Data Analysis

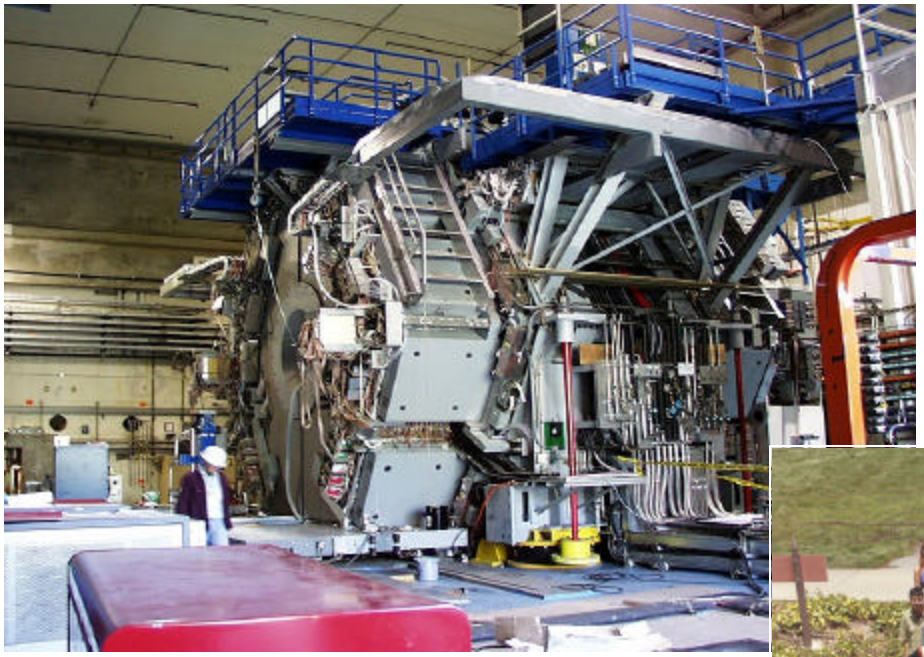
Wouter Verkerke
(University of California Santa Barbara / NIKHEF)

Course Overview

- Basic statistics – 24 pages
- Reducing backgrounds – 36 pages
- Estimation and fitting – 52 pages
- Significance, probability – 25 pages
- Systematic uncertainties – 12 pages

Speaker introduction

The BaBar Detector



Working for the **BaBar** experiment since 1998 -- *CP Violation in the B meson system*

The BaBar collaboration in 1999 →

Occasionally, I will take some examples from B physics, no material in this course is specifically tied to any experiment (including BaBar)

Basic Statistics

— Mean, Variance, Standard Deviation
— Gaussian Standard Deviation
— Covariance, correlations
— Basic distributions – Binomial, Poisson, Gaussian
— Central Limit Theorem
— Error propagation

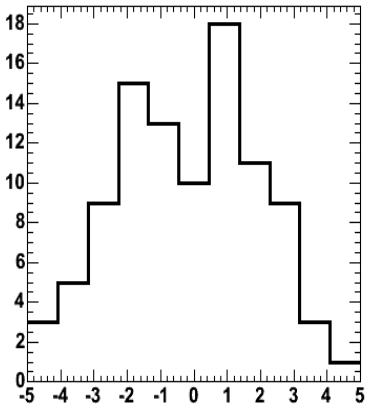
Wouter Verkerke, UCSB

Data – types of data

- **Qualitative** (numeric) vs **Quantitative** (non-numeric)

Not suitable for mathematical treatment

Discrete (Integers)



'Histograms'

Continuous (Reals)

{ 5.6354
7.3625
8.1635
9.3634
1.3846
0.2847
1.4763 }

'N-tuples'

Binning

Describing your data – the Average

- Given a set of *unbinned* data (measurements)

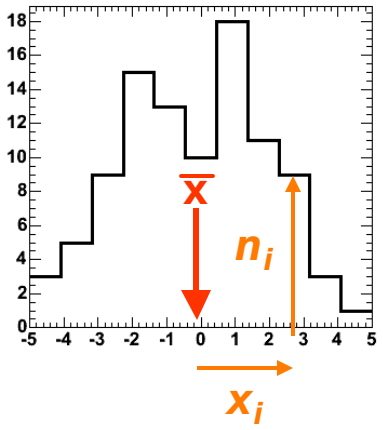
  $\{ x_1, x_2, \ldots, x_N \}$

  then the mean value of x is

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

- For *binned* data

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} n_i x_i$$



  - where n_i is bin count and x_i is bin center
  - Unbinned average more accurate due to rounding

Describing your data – Spread

- *Variance V(x)* of x expresses how much ***x*** is liable to vary from its mean value $\bar{x}$

$$
\begin{aligned}
V(x) &= \frac{1}{N}\sum_i (x_i - \bar{x})^2 \\
&= \frac{1}{N}\sum_i (x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\
&= \frac{1}{N}\sum_i x_i^2 - \frac{1}{N}2\bar{x}\sum_i x_i + \frac{1}{N}\bar{x}^2\sum_i 1) \\
&= \overline{x^2} - 2\bar{x}^2 + \bar{x}^2 \\
&= \overline{x^2} - \bar{x}^2
\end{aligned}
$$

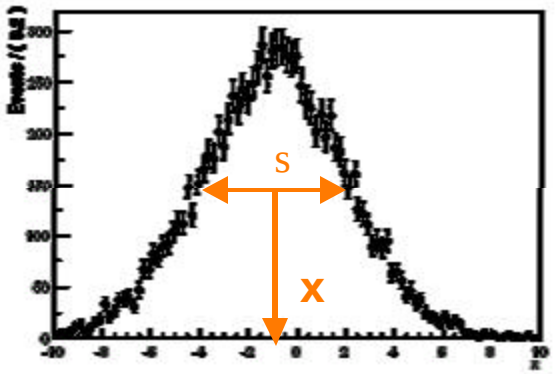
- ***Standard deviation*** $s \equiv \sqrt{V(x)} = \sqrt{\overline{x^2} - \bar{x}^2}$

Different definitions of the Standard Deviation

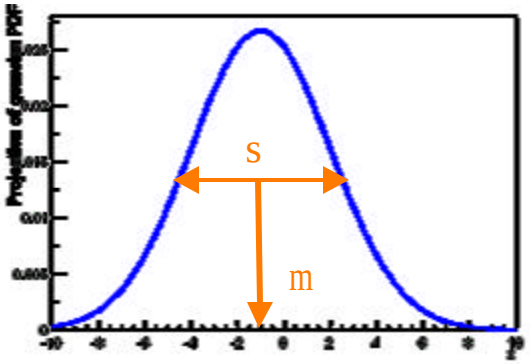
$$s = \sqrt{\frac{1}{N}\sum_i (x^2 - \bar{x})^2}$$ is the S.D. of the ***data sample***

- Presumably our data was taken from a parent distributions which has mean μ and S.F. σ

Data Sample



Parent Distribution
(from which data sample was drawn)



$\bar{x}$ – mean of our sample

σ – S.D. of our sample

$\Leftrightarrow$

μ – mean of our parent dist

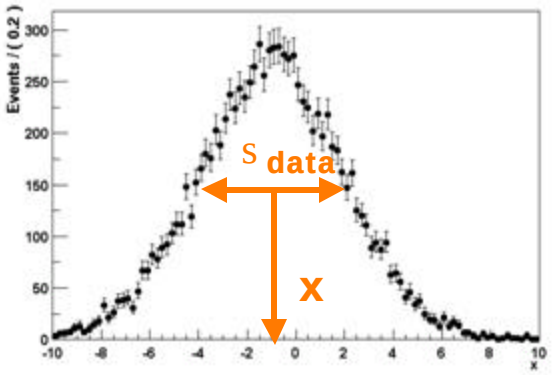
σ – S.D. of our parent dist

Beware Notational Confusion!

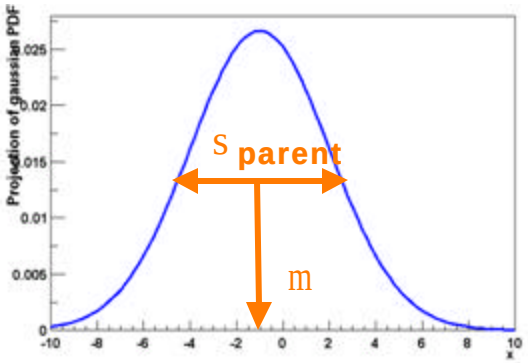
Different definitions of the Standard Deviation

- Which definition of σ you use, σ_{data} or σ_{parent}, is matter of preference, but be clear which one you mean!

Data Sample



Parent Distribution (from which data sample was drawn)



- In addition, you can get an unbiased estimate of s_{parent} from a given data sample using

$$\hat{s}_{\text{parent}} = \sqrt{\frac{1}{N-1}\sum_i (x^2 - \bar{x})^2} = \hat{s}_{\text{data}}\sqrt{\frac{N}{N-1}} \qquad \left(s_{\text{data}} = \sqrt{\frac{1}{N}\sum_i (x^2 - \bar{x})^2} \right)$$

More than one variable

- Given *2 variables x,y* and a dataset consisting of pairs of numbers

$$\{ (x_1,y_1),\ (x_2,y_2),\ \ldots\ (x_N,y_N) \}$$

- Definition of $\overline{x}$, $\overline{y}$, σ_x, σ_y as usual
- In addition, any *dependence between x,y* described by the ***co***variance

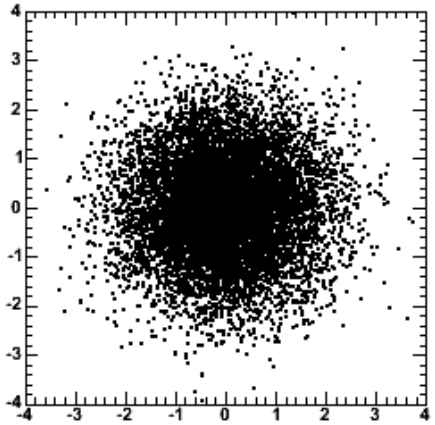
$$\begin{aligned} \mathrm{cov}(x,y) &= \frac{1}{N}\sum_i (x_i-\bar{x})(y_i-\bar{y}) \\ &= \overline{(x-\bar{x})(y-\bar{y})} \\ &= \overline{xy} - \bar{x}\,\bar{y} \end{aligned}$$

(has dimension D(x)D(y))

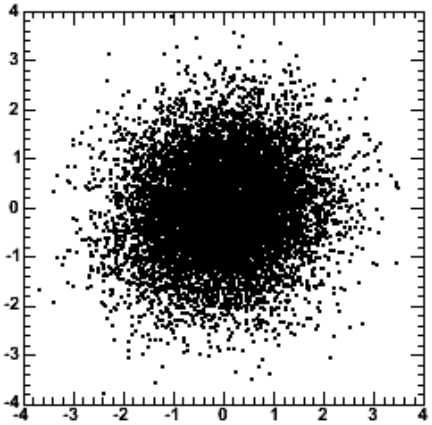
- The dimensionless ***correlation*** coefficient is defined as $r = \dfrac{\mathrm{cov}(x,y)}{s_x s_y} \in [-1,+1]$

Visualization of correlation

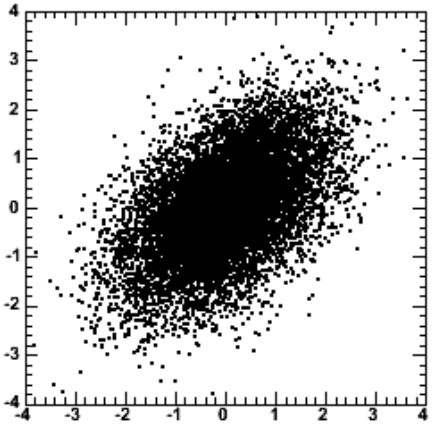
$r = 0$



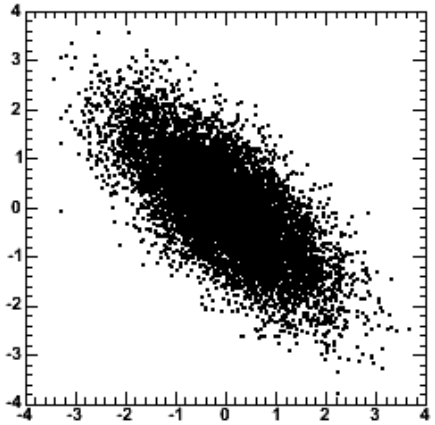
$r = 0.1$



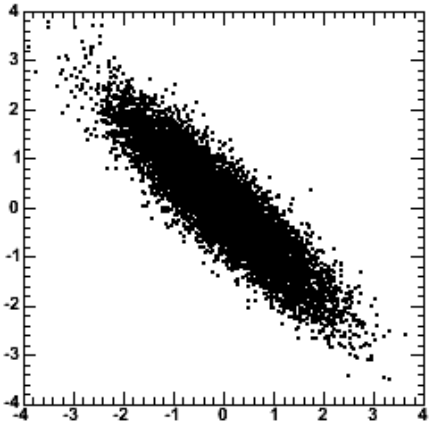
$r = 0.5$



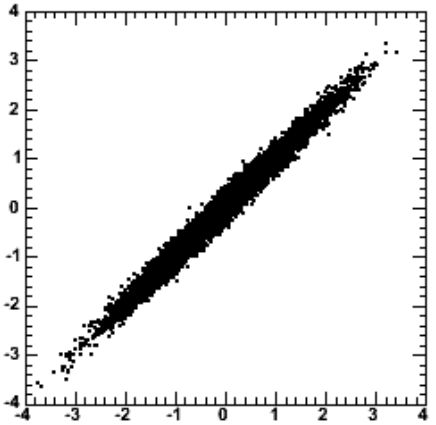
$r = -0.7$



$r = -0.9$



$r = 0.99$



Correlation & covariance in >2 variables

- Concept of covariance, correlation is easily extended to arbitrary number of variables

$$\mathrm{cov}(x_{(i)}, x_{(j)}) = \overline{x_{(i)} x_{(j)}} - \overline{x}_{(i)} \overline{x}_{(j)}$$

- so that $V_{ij} = \mathrm{cov}(x_{(i)}, x_{(j)})$ takes the form of a *n x n symmetric matrix*

- This is called the ***covariance matrix***, or ***error matrix***
- Similarly the correlation matrix becomes

$$r_{ij} = \frac{\mathrm{cov}(x_{(i)}, x_{(j)})}{s_{(i)} s_{(j)}} \longrightarrow V_{ij} = r_{ij} s_i s_j$$

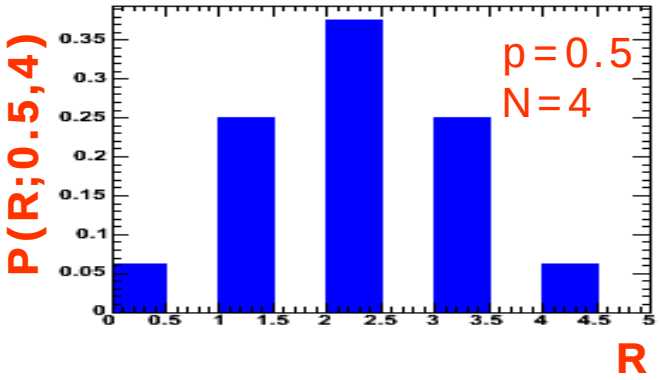
Basic Distributions – The binomial distribution

- Simple 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

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

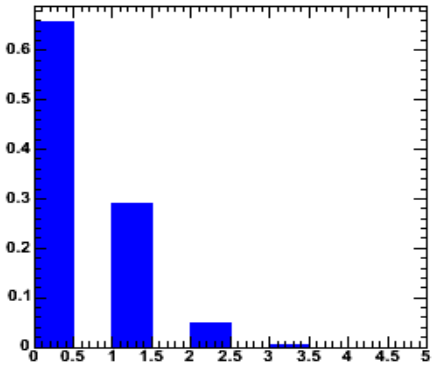


Binomial distribution

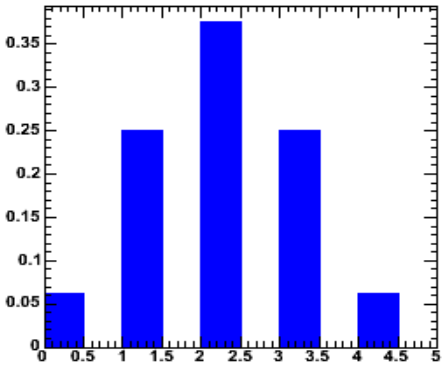
Properties of the binomial distribution

- Mean: $\langle r \rangle = n \cdot p$
- Variance: $V(r) = np(1-p) \quad \Rightarrow \quad s = \sqrt{np(1-p)}$

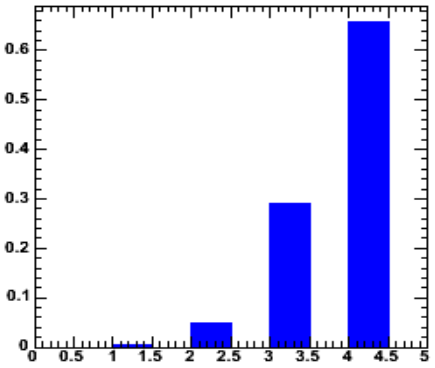
p=0.1, N=4



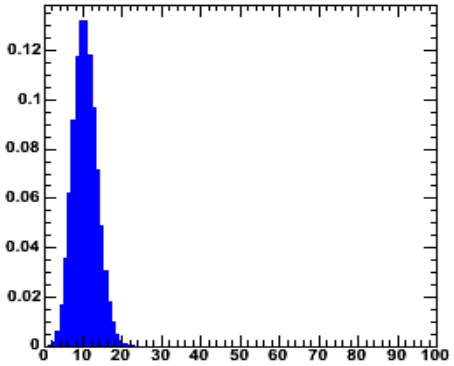
p=0.5, N=4



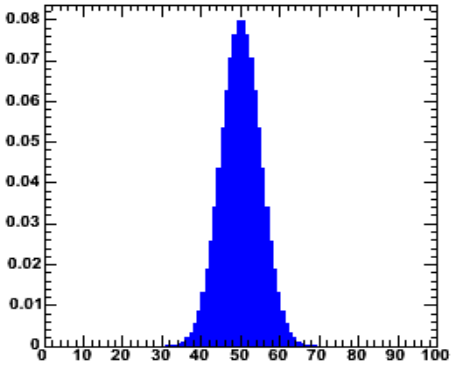
p=0.9, N=4



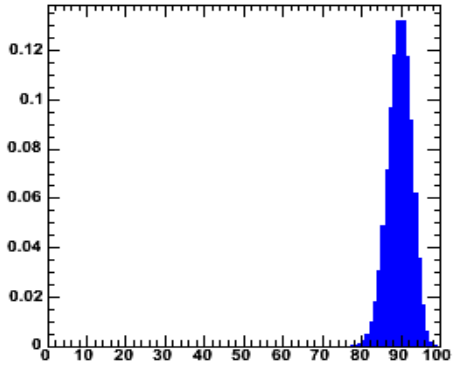
p=0.1, N=1000



p=0.5, N=1000



p=0.9, N=1000



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 to we expect in measurement over fixed amount of time?
  - Divide time interval λ in n finite chunks,
  - Take binomial formula with p=λ/n and let n→∞

$$P(r; l\ / n, n) = \frac{l^r}{n^r}(1-\frac{l}{n})^{n-r}\ \frac{n!}{r!(n-r)!}$$

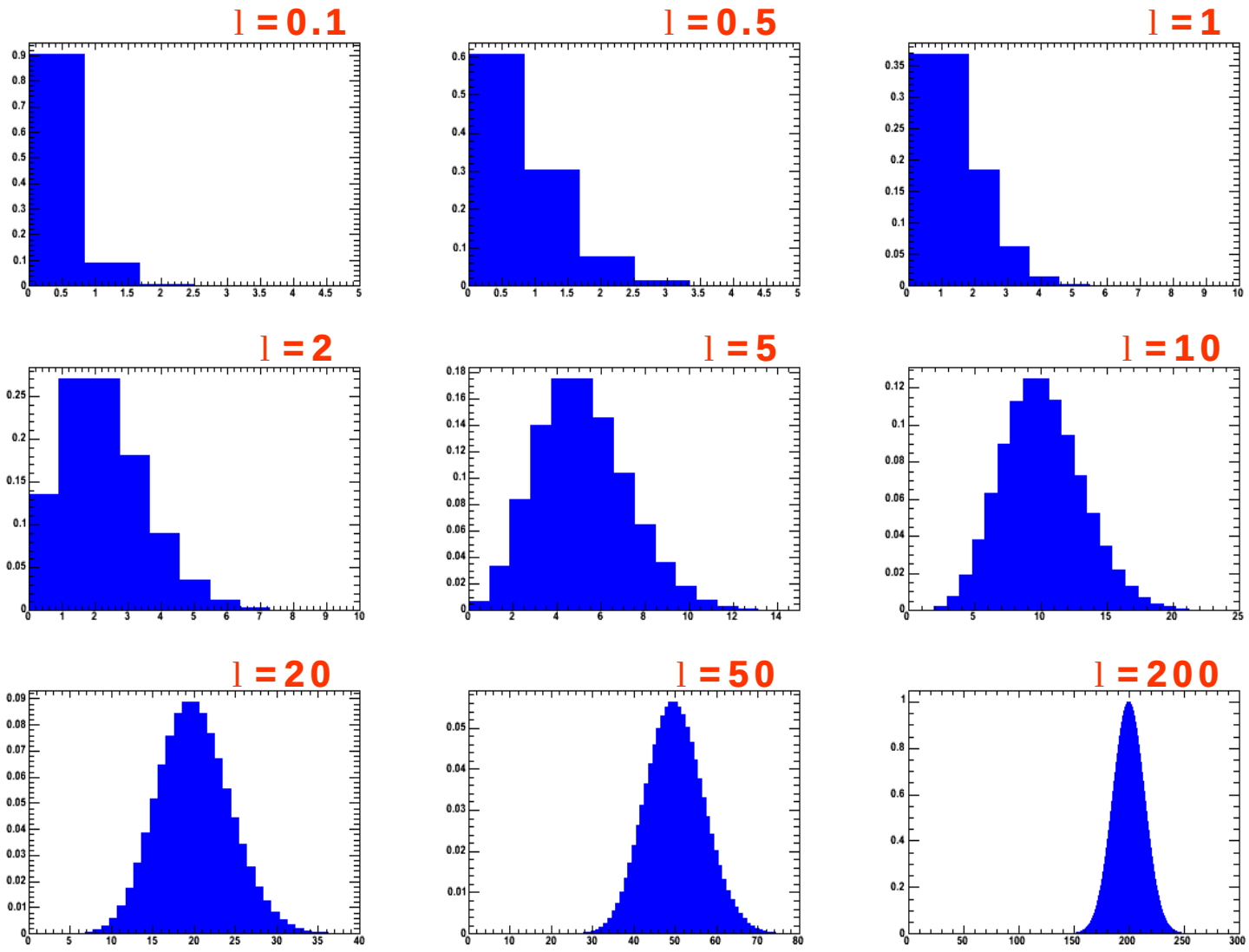
$$\lim_{n\to\infty} \frac{n!}{r!(n-r)!} = n^r,$$

$$\lim_{n\to\infty} (1-\frac{l}{n})^{n-r} = e^{-l}$$

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

←**Poisson distribution**

Properties of the Poisson distribution



More properties of the Poisson distribution $P(r;l)=\frac{e^{-l}\,l^{r}}{r!}$

- Mean, variance: $\langle r\rangle = l$

$$V(r) = l \quad \Rightarrow \quad s = \sqrt{l}$$

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

$$\begin{aligned}
P(r) &= \sum_{r_A=0}^{r} P(r_A;l_A)P(r-r_A;l_B) \\
&= e^{-l_A}e^{-l_B}\sum \frac{l_A^{r_A}\,l_B^{r-r_A}}{r_A!(r-r_A)!} \\
&= e^{-(l_A+l_B)}\frac{(l_A+l_B)^r}{r!}\sum_{r_{A=0}}^{r}\frac{r!}{(r-r_A)!}\left(\frac{l_A}{l_A+l_B}\right)^{r_A}\left(\frac{l_B}{l_A+l_B}\right)^{r-r_A} \\
&= e^{-(l_A+l_B)}\frac{(l_A+l_B)^r}{r!}\left(\frac{l_A}{l_A+l_B}+\frac{l_B}{l_A+l_B}\right)^{r} \\
&= e^{-(l_A+l_B)}\frac{(l_A+l_B)^r}{r!}
\end{aligned}$$

Basic Distributions – The Gaussian distribution

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

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

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

$$\ln(P(r;l)) = -l + r\ln l - (r\ln r - r) - \ln\sqrt{2pr}$$

$$= -l + r\left[\ln l - \ln\left(l\left(1+\frac{x}{l}\right)\right)\right] + (l+x) - \ln\sqrt{2pl}$$

$\ln(1+z) \approx z - z^2/2$

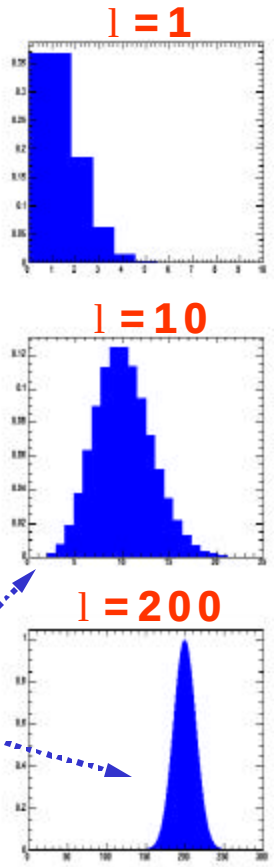
$$\approx x - (l-x)\left(\frac{x}{l} + \frac{x^2}{2l^2}\right) - \ln(2pl)$$

$$\approx \frac{-x^2}{2l} - \ln(2pl)$$

Take exp

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

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



Properties of the Gaussian distribution

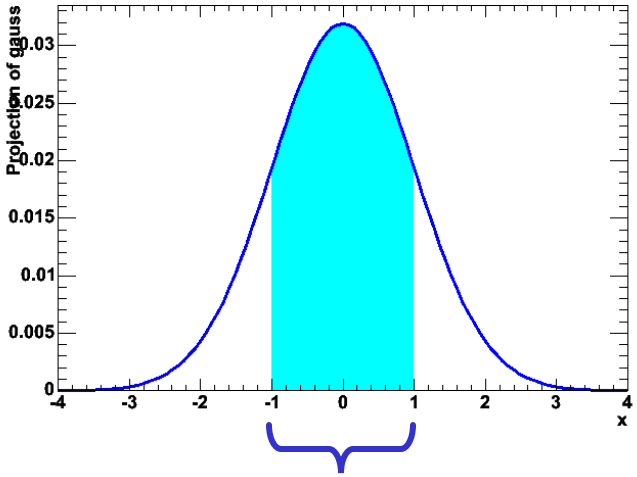
$$P(x;m,s) = \frac{1}{\sqrt{2ps}} e^{-(x-m)^2/2s^2}$$

- *Mean* and *Variance*

$$\langle x \rangle = \int_{-\infty}^{+\infty} xP(x;m,s)dx = m$$

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

$$s = s$$



- Integrals of Gaussian

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

Errors

- Doing an experiment → making measurements
- Measurements not perfect → imperfection quantified in resolution or error
- Common language to quote errors
  - Gaussian standard deviation = sqrt(V(x))
  - 68% probability that true values is within quoted errors

    [NB: 68% interpretation relies strictly on Gaussian sampling distribution, which is not always the case, more on this later]
- Errors are usually Gaussian if they quantify a result that is based on many independent measurements

The Gaussian as 'Normal distribution'

- Why are errors usually 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

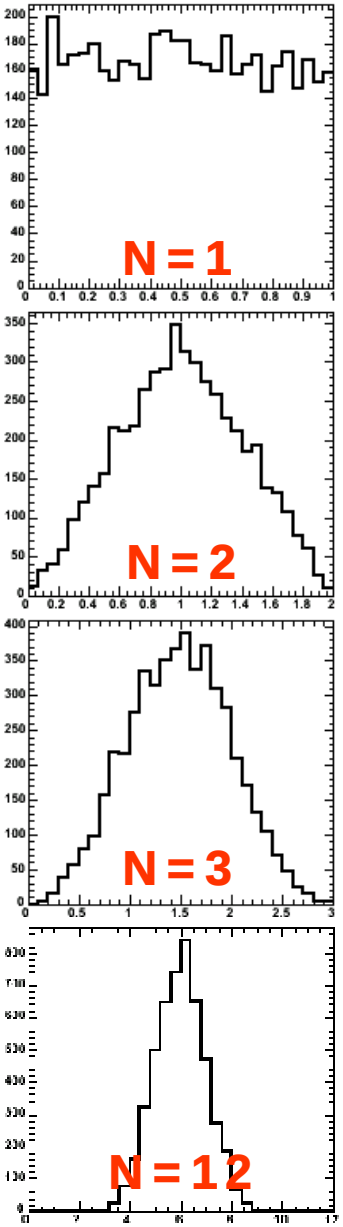
    (a) has expectation value $\langle X \rangle = \sum_i m_i$

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

    (c) becomes Gaussian as N → ¥

  - *Small print: tails converge very slowly in CLT, be careful in assuming Gaussian shape beyond 2s*

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

Central Limit Theorem – repeated measurements

- Common case 1 : Repeated identical measurements
  i.e. $\mu_i = \mu, \sigma_i = \sigma$ for all *i*

C.L.T

$$\langle X \rangle = \sum_i m_i = Nm \quad \Rightarrow \quad \langle \bar{x} \rangle = \frac{X}{N} = m$$

$$V(\bar{x}) = \sum_i V_i(\bar{x}) = \frac{1}{N^2} \sum_i V_i(X) = \frac{N s^2}{N^2} = \frac{s^2}{N}$$

$$s(\bar{x}) = \frac{s}{\sqrt{N}}$$

← Famous sqrt(N) law

Central Limit Theorem – repeated measurements

- Common case 2 : Repeated measurements with identical means but different errors (i.e weighted measurements, $\mu_i = \mu$)

$$\bar{x} = \frac{\sum x_i / s_i^2}{\sum 1/s_i^2}$$

Weighted average

$$V(\bar{x}) = \frac{1}{\sum 1/s_i^2} \Rightarrow s(\bar{x}) = \frac{1}{\sqrt{\sum 1/s_i^2}}$$

'Sum-of-weights' formula for error on weighted measurements

Error propagation – one variable

- Suppose we have $f(x) = ax + b$

- How do you calculate V(f) from V(x)?

$$
\begin{aligned}
V(f) &= \langle f^2 \rangle - \langle f \rangle^2 \\
&= \langle (ax+b)^2 \rangle - \langle ax+b \rangle^2 \\
&= a^2 \langle x^2 \rangle + 2ab\langle x \rangle + b^2 - a\langle x \rangle^2 - 2ab\langle x \rangle - b^2 \\
&= a^2 \left(\langle x^2 \rangle - \langle x \rangle^2 \right) \\
&= a^2 V(x) \qquad \leftarrow \textbf{i.e. } s_f = |a| s_x
\end{aligned}
$$

- More general: $V(f) = \left(\frac{df}{dx} \right)^2 V(x) \quad ; \quad s_f = \left| \frac{df}{dx} \right| s_x$

  – But only valid if *linear approximation is good in range of error*

Error Propagation – Summing 2 variables

- Consider $f = ax + by + c$

$$V(f) = a^2\left(\langle x^2\rangle - \langle x\rangle^2\right) + b^2\left(\langle y^2\rangle - \langle y\rangle^2\right) + 2ab\left(\langle xy\rangle - \langle x\rangle\langle y\rangle\right)$$

$$= a^2V(x) + b^2V(y) + 2ab\,\mathrm{cov}(x,y)$$

Familiar 'add errors in quadrature'
only valid in absence of correlations,
i.e. cov(x,y)=0

- More general

$$V(f) = \left(\frac{df}{dx}\right)^2 V(x) + \left(\frac{df}{dy}\right)^2 V(y) + 2\left(\frac{df}{dx}\right)\left(\frac{df}{dy}\right)\mathrm{cov}(x,y)$$

$$s_f^2 = \left(\frac{df}{dx}\right)^2 s_x^2 + \left(\frac{df}{dy}\right)^2 s_y^2 + 2\left(\frac{df}{dx}\right)\left(\frac{df}{dy}\right) r\, s_x s_y$$

But only valid if *linear approximation is good in range of error*

The correlation coefficient r [-1,+1] is 0 if x,y uncorrelated

Error propagation – multiplying, dividing 2 variables

- Now consider $f = x \cdot y$

$$V(f) = y^2 V(x) + x^2 V(y)$$

(math omitted)

$$\left(\frac{s_f}{f}\right)^2 = \left(\frac{s_x}{x}\right)^2 + \left(\frac{s_y}{y}\right)^2$$

  - Result similar for $f = x / y$

- Other useful formulas

$$\frac{s_{1/x}}{1/x} = \frac{s_x}{x} \quad ; \quad s_{\ln(x)} = \frac{s_x}{x}$$

Relative error on x,1/x is the same

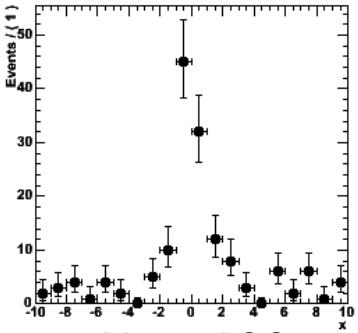
Error on log is just fractional error

Dealing with backgrounds

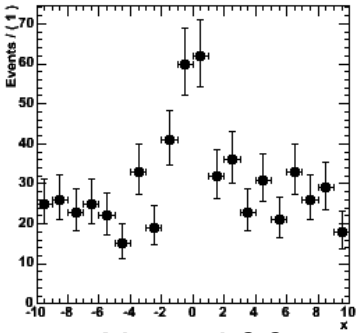
- Comparing discriminating variables
- Choosing the optimal cut
- Working in more than one dimension
- Approximating the optimal discriminant
- Techniques: Principal component analysis,
  Fisher Discriminant, Neural Network,
  Probability Density Estimate, Empirical Modeling

Backgrounds – Analysis strategy

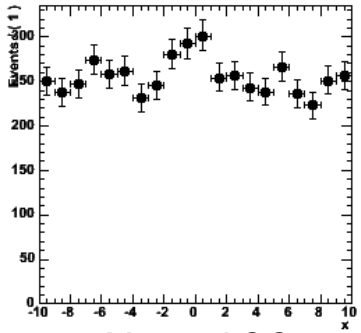
- Reducing backgrounds in a central theme in most HEP experiments and HEP data analyses
- For statistical analysis, problems introduced by background are two-fold
  1) Need to correct results for presence of background
     'subtract background' or 'include in fit'
  2) It reduces the significance of the measurement,
     10 events on top 1000 background events are less compelling evidence of any new particle than 10 events on top of 2 background events



$N_{sig}=100$
$N_{bkg}=50$



$N_{sig}=100$
$N_{bkg}=500$



$N_{sig}=100$
$N_{bkg}=5000$

Analysis strategy – General structure

- General strategy for data analysis in presence of background

1) Reduce backgrounds: 'Apply cuts'

- Exploiting information from your experiment to select a subset of events with less background

2) Account for remaining backgrounds: 'Fit the data'

- Developing procedures to incorporate uncertainty due to background into error on final result

Boundary between cutting and fitting is quite vague

3) Compute statistical significance of your result: 'Claim your signal (or not)'

- State your result in terms of absolute probabilities, e.g. 'the probability that background fakes my Higgs signal is less than 5×10^{-6}'

Analysis strategy – General structure

- General strategy for data analysis in presence of background

1) Reducing backgrounds: 'Apply cuts'

- Exploiting information from your experiment to select a subset of events with less background

2) Accounting for remaini… …ckgrounds:

- Developing procedures … to background into er…

3) Summarize statisti… result: 'Clai…

- State your result in ter… 'the probability that ba… is less than 5×10^{-6}'

We will now focus first on event selection techniques that reduce background:

how to find a set of criteria that reduces background a lot, signal as little as possible

Intermezzo – Role of simulation in HEP data analysis

- Simulation is an essential and pervasive aspects of all analysis step in HEP, e.g.

1) Reducing backgrounds: 'Apply cuts'

Samples of simulated events help you to understand the efficiency of your proposed cuts on signal and background and to determine the 'optimal' cut

2) Accounting for remaining backgrounds 'Fit the data'

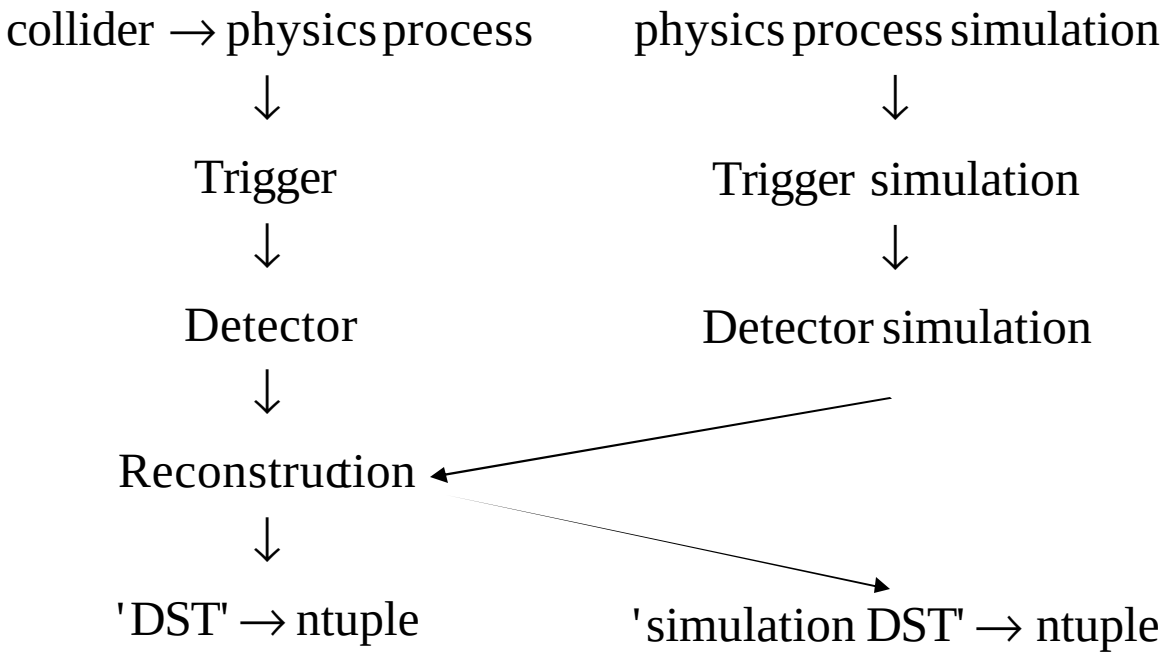
Simulation helps you to understand the behavior of your fit, explore the validity of functions and assumptions

3) Summarize statistical significance of your result: 'Claim your signal'

Simulation helps you to understand the robustness and validity of the statistical procedures that you have used

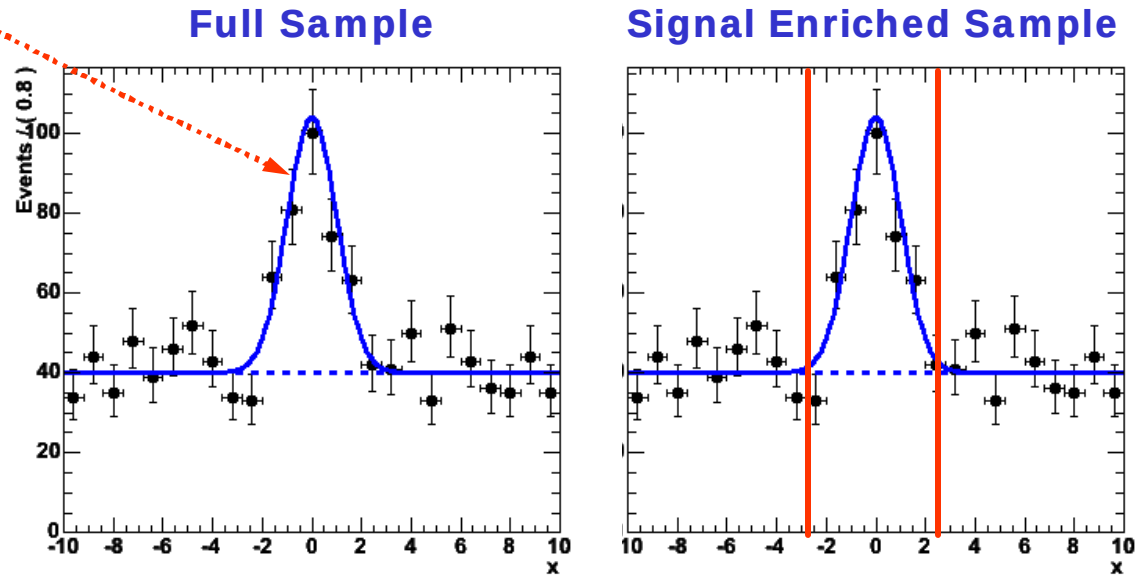
Intermezzo – Role of simulation in HEP data analysis

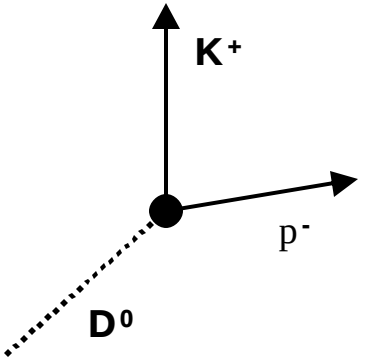
- Monte Carlo simulation is one of the most powerful tools to design and test HEP analyses
  - 'Monte Carlo' is a numerical technique generally directed at the problem of computing integrals. In HEP the 'sampling' aspect of the technique is especially useful to simulate events from given distribution functions
- Typical layout of simulation facilities of HEP experiments



Simple example – one discriminating variable

- Suppose we are looking at the decay $D^0 \rightarrow K^+\pi^-$.
  - We take two tracks and form the invariant mass m(Kπ)
  - Distribution of m(Kπ) will peak around m(D0) for signal
  - Distribution of m(Kπ) will be more or less flat for combinatorial background (random combinations of two tracks)



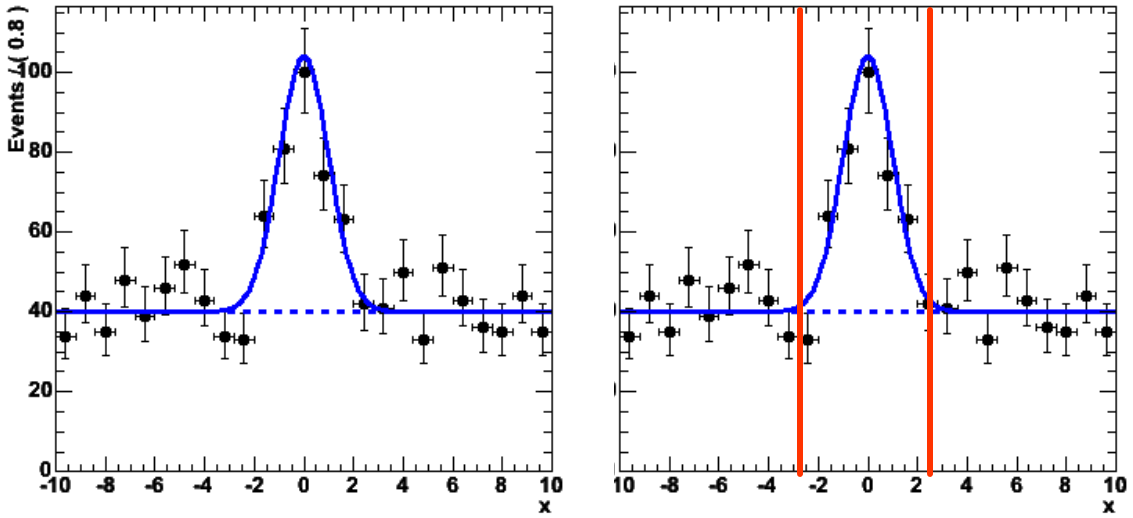


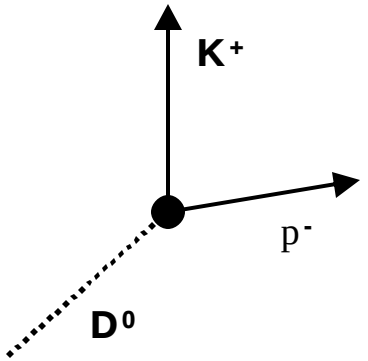
  - We can enhance the purity of our sample by cutting on m(Kπ)

Simple example – one discriminating variable

– We can enhance the purity of our sample by cutting on m(Kπ)

Full Sample **Signal Enriched Sample**

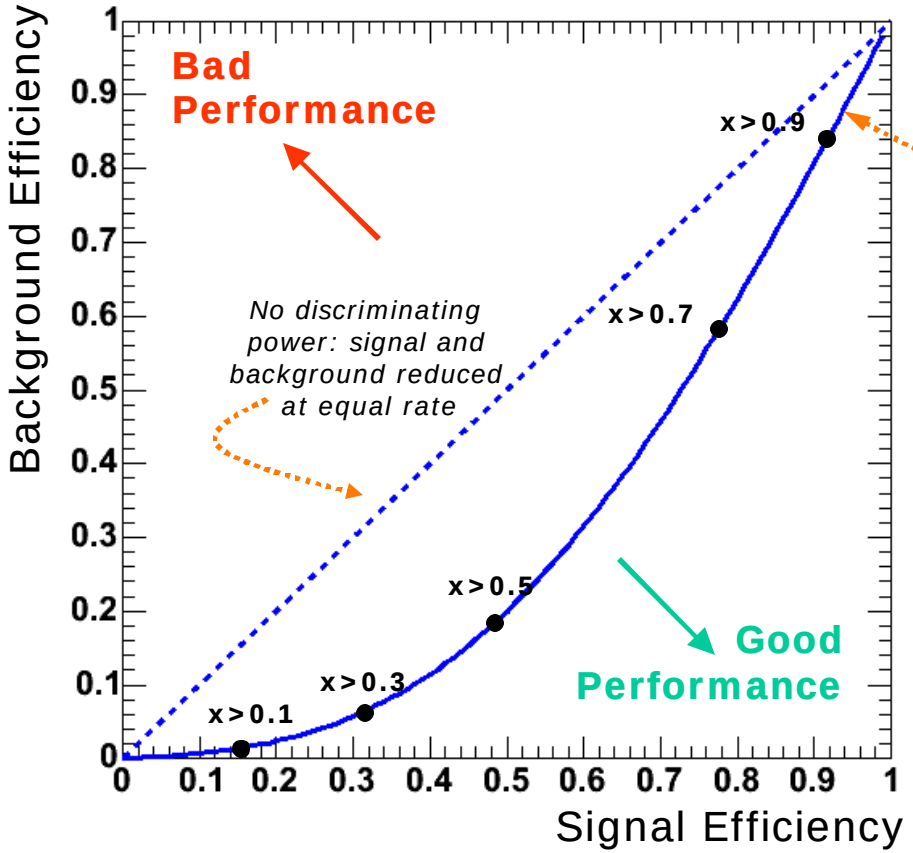




- **How do we decide that this cut is 'optimal'?**
  - We can choose cuts 'by eye' looking at the data – probably fine in this case, but not always so easy
  - More robust approach: Study separate samples of simulated signal and background events and make informed decision

Optimizing cuts – Looking at simulated events

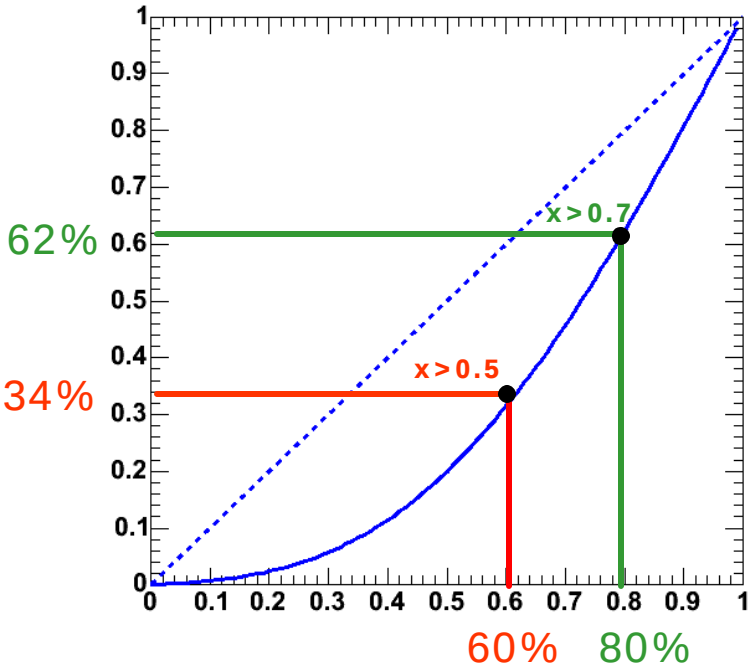
- Not all discriminating variables are equal – What is the selection power of your event variable?
  - Scan range of cut values and calculate signal, background efficiency for each point. Plot ε_{sig} versus ε_{bkg}



Background efficiency as function of signal efficiency

This type of plot is useful to compare the merits of various discriminating variables
but it doesn't tell you *where* to cut

Optimizing cuts – Looking at simulated events



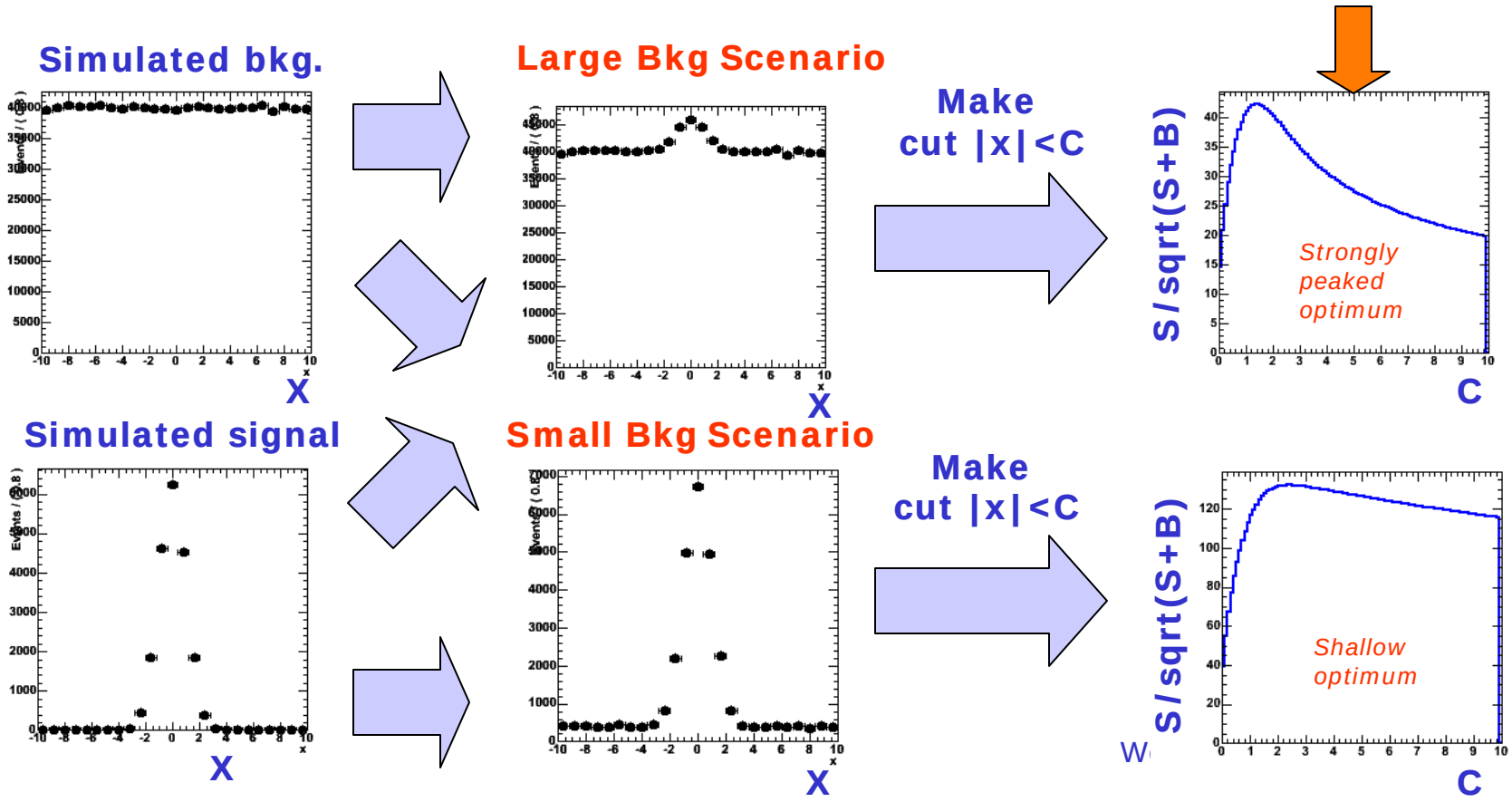
This type of plot is useful to compare the merits of various discriminating variables
but it doesn't tell you *where* to cut

- Choosing optimal cut require additional piece of information: the expected *amount* of signal, background
  - Lot of signal / little background → Cut looser
  - Little signal / lots of background → Cut harder
- Goal for optimization: minimize error on N(signal)

Optimizing your cut for the best signal significance

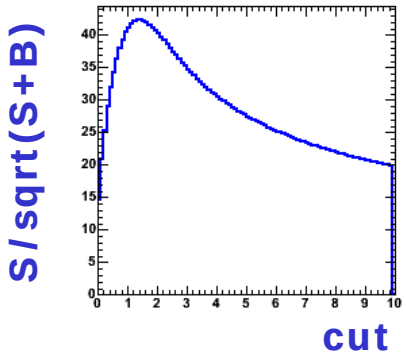
- Formula for approximate signal significance:
  - Formula only good for large N, asymmetric Poisson shape of distributions distorts results at low N

$$signif\,(N_{sig}) \propto \frac{N_{sig}}{\sqrt{N_{sig}+N_{bkg}}}$$



Optimizing your cut for the best signal significance

$$signif(N_{sig}) \propto \frac{N_{sig}}{\sqrt{N_{sig} + N_{bkg}}}$$



- If $N_{sig} << N_{bkg}$ then $N_{sig}+N_{bkg}$ can be approximated by N_{bkg}
- If you have no (good) background simulation, and N_{sig} is small you can also consider to replace $N_{sig}+N_{bkg}$ by N(DATA)
- In the limit of low data (MC) statistics, SSB curve may exhibit statistical fluctuations
  - Don't write algorithms that blindly finds the absolute maximum of S/sqrt(S+B)
  - Be especially careful if you use data as tuning to those statistical fluctations may bias your result

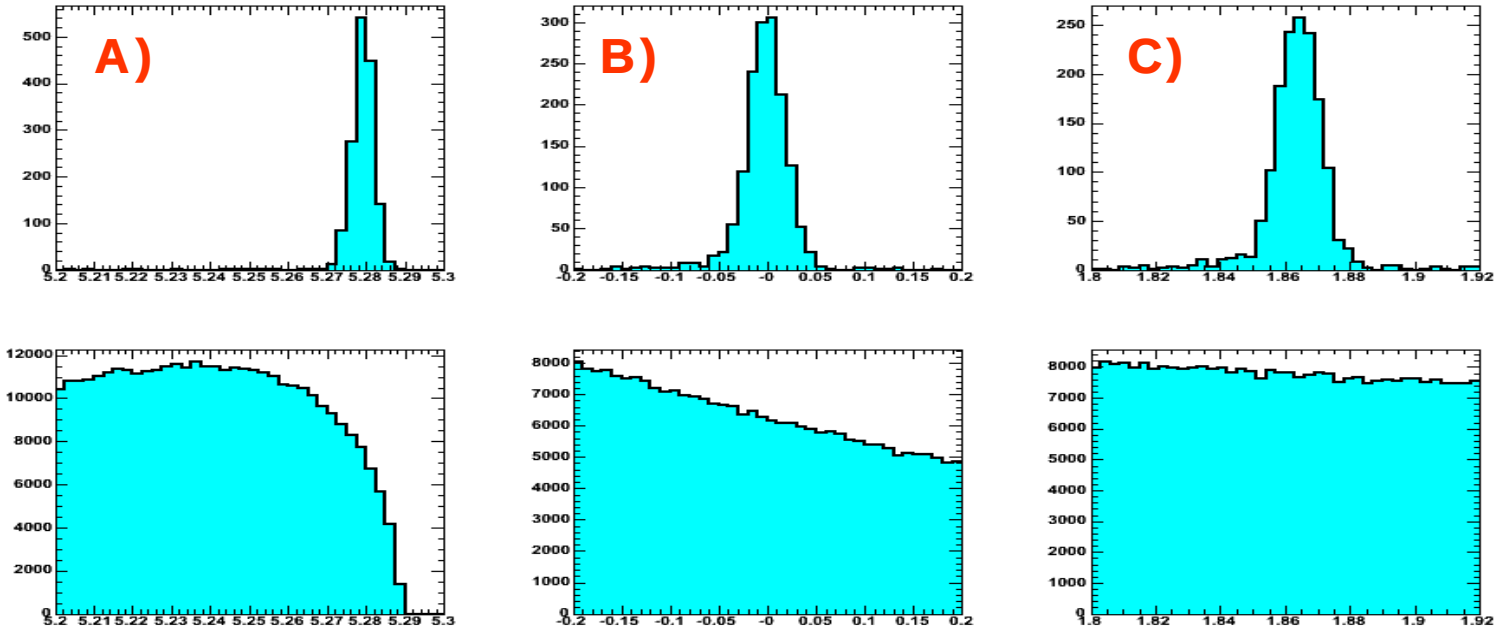
Optimizing a cut on multiple discriminating variables

- How to tune your cut if there is more than one discriminating variable?
- An example with three discriminating variables:
  $\Upsilon(4s) \rightarrow B^+B^-$, $B^- \rightarrow D^0 \pi^-$, $D^0 \rightarrow K^+\pi^-$

$m_{ES}(B^+)$
clever variation
on B^+ invariant mass

$E(B^+)$-$E(\Upsilon 4s/2)$
Measured vs expected E
of B^+ in Y4s 2-body system

$m(K^+p^-)$
D^0 candidate
invariant mass



Optimizing a cut on multiple discriminating variables

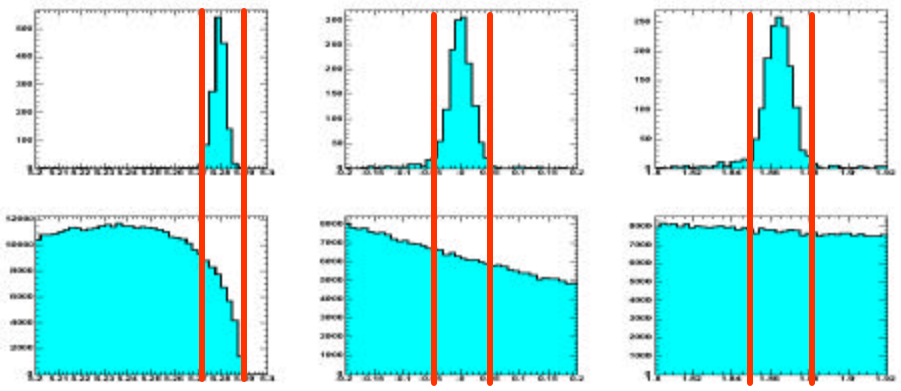
- Problem: need to find optimal S/sqrt(S+B) in 3-dim space
  - ***Difficult!***

- A possible way forward – Iterative approach

1) Start with reasonable 'by eye' cuts for m_{ES},ΔE,$m(K\pi)$
2) Tune each cut after all other cuts have been applied
3) Repeat step 2) until cuts no longer change

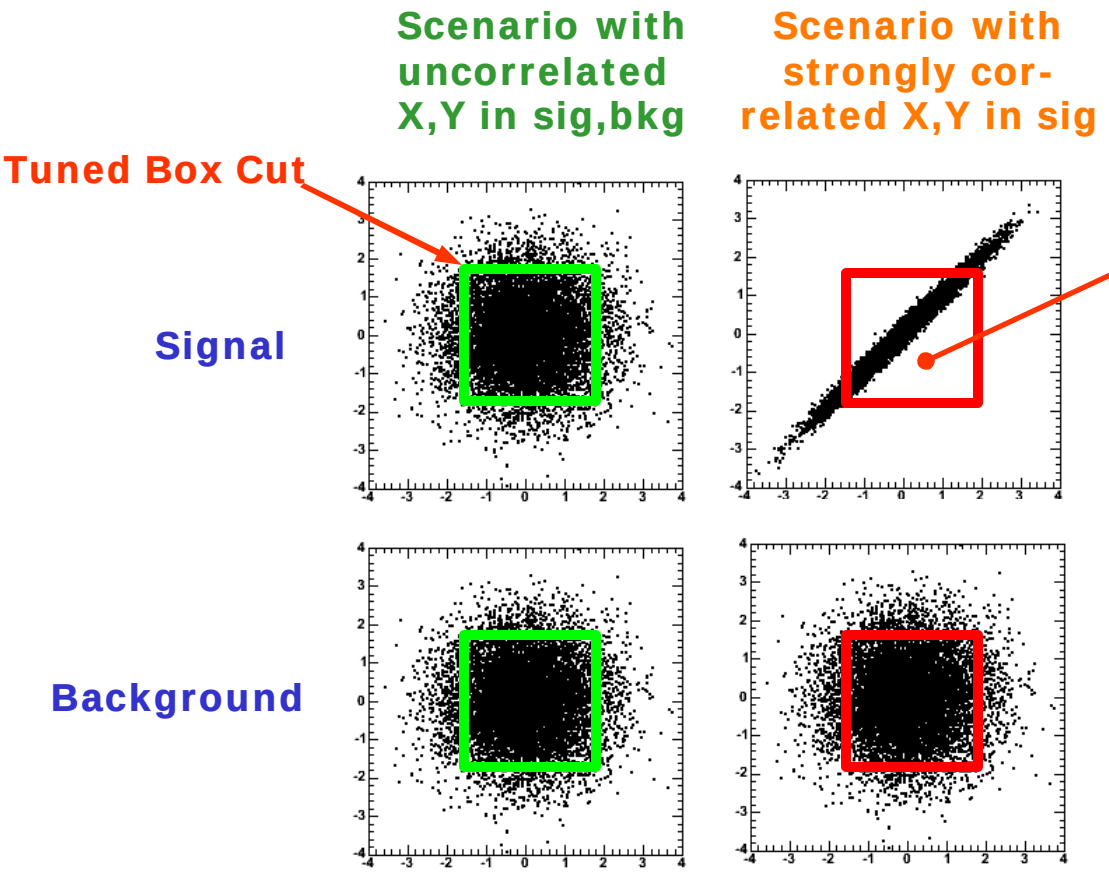
This ensures that the background reducing effects of the other cuts are taken into account in the tuning

Result: a (hyper) cube-shaped cut in the three observables



Multiple discriminating variables – correlations

- Warning: box cut may not be optimal if there are strong correlations between the variables



Additional background could have been reduced at no cost with a differently shaped cut

Need different approach…

Multivariate data selection – constructing a 1D discriminant

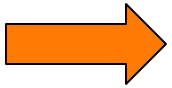
- Instead of tuning a box cut in N observables, *construct a 1-dim discriminant* that incorporates information from all N observables.
  - Why? It is awkward to work with many dimensions
  - How? Try to compactify data and not loose ability to discriminate between signal and background

- The Neyman-Pearson Lemma:
  - Given true signal and background probability

$$S(\vec{x}) \;\; ; \;\; B(\vec{x})$$

the highest purity at a given efficiency is obtained by requiring

$$\frac{S(\vec{x})}{B(\vec{x})} > C$$



Optimal Discriminant

$$D(\vec{x}) = \frac{S(\vec{x})}{B(\vec{x})}$$

where C controls the efficiency

Or any other function with a one-to-one mapping to this function like S/(S+B)

Multivariate data selection – constructing a 1D discriminant

- That's very nice but:
  we usually don't know true S(**x**) and true B(**x**)
  - But we can try to ***estimate*** it from data, simulation etc

- A variety of techniques exist to estimate D(**x**) from signal and background data samples such as
  - Neural net
  - Fisher discriminant
  - Likelihood description
  - Probability density estimate

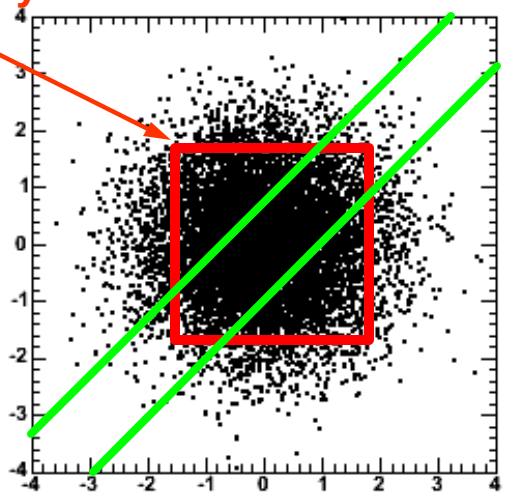
- We'll now explore some of these techniques
  - But keep in mind that the idea behind all these techniques is the same: approximate the optimal discriminant D(**x**)=S(**x**)/B(**x**)

Multivariate data selection – Principal Component Analysis

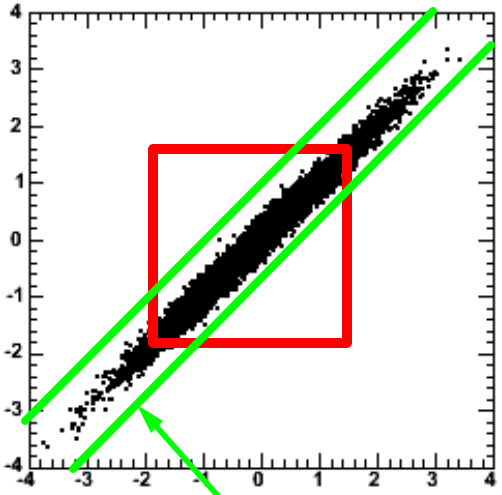
- Idea: reduce dimensionality of data
- Back to example of multi-dimensional box cut

Tuned box cut on original variables x, y

Background



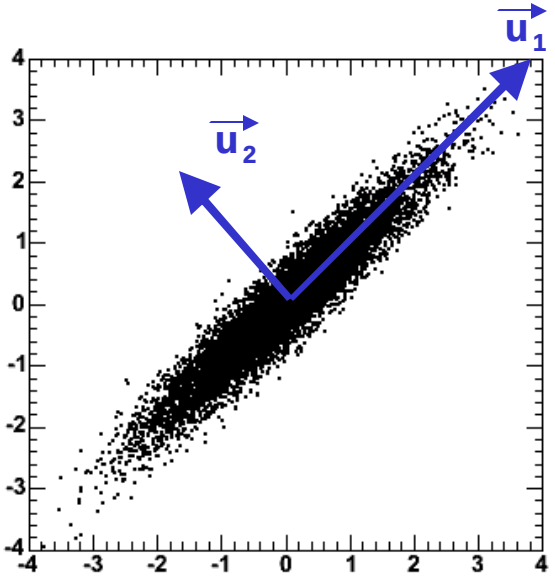
Signal



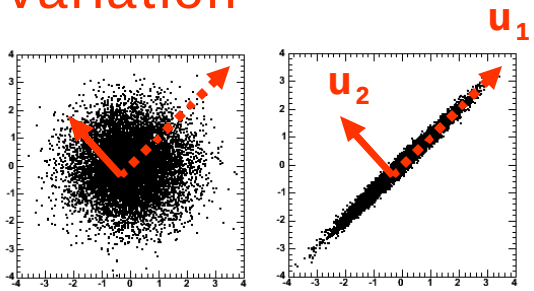
A better (1-dim) cut along axis with largest difference between signal and background

Multivariate data selection – Principal Component Analysis

- How to find principal axes of signal data sample
  - Goal: transform $X=(x_1,x_2)$ to $U=(u_1,u_2)$
  1) Compute variance matrix **Cov(X)**
  2) Compute eigenvalues λ_i and eigenvectors $\mathbf{v}_i$
  3) Construct rotation matrix $\mathbf{T} = \mathbf{Col}(\mathbf{v}_i)^T$
  4) Finally calculate $\mathbf{u}_i = \mathbf{T}\mathbf{x}_i$



- Eliminate $\mathbf{u}_i$ with smallest amount of variation
  - $\mathbf{u}_1$ in example
  - Just cut on $\mathbf{u}_2$



- Software tip: in ROOT the class **`TPrincipal`** does all the hard work for you

Combining discriminating variables – Linear discriminants

- A linear discriminant constructs D(x) from a linear combination of the variables x_i

$$t(\vec{x}) = \sum_{i=1}^{N} a_i x_i = \vec{a} \cdot \vec{x}$$

  - Optimize discriminant by chosing a_i to maximize separation between signal and background

- Most common form of the linear discriminant is the Fisher discriminant

$$F(\vec{x}) = \overbrace{\left(\vec{m}_S - \vec{m}_B\right)^T V^{-1}}^{\vec{a}} \vec{x}$$

Mean values in x_i for sig,bkg

Inverse of variance matrix of signal/background (assumed to be the same)

R.A. Fisher
Ann. Eugen. 7(1936) 179.

Multivariate data selection – Linear discriminants

$$F(\vec{x}) = \left(\vec{m}_S - \vec{m}_B\right)^T V^{-1} \vec{x}$$

R.A. Fisher
Ann. Eugen. 7(1936) 179.

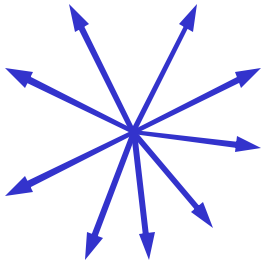
Mean values in x_i for sig,bkg

Inverse of variance matrix of signal/background (assumed to be the same)

- Advantage of Fisher Discriminant:
  - Ingredients $m_s, m_b, \mathbf{V}$ can all be calculated directly from data or simulation samples. No 'training' or 'tuning'
- Disadvantages of Fisher Discriminant
  - Fisher discriminant only exploits difference in means.
  - If signal and background have different variance, this information is not used.

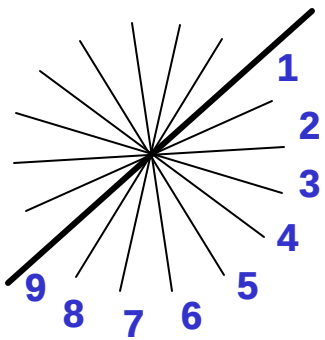
Example of Fisher discriminant

- The CLEO Fisher discriminant
  - Goal: distinguish between $e^+e^- \rightarrow \Upsilon(4S) \rightarrow b\bar{b}$ and $u\bar{u}, d\bar{d}, s\bar{s}, c\bar{c}$
  - Method: Measure energy flow in 9 concentric cones around direction of B candidate

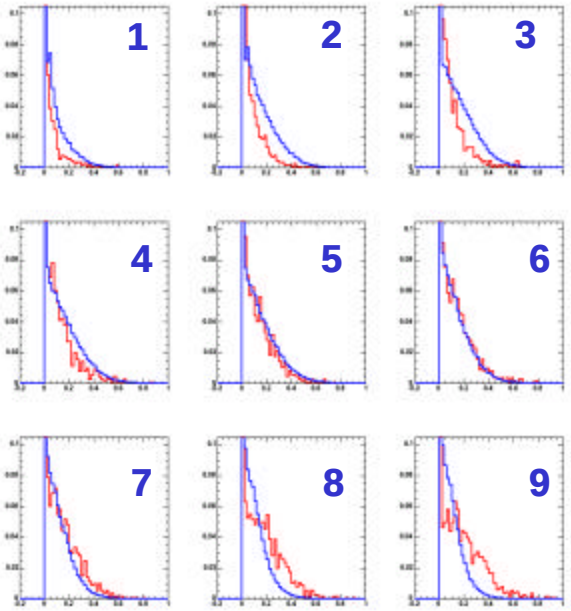


Energy flow in bb

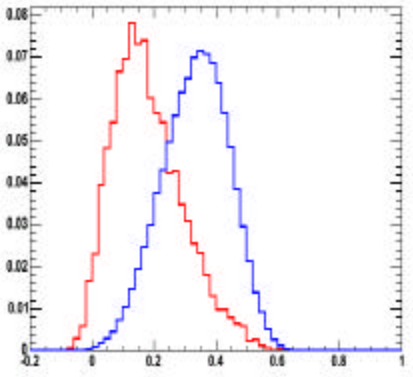
Energy flow in u,d,s,c



Cone Energy flows



F(x)



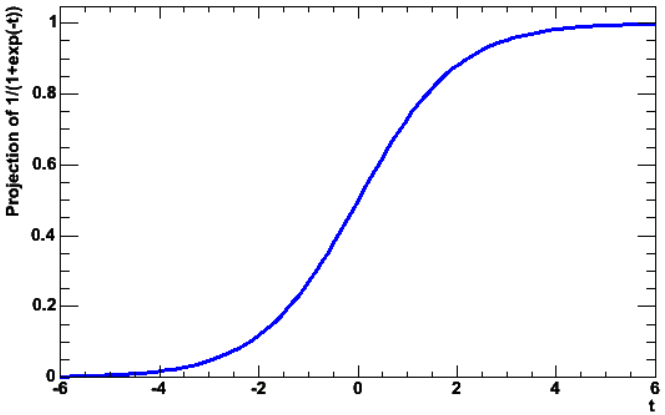
When is Fisher discriminant is the optimal discriminant?

- A very simple dataset

$$S = \prod_i Gauss(x_i; m_i^S, s_i)$$
$$B = \prod_i Gauss(x_i; m_i^B, s_i)$$

Multivariate Gaussian distributions with **different means** but **same width** for signal and background

- Fisher is optimal discriminant for this case
  - In this case we can also directly correlate F(x) to **absolute signal probability**



$$P(F) = \frac{1}{1+e^{-F}}$$

'Logistic sigmoid function'

Multivariate data selection – Neural networks

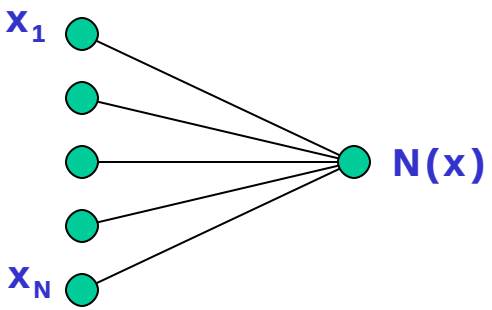
- Neural networks are used in neurobiology, pattern recognition, financial forecasting (and also HEP)

$$N(\vec{x}) = s\left(a_0 + \sum_i a_i x_i\right)$$

s(t) is the *activation function*, usually a logistic sigmoid

$$s(t) = \frac{1}{1+e^{-t}}$$

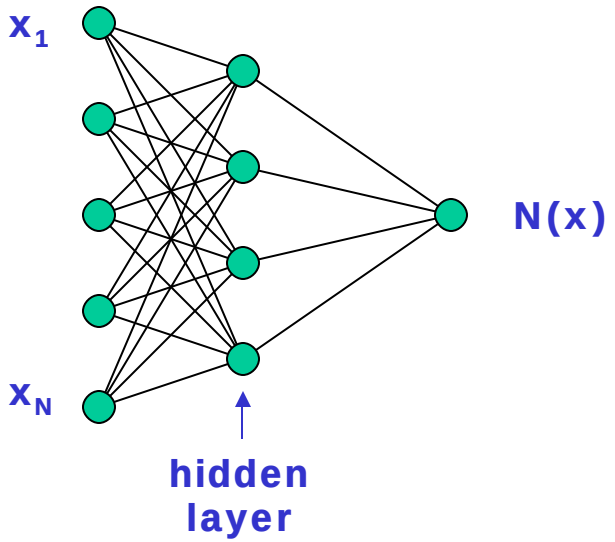
- This formula corresponds to the 'single layer perceptron'
  - Visualization of single layer network topology



Since activation function s(t) is monotonic, **the single layer N(x) is equivalent to the Fisher discriminant F(x)**

Neural networks – general structure

- The single layer model and easily be generalized to a ***multilayer*** perceptron



$$N(\vec{x}) = s(a_0 + \sum_{i=1,}^{m} a_i h_i(\vec{x}))$$

$$with\ h_i(\vec{x}) = s(w_{i0} + \sum_{j=1}^{n} w_{ij} x_j)$$

with a_i and w_{ij} weights (connection strengths)

- Easy to generalize to arbitrary number of layers
- Feed-forward net: values of a node depend only on earlier layers (usually only on preceding layer) 'the network architecture'
- More nodes bring N(x) close to optimal D(x)=S(x)/B(x) but with much more parameters to be determined

Neural networks – training

- Parameters of NN usually determined by minimizing the error function

$$e = \int \left(N(\vec{x}) - 0\right)^2 B(\vec{x}) d\vec{x} \quad + \quad \int \left(N(\vec{x}) - 1\right)^2 S(\vec{x}) d\vec{x}$$

NN target value for background

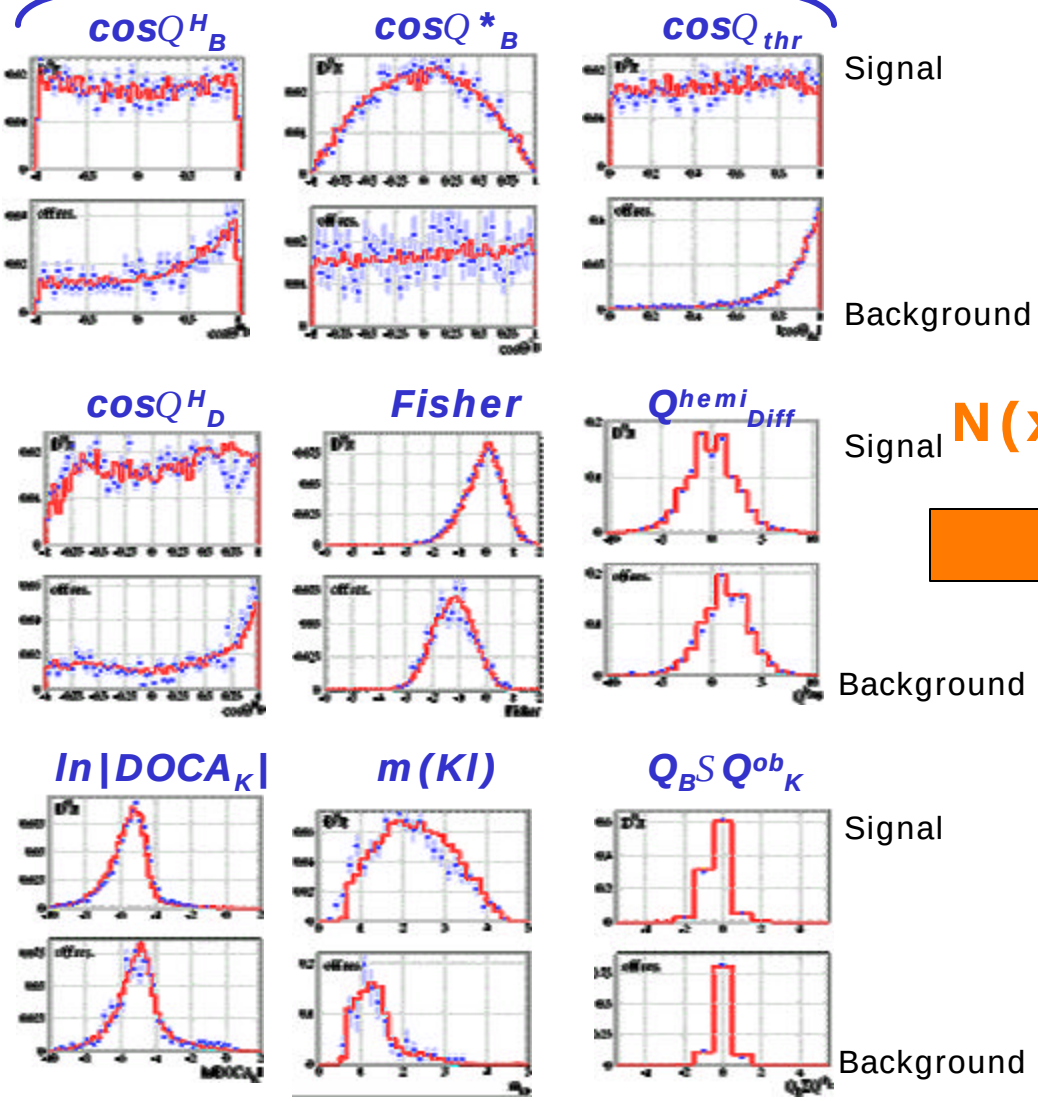
NN target value for signal

- Same principle as Fisher discriminant, but cannot solve analytically for general case
  - In practice replace ε with averages from training data from MC (Adjusting parameters → 'Learning')
  - Generally difficult, but many programs exist to do this for you ('error back propagation' technique most common)

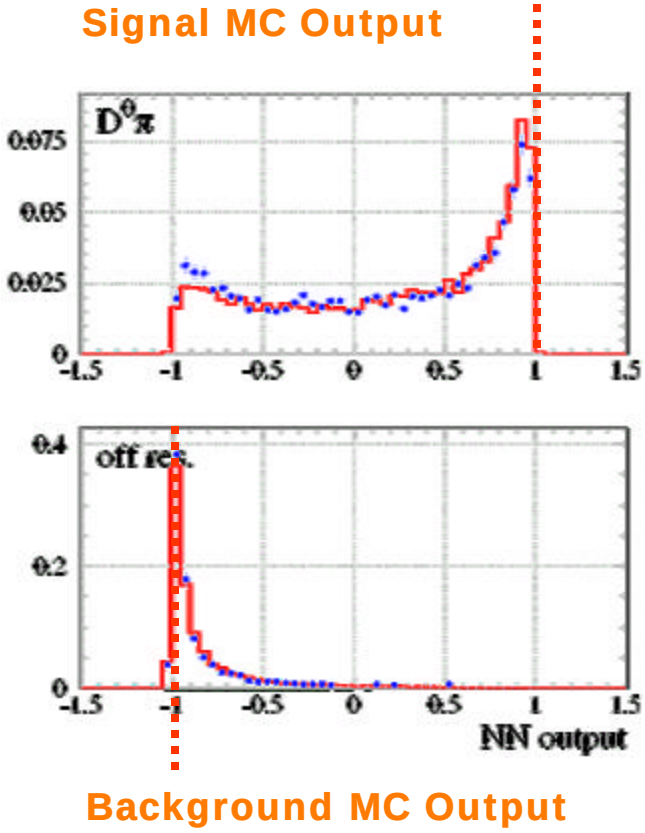
Neural networks – training example

Input Variables (9)

Output Variables (1)



N(x)

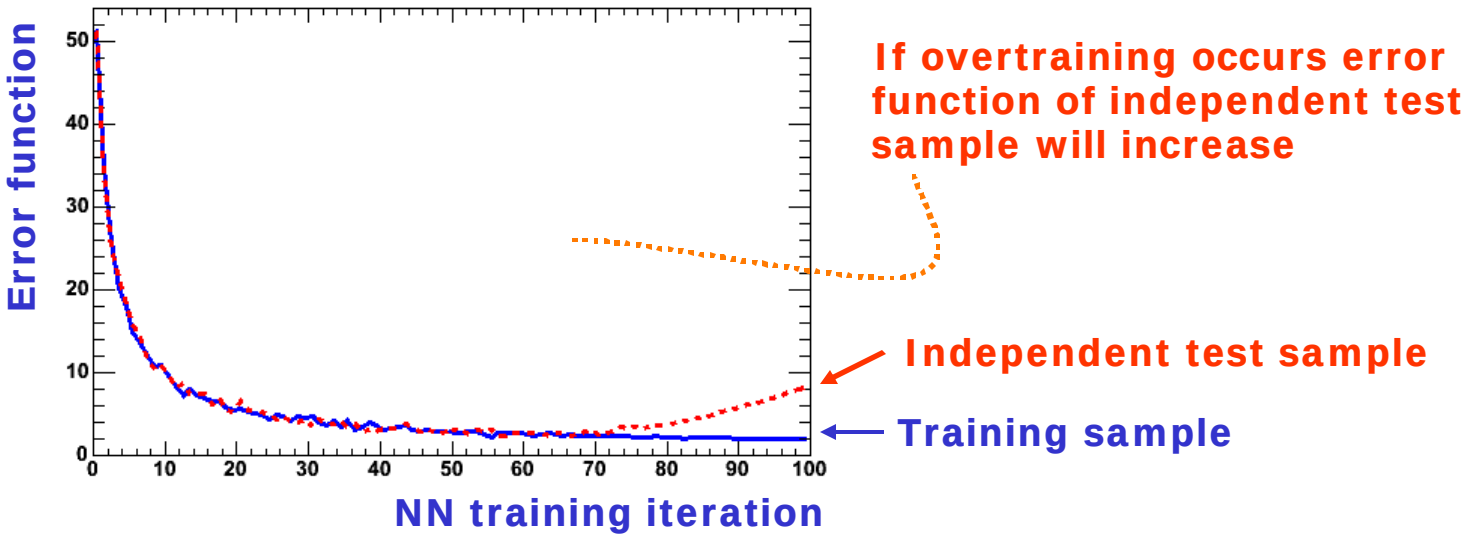


Practical aspects of Neural Net training

- **Choose input variables sensibly**
  - Don't include badly simulated observables (such as #tracks/evt)
  - Some input variables may be highly correlated → drop all but one
  - Some input variables may contain little or no discriminating power → drop them
  - Transform strongly peaked distributions into smooth ones (e.g. take log)
  - Fewer inputs → fewer parameters to be adjusted → parameters better determined for finite training data
- **Choose architecture sensibly**
  - No 'rules' for number of hidden layers, nodes
  - Usually better to start simple and gradually increase compexity and see how that pays off
- **Verify sensible behavior**
  - NN are not magic, understand what your trained NN is doing

Practical aspects of Neural Net training

- Training = iterative minimization of error function
- Beware risks of 'overtraining'
  - Overtraining = You network tunes to statistical fluctuations specific to your training sample that are not representative of the parent distribution
  - **How to avoid detect and avoid overtraining:**
    Look simultaneously at error function evaluated from independent input samples not used in training



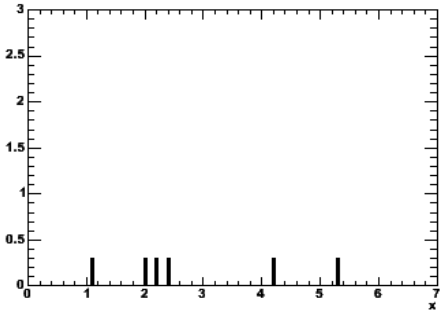
Neural networks – Software and literature

- Basic Neural Net implementations for analysis use
  - PAW: MLP (multi-layer perceptron) – built-in
  - ROOT: **`TMultiLayerPerceptron`** – built-in
  - Good enough for most basic analysis use
- More powerful standalone packages exist
  - For example JETNET
- Further reading
  - L. Lönnblad et al., Comp. Phys. Comm. 70 (1992), 167
  - C. Peterson et al., Comp. Phys. Comm. 81 (1994), 185
  - C.M. Bishop, *Neural Nets for Pattern Recognition*, Clarendon Press, Oxford (1995)
  - B. Muller et al., *Neural Networks: an Introduction*, 2nd edition, Springer, Berlin (1995)

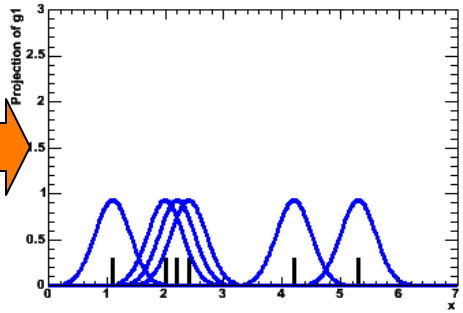
Multivariate data selection – Probability density estimates

- Probability Density Estimate technique aims to construct S(x) and B(x) separately
  - rather than D(x) directly, like NN does
  - Calculate $$D(\vec{x}) = \frac{S_{PDE}(\vec{x})}{B_{PDE}(\vec{x})}$$
- Idea (1-dim): represent each event of your MC sample as a Gaussian probability distribution
  - Add probability distributions from all events in sample
  - Example:

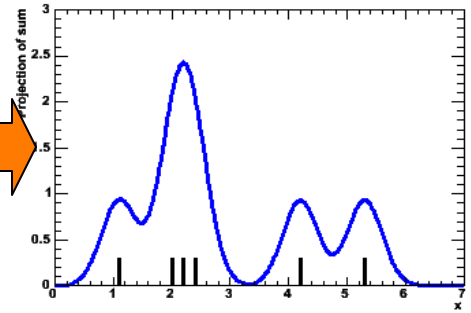
Sample of events



Gaussian probability distributions for each event



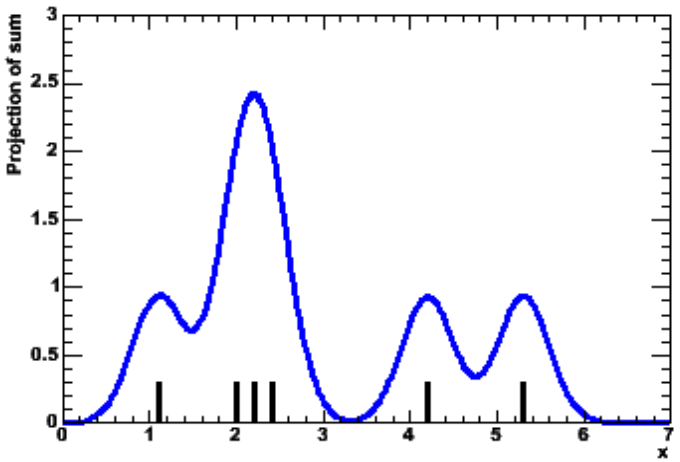
Summed probability distribution for all events in sample



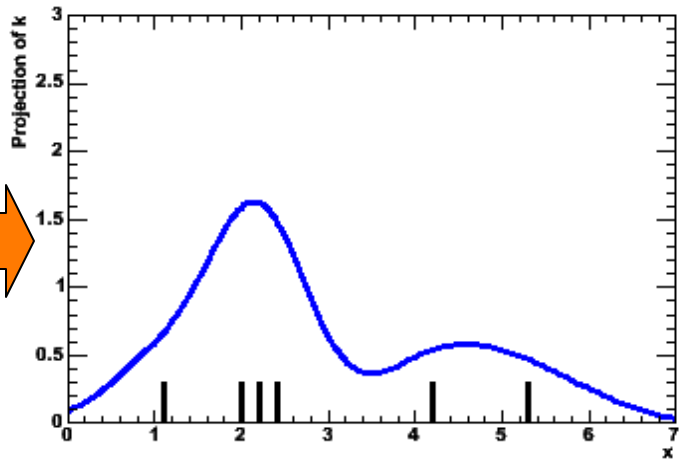
Probability Density Estimates – Adaptive Kernel

- Width of single event Gaussian can of course vary
  - Width of Gaussian tradeoff between smoothness and ability to describe small features
- Idea: 'Adaptive kernel' technique
  - Choose wide Gaussian if local density of events is low
  - Choose narrow Gaussian if local density of events is high
  - Preserves small features in high statistics areas, minimize jitter in low statistics areas

Static Kernel
(with of all Gaussian identical)



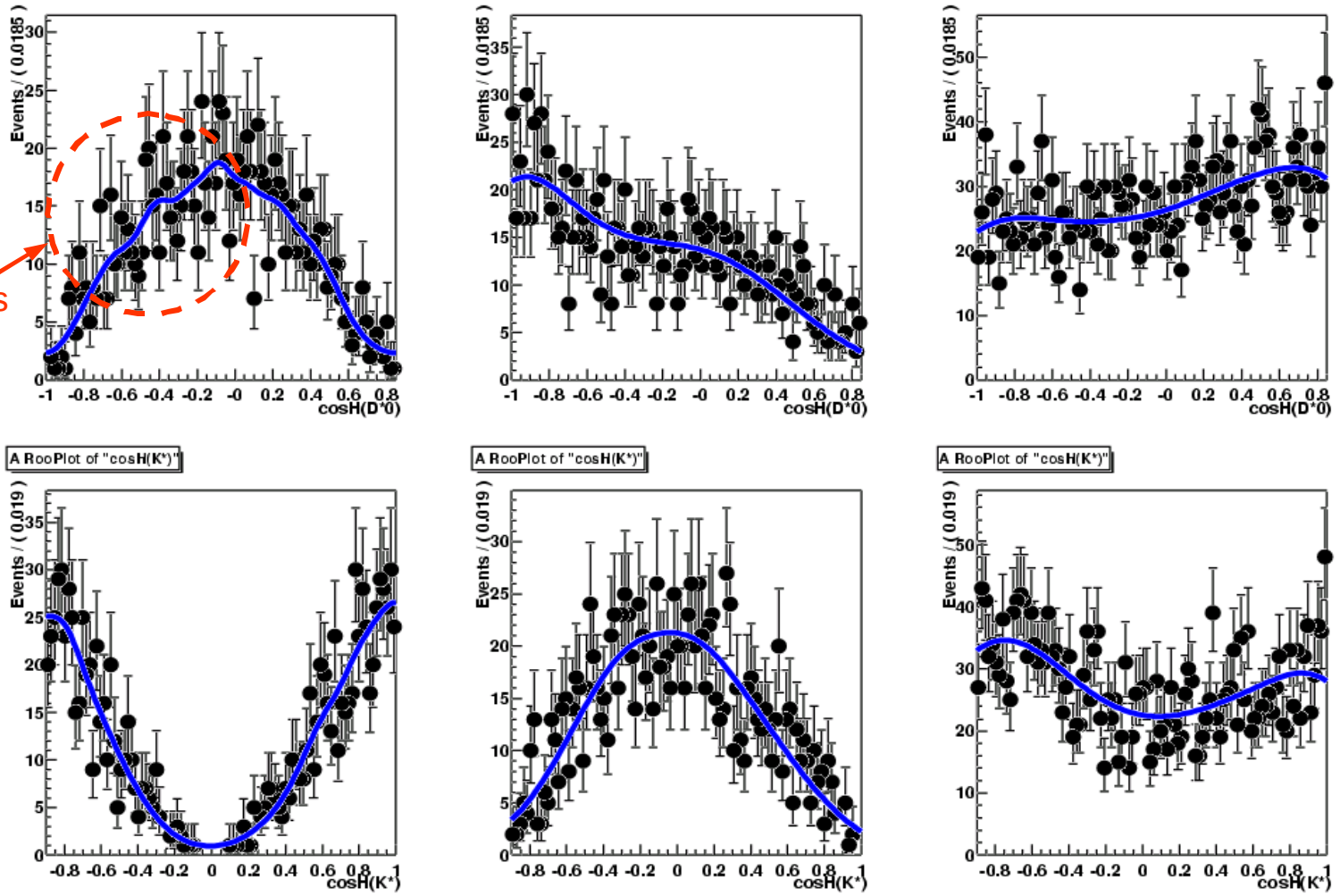
Adaptive Kernel
(width of all Gaussian depends on local density of events)



Probability Density Estimates – Some examples

- Illustration: some PDEs from realistic data samples

Some wobbliness due to limited statistics

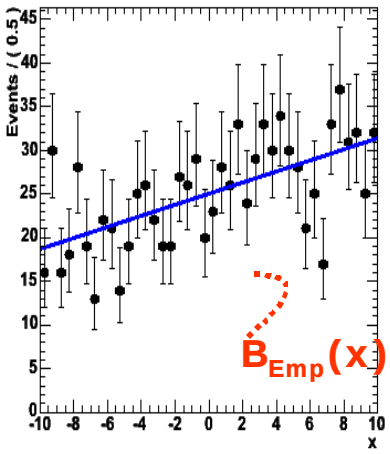


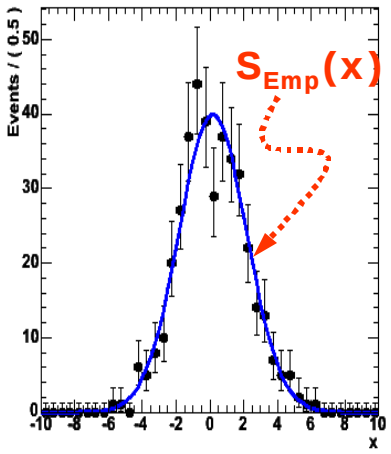
Probability Density Estimates

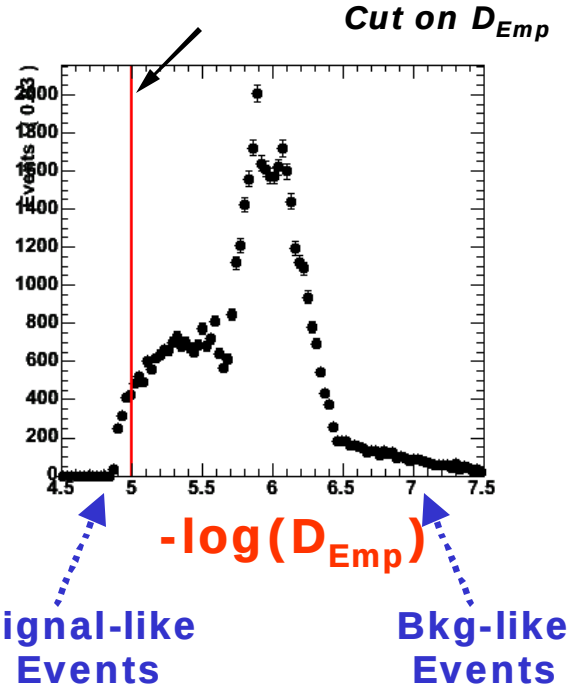
- Also works in multiple dimensions
  - Analogy in N dimensions straightforward
  - But watch for very low statistics regions, which are much more common in multi-dimensional problems
- Key features of PDE technique
  - Advantage: No training necessary, no functional form assumed
  - Disadvantage: Prone to effects of low statistics
- Further reading
  - K. Cranmer – Kernel Estimate Your Signal, Comp Phys Comm XXX
  - S. Towers – PhySTAT2003 conference

Multivariate data selection – empirical modeling

- Idea: Choose empirical model to describe your signal and background data
  - Works best if you have little training data and you have an approximate idea what the functional form will look like
  - Fit probability density functions $S_{Emp}(x;p_S)$, $B_{Emp}(x;p_B)$ functions to signal, background data to obtain best possible description for given model







- Calculate $D_{Emp}(x) = \dfrac{S_{Emp}(x)}{B_{Emp}(x)}$

Multivariate data selection – Likelihood description

- Most useful for multi-dimensional datasets
  - Application of technique in N dimensions straightforward

$$D(x,y) = \frac{S_x(x) \cdot S_y(y)}{B_x(x) \cdot B_y(y)}$$

Explicitly assumes that x and y are **uncorrelated** in signal and background
Easy, but possibly ignores information

alternatively

$$D(x,y) = \frac{S(x,y)}{B(x,y)}$$

Incorporates **correlations**.
Potentially **more powerful**, but **more work**

- Why cut on $D_{Emp}(x)$ rather than using the result from the fit directly?
  - Fitting multidimensional datasets is quite a bit of work
  - If function does not describe data perfectly (especially difficult in multiple dimensions with correlations), accounting for discrepancy in fit result a lot of work. Failing to do so may result in wrong answer.
  - With a cut on $D_{Emp}(x)$ efficiency of cut as measured on data or simulation will incorporate all such effects in the obtained cut efficiency

Summary of background rejection methods

Method	Merits	Drawbacks
Box cut	Easy to understand, explain	Correlations not handled, doesn't scale well to many variables
Principal Component Analysis	Easy to understand, explain, correlation taken into account	May not be close to optimal for complex problems
Fisher	Conceptually easy, implementation easy	Does not exploit difference in variance
Neural Net	Flexible, powerful	Training can be difficult
Probability	No free parameters, conceptually easy	Does not work well with low statistics
Empirical Function Method	Works well with low statistics training samples	Quality of discriminant depends strongly on you guessing the correct functional form

Finding the right method

- Which one is right for you? Depends on
  - Complexity of your problem
  - Time scale in which you would like to finish the analysis

- On finding the absolute best set of cuts
  - ***All*** methods for finding discriminants are approximate when used with finite training/tuning statistics
  - Your experiments event simulation is imperfect – your performance on data can be different (usually it is less)
  - You may a systematic error later that might depend on your choice of cuts
  - Don't hunt for upward statistical fluctuations in tuning data
  - If it takes you 6 months of work to reduce your error by 10% keep in mind that your experiment may have accumulated enough additional data by them to reduce your statistical error by a comparable or larger amount

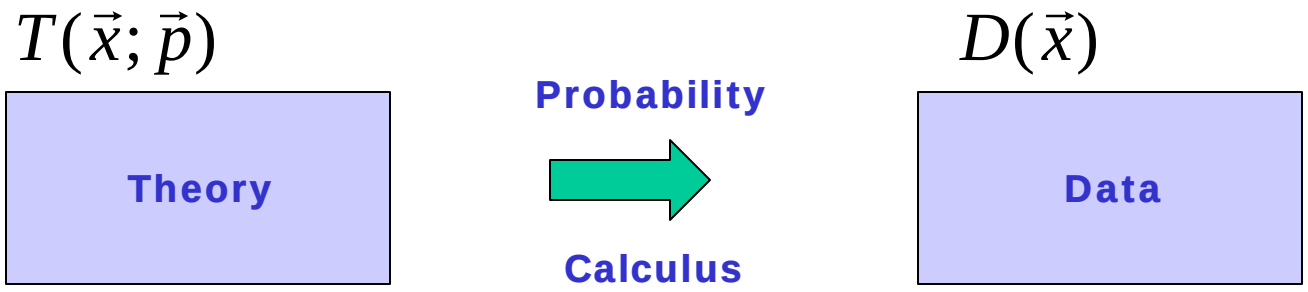
- It is more important to get the right(=unbiased) answer than the smallest possible statistical error
  - Don't use discriminating variables that you know are poorly modeled in simulation
  - Always try to find a way to cross check your performance on data, e.g. by using a control sample

Estimation & Fitting

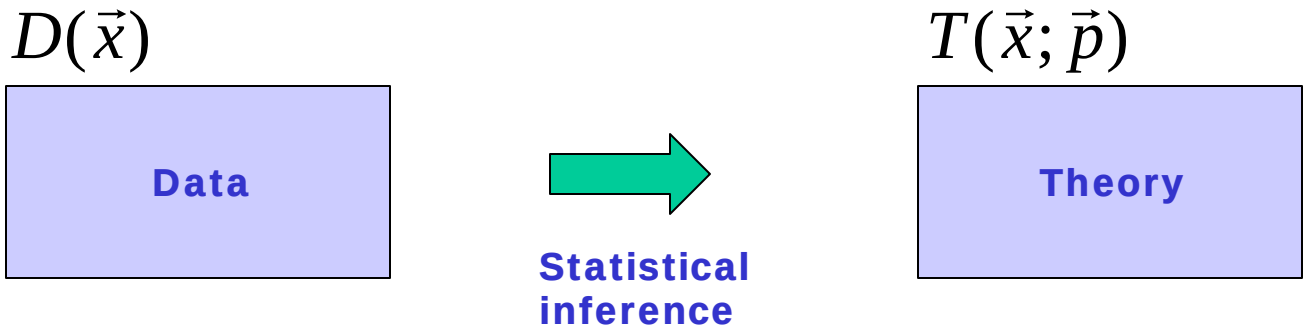
- Introduction to estimation
- Properties of χ^2, ML estimators
- Measuring and interpreting Goodness-Of-Fit
- Numerical issues in fitting
- Understanding MINUIT
- Mitigating fit stability problems
- Bounding fit parameters
- Fit validation studies
    - Fit validity issues at low statistics
- Toy Monte Carlo techniques
- Simultaneous fitting
- Multidimensional fitting

Wouter Verkerke, UCSB

Estimation – Introduction



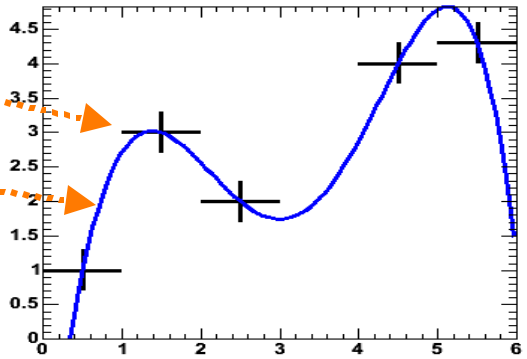
- Given the theoretical distribution parameters p, what can we say about the data



- **Need a procedure to estimate *p* from *D***
  - Common technique – fit!

A well known estimator – the χ^2 fit

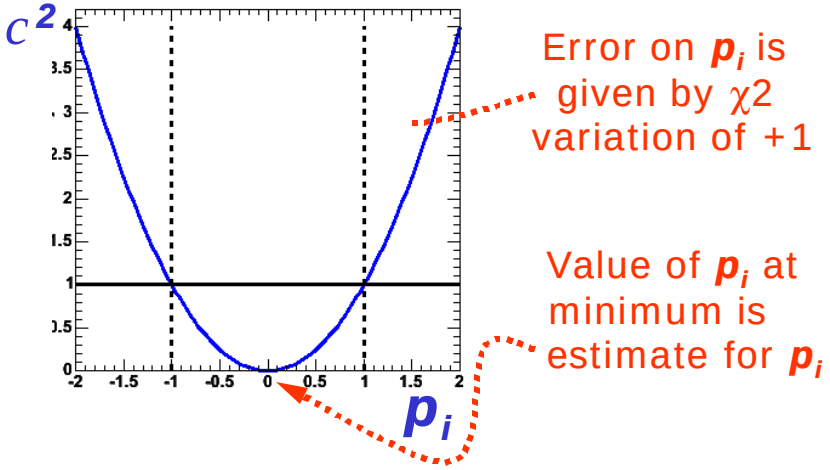
- Given a set of points $\{(\vec{x}_i, y_i, s_i)\}$ and a function ***f(x,p)*** define the χ^2



$$c^2(\vec{p}) = \sum_i \frac{(y_i - f(\vec{x};\vec{p}))^2}{s_y^2}$$

- Estimate parameters by minimizing the $\chi^2(p)$ with respect to all parameters p_i
  - *In practice, look for*

$$\frac{dc^2(p_i)}{dp_i} = 0$$



- Well known: but why does it work? Is it always right? Does it always give the best possible error?

Basics – What is an estimator?

- An ***estimator*** is a ***procedure*** giving a value for a parameter or a property of a distribution as a function of the actual data values, i.e.

$$\hat{m}(x) = \frac{1}{N}\sum_i x_i$$ ← **Estimator of the mean**

$$\hat{V}(x) = \frac{1}{N}\sum_i (x_i - \vec{m})^2$$ ← **Estimator of the variance**

- A perfect estimator is
  - Consistent: $\lim_{n\to\infty}(\hat{a}) = a$
  - Unbiased – *With finite statistics you get the right answer on average*
  - Efficient $V(\hat{a}) = \left\langle (\hat{a} - \langle \hat{a} \rangle)^2 \right\rangle$ ← This is called the **Minimum Variance Bound**
  - ***There are no perfect estimators!***

Likelihood – Another common estimator

- Definition of Likelihood
  - given $\mathbf{D(\vec{x})}$ and $\mathbf{F(\vec{x};\vec{p})}$

NB: Functions used in likelihoods must be Probability Density Functions:

$$\int F(\vec{x};\vec{p})d\vec{x} \equiv 1, \quad F(\vec{x};\vec{p}) > 0$$

$$L(\vec{p}) = \prod_i F(\vec{x}_i;\vec{p}), \quad \text{i.e.} \quad L(\vec{p}) = F(x_0;\vec{p}) \cdot F(x_1;\vec{p}) \cdot F(x_2;\vec{p}) \ldots$$

  - For convenience the negative log of the Likelihood is often used

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

- Parameters are estimated by maximizing the Likelihood, or equivalently minimizing –log(L)

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

Variance on ML parameter estimates

- The **estimator** for the **parameter variance** is

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

From Rao-Cramer-Frechet inequality

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

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

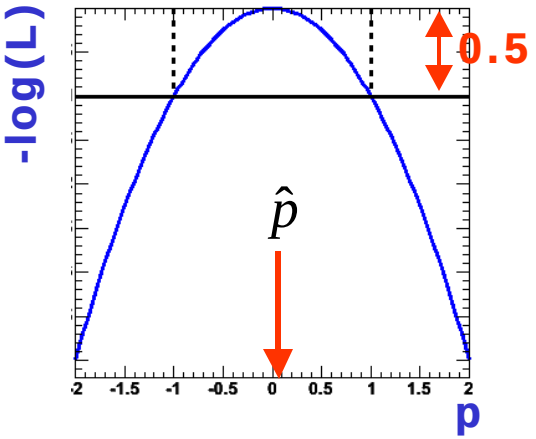
  - I.e. variance is estimated from 2nd derivative of –log(L) at minimum
  - Valid if estimator is **efficient** and **unbiased**!

- **Visual interpretation** of variance estimate
  - Taylor expand –log(L) around minimum

$$\ln L(p) = \ln L(\hat{p}) + \left. \frac{d \ln L}{dp} \right|_{p=\hat{p}} (p - \hat{p}) + \tfrac{1}{2} \left. \frac{d^2 \ln L}{d^2 p} \right|_{p=\hat{p}} (p - \hat{p})^2$$

$$= \ln L_{\max} + \left. \frac{d^2 \ln L}{d^2 p} \right|_{p=\hat{p}} \frac{(p - \hat{p})^2}{2}$$

$$= \ln L_{\max} + \frac{(p - \hat{p})^2}{2 \hat{s}_p^2} \quad \Rightarrow \quad \ln L(p \pm s) = \ln L_{\max} - \frac{1}{2}$$



Properties of Maximum Likelihood estimators

- In general, Maximum Likelihood estimators are
  - **Consistent** (gives right answer for N→∞)
  - **Mostly unbiased** (bias ∝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 $(\hat{p})^2 = \widehat{(p^2)}$

Use of 2nd derivative of –log(L) for variance estimate is usually OK

- MLE efficiency theorem: the MLE will be unbiased and efficient if an unbiased efficient estimator exists
  - Proof not discussed here for brevity
  - Of course this does not guarantee that any MLE is unbiased and efficient for any given problem

More about maximum likelihood estimation

- It's not 'right' it is just sensible

- It does not give you the 'most likely value of p' –
  it gives you *the value of p for which this data is most likely*

- Numeric methods are often needed to find
  the maximum of ln(L)
  - Especially difficult if there is >1 parameter
  - Standard tool in HEP: MINUIT (more about this later)

- Max. Likelihood does **not** give you a goodness-of-fit measure
  - If assumed F(x;p) is not capable of describing your data for any p,
    the procedure will not complain
  - The absolute value of L tells you nothing!

Properties of χ^2 estimators

- Properties of χ^2 estimator follow from properties of ML estimator

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

Probability Density Function in p for single data point $x_i(s_i)$ and function $f(x_i;p)$

Take log,
Sum over all points x_i

$$\ln L(\vec{p}) = -\tfrac{1}{2}\sum_i \left(\frac{y_i - f(x_i;\vec{p})}{s_i}\right) = -\tfrac{1}{2}\,c^2$$

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 s_i is truly Gaussian**
  - i.e. need $n_i > 10$ if y_i follows a Poisson distribution

- Bonus: Goodness-of-fit measure – $\chi^2 \approx 1$ per d.o.f

Maximum Likelihood or χ^2 – What should you use?

- χ^2 fit is fastest, easiest
  - Works fine at high statistics
  - Gives absolute goodness-of-fit indication
  - Make (incorrect) Gaussian error assumption on low statistics bins
  - Has bias proportional to 1/N
  - Misses information with feature size < bin size

- Full Maximum Likelihood estimators most robust
  - No Gaussian assumption made at low statistics
  - No information lost due to binning
  - Gives best error of all methods (especially at low statistics)
  - No intrinsic goodness-of-fit measure, i.e. no way to tell if 'best' is actually 'pretty bad'
  - Has bias proportional to 1/N
  - Can be computationally expensive for large N

- Binned Maximum Likelihood in between $-\ln L(p)_{\text{binned}} = \sum_{\text{bins}} n_{\text{bin}} \ln F(\vec{x}_{\text{bin-center}}; \vec{p})$
  - Much faster than full Maximum Likihood
  - Correct Poisson treatment of low statistics bins
  - Misses information with feature size < bin size
  - Has bias proportional to 1/N

Using weighted data in estimators

- χ^2 fit of histograms with weighted data are straightforward

$$y_i = \sum_i w_i \qquad c^2 = \sum_i \left(\frac{y_i - f(\vec{x}_i;\vec{p})}{s_i} \right)^2 \qquad s_i = \sqrt{1/\sum_i w_i^2}$$

From C.L.T — **From C.L.T**

  - NB: You may no longer be able to interpret $\hat{s}(p) \equiv \sqrt{\hat{V}(p)}$ as a Gaussian error (i.e. 68% contained in 1σ)

- In ML fits implementation of weights easy, but interpretation of errors is not!

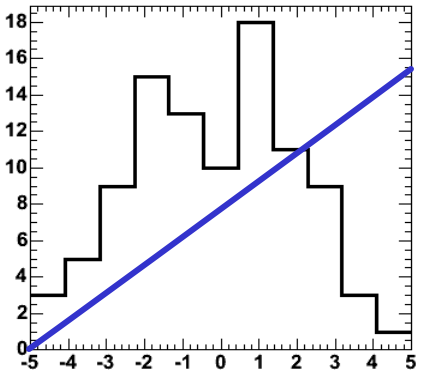
$$-\ln L(\vec{p})_{\text{weighted}} = -\sum_i w_i \ln F(\vec{x}_i;\vec{p})$$

Event weight

  - **Variance estimate on parameters will be proportional to $\sum_i w_i$**
  - If $\sum_i w_i < N$ errors will be too small, if $\sum_i w_i > N$ errors will be too large!
  - Interpretation of errors from weighted LL fits difficult -- Avoid it if you can

Estimating and interpreting Goodness-Of-Fit

- Fitting determines best set of parameters of given model to describe data
  - **Is 'best' good enough?,** i.e.
  - Is it an adequate description, or are there significant and incompatible differences?



'Not good enough'

- Most common test: **the c^2 test**

$$c^2 = \sum_i \left(\frac{y_i - f(\vec{x}_i; \vec{p})}{s_i} \right)^2$$

  - If f(x) describes data then $\chi^2 \approx N$, if $\chi^2 >> N$ something is wrong
  - How to quantify meaning of 'large χ^2'?

How to quantify meaning of 'large χ^2'

- Probability distr. for χ^2 is given by

$$c^2 = \sum_i \left(\frac{y_i - m_i}{s_i} \right)^2 \quad \Rightarrow \quad p(c^2, N) = \frac{2^{-N/2}}{\Gamma(N/2)} c^{N-2} e^{-c^2/2}$$

- To make judgement on goodness-of-fit, relevant quantity is integral of above:

$$P(c^2; N) = \int_{c^2}^{\infty} p(c^{2\prime}; N) dc^{2\prime}$$

- **What does c^2 probability $P(c^2,N)$ mean?**
  - It is the probability that a function which does genuinely describe the data on N points would give a χ^2 probability as large or larger than the one you already have.
    - Since it is a probability, it is a number in the range [0-1]

Goodness-of-fit – χ^2

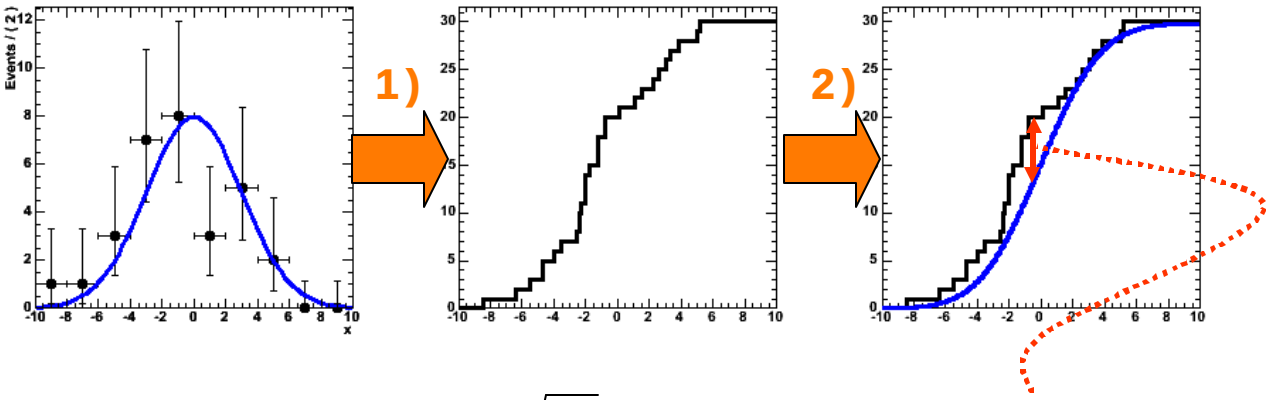
- Example for χ^2 probability
  - Suppose you have a function **f(x;p)** which gives a χ^2 of 20 for 5 points (histogram bins).
  - Not impossible that **f(x;p)** describes data correctly, just unlikely
  - How unlikely? $\int_{20}^{\infty} p(c^2,5)dc^2 = 0.0012$
- Note: If function has been fitted to the data
  - Then you need to account for the fact that parameters have been adjusted to describe the data

$$N_{\text{d.o.f.}} = N_{\text{data}} - N_{\text{params}}$$

- Practical tips
  - To calculate the probability in PAW '**`call prob(chi2,ndf)`**'
  - To calculate the probability in ROOT '**`TMath::Prob(chi2,ndf)`**'
  - For large N, sqrt($2\chi^2$) has a Gaussian distribution with mean sqrt(2N-1) and σ=1

Goodness-of-fit – Alternatives to χ^2

- When sample size is very small, it may be difficult to find sensible binning – Look for binning free test
- Kolmogorov Test
  1) Take all data values, arrange in increasing order and plot cumulative distribution
  2) Overlay cumulative probability distribution



– GOF measure: $d = \sqrt{N} \cdot \max\left|\mathrm{cum}(x) - \mathrm{cum}(p)\right|$

– 'd' large → bad agreement; 'd' small – good agreement

– Practical tip: in ROOT: **`TH1::KolmogorovTest(TF1&)`** calculates probability for you

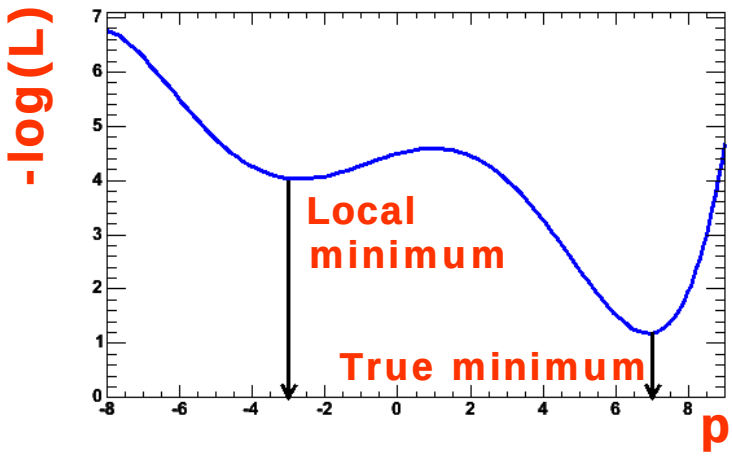
Practical estimation – Numeric χ^2 and -log(L) minimization

- For most data analysis problems minimization of χ^2 or –log(L) cannot be performed analytically
  - Need to rely on numeric/computational methods
  - In >1 dimension generally a difficult problem!

- But no need to worry – Software exists to solve this problem for you:
  - Function minimization workhorse in HEP many years: MINUIT
  - MINUIT does function minimization and error analysis
  - It is used in the PAW,ROOT fitting interfaces behind the scenes
  - It produces a lot of useful information, that is sometimes overlooked
  - Will look in a bit more detail into MINUIT output and functionality next

Numeric χ^2/-log(L) minimization – Proper starting values

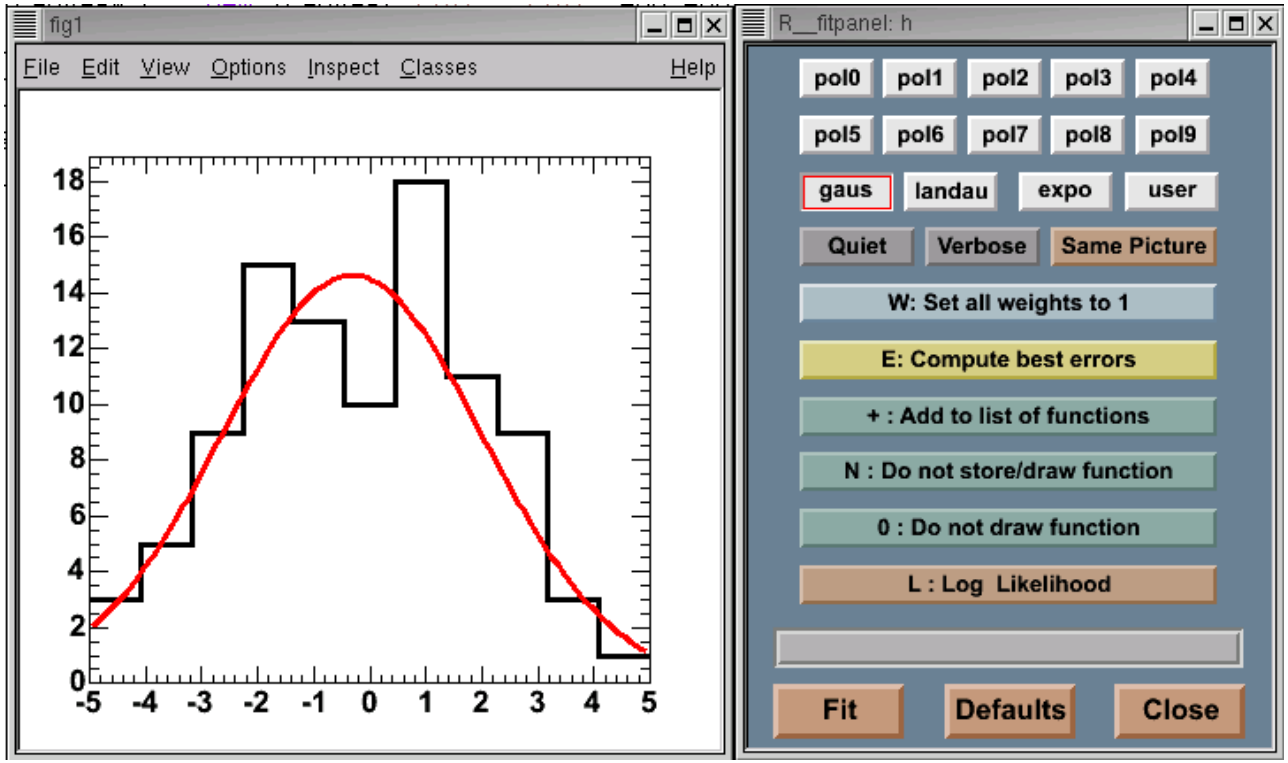
- For all but the most trivial scenarios it is not possible to automatically find reasonable starting values of parameters
  - This may come as a disappointment to some…
  - So you need to supply good starting values for your parameters



Reason: There may exist multiple (local) minima in the likelihood or c^2

  - Supplying good initial uncertainties on your parameters helps too
  - Reason: Too large error will result in MINUIT coarsely scanning a wide region of parameter space. It may accidentally find a far away local minimum

Example of interactive fit in ROOT



- What happens in MINUIT behind the scenes
  1) Find minimum in –log(L) or χ^2 – MINUIT function MIGRAD
  2) Calculate errors on parameters – MINUIT function HESSE
  3) Optionally do more robust error estimate – MINUIT function MINOS

Minuit function MIGRAD

- Purpose: find minimum

```
**********
**    13 **MIGRAD        1000           1
**********
(some output omitted)
MIGRAD MINIMIZATION HAS CONVERGED.
MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=257.304 FROM MIGRAD    STATUS=CONVERGED      31 CALLS          32 TOTAL
                     EDM=2.36773e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                          STEP         FIRST
 NO.   NAME      VALUE            ERROR                SIZE      DERIVATIVE
  1  mean         8.84225e-02   3.23862e-01   3.58344e-04  -2.24755e-02
  2  sigma        3.20763e+00   2.39540e-01   2.78628e-04  -5.34724e-02
                               ERR DEF= 0.5
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 1.049e-01  3.338e-04
 3.338e-04  5.739e-02
PARAMETER  CORRELATION COEFFICIENTS
      NO.   GLOBAL      1      2
       1  0.00430   1.000  0.004
       2  0.00430   0.004  1.000
```

Progress information, watch for errors here

Parameter values and approximate errors reported by MINUIT

Error definition (in this case 0.5 for a likelihood fit)

Minuit function MIGRAD

- Purpose: find minimum

```
**********
**   13 **MIGR
**********
(some output o
MIGRAD MINIMIZ
MIGRAD WILL VERIF                    ENCE AND ERROR MATRIX.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=257.304 FROM MIGRAD    STATUS=CONVERGED      31 CALLS          32 TOTAL
                    EDM=2.36773e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                          STEP         FIRST
  NO.   NAME      VALUE            ERROR                 SIZE      DERIVATIVE
   1  mean         8.84225e-02   3.23862e-01   3.58344e-04  -2.24755e-02
   2  sigma        3.20763e+00   2.39540e-01   2.78628e-04  -5.34724e-02
                               ERR DEF= 0.5
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 1.049e-01  3.338e-04
 3.338e-04  5.739e-02
PARAMETER  CORRELATION COEFFICIENTS
      NO.   GLOBAL      1      2
        1  0.00430   1.000  0.004
        2  0.00430   0.004  1.000
```

Value of c2 or likelihood at minimum

(NB: c^2 values are not divided by $N_{d.o.f}$)

Approximate Error matrix And covariance matrix

Minuit function MIGRAD

- Purpose: find minimu

Status:
Should be 'converged' but can be 'failed'

Estimated Distance to Minimum
should be small O(10^{-6})

Error Matrix Quality
should be 'accurate', but can be 'approximate' in case of trouble

```
**********
**   13 **MIGRAD        1000
**********
(some output omitted)
MIGRAD MINIMIZATION HAS CONVERGED.
MIGRAD WILL VERIFY CONVERGENCE AND E      MATRIX.
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=257.304 FROM MIGRAD    STATUS=CONVERGED      31 CALLS          32 TOTAL
                    EDM=2.36773e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                   STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  mean         8.84225e-02   3.23862e-01   3.58344e-04  -2.24755e-02
  2  sigma        3.20763e+00   2.39540e-01   2.78628e-04  -5.34724e-02
                               ERR DEF= 0.5
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 1.049e-01  3.338e-04
 3.338e-04  5.739e-02
PARAMETER  CORRELATION COEFFICIENTS
      NO.  GLOBAL      1      2
       1  0.00430   1.000  0.004
       2  0.00430   0.004  1.000
```

Minuit function HESSE

- Purpose: calculate error matrix from $\frac{d^2L}{dp^2}$

```
**********
**   18 **HESSE        1000
**********
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=257.304 FROM HESSE     STATUS=OK                 OTAL
                  EDM=2.36534e-06    STRATE         CURATE
 EXT PARAMETER                                 INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR       STEP SIZE       VALUE
  1  mean         8.84225e-02   3.23861e-01   7.16689e-05   8.84237e-03
  2  sigma        3.20763e+00   2.39539e-01   5.57256e-05   3.26535e-01
                              ERR DEF= 0.5
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 1.049e-01  2.780e-04
 2.780e-04  5.739e-02
PARAMETER  CORRELATION COEFFICIENTS
      NO.  GLOBAL      1      2
       1  0.00358   1.000  0.004
       2  0.00358   0.004  1.000
```

Symmetric errors calculated from 2nd derivative of –ln(L) or χ^2

Minuit function HESSE

- Purpose: calculate error matrix from $\frac{d^2L}{dp^2}$

```
**********
**
***
COV                                   UCCESSFULLY
FCN                                   TUS=OK              10 CALLS          42 TOTAL
                                      4e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
 EX                                                 INTERNAL      INTERNAL
 NO                                    ERROR        STEP SIZE       VALUE
  1                                 3.23861e-01   7.16689e-05   8.84237e-03
  2  sig                 3.20763e+00   2.39539e-01   5.57256e-05   3.26535e-01
                                    ERR DEF= 0.5
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 1.049e-01  2.780e-04
 2.780e-04  5.739e-02
PARAMETER  CORRELATION COEFFICIENTS
       NO.  GLOBAL      1      2
        1  0.00358   1.000  0.004
        2  0.00358   0.004  1.000
```

Error matrix (Covariance Matrix) calculated from

$$V_{ij} = \left(\frac{d^2(-\ln L)}{dp_i dp_j} \right)^{-1}$$

Minuit function HESSE

- Purpose: calculate error matrix from $\frac{d^2L}{dp^2}$

```
**********
**   18 **HESSE        1000
**********
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=257.304 FROM HESSE     STATUS=OK             10 CALLS          42 TOTAL
                    EDM=2.36534e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                   INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR          STEP SIZE       VALUE
  1  mean         8.84225e-02                                   8.84237e-03
  2  sigma        3.20763e+00                                   3.26535e-01

EXTERNAL ERROR MATRIX.    NDIM
 1.049e-01  2.780e-04
 2.780e-04  5.739e-02
PARAMETER  CORRELATION COEFFICIENT
      NO.  GLOBAL      1      2
       1  0.00358   1.000  0.004
       2  0.00358   0.004  1.000
```

Correlation matrix r_{ij} calculated from

$$V_{ij} = s_i s_j r_{ij}$$

Minuit function HESSE

- Purpose: calculate error matrix from $\frac{d^2L}{dp^2}$

```
**********
**   18 **HESSE        1000
**********
COVARIANCE MATRIX CALCULATED SUCCESSFULLY
FCN=257.304 FROM HESSE     STATUS=OK             10 CALLS          42 TOTAL
                    EDM=2.36534e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                   INTERNAL      INTERNAL
 NO.   NAME      VALUE            ERROR          STEP SIZE       VALUE
  1  mean                                       7.16689e-05   8.84237e-03
  2  sigma                                      5.57256e-05   3.26535e-01

EXTERNAL ERROR                                  2    ERR DEF=0.5
 1.049e-01  2.78
 2.780e-04  5.739e-
PARAMETER  CORRELA   N COEFFICIENTS
      NO.  GLOBAL      1      2
       1   0.00358   1.000  0.004
       2   0.00358   0.004  1.000
```

Global correlation vector: correlation of each parameter with *all other* parameters

Minuit function MINOS

- Purpose: More rigorous determination of errors
- Warning: Can be very CPU intensive for large number of parameters
- Optional – activated by option “E” in ROOT or PAW

```
**********
**    23 **MINOS        1000
**********
FCN=257.304 FROM MINOS     STATUS=SUCCESSFUL     52 CALLS          94 TOTAL
                     EDM=2.36534e-06    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                          PARABOLIC        MINOS ERRORS
  NO.   NAME       VALUE                    ERROR      NEGATIVE      POSITIVE
   1  mean          8.84225e-02   3.23861e-01   -3.24688e-01    3.25391e-01
   2  sigma         3.20763e+00   2.39539e-01   -2.23321e-01    2.58893e-01
                                ERR DEF= 0.5
```

Symmetric error

(repeated result from HESSE)

MINOS error
Can be asymmetric

(in this example the ‘sigma’ error is slightly asymmetric)

Practical estimation – Fit converge problems

- Sometimes fits don't converge because, e.g.
  - MIGRAD unable to find minimum
  - HESSE finds negative second derivatives (which would imply negative errors)
- Reason is usually numerical precision and stability problems, but
  - The underlying cause of fit stability problems is usually by **highly correlated parameters** in fit
- HESSE correlation matrix in primary investigative tool

PARAMETER	CORRELATION COEFFICIENTS		
NO.	GLOBAL	1	2
1	0.99835	1.000	0.998
2	0.99835	0.998	1.000

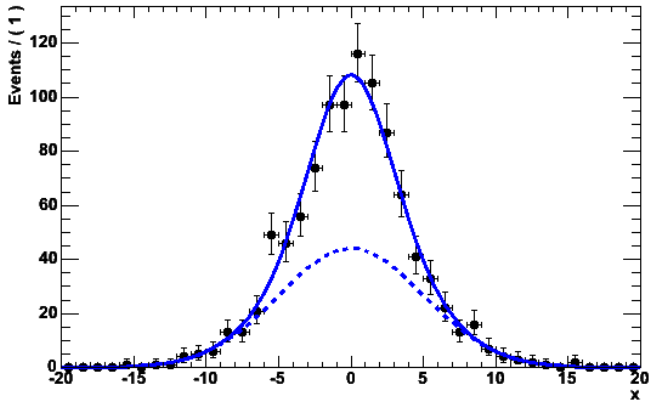
Signs of trouble…

  - In limit of 100% correlation, the usual point solution becomes a line solution (or surface solution) in parameter space. Minimization problem is no longer well defined

Mitigating fit stability problems

- Strategy I – More orthogonal choice of parameters
  - Example: fitting sum of 2 Gaussians of similar width

$$F(x; f, m, s_1, s_2) = fG_1(x; s_1, m) + (1-f)G_2(x; s_2, m)$$



HESSE correlation matrix

Widths s_1,s_2 strongly correlated fraction f

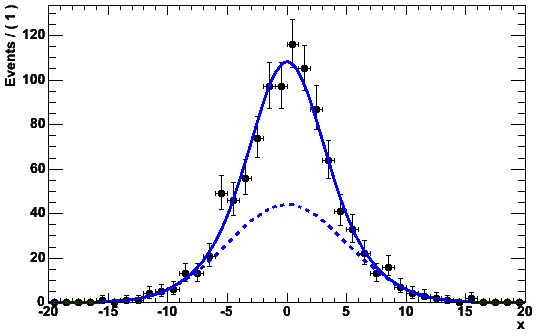
PARAMETER CORRELATION COEFFICIENTS					
NO.	GLOBAL	[f]	[m]	[s1]	[s2]
[f]	0.96973	1.000	-0.135	0.918	0.915
[m]	0.14407	-0.135	1.000	-0.144	-0.114
[s1]	0.92762	0.918	-0.144	1.000	0.786
[s2]	0.92486	0.915	-0.114	0.786	1.000

Mitigating fit stability problems

– Different parameterization:

$$fG_1(x; s_1, m_1) + (1-f)G_2(x; s_1 \cdot s_2, m_2)$$

PARAMETER	CORRELATION COEFFICIENTS				
NO.	GLOBAL	[f]	[m]	[s1]	[s2]
[f]	0.96951	1.000	-0.134	0.917	-0.681
[m]	0.14312	-0.134	1.000	-0.143	0.127
[s1]	0.98879	0.917	-0.143	1.000	-0.895
[s2]	0.96156	0.681	0.127	-0.895	1.000



– Correlation of width s2 and fraction f reduced from 0.92 to 0.68

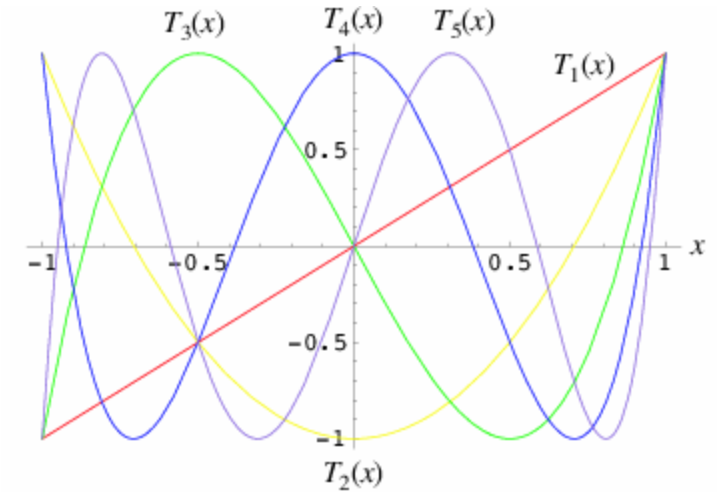
– Choice of parameterization matters!

- Strategy II – Fix all but one of the correlated parameters
  – If floating parameters are highly correlated, some of them may be redundant and not contribute to additional degrees of freedom in your model

Mitigating fit stability problems -- Polynomials

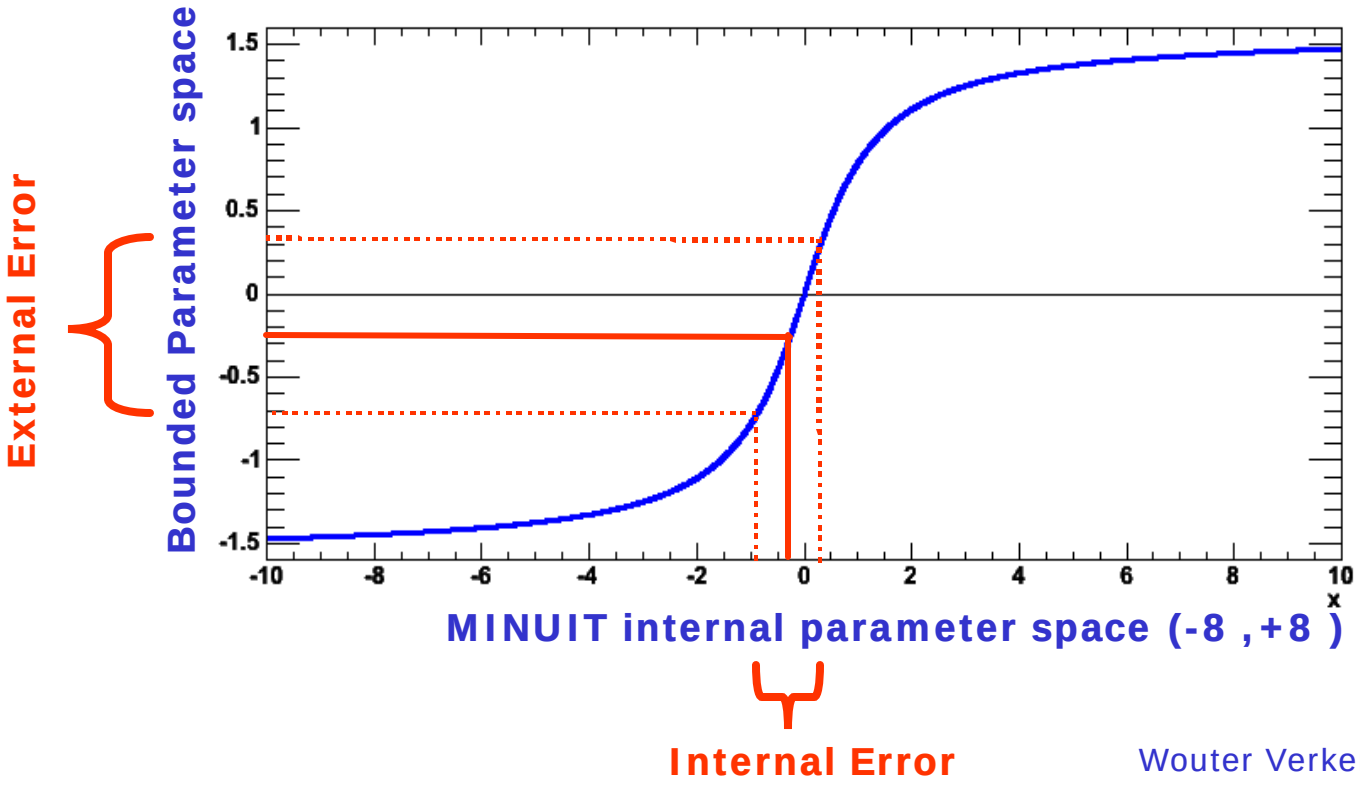
- **Warning:** Regular parameterization of polynomials $a_0+a_1x+a_2x^2+a_3x^3$ nearly always results in strong correlations between the coefficients a_i.
  - *Fit stability problems, inability to find right solution common at higher orders*
- **Solution:** Use existing parameterizations of polynomials that have (mostly) uncorrelated variables
  - ***Example: Chebychev polynomials***

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1.
\end{aligned}
$$



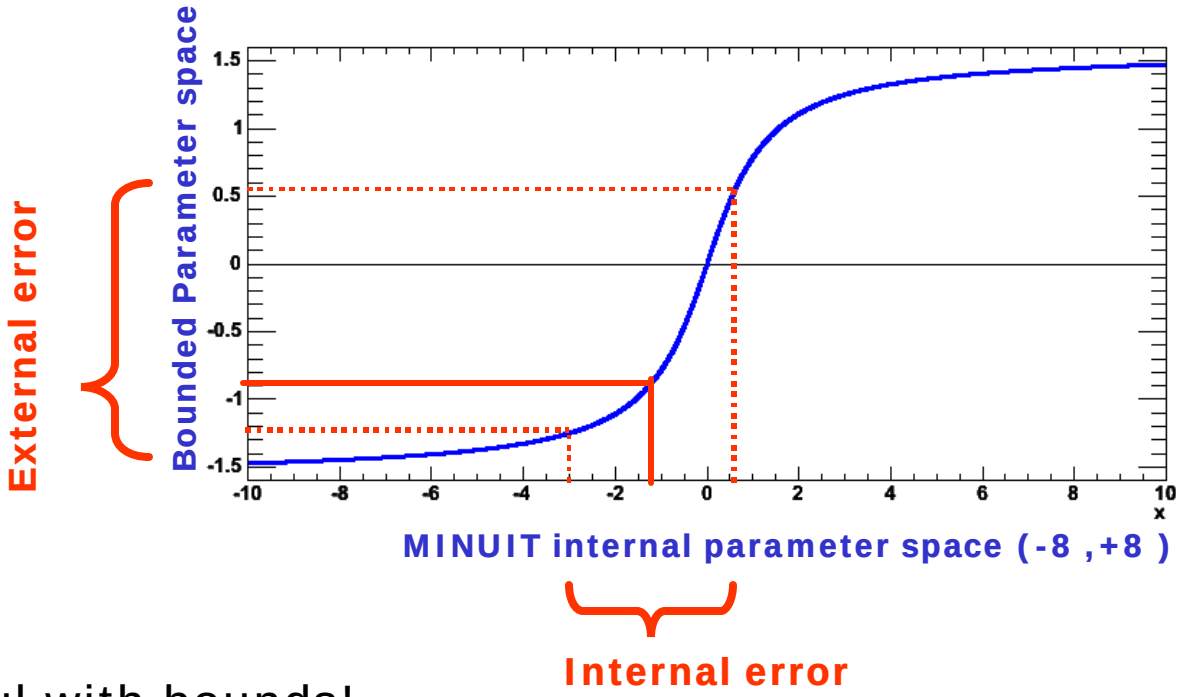
Practical estimation – Bounding fit parameters

- Sometimes is it desirable to bound the allowed range of parameters in a fit
  - Example: a fraction parameter is only defined in the range [0,1]
  - MINUIT option 'B' maps finite range parameter to an internal infinite range using an arcsin(x) transformation:



Practical estimation – Bounding fit parameters

- If fitted parameter values is close to boundary, errors will become asymmetric (and possible incorrect)



- So be careful with bounds!
  - If boundaries are imposed to avoid region of instability, look into other parameterizations that naturally avoid that region
  - If boundaries are imposed to avoid 'unphysical', but statistically valid results, consider not imposing the limit and dealing with the 'unphysical' interpretation in a later stage

Practical Estimation – Verifying the validity of your fit

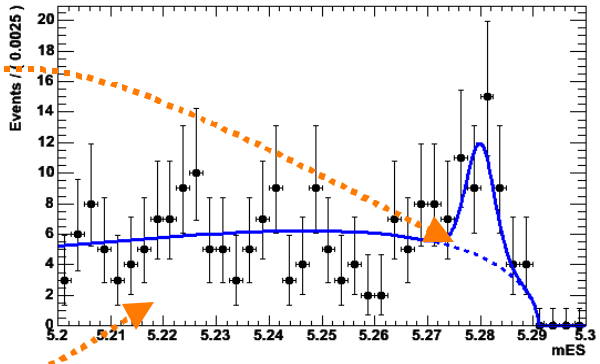
- How to validate your fit? – You want to demonstrate that
  1) Your fit procedure gives on average the correct answer **'no bias'**
  2) The uncertainty quoted by your fit is an accurate measure for the statistical spread in your measurement **'correct error'**

- **Validation is important for low statistics fits**
  - **Correct behavior not obvious a priori due to intrinsic ML bias proportional to 1/N**

- Basic validation strategy – A simulation study
  1) Obtain a large sample of simulated events
  2) Divide your simulated events in O(100-1000) samples with the same size as the problem under study
  3) Repeat fit procedure for each data-sized simulated sample
  4) Compare average value of fitted parameter values with generated value → **Demonstrates (absence of) bias**
  5) Compare spread in fitted parameters values with quoted parameter error → **Demonstrates (in)correctness of error**

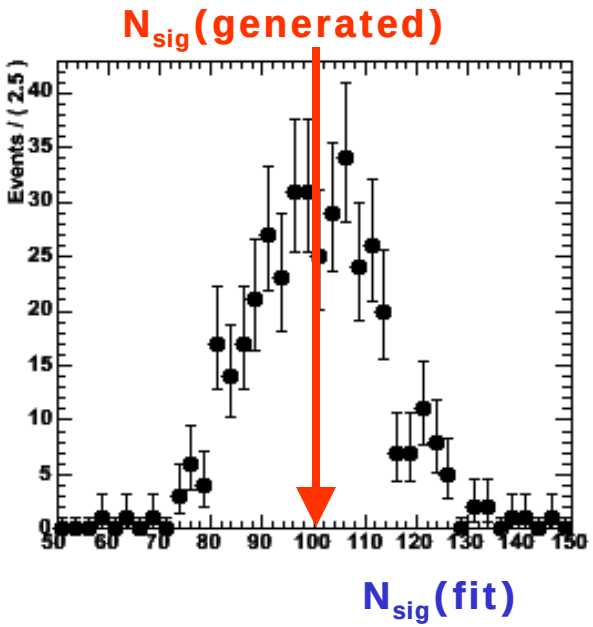
Fit Validation Study – Practical example

- Example fit model in 1-D (B mass)
  - Signal component is Gaussian centered at B mass
  - Background component is Argus function (models phase space near kinematic limit)



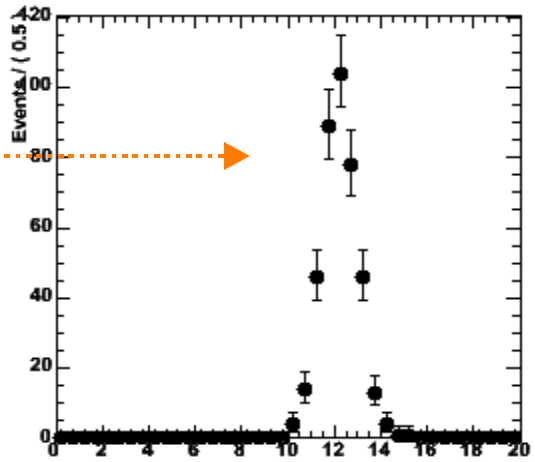
$$F(m; N_{\text{sig}}, N_{\text{bkg}}, \vec{p}_S, \vec{p}_B) = N_{\text{sig}} \cdot G(m; p_S) + N_{\text{bkg}} \cdot A(m; p_B)$$

- Fit parameter under study: N_{sig}
  - Results of simulation study: 1000 experiments with $\mathbf{N_{SIG}(gen)=100}$, $\mathbf{N_{BKG}(gen)=200}$
  - Distribution of N_{sig}(fit)
  - This particular fit looks unbiased…



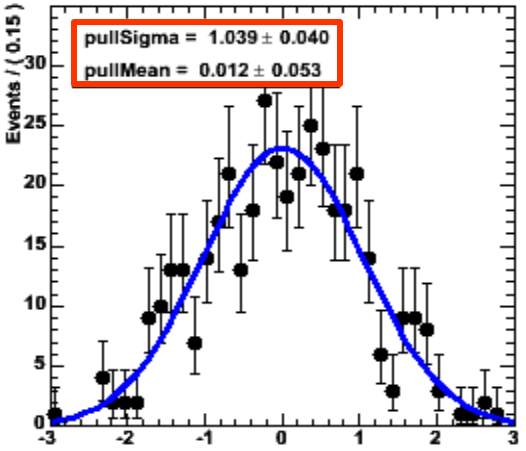
Fit Validation Study – The pull distribution

- What about the validity of the error?
  - Distribution of error from simulated experiments is difficult to interpret…
  - We don't have equivalent of N_{sig}(generated) for the error



$s(N_{sig})$

- Solution: look at the ***pull distribution***
  - Definition: $\text{pull}(N_{sig}) = \dfrac{N_{sig}^{fit} - N_{sig}^{true}}{s_N^{fit}}$
  - Properties of pull:
    - **Mean is 0 if there is no bias**
    - **Width is 1 if error is correct**
  - In this example: no bias, correct error within statistical precision of study .....



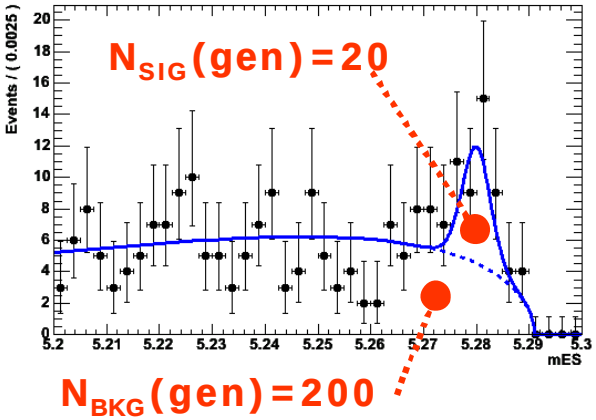
pull(N_{sig})

Fit Validation Study – Low statistics example

- Special care should be taken when fitting small data samples
  - Also if fitting for small signal component in large sample
- Possible causes of trouble
  - χ^2 estimators may become approximate as Gaussian approximation of Poisson statistics becomes inaccurate
  - ML estimators may no longer be efficient
    → error estimate from 2nd derivative may become inaccurate
  - Bias term proportional to 1/N of ML and χ^2 estimators may no longer be small compared to 1/sqrt(N)
- In general, absence of bias, correctness of error can not be assumed. How to proceed?
  - Use unbinned ML fits only – most robust at low statistics
  - **Explicitly verify the validity of your fit**

Demonstration of fit bias at low N – pull distributions

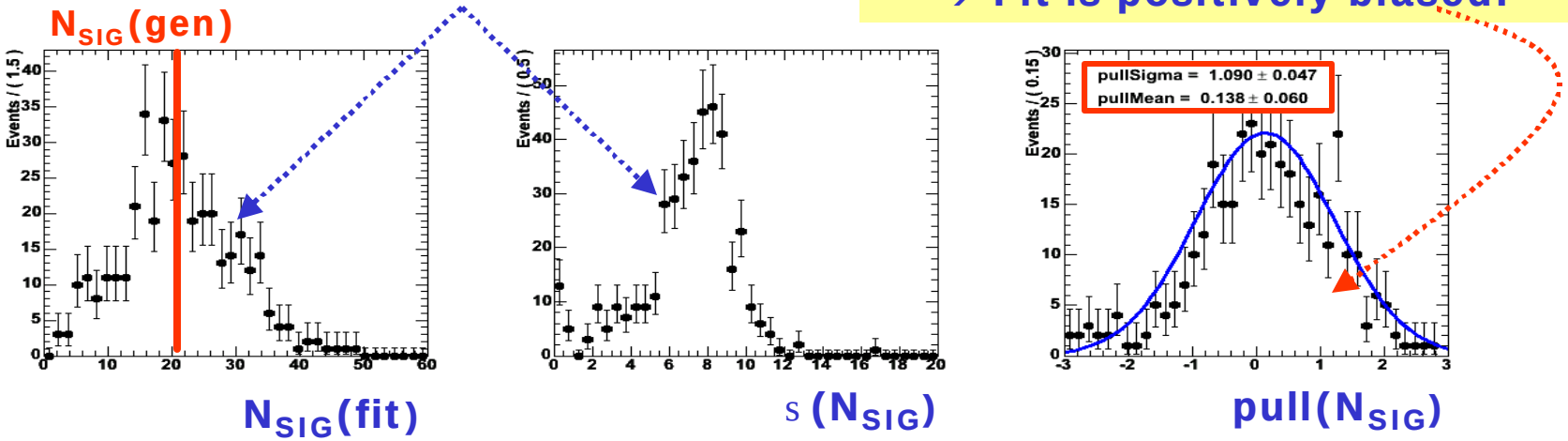
- Low statistics example:
  - Scenario as before but now with 200 bkg events and **only 20 signal events** (instead of 100)



- Results of simulation study

Distributions become asymmetric at low statistics

Pull mean is 2.3s away from 0 → Fit is positively biased!



- *Absence of bias, correct error at low statistics not obvious!*
  - *Small yields are typically overestimated*

Fit Validation Study – How to obtain 10.000.000 simulated events?

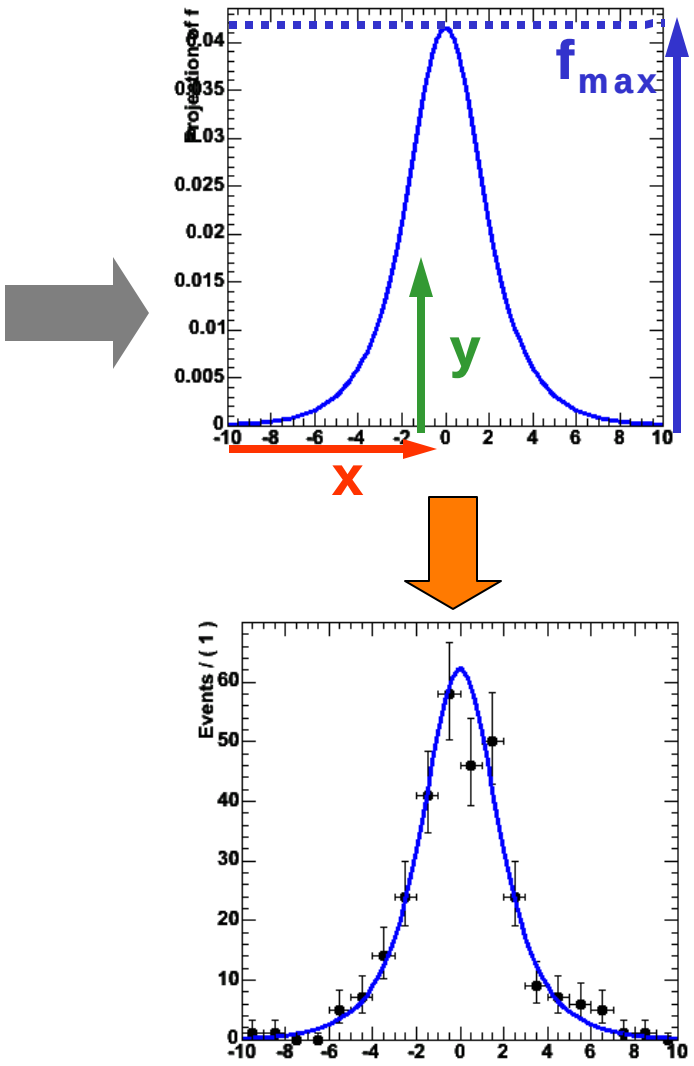
- Practical issue: usually you need very large amounts of simulated events for a fit validation study
  - Of order 1000x number of events in your fit, easily >1.000.000 events
  - Using data generated through a full GEANT-based detector simulation can be prohibitively expensive

- Solution: Use events sampled directly from your fit function
  - Technique named '*Toy Monte Carlo*' sampling
  - Advantage: Easy to do and very fast
  - Good to determine fit bias due to low statistics, choice of parameterization, boundary issues etc
  - Cannot be used to test assumption that went into model (e.g. absence of certain correlations). Still need full GEANT-based simulation for that.

Toy MC generation – Accept/reject sampling

- *How to sample events directly from your fit function?*
- Simplest: 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)**



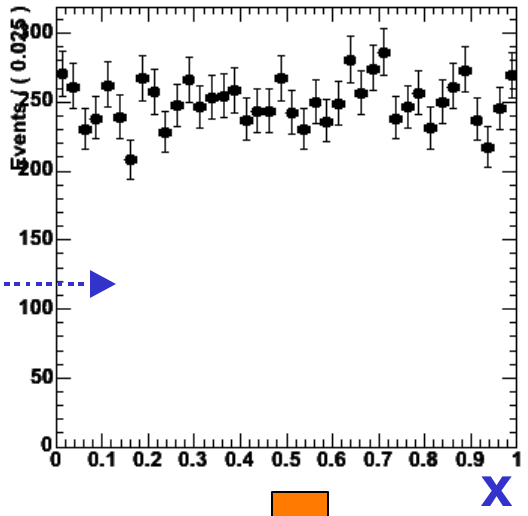
– PRO: Easy, always works

– CON: It can be inefficient if function is strongly peaked.
Finding maximum empirically through random sampling can be lengthy in >2 dimensions

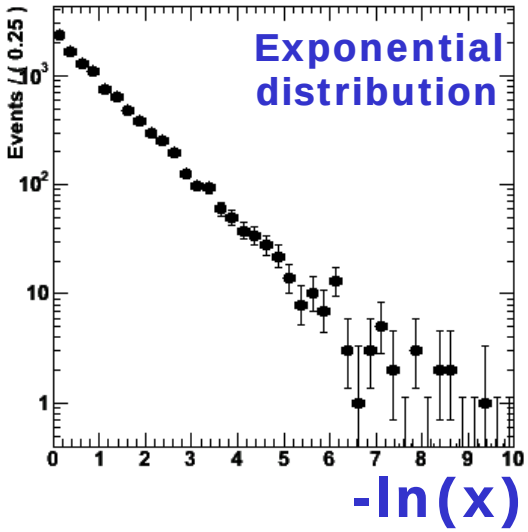
Toy MC generation – Inversion method

- 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)

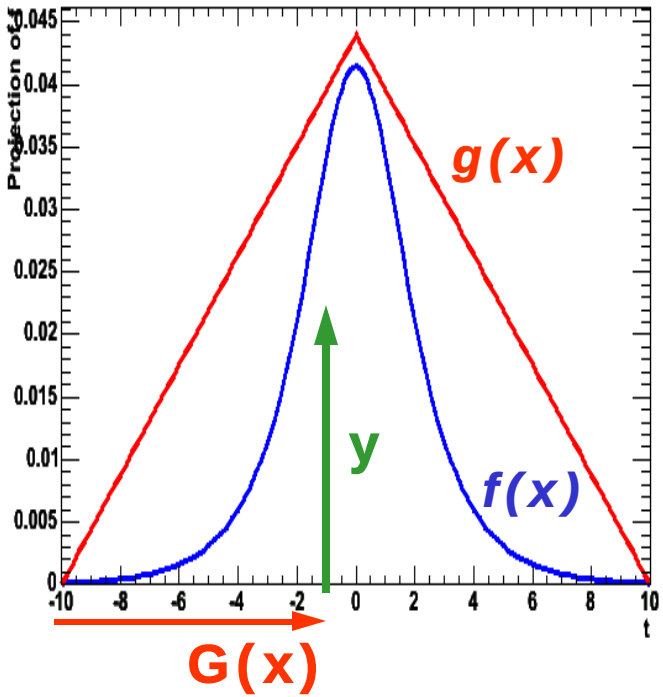


- PRO: Maximally efficient
- CON: Only works for invertible functions

Toy MC Generation in a nutshell

- Hybrid: Importance sampling

1) Find 'envelope function' g(x) that is invertible into G(x) and that fulfills g(x)>=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

Multi-dimensional fits – Benefit analysis

- Fits to multi-dimensional data sets offer opportunities but also introduce several headaches

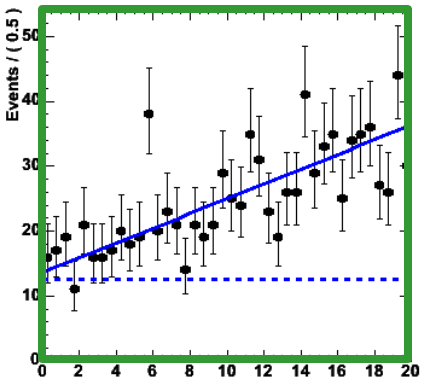
Pro

- Enhanced in sensitivity because more data and information is used simultaneously
- Exploit information in correlations between observables

Con

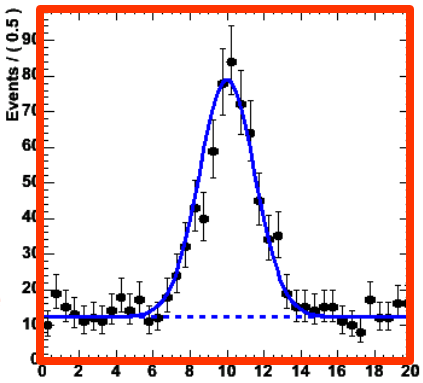
- More difficult to visualize model, model-data agreement
- More room for hard-to-find problems
- Just a lot more work

- It depends very much on your particular analysis if fitting a variable is better than cutting on it



← **No obvious cut, may be worthwile to include in n-D fit**

Obvious where to cut, probably not worthwile to include in n-D fit →



Ways to construct a multi-D fit model

- Simplest way: take product of N 1-dim models, e.g

$$FG(x,y) = F(x) \cdot G(y)$$

  - Assumes x and y are uncorrelated in data. If this assumption is unwarranted you may get a wrong result: Think & Check!

- Harder way: explicitly model correlations by writing a 2-D model

$$H(x,y) = \exp\left[-\left((x+y)/2\right)^2\right]$$

- Hybrid approach:
  - Use conditional probabilities

$$FG(x,y) = F(x \mid y) \cdot G(y)$$

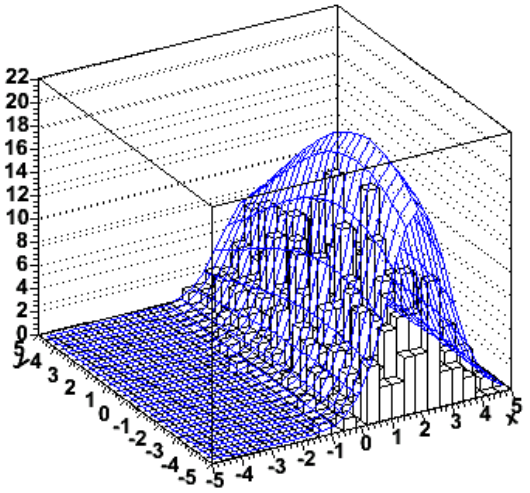
Probability for y $\int G(y)dy \equiv 1$

Probability for x, given a value of y

$$\int F(x,y)dx \equiv 1 \quad \text{for all values of y}$$

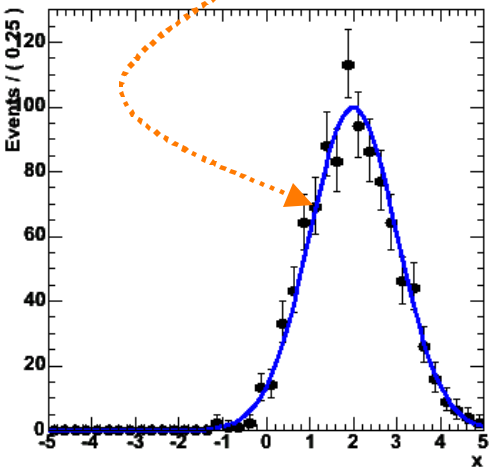
Multi-dimensional fits – visualizing your model

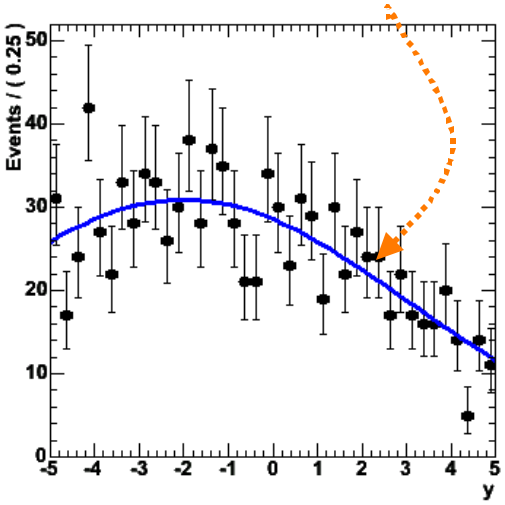
- Overlaying a 2-dim PDF with a 2D (lego) data set doesn't provide much insight



- 1-D projections usually easier

$$f_y(x) = \int F(x,y)dy \qquad f_x(y) = \int F(x,y)dx$$



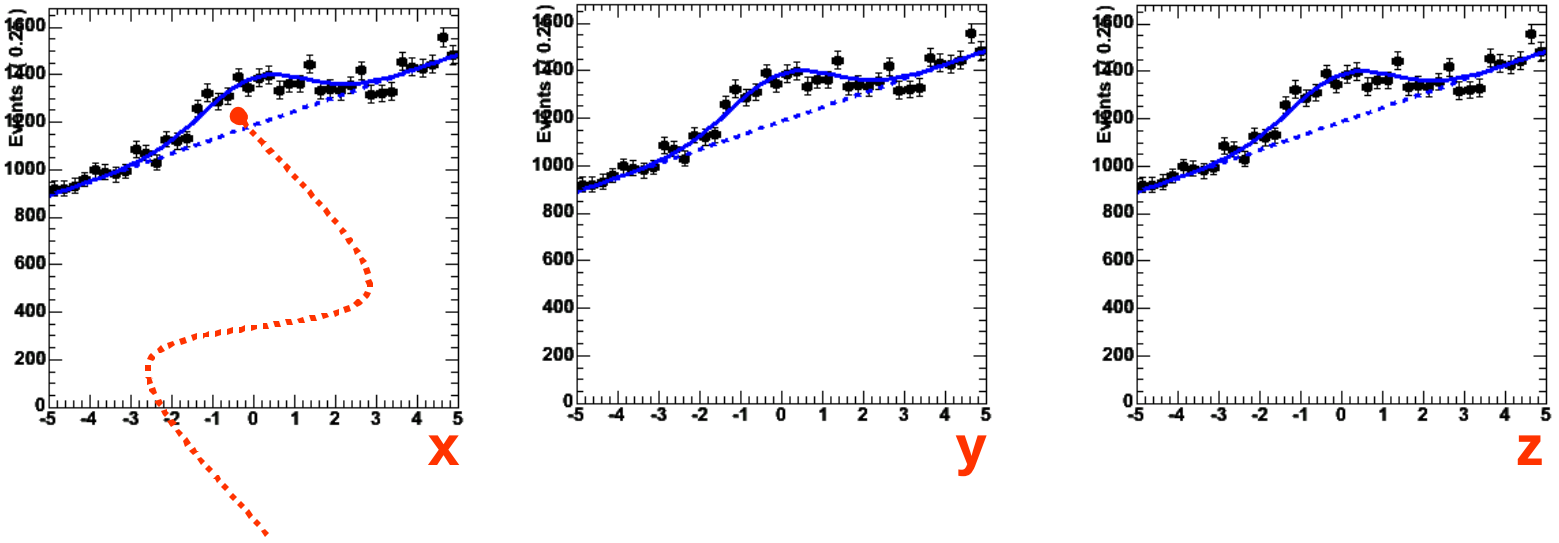


x-y correlations in data and / or model difficult to visualize

Multi-dimensional fits – visualizing your model

- However: plain 1-D projections often don't do justice to your fit
  - Example: 3-Dimensional dataset with 50K events, 2500 signal events
  - Distributions in x,y and z chosen identical for simplicity

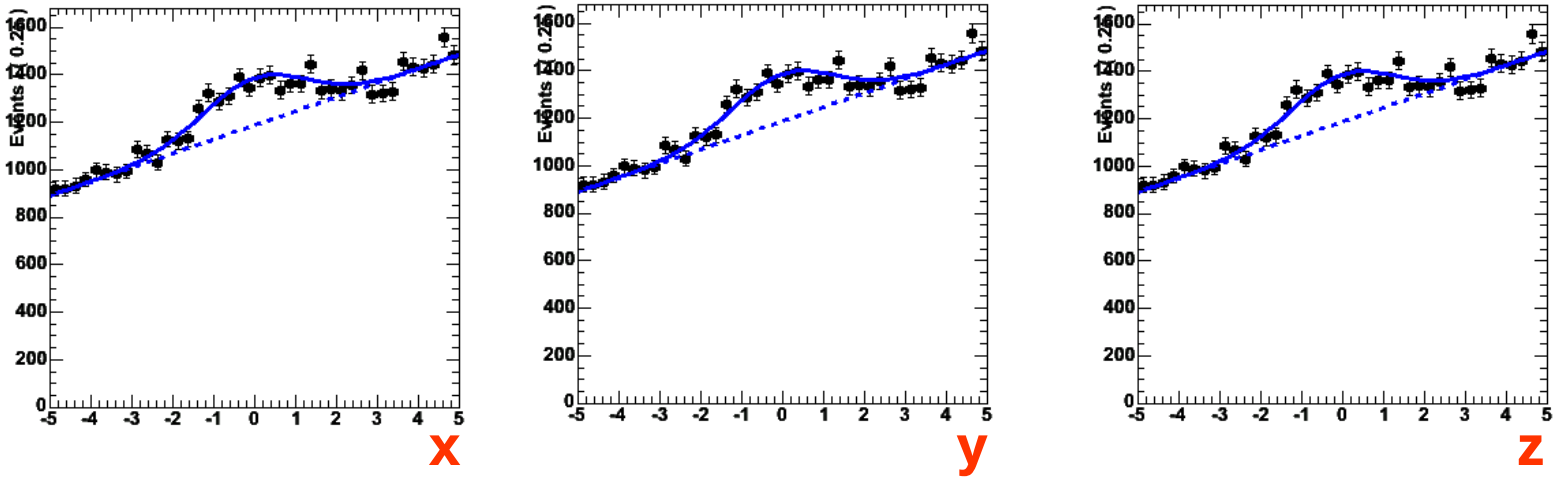
- Plain 1-dimensional projections in x,y,z



- Fit of 3-dimensional model finds $N_{sig} = 2440 \pm 64$
  - Difficult to reconcile with enormous backgrounds in plots

Multi-dimensional fits – visualizing your model

- Reason for discrepancy between precise fit result and large background in 1-D projection plot
  - Events in shaded regions of y,z projections can be discarded without loss of signal



- Improved projection plot: show only events in x projection that are likely to be signal in (y,z) projection of fit model
  - Zeroth order solution: make box cut in (x,y)
  - Better solution: cut on signal probability according to fit model in (y,z)

Multi-dimensional fits – visualizing your model

- Goal: Projection of model and data on x, with a cut on the signal probability in (y,z)
- First task at hand: calculate signal probability according to PDF using only information in (y,z) variables
  - Define 2-dimensional signal and background PDFs in (y,z) by integrating out x variable (and thus discarding any information contained in x dimension)

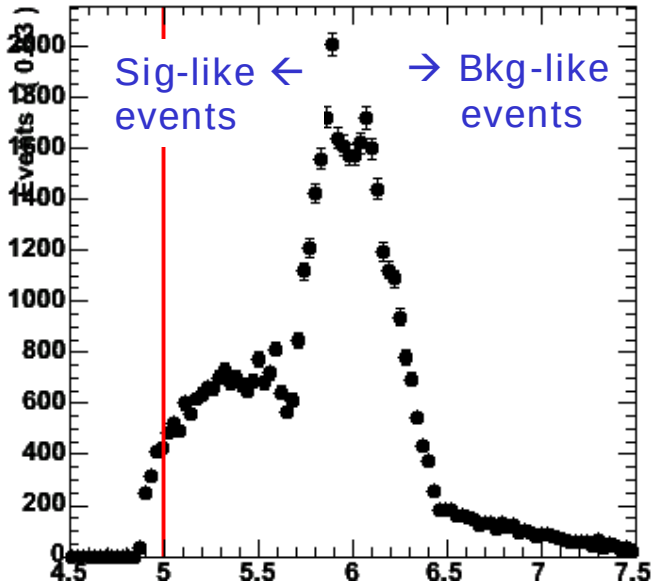
$$F_{SIG}(y,z) = \int S(x,y,z)dx$$

$$F_{BKG}(y,z) = \int B(x,y,z)dx$$

  - Calculate signal probability P(y,z) for all data points (x,y,z)

$$P_{SIG}(y,z) = \frac{F_{SIG}(y,z)}{F_{SIG}(y,z) + F_{BKG}(y,z)}$$

  - Choose sensible cut on P(y,z)

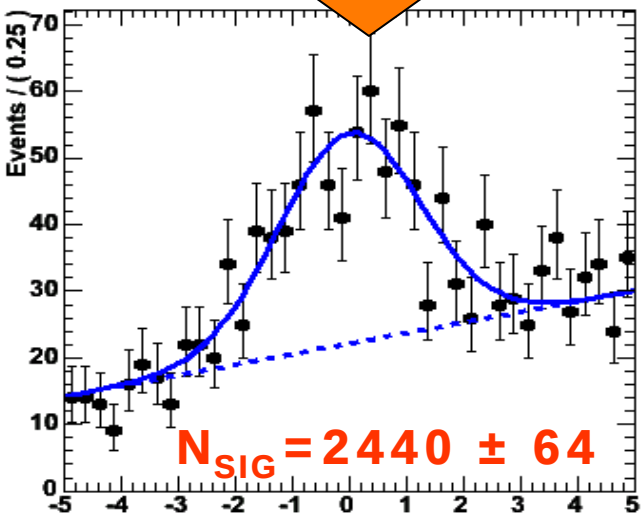


$-\log(P_{SIG}(y,z))$

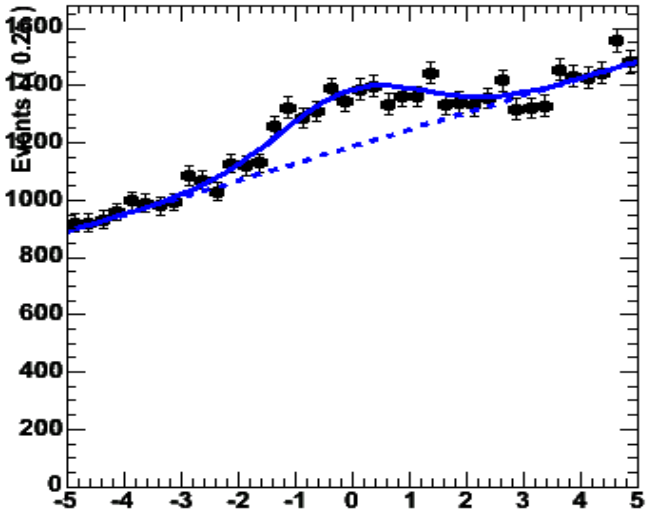
Plotting regions of a N-dim model – Case study

- Next: plot distribution of data, model with cut on $P_{SIG}(y,z)$
  - Data: Trivial
  - Model: Calculate projection of selected regions with Monte Carlo method
    1) **Generate a toy Monte Carlo dataset $D_{TOY}(x,y,z)$ from $F(x,y,z)$**
    2) **Select subset of D_{TOY} with $P_{SIG}(y,z)<C$**
    3) **Plot** $f_C(x) = \sum_{D_{TOY}} F(x, y_i, z_i)$

Likelihood ratio projection



Plain projection (for comparison)



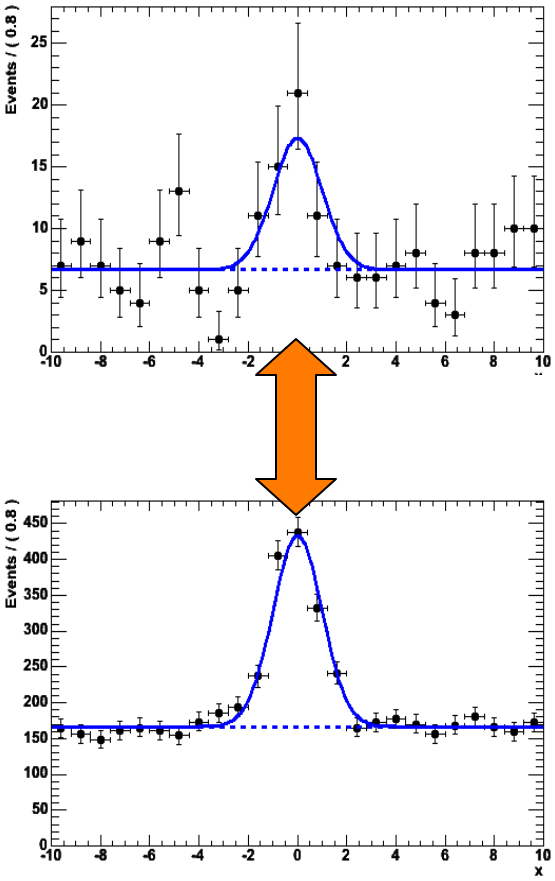
Multidimensional fits – Goodness-of-fit determination

- Warning: Goodness-of-fit measures for multi-dimensional fits are difficult
  - Standard χ^2 test does not work very will in N-dim because of natural occurrence of large number of empty bins
  - Simple equivalent of (unbinned) Kolmogorov test in >1-D does not exist

- This area is still very much a work in progress
  - Several new ideas proposed but sometimes difficult to calculate, or not universally suitable
  - Some examples
    - Cramer-von Mises (close to Kolmogorov in concept)
    - Anderson-Darling
    - 'Energy' tests
  - **No magic bullet here**
  - Some references to recent progress:
    - PHYSTAT2001, PHYSTAT2003

Practical fitting – Error propagation between samples

- Common situation: you want to fit a small signal in a large sample
  - Problem: small statistics does not constrain shape of your signal very well
  - Result: errors are large



- Idea: Constrain shape of your signal from a fit to a control sample
  - Larger/cleaner data or MC sample with similar properties

- Needed: a way to propagate the information from the control sample fit (parameter values *and* errors) to your signal fit

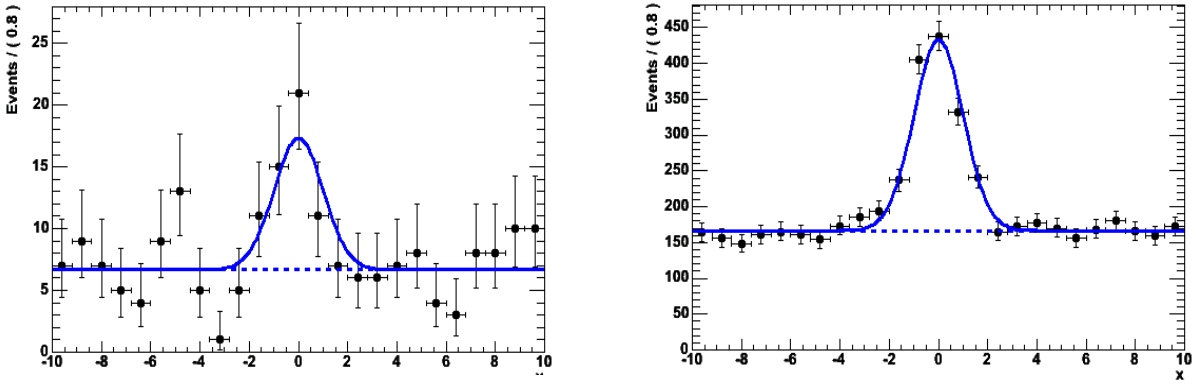
Practical fitting – Error propagation between samples

- 0th order solution:
  - **Fit control sample first, signal sample second** – signal shape parameters fixed from values of control sample fit
  - Signal fit will give correct parameter estimates
  - But error on signal will be underestimated because uncertainties in the determination of the signal shape from the control sample are not included
- 1st order solution
  - **Repeat fit on signal sample at $p \pm s_p$**
  - Observe difference in answer and add this difference in quadrature to error: $$s^2_{tot} = s^2_{stat} + (N^{p-s_p}_{sig} - N^{p+s_p}_{sig})^2 / 2$$
  - Problem: Error estimate will be incorrect if there is >1 parameter in the control sample fit and there are correlations between these parameters
- Best solution: a simultaneous fit

Practical fitting – Simultaneous fit technique

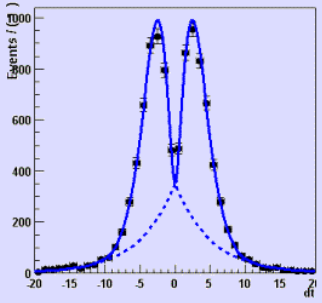
- given data $D_{sig}(\mathbf{x})$ and model $F_{sig}(\mathbf{x};\mathbf{p_{sig}})$ and data $D_{ctl}(\mathbf{x})$ and model $F_{ctl}(\mathbf{x};\mathbf{p_{ctl}})$
  - construct $\chi^2_{sig}(\mathbf{p_{sig}})$ and $\chi^2_{ctl}(\mathbf{p_{ctl}})$ and

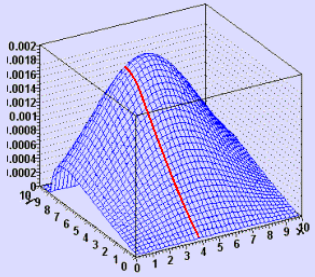
$D_{sig}(x)$, $F_{sig}(x;p_{sig})$ $D_{ctl}(x)$, $F_{ctl}(x;p_{ctl})$

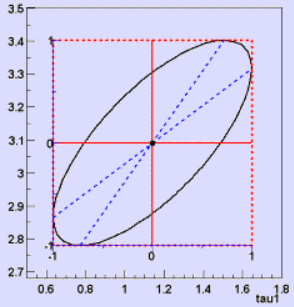


- Minimize $\chi^2(\mathbf{p_{sig}},\mathbf{p_{ctl}}) = \chi^2_{sig}(\mathbf{p_{sig}}) + \chi^2_{ctl}(\mathbf{p_{ctl}})$
  - All parameter errors, correlations automatically propagated

Commercial Break







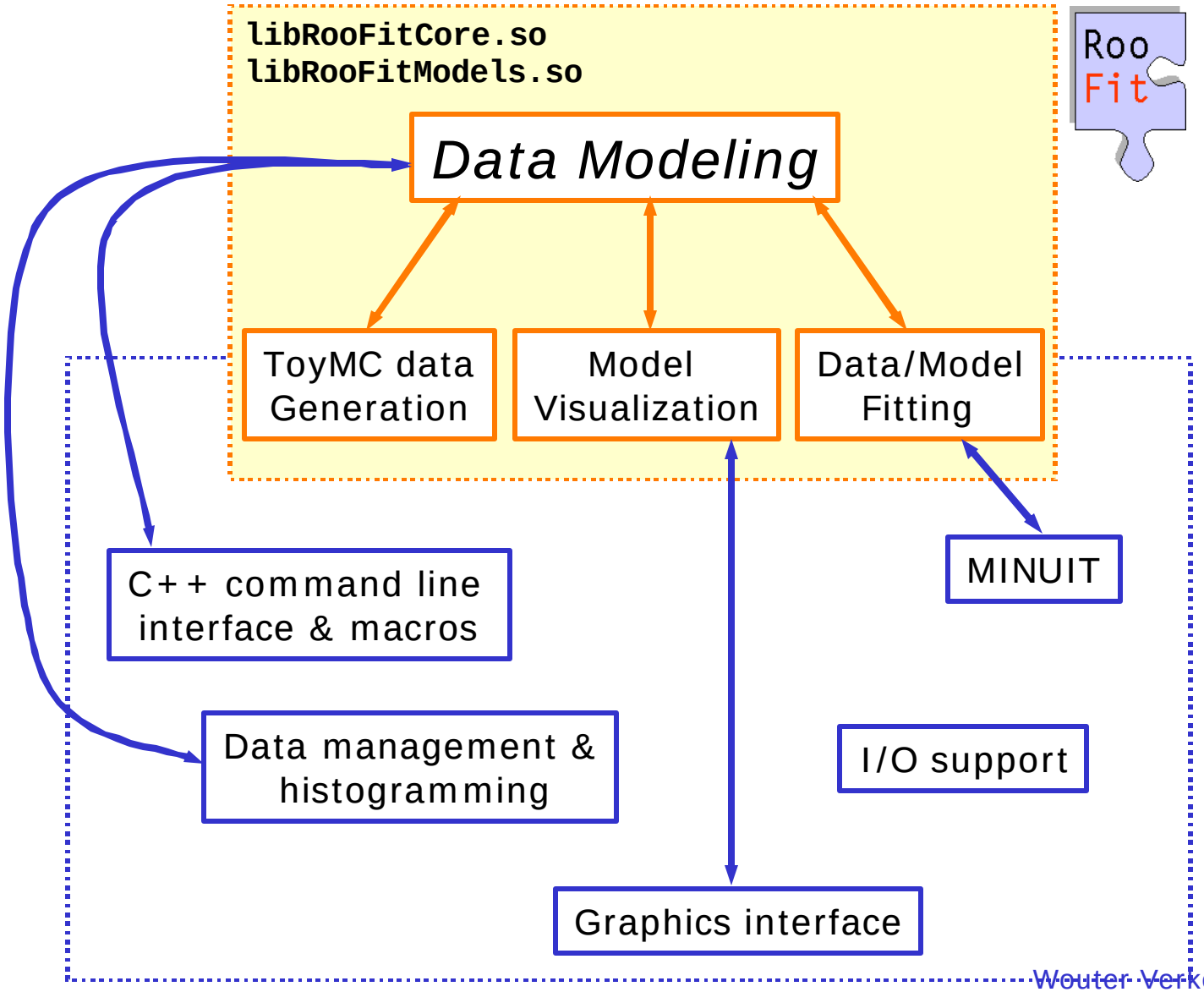
This course comes with free software that helps you do many labor intensive analysis and fitting tasks much more easily

RooFit

A general purpose tool kit for data modeling

Wouter Verkerke (UC Santa Barbara)
David Kirkby (UC Irvine)

Implementation – Add-on package to ROOT



Data modeling – OO representation

- Mathematical objects are represented as C++ objects

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

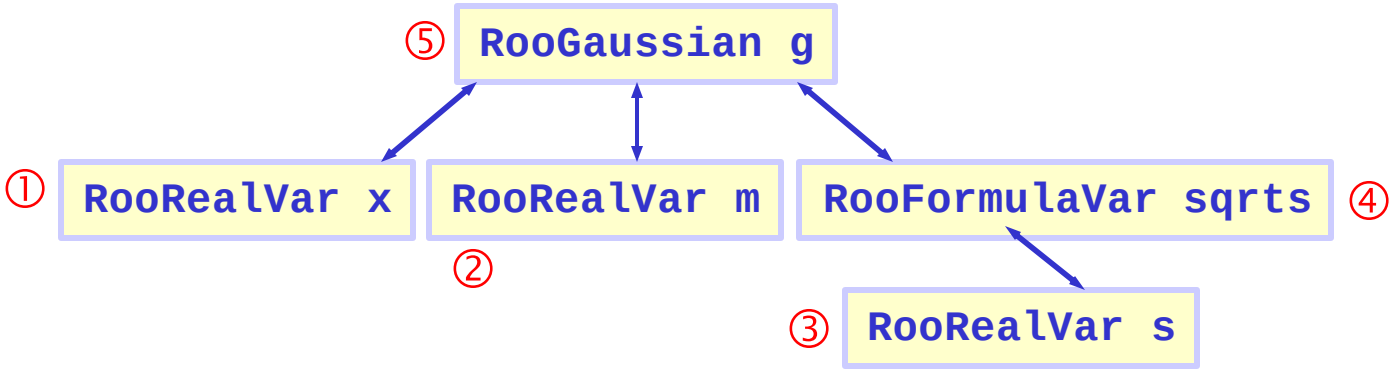
Data modeling – Constructing composite objects

- Straightforward correlation between mathematical representation of formula and RooFit code

Math

$$G(x, m, \sqrt{s})$$

RooFit diagram

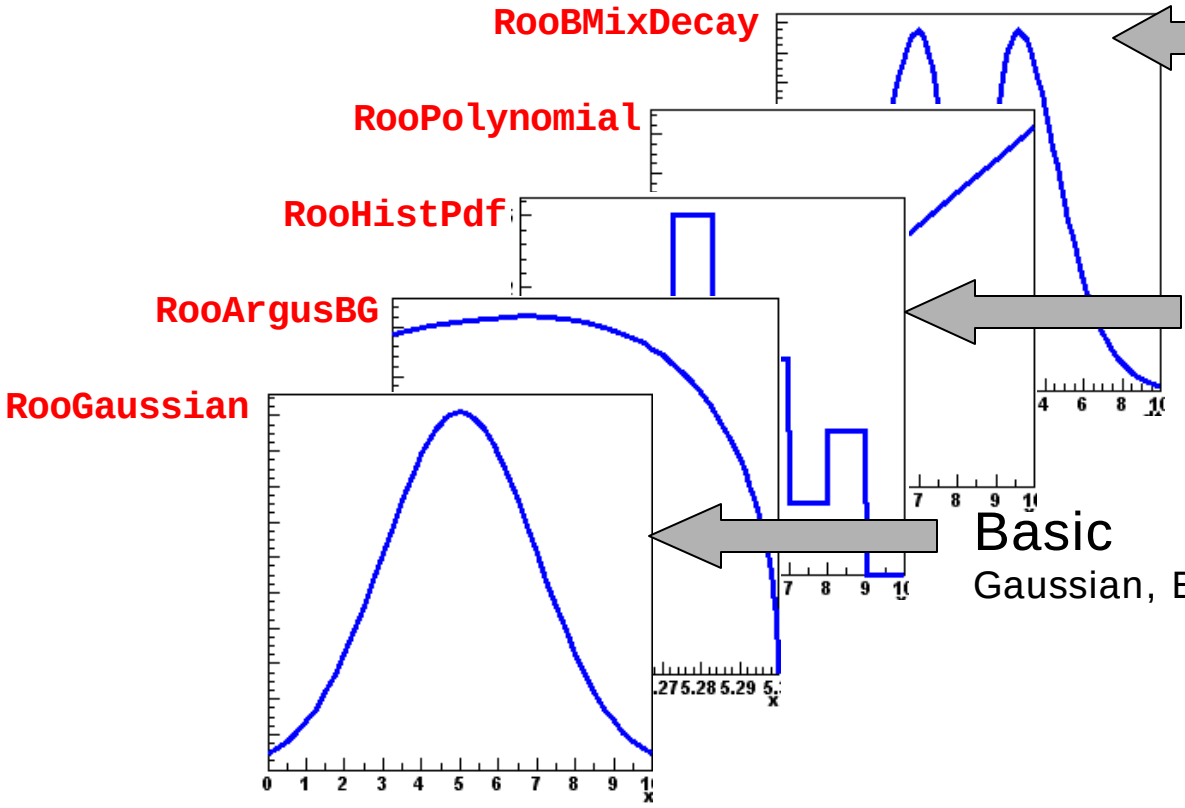


RooFit code

```
① RooRealVar x("x","x",-10,10) ;
② RooRealVar m("m","mean",0) ;
③ RooRealVar s("s","sigma",2,0,10) ;
④ RooFormulaVar sqrts("sqrts","sqrt(s)",s) ;
⑤ RooGaussian g("g","gauss",x,m,sqrts) ;
```

Model building – (Re)using standard components

- RooFit provides a collection of compiled standard PDF classes



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

Non-parametric
Histogram, KEYS
(Probability Density Estimate)

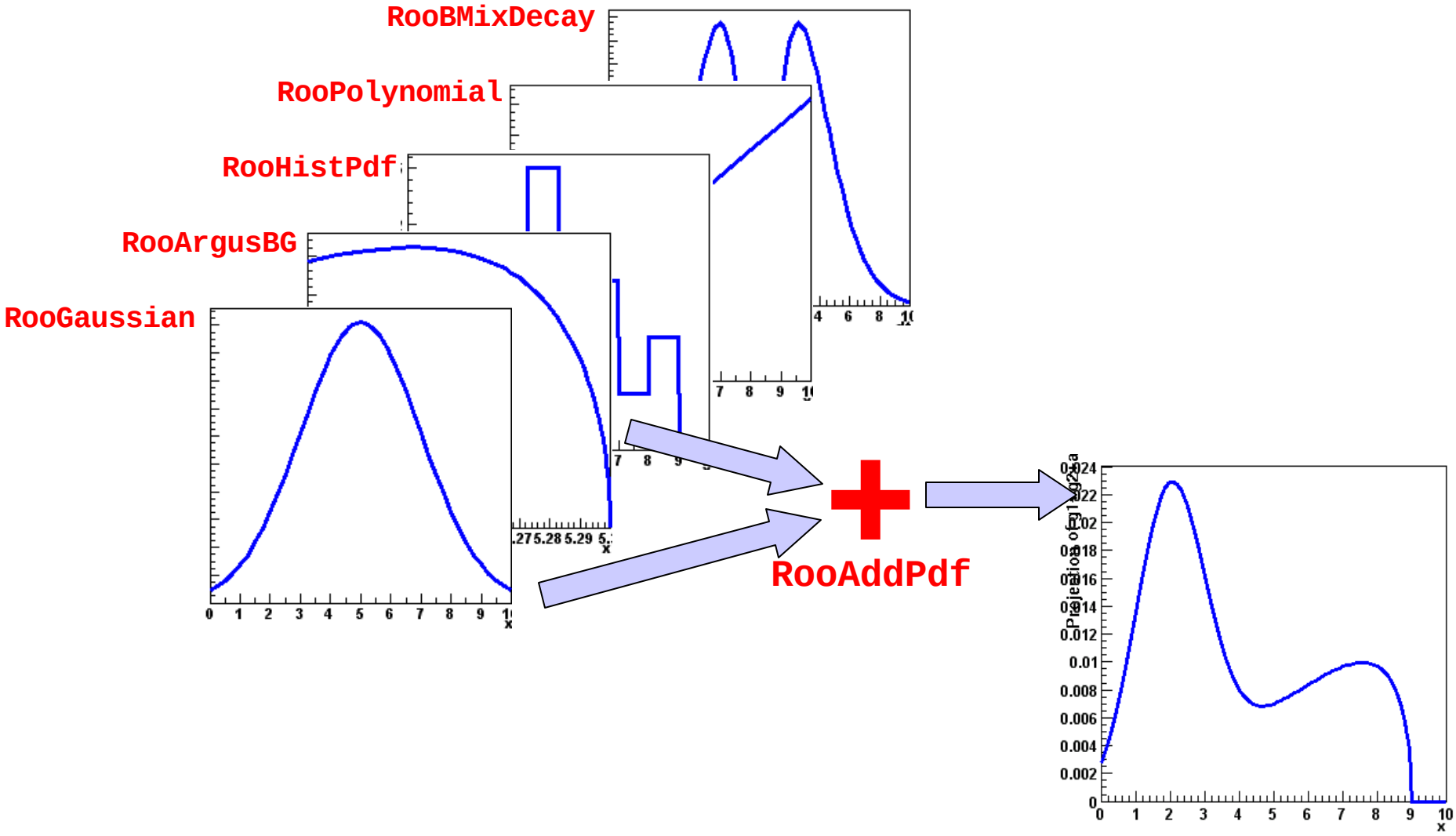
Basic
Gaussian, Exponential, Polynomial,...

PDF Normalization

- By default RooFit uses numeric integration to achieve normalization
- Classes can optionally provide (partial) analytical integrals
- Final normalization can be hybrid numeric/analytic form

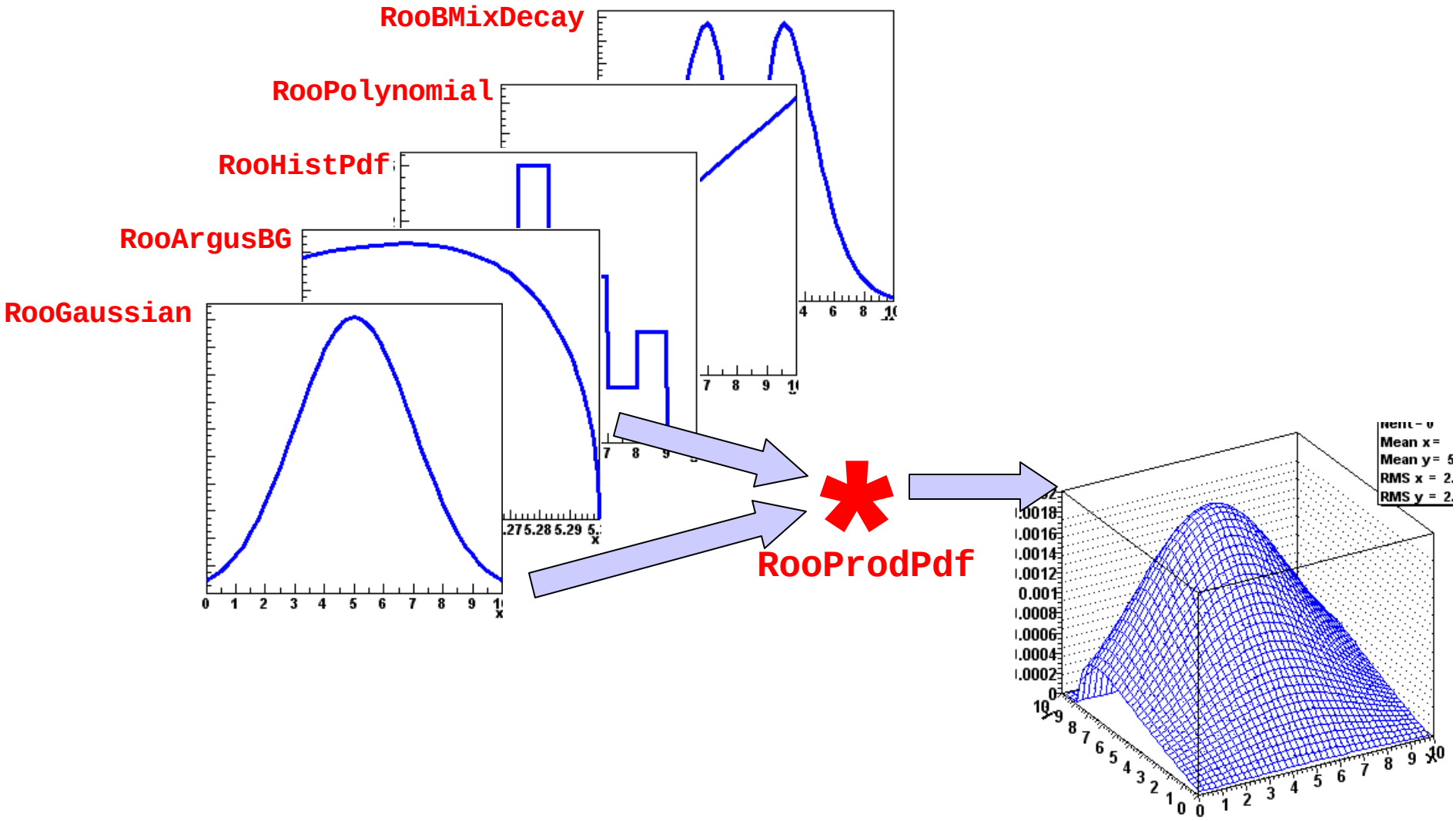
Model building – (Re)using standard components

- Most physics models can be composed from ‘basic’ shapes



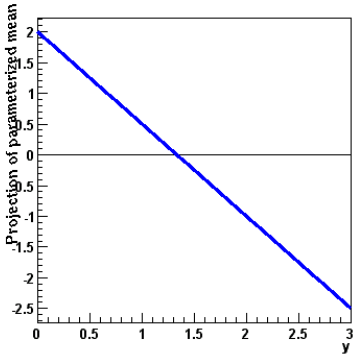
Model building – (Re)using standard components

- Most physics models can be composed from ‘basic’ shapes

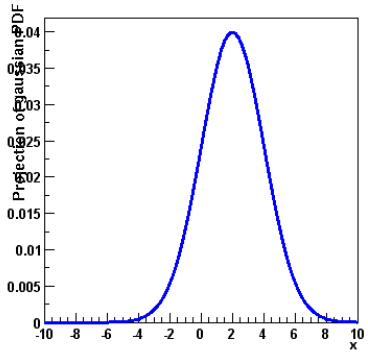


Model building – (Re)using standard components

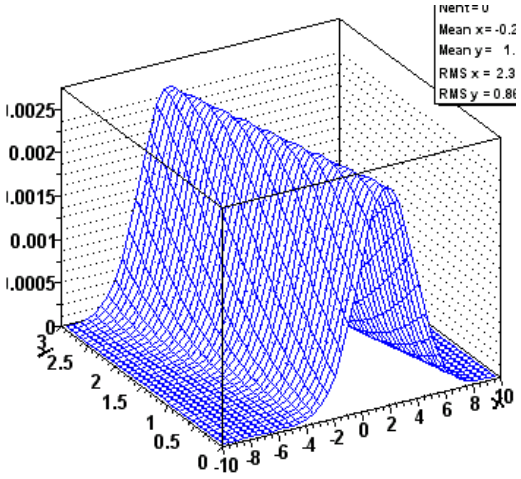
- Building blocks are *flexible*
  - Function *variables can be functions* themselves
  - Just plug in *anything* you like
  - Universally supported by core code
    (PDF classes don't need to implement special handling)



$m(y;a_0,a_1)$



$g(x;m,s)$

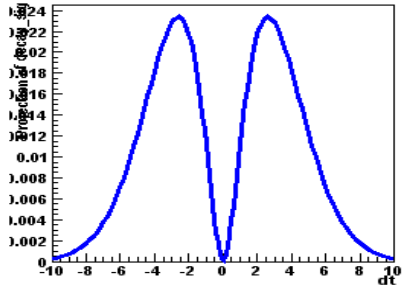


$g(x,y;a_0,a_1,s)$

```
RooPolyVar  m("m",y,RooArgList(a0,a1)) ;
RooGaussian g("g","gauss",x,m,s) ;
```

Model building – Expression based components

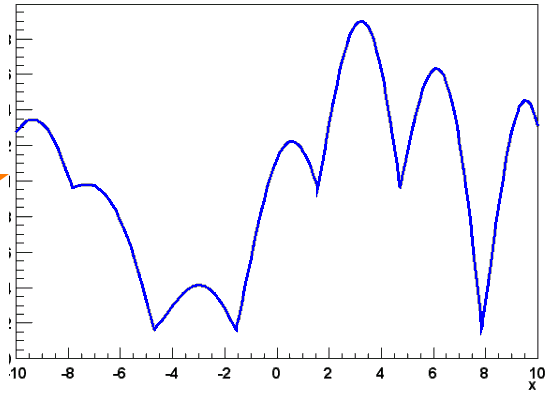
- **RooFormulaVar** – Interpreted real-valued function
  - Based on ROOT **TFormula** class
  - Ideal for modifying parameterization of existing compiled PDFs



RooBMixDecay(t,tau,w,...)

RooFormulaVar w("w","1-2*D",D) ;

- **RooGenericPdf** – Interpreted PDF
  - Based on ROOT **TFormula** class
  - User expression doesn't need to be normalized
  - Maximum flexibility




```
RooGenericPdf f("f","1+sin(0.5*x)+abs(exp(0.1*x)*cos(-1*x))",x)
```

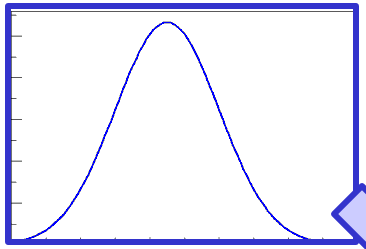
Using models - Overview

- *All* RooFit models provide *universal and complete fitting* and Toy Monte Carlo *generating* functionality
  - Model complexity only limited by available memory and CPU power
    - models with >16000 components, >1000 fixed parameters and>80 floating parameters have been used (published physics result)
  - Very easy to use – Most operations are one-liners

Fitting

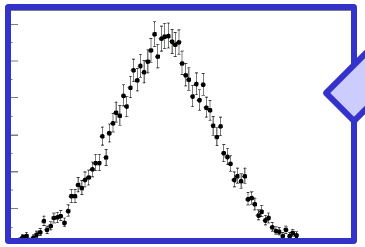
Generating

```
data = gauss.generate(x,1000)
```

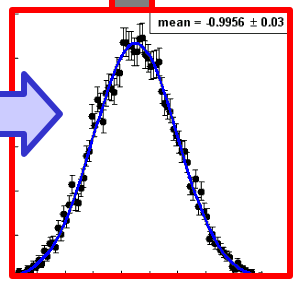


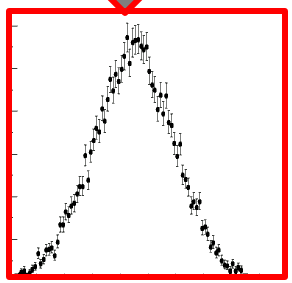
RooAbsPdf

```
gauss.fitTo(data)
```



RooAbsData

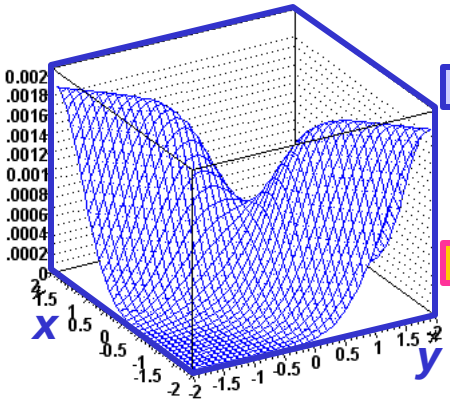




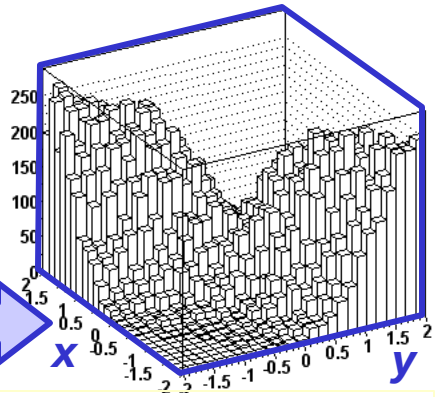
RooDataSet

Using models – Toy MC Generation

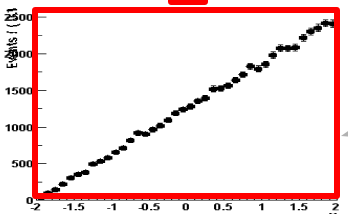
- Generate "Toy" Monte Carlo samples from *any* PDF
  - Sampling method used by default, but PDF components can advertise alternative (more efficient) generator methods
  - No limit to number of dimensions, discrete-valued dimensions also supported

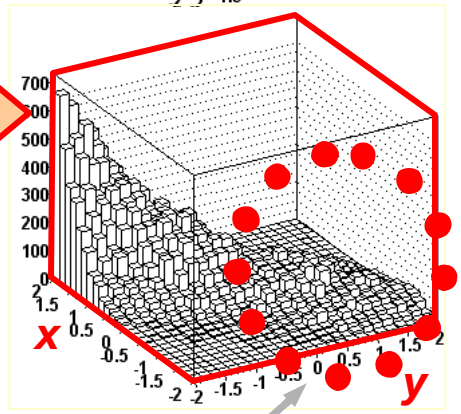


`data=pdf.generate(x,y,1000)`



`data=pdf.generate(x,ydata)`

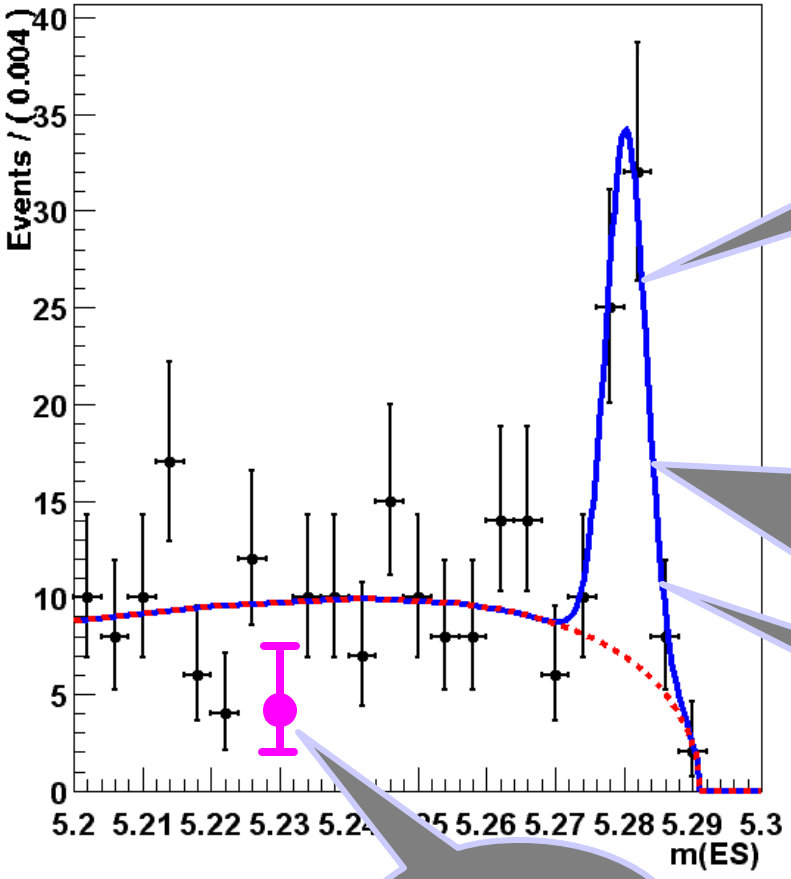




  - **Subset of variables can be taken from a prototype dataset**
    - E.g. to more accurately model the statistical fluctuations in a particular sample.
    - ***Correlations*** with prototype observables ***correctly taken into account***

Using models – Plotting

- **RooPlot** – View of 31 datasets/PDFs projected on the same dimension



Curve always normalized to last plotted dataset in **frame**

For multi-dimensional PDFs: appropriate 1-dimensional projection is automatically created:

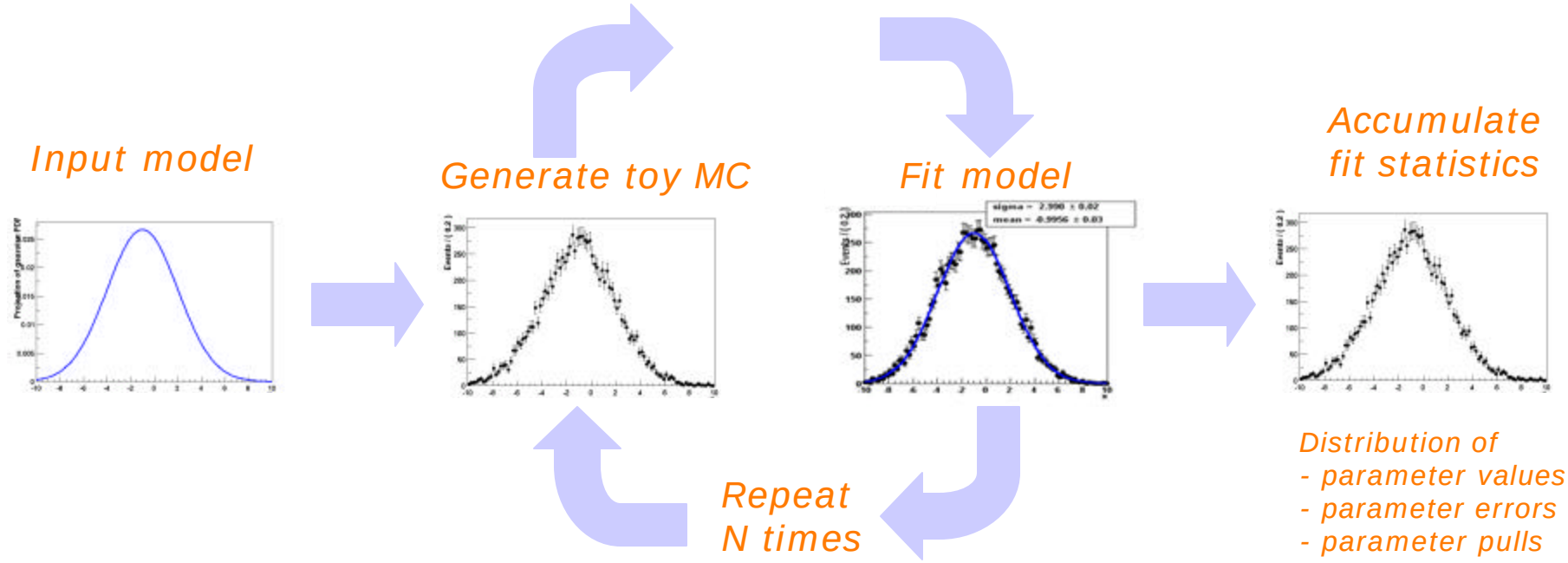
$$\text{Projection}[F](x) = N \cdot \frac{\int F(x,\vec{y})d\vec{y}}{\int F(x,\vec{y})dxd\vec{y}}$$

Adaptive spacing of curve points to achieve 1‰ precision, regardless of data binning

Poisson errors on histogram

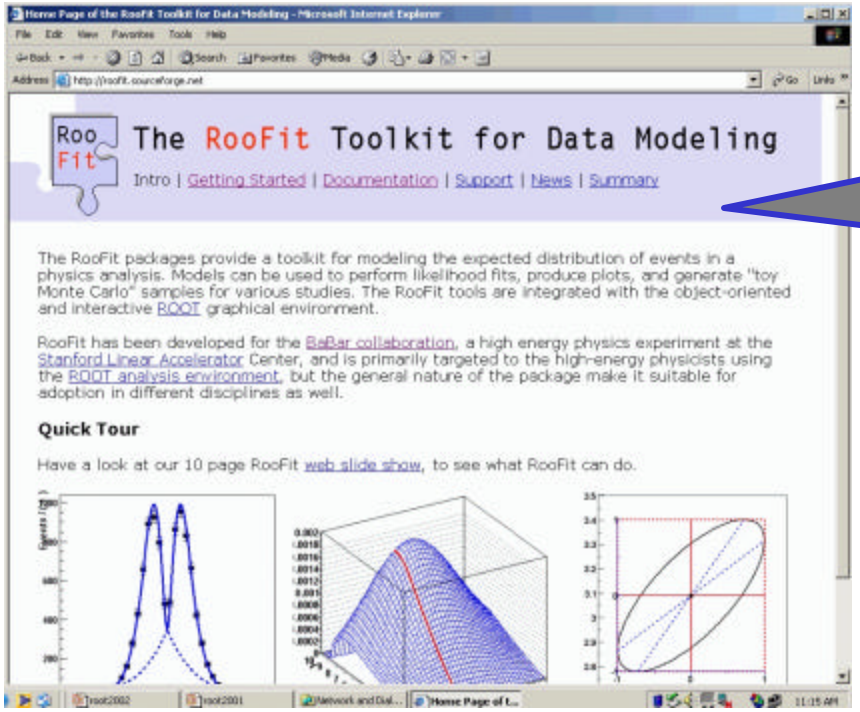
Advanced features – Task automation

- Support for routine task automation, e.g. goodness-of-fit study

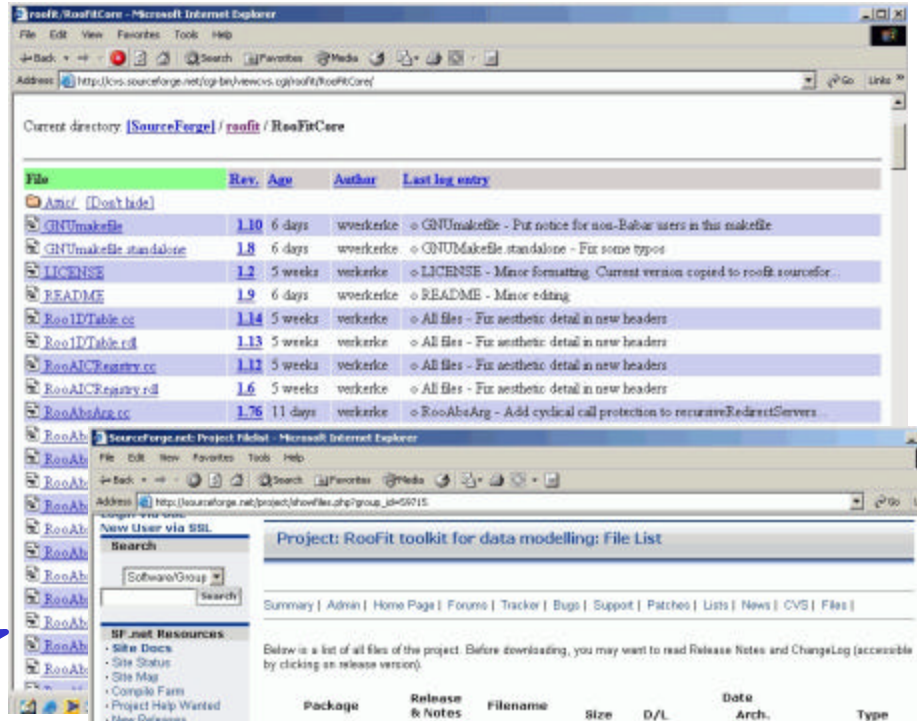



```
// Instantiate MC study manager
RooMCStudy mgr(inputModel) ;

// Generate and fit 100 samples of 1000 events
mgr.generateAndFit(100,1000) ;

// Plot distribution of sigma parameter
mgr.plotParam(sigma)->Draw()
```

RooFit at SourceForge - **roofit.sourceforge.net**



RooFit available at SourceForge
to facilitate access and
communication
with all users



Code access

- *–CVS repository via pserver*
- *–File distribution sets for production versions*

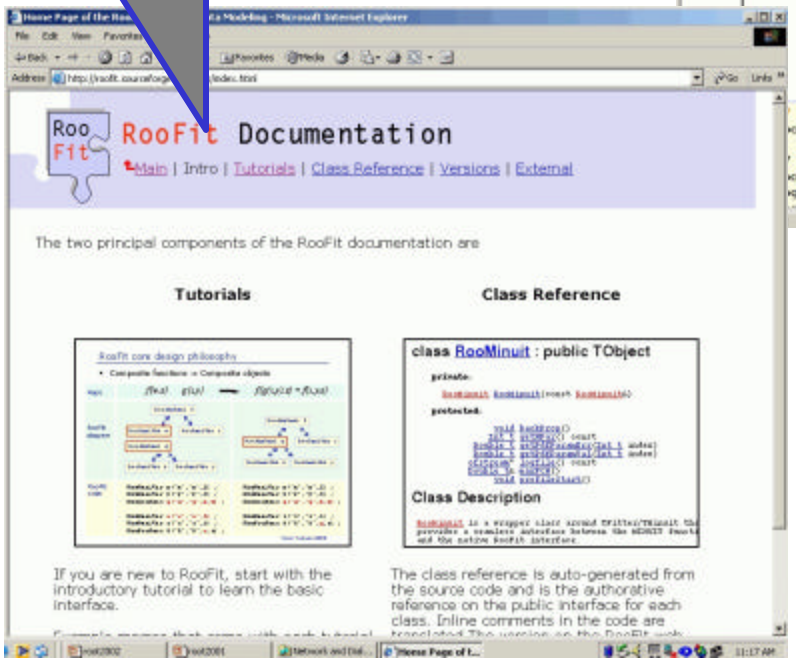
RooFit at SourceForge - Documentation

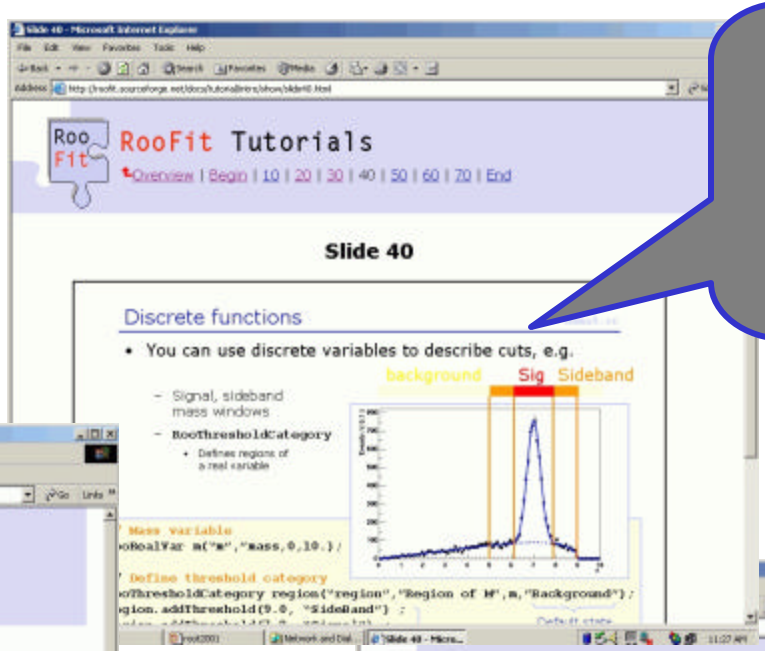
Documentation

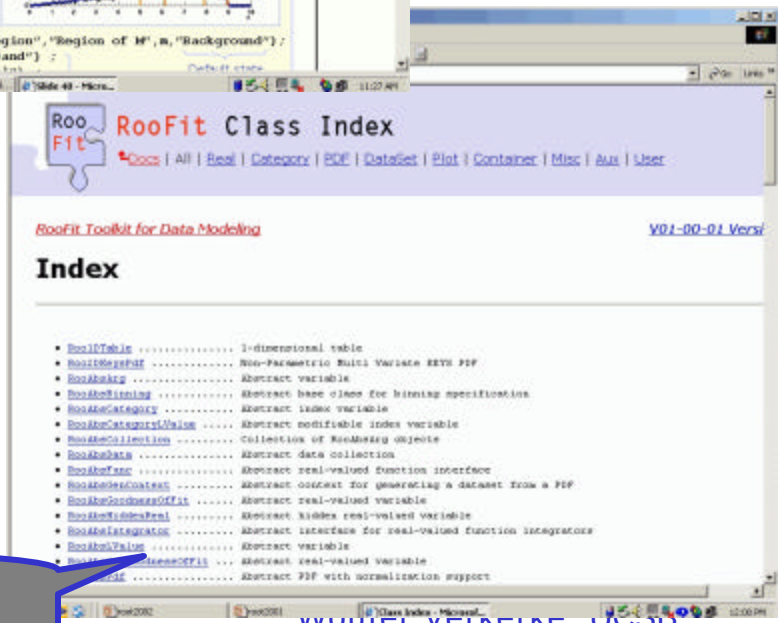
Comprehensive set of tutorials (PPT slide show + example macros)

Five separate tutorials

More than 250 slides and 20 macros in total







Class reference in `THtml` style

Significance & probability

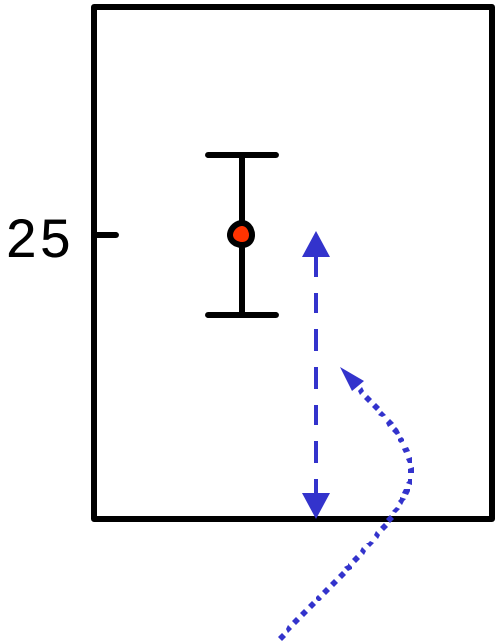
— CLT revisited – interpreting your error beyond 2s as Gaussian
— Null Hypothesis testing – P-values
— Classical or ‘frequentist’ confidence intervals
— Issues that arise in interpretation of fit result
— Bayesian statistics and intervals

Wouter Verkerke, UCSB

Significance and probability – introduction

- Suppose you have the final result from your analysis, e.g.

$$N_{sig} = 25 \pm 7$$



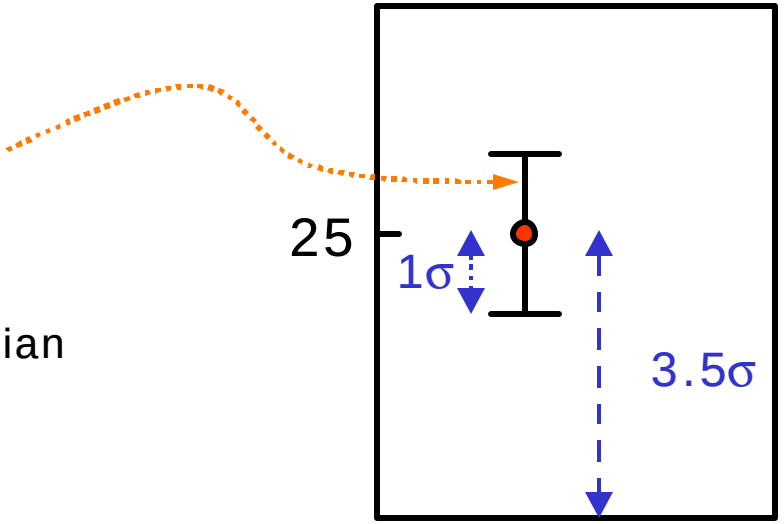
- What does this mean?
  - Is it sufficiently different from 0 to claim discovery of a new particle?
  - Or should you be more cautious?

- Need to state results in terms of absolute probabilities
  - For example, probability result is due to bkg fluctuation is <0.001%

Significance – Gaussian error assumption

- Naïve interpretation of $\mathbf{N_{sig} = 25 \pm 7}$ **:**

significance is $\frac{25-0}{7} = 3.5s$



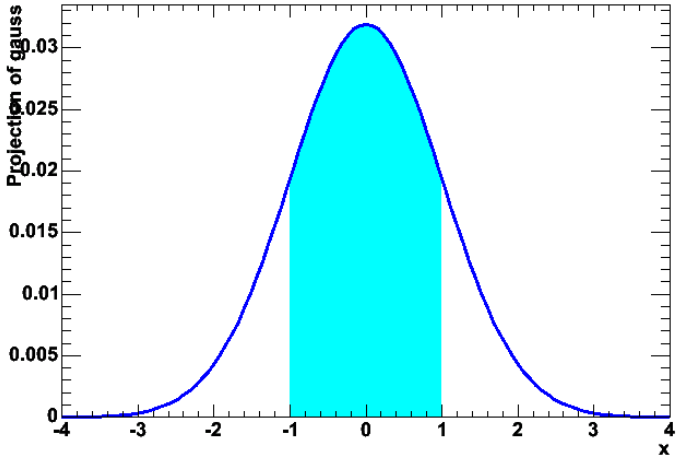
  - So probability that signal is fake corresponds to fraction of Gaussian beyond 3.5σ, which is < 0.1%
  - **Is this correct?**

- Assumption made: Your sampling distribution is Gaussian
  - In other words, if you would repeat the experiment many times the resulting distribution of results is perfectly Gaussian
  - Not necessarily bad assumption: Central Limit Theorem predicts converge to a Gaussian sampling distribution at high statistics, but convergence beyond 2-3s range can take relatively large N

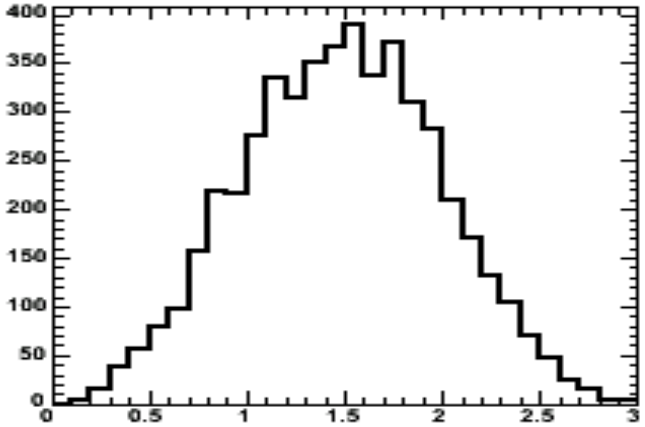
Significance – Gaussian sampling distribution

- Sampling distribution = Distribution you obtain if you repeat experiment many times

Assumed distribution with Gaussian error interpretation



Actual sampling distribution for hypothetical low N measurement



Gaussian integral fractions	Relative Discrepancy	Actual integral fractions
31.73% outside 1s	← 1.6% →	**32.23% within 1s**
4.57% outside 2σ	← 4.8% →	4.35% within 2σ
0.27% outside 3σ	← 33% →	0.18% within 3σ

Tails of sampling distribution converge more slowly to Gaussian

Significance – Example N=25±7 continued

- So be careful assigning Gaussian probabilities when looking at $>2\sigma$ deviations
  - Monte Carlo study of sampling distribution may be necessary

- But wait – **there is another issue!**
  - Just measured probability that true signal yield is zero, given a measurement of 25 ± 7 events
  - This is not the number you're most interested in to claim a discovery…

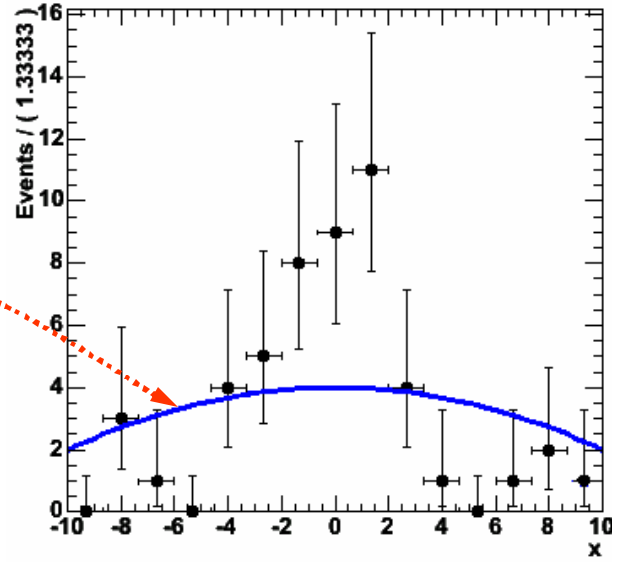
- What you really want know
  - What is the probability that my **background** will **fluctuate upwards** to 25 events and **fake** the **signal** we observe
  - Technical term: '**P-value**' – Probability that the null hypothesis (in this case 0 signal events) reproduces the observed signal

*These numbers are generally **not** the same*

Calculating P-values – Pearson's χ^2 test

- Idea: Calculate χ^2 of data with respect to null hypotheses

$$c^2 = \sum_{i=1}^{N} \frac{(n_i - f_i^{null})^2}{f_i^{null}}$$



- P-value given by

$$P(c^2; N) = \int_{c^2}^{\infty} p(c^{2\prime}; N) dc^{2\prime}$$

- Example: χ^2 = 29.8 for N=20 d.o.f → P(χ^2)=0.073 = P-value

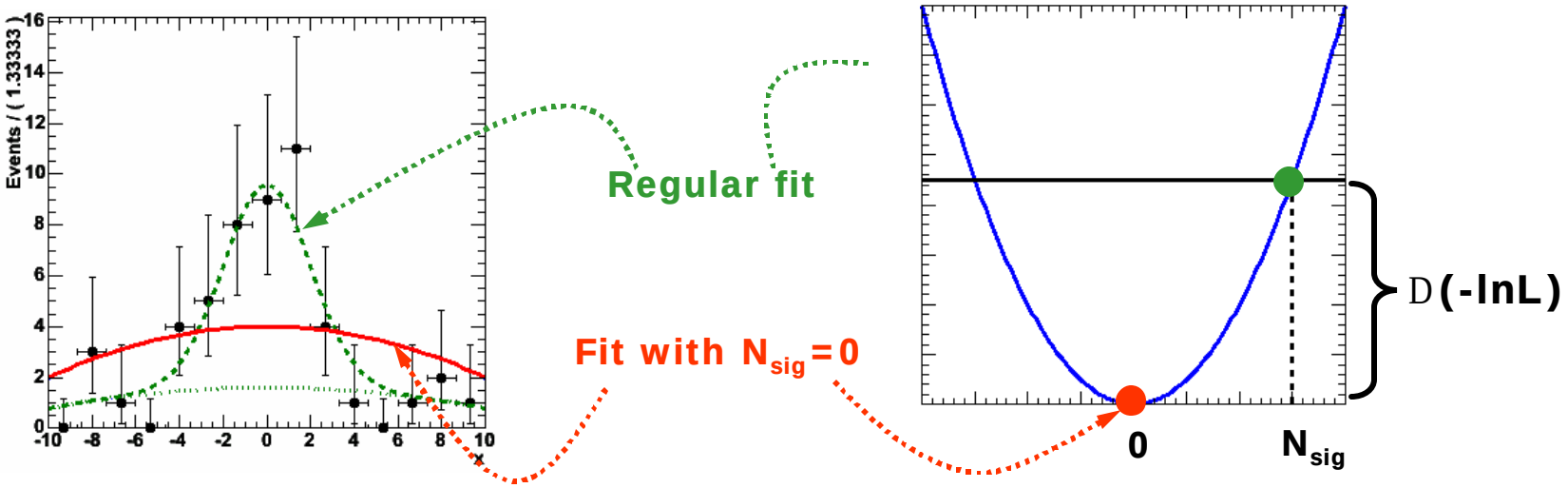
- Warning: P(χ^2) probability interpretation only valid for normal sampling distribution.
  - If statistics are low P(χ^2) distribution will distort → Use Monte Carlo study to calculate correct shape for your sample
  - Monte Carlo adjusted result for above example P(χ^2) = 0.11

Calculating P-values – Δln(L) method

- Significance from ML fit is similar to Pearson's χ^2 test
  1) Perform regular Maximum Likelihood fit to determine N_{sig}
  2) Repeat ML fit with N_{sig} parameter fixed
  – From difference in log(L) values in fits 1) and 2) calculate

$$\Delta(-\ln L) = \tfrac{1}{2} s^2$$ ← P-value from Gaussian σ interpretation

  – Significance interpretation assumes normal sampling distribution



Significance, Normal Sampling & Confidence intervals

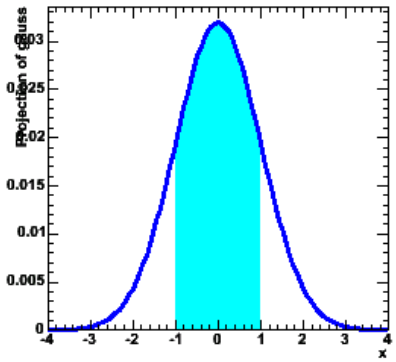
- Calculating the significance of a result by means of a P-value is straightforward for normal sampling distributions
  - If statistics become low, methods discussed are in inaccurate
  - But you can correct these method through Monte Carlo studies (e.g. computing the distorted χ^2 distribution for a low statistics sample rather than relying on the standard χ^2 distribution)
- You can avoid this altogether when you explicitly construct a confidence interval for your result
- Example of Confidence interval

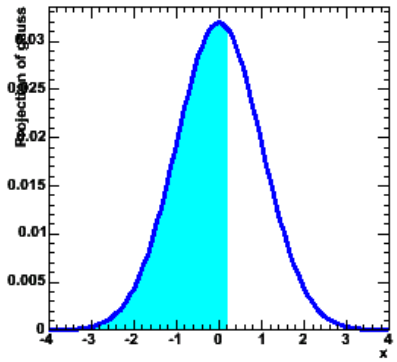
  $18 < N < 32$ at 68% Confidence Level (C.L.)

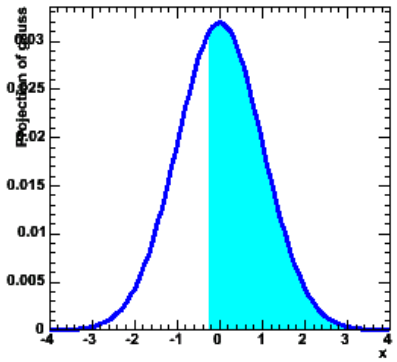
  - No Gaussian assumptions made in this statement
  - Confidence intervals often used for results where interpretation of uncertainties is non-trivial (i.e. non-Gaussian)

Confidence intervals – the basics

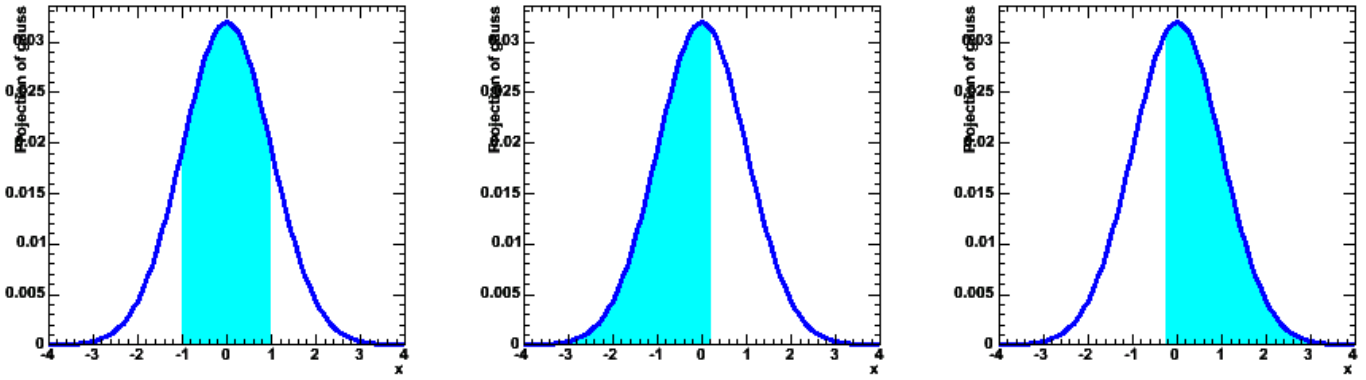
- Definition of a classic or 'frequentist' confidence interval at CL% confidence level
  - If you repeat a measurement X many times and calculate a confidence interval $[X_-,X_+]$ for each of them, CL% of the time the true value will be contained in the interval.
    - Note that a frequentist confidence interval *makes no statement about the true value of x*. For a given experiment and corresponding interval, the true value either is or isn't in the interval, no statement is made about that. It just says that if you repeat the experiment and interval calculation many times, CL% of the time the true value is inside the interval
- Note: this definition is ambiguous
  - Examples below are all 3 valid 68% C.L confidence intervals of a Gaussian sampling distribution







Confidence intervals – the basics



- Resolve ambiguity in definition by requiring either
  - 1) Symmetry: x-,x+ are equidistant from mean
  - 2) Shortest interval: choose x-,x+ such that $|x_- - x_+|$ is smallest
  - 3) Central $\int_{-\infty}^{x_-} P(x)dx = \int_{x_+}^{\infty} P(x)dx = \frac{1-C.L.}{2}$ ← **Most sensible**
  - For Gaussian sampling distributions all 3 requirements result in same confidence interval

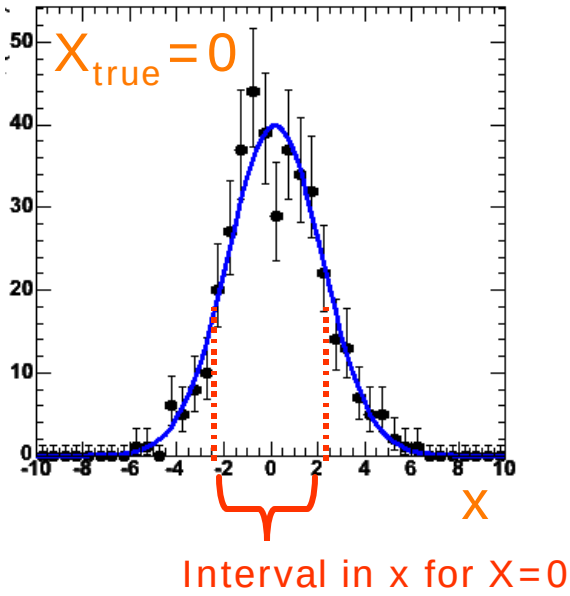
Confidence intervals – How do you make one?

- Example: Given a measurement x_{obs} of a true value X_{true}

- Step 1: For a given value of X_{true} find interval $[x_+, x_-]$ that contain 68% of the values of x_{obs}
  - Monte Carlo approach common:

1) Generate O(1000) data samples with true value X
2) For each sample, execute analysis and find measured value x
3) Find interval in x that contains 68% of values of x

[NB: This interval is in x_{obs}.
It is NOT the confidence interval, which will be in X_{true}]



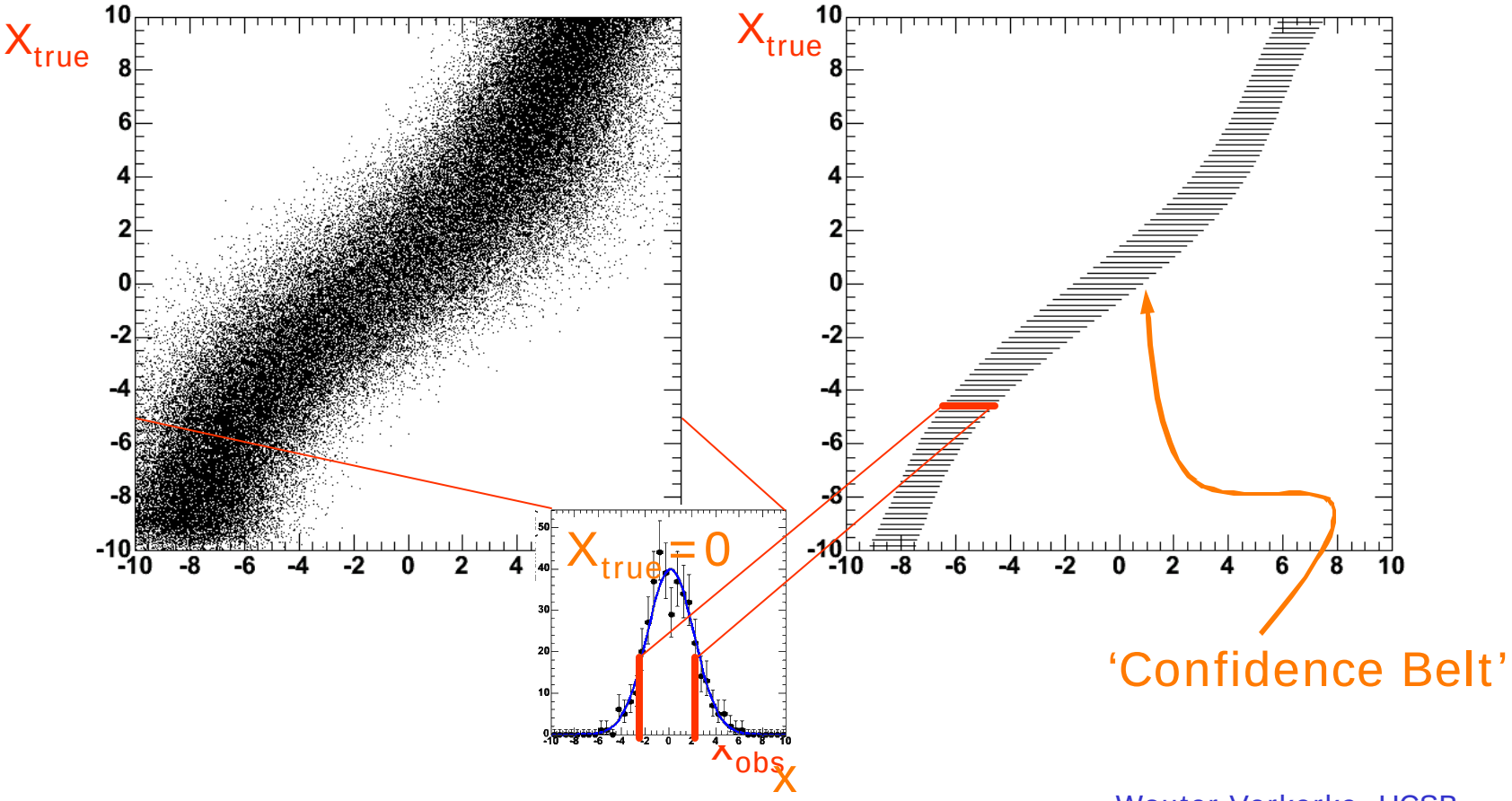
- Repeat procedure for a wide range of value for X_{true}

Frequentist confidence intervals – the Confidence Belt

- Result of step 1

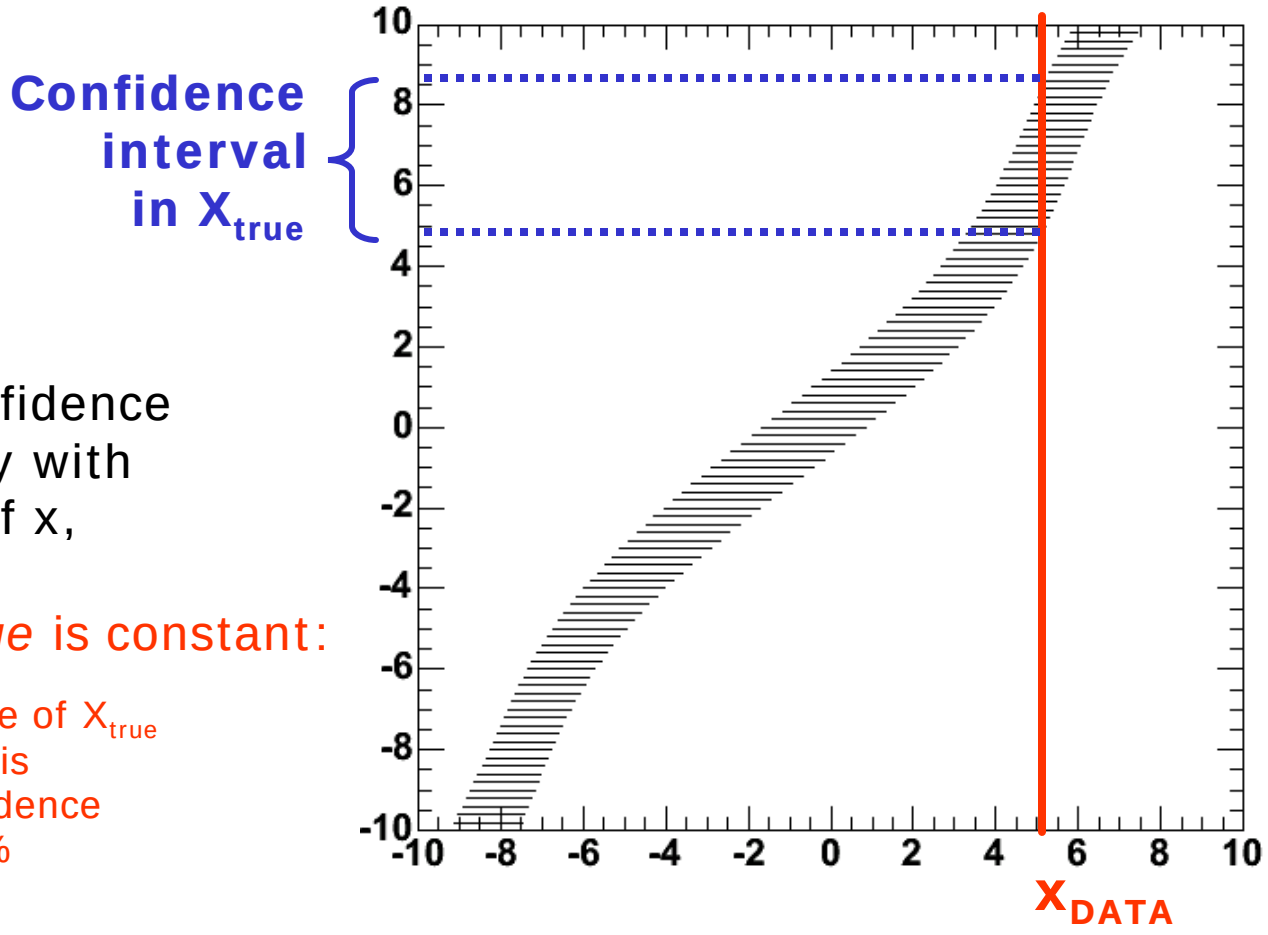
Each point measurement x_{obs} from a MC dataset generated with X_{true}

Intervals that contains 68% of values of x_{obs} for each X_{true}



Frequentist confidence intervals – the Confidence Belt

- Step 2 – Given data measurement of x_{obs} read off confidence interval in X_{true}



NB: Width of confidence interval may vary with observed value of x,

But 68% *coverage* is constant:

regardless of the value of X_{true} the probability that it is contained in the confidence interval is always 68%

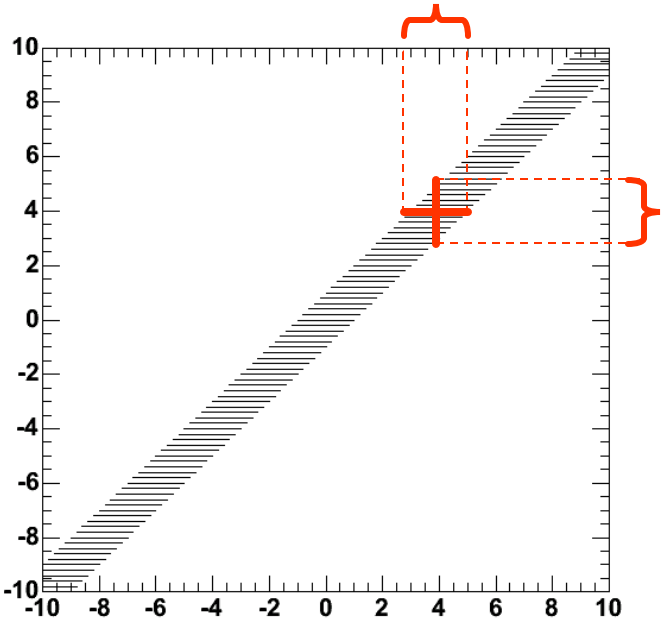
Important concept in frequentist intervals

Frequentist Confidence Interval – the Gaussian case

- Confidence intervals make no assumption of a Gaussian sampling distribution
  - but what do they look like if we have one?
  - Gaussian sampling distribution: $x_{\text{obs}}(X_{\text{true}}) = \exp\left[-\frac{1}{2}\left((x_{\text{obs}} - X_{\text{true}})/s\right)^2\right]$

- Result of step 1 with Gaussian sampling distribution

Width of confidence belt in x_{obs} = 2s



Width of confidence belt in X_{true} = 2s

$$68\%\ \text{C.I.} = \left[X - s\ ,\ X + s\right]$$

Gaussian case gives familiar result

Frequentist Confidence Interval – Eternal sunshine?

- Frequentist confidence intervals are properly defined in case of non-Gaussian sampling distributions
  - Valid intervals are obtained for e.g. low statistics fits
  - In case of a Gaussian sampling distribution the familiar Gaussian errors are obtained
  - But does it always tell you want you want to know?

- Two (arbitrary) examples at low statistics
  - A) we measure $N_{sig} = 20 \pm 10$ → [0,+40] 68% C.L.
  - B) we measure $N_{sig} = -20 \pm 10$ → **[-40, 0] 68% C.L.**

- In case A) we are happy, no questions asked…
- In case B) we are not: We 'know' that N_{sig} must be >0!
  - Nevertheless the interval is well defined! If you repeat the experiment many times 68% of the reported confidence intervals will contain the true value

Experimental summary versus Interpretation

- Key problem: Interval is statistically well defined,
  but physical interpretation makes no sense

$-40 < N_{sig} < 0$ at 68% C.L. ? N_{sig} my is number of Higgs decays so it must be ≥ 0.

- Solution depends on what you want!
  1) Summary of experimental result, or
  2) Incorporate physical interpretation/constraints in your result
  - These are two different things,
    and cannot really be accomplished simultaneously

- Frequentist Confidence Interval accomplishes 1), how do you do 2)?

Bayesian statistics – Decision making

- Bayesian statistics interprets probabilities very differently from Frequentist statistics
  - It provides a natural framework to include prior beliefs (such as $N_{sig}>0$)
- Essential Bayesian formulas:

Notation of conditional probabilities:
p(A|B) is probability of A given B

Bayes Theorem:

$$p(A|B) = \frac{p(B|A)}{p(B)}\,p(A)$$

Say 'A' = 'theory'
'B' = 'exp. result'

Law of Total Probability

$$p(res) = \sum_i P(res|the_i)P(the_i)$$

$$p(the|res) = \frac{p(res|the)\,p(the)}{\sum_i p(res|the_i)\,p(the_i)}$$

Bayesian statistics – Decision making

- How to read this formula

P(res|the) = Your measurement the probability of an experimental under a given theoretical hypothesis

P(the) = Prior Belief (e.g $N_{sig}>0$)

$$p(the \mid res) = \frac{p(res \mid the)\, p(the)}{\sum_i p(res \mid the_i)\, p(the_i)}$$

Normalization term

P(the|res) = Your new belief in the theory, given the just obtained experimental result 'interpretation'

Bayesian statistics – Medical example

- Medical example: P(disease) = 0.001
  - Prior belief (your input theory)
  - E.g. based on population average

- Consider test for disease, result is either + or –
  - **P(+|+) = 0.98** – Prob that test will correctly ID disease
  - **P(-|+) = 0.02** – Prob that test will give false negative
  - **P(+|-) = 0.03** – Prob that test will give false positive
  - **P(-|-) = 0.97** – Prob that test will correctly ID absence of disease

- Suppose you test positive – should you be worried?

$$P(+|+) = \frac{P(+|+)P(disease)}{P(+|+)P(+) \quad + \quad P(+|-)P(-)} = \frac{0.90 \cdot 0.001}{0.98 \cdot 0.001 + 0.03 \cdot 0.999} = 0.032$$

  - **Posterior belief is 0.032, larger than initial belief but still not very large!**

Bayesian statistics – Medical example

$$P(+|+) = \frac{P(+|+)P(disease)}{P(+|+)P(+) \quad + \quad P(+|-)P(-)} = \frac{0.90 \cdot 0.001}{0.98 \cdot 0.001 + 0.03 \cdot 0.999} = 0.032$$

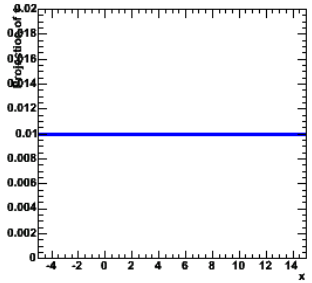
- Medical example deals with simple hypothesis (true or false)
- In physics we often deal with composite hypothesis
  - I.e. our hypothesis has parameters
  - We will use Probability Density Functions as function of vector of parameters **a** rather than with total probabilities, i.e.

$$p(the) \quad \rightarrow \quad p(the;\vec{a})$$
$$p(res\,|\,the) \quad \rightarrow \quad p(res\,|\,the;\vec{a})$$

Physics Example – Measurement of parameter $Q = Q_0 \pm \sigma(Q)$

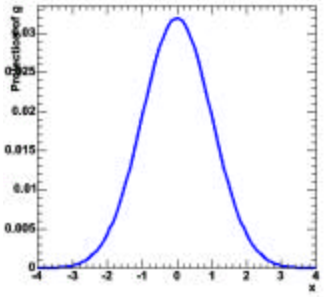
$$\frac{p(the;Q)\cdot p(res\,|\,the;Q)}{\sum_i p(res\,|\,the_i;Q)\,p(the_i;Q)} = p(the\,|\,res;Q)$$

Initial belief on Q 'prior' : we know nothing



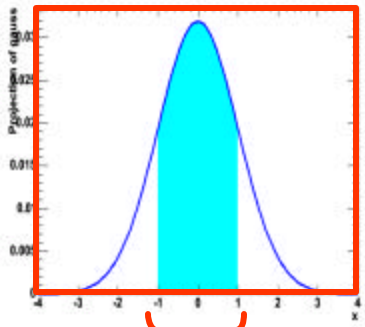
X

Measurement of $Q = Q_0 \pm \sigma(Q)$: Gaussian PDF with mean of Q_0 and width of $\sigma(Q)$



=

Posterior belief on Q is product of prior belief and measurement



Bayesian 68% interval = Area that integrates 68% of posterior Bayesian distribution

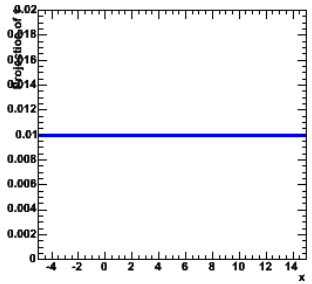
(Resolve ambiguity in definition in the same way as for a frequentist confidence interval)

NB: In this Gaussian example Bayesian interval is same as Frequentist interval

Bayesian Physics Example – Incorporating any measurements

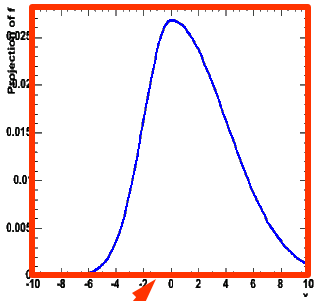
$$\frac{p(the;Q)\cdot p(res\mid the;Q)}{\sum_i p(res\mid the_i;Q)\,p(the_i;Q)} = p(the\mid res;Q)$$

Initial belief on Q 'prior' : we know nothing



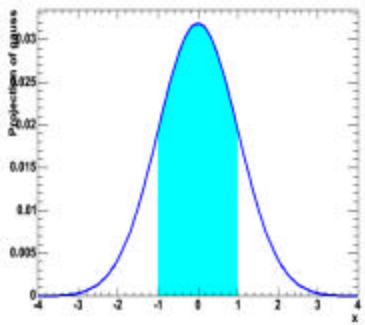
X

Measurement of Q from ML fit



=

Posterior belief on Q is product of prior belief and measurement

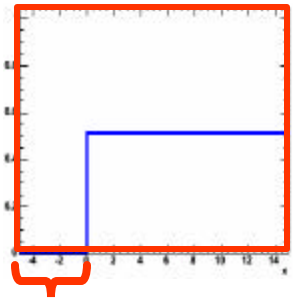


Very practical aspect of Bayesian analysis:
Measurement of Q = Likelihood distribution from fit!

Including prior knowledge – Using a non-trivial prior

$$\frac{p(the;Q)\cdot p(res\mid the;Q)}{\sum_i p(res\mid the_i;Q)\,p(the_i;Q)} = p(the\mid res;Q)$$

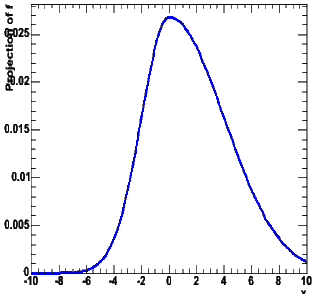
New initial belief on Q
it must be >0



Values below
0 now a priori
forbidden

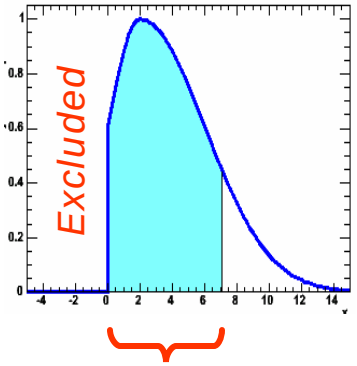
X

Measurement of Q



=

Posterior belief on Q
is product of prior belief
and measurement



Bayesian interval
changed to take
initial belief into account

Baysian interval with this prior will be different from Frequent interval

Bayesian statistics – a word of caution

- Bayesian framework provides easy mechanism to incorporate 'prior' knowledge via ***p(the;a)***
  - Difficulties arise when we want to express 'no prior knowledge', i.e.

- Is a flat prior really equivalent to complete ignorance?
  - Apparent problem: if we declare all values of Q to be equally probable, then all values of Q^2 are not equally probable!
  - Example: Complete ignorance in Q translates into prior preference for low values of Q^2
  - Posterior Bayes distribution, interval will depend on choice of parameterization…

- Be careful with concept of 'prior ignorance'
  - If you go for prior ignorance, try a few choices of parameterization
  - If it matters, be warned!

One-sided versus two-sided intervals

- By default confidence intervals are two-sided intervals e.g.

  $18 < X < 30$ (68% C.L.)

- In preceding Bayesian example we have explicitly excluded the range below zero through a prior

  $0 < X < 5.3$ (68% C.L.)

  which is then usually quoted as a one-sided confidence interval

  $X < 5.3$ (68% C.L.)

  – One sided intervals are customarily quoted when you see no signal.

Special issue – Frequentist confidence intervals with constraints

- There exist recent methods to construct proper frequentist confidence intervals in the presence of boundaries
  - Example: 'A Unified Approach to the Classical Statistical Analysis of Small Signals', Gary J Feldman and Robert D Cousins [PRD 57, 3873 (1998)]
  - Treatment of Feldman & Cousins beyond scope of this lecture

- Main feature of Feldman & Cousins: it decides for you

  when to quote a 1-sided interval [X<N] at X% C.L and
  when to quote a 2-sided interval [X<N<Y] at X% C.L.

  - Preserves main characteristic of frequentist interval: coverage is independent of true value of N
  - If you would decide by yourself this would be the case probably.

- Does this help you? Sometimes
  - Intrinsic problem with 1-sided intervals remains: they are difficult to average a posteriori. E.g. given two results A,B
    - $N_A = 15 \pm 13$, $N_B = 10 \pm 7$ → $N_{AB} = 13 \pm 6$
    - $N_A < 28$, $N_B < 17$ → NAB=???

Frequent vs Bayesian – Summary of options

- NB: This is often a hotly debated topic among physicists!
- Frequentist confidence intervals
  - Provide ‘summary of information content’ of measurement
  - No interpretation of result is made → Intervals may include values deemed unphysical (though Feldman & Cousins can help here)
- Bayesian intervals
  - Support physical interpretation of result.
  - Provides easy framework for incorporating physical constraints etc (these are all ‘prior’ beliefs)
  - But you can run into difficulties incorporating prior ignorance
- For normal (Gaussian) sampling distributions Bayesian interval with uniform prior and Frequentist intervals are identical
  - In that case both are also identical to interval defined by $\Delta(-\ln L)=0.5$

Systematic errors

— Sources of systematic errors
— Sanity checks versus systematic error studies
— Common issues in systematic evaluations
— Correlations between systematic uncertainties
— Combining statistical and systematic error

Wouter Verkerke, UCSB

Systematic errors vs statistical errors

- Definitions

Statistical error = any error in measurement due to statistical fluctuations in data

Systematic errors = **all other errors**

Systematic uncertainty ≡ Systematic error

- But Systematic **error** ≠ Systematic **mistake**!
  - Suppose we know our measurement needs to be corrected by a factor of 1.05 ± 0.03
  - Not correcting the data by factor 1.05 introduces a systematic mistake
  - Right thing to do: correct data by factor 1.05 and take uncertainty on factor (0.03) as a systematic error

Source of systematic errors – 'Good' and 'Bad' errors

- 'Good' errors arise from clear causes and can be evaluated
  - Clear cause of error
  - Clear procedure to identify and quantify error
  - Example: Calibration constants,
    efficiency corrections from simulation

- 'Bad' errors arise from clear causes, but can *not* be evaluated
  - Still clear cause
  - But no unambiguous procedure to quantify uncertainty
  - Example: theory error:
    - Given 2 or more choices of theory model you get 2 or more different answers.
    - What is the error?

Sources of systematic errors – 'Ugly' errors

- 'Ugly' errors arise from sources that have been overlooked
  - Cause unknown → error unquantifiable

- 'Ugly' errors are usually found through failed sanity checks
  - Example: measurement of CP violation on a sample of events that is known to have no CP-violation: You find $A_{CP} = 0.10 \pm 0.01$
  - Clearly something is wrong – What to do?

  - 1) **Check your analysis**
  - 2) Check your analysis again
  - 3) Phone a friend
  - 4) Ask the audience

    ...
  - 99) **Incorporate as systematic error as last and desperate resort!**

What about successful sanity checks?

- Do not incorporate successful checks in your systematic uncertainty
  - Infinite number of successful sanity checks would otherwise lead to infinitely large systematic uncertainty. Clearly not right!

- Define beforehand if a procedure is a sanity check or an evaluation of an uncertainty
  - If outcome of procedure can legitimately be different from zero, it is a systematic uncertainty evaluation
  - If outcome of procedure can only significantly different from zero due to mistake or unknown cause, it is a sanity check

Common scenarios in evaluating systematic errors

- Two values – corresponding to use of two (theory) models A,B
  - What is a good estimate for your systematic uncertainty?

- I) If A and B are *extreme scenarios*, and the truth must always be between A and B
  - Example: fully transverse and fully longitudinal polarization
  - Error is root of variance with uniform distribution with width A-B

$$s = \frac{|A-B|}{\sqrt{12}} \qquad V(x) = \langle x \rangle^2 - \langle x^2 \rangle = \left(\frac{1}{2}\right)^2 - \int_0^1 x^2 dx = \frac{1}{4} - \frac{1}{3} = \frac{1}{12}$$

  - Popular method because sqrt(12) is quite small, but only justified if A,B are truly extremes!

- II) If A and B are typical scenarios
  - Example: JETSET versus HERWIG (different Physics simulation packages)
  - Error is difference divided by sqrt(2)

$$s = \frac{|A-B|}{2} \cdot \sqrt{2} = \frac{|A-B|}{\sqrt{2}}$$

Factor $\sqrt{\frac{N}{N-1}}$ *to get unbiased estimate of* s_{parent}

Common scenarios in evaluating systematic errors

- Two variations of the analysis procedure on the *same* data
  - Example: fit with two different binnings giving A ± σ_A and B ± σ_B
  - Clearly, results A,B are correlated so $\frac{|A-B|}{\sqrt{s_A^2+s_B^2}}$ is not a good measure of smallness of error

- Generally difficult to calculate, but can estimate uppper,lower bound on systematic uncertainty

$$\sqrt{s_A^2-s_0^2}-\sqrt{s_B^2-s_0^2}\le s_{A-B}\le\sqrt{s_A^2-s_0^2}+\sqrt{s_B^2-s_0^2}$$

  - Where $\sigma_A>\sigma_B$ and σ_0 is the Minimum Variance Bound. $s_0^2(\hat{a})=\left\langle(\hat{a}-\langle\hat{a}\rangle)^2\right\rangle$
  - If the better technique (B) saturates the MVB the range reduces to

$$s_{A-B}^2=s_A^2-s_B^2$$

  - If MVB is not saturated (e.g. you have low statistics) you will need to use a toy Monte Carlo technique to evaluate $\sigma_{A\text{-}B}$

Common scenarios in evaluating systematic errors

- Perhaps most common scenario in HEP analysis: you need to assign systematic uncertainty to (in)accuracy of full Monte Carlo simulation
- Popular technique: 'Cut variation'
  - Procedure: vary each of your cuts by a little bit. For each change,

1) Measure new yield on data
2) Correct with new MC efficiency.
3) Difference between efficiency corrected results is systematic uncertainty.

  - Example, for a nominal cut in x at 'p' you find N(data)=105, with a MC efficiency $\varepsilon_{MC}=0.835$ so that N(corrected)=125.8

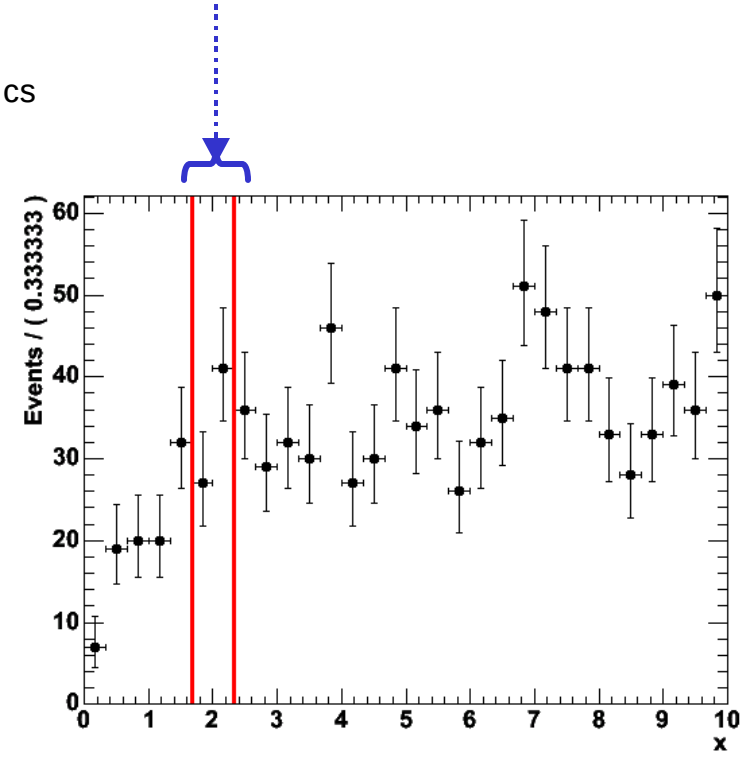
	N(data)	ε(MC)	N(corrected)
p+Δp	110	0.865	127.2
p-Δp	100	0.803	124.5

$$s^{p}_{sys} = (127.2 - 124.5)/2 = 1.4$$

$$x = 125.8 \pm 1.4$$

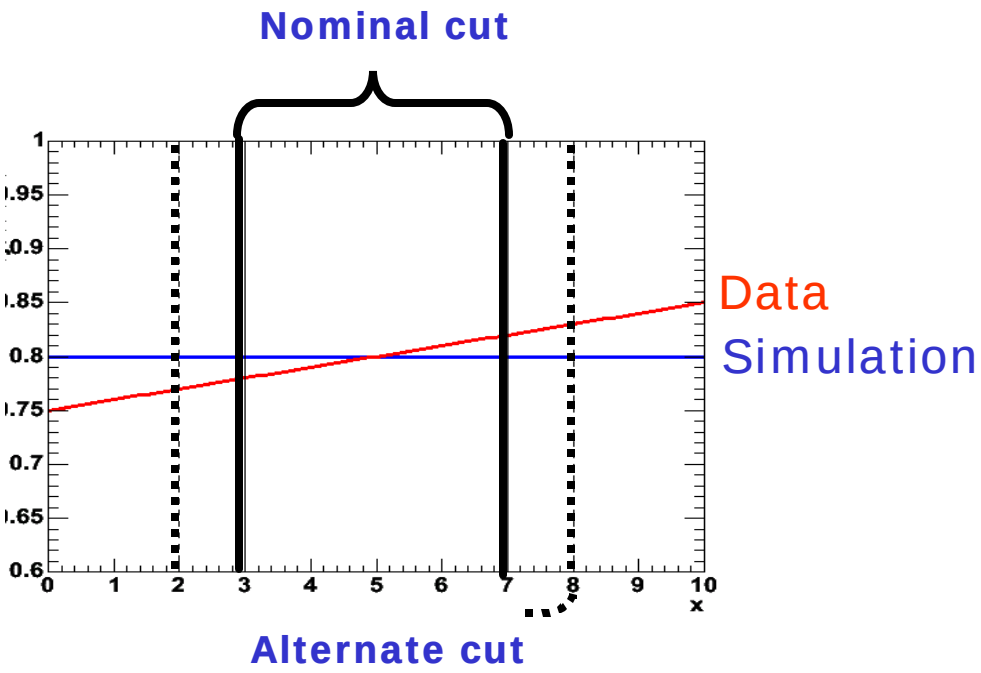
Common scenarios in evaluating systematic errors

- Warning I: Cut variation does not give an precise measure of the systematic uncertainty due data/MC disagreement!
  - Your systematic error is dominated by a potentially large statistical error from the small number of events in data between your two cut alternatives
    - This holds independent of your MC statistics
  - You could see a large statistical fluctuation
    → **error overestimated**
  - You could see no change due to a statistical fluctuation
    → **error underestimated**



Common scenarios in evaluating systematic errors

- Warning II: Cut variation doesn't catch all types of data/MC discrepancies that may affect your analysis
  - Error may be fundamentally underestimated
  - Example of discrepancy missed by cut variation:



Data and Simulation give same efficiency for nominal and alternate cut, sp

Zero systematic is evaluated (in limit N→8)

Even though data and MC are clearly different

Cut variation is a good sanity check,
but not necessarily a good estimator for systematic uncertainty

Systematic errors and correlations

- Pay attention to correlation between systematic errors

$$s_{xy}^2 = \left(\frac{df}{dx}\right)^2 s_x^2 + \left(\frac{df}{dy}\right)^2 s_y^2 + 2\left(\frac{df}{dx}\right)\left(\frac{df}{dy}\right) r\, s_x s_y$$

- If error uncorrelated, $\rho = 0$
  - Add in quadrature

- If error 100% correlated, then $\rho = 1$.
  - E.g. tracking efficiency uncertainty per track for 6 tracks,

$$\sigma_{3trk} = \sigma_{trk} + \sigma_{trk} + \sigma_{trk} = 3 \cdot \sigma_{trk} \quad (\textit{not } \sqrt{3} \cdot \sigma_{trk})$$

- If errors 100% anti-correlated, then $\rho = -1$
  - This can really happen!
  - Example BF($D^{*0} \rightarrow D^0\pi^0$) =67% and BF($D^{*0} \rightarrow D^0\gamma$) =33%

Combining statistical and systematic uncertainty

- Systematic error and statistical error are independent
  - They can be added in quadrature to obtain combined error
  - Nevertheless always quote (also) separately!
  - Also valid procedure if systematic error is not Gaussian: Variances can be added regardless of their shape
  - Combined error usually approximately Gaussian anyway (C.L.T)

- Combining errors a posteriori not only option
  - You can include any systematic error directly in your χ^2 or ML fit:

In χ^2 fit **In ML fit**

$$c^2 = c^2_{nom} + \left(\frac{p - p_0}{s_p} \right)^2 \quad ; \quad -\ln L = -\left[\ln L_{nom} + \tfrac{1}{2} \left(\frac{p - p_0}{s_p} \right)^2 \right]$$

  - Or, for multiple uncertainties with correlations

$$c_{pen} = \vec{p}^{\,T} V^{-1} \vec{p} \quad ; \quad -\ln L_{pen} = -\tfrac{1}{2} (\vec{p}^{\,T} V^{-1} \vec{p})$$

The End

- This course will be available at

  http://www.slac.stanford.edu/~verkerke/bnd2004/data_analysis.pdf

- Some material for further reading
  - R. Barlow, *Statistics: A Guide to the Use of Statistical Methods in the Physical Sciences*, Wiley, 1989
  - L. Lyons, *Statistics for Nuclear and Particle Physics*, Cambridge University Press,
  - G. Cowan, *Statistical Data Analysis*, Clarendon, Oxford, 1998
    (See also his 10 hour post-graduate web course:
    http://www.pp.rhul.ac.uk/~cowan/stat_course)