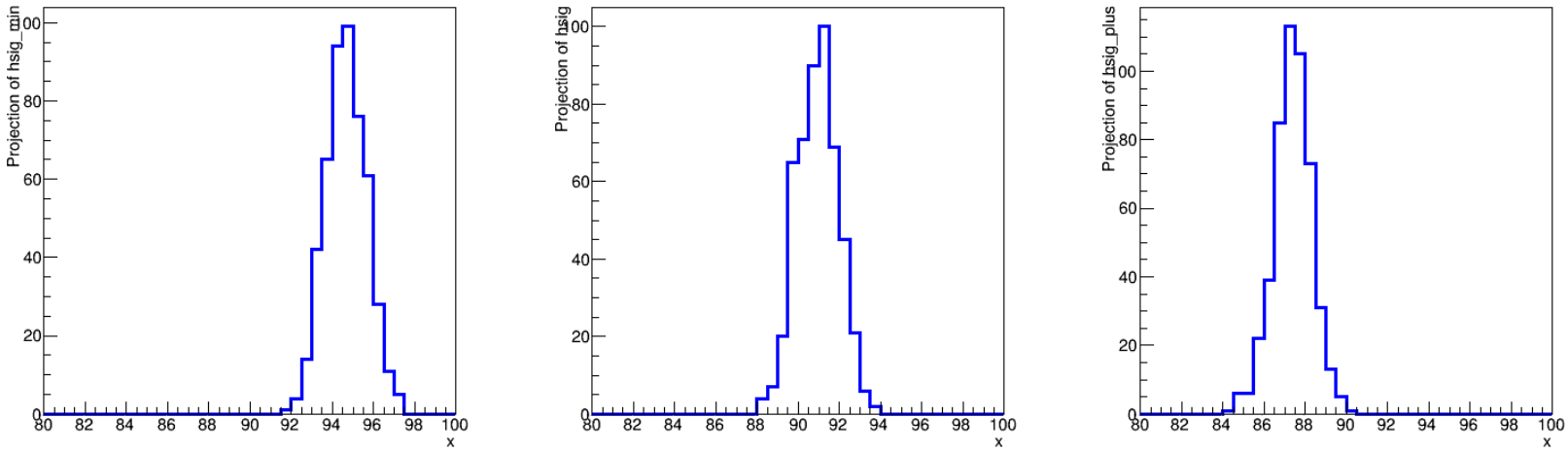
Subsidiary measurement

- But what about a systematic uncertainty that shifts the mean, or affects the distribution in another way?

Modeling of shape systematics in the likelihood

- Effect of *any* systematic uncertainty that affects the shape of a distribution can in principle be obtained from MC simulation chain
  - Obtain histogram templates for distributions at '+1σ' and '-1σ' settings of systematic effect

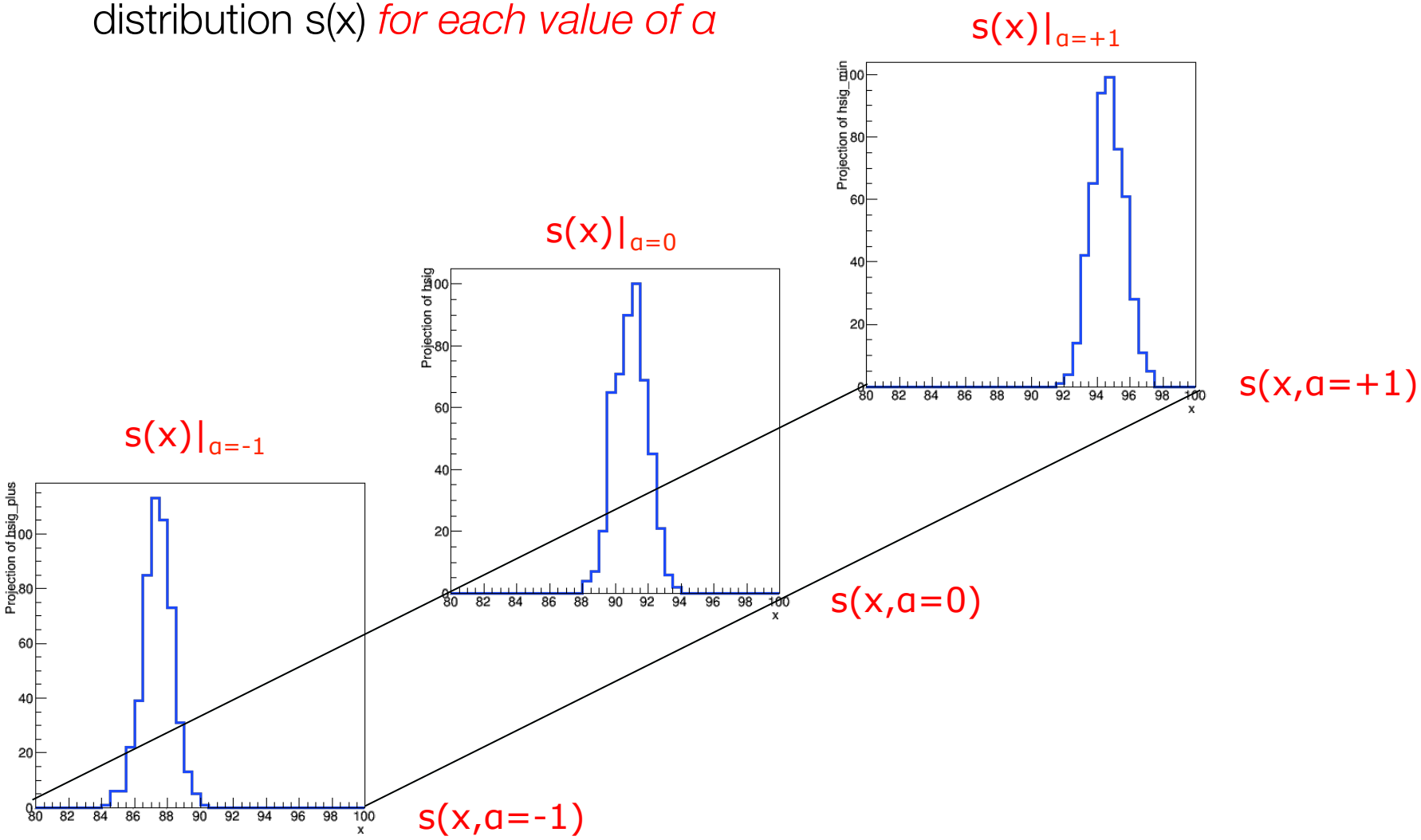
'-1σ' 'nominal' '+1σ'



- Challenge: **construct an empirical response function based on the interpolation of the shapes of these three templates**.

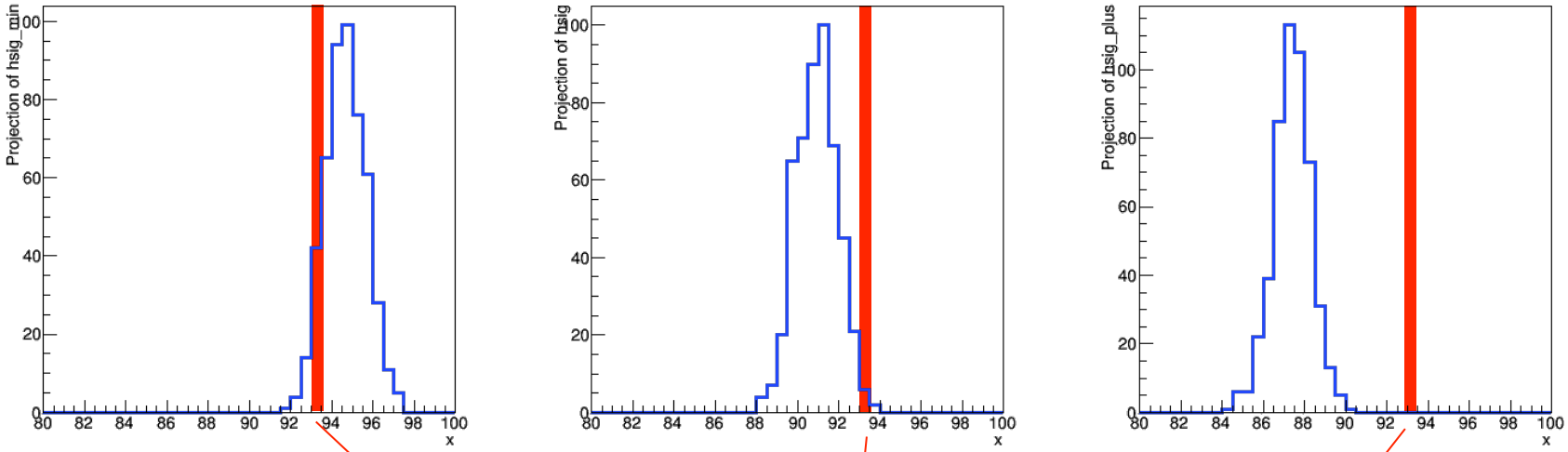
Need to interpolate between template models

- Need to define 'morphing' algorithm to define distribution s(x) *for each value of α*

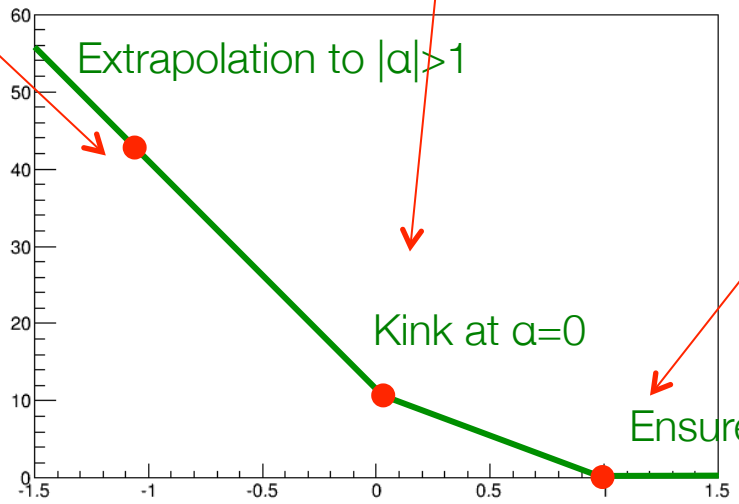


Piecewise linear interpolation

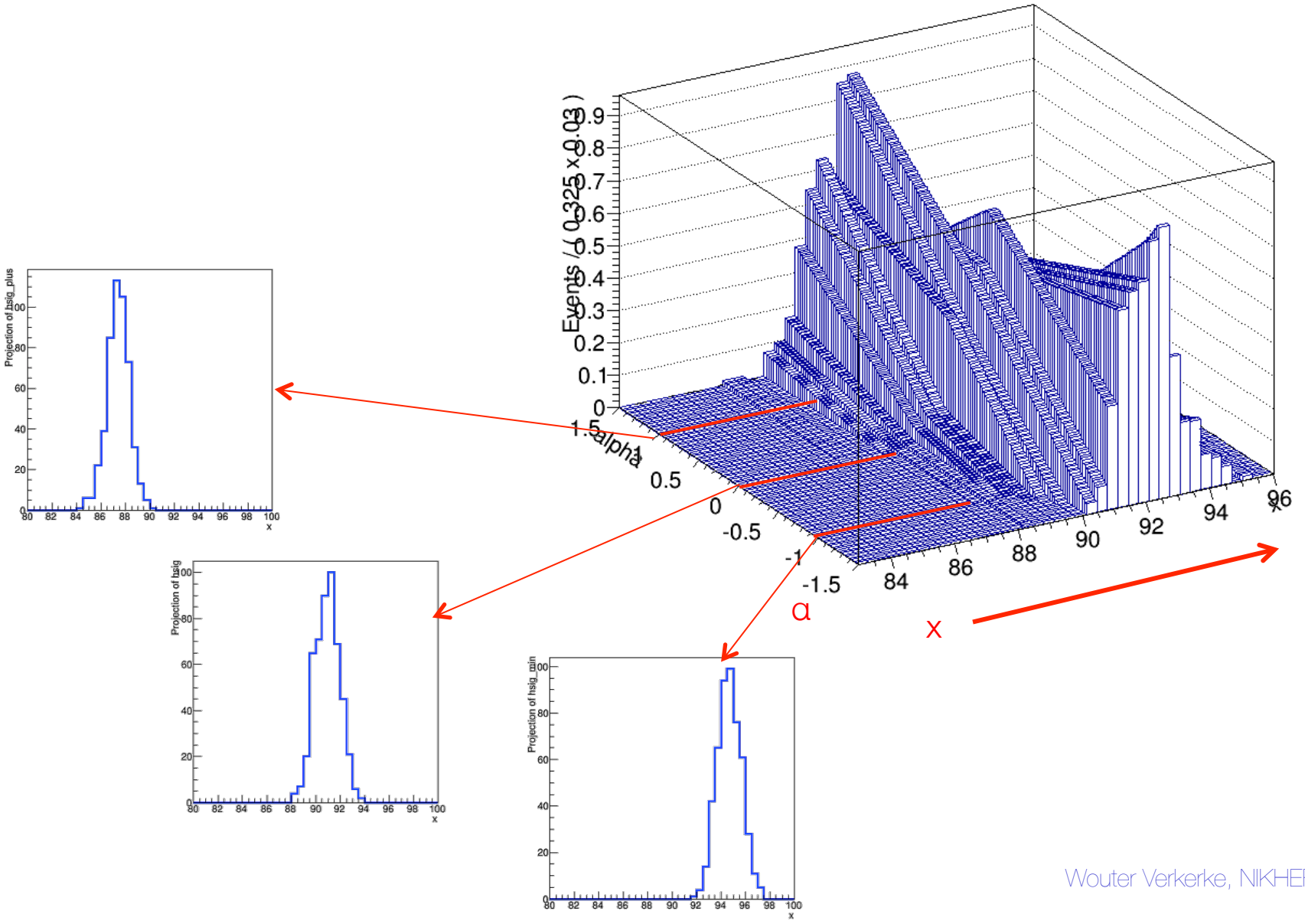
- Simplest solution is piece-wise linear interpolation for each bin



Piecewise linear interpolation response model for a one bin



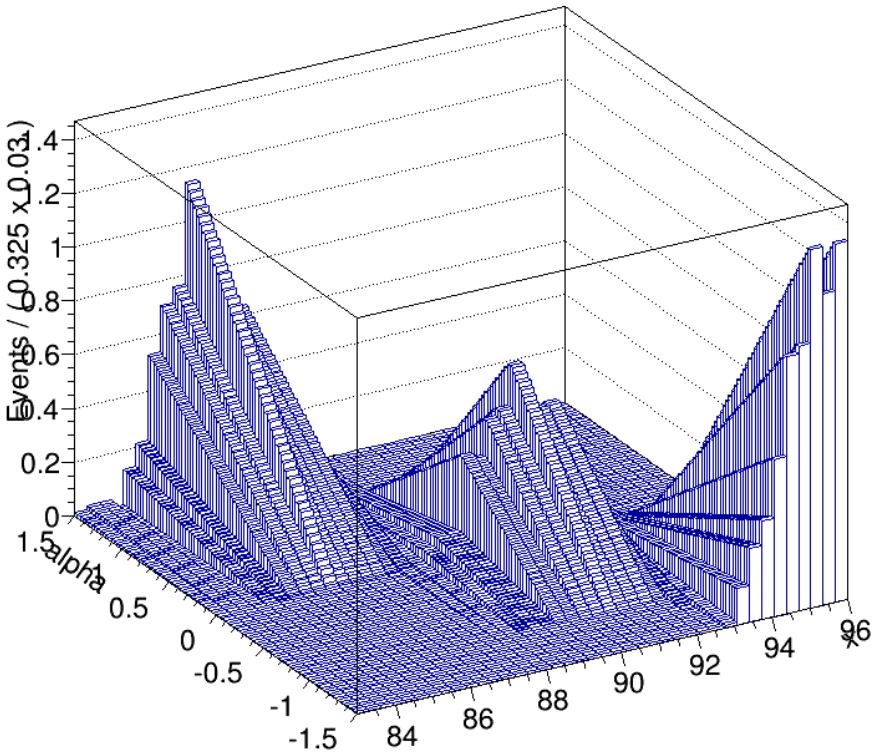
Visualization of bin-by-bin linear interpolation of distribution

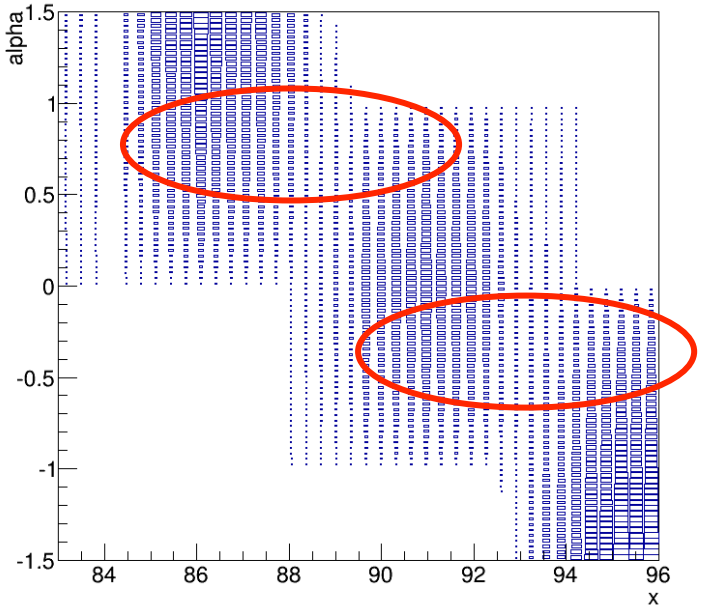


Limitations of piece-wise linear interpolation

- Bin-by-bin interpolation looks spectacularly easy and simple, but be aware of its limitations
  - Same example, but with larger 'mean shift' between templates

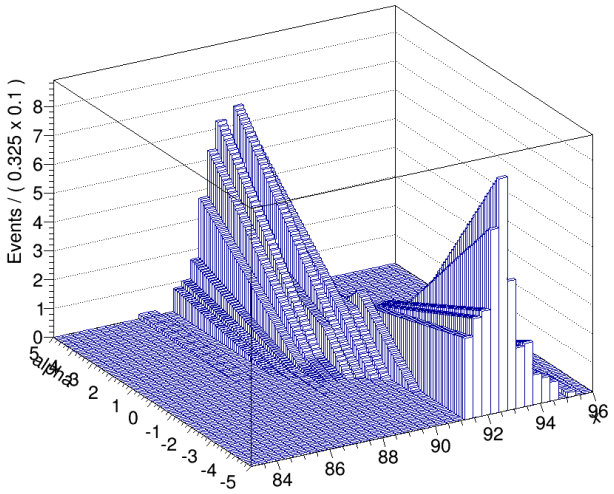
Note double peak structure around |α|=0.5

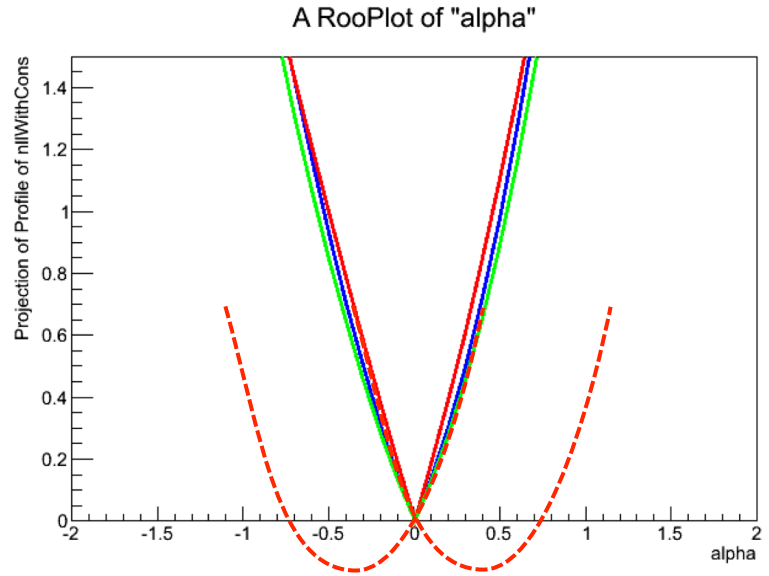




Non-linear interpolation options

- Piece-wise linear interpolation leads to kink in response functions that may result in pathological likelihood functions

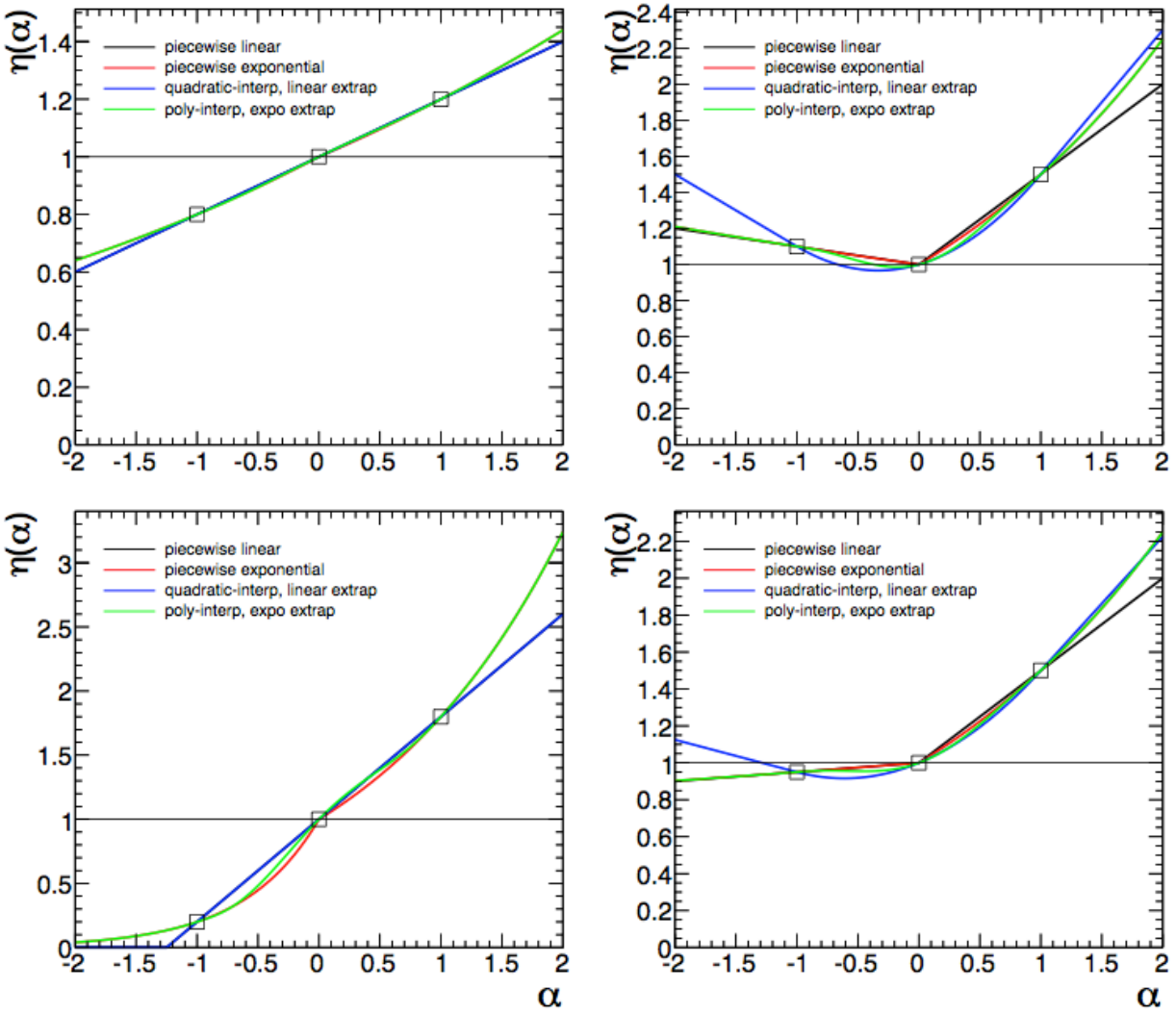




- A variety of other interpolation options exist that improve this
  - Parabolic interpolation/linear extrapolation (but causes shift of minimum)
  - Polynomial interpolation [orders 1,2,4,6]/linear extrapolation (order 1 term allows for asymmetric modeling of templates)

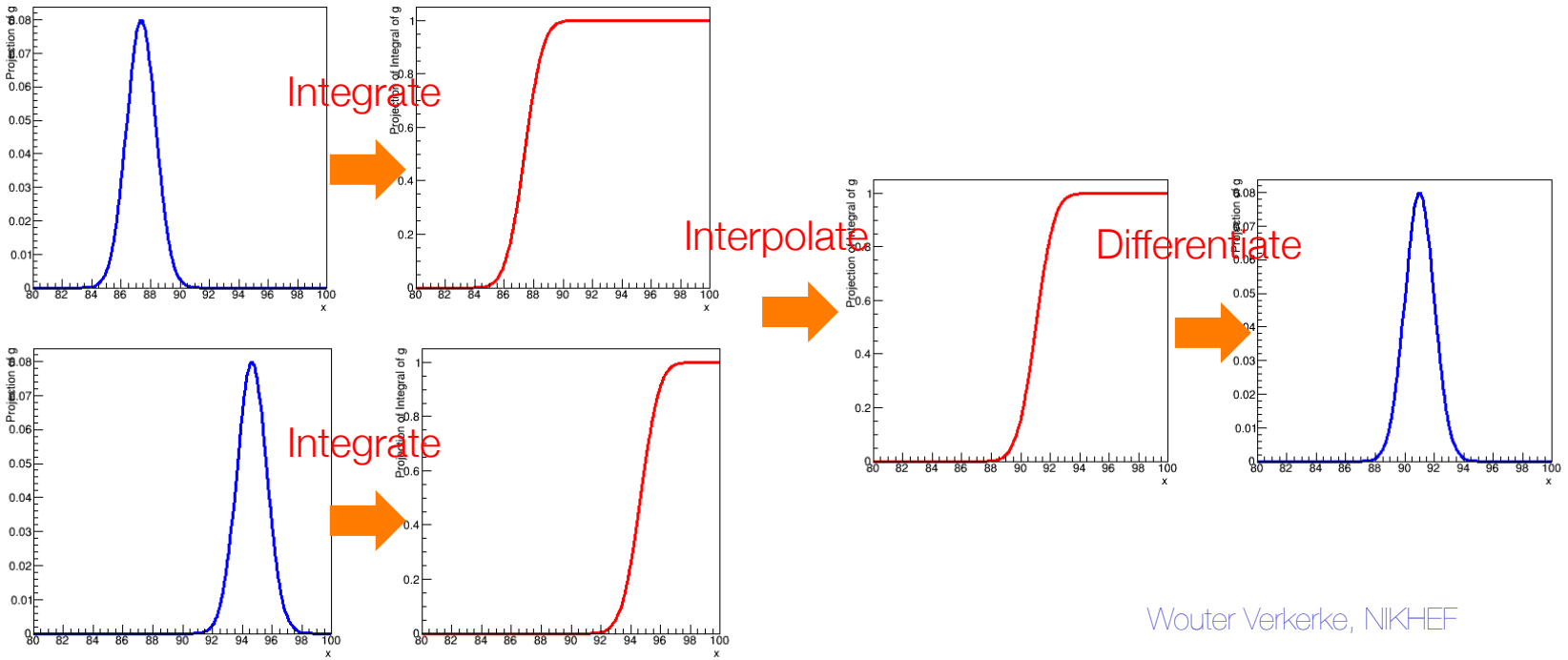
Non-linear interpolation options

- Comparison of common interpolation options



Other morphing strategies – 'horizontal morphing'

- Other template morphing strategies exist that are less prone to unintended side effects
- A 'horizontal morphing' strategy was invented by Alex read.
  - Interpolates the cumulative distribution function instead of the distribution
  - Especially suitable for shifting distributions
  - Here shown on a continuous distribution, but also works on histograms
  - Drawback: computationally expensive, algorithm only worked out for 1 NP



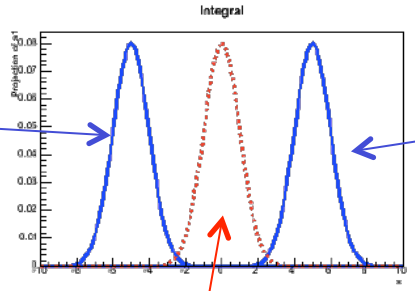
Yet another morphing strategy – 'Moment morphing'

M. Baak & S. Gadatsch

- Given two template model $f_-(x)$ and $f_+(x)$ the strategy of moment morphing considers first two moment of template models (mean and variance)

$$\mu_- = \int x \cdot f_-(x)dx \qquad \mu_+ = \int x \cdot f_+(x)dx$$

$$V_- = \int (x-\mu_-)^2 \cdot f_-(x)dx \qquad V_+ = \int (x-\mu_+)^2 \cdot f_+(x)dx$$



- The goal of moment morphing is to construct an interpolated function that has linearly interpolated moments

$$\mu(\alpha) = \alpha\mu_- + (1-\alpha)\mu_+$$
$$V(\alpha) = \alpha V_- + (1-\alpha)V_+ \quad [1]$$

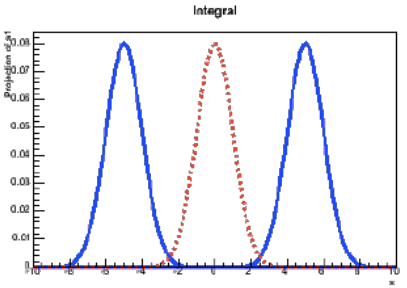
- It constructs this morphed function as combination of linearly transformed input models

$$f(x,\alpha) \rightarrow \alpha f_-(ax+b) + (1-\alpha) f_+(cx-d)$$

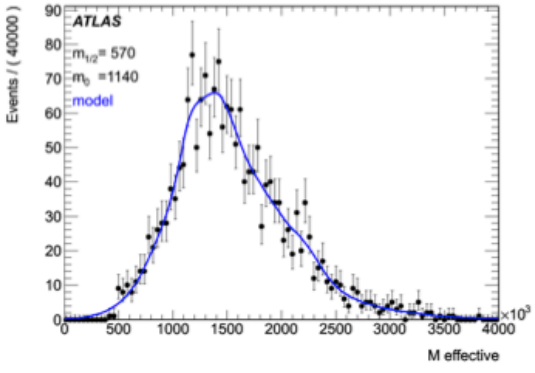
  - Where constants a,b,c,d are chosen such so that f(x,α) satisfies conditions [1]

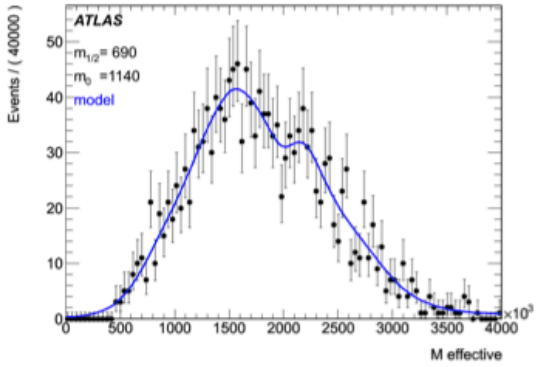
Yet another morphing strategy – 'Moment morphing'

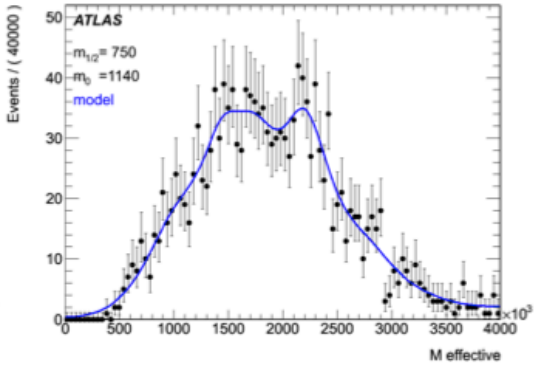
- For a Gaussian probability model with linearly changing mean and width, moment morphing of two Gaussian templates is the exact solution



- But also works well on 'difficult' distributions





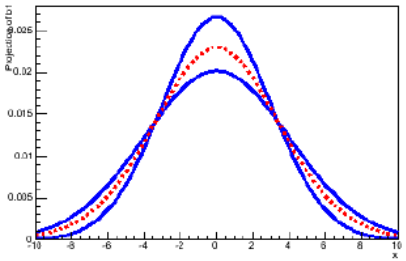
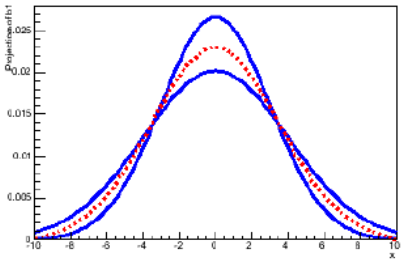
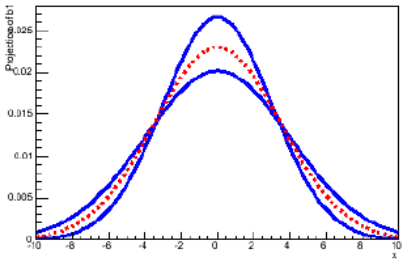
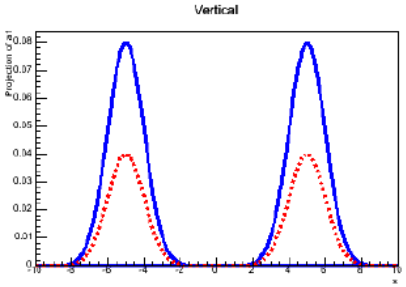
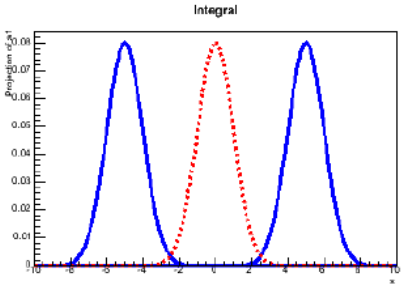
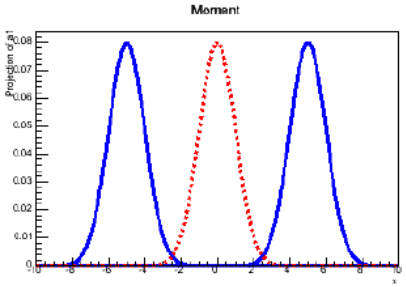
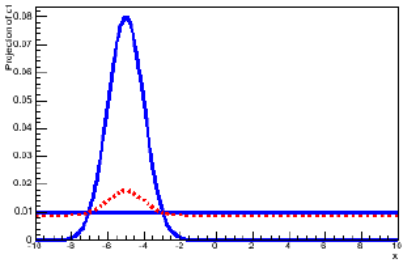
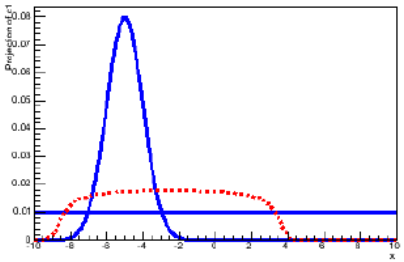
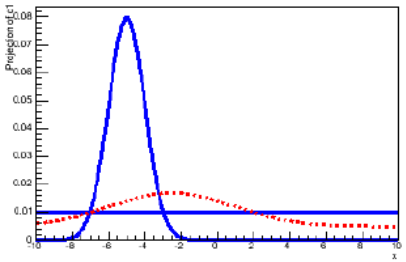


- Good computational performance
  - Calculation of moments of templates is expensive, but just needs to be done once, otherwise very fast (just linear algebra)

$$f(x,\alpha) \rightarrow \alpha f_{-}(ax+b) + (1-\alpha) f_{+}(cx-d)$$

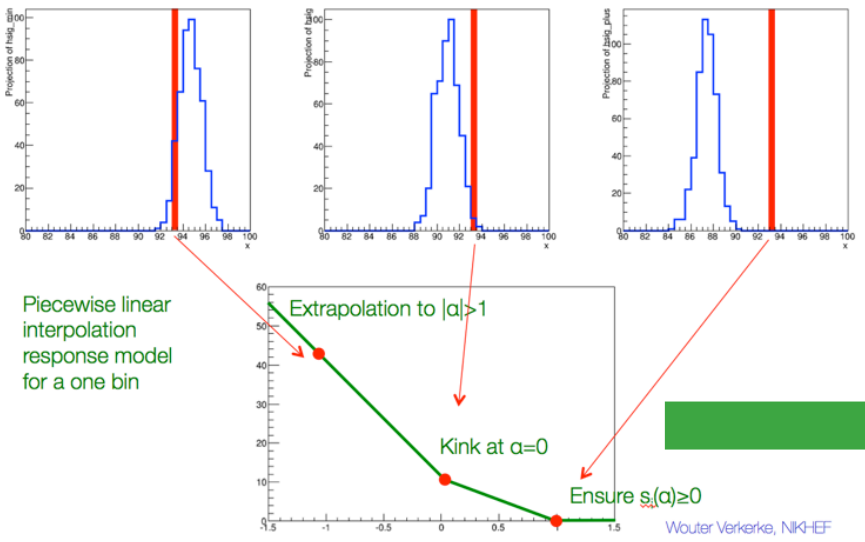
- Multi-dimensional interpolation strategies exist

There are other morphing algorithms to choose from

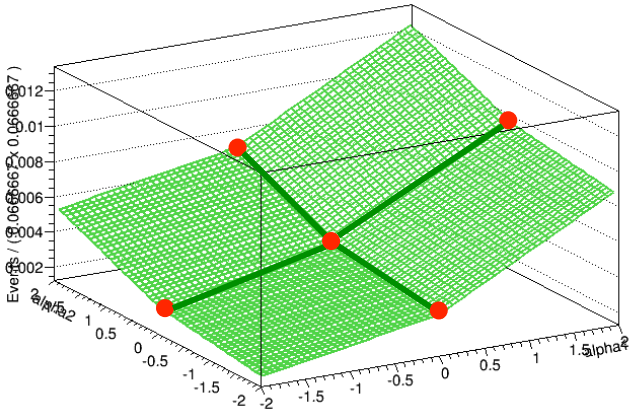
	Vertical Morphing	Horizontal Morphing	Moment Morphing
Gaussian varying width			
Gaussian varying mean			
Gaussian to Uniform (this is conceptually ambigous!)			
n-dimensional morphing?	✔	✗	✔

Piece-wise interpolation for >1 nuisance parameter

- Concept of piece-wise linear interpolation can be trivially extended to apply to morphing of >1 nuisance parameter.
  - Difficult to visualize effect on full distribution, but easy to understand concept at the individual bin level

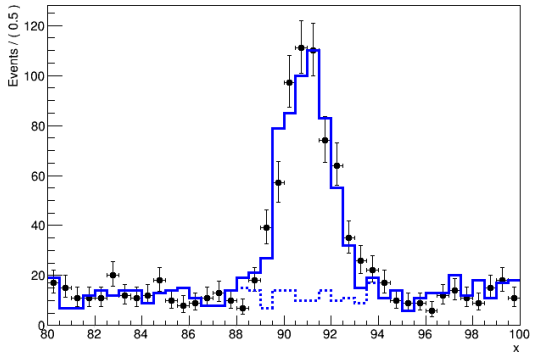


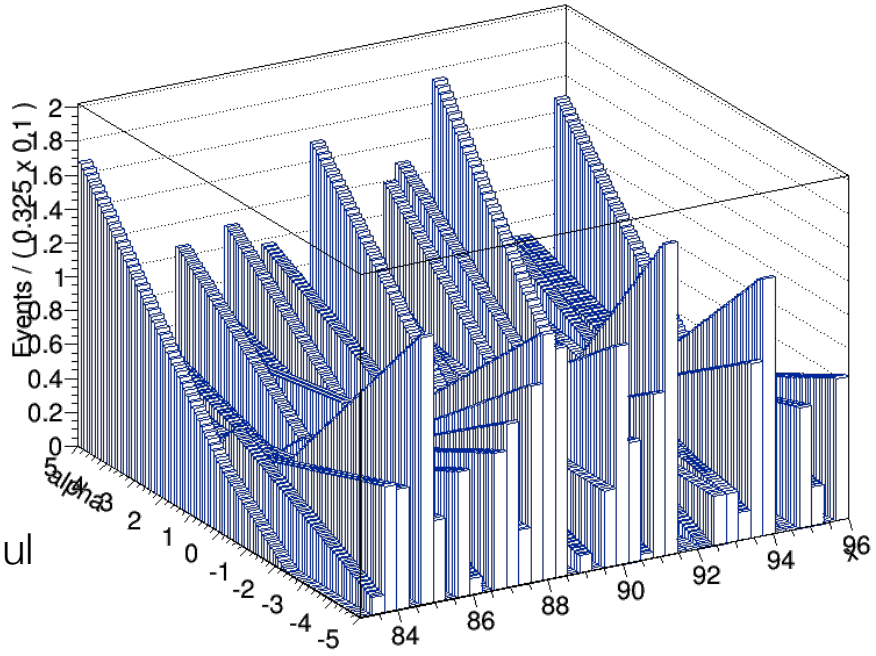
Visualization of 2D interpolation



Shape, rate or no systematic?

- Be judicious with modeling of systematic with little or no significant change in shape (w.r.t MC template statistics)
  - Example morphing of a very subtle change in the background model
  - Is this a meaningful new degree of freedom in the likelihood model?

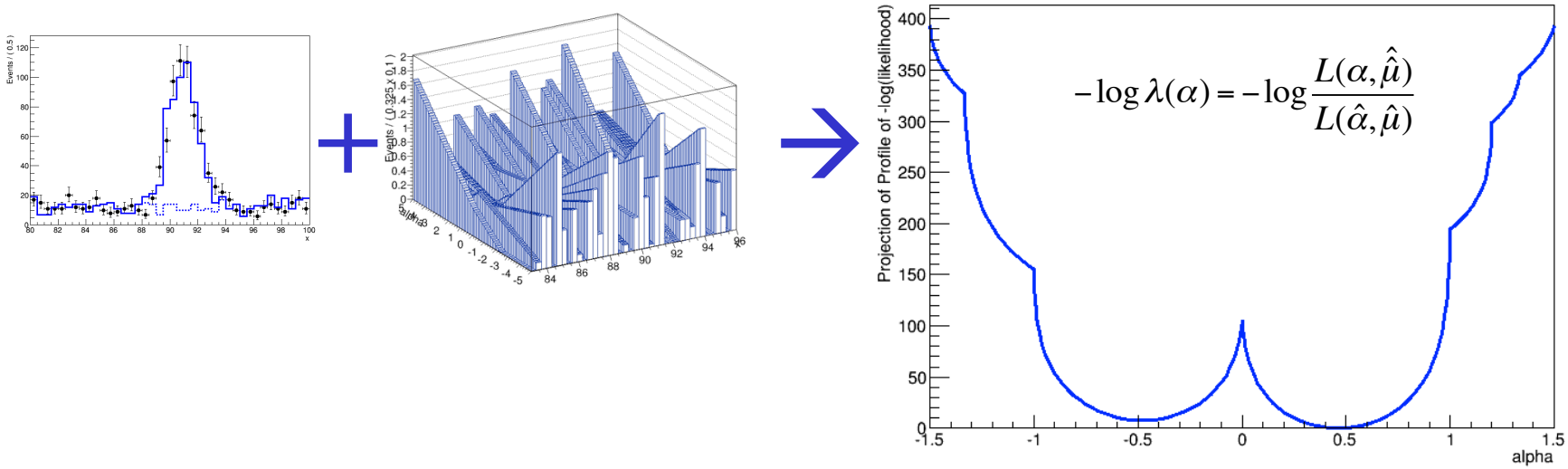




  - A χ2 or KS test between nominal and alternate template can help to decide if a shape uncertainty is meaningul
  - Most systematic uncertainties affect both rate and shape, but can make independent decision on modeling rate (which less likely to affect fit stability)

Fit stability due to insignificant shape systematics

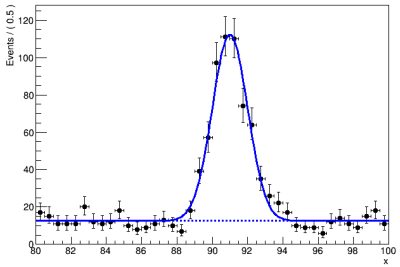
- Shape of profile likelihood in NP α clearly raises two points



- 1) Numerical minimization process will be 'interesting'
- 2) MC statistical effects induce strongly defined minima that are fake
  - Because for this example all three templates were sampled from the same parent distribution (a uniform distribution)

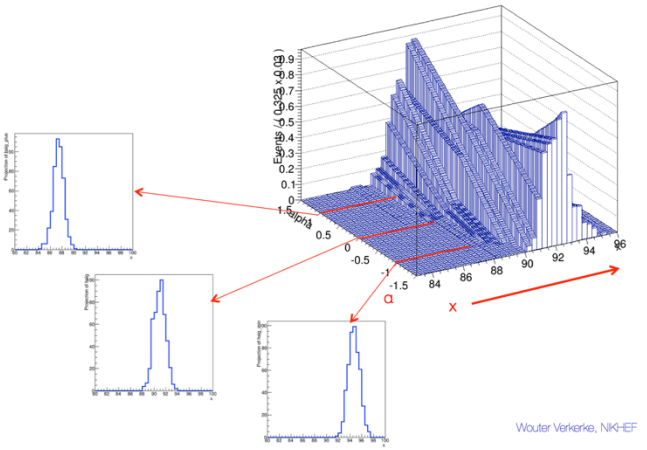
Recap on shape systematics & template morphing

- Implementation of shape systematic in likelihoods modeling distributions conceptually no different that rate systematics in counting experiments



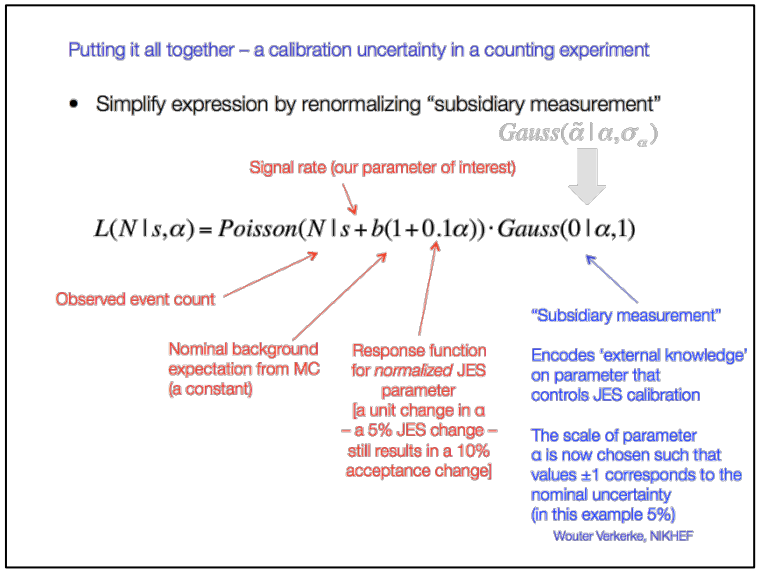
$$L(\vec{m}_{ll} \mid \mu, \alpha_{LES}) = \prod_i \left[\mu \cdot \text{Gauss}(m_{ll}^{(i)}, 91 \cdot (1 + 2\alpha_{LES}, 1) + (1 - \mu) \cdot \text{Uniform}(m_{ll}^{(i)}) \right] \cdot Gauss(0 \mid \alpha_{LES}, 1)$$

- For template modes obtained from MC simulation template provides a technical solution to implement response function
  - Simplest strategy piecewise linear interpolation, but only works well for small changes
  - Moment morphing better adapted to modeling of shifting distributions
  - Both algorithms extend to n-dimensional interpolation to model multiple systematic NPs in response function
  - Be judicious in modeling 'weak' systematics: MC systematic uncertainties will dominate likelihood



Example 1: counting expt

- Will now demonstrate how to construct a model for a counting experiment with a systematic uncertainty

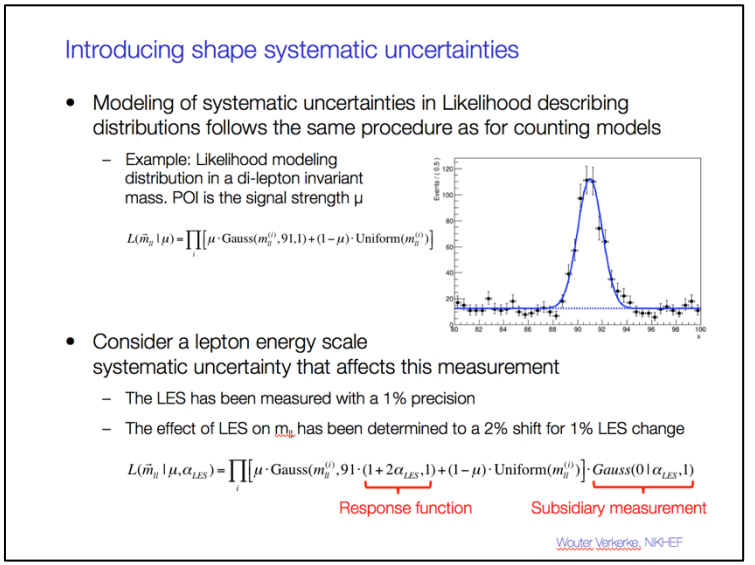


$$L(N\,|\,s,\alpha) = Poisson(N\,|\,s+b(1+0.1\alpha))\cdot Gauss(0\,|\,\alpha,1)$$

```
// Subsidiary measurement of alpha
w.faxtory("Gaussian::subs(0,alpha[-5,5],1)") ;

// Response function mu(alpha)
w.factory("expr::mu('s+b(1+0.1*alpha)',s[20],b[20],alpha)") ;

// Main measurement
w.factory("Poisson::p(N[0,10000],mu)");

// Complete model Physics*Subsidiary
w.factory("PROD::model(p,subs)") ;
```

Example 2: unbinned L with syst.

- Will now demonstrate how to code complete example of the unbinned profile likelihood of Section 5:



$$L(\vec{m}_{ll} \mid \mu, \alpha_{LES}) = \prod_i \left[\mu \cdot \text{Gauss}(m_{ll}^{(i)}, 91 \cdot (1+2\alpha_{LES}), 1) + (1-\mu) \cdot \text{Uniform}(m_{ll}^{(i)}) \right] \cdot Gauss(0 \mid \alpha_{LES}, 1)$$

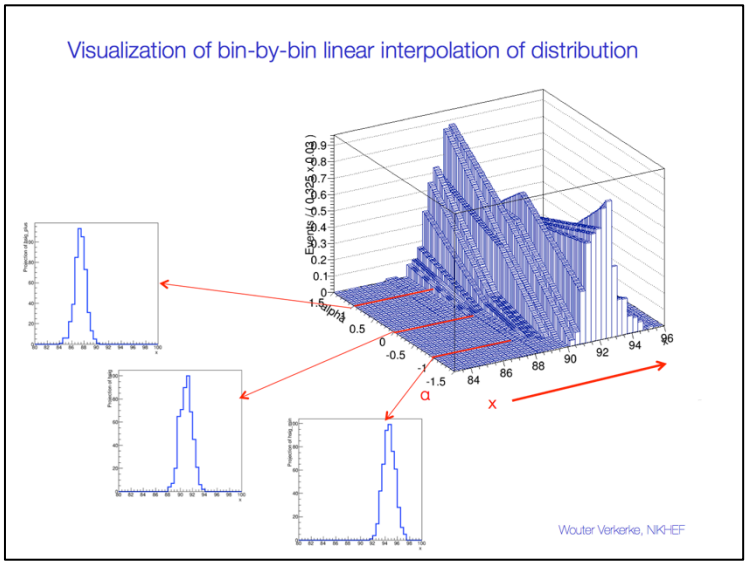
```
// Subsidiary measurement of alpha
w.factory("Gaussian::subs(0,alpha[-5,5],1)");

// Response function m(alpha)
w.factory("expr::m_a("m*(1+2alpha)",m[91,80,100],alpha)") ;

// Signal model
w.factory("Gaussian::sig(x[80,100],m_a,s[1])")

// Complete model Physics(signal plus background)*Subsidiary
w.factory("PROD::model(SUM(mu[0,1]*sig,Uniform::bkg(x)),subs)") ;
```

Example 3 : binned L with syst

- Example of template morphing systematic in a binned likelihood

$$s_i(\alpha,\ldots) = \begin{cases} s_i^0 + \alpha\cdot(s_i^+ - s_i^0) & \forall\alpha > 0 \\ s_i^0 + \alpha\cdot(s_i^0 - s_i^-) & \forall\alpha < 0 \end{cases}$$

$$L(\vec{N}\,|\,\alpha,\vec{s}^-,\vec{s}^0,\vec{s}^+) = \prod_{bins} P(N_i\,|\,s_i(\alpha,s_i^-,s_i^0,s_i^+))\cdot G(0\,|\,\alpha,1)$$




```
// Import template histograms in workspace
w.import(hs_0,hs_p,hs_m) ;

// Construct template models from histograms
w.factory("HistFunc::s_0(x[80,100],hs_0)") ;
w.factory("HistFunc::s_p(x,hs_p)") ;
w.factory("HistFunc::s_m(x,hs_m)") ;

// Construct morphing model
w.factory("PiecewiseInterpolation::sig(s_0,s_,m,s_p,alpha[-5,5])") ;

// Construct full model
w.factory("PROD::model(ASUM(sig,bkg,f[0,1]),Gaussian(0,alpha,1))") ;
```