

Statistics

W. Verkerke

What do we want to know?

- **Physics questions we have...**
 - Does the (SM) Higgs boson exist?
 - What is its production cross-section?
 - What is its boson mass?

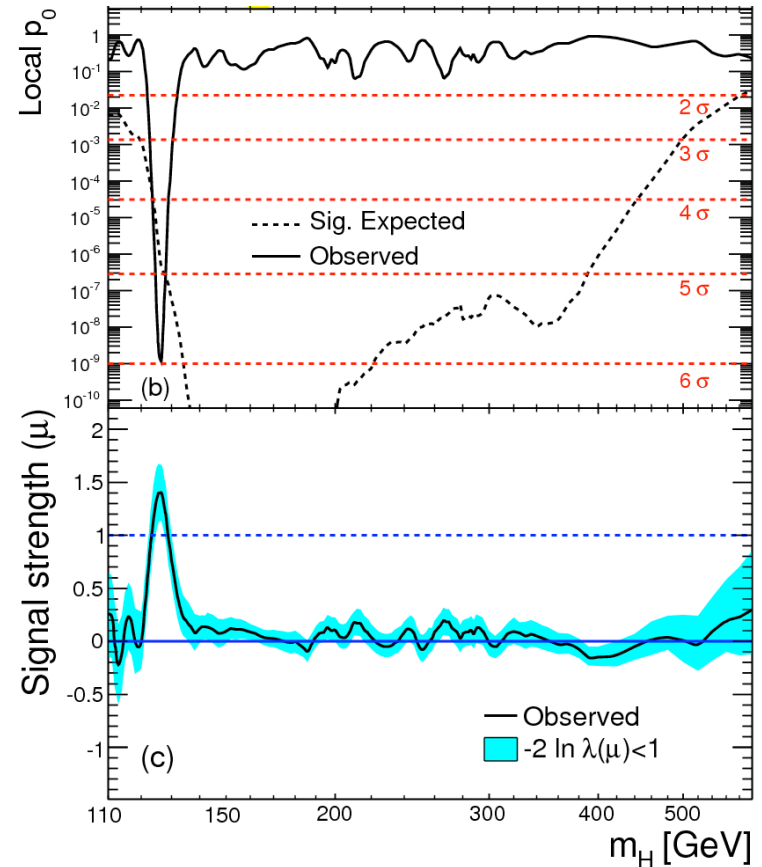


- **Statistical tests construct probabilistic statements:** $p(\text{theo}|\text{data})$, or $p(\text{data}|\text{theo})$
 - Hypothesis testing (discovery)
 - (Confidence) intervals
Measurements & uncertainties



- **Result: *Decision* based on tests**

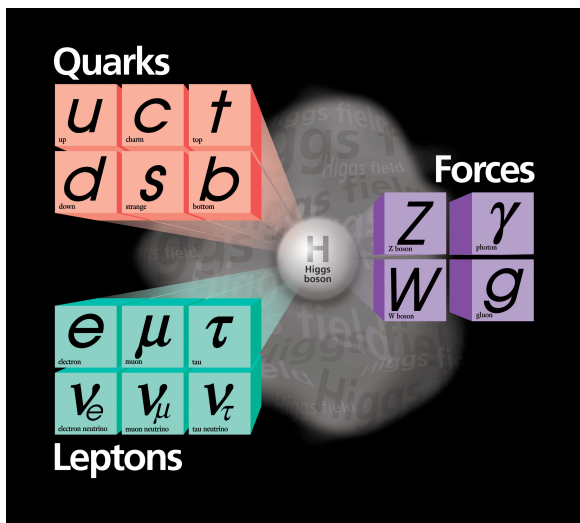
“As a layman I would now say: I think we have it”



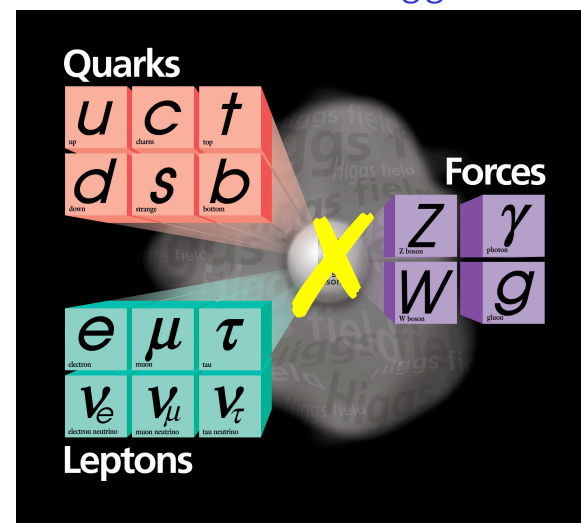
How do we do this?

- All experimental results start with formulation of a (physics) theory
- Examples of HEP **physics** models being tested

The Standard Model

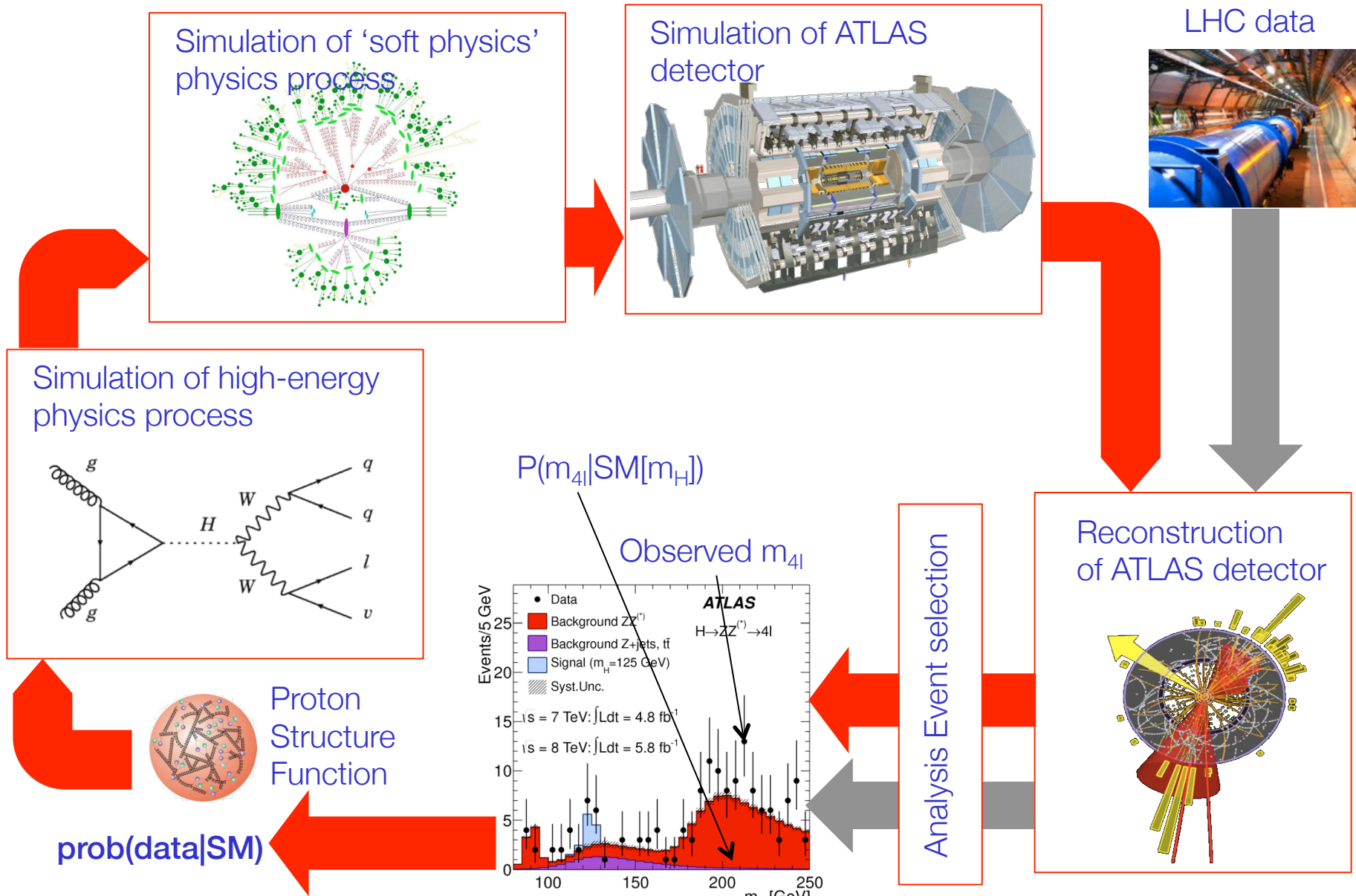


The SM without a Higgs boson



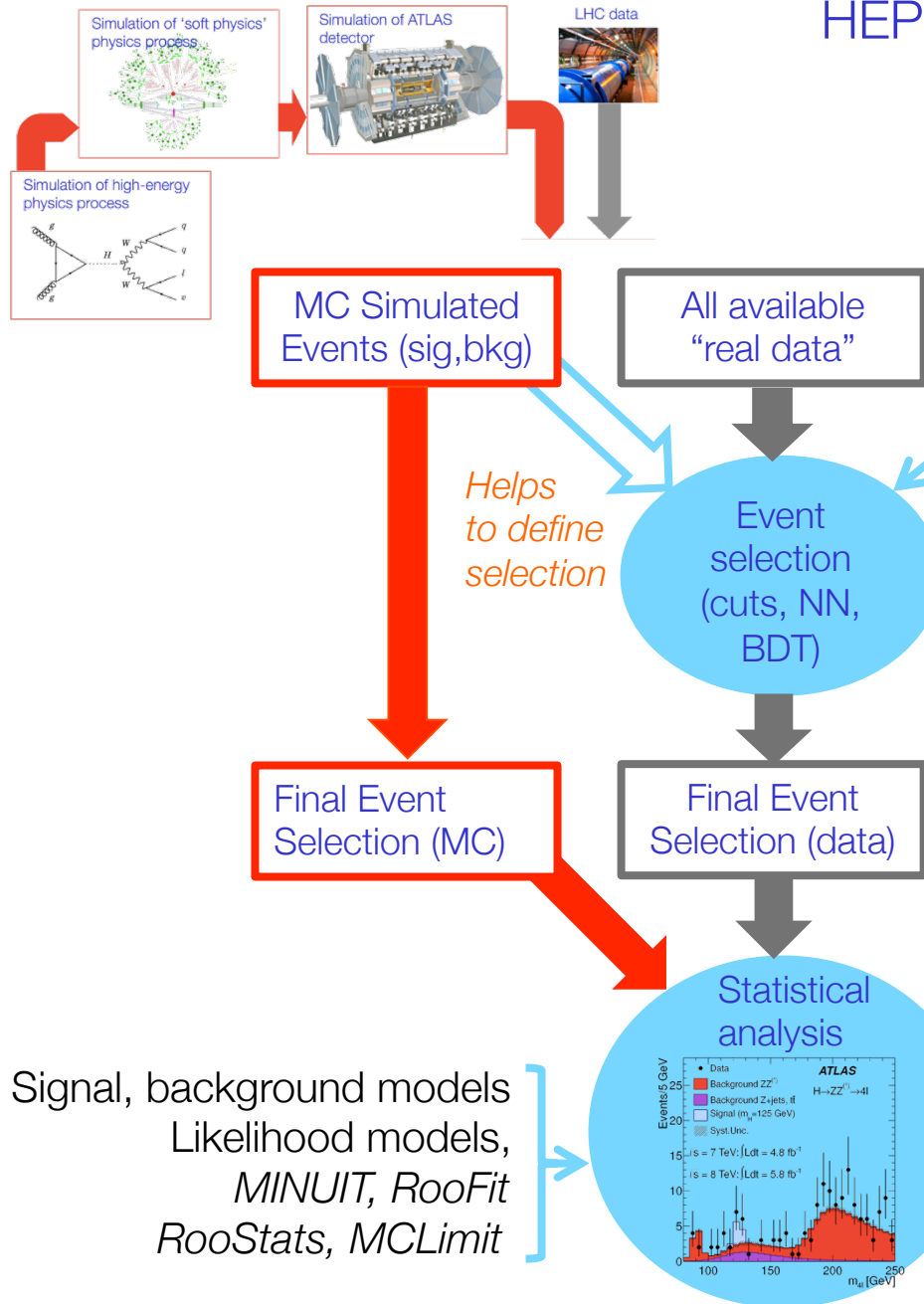
- Next, you design a measurement to be able to *test* model
 - Via chain of physics simulation, showering MC, detector simulation and analysis software, a physics model is reduced to a **statistical** model

An overview of HEP data analysis procedures

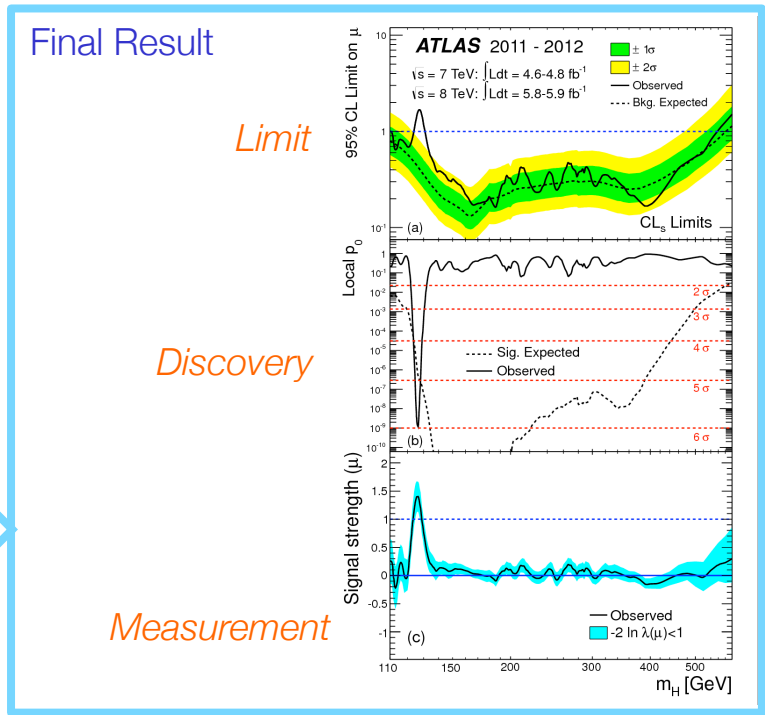


An overview of HEP data analysis procedures

HEP workflow: data analysis in practice



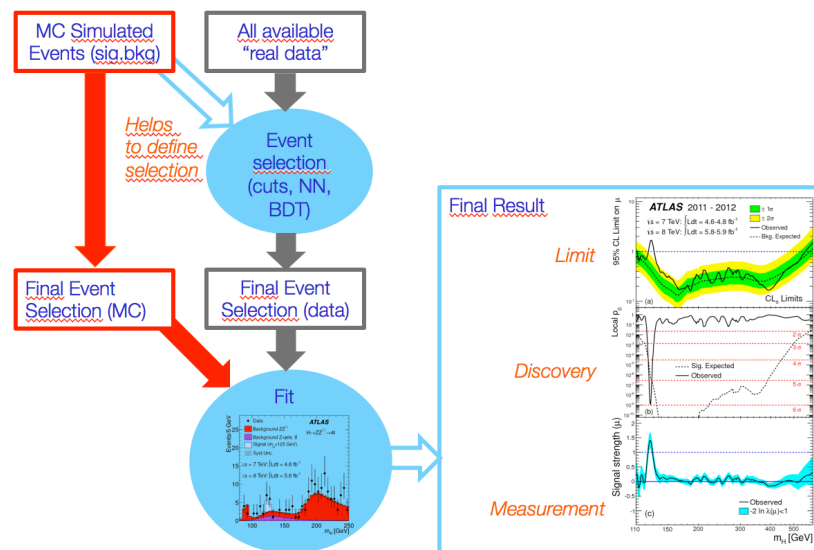
N-tuples
Cut-flows,
Multi-variate analysis (NN, BDT)
ROOT, TMVA, NeuroBayes



From physics theory to statistical model

- HEP “Data Analysis” is for large part **the reduction of a physics theory to a statistical model**

Physics Theory: Standard Model with 125 GeV Higgs boson

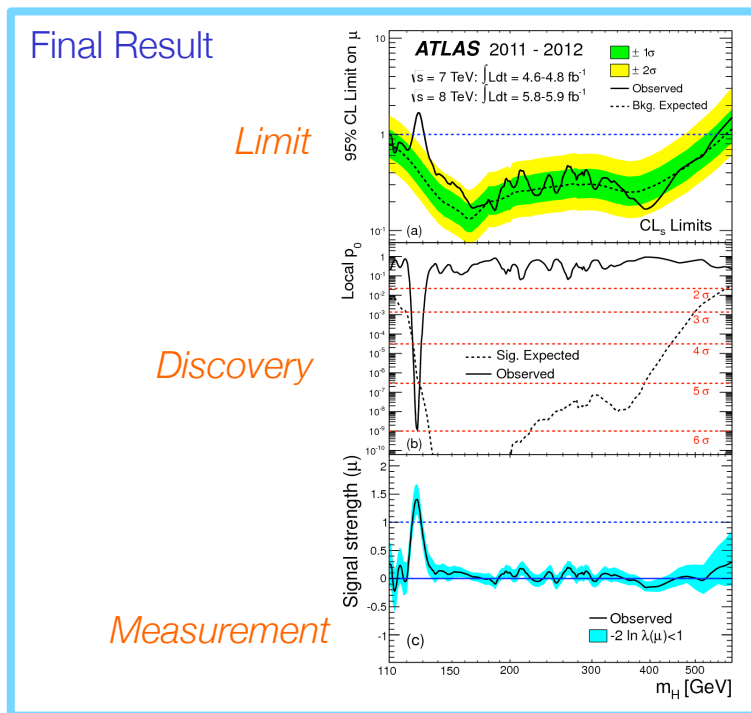


Statistical Model: *Given a measurement x (e.g. an event count) what is the probability to observe each possible value of x , under the hypothesis that the physics theory is true.*

Once you have a statistical model, all physics knowledge has been abstracted into the model, and further steps in statistical inference are ‘procedural’ (no physics knowledge is required in principle)

From statistical model to a result

- The next step of the analysis is to confront your model with the data, and summarize the result in a probabilistic statement of some form



‘Confidence/Credible Interval’

$$\sigma/\sigma_{\text{SM}} (\text{H} \rightarrow \text{ZZ}) |_{m_{\text{H}}=150} < 0.3 \text{ @ 95\% C.L.}$$

‘p-value’

“Probability to observed this signal or more extreme, under the hypothesis of background-only is 1×10^9 ”

‘Measurement with variance estimate’

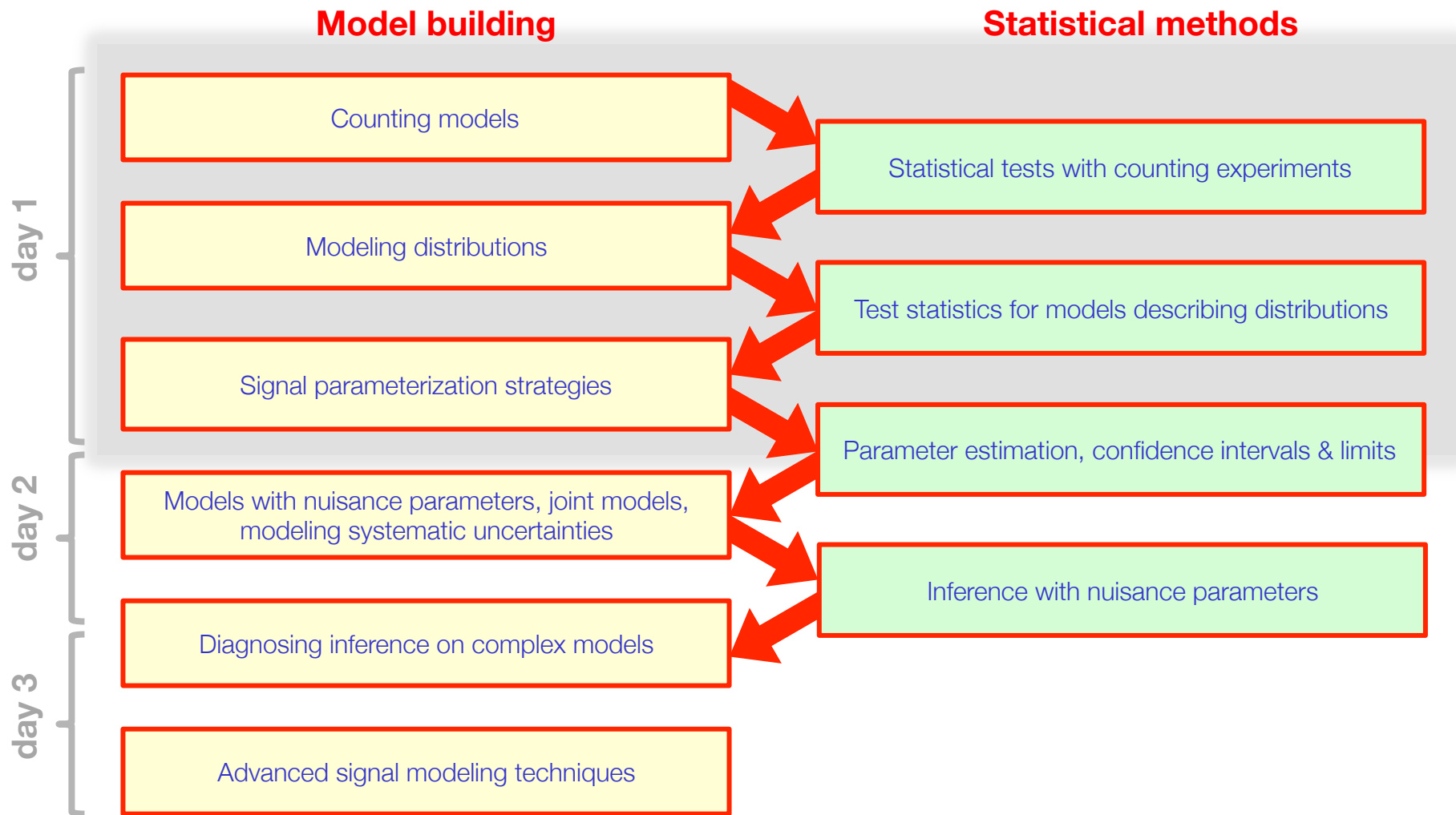
$$\sigma/\sigma_{\text{SM}} (\text{H} \rightarrow \text{ZZ}) |_{m_{\text{H}}=126} = 1.4 \pm 0.3$$

- The last step, usually not in a (first) paper, that you, or your collaboration, *decides* if your theory is valid



Roadmap of this course

- Start with basics, gradually build up to complexity

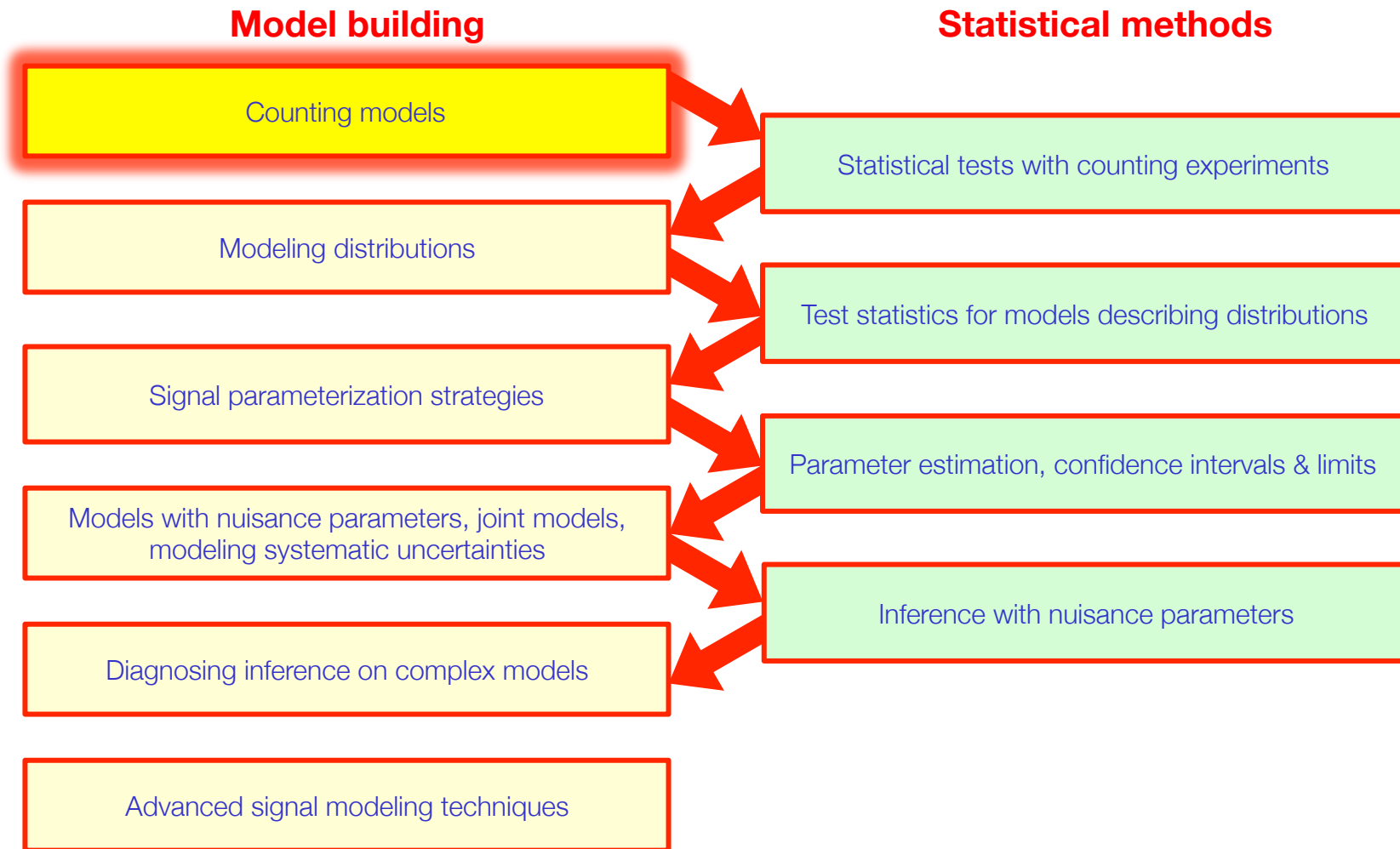


Model building 1

Basic distributions: Binomial, Poisson, Gaussian

Roadmap of this course

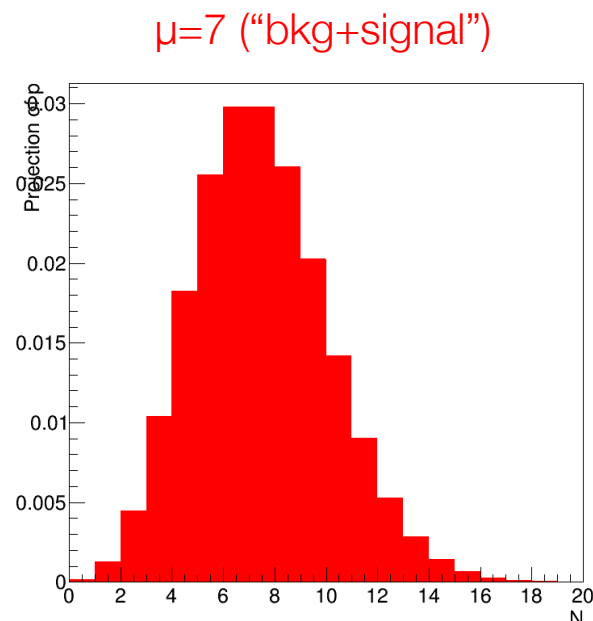
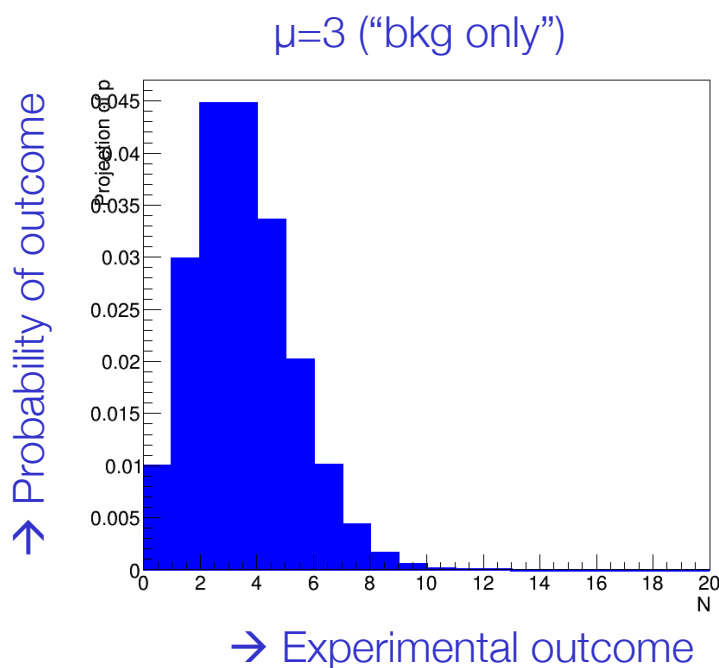
- Start with basics, gradually build up to complexity



The statistical world

- Central concept in statistics is the ‘**probability model**’
- *A probability model assigns a probability to each possible experimental outcome.*
- Example: a HEP counting experiment
 - Count number of ‘events’ in a fixed time interval → Poisson distribution
 - Given the *expected event count*, the probability model is fully specified

$$P(N | \mu) = \frac{\mu^N e^{-\mu}}{N!}$$



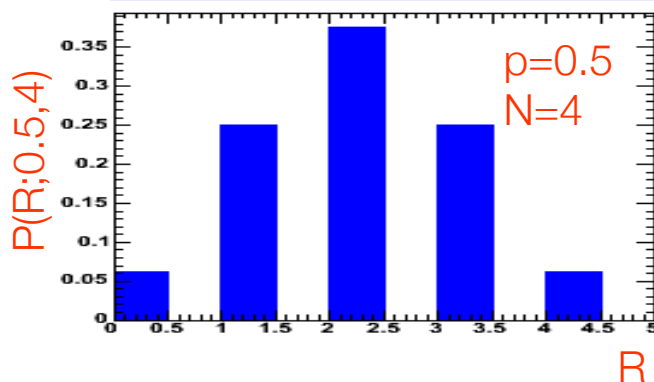
Intermezzo on distributions – The binomial distribution

- Simple **counting** experiment – Drawing marbles from a bowl
 - Bowl with marbles, **fraction p are black**, others are white
 - **Draw N marbles** from bowl, *put marble back after each drawing*
 - Distribution of R black marbles in drawn sample:

Probability of a
specific outcome
e.g. 'BBBWBWW'

Number of equivalent
permutations for that
outcome

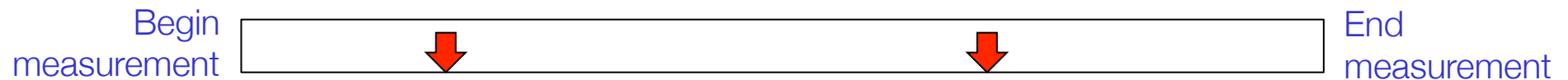
$$P(R; p, N) = p^R (1 - p)^{N-R} \frac{N!}{R!(N - R)!}$$



Binomial distribution

Basic Distributions – the Poisson distribution

- Sometimes we don't know the equivalent of the number of drawings
 - Example: Geiger counter
 - Sharp events occurring in a (time) continuum



- What distribution do we expect in measurement over a fixed amount of time?
 - Can be related to Binomial distribution by dividing time interval in fixed number of small intervals, counting #intervals with a collision



A probability model for LHC collisions

- For k expected collisions in measurement, probability of collision in one of N intervals is $k/N \rightarrow$ Now back to binomial distribution



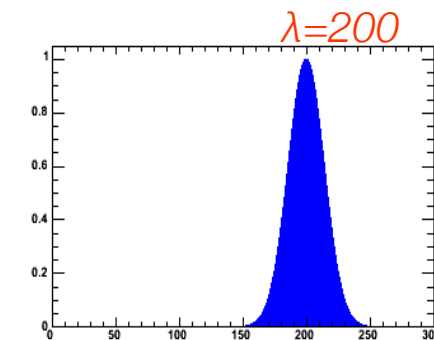
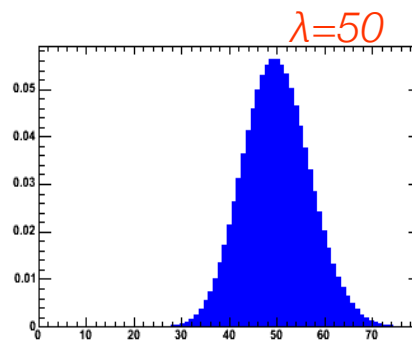
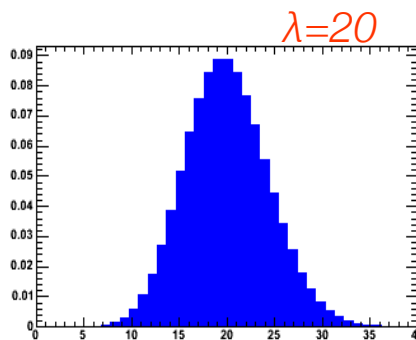
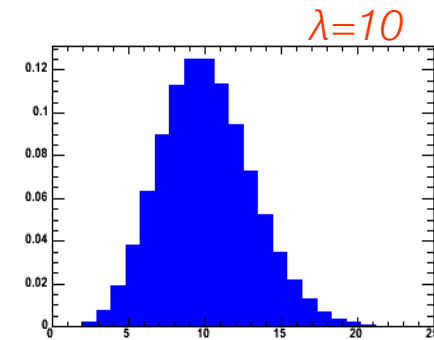
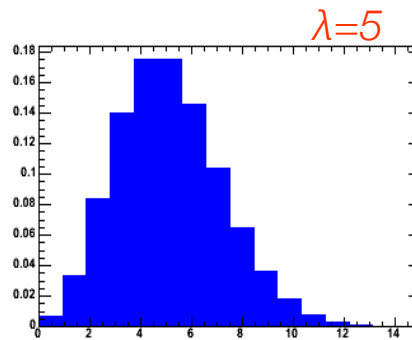
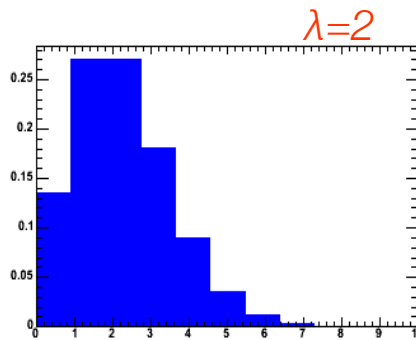
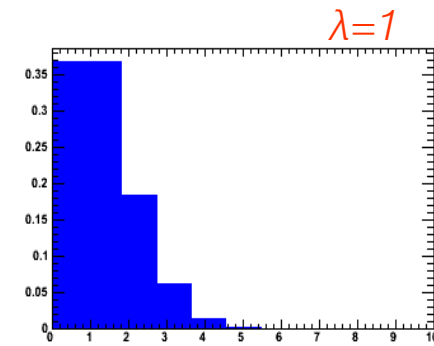
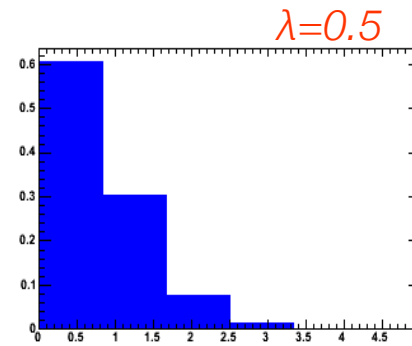
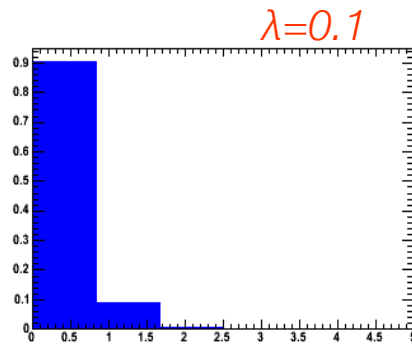
$$p(r \mid \frac{k}{N}, N) = \frac{k^r}{N^r} \left(1 - \frac{k}{N}\right)^{N-r} \frac{N!}{r!(N-r)!}$$

- Now take limit $N \rightarrow \infty$
(to avoid possibility of >1 collision per interval)

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{n!}{(n-r)!} &= n^r \\ \lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^{n-r} &= e^{-\lambda} \end{aligned} \quad \rightarrow \quad p(r \mid k) = \frac{e^{-k} k^r}{r!}$$

The Poisson distribution for values value of λ

$$p(r | k) = \frac{e^{-k} k^r}{r!}$$



Named after Simeon de Poisson – who was investigating the occurrence of judgement errors in the French judicial system

More properties of the Poisson distribution

$$P(r; \lambda) = \frac{e^{-\lambda} \lambda^r}{r!}$$

- Mean, variance:

$$\langle r \rangle = \lambda$$

$$V(r) = \lambda \quad \Rightarrow \quad \sigma = \sqrt{\lambda}$$

- Convolution of 2 Poisson distributions is also a Poisson distribution with $\lambda_{ab} = \lambda_a + \lambda_b$

$$P(r) = \sum_{r_A=0}^r P(r_A; \lambda_A) P(r - r_A; \lambda_B)$$

$$= e^{-\lambda_A} e^{-\lambda_B} \sum \frac{\lambda_A^{r_A} \lambda_B^{r-r_A}}{r_A! (r-r_A)!}$$

$$= e^{-(\lambda_A + \lambda_B)} \frac{(\lambda_A + \lambda_B)^r}{r!} \sum_{r_A=0}^r \frac{r!}{(r-r_A)!} \left(\frac{\lambda_A}{\lambda_A + \lambda_B} \right)^{r_A} \left(\frac{\lambda_B}{\lambda_A + \lambda_B} \right)^{r-r_A}$$

$$= e^{-(\lambda_A + \lambda_B)} \frac{(\lambda_A + \lambda_B)^r}{r!} \left(\frac{\lambda_A}{\lambda_A + \lambda_B} + \frac{\lambda_B}{\lambda_A + \lambda_B} \right)^r$$

$$= e^{-(\lambda_A + \lambda_B)} \frac{(\lambda_A + \lambda_B)^r}{r!}$$

Basic Distributions – The Gaussian distribution

- Look at **Poisson distribution** in limit of **large N**

$$P(r; \lambda) = e^{-\lambda} \frac{\lambda^r}{r!}$$

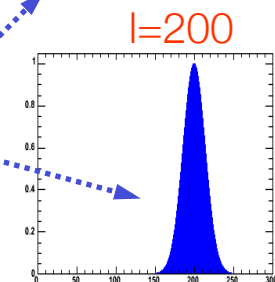
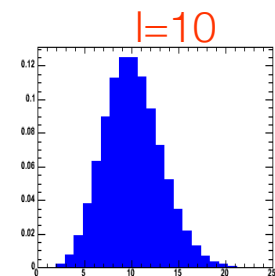
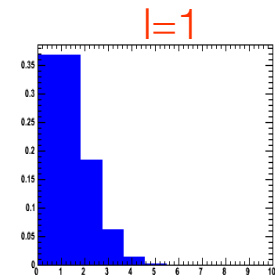
Take log, substitute, $r = l + x$,
and use $\ln(r!) \approx r \ln r - r + \ln \sqrt{2\pi r}$

$$\begin{aligned} \ln(P(r; \lambda)) &= -\lambda + r \ln \lambda - (r \ln r - r) - \ln \sqrt{2\pi r} \\ &= -\lambda + r \left[\ln \lambda - \ln \left(\lambda \left(1 + \frac{x}{\lambda} \right) \right) \right] + (\lambda + x) - \ln \sqrt{2\pi \lambda} \\ &\approx x - (\lambda - x) \left(\frac{x}{\lambda} + \frac{x^2}{2\lambda^2} \right) - \ln(2\pi \lambda) \\ &\approx \frac{-x^2}{2\lambda} - \ln(2\pi \lambda) \end{aligned}$$

Take exp

$$P(x) = \frac{e^{-x^2/2\lambda}}{\sqrt{2\pi\lambda}}$$

Familiar Gaussian distribution,
(approximation reasonable for $N > 10$)



Properties of the Gaussian distribution

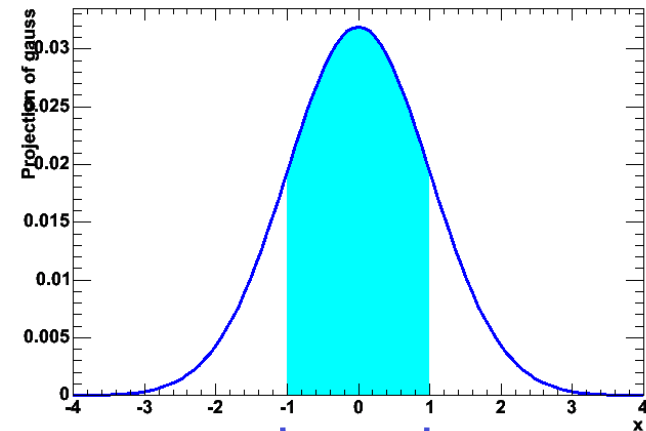
$$P(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2}$$

- *Mean* and *Variance*

$$\langle x \rangle = \int_{-\infty}^{+\infty} x P(x; \mu, \sigma) dx = \mu$$

$$V(x) = \int_{-\infty}^{+\infty} (x - \mu)^2 P(x; \mu, \sigma) dx = \sigma^2$$

$$\sigma = \sigma$$



- Integrals of Gaussian

68.27% within 1σ	90% → 1.645σ
95.43% within 2σ	95% → 1.96σ
99.73% within 3σ	99% → 2.58σ
	99.9% → 3.29σ

The Gaussian as 'Normal distribution'

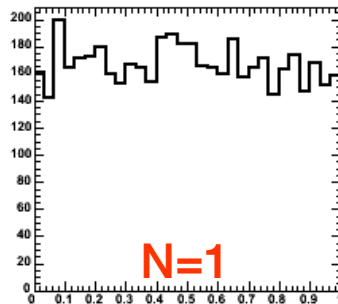
- Why are distributions often Gaussian?
- The **Central Limit Theorem** says
- If you take the sum X of N independent measurements x_i , each taken from a distribution of mean m_i , a variance $V_i = \sigma_i^2$, the distribution for x

(a) has expectation value $\langle X \rangle = \sum_i \mu_i$

(b) has variance $V(X) = \sum_i V_i = \sum_i \sigma_i^2$

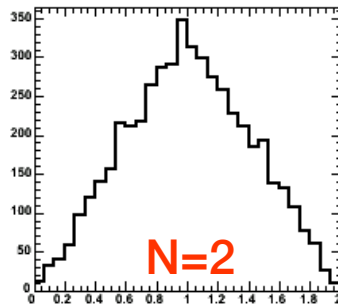
(c) becomes Gaussian as $N \rightarrow \infty$

Demonstration of Central Limit Theorem



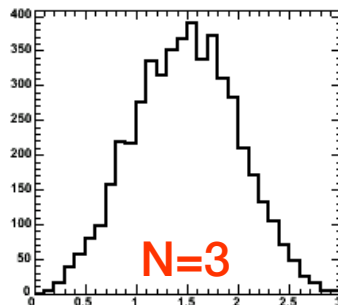
← 5000 numbers taken at random from a uniform distribution between $[0, 1]$.

– Mean = $1/2$, Variance = $1/12$

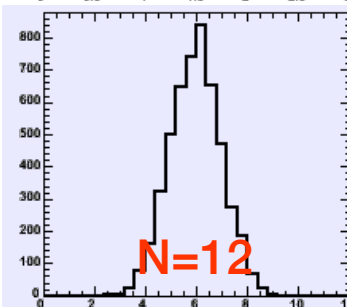


← 5000 numbers, each the sum of 2 random numbers, i.e. $X = x_1 + x_2$.

– Triangular shape



← Same for 3 numbers,
 $X = x_1 + x_2 + x_3$



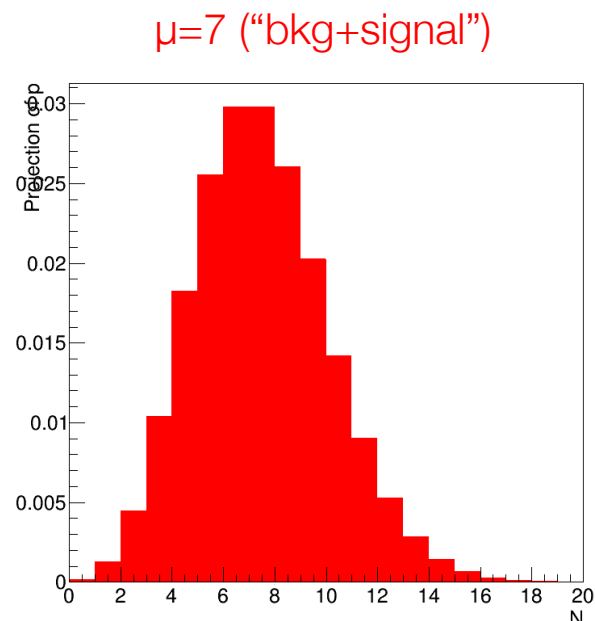
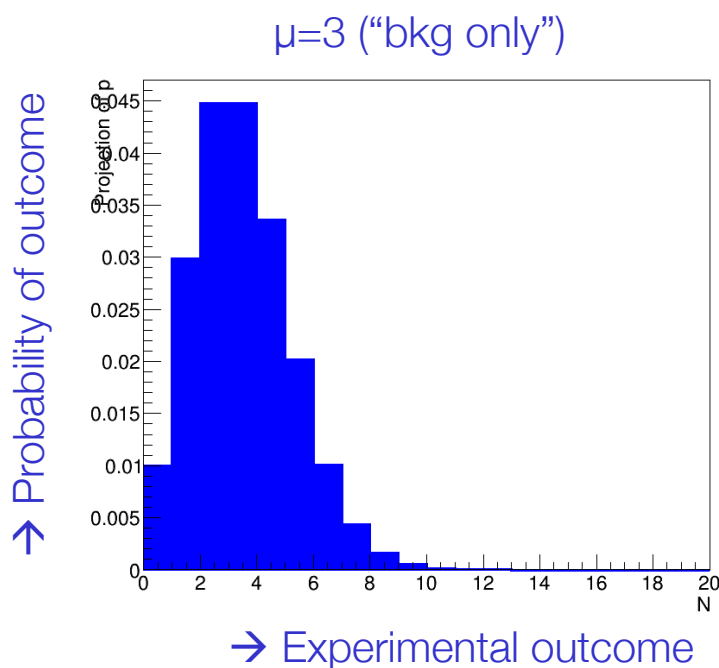
← Same for 12 numbers, overlaid curve is exact Gaussian distribution

Important: tails of distribution converge very slowly CLT often *not* applicable for '5 sigma' discoveries

The statistical world

- Central concept in statistics is the ‘**probability model**’
- *A probability model assigns a probability to each possible experimental outcome.*
- Example: a HEP counting experiment
 - Count number of ‘events’ in a fixed time interval → Poisson distribution
 - Given the *expected event count*, the probability model is fully specified

$$P(N | \mu) = \frac{\mu^N e^{-\mu}}{N!}$$

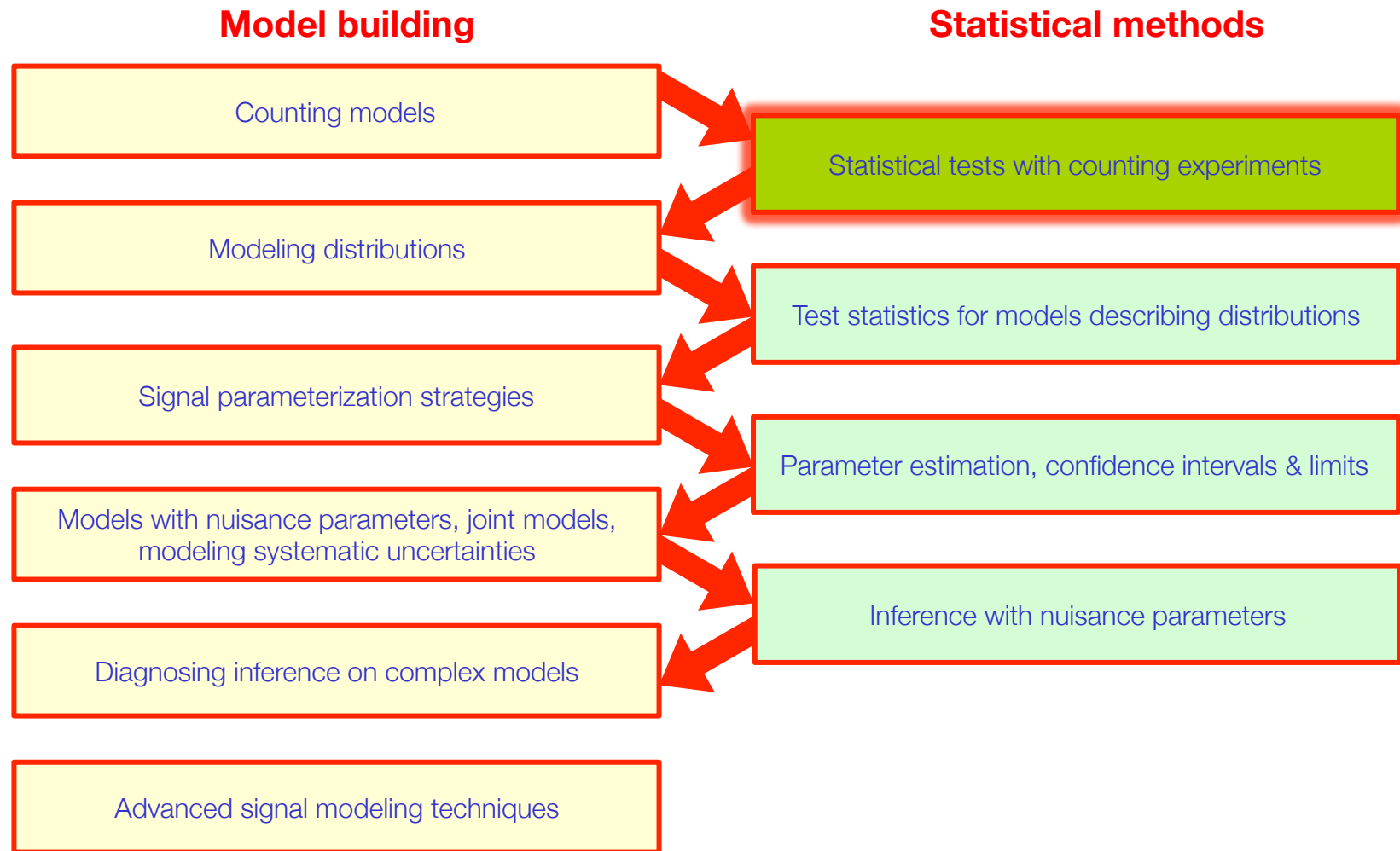


Statistical methods 1

Hypothesis testing, p-values, odds ratios (demonstrated on simple
Poisson counting experiments)

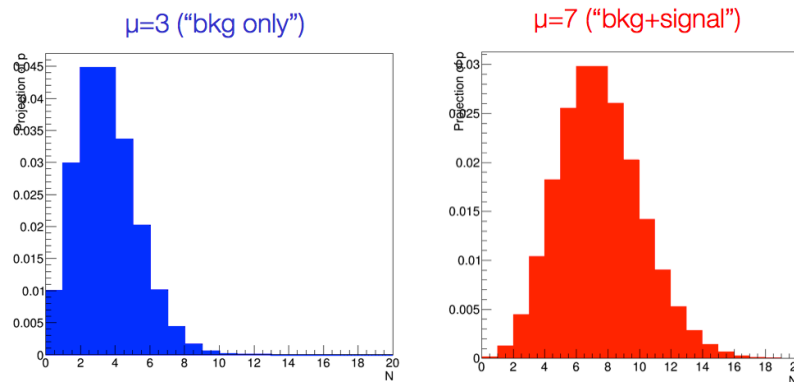
Roadmap of this course

- Start with basics, gradually build up to complexity



Probabilities vs conditional probabilities

- Note that probability models strictly give *conditional* probabilities (with the condition being that the underlying hypothesis is true)



Definition:
 $P(\text{data}|\text{hypo})$ is called
the likelihood

$$P(N) \rightarrow P(N | H_{bkg}) \quad P(N) \rightarrow P(N | H_{sig+bkg})$$

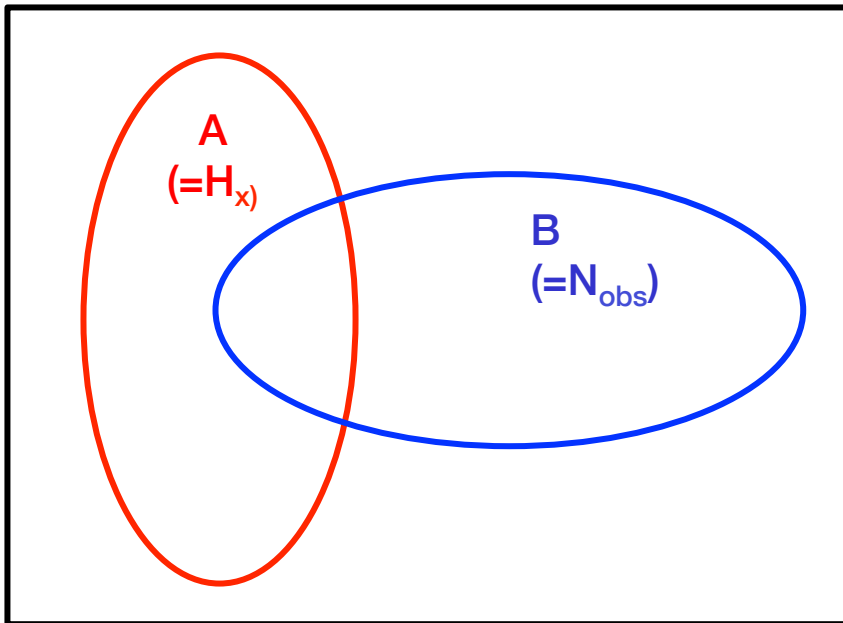
- Suppose we measure $N=7$ then can calculate

$$L(N=7|H_{bkg})=2.2\% \quad L(N=7|H_{sig+bkg})=14.9\%$$

- Data is more likely under sig+bkg hypothesis than bkg-only hypo*
- Is this what we want to know? Or do we want to know $L(H_{s+b}|N=7)$?

Inverting the conditionality on probabilities

- Do $L(7|H_b)$ and $L(7|H_{sb})$ provide you enough information to calculate $P(H_b|7)$ and $P(H_{sb}|7)$
- **No!**
- Image the 'whole space' and two subsets A and B



$$P(A) = \frac{\text{small blue oval}}{\text{large blue rectangle}}$$
$$P(B) = \frac{\text{small blue oval}}{\text{large blue rectangle}}$$
$$P(A|B) = \frac{\text{tiny blue oval}}{\text{large blue oval}}$$
$$P(B|A) = \frac{\text{tiny blue oval}}{\text{large blue oval}}$$

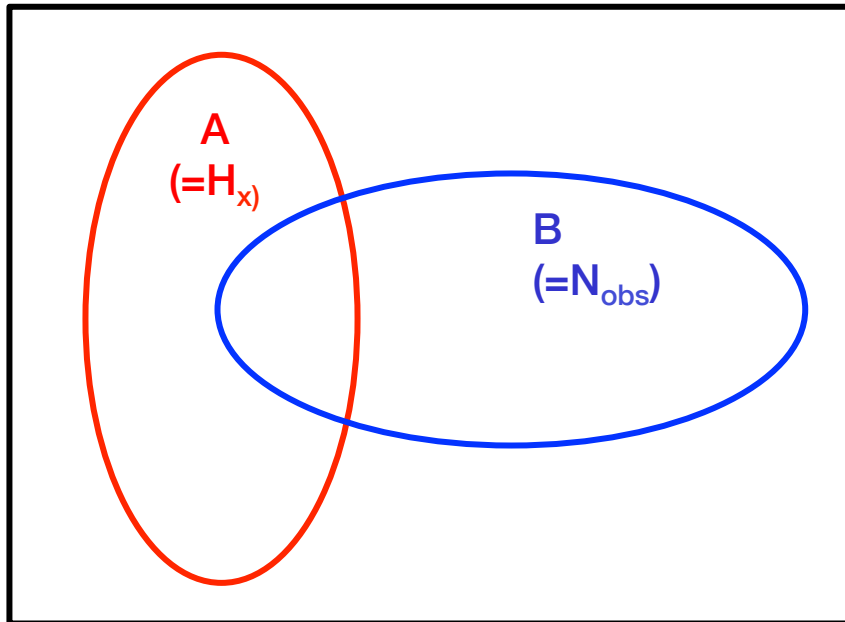
↓

$$P(A|B) \neq P(B|A)$$

↓

$$P(7|H_b) \neq P(H_b|7)$$

Inverting the conditionality on probabilities



$$P(A) = \frac{\text{blue oval}}{\text{blue square}} \quad P(B) = \frac{\text{blue oval}}{\text{blue square}}$$
$$P(A|B) = \frac{\text{small blue oval}}{\text{large blue oval}} \quad P(B|A) = \frac{\text{small blue oval}}{\text{large blue oval}}$$



$$P(A|B) \neq P(B|A)$$



but you can deduce
their relation



$$\Rightarrow P(B|A) = P(A|B) \times P(B) / P(A)$$

$$P(A) \times P(B|A) = \frac{\text{blue oval}}{\text{blue square}} \times \frac{\text{small blue oval}}{\text{large blue oval}} = \frac{\text{small blue oval}}{\text{blue square}} = P(A \cap B)$$

$$P(B) \times P(A|B) = \frac{\text{blue oval}}{\text{blue square}} \times \frac{\text{small blue oval}}{\text{large blue oval}} = \frac{\text{small blue oval}}{\text{blue square}} = P(A \cap B)$$

Inverting the conditionality on probabilities

- This conditionality inversion relation is known as **Bayes Theorem**

$$P(B|A) = P(A|B) \times P(B)/P(A)$$

Essay "Essay Towards Solving a Problem in the Doctrine of Chances" published in Philosophical Transactions of the Royal Society of London in 1764



Thomas Bayes (1702-61)

- And choosing A=data and B=theory

$$P(\text{theo}|\text{data}) = P(\text{data}|\text{theo}) \times P(\text{theo}) / P(\text{data})$$

- *Return to original question:*

Do you $L(7|H_b)$ and $L(7|H_{sb})$ provide you enough information to calculate $P(H_b|7)$ and $P(H_{sb}|7)$

- **No!** → Need $P(A)$ and $P(B)$ → **Need $P(H_b)$, $P(H_{sb})$ and $P(7)$**

Inverting the conditionality on probabilities

- **What is P(data)?**

$$P(\text{theo}|\text{data}) = P(\text{data}|\text{theo}) \times P(\text{theo}) / P(\text{data})$$

- It is the probability of the data under *any* hypothesis
 - For Example for two competing hypothesis H_b and H_{sb}

$$P(N) = L(N|H_b)P(H_b) + L(N|H_{sb})P(H_{sb})$$

and generally for N hypotheses

$$P(N) = \sum_i P(N|H_i)P(H_i)$$

- Bayes theorem reformulated using law of total probability

$$P(\text{theo}|\text{data}) = \frac{L(\text{data}|\text{theo}) \times P(\text{theo})}{\sum_i L(\text{data}|\text{theo-i})P(\text{theo-i})}$$

- *Return to original question:* Do you $L(7|H_b)$ and $L(7|H_{sb})$ provide you enough information to calculate $P(H_b|7)$ and $P(H_{sb}|7)$
No! → Still need $P(H_b)$ and $P(H_{sb})$

Prior probabilities

- What is the **meaning** of $P(H_b)$ and $P(H_{sb})$?
 - They are the probability assigned to hypothesis H_b *prior to the experiment*.
- What are the **values** of $P(H_b)$ and $P(H_{sb})$?
 - Can be result of an earlier measurement
 - Or more generally (e.g. when there are no prior measurement) they quantify *a prior degree of belief* in the hypothesis
- **Example** – suppose prior belief $P(H_{sb})=50\%$ and $P(H_b)=50\%$

$$\begin{aligned} P(H_{sb}|N=7) &= \frac{P(N=7|H_{sb}) \times P(H_{sb})}{[P(N=7|H_{sb})P(H_{sb})+P(N=7|H_b)P(H_b)]} \\ &= \frac{0.149 \times 0.50}{[0.149 \times 0.5 + 0.022 \times 0.5]} = 87\% \end{aligned}$$

- Observation $N=7$ strengthens belief in hypothesis H_{sb} (and weakens belief in $H_b \rightarrow 13\%$)

Interpreting probabilities

- We have seen

probabilities assigned observed experimental outcomes

(probability to observed 7 events under some hypothesis)

probabilities assigned to hypotheses

(prior probability for hypothesis H_{sb} is 50%)

which are conceptually different.

- How to interpret probabilities – two schools

Bayesian probability = (subjective) degree of belief $P(\text{theo}|\text{data})$
 $P(\text{data}|\text{theo})$

Frequentist probability = fraction of outcomes in $P(\text{data}|\text{theo})$
future repeated identical experiments

*“If you’d repeat this experiment identically many times,
in a fraction P you will observe the same outcome”*

Interpreting probabilities

- Frequentist:
Constants of nature are fixed – you cannot assign a probability to these. Probabilities are restricted to observable experimental results
 - “The Higgs either exists, or it doesn’t” – you can’t assign a probability to that
 - Definition of $P(\text{data}|\text{hypo})$ is objective (and technical)
- Bayesian:
Probabilities can be assigned to constants of nature
 - Quantify your *belief* in the existence of the Higgs – can assign a probability
 - But it can be very difficult to assign a meaningful number (e.g. Higgs)
- **Example of weather forecast**

Bayesian: “*The probability it will rain tomorrow is 95%*”

- Assigns probability to constant of nature (“rain tomorrow”)
 $P(\text{rain-tomorrow}|\text{satellite-data}) = 95\%$

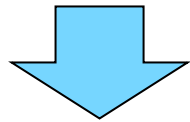
Frequentist: “*If it rains tomorrow,
95% of time satellite data looks like what we observe now*”

- Only states $P(\text{satellite-data}|\text{rain-tomorrow})$

Back to H_b/H_{sb} - Formulating evidence for discovery of H_{sb}

- Given a scenario with exactly two competing hypotheses
- In the Bayesian school you can cast evidence as an odd-ratio

$$O_{prior} \equiv \frac{P(H_{sb})}{P(H_b)} = \frac{P(H_{sb})}{1 - P(H_{sb})} \quad \text{If } p(H_{sb})=p(H_b) \rightarrow \text{Odds are 1:1}$$



'Bayes Factor' K multiplies prior odds

$$O_{posterior} \equiv \frac{L(x | H_{sb})P(H_{sb})}{L(x | H_b)P(H_b)} = \frac{L(x | H_{sb})}{L(x | H_b)} O_{prior}$$

If $\frac{P(\text{data}|H_b)=10^{-7}}{P(\text{data}|H_{sb})=0.5}$ $K=2.000.000 \rightarrow$ Posterior odds are 2.000.000 : 1

Formulating evidence for discovery

- In the frequentist school you restrict yourself to $P(\text{data}|\text{theory})$ and there is no concept of ‘priors’
 - But given that you consider (exactly) 2 competing hypothesis, very low probability for data under H_b lends credence to ‘discovery’ of H_{sb} (since H_b is ‘ruled out’). Example

$$\begin{array}{l} P(\text{data}|H_b)=10^{-7} \\ P(\text{data}|H_{sb})=0.5 \end{array} \quad \rightarrow \quad \text{“}H_b \text{ ruled out”} \rightarrow \text{“Discovery of } H_{sb}\text{”}$$

- Given importance to interpretation of the lower probability, it is customary to quote it in “physics intuitive” form: Gaussian σ .
 - E.g. ‘5 sigma’ \rightarrow probability of 5 sigma Gaussian fluctuation $=2.87 \times 10^{-7}$
- No formal rules for ‘discovery threshold’
 - Discovery also assumes data is not too unlikely under H_{sb} . If not, no discovery, but again no formal rules (“your good physics judgment”)
 - NB: In Bayesian case, both likelihoods low reduces Bayes factor K to $O(1)$

Taking decisions based on your result

- What are you going to do with the results of your measurement?
- Usually basis for a decision
 - **Science**: declare discovery of Higgs boson (or not), make press release, write new grant proposal
 - **Finance**: buy stocks or sell
- Suppose you believe $P(\text{Higgs}|\text{data})=99\%$.
- **Should declare discovery, make a press release?**
A: Cannot be determined from the given information!
- Need in addition: the utility function (or cost function),
 - The cost function specifies the relative costs (to You) of a Type I error (declaring model false when it is true) and a Type II error (not declaring model false when it is false).

Taking decisions based on your result

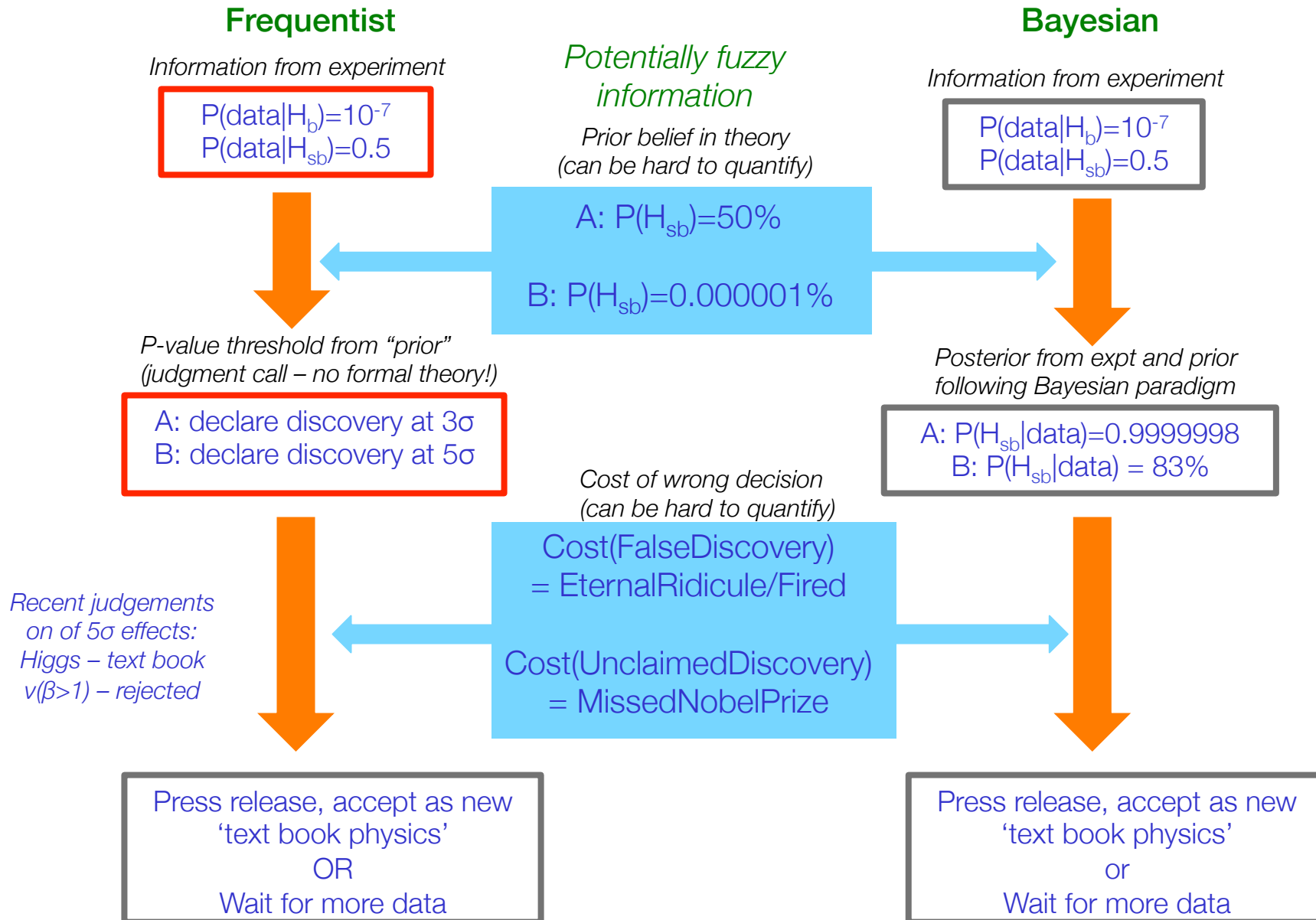
- Thus, your *decision*, such as where to invest your time or money, requires two subjective inputs:

Your **prior probabilities**, and

the **relative costs to You of outcomes**.

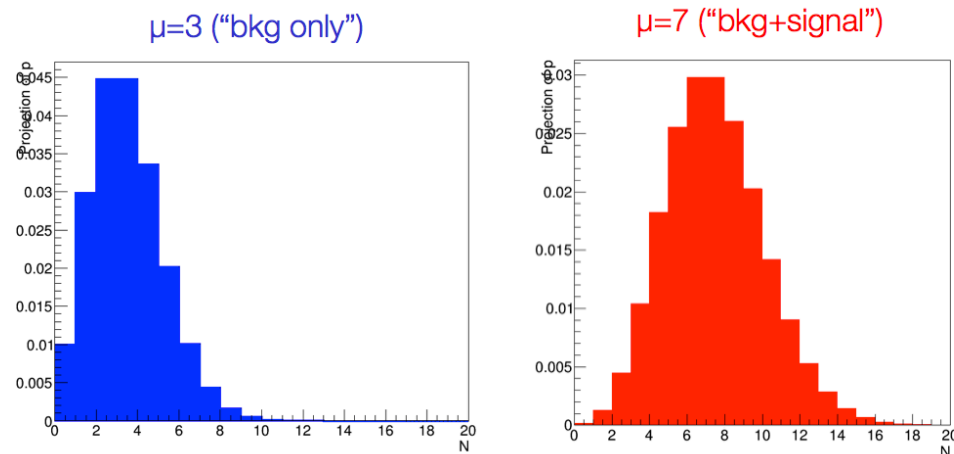
- Statisticians often focus on decision-making; in HEP, the tradition thus far is to communicate experimental results (well) short of formal decision calculations.
- Costs can be difficult to quantify in science.
 - What is the cost of declaring a false discovery?
 - Can be high (“Fleischman and Pons”), but hard to quantify
 - What is the cost of missing a discovery (“Nobel prize to someone else”), but also hard to quantify

How a theory becomes text-book physics



Summary on statistical test with simple hypotheses

- So far we considered simplest possible experiment we can do: counting experiment
- For a set of 2 or more completely specified (i.e. simple) hypotheses



→ Given probability models $P(N|bkg)$, and $P(N|sig)$
we can calculate $P(N_{obs}|H_x)$ under either hypothesis

→ With additional information on $P(H_i)$ we can also calculate $P(H_x|N_{obs})$

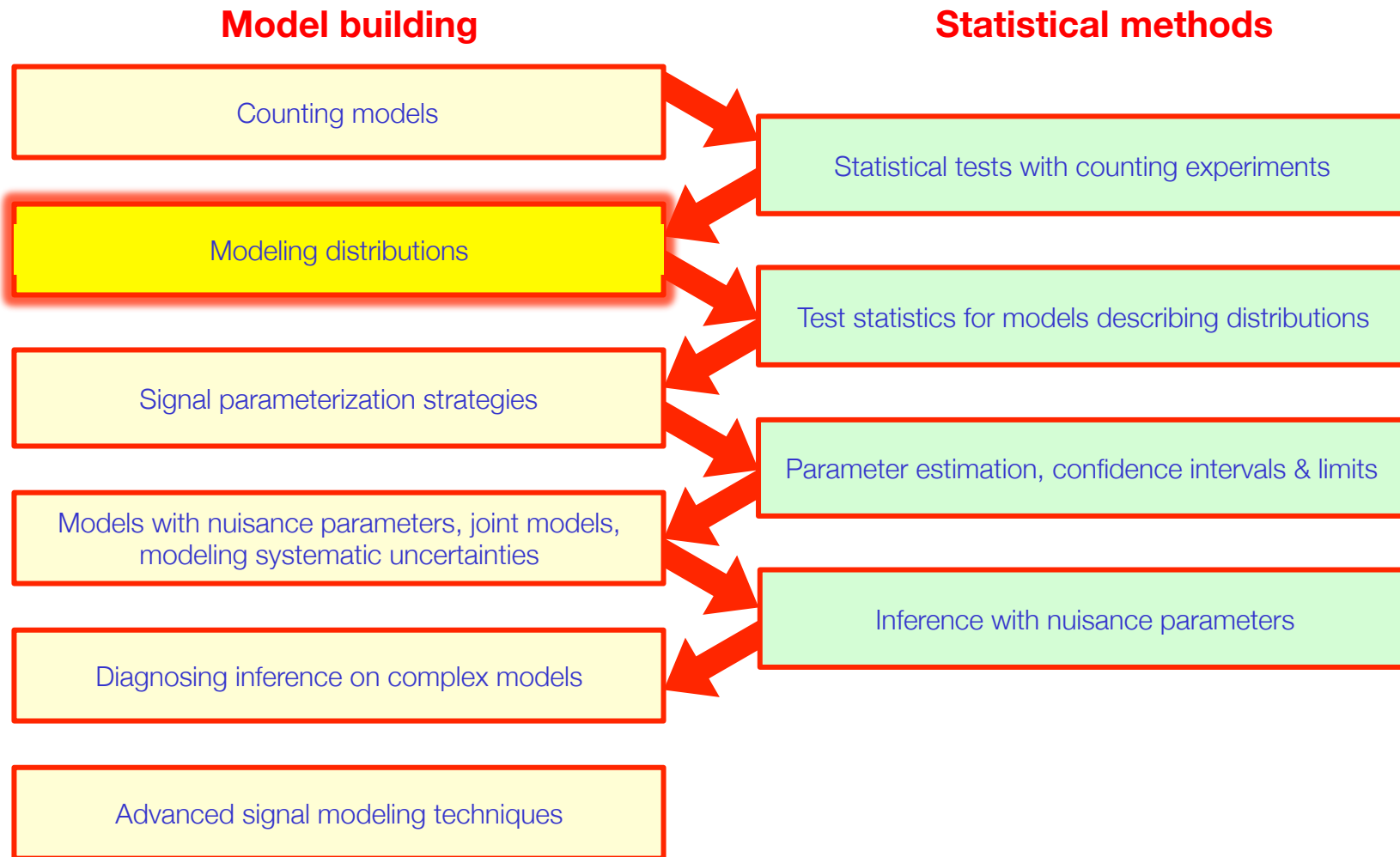
- In principle, *any potentially complex measurement (for Higgs, SUSY, top quarks) can ultimately take this a simple form.*
But there is some 'pre-work' to get here – examining (multivariate) discriminating distributions → Now try to incorporate that

Model building 2

Modelling distributions –
template based models or
analytical models

Roadmap of this course

- Start with basics, gradually build up to complexity

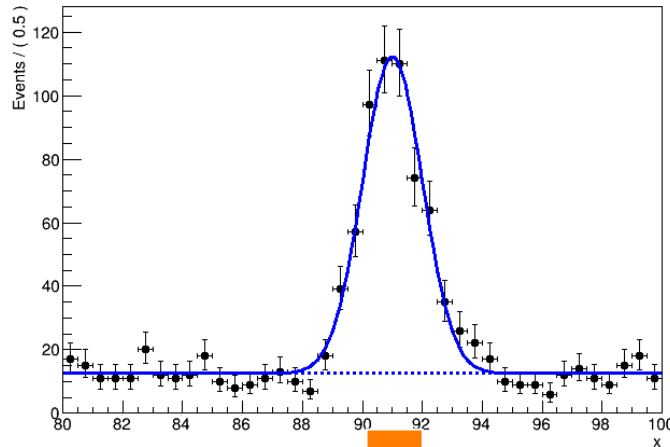


Discriminating observables & counting experiments

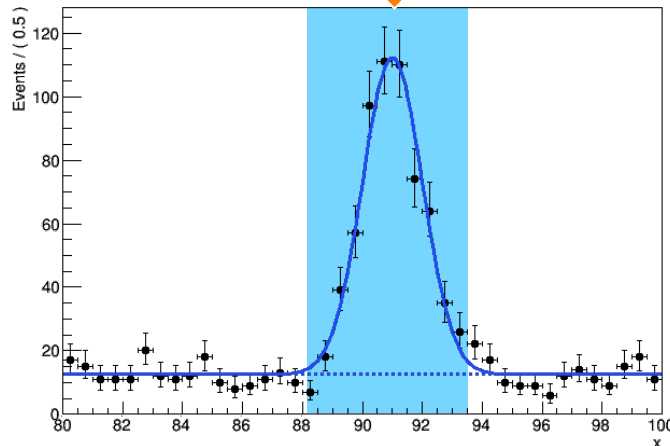
- HEP experimental data usually has many discriminating observables that carry information that can distinguish signal from background hypothesis
- In principle can use them all directly in an elaborate hypothesis test.
 - But would need to formulate a model that describe the expected distribution of all of these → Complicated
 - If expectations are uncertain (from simulation or theory) process of modeling becomes even more complex
- A pragmatic solution to reduce complexity is to split task in two
 - Define empirical selection of events enriched in signal using one or more observable properties of the event (invariant masses, distributions, angles etc)
 - Perform statistical test (hypothesis test, parameter estimation etc) on sample that reduced in size and in dimensionality of discriminating observables that are modeled
 - Most extreme reduction of dimensionality is to zero → counting experiment

Discriminating observables & counting experiments

- Example 1 – **Discrimination in selection stage only**



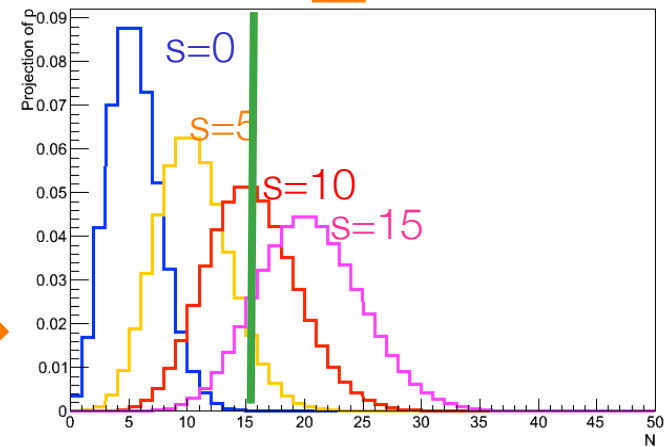
Event selection:
reduce sample size
and dimensionality



NB1: All discriminating power in selection step,
none in inference step. *This is a design choice!*

NB2: Selection must be tuned on a 'figure of merit'
usually a simplified statistical inference test

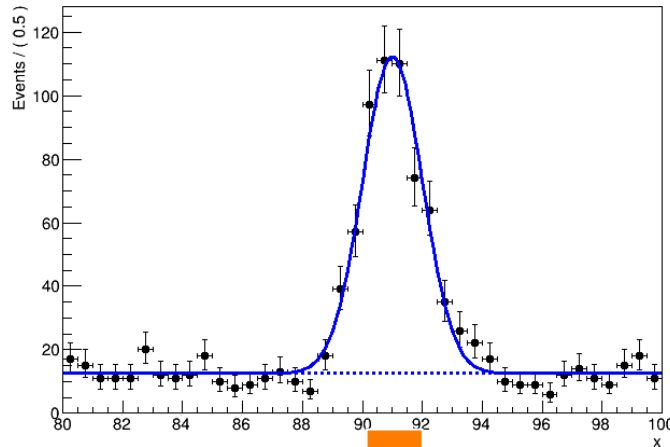
Statistical inference:
 $L(15|5) = 1.5 \cdot 10^{-4}$



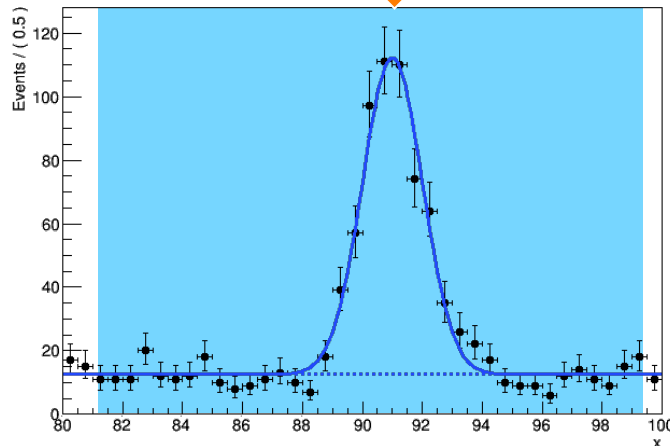
Formulation of probability model of reduced sample:
 $Poisson(N|s+b)$

Modeling discriminating observables

- Example 2 – **Discrimination in inference stage**



*Event selection:
reduce sample size
and dimensionality*

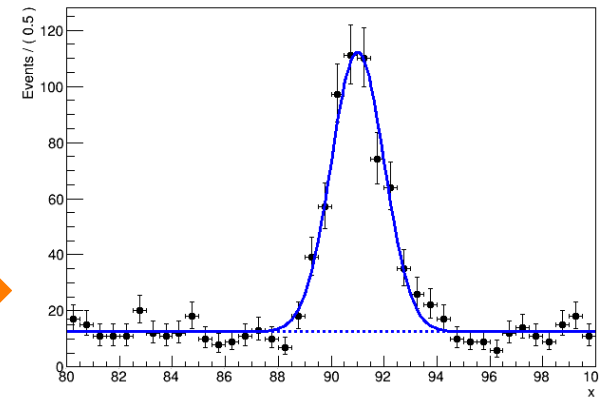


NB1: Most discrimination power in inference step.
This is again design choice!

NB2: Optimal selection less critical

NB3: Correct description of selected sample
more complex

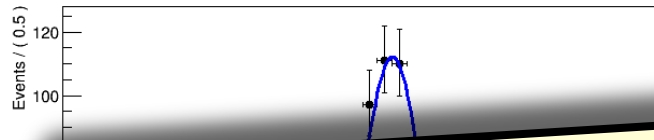
Statistical inference:
 $L(\text{data}|\text{hypo})=\text{something}$



*Formulation of probability model of reduced sample:
 $N_{bkg} * \text{Uniform}(x) + N_{sig} * \text{Gaussian}(x)$*

Modeling discriminating observables

- Example 2 – full dataset has one discriminating observable: x



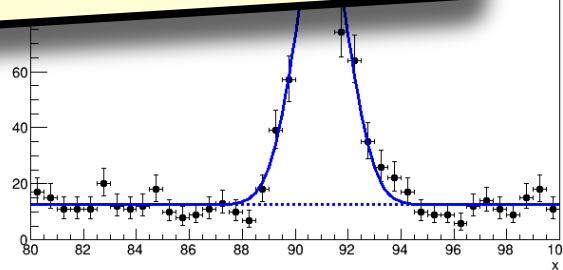
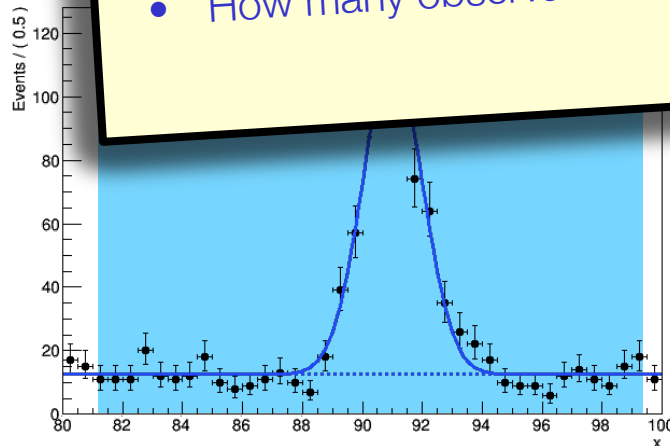
NB1: Most discrimination power in inference step.
This is again design choice!

Q: Which strategy is better?
A: Depends on how 'better' is defined?

For hypothesis testing 'discovery of a new article'
the 'power' of the test can be the same, but doesn't need to be

Choice is real life largely dictated by practicalities

- How easy is it to formulate a description of the observables?
- How many observables are important?



Formulation of probability model of reduced sample:
 $Nbkg * Uniform(x) + Nsig * Gaussian(x)$

Formulating probability models for discriminating observables

- For **counting experiments could derive Poisson($N|\mu$)** from first principles ('random discrete events measured in fixed time interval)
- For **experiments with discriminating observables**, description should ideally **also derive from underlying (physics) hypothesis/theory**
 - In many cases this is possible, but not always without assumptions.
 - Assumptions lead to uncertainties in predictions → we'll revisit later how to deal with those.
- Example: common underlying principle in (signal) model is that discriminating observable is sum/average of many components
 - E.g. light collected by photomultiplier has contributions from $\gg 1$ photons
 - Tracks reconstructed in detector have contributions $\gg 1$ hits
 - **Central Limit Theorem: for large N → Can be analytically described by Gaussian**
- In case there is no easy analytical solution → empirical models (polynomial) or numerical solution (simulation-based histogram)

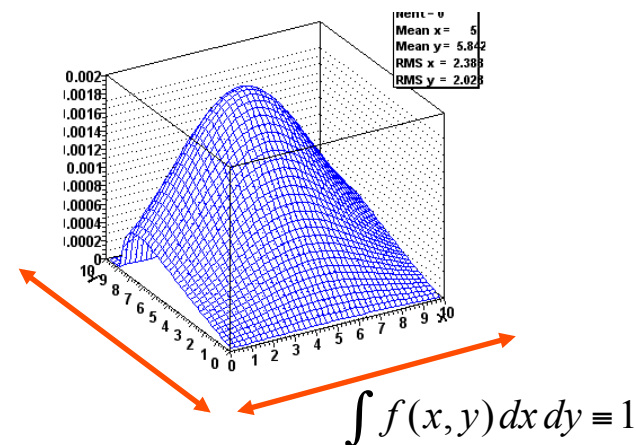
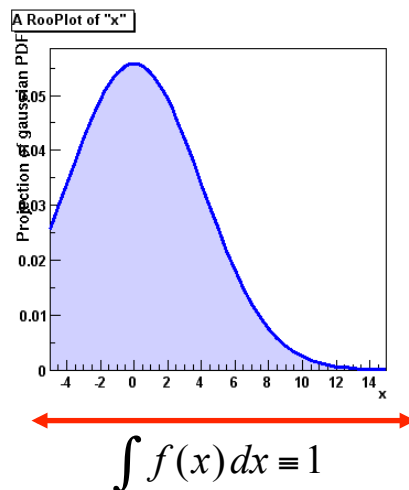
Mathematical formulation of models for observables

- Mathematical description for counting expt is probability model

$$P(N) \geq 0 \quad \forall N$$
$$\sum_{N=0}^{\infty} P(N) \equiv 1$$

- Mathematical description for distribution of discriminating observable is a probability density model:

$$f(\vec{x}) \geq 0 \quad \forall \vec{x}$$
$$\int f(\vec{x}) d\vec{x} \equiv 1$$



Mathematical formulation of models for observables

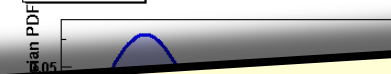
- Mathematical description for **counting** expt is **probability model**

$$P(N) \geq 0 \quad \forall N \quad \sum_{N=0}^{\infty} P(N) \equiv 1$$

- Mathematical description for distribution of **discriminating observable** is a **probability density model**:

$$f(\vec{x}) \geq 0 \quad \forall \vec{x} \quad \int f(\vec{x}) d\vec{x} \equiv 1$$

A RooPlot of "x"



Note that $f(x)$ itself is **not** a probability, but a probability density.

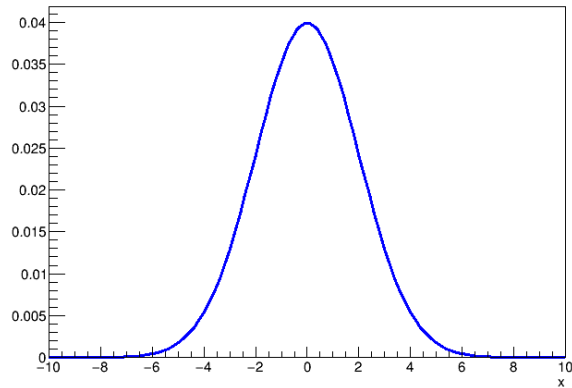
However any integral $\int_a^b f(x) dx$ **is** a probability (for x to be in $[a,b]$)

$$\int f(x) dx \equiv 1$$

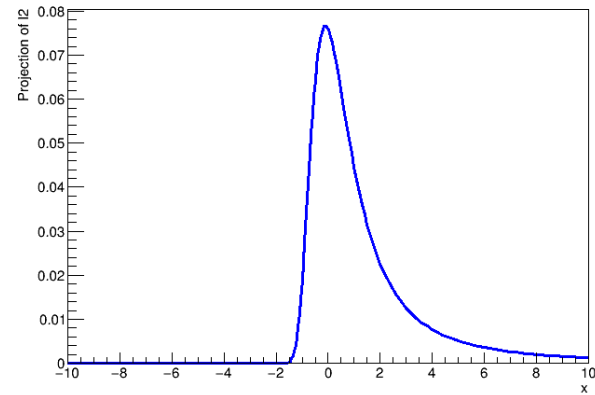
$$\int f(x, y) dx dy \equiv 1$$

Some examples of physics-inspired probability density models

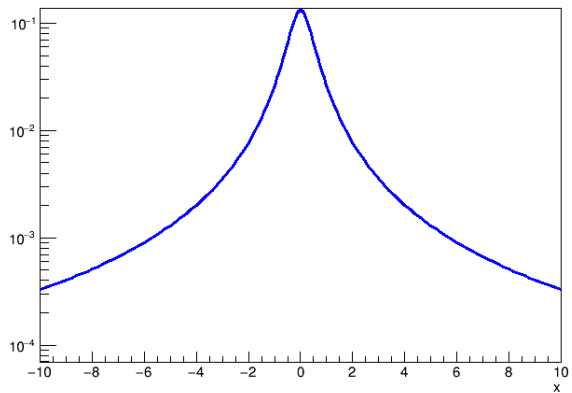
Gaussian
(anything in CLT regime)



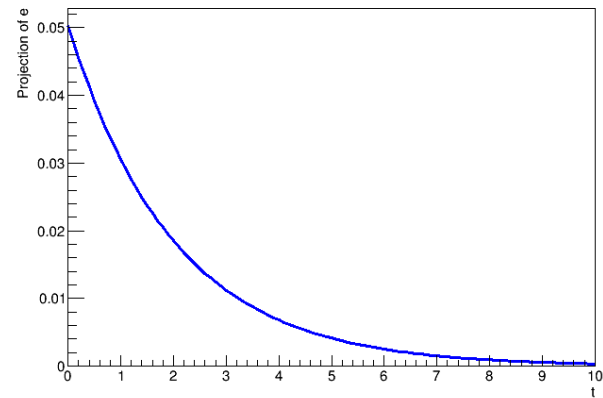
Landau
(energy loss in matter)



Breit-Wigner
(resonant mass)

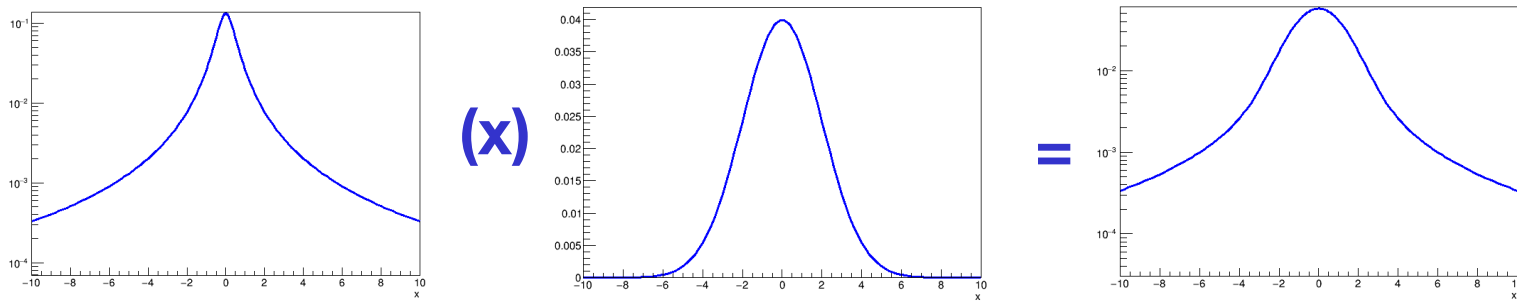


Exponential
(decay time)

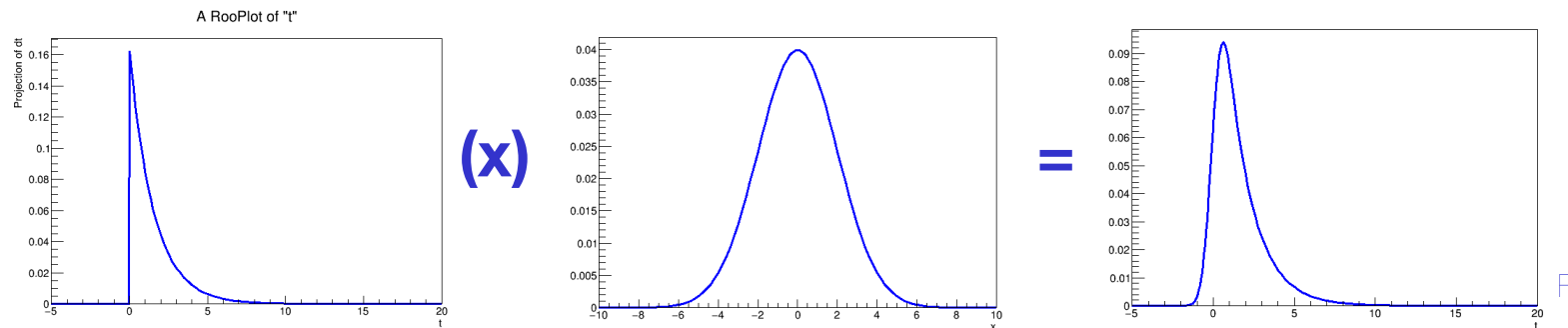


Signal models are often convolutions!

- Observable distributions are often well described by convolutions of physics distributions with (experimental) resolution functions.
 - Often can be calculated analytically, otherwise numerically use FFT
- Example 1: Resonance mass (x) detector resolution



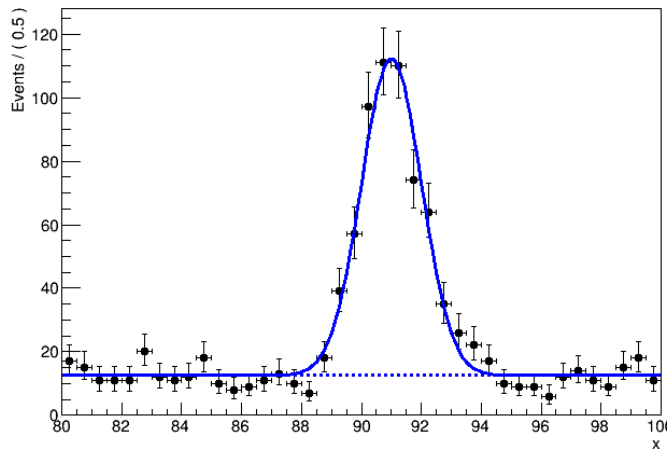
- Example 2: Decay life time (x) detector resolution



PDFs with multiple process contributions

- Analogous to the counting model Poisson(N|S+B), probability density models can describe the distribution of such hypothesis through simple addition

$$f(x) = f_{\text{sig}} \text{Gaussian}(x) + (1 - f_{\text{sig}}) \text{Uniform}(x)$$



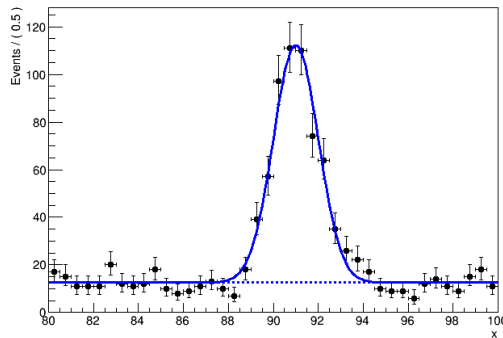
↑
If Gaussian(x) and Uniform(x) are pdfs, then their sum is also a pdf, provided the sum of the coefficients is also 1

- Given a data sample $D(x)$ of N *independent identically distributed* observations of x , the Likelihood is

$$L(\vec{x}) = \prod_{i=0 \dots N} f(x_i)$$

PDFs with multiple process contributions

- Note that the Likelihood $L(x)$ of a probability density function $f(x)$ for a data sample $D(x)$ with N entries **only exploits the differential distribution in x , but not the event count N of the data**
- In many cases the event count can also distinguish the S/B hypothesis (more events expected if signal is present). If so, **the probability model for the event count can be explicitly included in the Likelihood (often called 'extended likelihood')**



$$f(x) = f_{\text{sig}} \text{Gaussian}(x) + (1-f_{\text{sig}}) \text{Uniform}(x)$$

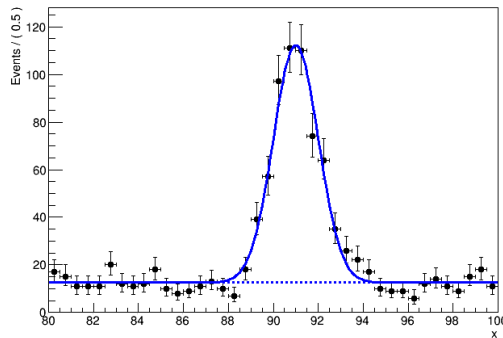
$$P(N) = \text{Poisson}(N | N_{\text{exp}})$$

$$L(\vec{x}, N) = \prod_{i=0 \dots N} f(x_i | f_{\text{sig}}) \cdot \text{Poisson}(N | N_{\text{exp}})$$

- In the common case of a signal and background, with a respective expected event S and B , one can reparameterize $(f_{\text{sig}}, N_{\text{exp}}) \rightarrow (S, B)$

PDFs with multiple process contributions

- Note that the Likelihood $L(x)$ of a probability density function $f(x)$ for a data sample $D(x)$ with N entries **only exploits the differential distribution in x , but not the event count N of the data**
- In many cases the event count can also distinguish the S/B hypothesis (more events expected if signal is present). If so, the probability model for the event count can be explicitly included in the Likelihood (often called 'extended likelihood')



$$f(x) = \frac{S}{S+B} \text{Gaussian}(x) + \frac{B}{S+B} \text{Uniform}(x)$$

$$P(N) = \text{Poisson}(N \mid S+B)$$

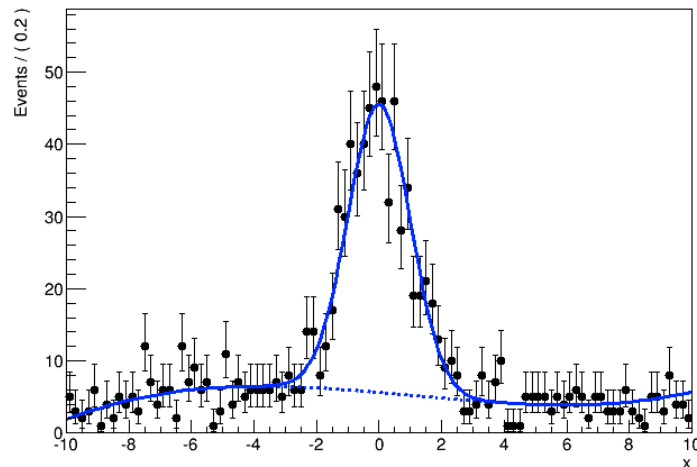
$$L(\vec{x}, N) = \prod_{i=0 \dots N} f(x_i \mid S, B) \cdot \text{Poisson}(N \mid S + B)$$

- In the common case of a signal and background, with a respective expected event S and B , one can reparameterize $(f_{\text{sig}}, N_{\text{exp}}) \rightarrow (S, B)$

Empirical probability models

- In case no description from first principles exists for a differential distribution, empirical or simulation-based models can be deployed

Empirical models

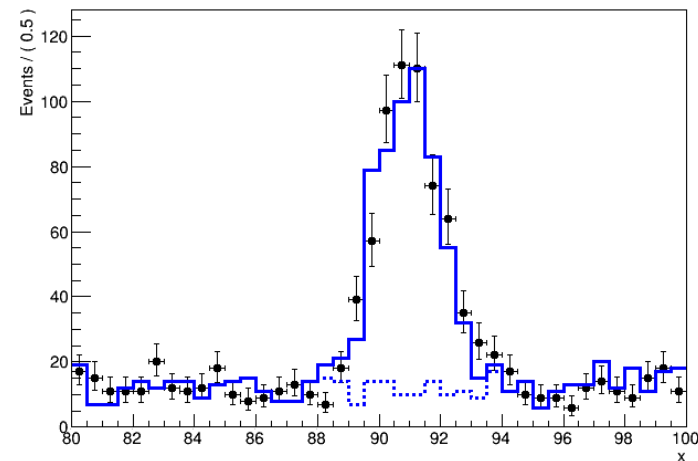


$$B(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \dots$$

Drawbacks:

- **Arbitrariness in parameterization**, e.g. which order to choose for a polynomial

Simulation-based models



$$B(x) = \text{histogram}$$

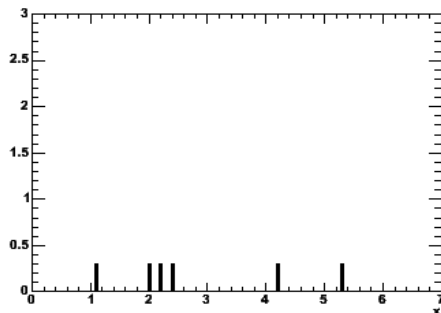
Drawbacks:

- **Quantization** of model prediction in bins
- Poor modeling in regions with **low simulation statistics**

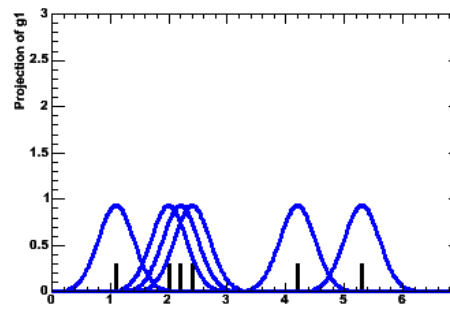
Modeling low-statistics simulation predictions

- For low-statistics simulation predictions, **kernel estimation techniques** can improve modeling substantially
- Procedure:
 - Assign a **Gaussian probability** density distribution to each simulated event.
 - **Sum** Gaussian probability **densities** of all events
 - Started from unbinned data → no binning effects

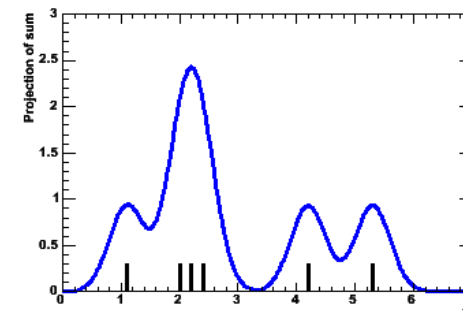
Sample of events



Gaussian probability distributions for each event



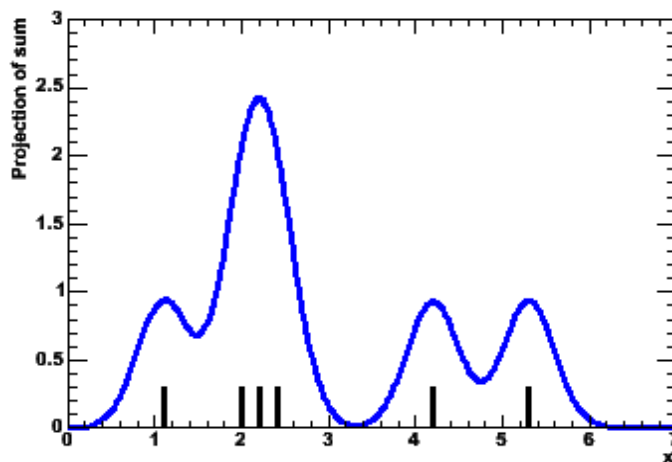
Summed probability distribution for all events in sample



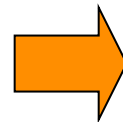
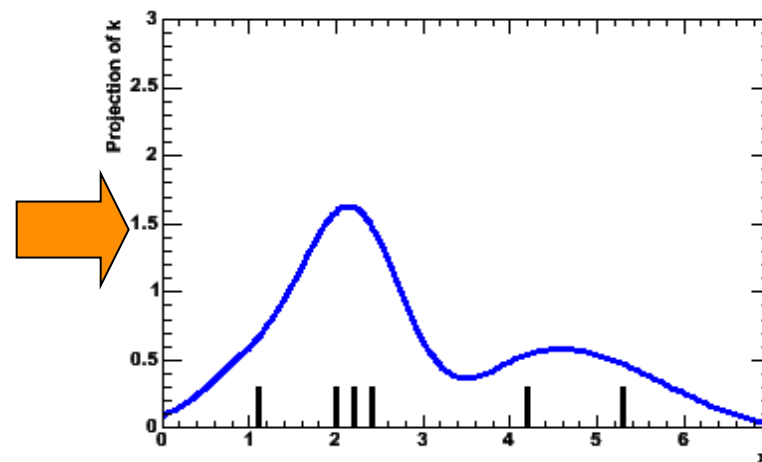
Modeling low-statistics simulation predictions

- Technique does *not* require that all Gaussian kernels have same width
- Improved procedure: 'adaptive kernel'
 - Adjust width of Gaussian kernels depending on local event density
 - High density \rightarrow narrow kernels \rightarrow preserve more detail
 - Low density \rightarrow wide kernels \rightarrow promote smoothness

Static Kernel
(with of all Gaussian identical)



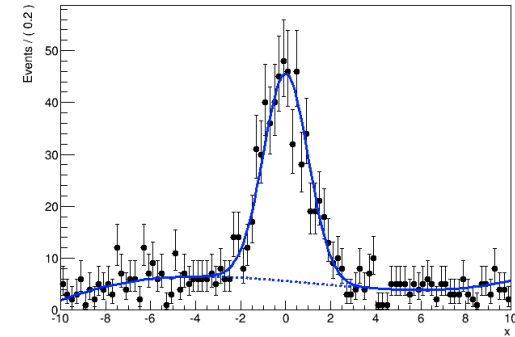
Adaptive Kernel
(width of all Gaussian depends
on local density of events)



Binned vs unbinned likelihoods

- Analytical probability density functions describe data vectors x = unbinned ‘raw’ distribution of x
 - Constructs statistical tests with the highest power, in particular at low event counts

$$L(\vec{x}, N) = \prod_{i=0 \dots N} f(x_i | f_{sig}) \cdot \text{Poisson}(N | N_{exp})$$



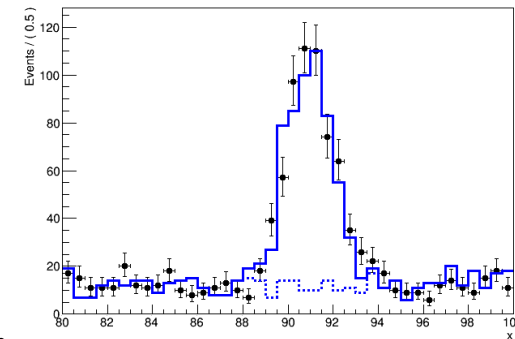
- In the limit of large N unbinned likelihoods become very CPU consuming with diminishing returns
 - Can approximate unbinned likelihood with a binned likelihood (calculation time will scale with $N(\text{bin})$ rather $N(\text{data})$)

$$L(\vec{n}) = \prod_{i=0 \dots N} \text{Poisson}(n_i | \mu_i)$$

$$\mu_i = \int_{x_i^{low}}^{x_i^{high}} f(x) dx \cdot N_{exp}$$

$$\approx f(x_i^{mid})(x_i^{high} - x_i^{low}) \cdot N_{exp}$$

(Exact for binned models)



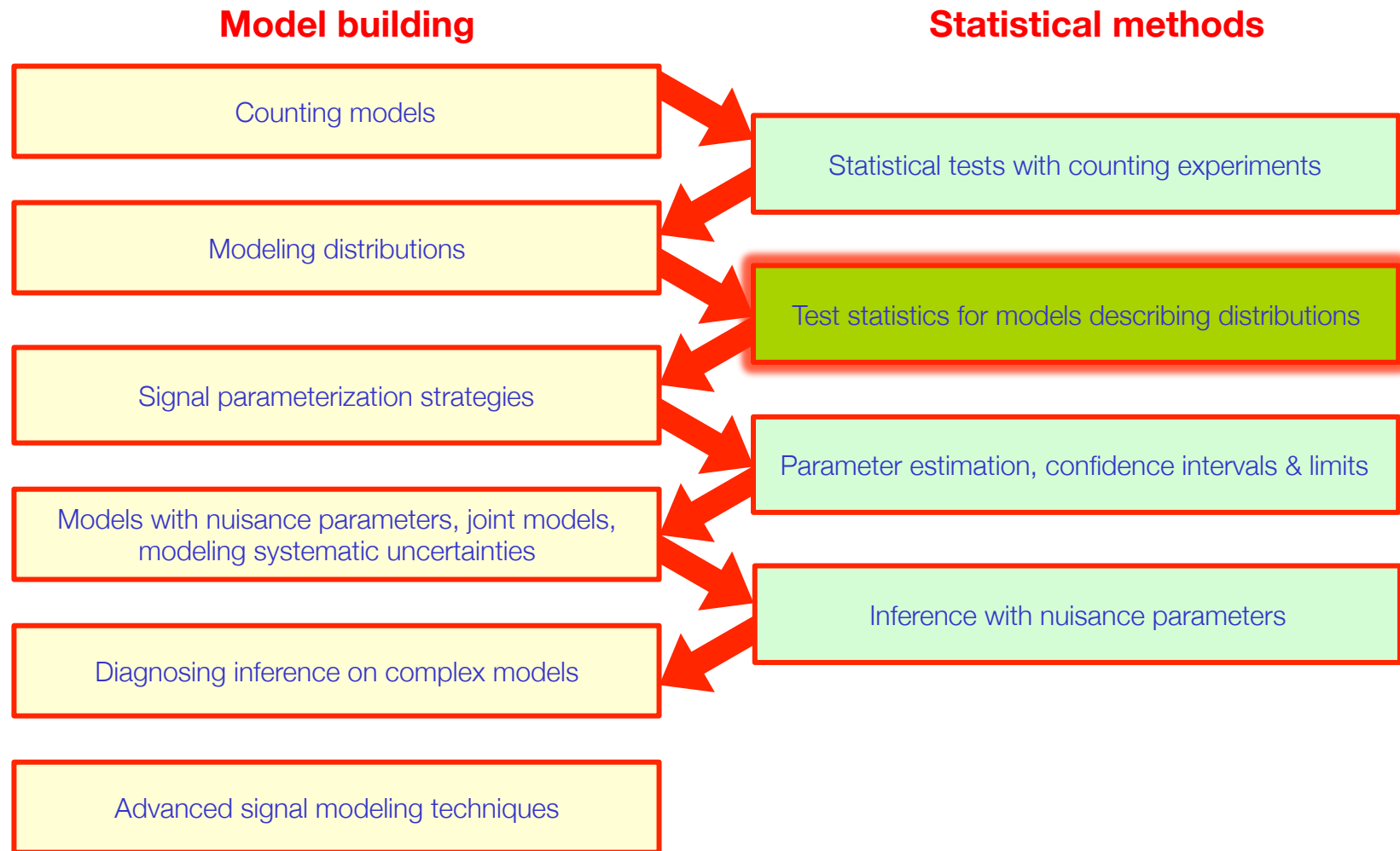
Wouter Verkerke, NIKHEF

Statistical methods 2

Adapting statistical methods to use with distributions:
test statistics as ordering principle, likelihood ratios,
contrast with Bayesian methods, the likelihood principle.
Practical aspects of toy MC sampling

Roadmap of this course

- Start with basics, gradually build up to complexity



Working with Likelihood functions for distributions

- **How do the statistical inference procedures change** for Likelihoods describing distributions?
- Bayesian calculation of $P(\text{theo}|\text{data})$ they are *exactly the same*.
 - Simply substitute counting model with binned distribution model

$$P(H_{s+b} | \vec{N}) = \frac{L(\vec{N} | H_{s+b})P(H_{s+b})}{L(\vec{N} | H_{s+b})P(H_{s+b}) + L(\vec{N} | H_b)P(H_b)}$$

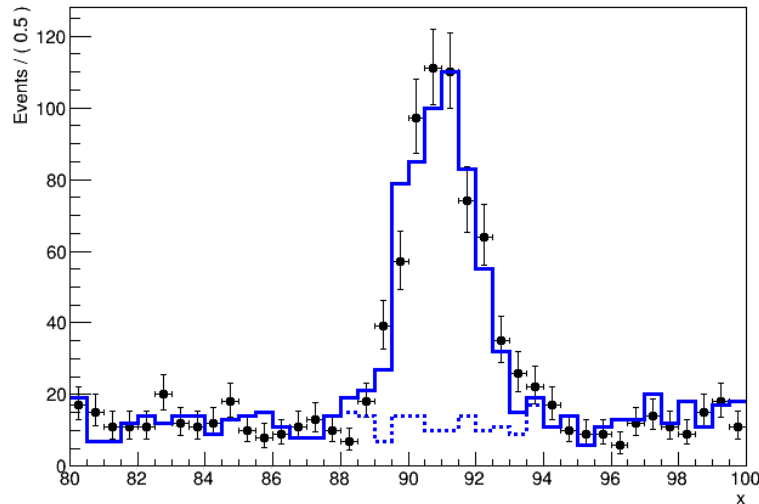


Simply fill in new Likelihood function
Calculation otherwise unchanged

$$P(H_{s+b} | \vec{N}) = \frac{\prod_i \text{Poisson}(N_i | \tilde{s}_i + \tilde{b}_i)P(H_{s+b})}{\prod_i \text{Poisson}(N_i | \tilde{s}_i + \tilde{b}_i)P(H_{s+b}) + \prod_i \text{Poisson}(N_i | \tilde{b}_i)P(H_b)}$$

Working with Likelihood functions for distributions

- Frequentist calculation of $P(\text{data}|\text{hypo})$ also unchanged, but **question arises if $P(\text{data}|\text{hypo})$ is still relevant?**



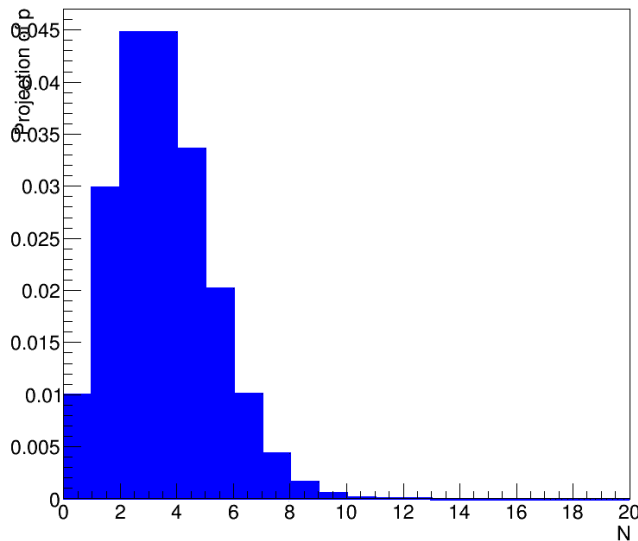
$$L(\vec{N} | H_b) = \prod_i \text{Poisson}(N_i | \tilde{b}_i)$$

$$L(\vec{N} | H_{s+b}) = \prod_i \text{Poisson}(N_i | \tilde{s}_i + \tilde{b}_i)$$

- **$L(N|H)$ is probability to obtain *exactly* the histogram observed.**
- *Is that what we want to know?* Not really.. We are interested in probability to observe any ‘similar’ dataset to given dataset, or in practice dataset ‘similar or more extreme’ that observed data
- **Need a way to quantify ‘similarity’ or ‘extremity’ of observed data**

Working with Likelihood functions for distributions

- *Definition*: a test statistic $T(x)$ is *any* function of the data x
- We need a test statistic that will **classify ('order') all possible observations** in terms of 'extremity' (definition to be chosen by physicist)
- NB: For a counting measurement the count itself is already a useful test statistic for such an ordering (i.e. $T(x) = x$)

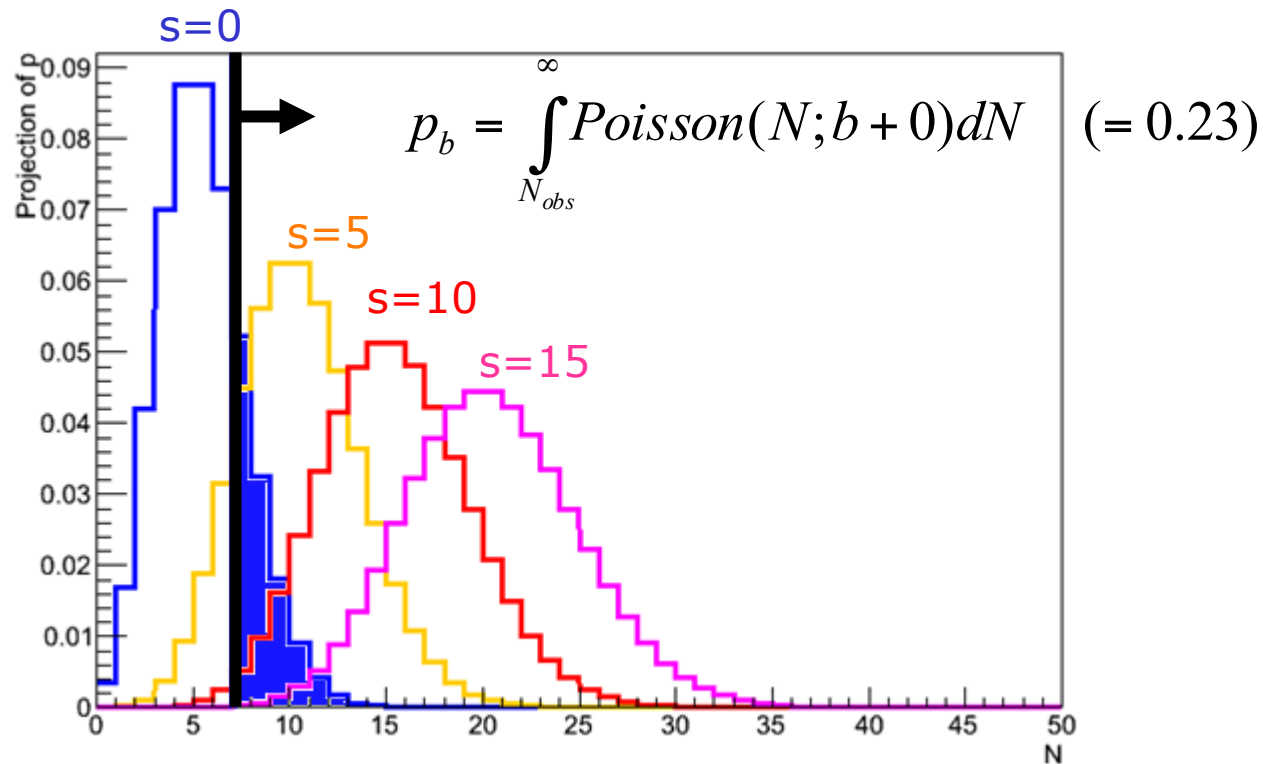


Test statistic $T(N) = N_{\text{obs}}$ orders observed events count by estimated signal yield

Low $N \rightarrow$ low estimated signal
High $N \rightarrow$ large estimated signal

P-values for counting experiments

- Now make a measurement $N=N_{\text{obs}}$ (example $N_{\text{obs}}=7$)
- **Definition: p-value:**
probability to obtain the observed data, or more extreme in future repeated identical experiments
 - Example: p-value for background-only hypothesis



Ordering distributions by ‘signal-likeness’ aka ‘extremity’

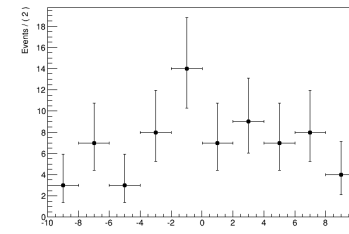
- How to define ‘extremity’ if observed data is a distribution

Observation

Counting

$$N_{\text{obs}}=7$$

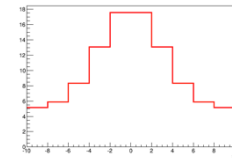
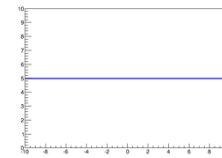
Histogram



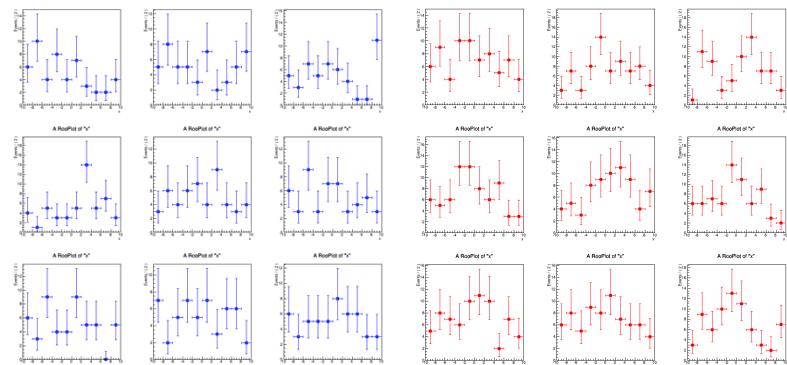
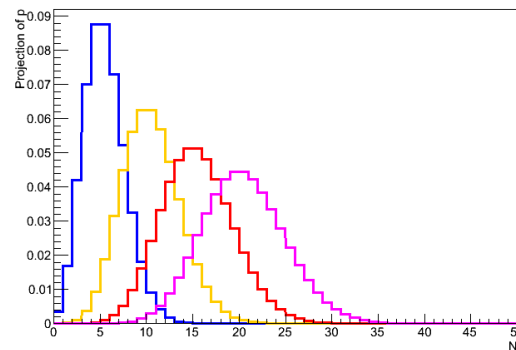
Median expected by hypothesis

$$N_{\text{exp}}(s=0) = 5$$

$$N_{\text{exp}}(s=5) = 10$$



Predicted distribution of observables



Which histogram is more ‘extreme’?

The Likelihood Ratio as a test statistic

- Given two hypothesis H_b and H_{s+b} the ratio of likelihoods is a useful test statistic

$$\lambda(\vec{N}) = \frac{L(\vec{N} | H_{s+b})}{L(\vec{N} | H_b)}$$

- Intuitive picture:

→ If data is likely under H_b ,
 $L(N|H_b)$ is **large**,
 $L(N|H_{s+b})$ is smaller

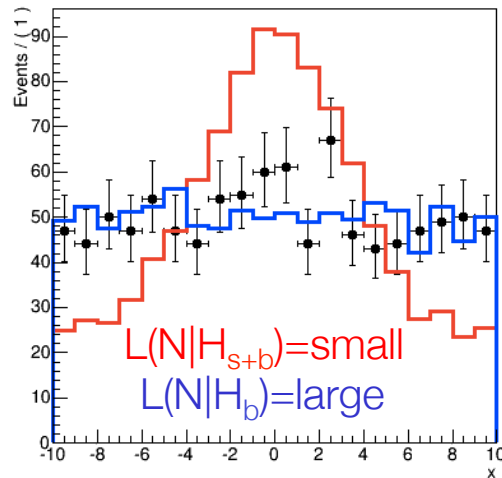
$$\lambda(\vec{N}) = \frac{\text{small}}{\text{large}} = \text{small}$$

→ If data is likely under H_{s+b}
 $L(N|H_{s+b})$ is **large**,
 $L(N|H_b)$ is smaller

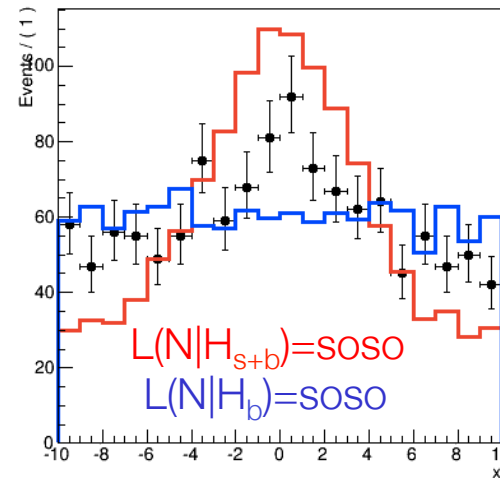
$$\lambda(\vec{N}) = \frac{\text{large}}{\text{small}} = \text{large}$$

Visualizing the Likelihood Ratio as ordering principle

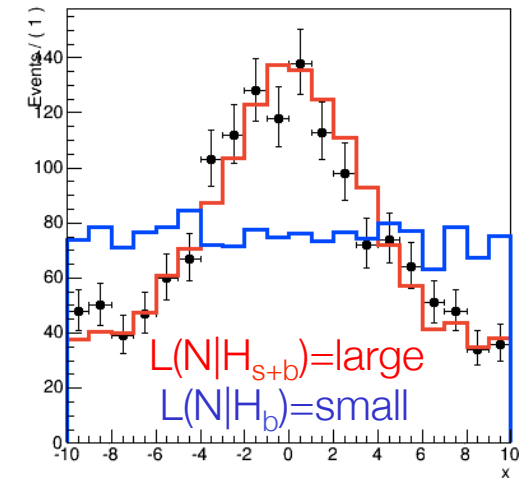
- The Likelihood ratio as ordering principle



$$\lambda(N)=0.0005$$



$$\lambda(N)=0.47$$

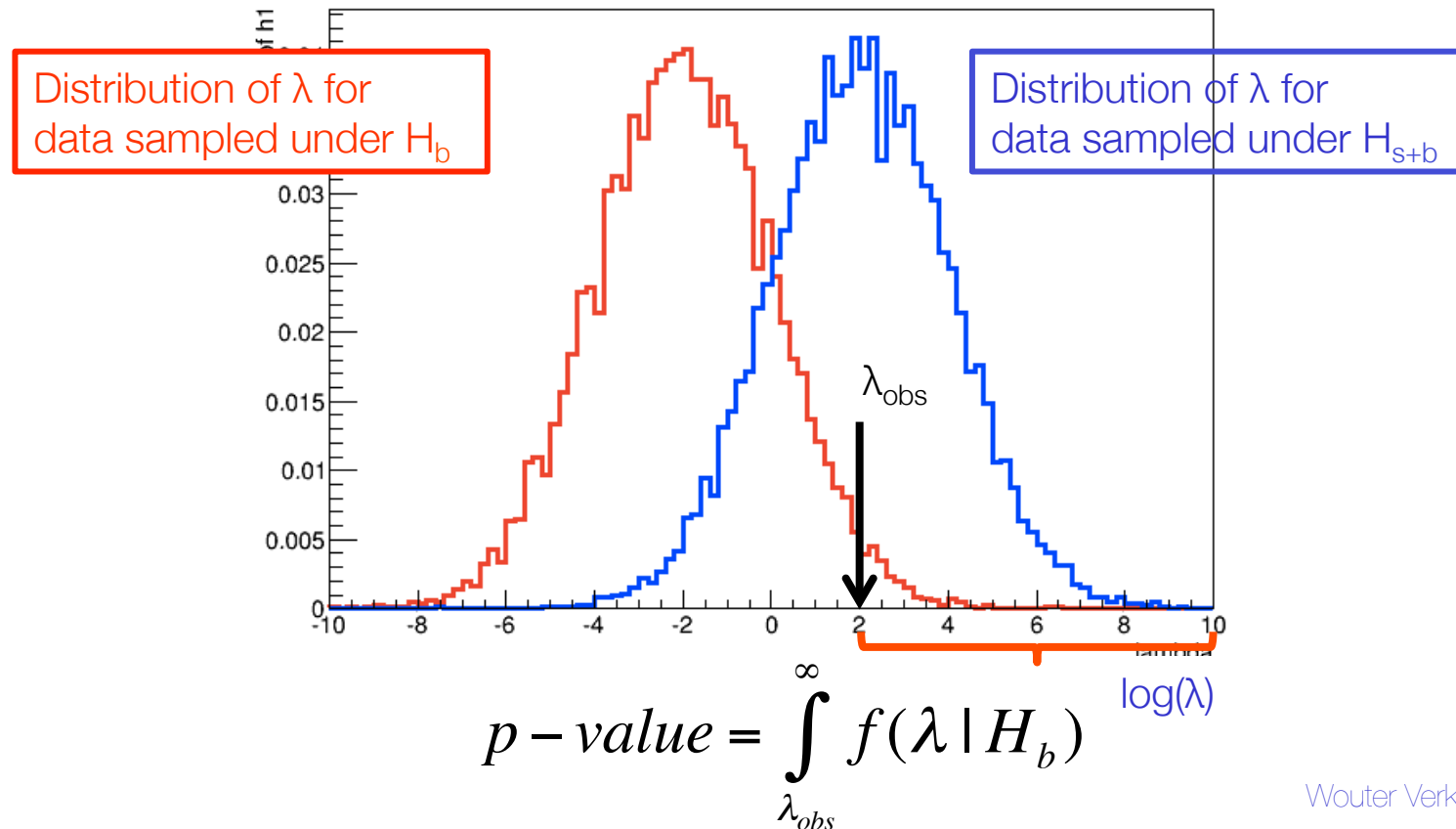


$$\lambda(N)=5000$$

- Frequentist solution to ‘relevance of $P(\text{data}|\text{theory})$ ’ is to order all observed data samples using a (Likelihood Ratio) test statistic
 - Probability to observe ‘similar data or more extreme’ then amounts to calculating ‘probability to observe test statistic $\lambda(N)$ as large or larger than the observed test statistic $\lambda(N_{\text{obs}})$ ’

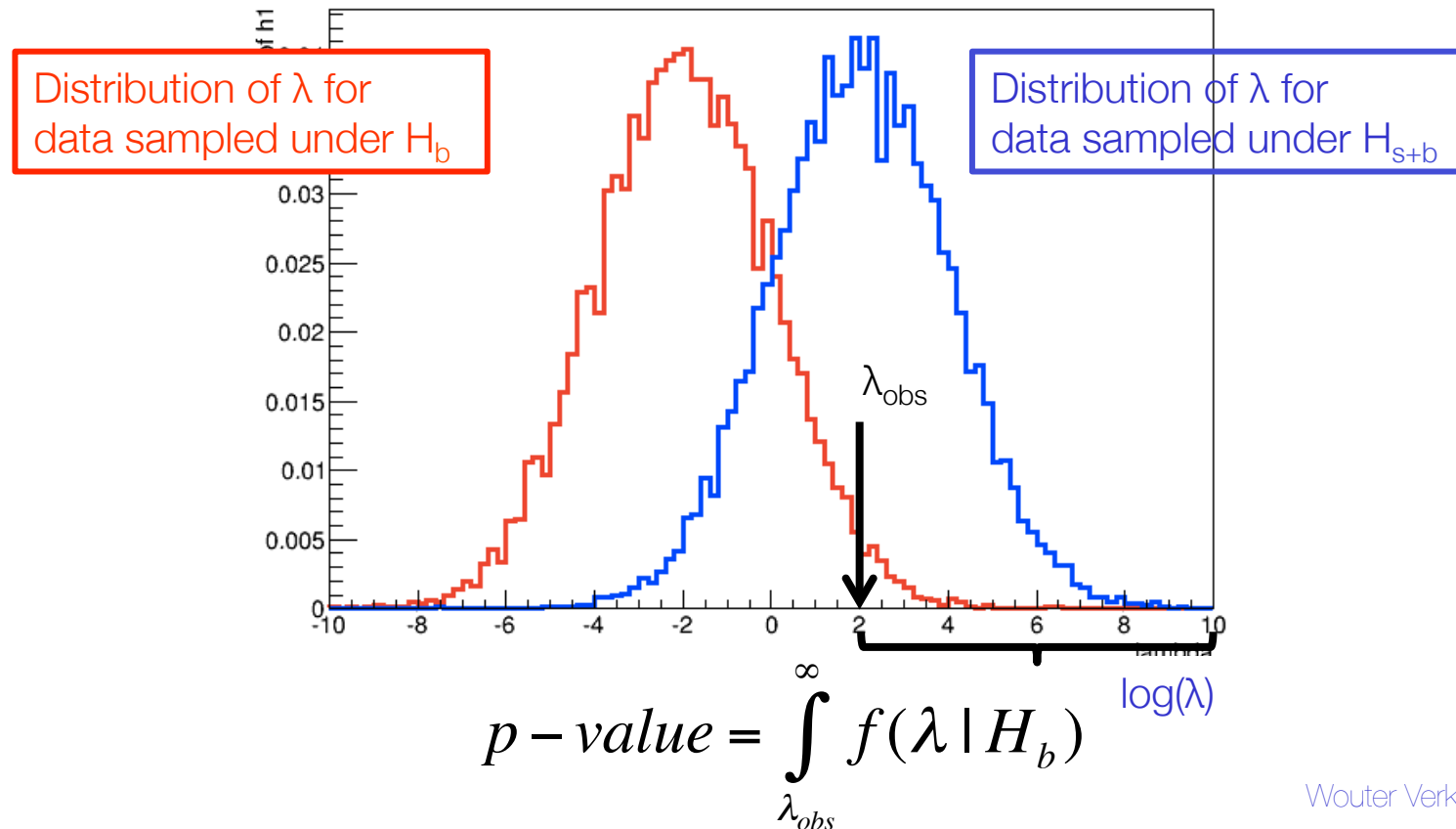
The distribution of the test statistic

- Distribution of a test statistic is *generally not known*
- Use toy MC approach to approximate distribution
 - Generate many toy datasets N under H_b and H_{s+b} and evaluate $\lambda(N)$ for each dataset



The distribution of the test statistic

- **Definition: p-value:**
probability to obtain the observed data, or more extreme
in future repeated identical experiments
(extremity define in the precise sense of the (LR) ordering rule)



Likelihoods for distributions - summary

- **Bayesian inference unchanged**

→ simply insert L of distribution to calculate $P(H|\text{data})$

$$P(H_{s+b} | \vec{N}) = \frac{L(\vec{N} | H_{s+b})P(H_{s+b})}{L(\vec{N} | H_{s+b})P(H_{s+b}) + L(\vec{N} | H_b)P(H_b)}$$

- **Frequentist inference procedure *modified***

→ Pure $P(\text{data}|\text{hypo})$ not useful for non-counting data

→ Order all possible data with a (LR) test statistic in ‘extremity’

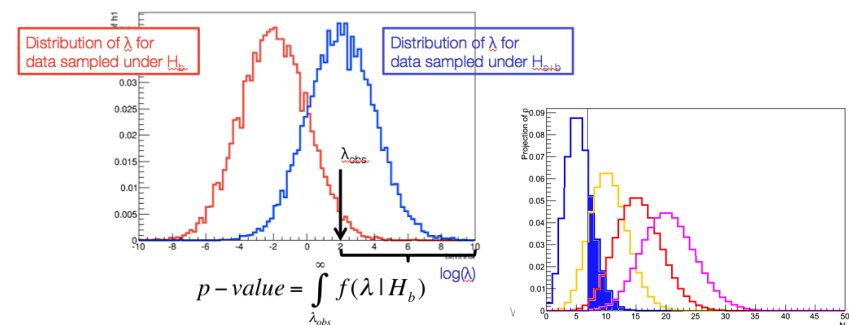
→ Quote $p(\text{data}|\text{hypo})$ as ‘p-value’ for hypothesis

Probability to obtain observed data, *or more extreme*, is X%

‘Probability to obtain 13 or more 4-lepton events under the no-Higgs hypothesis is 10^{-7} ’

‘Probability to obtain 13 or more 4-lepton events under the SM Higgs hypothesis is 50%’

- **Definition: p-value**



The likelihood principle

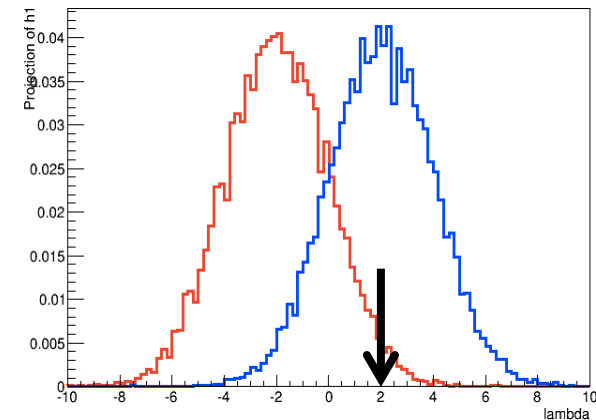
- Note that ‘ordering procedure’ introduced by test statistic also has a profound implication on interpretation
- Bayesian inference only uses the Likelihood of the observed data

$$P(H_{s+b} | \vec{N}) = \frac{L(\vec{N} | H_{s+b})P(H_{s+b})}{L(\vec{N} | H_{s+b})P(H_{s+b}) + L(\vec{N} | H_b)P(H_b)}$$

- While the observed Likelihood Ratio also only uses likelihood of observed data.

$$\lambda(\vec{N}) = \frac{L(\vec{N} | H_{s+b})}{L(\vec{N} | H_b)}$$

- **Distribution $f(\lambda|N)$, and thus p-value, also uses likelihood of non-observed outcomes** (in fact Likelihood of every possible outcome is used)



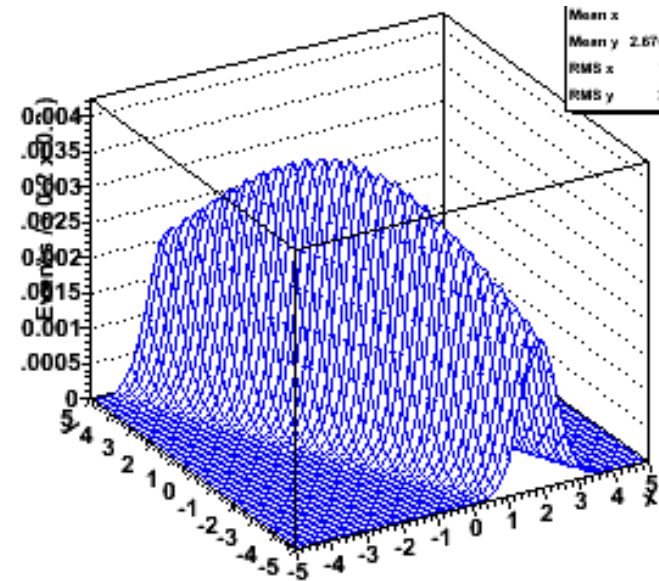
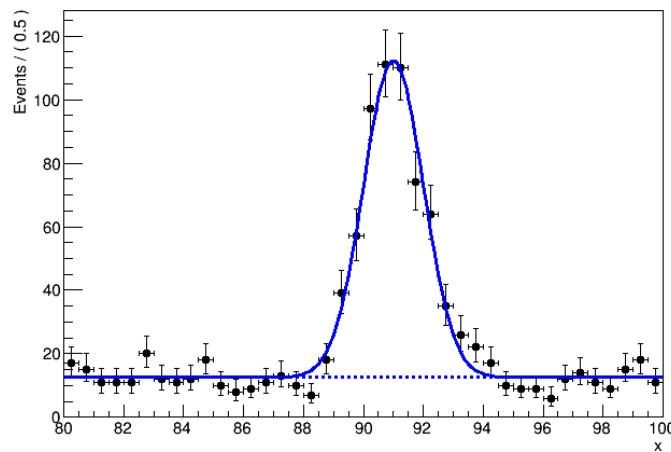
Likelihood Principle

- In **Bayesian** methods and **likelihood-ratio** based methods, the probability (density) for obtaining the *data at hand is used (via the likelihood function)*, *but probabilities for obtaining other data are not used!*
- In contrast, in typical **frequentist** calculations (e.g., a p-value which is the probability of obtaining a value as extreme or *more extreme than that observed*), *one uses probabilities of data not seen.*
- This difference is captured by the *Likelihood Principle**:

If two experiments yield likelihood functions which are proportional, then Your inferences from the two experiments should be identical.

Generalizing to multiple dimensions

- Can also generalize likelihood models to distributions in *multiple* observables



$$L(\vec{x}) = \prod_i f(x_i)$$

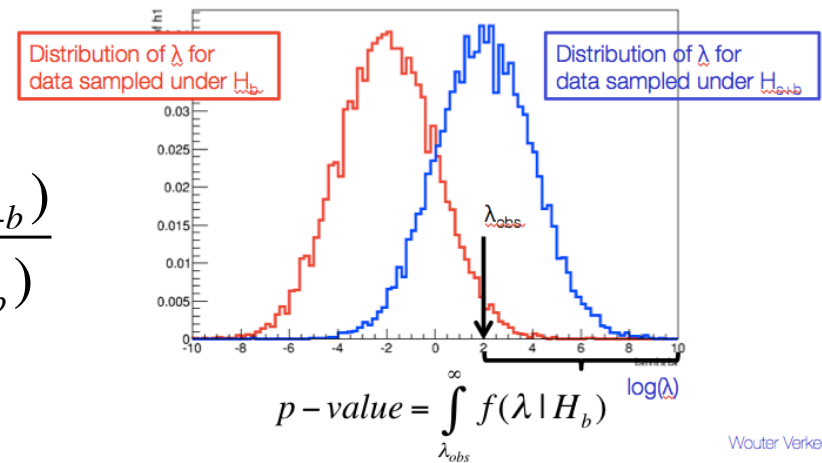
$$L(\vec{x}, \vec{y}) = \prod_i f(x_i, y_i)$$

- Neither generalization (binned \rightarrow continuous, one \rightarrow multiple observables) has any further consequences for Bayesian or Frequentist inference procedures

The Likelihood Ratio test statistic as tool for event selection

- Note that hypothesis testing with two simple hypotheses for observable distributions, exactly describes ‘event selection’ problem
- In fact we have already ‘solved’ the optimal event selection problem! Given two hypothesis H_{s+b} and H_b that predict an complex multivariate distribution of observables, **you can always classify all events in terms of ‘signal-likeness’ (a.k.a ‘extremity’) with a likelihood ratio**

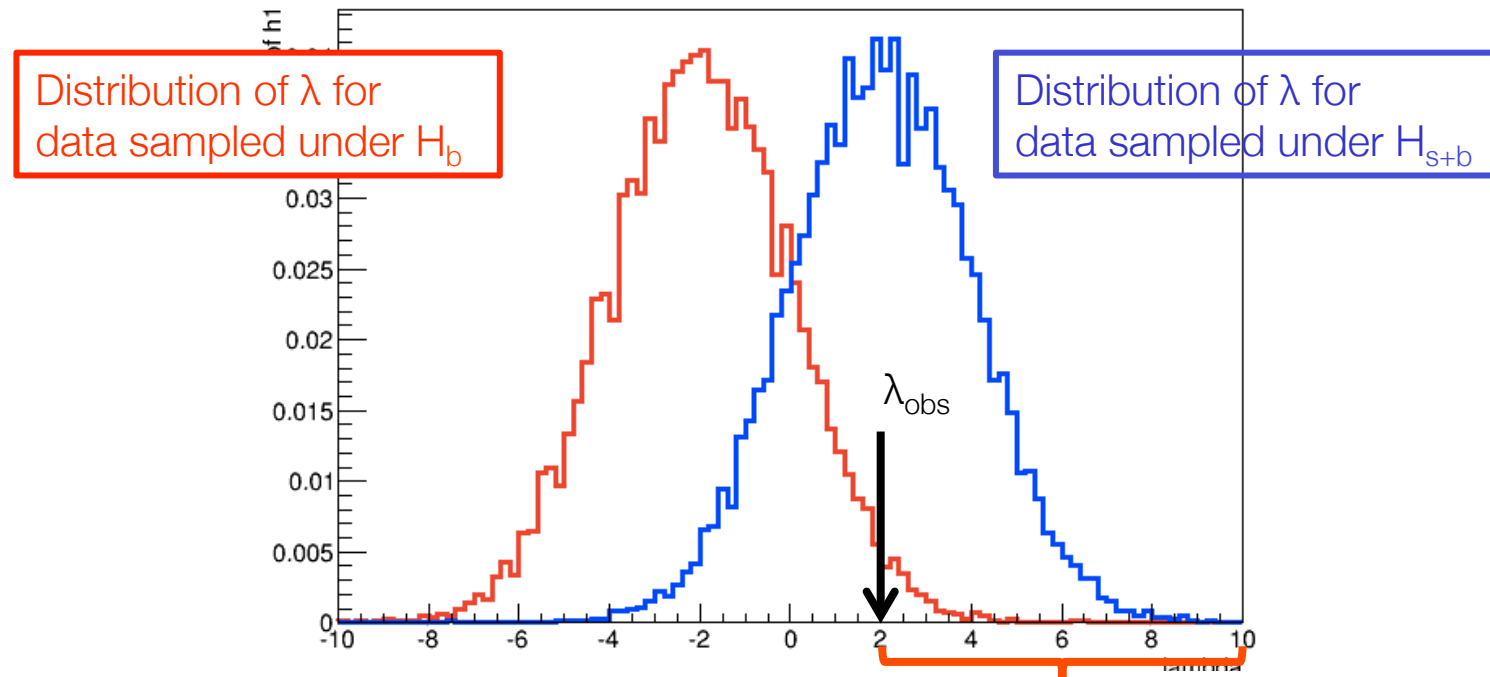
$$\lambda(\vec{x}, \vec{y}, \vec{z}, \dots) = \frac{L(\vec{x}, \vec{y}, \vec{z}, \dots | H_{s+b})}{L(\vec{x}, \vec{y}, \vec{z}, \dots | H_b)}$$



- So far we have exploited λ to calculate a frequentist p-value **now explore properties ‘cut on λ ’ as basis of (optimal) event selection**

The distribution of the test statistic

- Distribution of a test statistic is *generally not known*
- Use toy MC approach to approximate distribution
 - Generate many toy datasets N under H_b and H_{s+b} and evaluate $\lambda(N)$ for each dataset



$$p - value = \int_{\lambda_{obs}}^{\infty} f(\lambda | H_b) \log(\lambda)$$

Intermezzo – Generating toy data

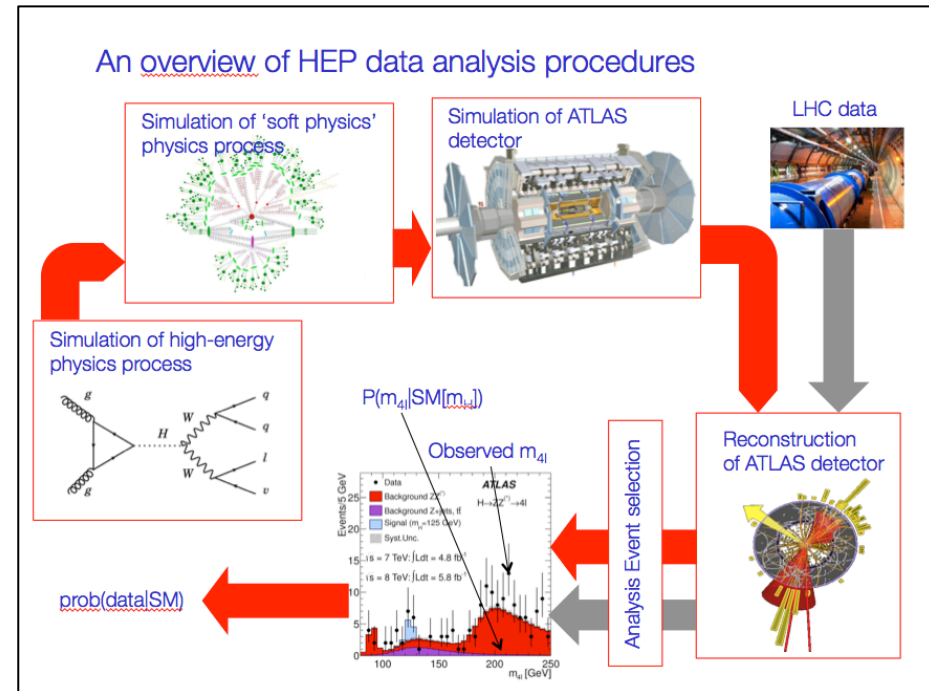
- Two approaches to obtaining simulated data

- First approach is ‘Physics Monte Carlo Chain’, described earlier

- Time consuming, but injects detailed knowledge about physics, detector, output is full collision information, and relation to underlying theory details

- Alternative approach is sample sampling the probability model ‘toy MC’

- Fast (generally), only requires access to probability model
- Can only produce datasets with observables that are described by the probability model → Sufficient to study distribution of test statistics

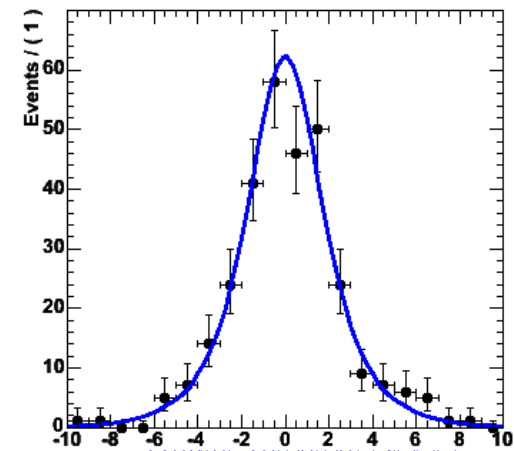
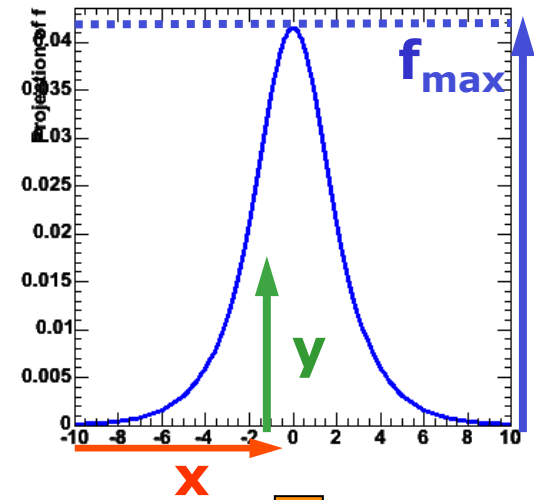


How do you efficiently generate a toy dataset from a probability model?

- Simplest method is accept/reject sampling

- 1) Determine maximum of function f_{\max}
- 2) Throw random number x
- 3) Throw another random number y
- 4) If $y < f(x)/f_{\max}$ keep x ,
otherwise return to step 2)

- PRO: Easy, always works
- CON: It can be inefficient if function is strongly peaked.
Finding maximum empirically through random sampling can be lengthy in >2 dimensions

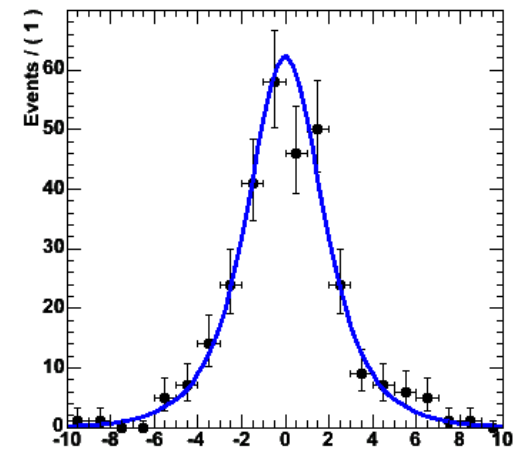
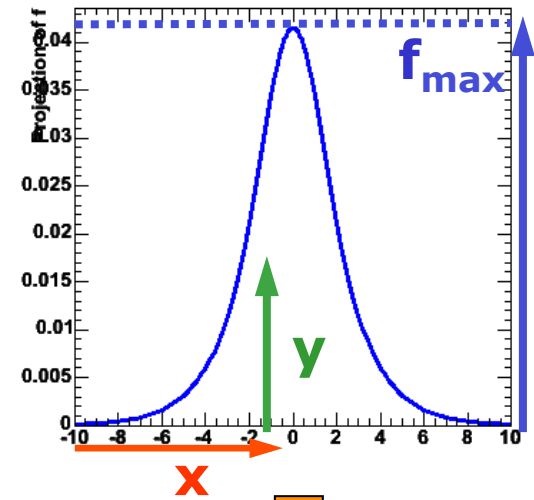


How do you efficiently generate a toy dataset from a probability model?

- Simplest method is accept/reject sampling

- 1) Determine maximum of function f_{\max}
- 2) Throw random number x
- 3) Throw another random number y
- 4) If $y < f(x)/f_{\max}$ keep x ,
otherwise return to step 2)

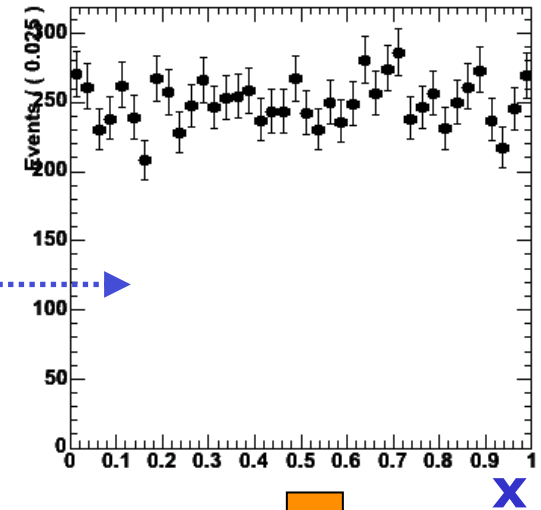
- PRO: Easy, always works
- CON: It can be inefficient if function is strongly peaked.
Finding maximum empirically through random sampling can be lengthy in >2 dimensions



Toy MC generation – Inversion method

- Fastest: function inversion

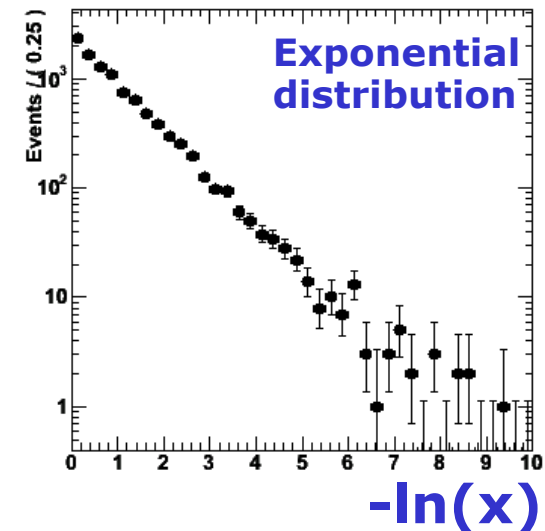
- 1) Given $f(x)$ find inverted function $F(x)$ so that $f(F(x)) = x$
- 2) Throw uniform random number x
- 3) Return $F(x)$



Take $-\log(x)$



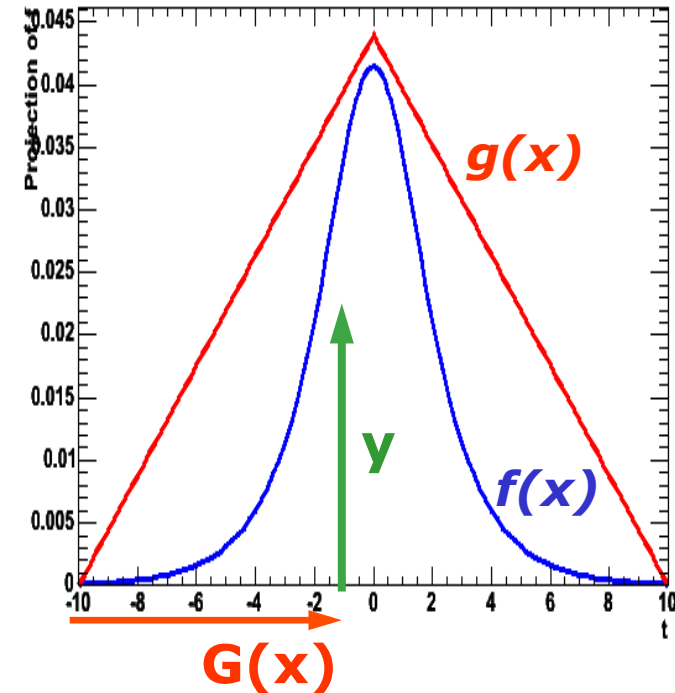
- PRO: Maximally efficient
- CON: Only works for invertible functions



Toy MC Generation – importance sampling

- Hybrid: Importance sampling

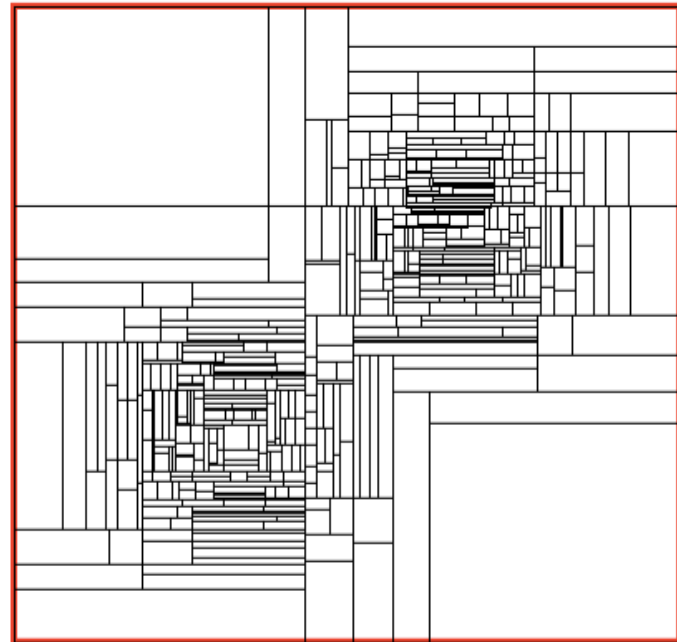
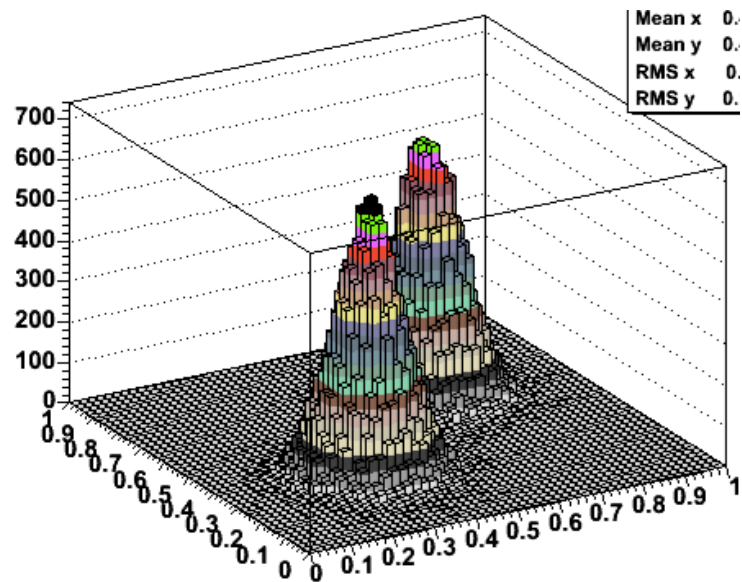
- 1) Find 'envelope function' $g(x)$ that is invertible into $G(x)$ and that fulfills $g(x) \geq f(x)$ for all x
- 2) Generate random number x from G using inversion method
- 3) Throw random number 'y'
- 4) If $y < f(x)/g(x)$ keep x , otherwise return to step 2



- PRO: Faster than plain accept/reject sampling
Function does not need to be invertible
- CON: Must be able to find invertible envelope function

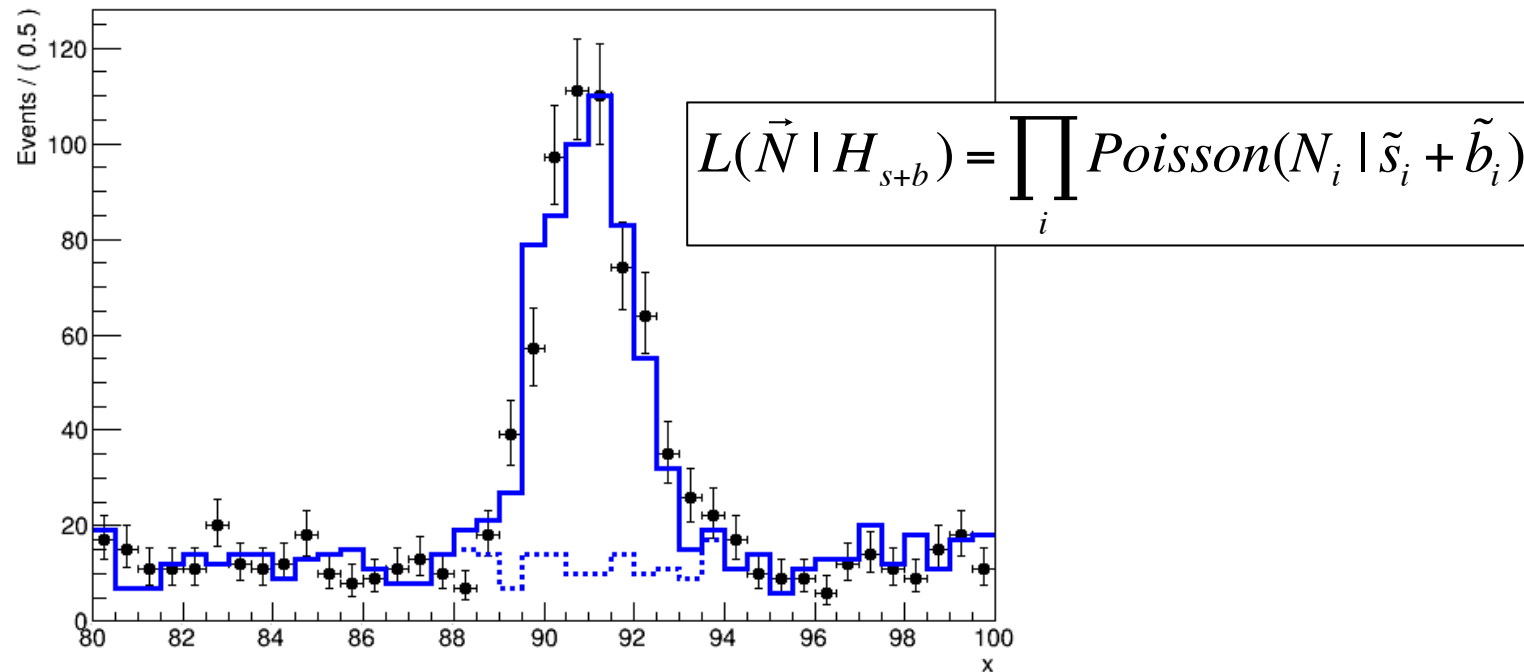
Toy MC Generation – importance sampling in $>1D$

- General algorithms exist that can construct empirical envelope function
 - Divide observable space recursively into smaller boxes and take uniform distribution in each box
 - Example shown below from FOAM algorithm



Toy MC Generation – importance sampling in >1D

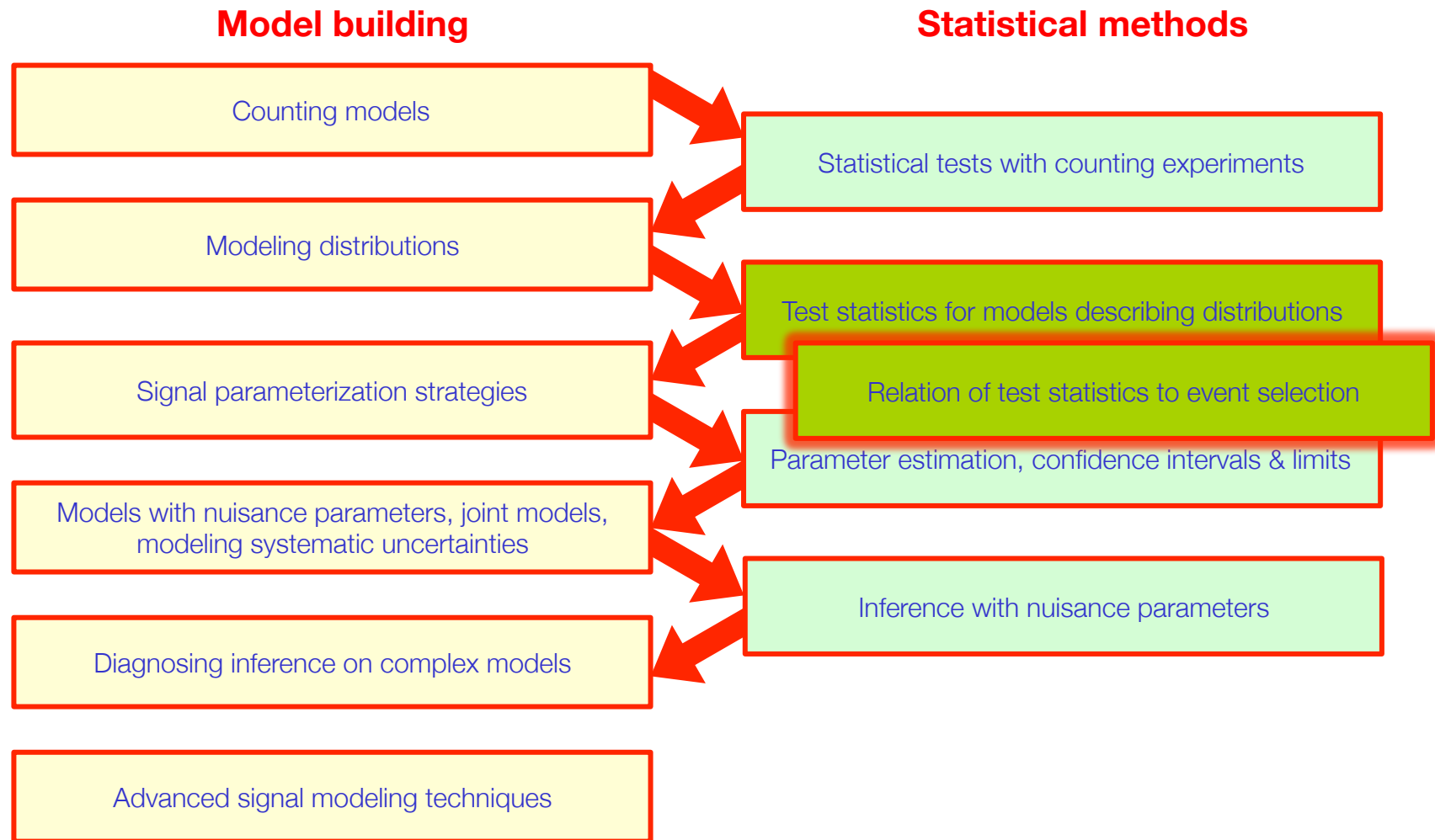
- For *binned distributions*, can generate content of each bin on toy dataset independently, using a Poisson process



- Note that efficient generation of Poisson random number relies on a combination of importance sampling (for small μ , using exponential envelope, for large μ using Cauchy distribution)

Roadmap of this course

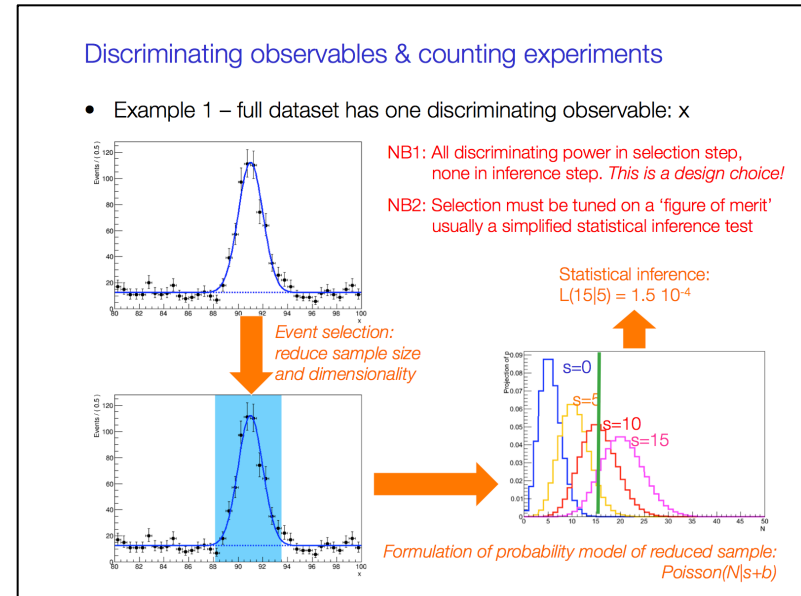
- Start with basics, gradually build up to complexity



Deciding on a split

- HEP data analysis often a 2-step process:

first selection,
then inference



- Focus in this course on inference, but Likelihood Ratio as test statistics shows that there is a **general optimal solution for any event selection problem**: the ratio will order all event by signal-likeness

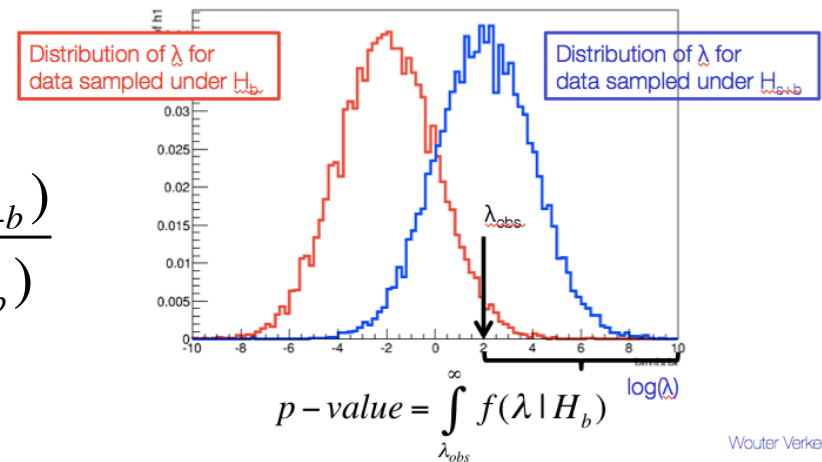
$$\lambda(\vec{x}, \vec{y}, \vec{z}, \dots) = \frac{L(\vec{x}, \vec{y}, \vec{z}, \dots | H_{s+b})}{L(\vec{x}, \vec{y}, \vec{z}, \dots | H_b)}$$

- Hence if we can construct λ , a selection defined by $\lambda > \lambda_c$ will always be optimal for some stated level of desired purity

The Likelihood Ratio test statistic as tool for event selection

- Note that hypothesis testing with two simple hypotheses for observable distributions, exactly describes ‘event selection’ problem
- In fact we have already ‘solved’ the optimal event selection problem! Given two hypothesis H_{s+b} and H_b that predict an complex multivariate distribution of observables, **you can always classify all events in terms of ‘signal-likeness’ (a.k.a ‘extremity’) with a likelihood ratio**

$$\lambda(\vec{x}, \vec{y}, \vec{z}, \dots) = \frac{L(\vec{x}, \vec{y}, \vec{z}, \dots | H_{s+b})}{L(\vec{x}, \vec{y}, \vec{z}, \dots | H_b)}$$



- So far we have exploited λ to calculate a frequentist p-value **now explore properties ‘cut on λ ’ as basis of (optimal) event selection**

Event selection

- The event selection problem:
 - Input: Two classes of events “signal” and “background”
 - Output: Two categories of events “selected” and “rejected”
- Goal: select as many signal events as possible, reject as many background events as possible
- Note that optimization goal as stated is ambiguous.
 - But can choose a well-defined by optimization goal by e.g. fixing desired background acceptance rate, and then choose procedure that has highest signal acceptance.
- Relates to “classical hypothesis testing”
 - Two competing hypothesis (traditionally named ‘null’ and ‘alternate’)
 - Here null = background, alternate = signal

Terminology of classical hypothesis testing

- Definition of terms

- Rate of type-I error = α
- Rate of type-II error = β
- Power of test is $1-\beta$

		Actual condition	
		Guilty	Not guilty
Decision	Verdict of 'guilty'	True Positive	False Positive (i.e. guilt reported unfairly) Type I error
	Verdict of 'not guilty'	False Negative (i.e. guilt not detected) Type II error	True Negative

- Treat hypotheses asymmetrically

- Null hypo is usually special → Fix rate of type-I error
- Criminal convictions: Fix rate of unjust convictions
- Higgs discovery: Fix rate of false discovery
- Event selection: Fix rate of background that is accepted

- Now can define a well stated goal for optimal testing

- Maximize the power of test (minimized rate of type-II error) for given α
- Event selection: Maximize fraction of signal accepted

The Neyman-Pearson lemma

- In 1932-1938 Neyman and Pearson developed a theory in which one must consider competing hypotheses
 - Null hypothesis (H_0) = Background only
 - Alternate hypotheses (H_1) = e.g. Signal + Background

and proved that

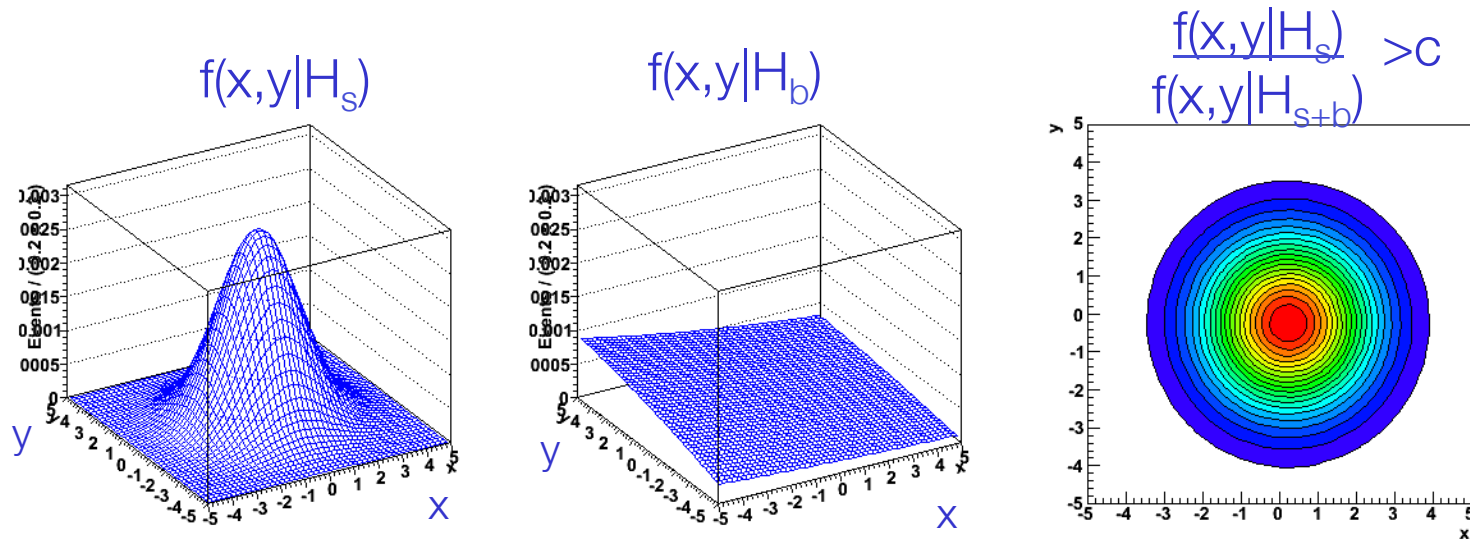
- The region W that minimizes the rate of the type-II error (not reporting true discovery) is a contour of the Likelihood Ratio

$$\frac{P(x|H_1)}{P(x|H_0)} > k_\alpha$$

- Any other region of the same size will have less power

The Neyman-Pearson lemma

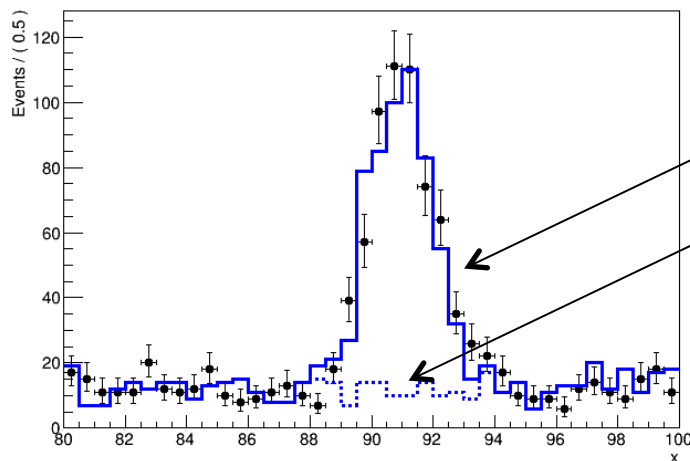
- Example of application of NP-lemma with two observables



- Cut-off value c controls type-I error rate ('size' = bkg rate)
Neyman-Pearson: LR cut gives best possible 'power' = signal eff.
- **So why don't we *always* do this?** (instead of training neural networks, boosted decision trees etc)

Why Neyman-Pearson doesn't always help

- The problem is that we usually don't have explicit formulae for the pdfs $f(\vec{x}|\mathbf{s})$, $f(\vec{x}|\mathbf{b})$.
- Instead we may have Monte Carlo samples for signal and background processes
 - Difficult to reconstruct analytical distributions of pdfs from MC samples, especially if number of dimensions is large
- If physics problem has only few observables can still estimate estimate pdfs with histograms or kernel estimation,
 - But in such cases one can also forego event selection and go straight to hypothesis testing / parameter estimation with all events



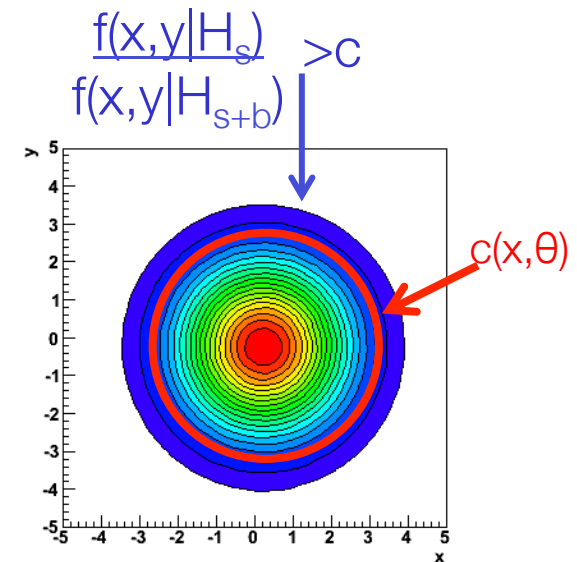
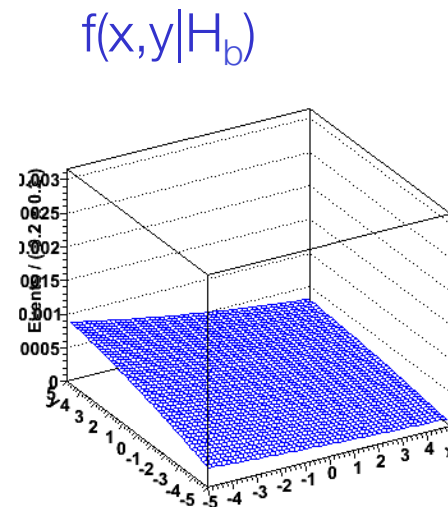
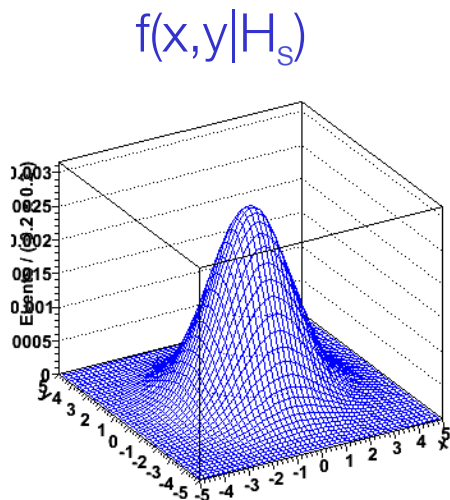
Approximation of true $f(x|\mathbf{s})$

Approximation of true $f(x|\mathbf{b})$

Hypothesis testing with a large number of observables

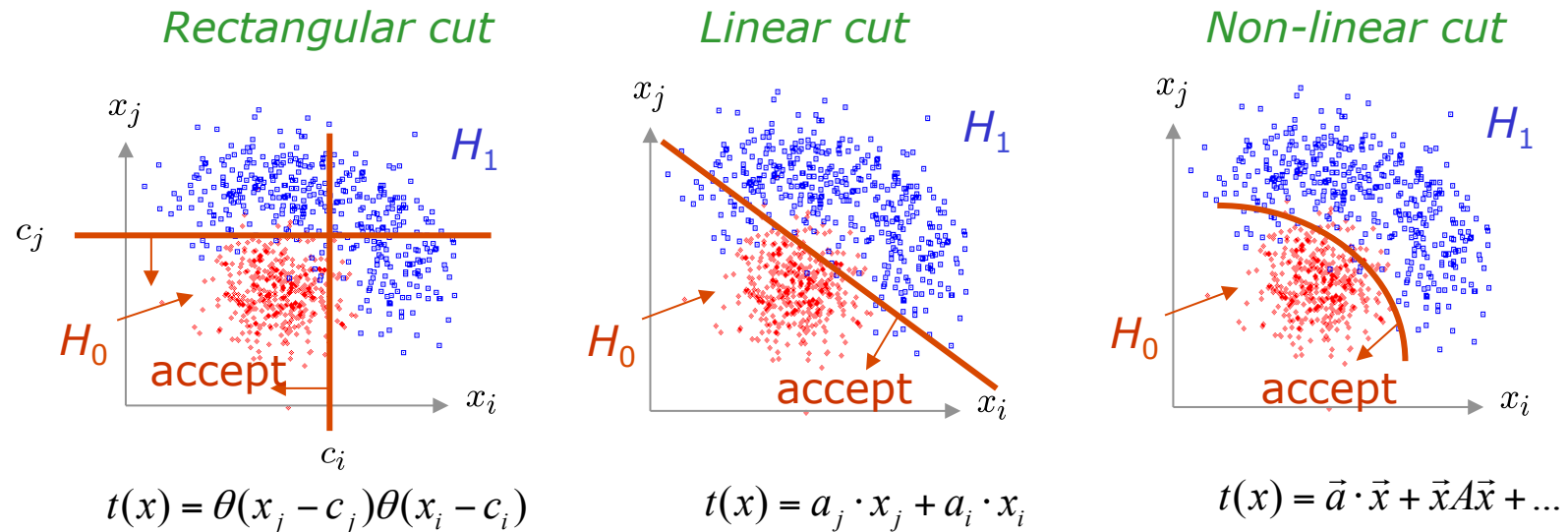
- When number of observables is large follow different strategy
- Instead of aiming at approximating p.d.f.s $f(x|s)$ and $f(x|b)$ aim to approximate decision boundary with an empirical parametric form

$$A_\alpha(\vec{x}) = \left[\frac{f(\vec{x}|s)}{f(\vec{x}|s+b)} > \alpha \right] \Rightarrow A_\alpha(\vec{x}) = c(\vec{x}, \vec{\theta})$$




Empirical parametric forms of decision boundaries

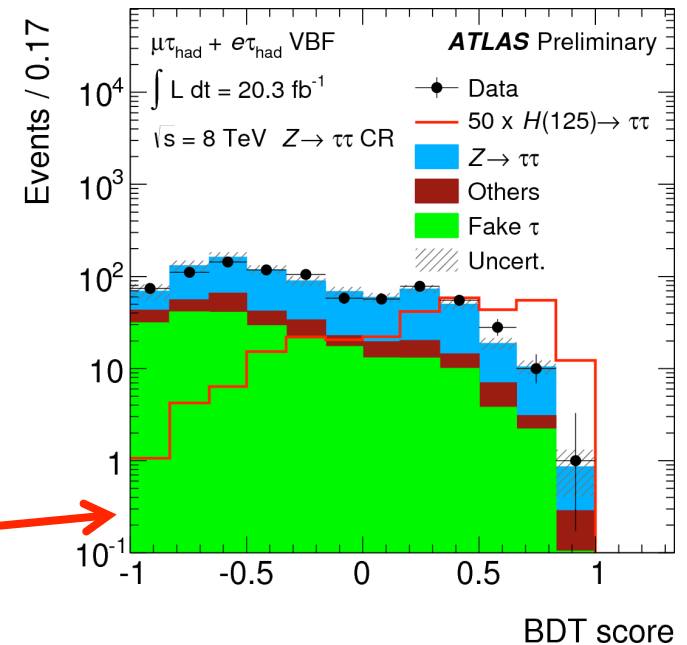
- Can in principle choose any type of Ansatz parametric shape



- Goal of Ansatz form is estimate of a ‘signal probability’ for every event in the observable space x (just like the LR)
- Choice of desired type-I error rate (selected background rate), can be set later by choosing appropriate cut on Ansatz test statistic.

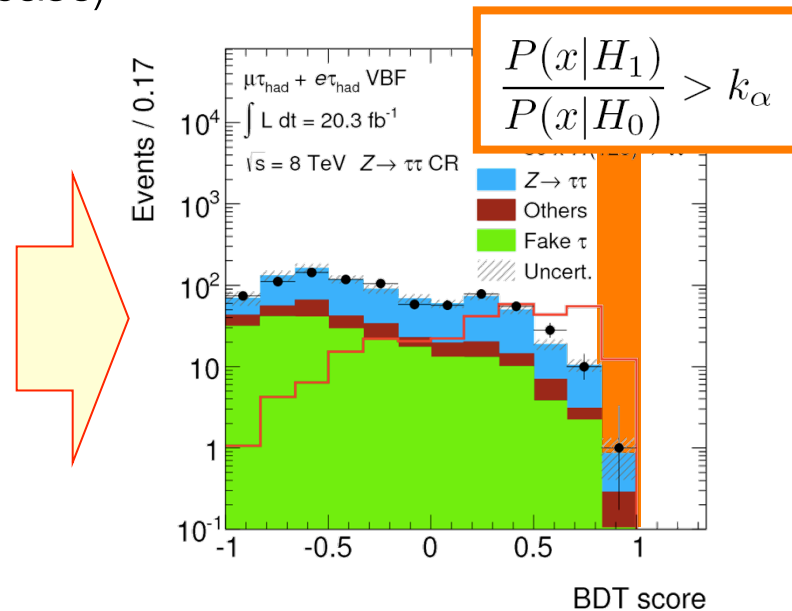
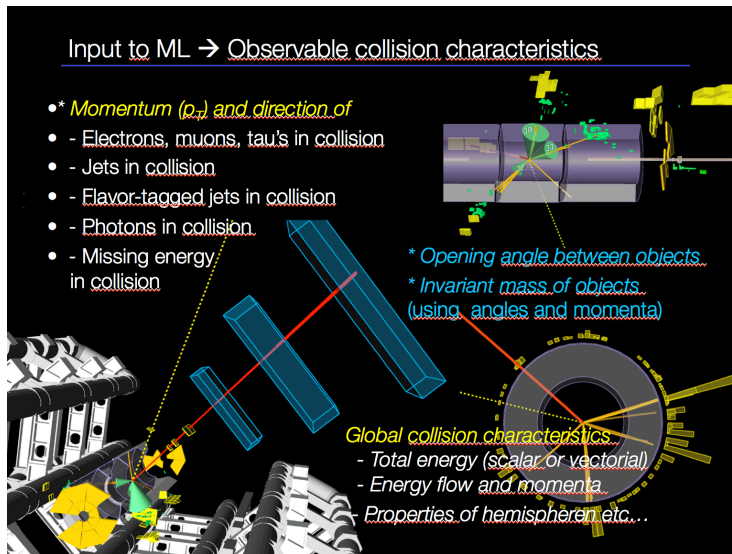
Machine learning and all that

- A wide range of modern tools exist to perform supervised learning of a multivariate discriminant with the aim to approximate the optimal Neyman-Pearson discriminant.
 - Deep Learning, Boosted Decision Trees, GAN's etc etc.
- Variation in
 - Ansatz (empirical parametric form of discriminant)
 - Learning process (error back propagation, Bayesian)
- Commonality in
 - Input (labeled simulation samples)
 - Output (single function that maps signal probability) 
- In all cases output functions is functionally comparable to likelihood ratio discriminant (modulo some trivial transformations)

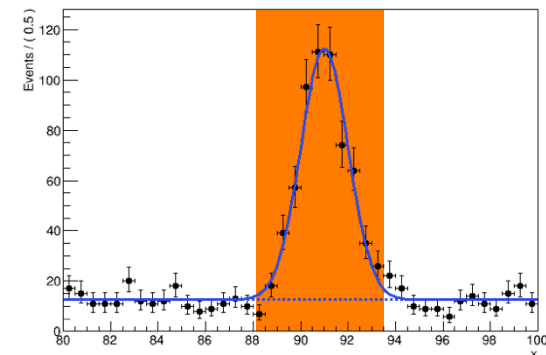


Event selection as dimensionality reduction

- In the limit of an optimal discriminant – **the event selection step is effectively (and only) a reduction of dimensionality of the data** without loss of information (in the optimal case)

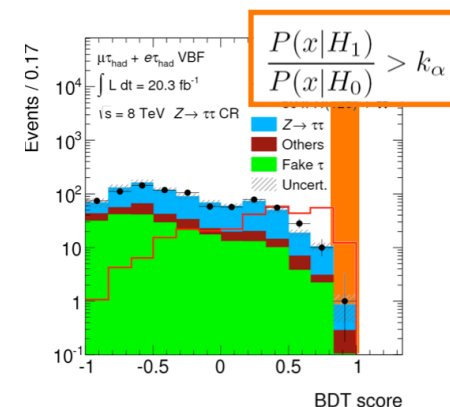


- In case the full discriminant distribution is tested → no loss of information
 - But need for pdf that model distribution
- But can also select high-signal region and perform simplified inference
 - e.g. counting model in that region



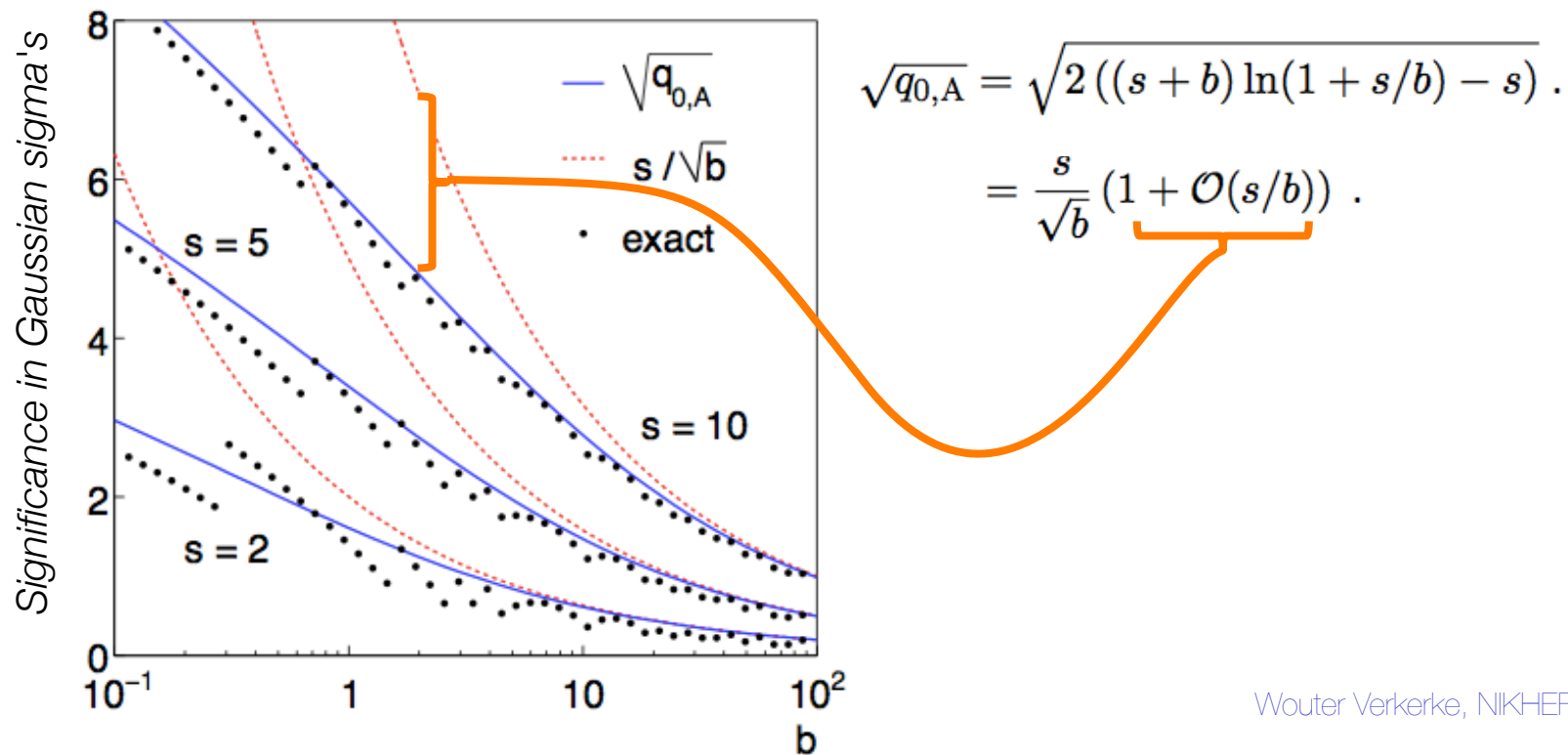
Choosing the ‘best’ high-signal region

- A common scenario for searches in a low-statistics regime is to perform a simplified analysis
 1. Train MVA to obtain discriminant D
 2. Apply a cut on D
 3. Perform only a counting analysis
- And a common question is then – **what is the ‘optimal cut on D’?**
 - NB: the question arise due to choice for simplified counting in step 3). If a *probability density model* is used for the analysis the answer is always ‘the full range of the discriminant’
 - To answer question a ‘figure of merit’ (FOM) must be chosen that quantifies the optimality of the selection. **The ideal FOM for a search is usually the expected signal significance.**
 - A ‘traditional’ choice is $FOM=s/\sqrt{b}$. For low-statistic searches this is a bad choice! It assumes Gaussian distribution, whereas the true distribution is Poisson, which is quite unlike Gaussian especially in the tails at low N
 - A better, and equally easy to use, equation exists based on a Poisson calculation



Choosing the 'best' high-signal region

- The estimated significance assuming a Poisson process modeled by $\text{Poisson}(N|S+B)$ is $\sqrt{2((s+b)\ln(1+s/b) - s)}$.
- E.g. for 'discovery FOM' s/\sqrt{b} illustration of approximation for $s=2,5,10$ and b in range $[0.01-100]$ *shows significant deviations of s/\sqrt{b} from actual significance at low b*

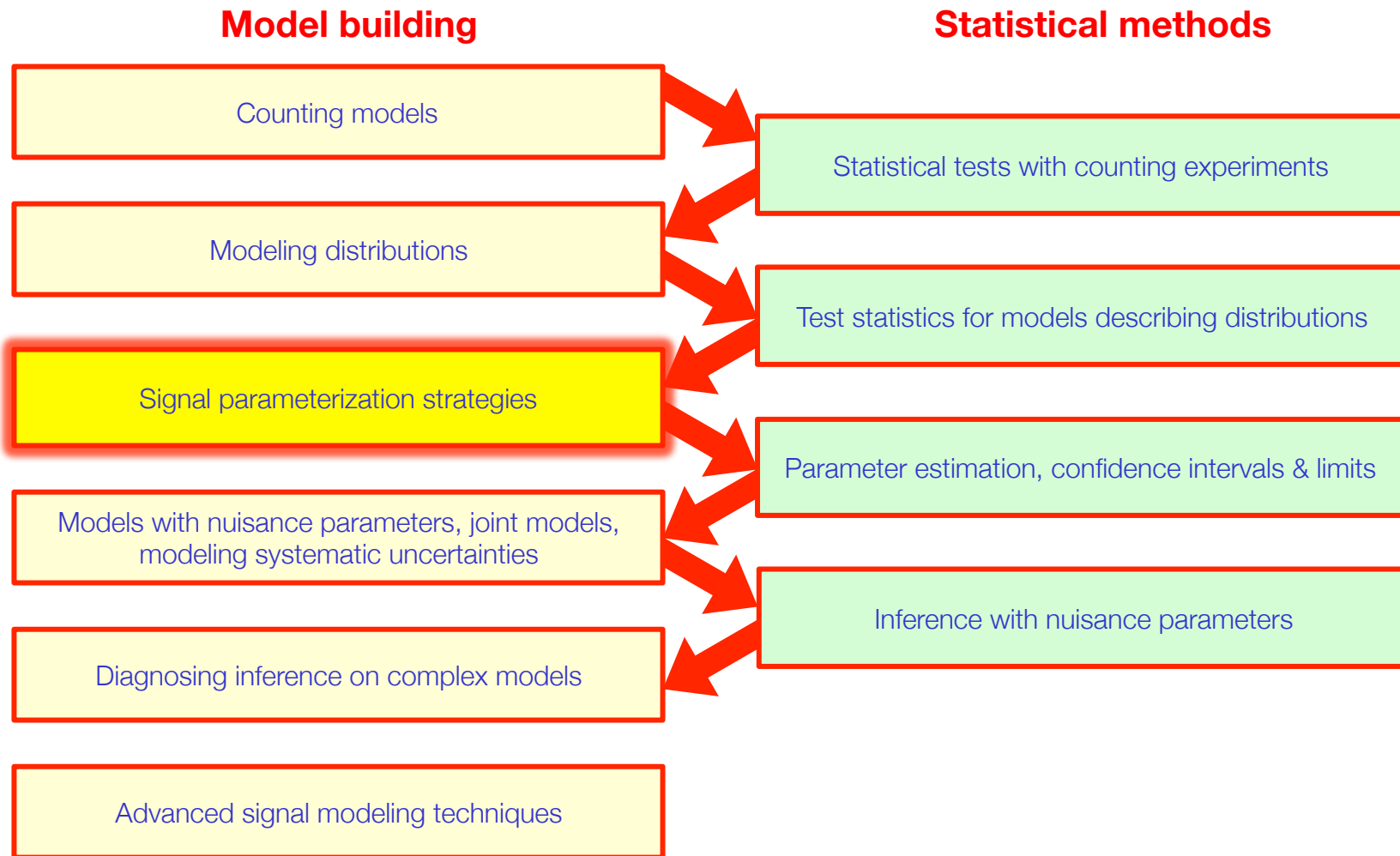


Model building 3

Models with parameters I -
analytical parametric models,
multi-dimensional models
template morphing approach for
histogram-based models

Roadmap of this course

- Start with basics, gradually build up to complexity

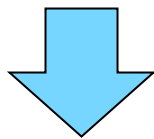


Introduce concept of composite hypotheses

- In most cases in physics, a hypothesis is not “simple”, but “composite”
- **Composite hypothesis** = Any hypothesis which does *not* specify the population distribution completely
- Example: counting experiment with signal and background, that leaves signal expectation unspecified

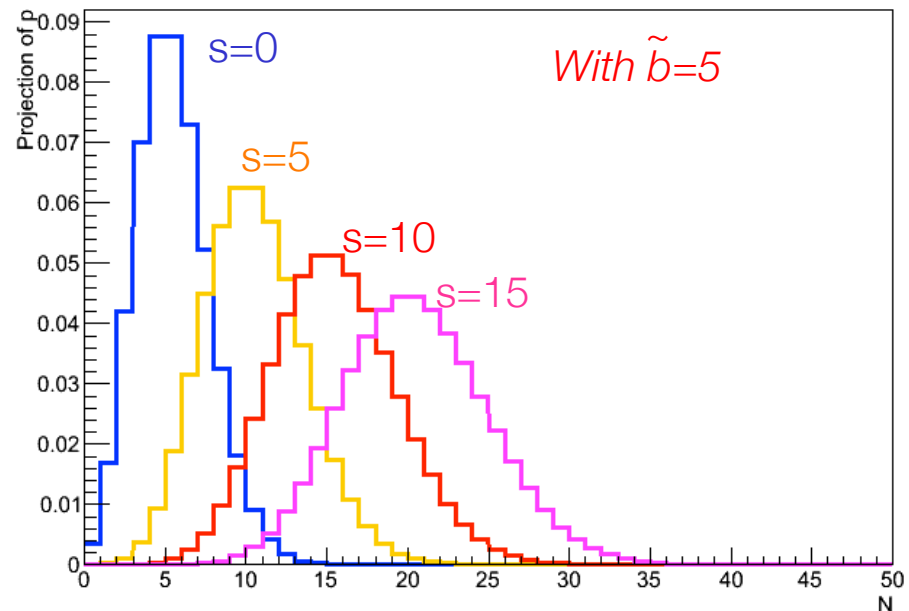
Simple hypothesis

$$L = \text{Poisson}(N | \tilde{s} + \tilde{b})$$



$$L(s) = \text{Poisson}(N | s + \tilde{b})$$

Composite hypothesis



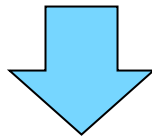
(My) notation convention: all symbols with \sim are constants

A common convention in the meaning of model parameters

- A common convention is to recast signal rate parameters into a normalized form (e.g. w.r.t the Standard Model rate)

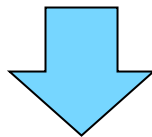
Simple hypothesis

$$L = \text{Poisson}(N | \tilde{s} + \tilde{b})$$



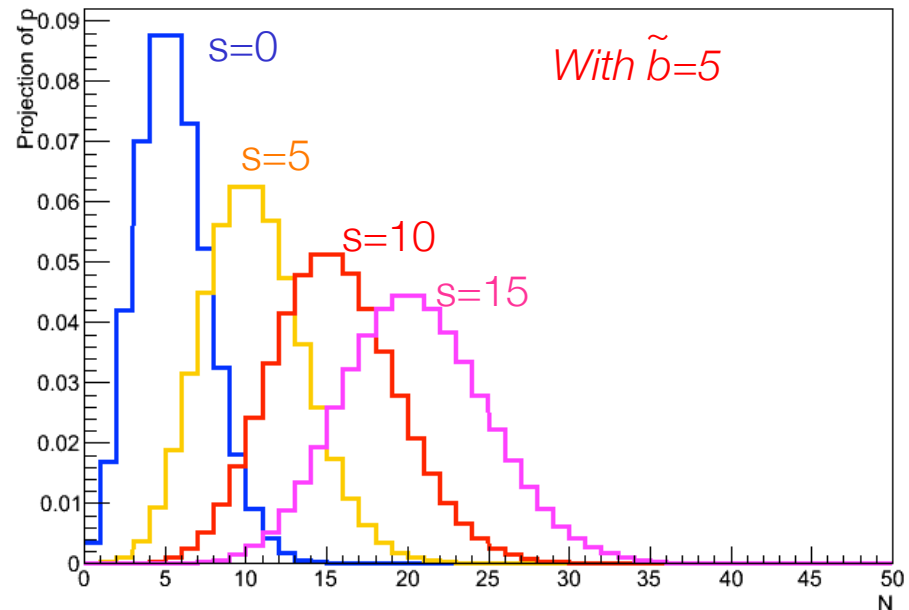
$$L(s) = \text{Poisson}(N | s + \tilde{b})$$

Composite hypothesis



$$L(\mu) = \text{Poisson}(N | \mu \cdot \tilde{s} + \tilde{b})$$

Composite hypothesis
with normalized rate parameter



*'Universal' parameter interpretation
makes it easier to work with your models*

$\mu=0 \rightarrow$ no signal

$\mu=1 \rightarrow$ expected signal

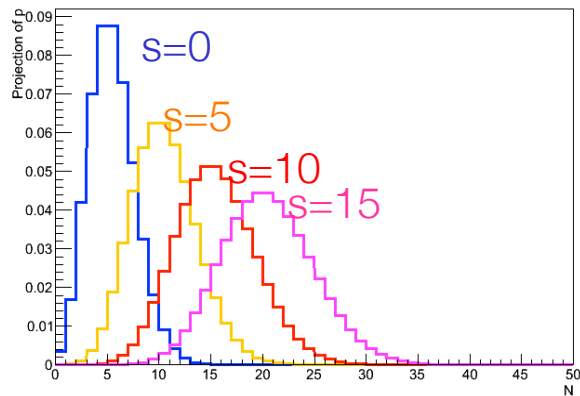
$\mu>1 \rightarrow$ more than expected signal

What can we do with composite hypothesis

- With simple hypotheses – inference is restricted to making statements about $P(D|\text{hypo})$ or $P(\text{hypo}|D)$
- With composite hypotheses – many more options
- 1 Parameter estimation and variance estimation
 - What is value of s for which the observed data is most probable?
 - What is the variance (std deviation squared) in the estimate of s ? } $s=5.5 \pm 1.3$
- 2 Confidence intervals
 - Statements about model parameters using frequentist concept of probability
 - $s < 12.7$ at 95% confidence level
 - $4.5 < s < 6.8$ at 68% confidence level
- 3 Bayesian credible intervals
 - Bayesian statements about model parameters
 - $s < 12.7$ at 95% credibility

Model building for discovery, X-section \rightarrow yield parameter

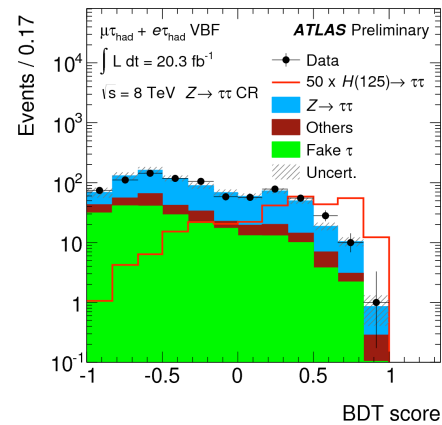
0-dimensional (counting)



$$\text{Poisson}(N|\mathbf{S}+B)$$

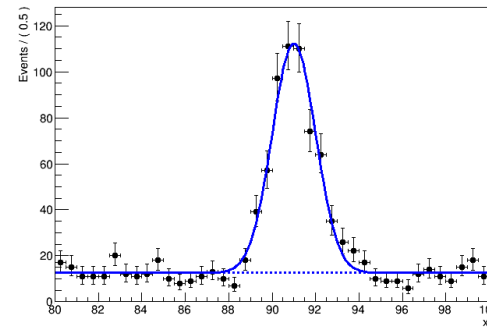
1-dimensional (discriminant)

MVA discriminant



$$\mathbf{S}^* \text{sig}(x) + B^* \text{bkg}(x)$$

Physics-inspired discriminant

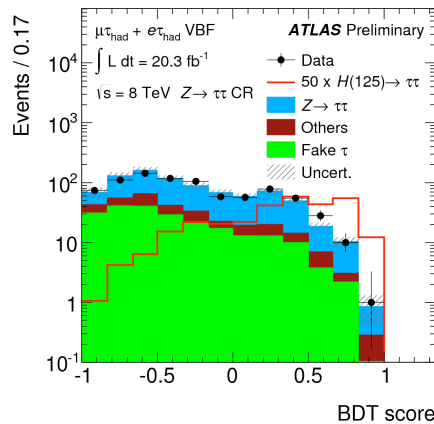


$$\mathbf{S}^* \text{sig}(x) + B^* \text{bkg}(x)$$

Models for discovery, X-section \rightarrow yield parameter

1-dimensional (discriminant)

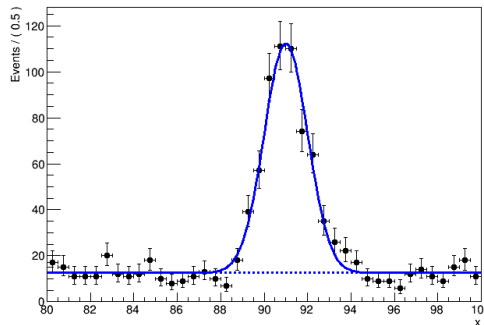
MVA discriminant



$$S^*sig(x)+B^*bkg(x)$$



Physics-inspired discriminant



$$S^*sig(x)+B^*bkg(x)$$

2-dimensional?

Q: When is it useful to build probability models in ≥ 2 observables?

A1: When you have a physics model with a clear prediction for the full 2D model..

Often you don't and then you let an MVA reduce the n-Dim space to 1-dimension

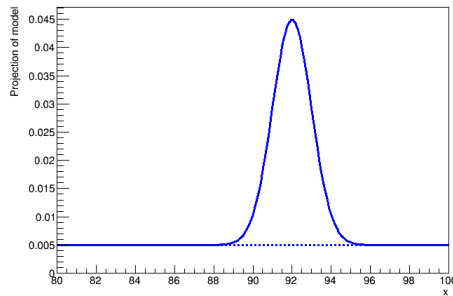
But sometimes you have clear models described 2 or more observables \rightarrow No point in letting an MVA approximate what you know analytically.

Case study – dependence of 1-D model on another observable

- A common scenario for 2D modelling is the following: You observe that the mean reconstructed mass of some particle depends on another observable

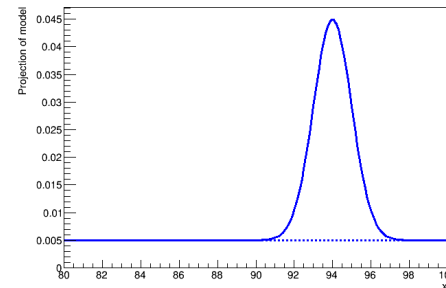
Model for mass at ($y=0$)

$$\text{sig}(m) = \text{Gaussian}(m, 92, 1)$$



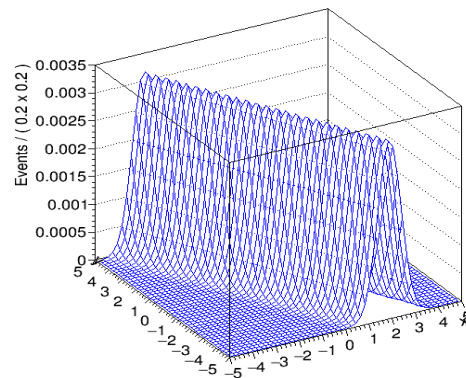
Model for mass at ($y=3$)

$$\text{sig}(m) = \text{Gaussian}(m, 94, 1)$$



$$\text{sig}(m, y) = \text{Gaussian}(m, \text{mean}(y), 1)$$

Solution:
introduce a
function **mean(y)**
that describes
dependence
of mean of y

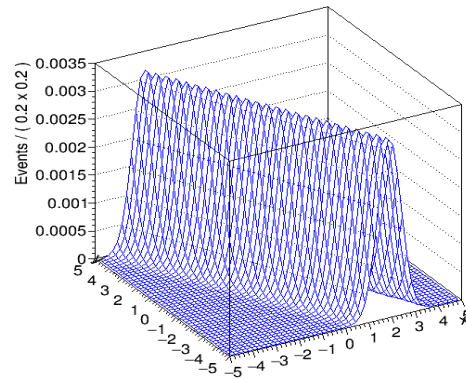


Q: Is $\text{sig}(m, y)$ a proper
2-dimensional model?

Case study – dependence of 1-D model on another observable

$$\text{sig}(m, y) = \text{Gaussian}(m, \text{mean}(y), 1)$$

Solution:
introduce a
function **mean(y)**
that describes
dependence
of mean of y



Q: Is sig(m,y) a proper
2-dimensional model?



A: No!
Distribution in y is
unlikely to be flat...

- Challenge for 2D models: distributions in x,y and all correlations must all be correct! Seems intractable, but solutions exists
- Instead of immediately defining a 2D model **f(x,y)**, define first the *conditional* probability density function **f(x|y)**

$$\begin{aligned} & f(x,y) \\ & = \\ & \text{2D model for} \\ & \text{both x and y} \\ & \int f(x,y) dx dy \equiv 1 \end{aligned}$$

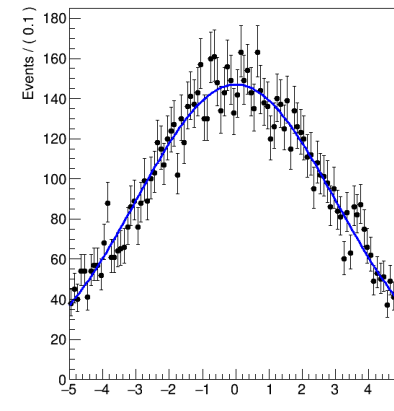
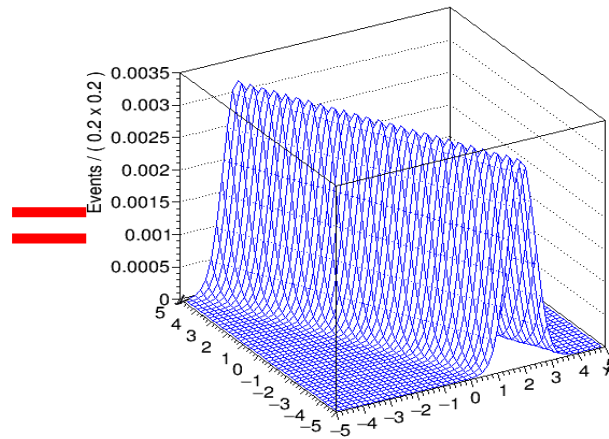
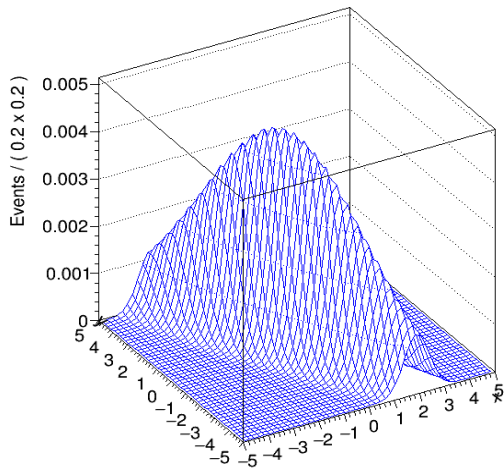
$$\begin{aligned} & f(x|y) \\ & = \\ & \text{1D model for x} \\ & \text{at a given value of y} \\ & \int f(x,y) dx \equiv 1 \quad \forall y \end{aligned}$$

*This is really what
we meant when we
formulated this:*
Gaussian(m, **mean(y)**, 1)

Case study – dependence of 1-D model on another observable

- Given a conditional model $f(x|y)$ can build full 2D model by multiplying with a model $g(y)$

$$\text{sig}(m,y) = \text{sig}_m(m|y) * \text{sig}_y(y)$$

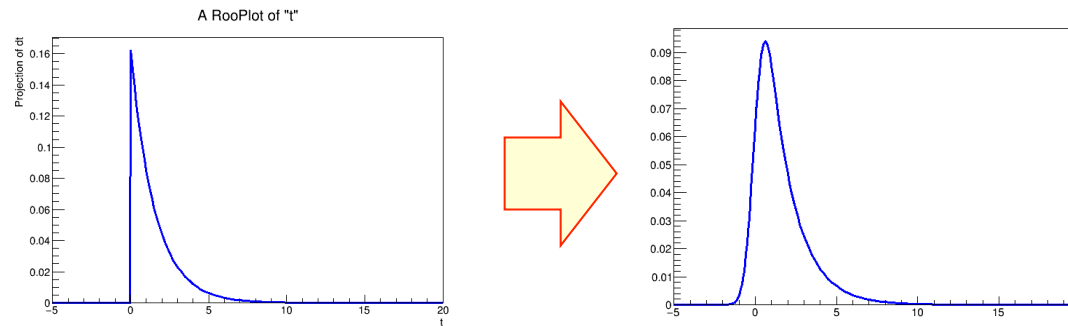
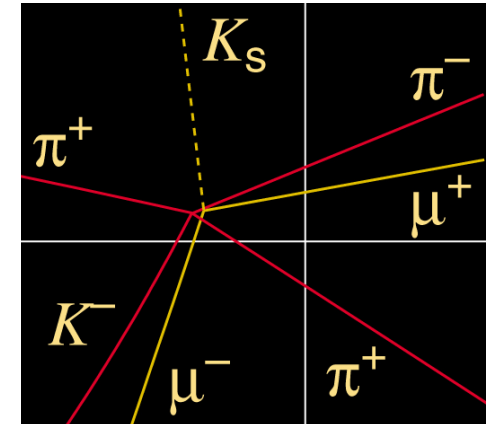


Gaussian($m, \text{mean}(y), 1$)

Gaussian(y)

Case study – per-event errors

- Another common variant of this type of modeling problem is the so-called ‘per-event’ error
- Example: observable = decay time distribution, measured from reconstructed vertex.
 - In absence of a detector resolution, exponential decay distribution
 - In real life, distribution is convoluted with (Gaussian) reconstruction resolution



- But vertex reconstruction gives also estimate of uncertainty for every reconstructed vertex → the ‘per-event error’
 - Can take this into account: well-reconstructed events carry more information
- How? Scale assumed resolution with per-event error

$$f(t | \delta t) = \text{Decay}(t) \otimes \text{Gaussian}(t, 0, \sigma \cdot \delta t)$$

Case study – per-event errors

- Visualization of decay function with variable resolution

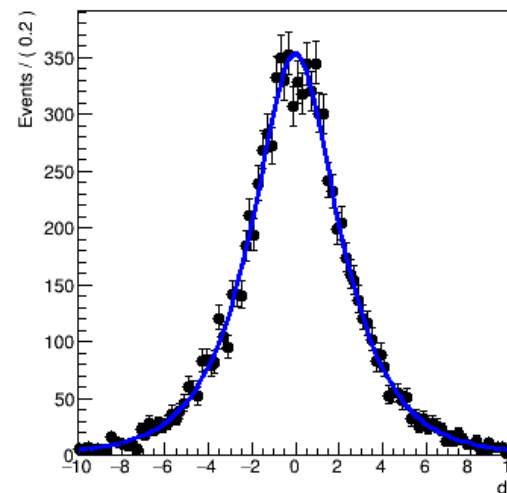
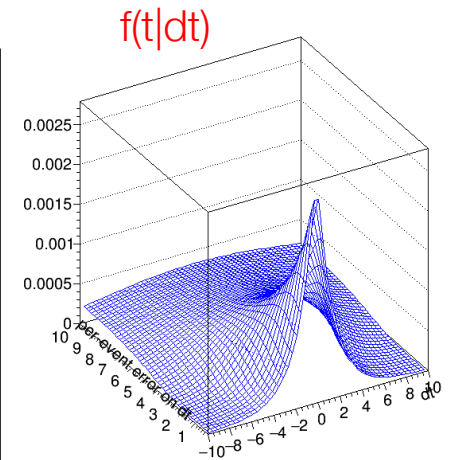
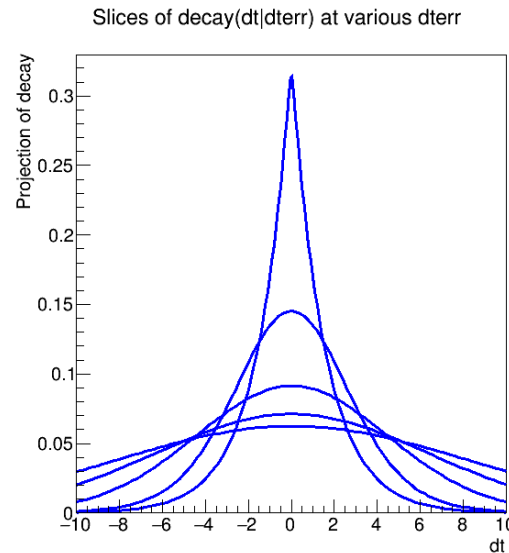
Decay function (symmetrized)
convoluted with Gaussian resolution
at 4 different values of per-event error

$$f(t | \delta t) = \text{Decay}(t) \otimes \text{Gaussian}(t, 0, \sigma \cdot \delta t)$$

Gain: high-resolution events
carry more weight in likelihood →
better estimate of model parameters

Full 2D-model:
 $F(t, dt) = F_1(t|dt) * F_2(dt)$

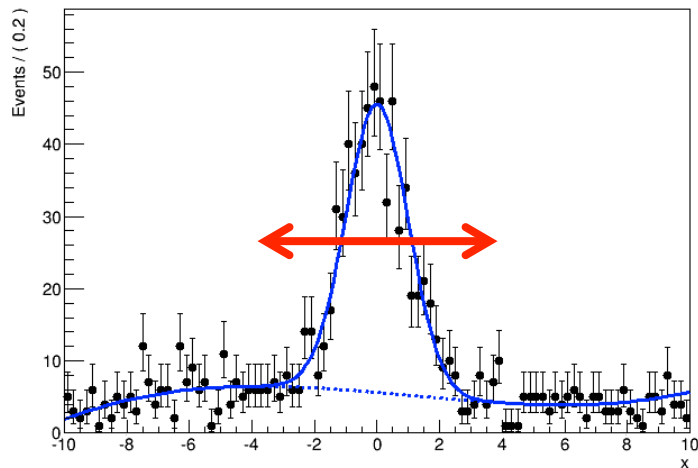
Shown here: *projection on t*
 $F(t) = \int F_1(t|dt) * F_2(dt) dt$



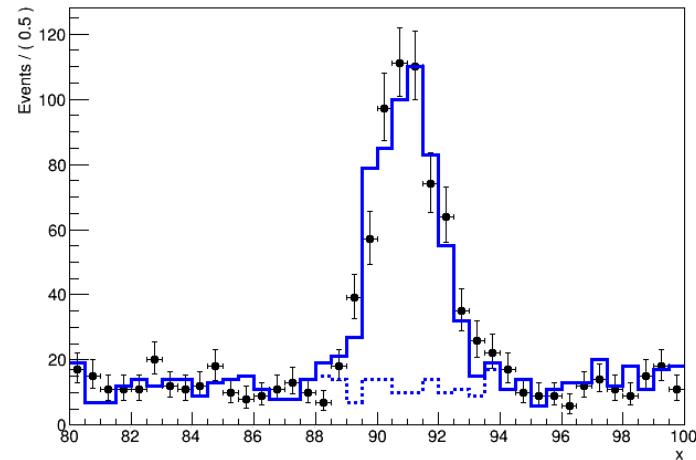
Model building for measurements → shape parameter

- Beyond discovery/rate measurements, can also build models to measure properties of particles (e.g mass)
→ introduce shape parameters
- Often trivial for analytical models,
less so for simulation-based models

$$F(x|\mathbf{m}) = \text{Gaussian}(x, \mathbf{m}, \sigma) + \text{bkg}$$

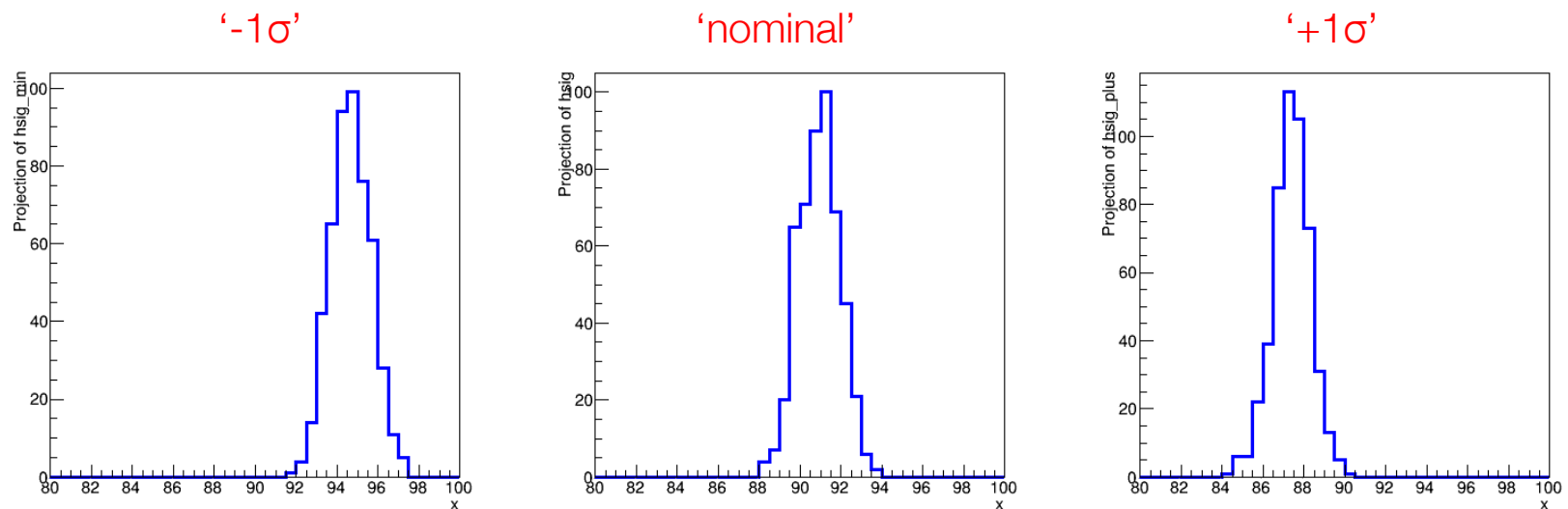


$$F(x|\mathbf{m}) = ??$$



Modeling of shape variations in the likelihood

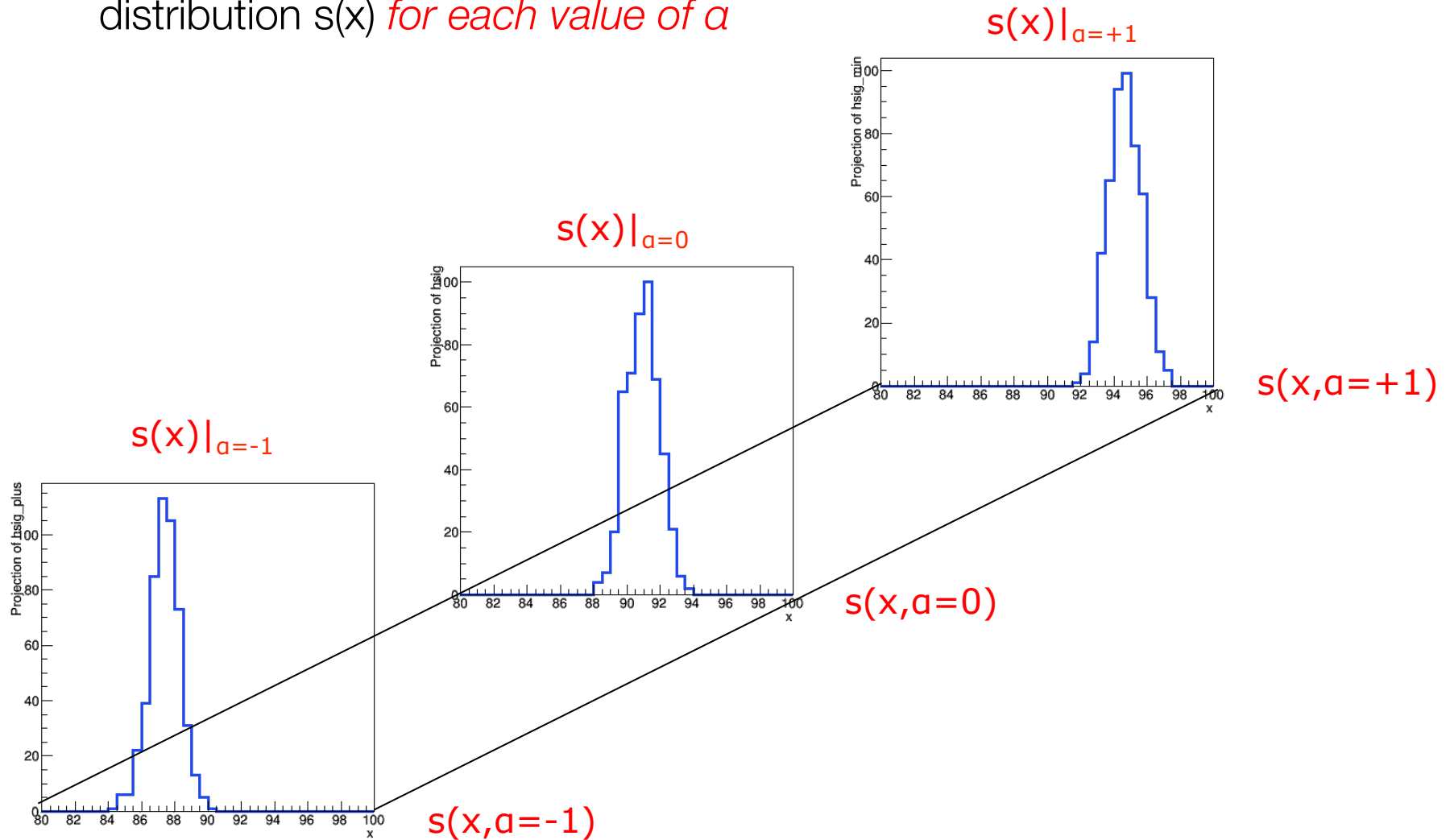
- If underlying simulation has free parameter θ , can assess impact on reconstructed shapes by rerunning simulation at different values
 - Obtain histogram templates for distributions at ‘+1 σ ’ and ‘-1 σ ’ settings of systematic effect



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

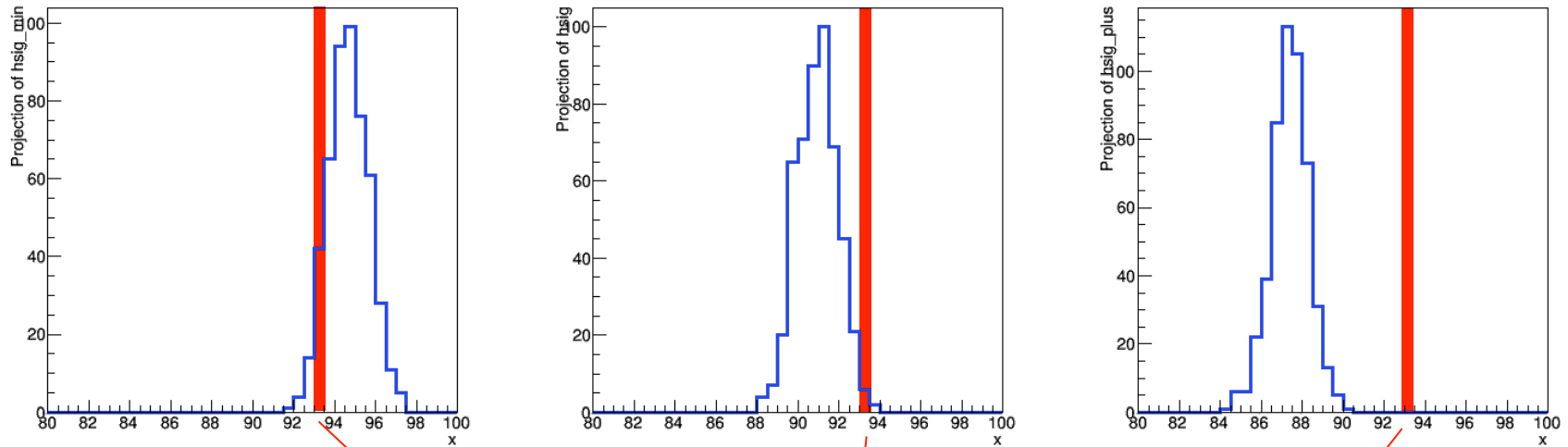
Need to interpolate between template models

- Need to define ‘morphing’ algorithm to define distribution $s(x)$ *for each value of a*

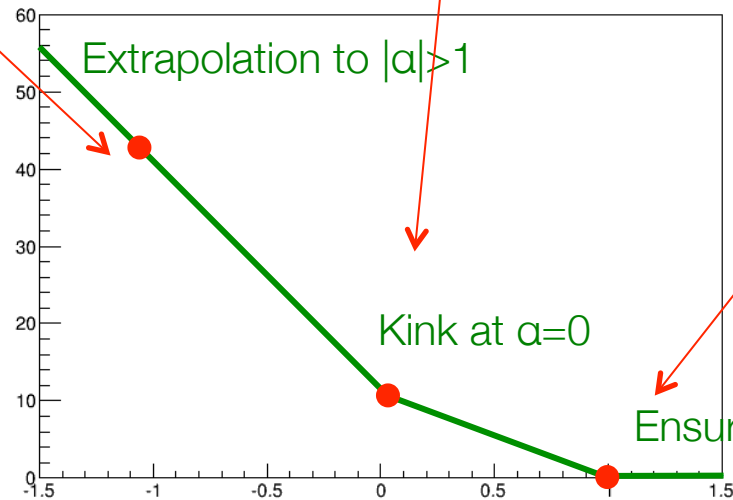


Piecewise linear interpolation

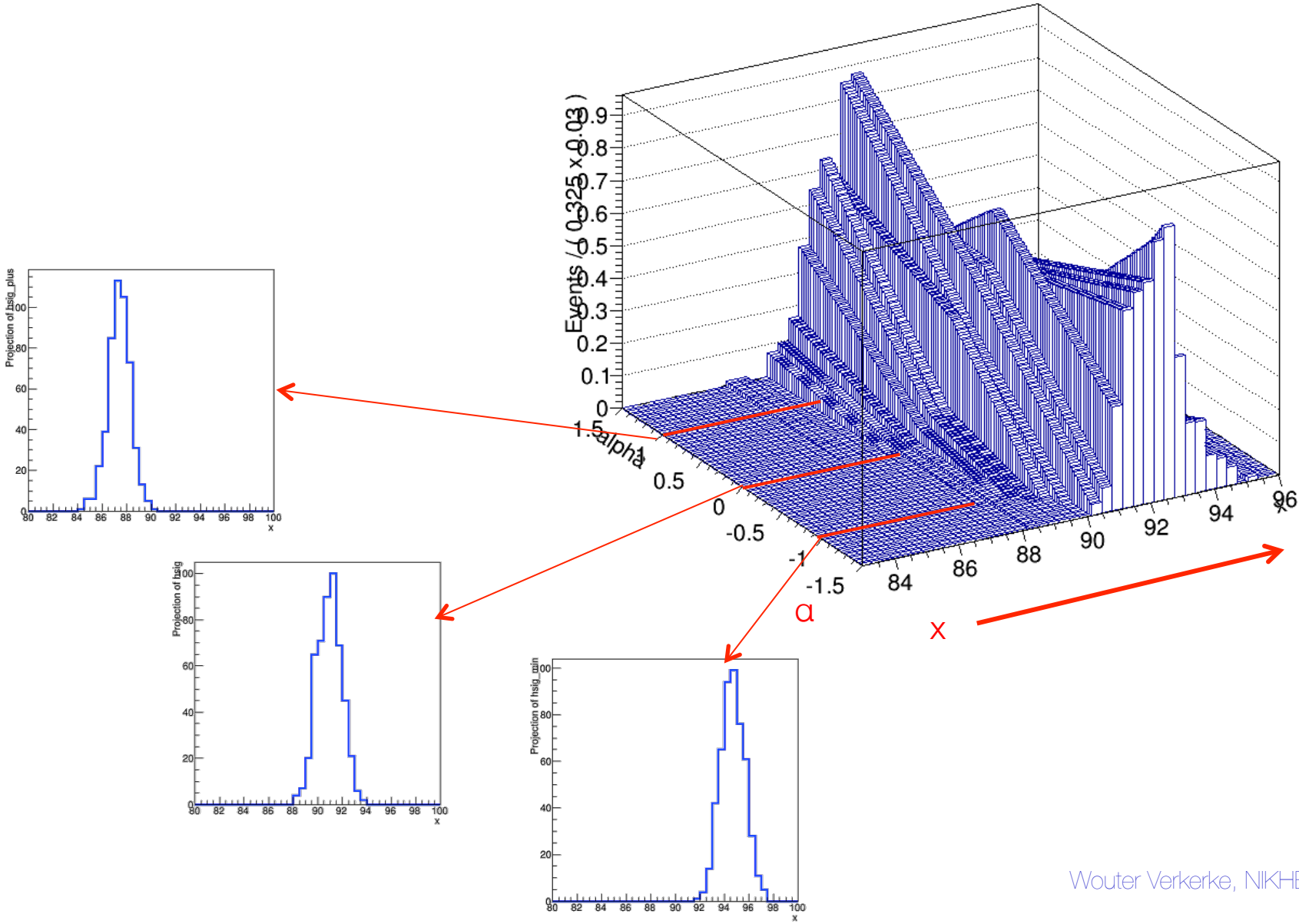
- Simplest solution is piece-wise linear interpolation for each bin



Piecewise linear interpolation response model for a one bin

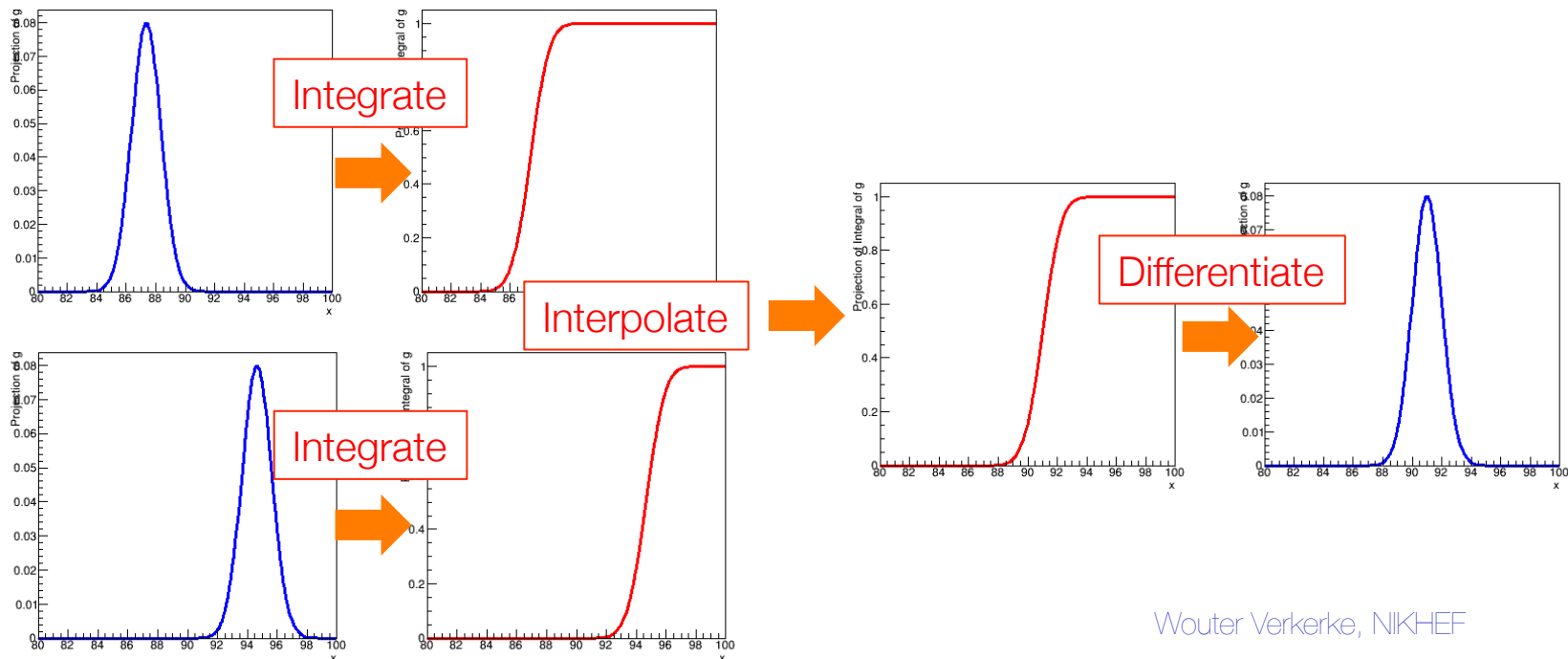


Visualization of bin-by-bin linear interpolation of distribution



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



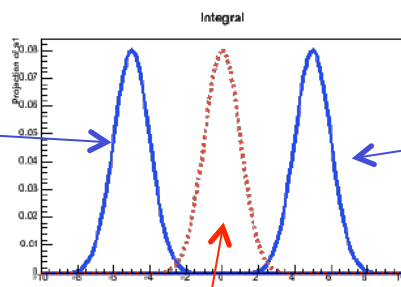
Yet another morphing strategy – ‘Moment morphing’

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)

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

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



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

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

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

$$\begin{aligned} \mu(\alpha) &= \alpha\mu_- + (1 - \alpha)\mu_+ \\ V(\alpha) &= \alpha V_- + (1 - \alpha)V_+ \end{aligned} \quad [1]$$

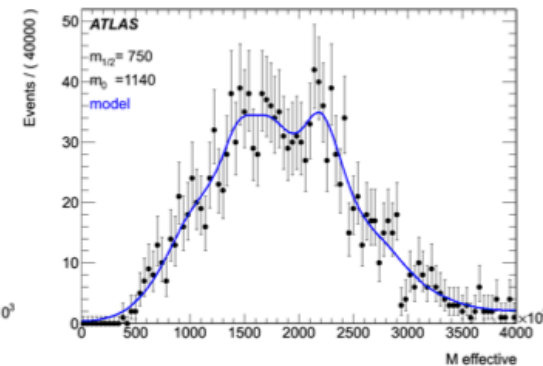
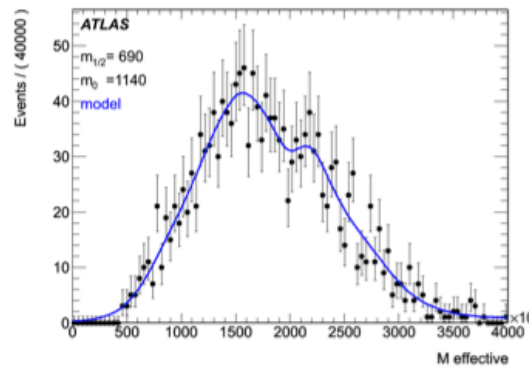
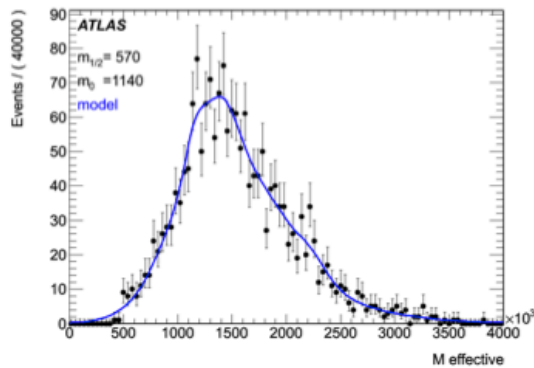
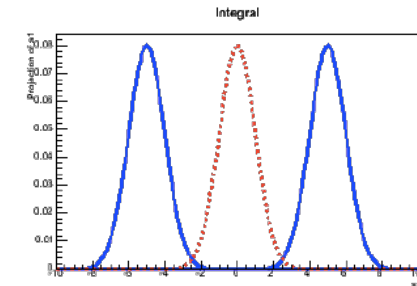
- It constructs this morphed function as combination of linearly transformed input models

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

- Where constants a,b,c,d are chosen such so that $f(x, \alpha)$ 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

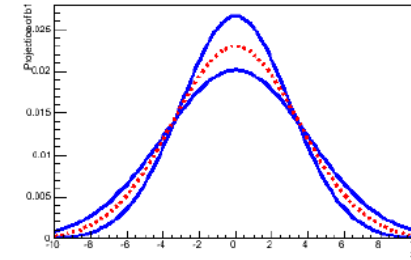
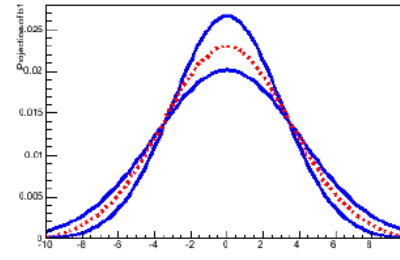
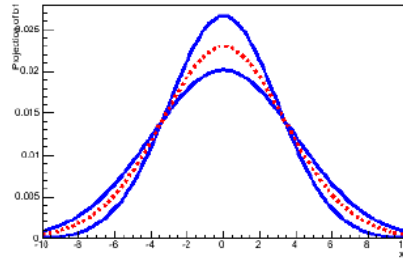
There are other morphing algorithms to choose from

Vertical Morphing

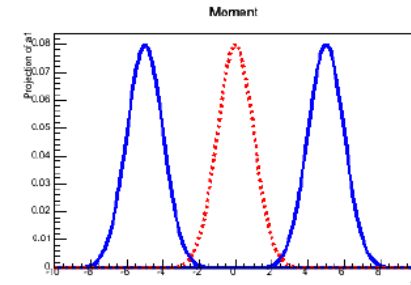
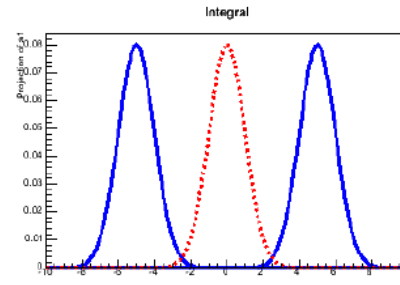
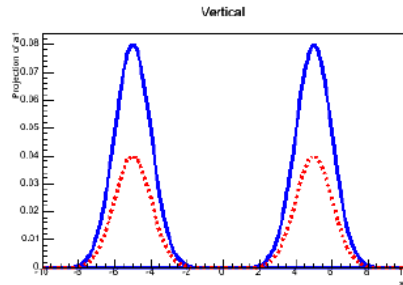
Horizontal Morphing

Moment Morphing

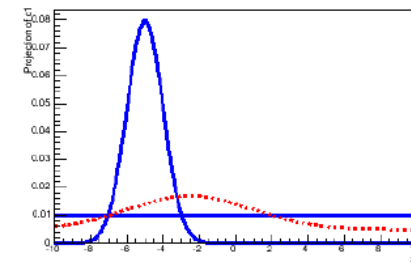
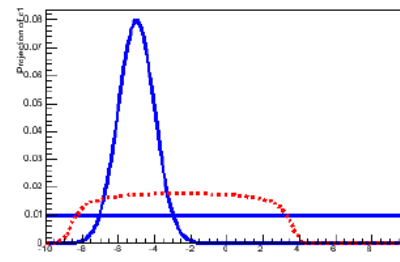
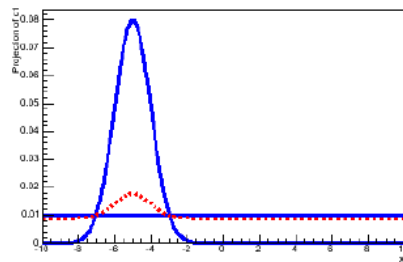
Gaussian
varying
width



Gaussian
varying
mean



Gaussian
to
Uniform
(this is
conceptually ambiguous!)



n-dimensional
morphing?

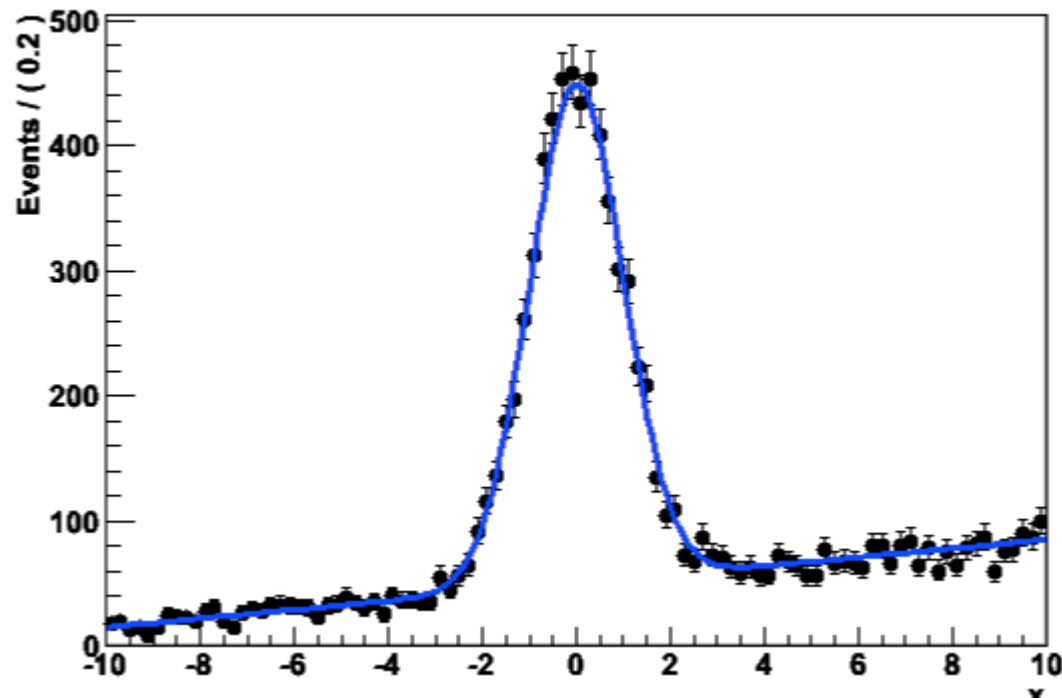


Software tools 1

Basic RooFit modeling

Roofit – Focus: coding likelihood functions

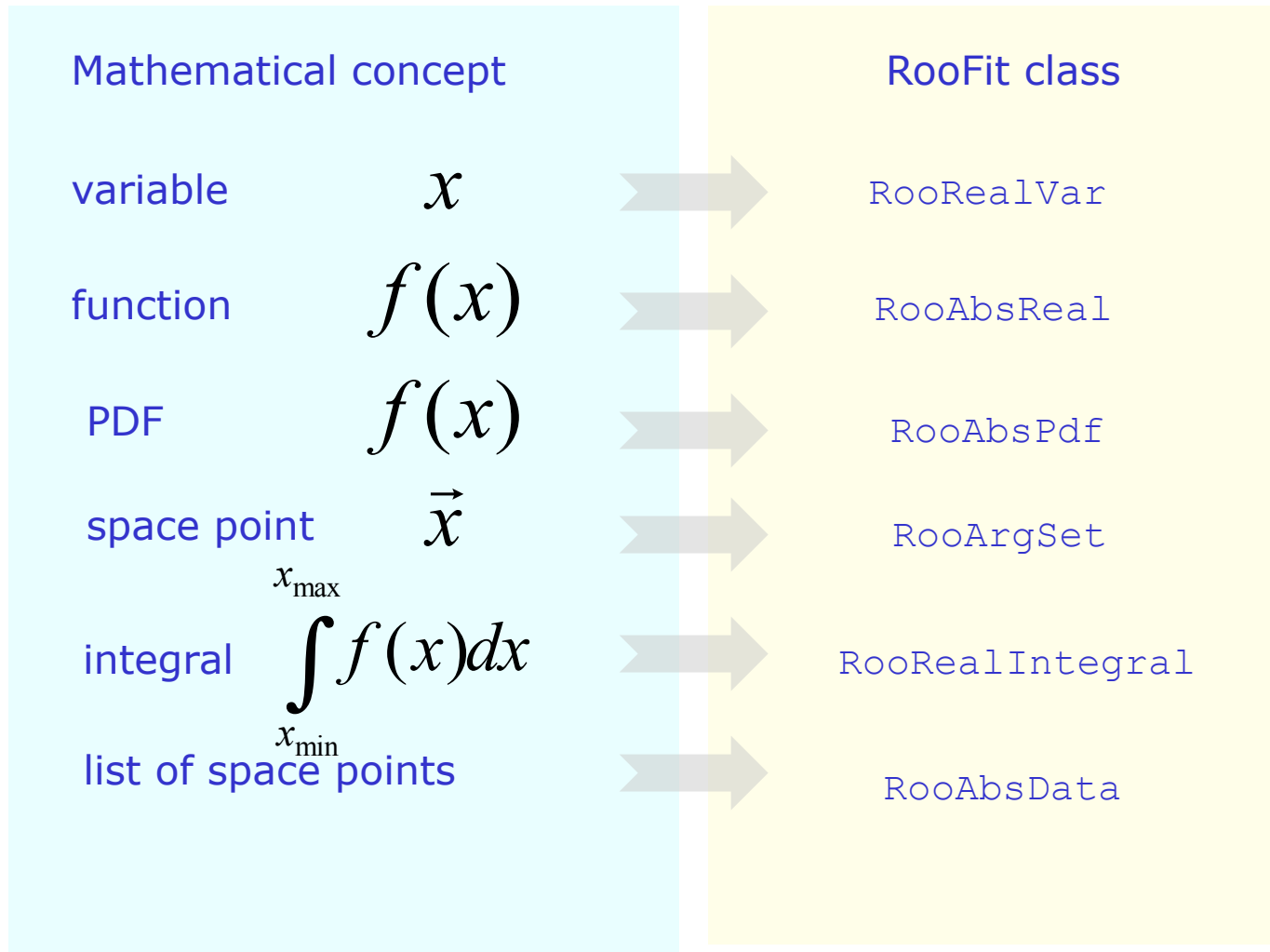
- Focus on one practical aspect of many data analysis in HEP: **How do you formulate your likelihood functions in ROOT**
 - For ‘simple’ problems (gauss, polynomial) this is easy



- 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



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	<pre> graph TD g[RooGaussian g] --> x[RooRealVar x] g --> y[RooRealVar y] g --> z[RooRealVar z] g <--> y </pre>
RooFit code	<pre> 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) ; </pre>

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

Setup gaussian PDF and plot

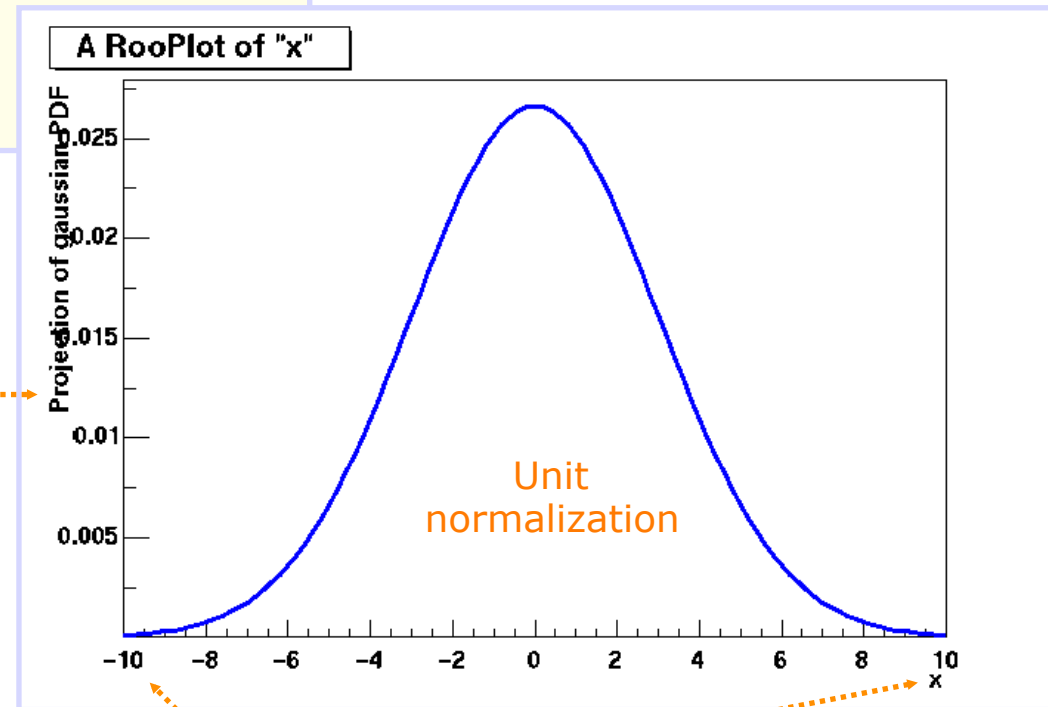
```
// Create an empty plot frame
RooPlot* xframe = w::x.frame() ;

// 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 its variable



Plot range taken from limits of `x`

Basics – Generating toy MC events

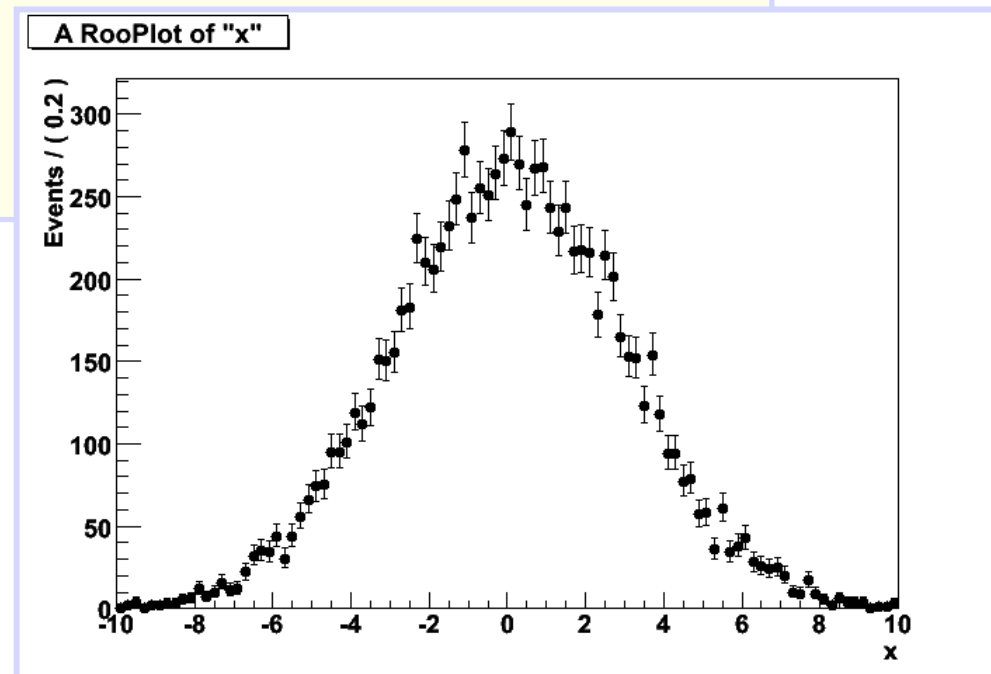
Generate 10000 events from Gaussian p.d.f and show distribution

```
// 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

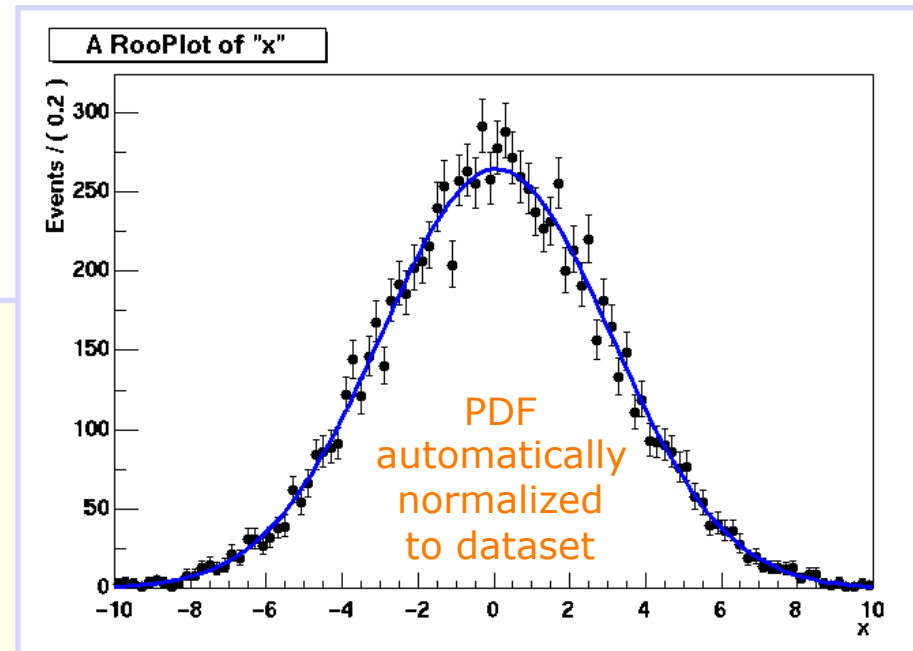


Basics – ML fit of p.d.f to *unbinned* data

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

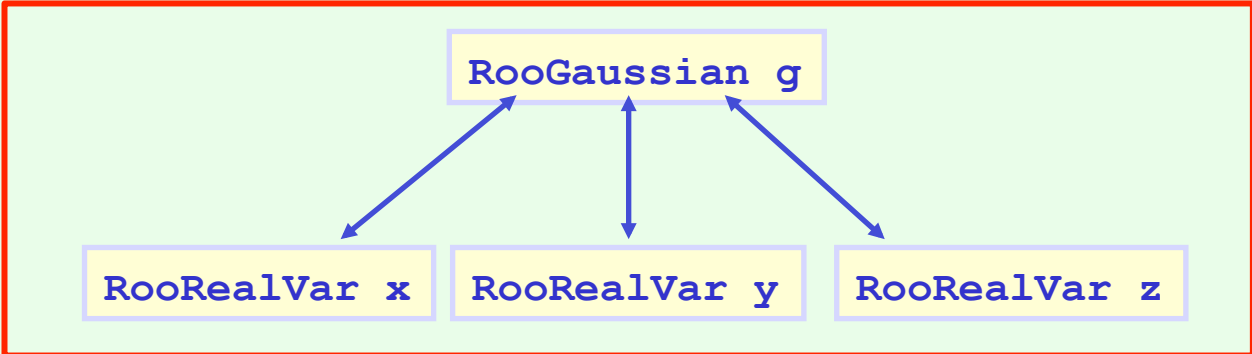
// Parameters if gauss now
// reflect fitted values
w::mean.Print()
RooRealVar::mean = 0.0172335 +/- 0.0299542
w::sigma.Print()
RooRealVar::sigma = 2.98094 +/- 0.0217306

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



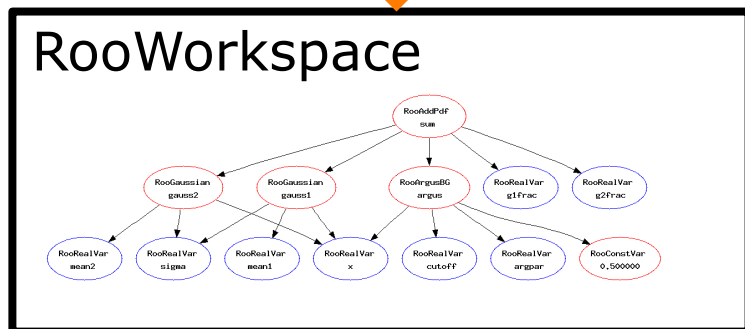
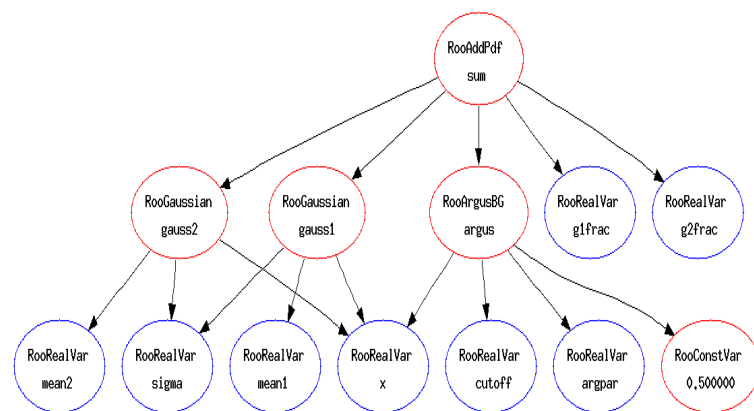
RooFit core design philosophy - Workspace

- The workspace serves a container class for all objects created

Math	$\text{Gauss}(x, \mu, \sigma)$
RooFit diagram	<p style="color: red; text-align: center;">RooWorkspace</p>  <pre> graph TD g[RooGaussian g] <--> x[RooRealVar x] g <--> y[RooRealVar y] g <--> z[RooRealVar z] </pre>
RooFit code	<pre> 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) ; </pre>

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.

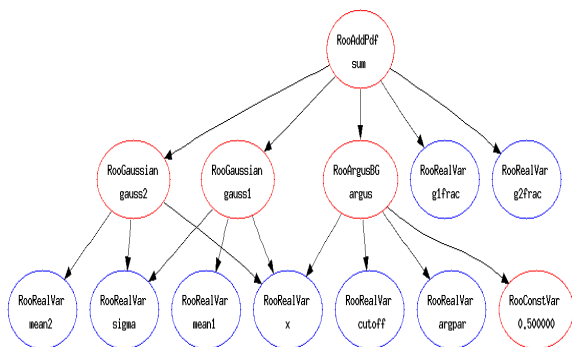
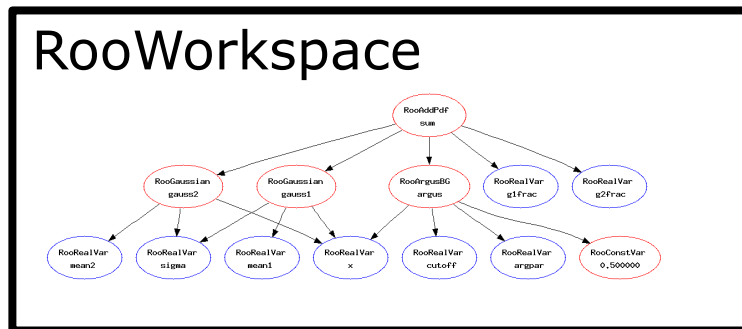


```
RooWorkspace w("w") ;  
w.import(sum) ;  
w.writeToFile("model.root") ;
```

model.root



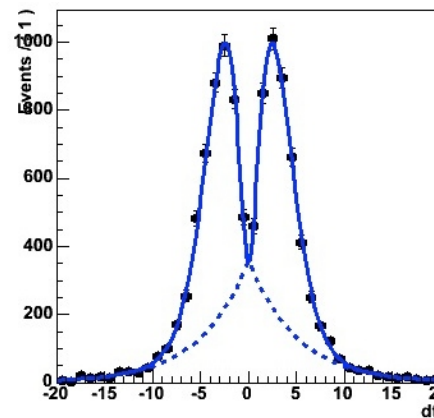
Using a workspace



```
// 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) ;
```



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

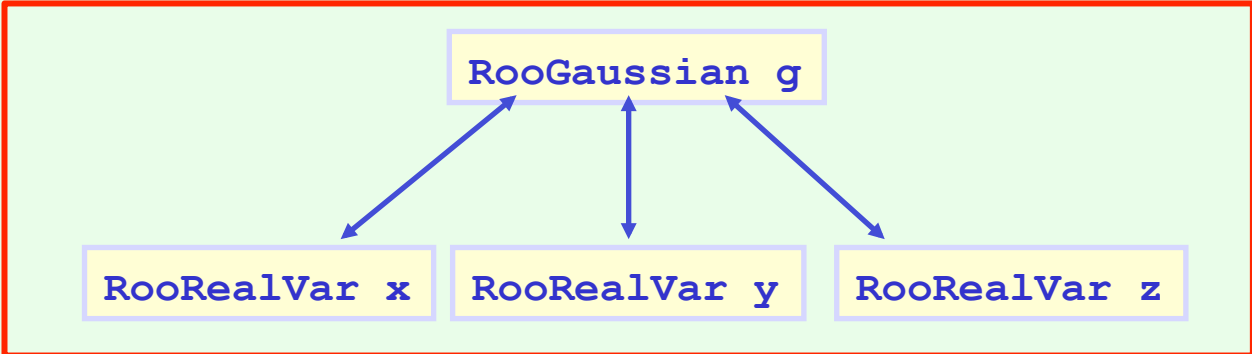
```
RooRealVar x("x","x",-10,10) ;  
RooRealVar mean("mean","mean",5) ;  
RooRealVar sigma("sigma","sigma",3) ;  
RooGaussian f("f","f",x,mean,sigma) ;  
w.import(f) ;
```



```
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	<p style="text-align: center; color: red;">RooWorkspace</p>  <pre> graph TD g[RooGaussian g] <--> x[RooRealVar x] g <--> y[RooRealVar y] g <--> z[RooRealVar z] </pre>
RooFit code	<pre> 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) ; </pre>

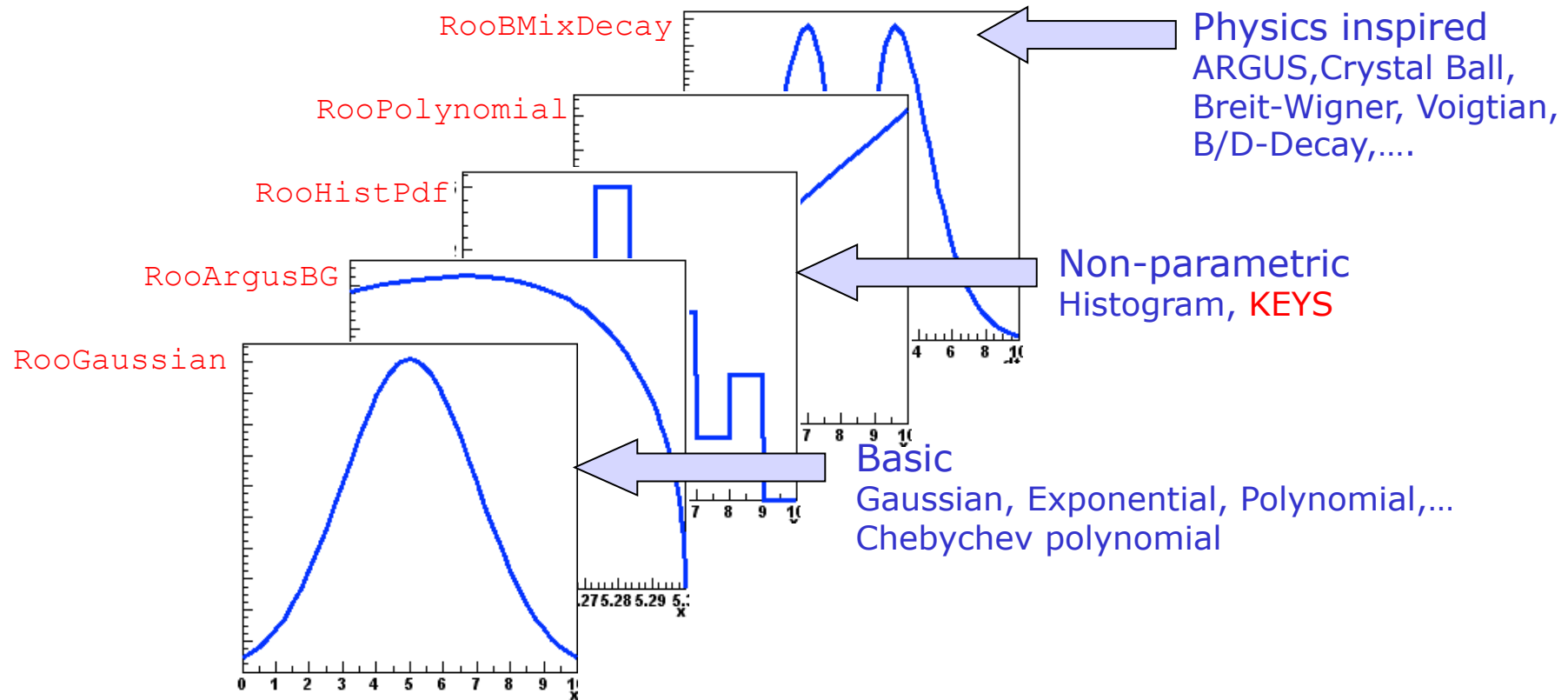
Populating a workspace the easy way – “the factory”

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

Math	$\text{Gauss}(x, \mu, \sigma)$
	RooWorkspace
RooFit diagram	<pre>graph TD; f[RooAbsReal f] --> x[RooRealVar x]; f --> y[RooRealVar y]; f --> z[RooRealVar z]; y <--> f;</pre>
RooFit code	<pre>RooWorkspace w("w") ; w.factory("RooGaussian::g(x[-10,10],m[-10,10],z[3,0.1,10])") ;</pre>

Model building – (Re)using standard components

- RooFit provides a collection of compiled standard PDF classes



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

Exponential::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)

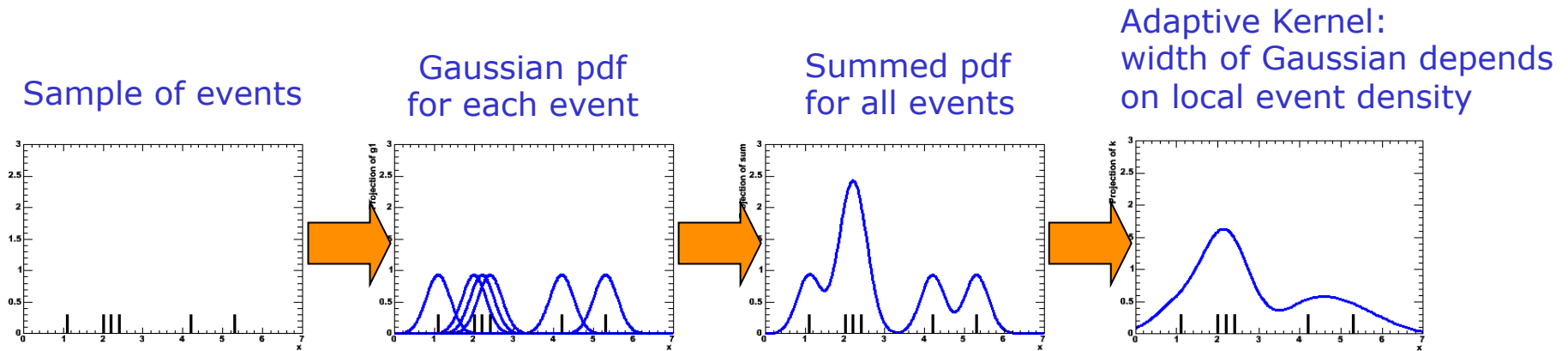
Voigtian

Voigtian::v(x, mean, gamma, sigma)

(=BW⊗G)

The power of pdf as building blocks – Advanced algorithms

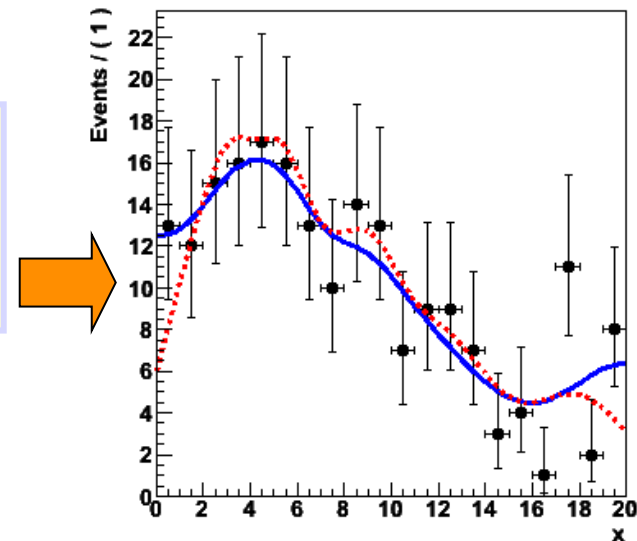
- Example: a ‘kernel estimation probability model’
 - Construct smooth pdf from unbinned data, using kernel estimation technique



- 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

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

also class RooGaussian!

$Gauss(x | \underbrace{\mu \cdot (1 + 2\alpha)}, \sigma)$

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 interpreted function expression on the fly

The power of building blocks – operator expressions

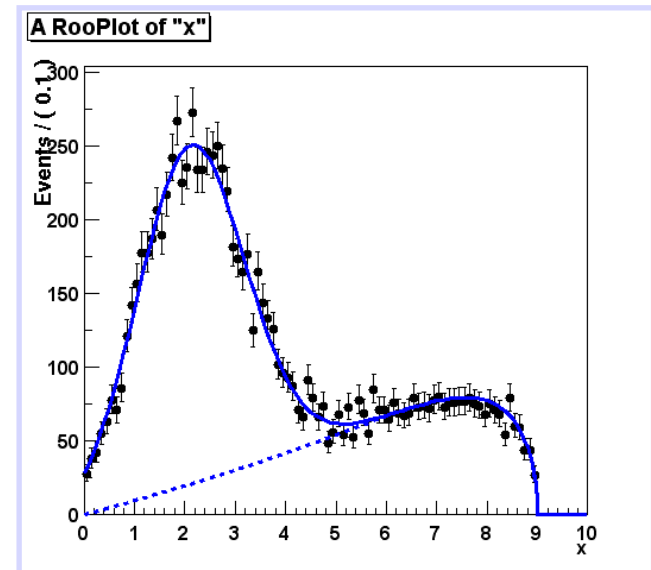
- Create a SUM expression to represent a sum of probability models

```
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)) ;
```

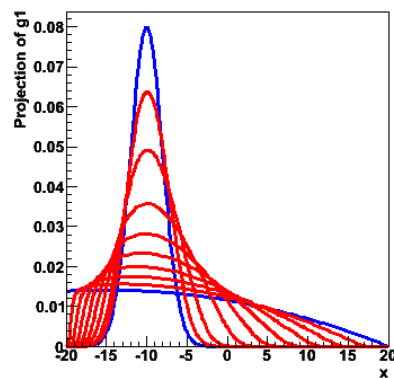


Powerful operators – Morphing interpolation

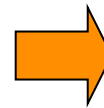
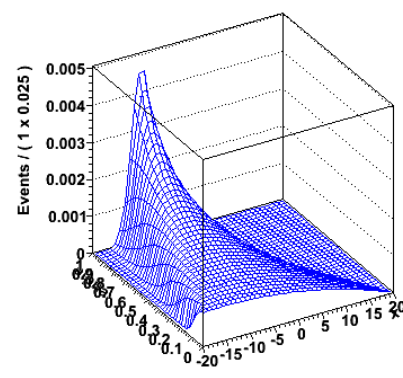
- Special operator pdfs can interpolate existing pdf shapes
 - Ex: interpolation between Gaussian and Polynomial

```
w.factory("Gaussian::g(x[-20,20],-10,2)") ;
w.factory("Polynomial::p(x,{-0.03,-0.001})") ;
w.factory("IntegralMorph::gp(g,p,x,alpha[0,1])") ;
```

A RooPlot of "x"

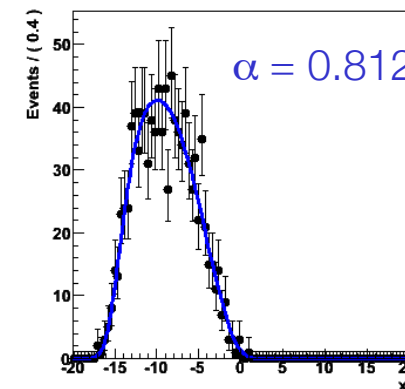


Histogram of hh_x_alpha



Fit to data

A RooPlot of "x"



- Three morphing operator classes available
 - `IntegralMorph` (Alex Read).
 - `MomentMorph` (Max Baak).
 - `PiecewiseInterpolation` (via HistFactory)

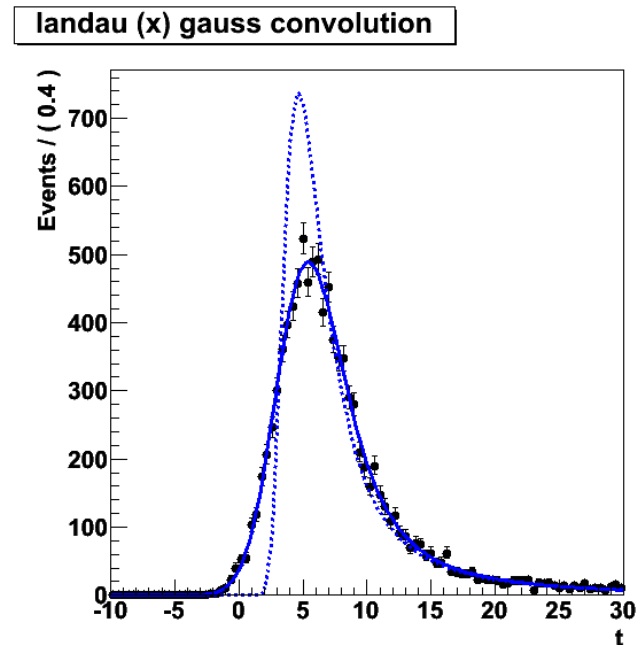
Powerful operators – Fourier convolution

- Convolve any two arbitrary pdfs with a 1-line expression

```
w.factory("Landau::L(x[-10,30],5,1)") :
w.factory("Gaussian::G(x,0,2)") ;

w::x.setBins("cache",10000) ; // FFT sampling density
w.factory("FCONV::LGf(x,L,G)") ; // FFT convolution
```

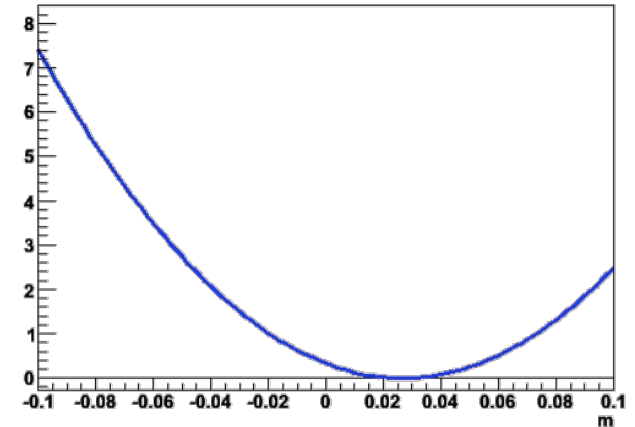
- Exploits power of FFTW package available via ROOT
 - Hand-tuned assembler code for time-critical parts
 - Amazingly fast: unbinned ML fit to 10.000 events take ~5 seconds!



Working with the likelihood function

- Plot the likelihood function versus a parameter

```
RooAbsReal* nll = w::model.createNLL(data) ;  
  
RooPlot* frame = w::param.frame() ;  
nll->plotOn(frame, ShiftToZero()) ;
```



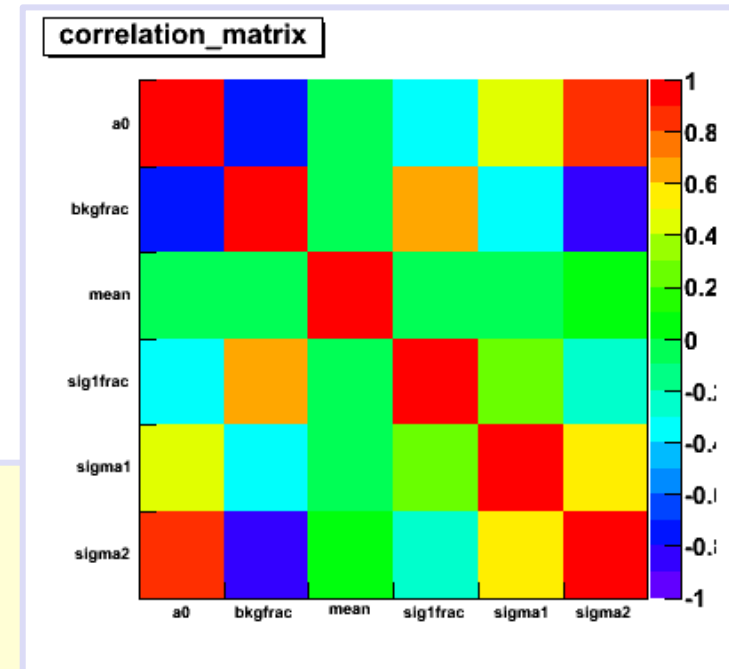
- Maximum Likelihood estimation of parameters and variance

```
RooMinimizer m(*nll) ;  
  
// ML Parameter estimation  
m.minimize("Minuit2", "migrad") ;  
  
// Variance estimation  
m.hesse() ;  
  
// Alternatively - all this in one line  
pdf->fitTo(*data) ;
```

Working with covariance and correlation matrices

- Detailed information on parameter and covariance estimates can be saved for detailed information

```
RoofMinimizer m(*nll) ;  
m.minimize("Minuit2","migrad") ;  
m.hesse() ;  
RooFitResult* r = m.save() ;  
  
// Visualize correlation matrix  
r->correlationHist->Draw("colz") ;  
  
// Extract correlation,covariance matrix  
TMatrixDSym cov = fr->covarianceMatrix() ;  
TMatrixDSym cov = fr->covarianceMatrix(a,b) ;
```



Use covariance matrices for correlated error propagation

- Can (as visual aid) propagate errors in covariance matrix of a fit result to a pdf projection

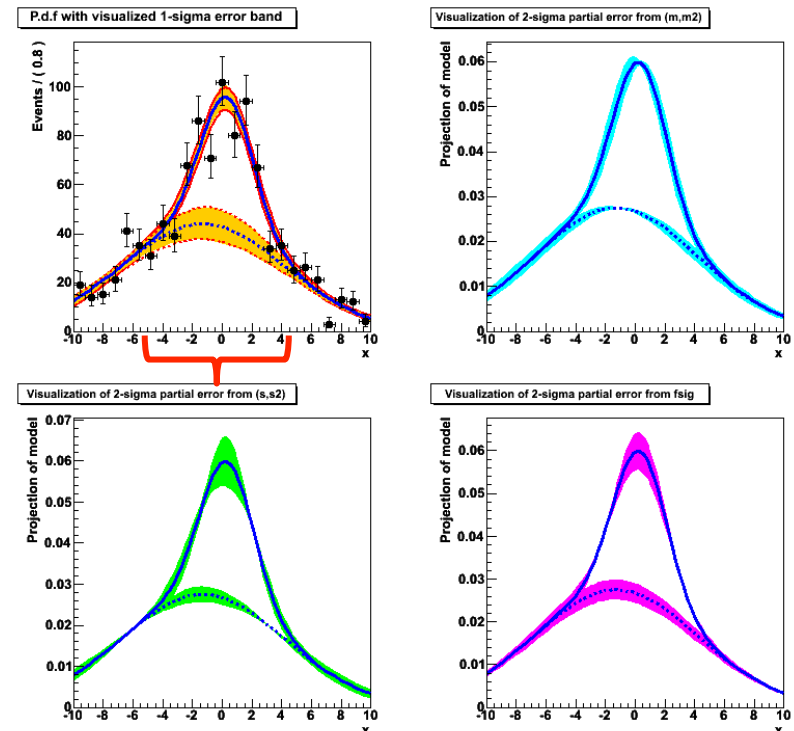
```
w::model.plotOn(frame, VisualizeError(*fitresult)) ;
w::model.plotOn(frame, VisualizeError(*fitresult, fsig)) ;
```

- Linear propagation on pdf projection $\Delta = \vec{E}V^{-1}\vec{E}$

- Propagated error can be calculated on arbitrary function
 - E.g fraction of events in signal range

```
RooAbsReal* fracSigRange =
  w::model.createIntegral(x,x,"sig") ;

Double_t err =
  fracSigRange.getPropagatedError(*fr) ;
```



Some RooFit practical examples – from start to end

- Signal + Background (analytical)

```
RooWorkspace w("w") ;
```

```
// Construct exponential background model  
w.factory("Exponential::bkg(x[10,100],alpha[-0.04,-0.1,-0])") ;
```

```
// Construct Gaussian signal model  
w.factory("Gaussian::sig(x,mean[40],width[3])") ;
```

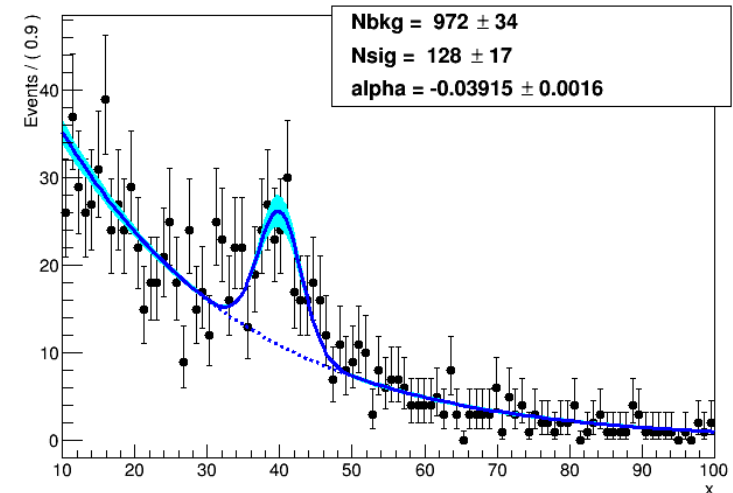
```
// Construct extended ML model of sum of signal and background  
w.factory("SUM::modelsum(Nsig[100,0,200]*sig,Nbkg[1000,0,2000]*bkg)") ;
```

```
// Generate a toy dataset (unbinned) from model, data sample size obtained from expected event count  
RooDataSet* d = w.pdf("modelsum")->generate(*w.var("x")) ;
```

```
// Fit model to toy data  
RooFitResult* r3 = w.pdf("modelsum")->fitTo(*d,Save()) ;
```

```
// Plot data  
RooPlot* frame = w.var("x")->frame() ;  
d->plotOn(frame) ;
```

```
// Plot model (background component separately) and visualization of uncertainties from fit  
w.pdf("modelsum")->plotOn(frame,VisualizeError(*r3)) ;  
w.pdf("modelsum")->plotOn(frame) ;  
w.pdf("modelsum")->plotOn(frame,Components("bkg"),LineStyle(kDashed)) ;  
w.pdf("modelsum")->paramOn(frame) ;  
frame->Draw() ;
```



Some RooFit practical examples – from start to end

- Two-dimensional signal: $f(x|y)*g(y)$

```
RooWorkspace w("w") ;

// Construct g(x|fy,0.5) where the mean of the gaussian
// is a polynomial fy=a0+a1*y
w.factory("PolyVar::fy(y[-5,5],{a0[-0.5,-5,5],a1[-0.5,-1,1]})") ;
w.factory("Gaussian::gx(x[-5,5],fy,sigma[0.5])") ;

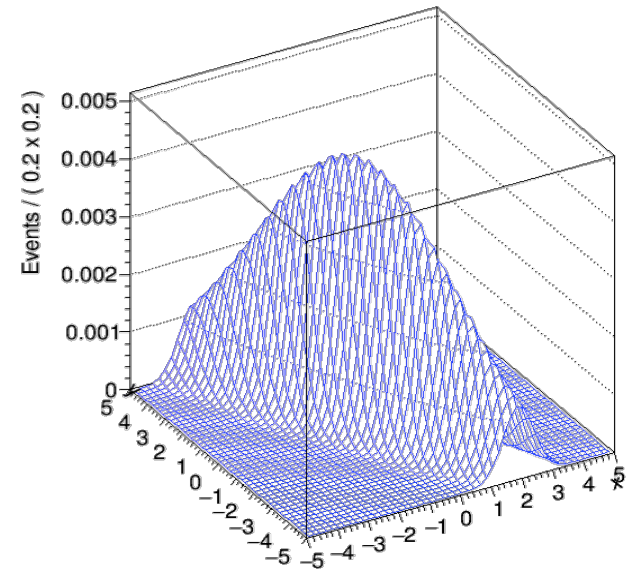
// Construct g(y)
w.factory("Gaussian::gy(y,0,3)") ;

// Construct the conditional product g(x|y)*g(y)
w.factory("PROD::model(gx|y,gy)") ;

// Generate 1000 events in x and y from model
RooDataSet *data = w.pdf("model")->generate(RooArgSet(*w.var("x"),*w.var("y")),10000) ;

// Plot x distribution of data and projection of model on x = Int(dy) model(x,y)
RooPlot* xframe = w.var("x")->frame() ;
data->plotOn(xframe) ;
w.pdf("model")->plotOn(xframe) ;

// Make two-dimensional plot in x vs y
TH1* hh_model = w.pdf("model")->createHistogram("hh_model",*w.var("x"),Binning(50),
                                                YVar(*w.var("y"),Binning(50))) ;
hh_model->SetLineColor(kBlue) ;
```



Some RooFit practical examples – from start to end

- **Signal + Background (templates)**
Method 1: Construct unit-normalized pdf from histograms
Model parameters are absolute event counts

```
RooWorkspace w("w") ;

// Import template histograms into workspace
w.import(*h_bkg,Rename("histo_bkg")) ;
w.import(*h_sig,Rename("histo_sig")) ;

// Construct sum of histogram-shaped templates
w.factory("SUM::modelsum(Nsig[100,0,200]*HistPdf::sig(x[10,100],histo_sig),
          Nbkg[1000,0,2000]*HistPdf::bkg(x,histo_bkg))") ;

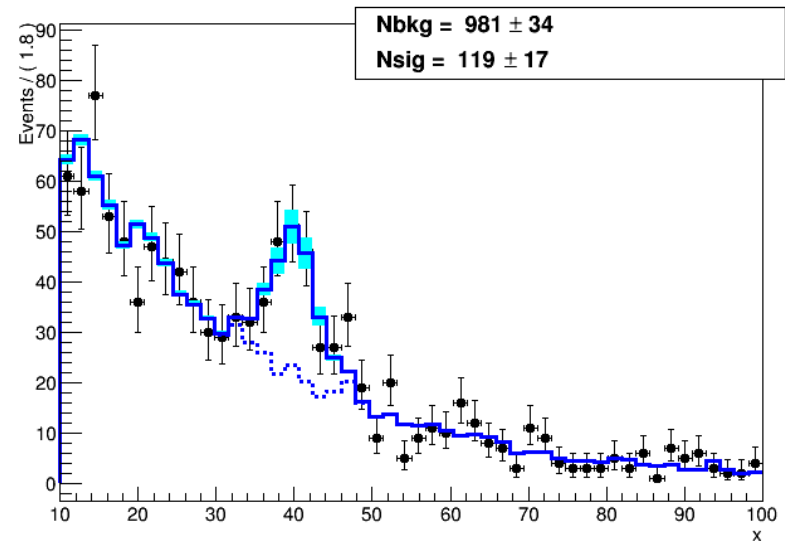
// Generate a toy dataset (unbinned) from model, data sample size obtained from expected event count
RooDataSet* d = w.pdf("modelsum")->generate(*w.var("x")) ;

// Fit model to toy data
RooFitResult* r3 = w.pdf("modelsum")->fitTo(*d,Save()) ;

// Plot data
RooPlot* frame = w.var("x")->frame() ;
d->plotOn(frame) ;

// Plot model (background component separately) and visualization of uncertainties from fit
w.pdf("modelsum")->plotOn(frame,VisualizeError(*r3)) ;
w.pdf("modelsum")->plotOn(frame) ;
w.pdf("modelsum")->plotOn(frame,Components("bkg"),LineStyle(kDashed)) ;
w.pdf("modelsum")->paramOn(frame) ;

frame->Draw() ;
```



Some RooFit practical examples – from start to end

- **Signal + Background (templates)**
Method 2: Construct event-count scaled pdf from histograms
Model parameters are scale factors relative histogram counts

```
RooWorkspace w("w") ;

// Import template histograms into workspace
w.import(*h_bkg,Rename("histo_bkg")) ;
w.import(*h_sig,Rename("histo_sig")) ;

// Construct sum of histogram-shaped templates
w.factory("ASUM::modelsum(kappa_sig[0.01,-0.1,1]*HistFunc::sig(x[10,100],histo_sig),
          kappa_bkg[0.1,-0.1,1]*HistFunc::bkg(x,histo_bkg))") ;

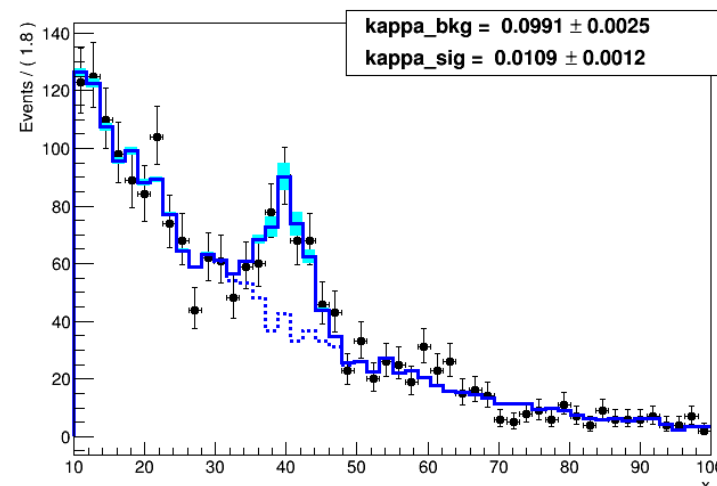
// Generate a toy dataset (unbinned) from model, data sample size obtained from expected event count
RooDataSet* d = w.pdf("modelsum")->generate(*w.var("x")) ;

// Fit model to toy data
RooFitResult* r3 = w.pdf("modelsum")->fitTo(*d,Save()) ;

// Plot data
RooPlot* frame = w.var("x")->frame() ;
d->plotOn(frame) ;

// Plot model (background component separately) and visualization of uncertainties from fit
w.pdf("modelsum")->plotOn(frame,VisualizeError(*r3)) ;
w.pdf("modelsum")->plotOn(frame) ;
w.pdf("modelsum")->plotOn(frame,Components("bkg"),LineStyle(kDashed)) ;
w.pdf("modelsum")->paramOn(frame) ;

frame->Draw() ;
```



Some RooFit practical examples – from start to end

- **Signal + Background (templates)**
With morphing shape parameter

```
Rooworkspace w("w") ;
```

```
// Import template histograms into workspace  
w.import(*h_bkg,Rename("histo_bkg")) ;  
w.import(*h_sig_lo,Rename("histo_sig_lo")) ;  
w.import(*h_sig_nom,Rename("histo_sig_nom")) ;  
w.import(*h_sig_hi,Rename("histo_sig_hi")) ;
```

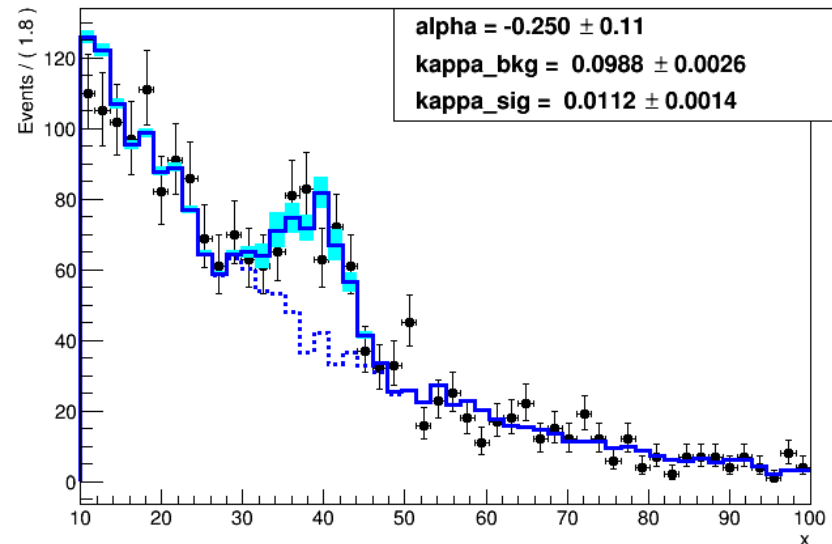
```
w.factory("PiecewiseInterpolation::sig_morph(HistFunc::sig_nom(x,histo_sig_nom),  
HistFunc::sig_lo(x,histo_sig_lo),  
HistFunc::sig_hi(x,histo_sig_hi),alpha[-5,5])") ;
```

```
// Construct sum of histogram-shaped templates  
w.factory("ASUM::modelsum(kappa_sig[0.01,-0.1,1]*sig_morph,  
kappa_bkg[0.1,-0.1,1]*HistFunc::bkg(x,histo_bkg))") ;
```

```
// Generate a toy dataset (unbinned) from model, data sample size obtained from expected event count  
RooDataSet* d = w.pdf("modelsum")->generate(*w.var("x")) ;
```

```
// Fit model to toy data  
RooFitResult* r3 = w.pdf("modelsum")->fitTo(*d,Save()) ;
```

```
// Plot data  
RooPlot* frame = w.var("x")->frame() ;  
d->plotOn(frame) ;
```

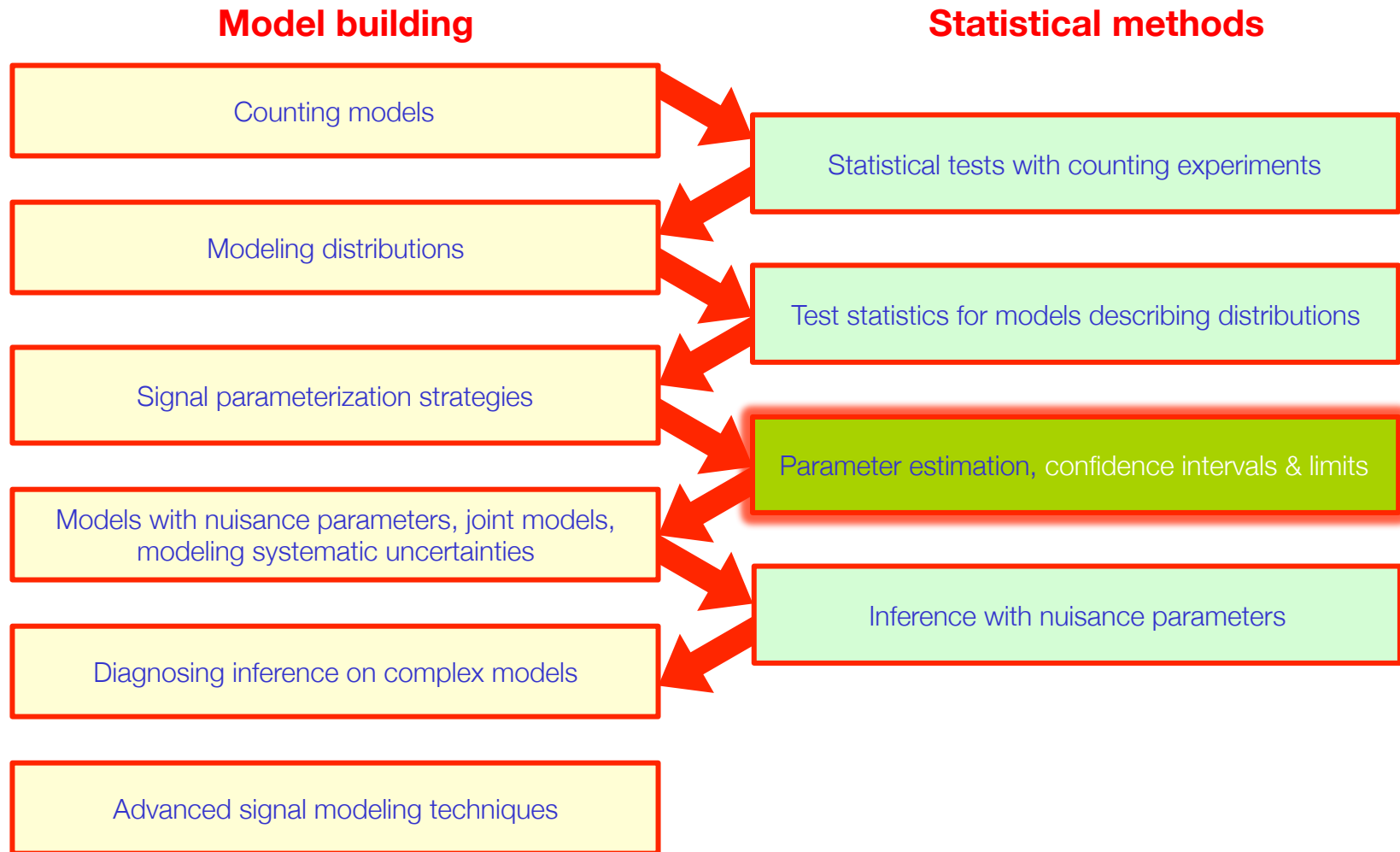


Statistical methods 3

Inference with parameters:
maximum likelihood, confidence
intervals, upper limits, likelihood ratio
and asymptotic formulae

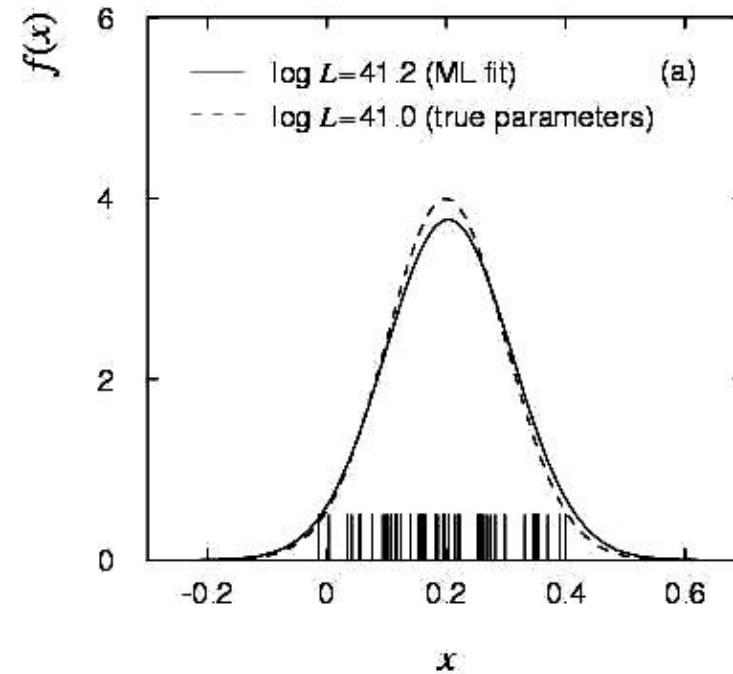
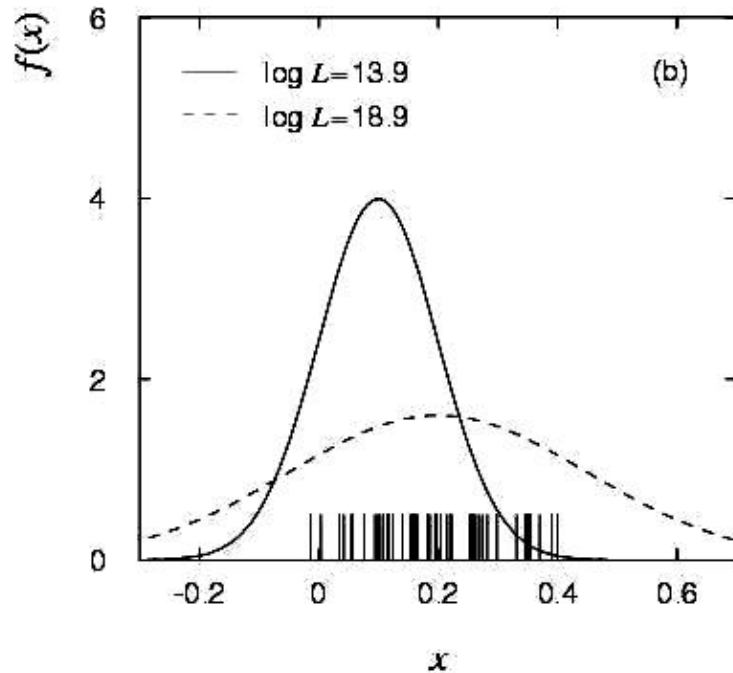
Roadmap of this course

- Start with basics, gradually build up to complexity



Parameter estimation using Maximum Likelihood


- Likelihood is high for values of p that result in distribution similar to data



- Define the **maximum likelihood (ML) estimator** to be the procedure that finds the parameter value for which the likelihood is maximal.

Parameter estimation – Maximum likelihood

- Practical estimation of maximum likelihood performed by minimizing the negative log-Likelihood

$$L(\vec{p}) = \prod_i f(\vec{x}_i; \vec{p})$$

$$-\ln L(\vec{p}) = -\sum_i \ln F(\vec{x}_i; \vec{p})$$

- Advantage of log-Likelihood is that contributions from events can be summed, rather than multiplied (computationally easier)
- In practice, find point where derivative of $-\log L$ is zero

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

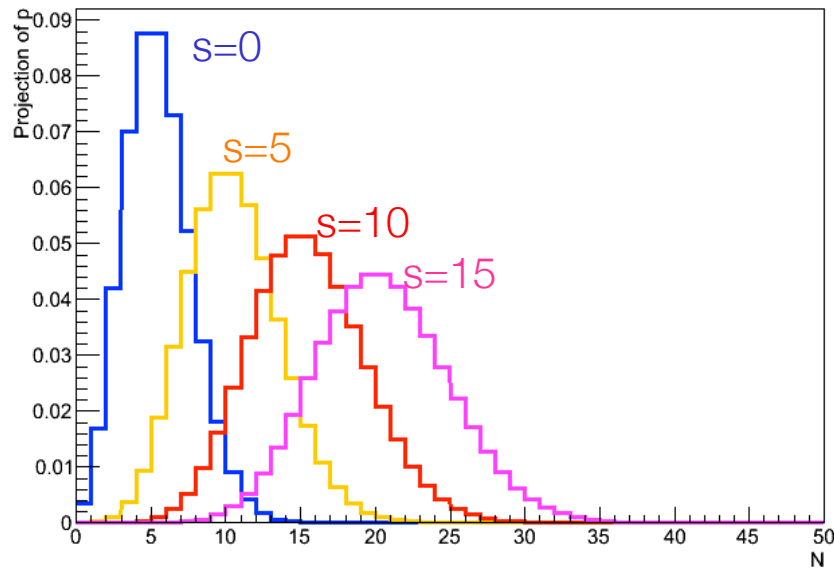
- Standard notation for ML estimation of p is \hat{p}

Example of Maximum Likelihood estimation

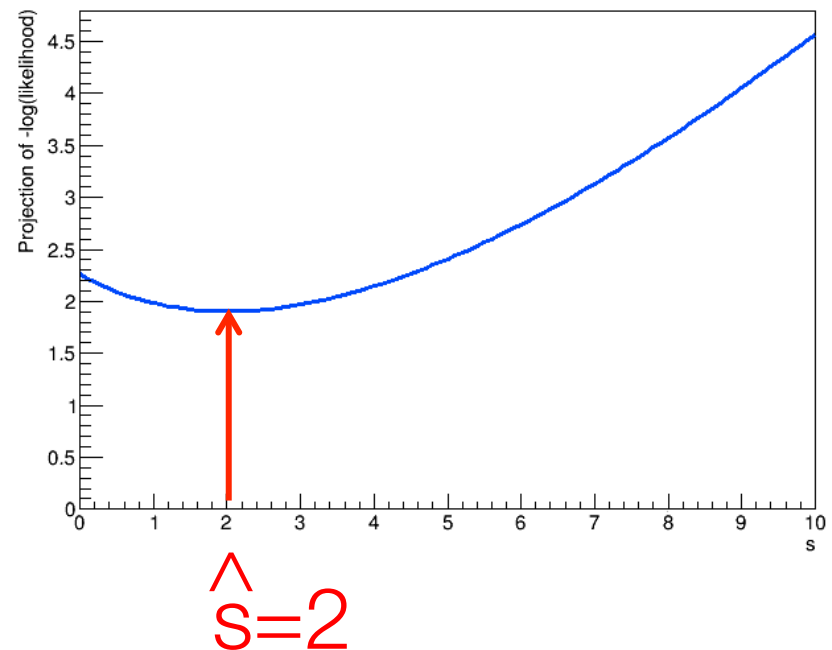
- Illustration of ML estimate on Poisson counting model

$$L(N | s) = \text{Poisson}(N | s + \tilde{b})$$

$-\log L(N|s)$ versus N [$s=0,5,10,15$]



$-\log L(N|s)$ versus s [$N=7$]



- Note that Poisson model is discrete in N , *but continuous in s !*

Properties of Maximum Likelihood estimators

- In general, Maximum Likelihood estimators are
 - Consistent (gives right answer for $N \rightarrow \infty$)
 - Mostly unbiased (bias $\propto 1/N$, may need to worry at small N)
 - Efficient for large N (you get the smallest possible error)
 - Invariant: (a transformation of parameters will Not change your answer, e.g. $(\hat{p})^2 = \widehat{(p^2)}$)
- MLE efficiency theorem: the MLE will be *unbiased and efficient* if an unbiased efficient estimator exists
 - Proof not discussed here
 - Of course this does not guarantee that any MLE is unbiased and efficient for any given problem

Relation between Likelihood and χ^2 estimators

- Properties of χ^2 estimator follow from properties of ML estimator using *Gaussian probability density functions*

$$F(x_i, y_i, \sigma_i; \vec{p}) = \prod_i \exp \left[- \left(\frac{y_i - f(x_i; \vec{p})}{\sigma_i} \right)^2 \right]$$

← Gaussian Probability Density Function in p for single measurement $y \pm \sigma$ from a predictive function $f(x|p)$



Take log,
Sum over all points (x_i, y_i, σ_i)

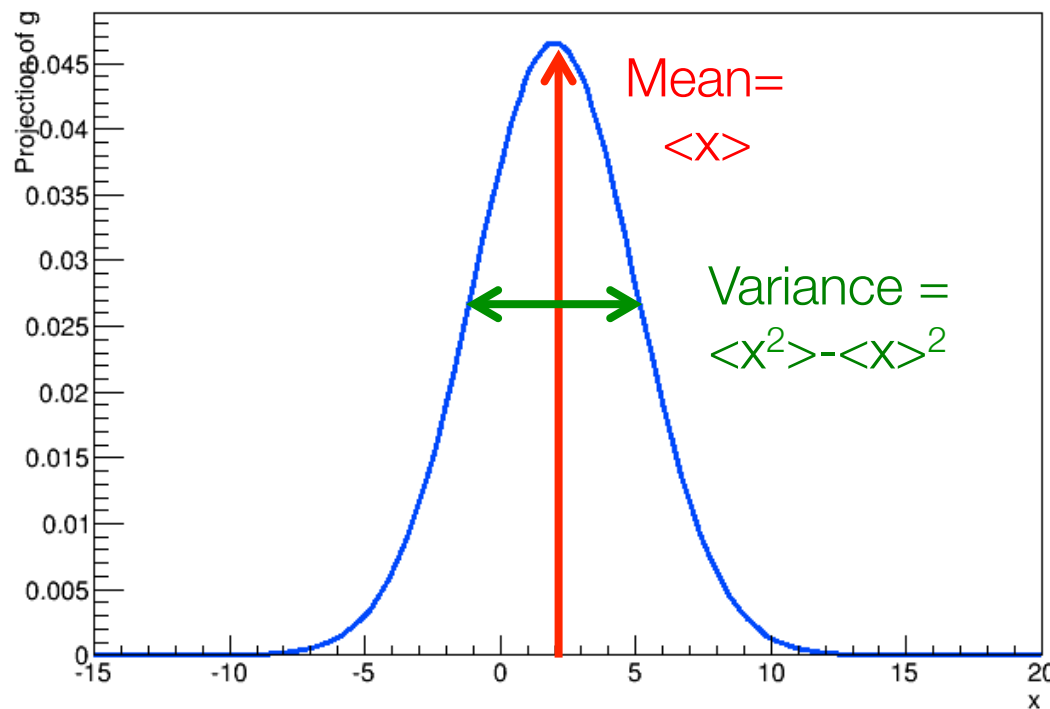
$$-\ln L(\vec{p}) = \frac{1}{2} \sum_i \left(\frac{y_i - f(x_i; \vec{p})}{\sigma_i} \right)^2 = \frac{1}{2} \chi^2$$

← The Likelihood function in p for given points $x_i(s_i)$ and function $f(x_i; p)$

- The χ^2 estimator follows from ML estimator, i.e it is
 - Efficient, consistent, bias $1/N$, invariant,
 - But only in the limit that the error **on x_i** is truly Gaussian

Estimating parameter variance

- Note that ‘uncertainty’ on a parameter estimate is an ambiguous statement
- Can either mean an **interval with a stated confidence or credible level (e.g. 68%)**, or simply assume it is the **square-root of the variance** of a distribution



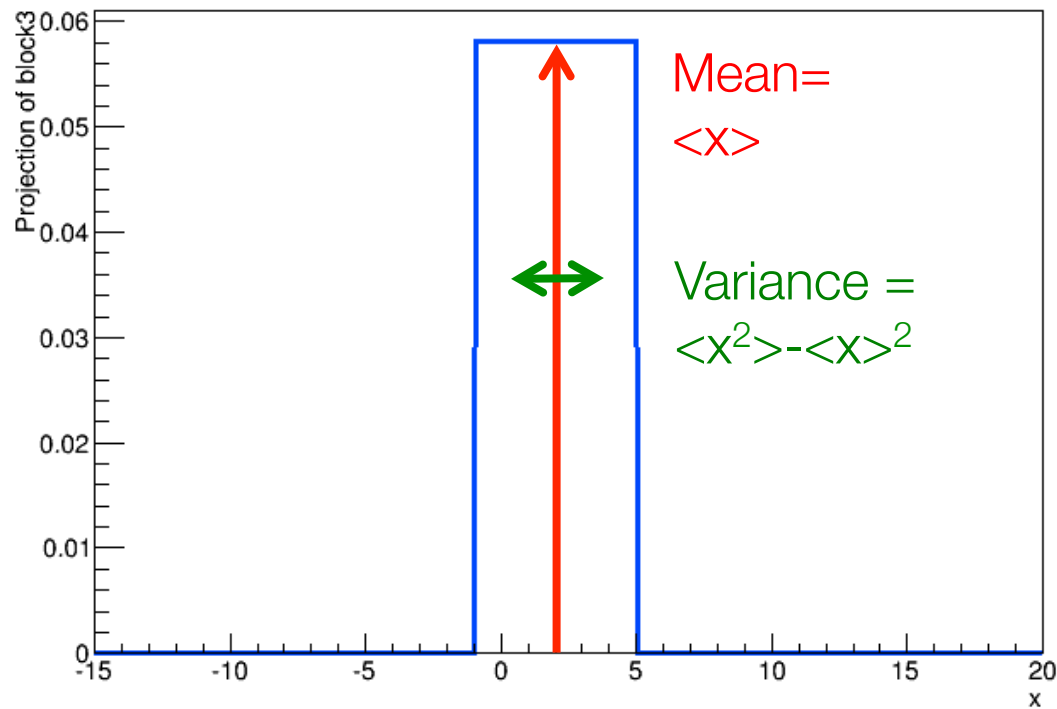
For a Gaussian distribution mean and variance map to parameters for *mean* and *sigma*²

and interval defined by \sqrt{V} contains 68% of the distribution (=‘1 sigma’ by definition)

Thus for Gaussian distributions all common definitions of ‘error’ work out to the same numeric value

Estimating parameter variance

- Note that ‘error’ or ‘uncertainty’ on a parameter estimate is an ambiguous statement
- Can either mean an **interval with a stated confidence or credible level (e.g. 68%)**, or simply assume it is the **square-root of the variance** of a distribution



For other distributions intervals by \sqrt{V} do not necessarily contain 68% of the distribution

Estimating variance on parameters

- Variance on of parameter can also be estimated from Likelihood using the variance estimator

$$\hat{\sigma}(p)^2 = \hat{V}(p) = \left(\frac{d^2 \ln L}{d^2 p} \right)^{-1}$$

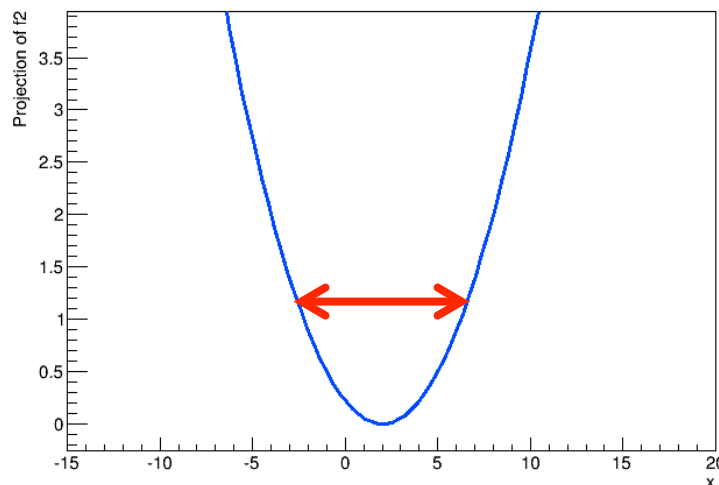
From Rao-Cramer-Frechet inequality

$$V(\hat{p}) \geq 1 + \frac{db}{dp} \left/ \left(\frac{d^2 \ln L}{d^2 p} \right) \right.$$

b = bias as function of p, inequality becomes equality in limit of efficient estimator

- Valid if estimator is **efficient** and **unbiased!**

- Illustration of Likelihood Variance estimate on a Gaussian distribution



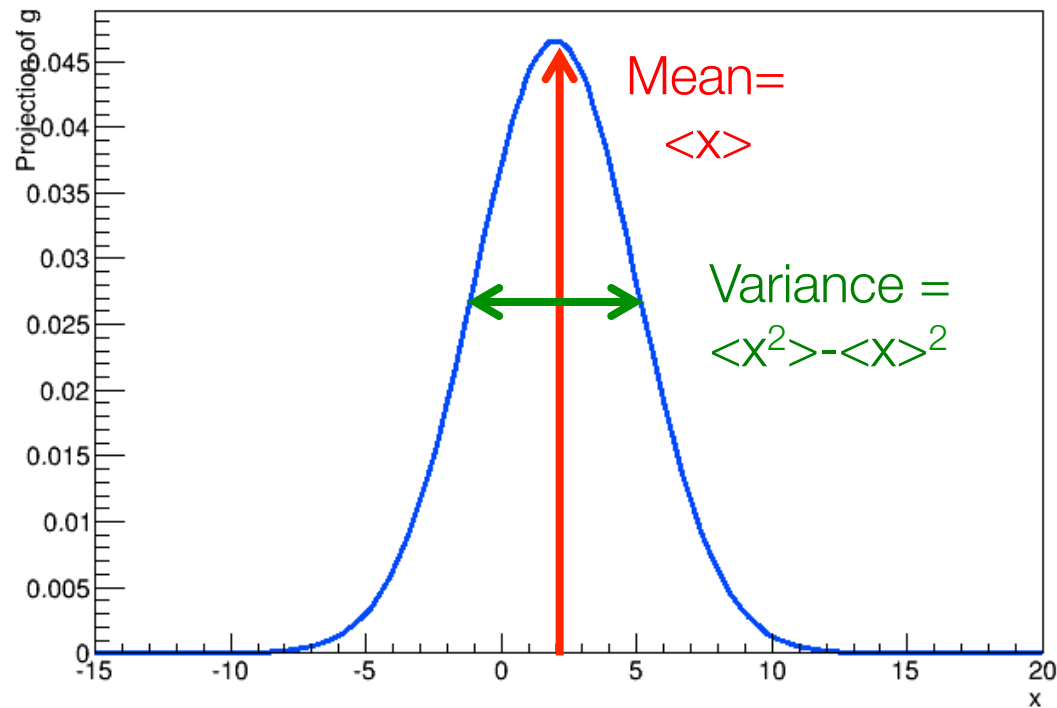
$$f(x | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2 \right]$$

$$\ln f(x | \mu, \sigma) = -\ln \sigma - \ln \sqrt{2\pi} + \frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2$$

$$\left. \frac{d \ln f}{d \sigma} \right|_{x=\mu} = \frac{-1}{\sigma} \Rightarrow \left. \frac{d^2 \ln f}{d^2 \sigma} \right|_{x=\mu} = \frac{1}{\sigma^2}$$

Bayesian parameter estimation

- Bayesian parameter estimate is the posterior mean
- Bayesian variance is the posterior variance



$$\hat{\mu} = \int \mu P(\mu | N) d\mu$$

$$\hat{V} = \int (\hat{\mu} - \mu)^2 P(\mu | N) d\mu$$

What can we do with composite hypothesis

- With simple hypotheses – inference is restricted to making statements about $P(D|\text{hypo})$ or $P(\text{hypo}|D)$
- With composite hypotheses – many more options

- **1 Parameter estimation and variance estimation**

- What is value of s for which the observed data is most probable?
 - What is the variance (std deviation squared) in the estimate of s ?
- } $s=5.5 \pm 1.3$

- **2 Confidence intervals**

- Statements about model parameters using frequentist concept of probability
- $s < 12.7$ at 95% confidence level
- $4.5 < s < 6.8$ at 68% confidence level

- **3 Bayesian credible intervals**

- Bayesian statements about model parameters
- $s < 12.7$ at 95% credibility