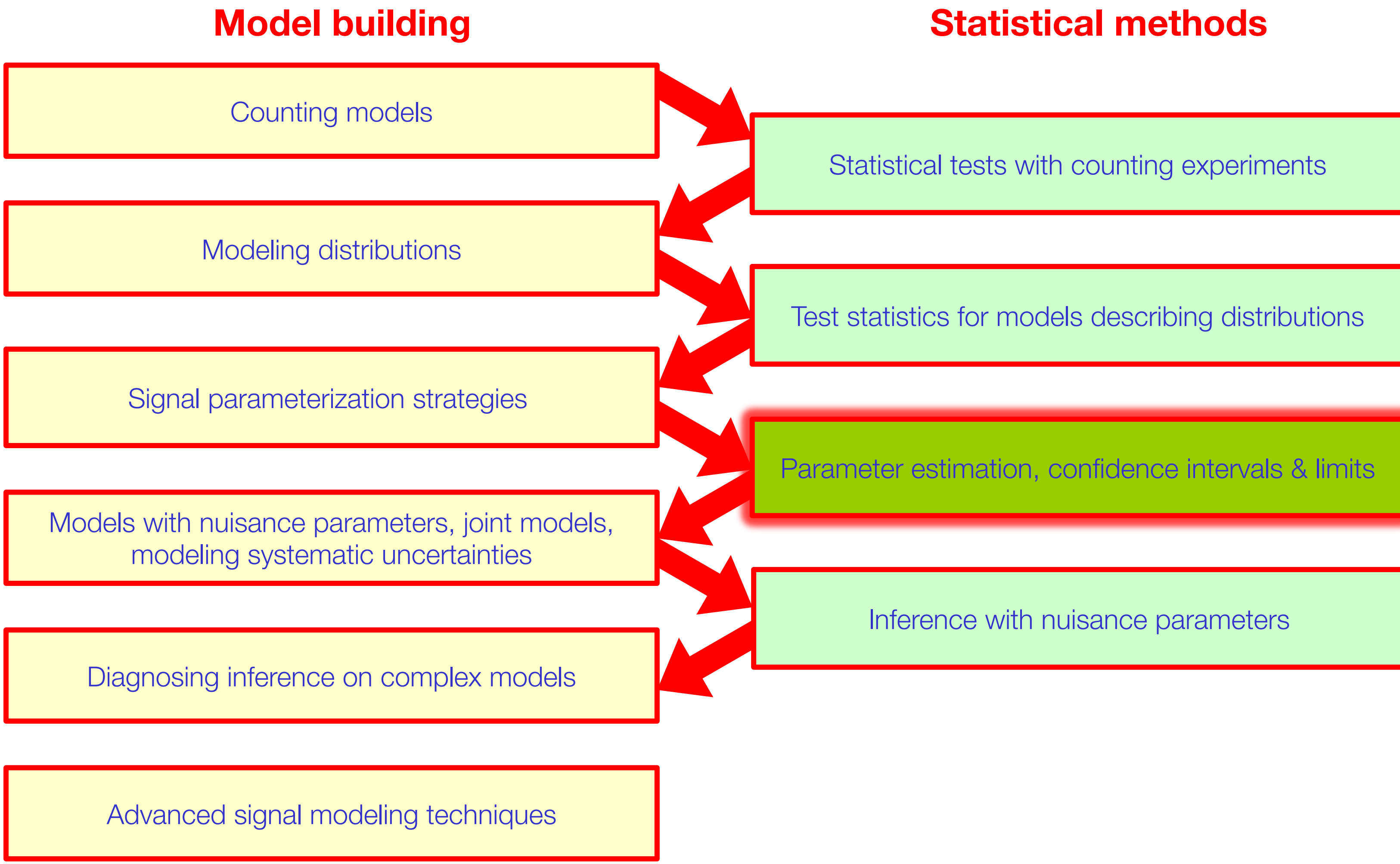
Statistical methods 3b (continued)

Expected results, upper limits and asymptotic formulae

Wouter Verkerke, NIKHEF

Roadmap of this course

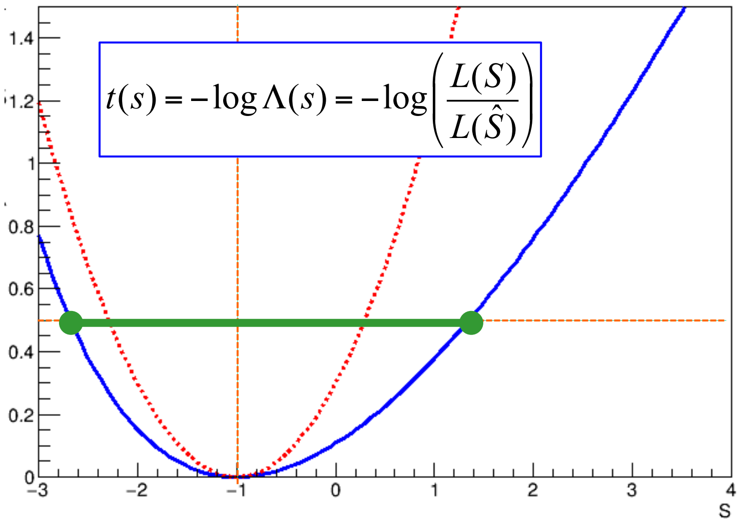
- Start with basics, gradually build up to complexity



Physics or statistics?

- An important and recurring dilemma facing analyzers is what to do with inference results of a statistical model that cover unphysical regions in the parameter space of the underlying theory
- Simplest example: Poisson counting experiment P(N|S+B)
  - Expect 5 background events, and 3 signal event
  - We observe 4 events – What result will we report? What conclusion will we draw?
- The data tells us precisely this : Likelihood L(s)=Poisson (4|S+5)
- Estimation procedures report:

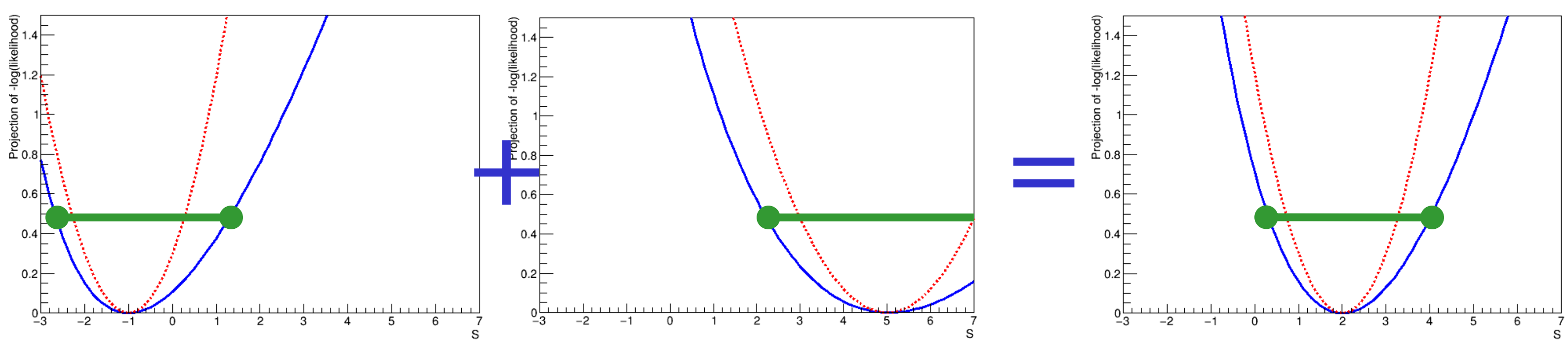
  ML parameter estimate → S= -1
  ML variance estimate → √V(S) = 1.83
  MINOS Conf. Interval → [-1.68,2.34] 68% C.L.



- Only S>0 is physical, what do we report?
  - Option A) Report as is?
  - Option B) Try to exclude unphysical regions from result

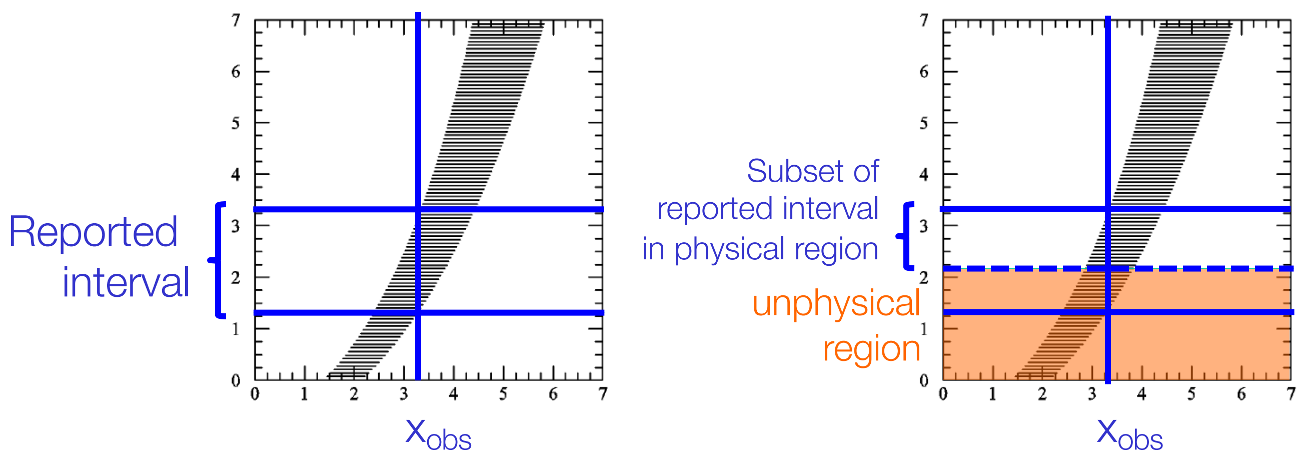
Physics or statistics?

- Q: Only S>0 is physical, what do we report?
  - Option A) Report as is?
  - Option B) Try to exclude unphysical regions from result?
- A: Depends on your goal!
- Goal 1: reporting, as accurately as possible, result of experiment
  - Observed result is not peculiar:
    44% of experiments of hypothesis S=0 with B=5 result in Nobs<5
    10% of experiments of hypothesis S=3 with B=5 result in Nobs<5
  - Problem arises only in interpretation of N in terms of S+B → defer interpretation
  - Report S, V(S), or confidence on S as usual (as proxy for the full likelihood)
  - Downside: interpretation deferred
  - Upside: easy to combine results of multiple experiments reported in this form (combination = inference on product of likelihoods



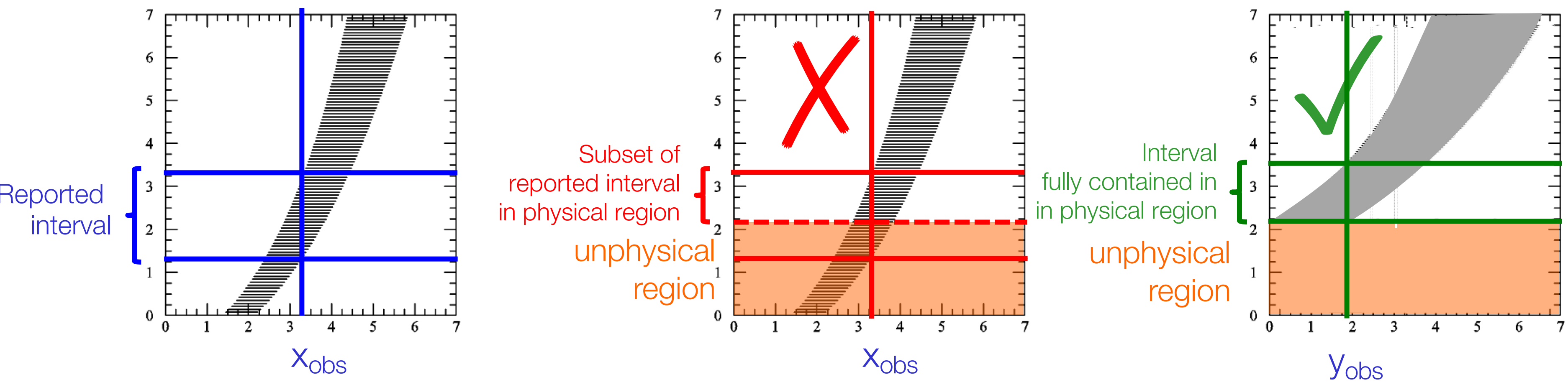
Physics or statistics?

- Q: Only S>0 is physical, what do we report?
  - Option A) Report as is?
  - Option B) Try to exclude unphysical regions from result?
- A: Depends on your goal!
- Goal 2: make physics interpretation of your model
  - Confidence interval should in that case not cover unphysical values
  - But you cannot simply exclude unphysical region without spoiling coverage properties (=calibration of 68%/95% promise)



Physics or statistics?

- Goal 2: make physics interpretation of your model
  - Confidence interval should in that case not cover unphysical values
  - But you cannot simply exclude unphysical region without spoiling coverage properties (=calibration of 68%/95% promise)
  - But you are allowed to modify the test statistic (=observed quantity) so that confidence belt never enters the unphysical region



- Can we modify test statistic such that boundaries are obeyed?

Physical boundaries frequentist confidence intervals

- Solution is to modify the statistic to avoid unphysical region

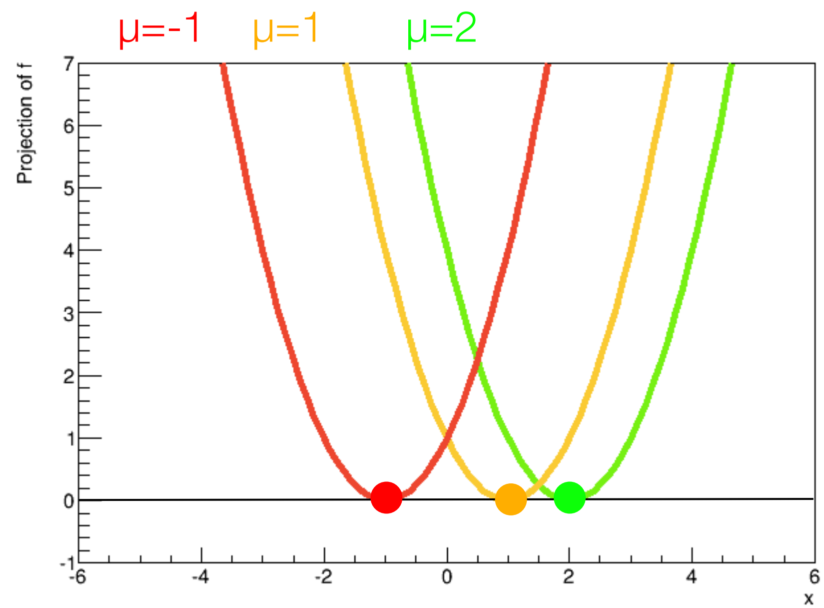
$$t_\mu(x) = -2\log\frac{L(x|\mu)}{L(x|\hat{\mu})}$$

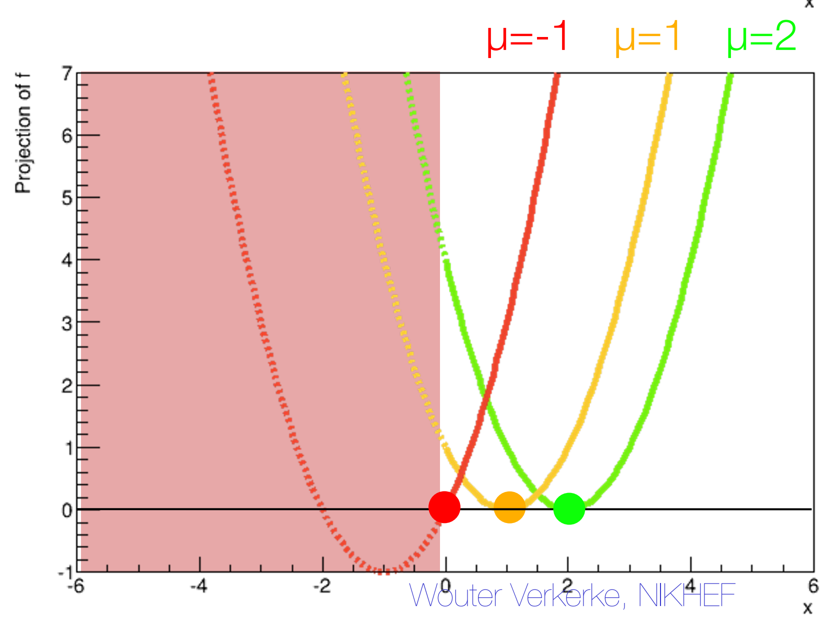
Introduce “physical bound” μ>0

$$\tilde{t}_\mu(x) = \begin{cases} -2\log\dfrac{L(x|\mu)}{L(x|\hat{\mu})} & \forall\hat{\mu}\geq 0 \\ -2\log\dfrac{L(x|\mu)}{L(x|0)} & \forall\hat{\mu}<0 \end{cases}$$

If μ<0, use 0 in denominator

→ *Declare data maximally compatible with hypothesis μ=0*





Physical boundaries in frequentist confidence intervals

- What is effect on *distribution* of test statistic?

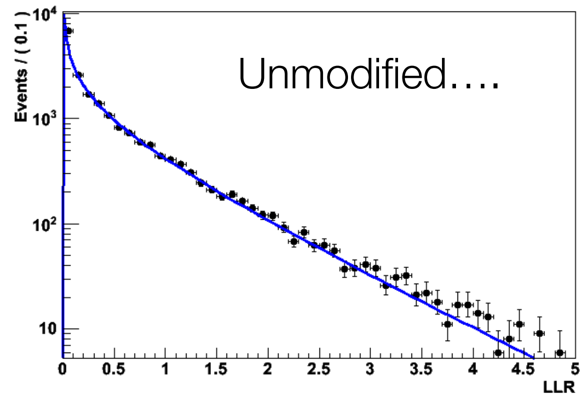
$$t_\mu(x) = -2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})}$$

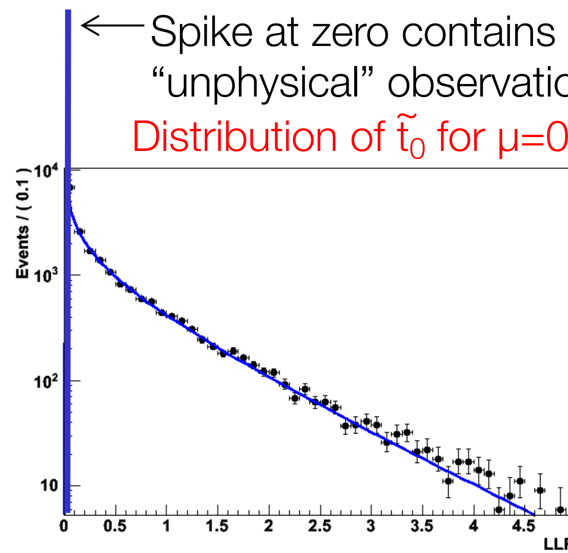
Introduce “physical bound” μ>0

$$\tilde{t}_\mu(x) = \begin{cases} -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall\hat{\mu}\geq 0 \\ -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,0)} & \forall\hat{\mu}< 0 \end{cases}$$

If μ<0, use 0 in denominator
→ *Declare data maximally compatible with hypothesis μ=0*

Distribution of $\tilde{t}_0$ for μ=2



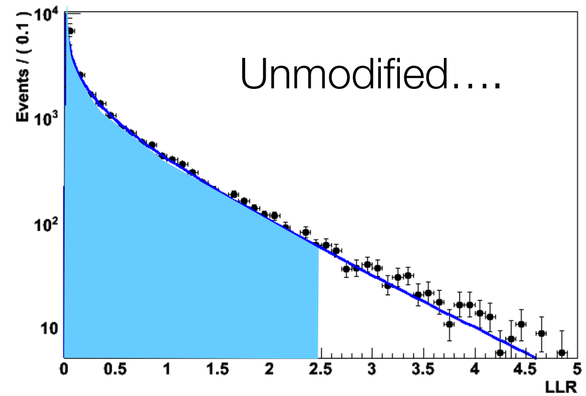


Distribution of $\tilde{t}_0$ for μ=0

Physical boundaries frequentist confidence intervals

- What is effect on *acceptance interval* of test statistic?

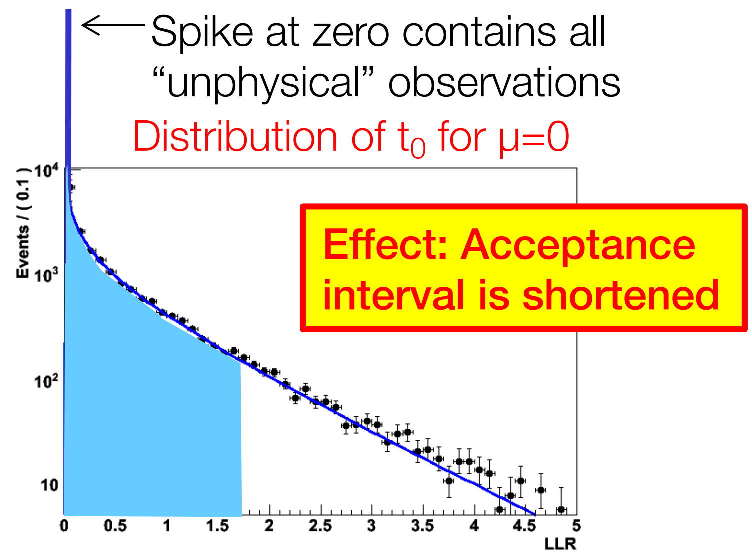
$$t_\mu(x) = -2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})}$$



Introduce
“physical bound”
μ>0

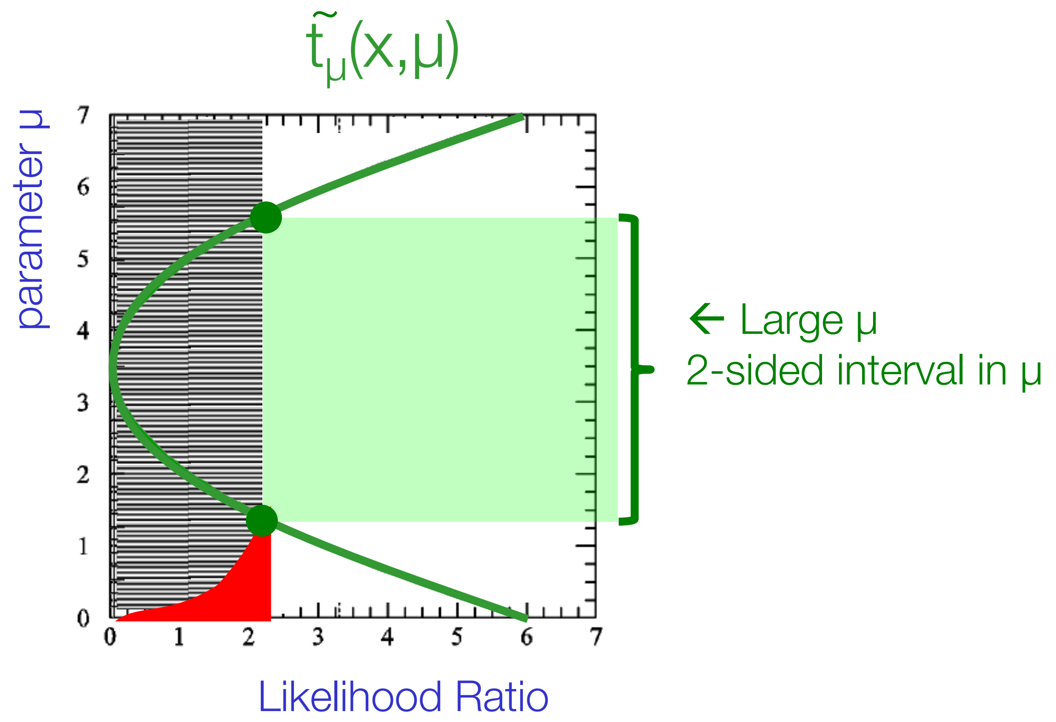
$$\tilde{t}_\mu(x) = \begin{cases} -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall\,\hat{\mu} \geq 0 \\ -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,0)} & \forall\,\hat{\mu} < 0 \end{cases}$$

If μ<0, use 0 in denominator
→ *Declare data maximally compatible with hypothesis μ=0*



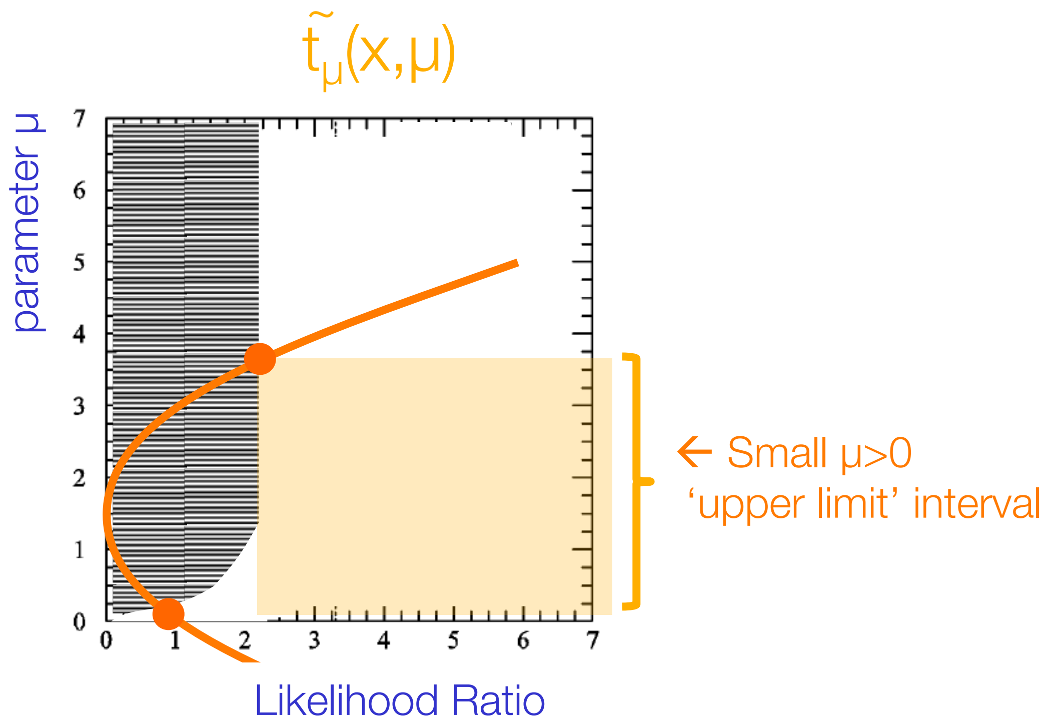
Physical boundaries frequentist confidence intervals

- Putting everything together – the confidence with modified t_μ
- Confidence belt 'pinches' towards physical boundary
- Offsetting of likelihood curves for measurements that prefer $\mu<0$



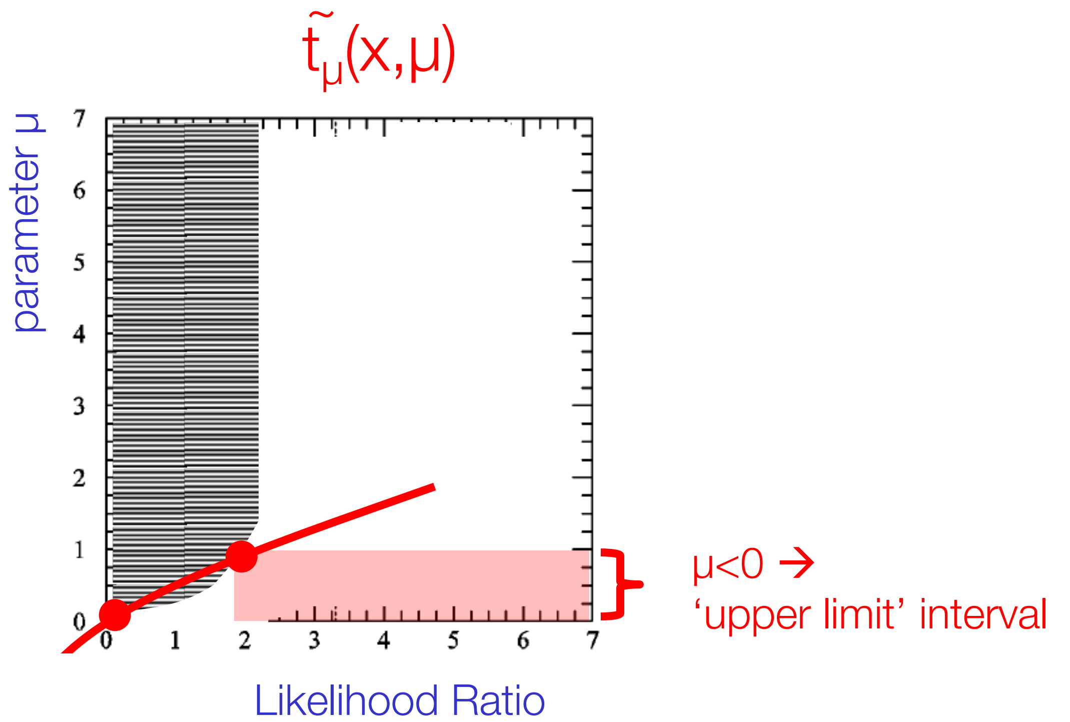
Physical boundaries frequentist confidence intervals

- Putting everything together – the confidence with modified t_μ
- Confidence belt 'pinches' towards physical boundary
- Offsetting of likelihood curves for measurements that prefer μ<0



Physical boundaries frequentist confidence intervals

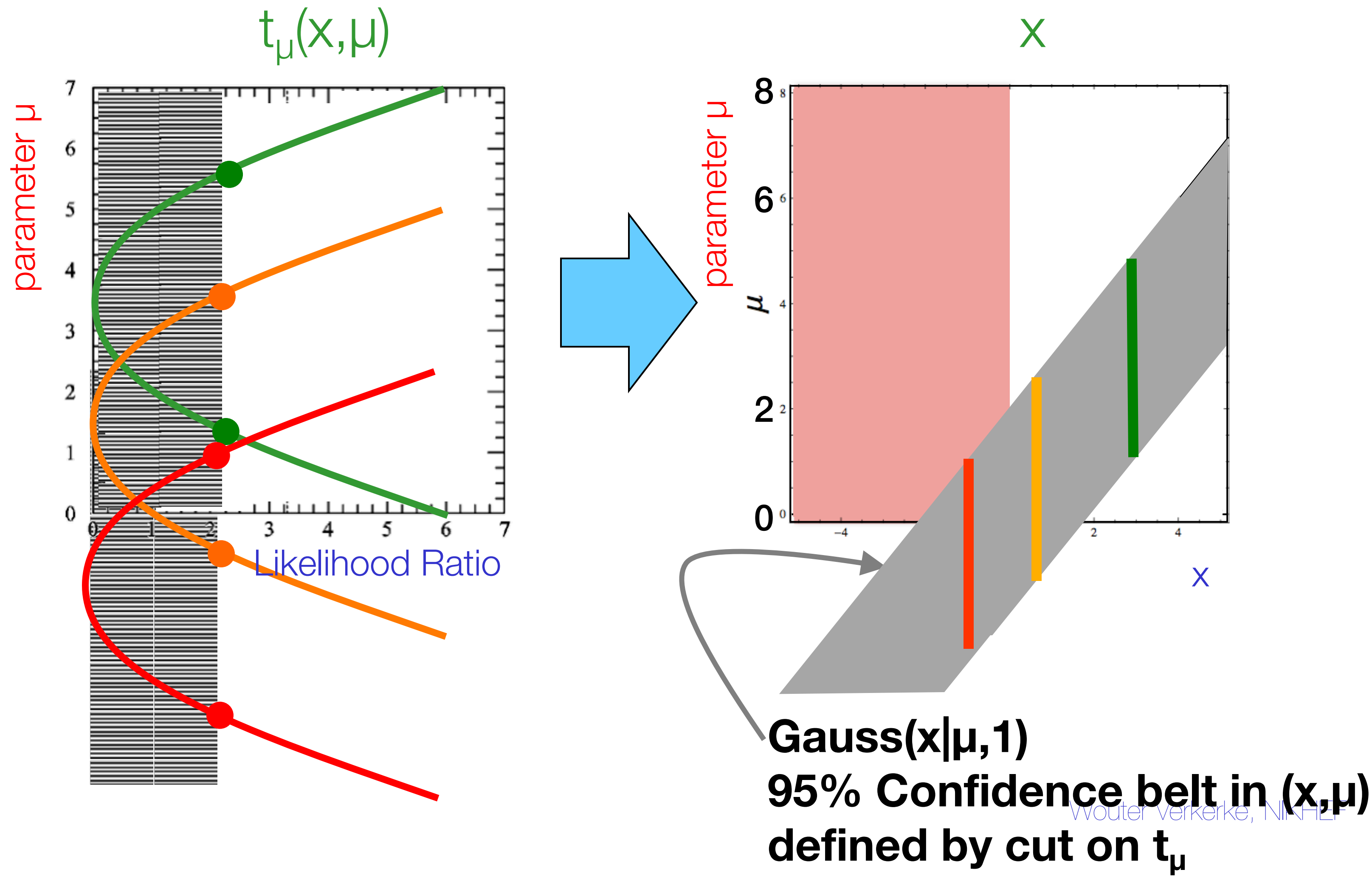
- Putting everything together – the confidence with modified t_μ
- Confidence belt 'pinches' towards physical boundary
- Offsetting of likelihood curves for measurements that prefer μ<0



Physical boundaries frequentist confidence intervals

- Example for *unconstrained* unit Gaussian measurement

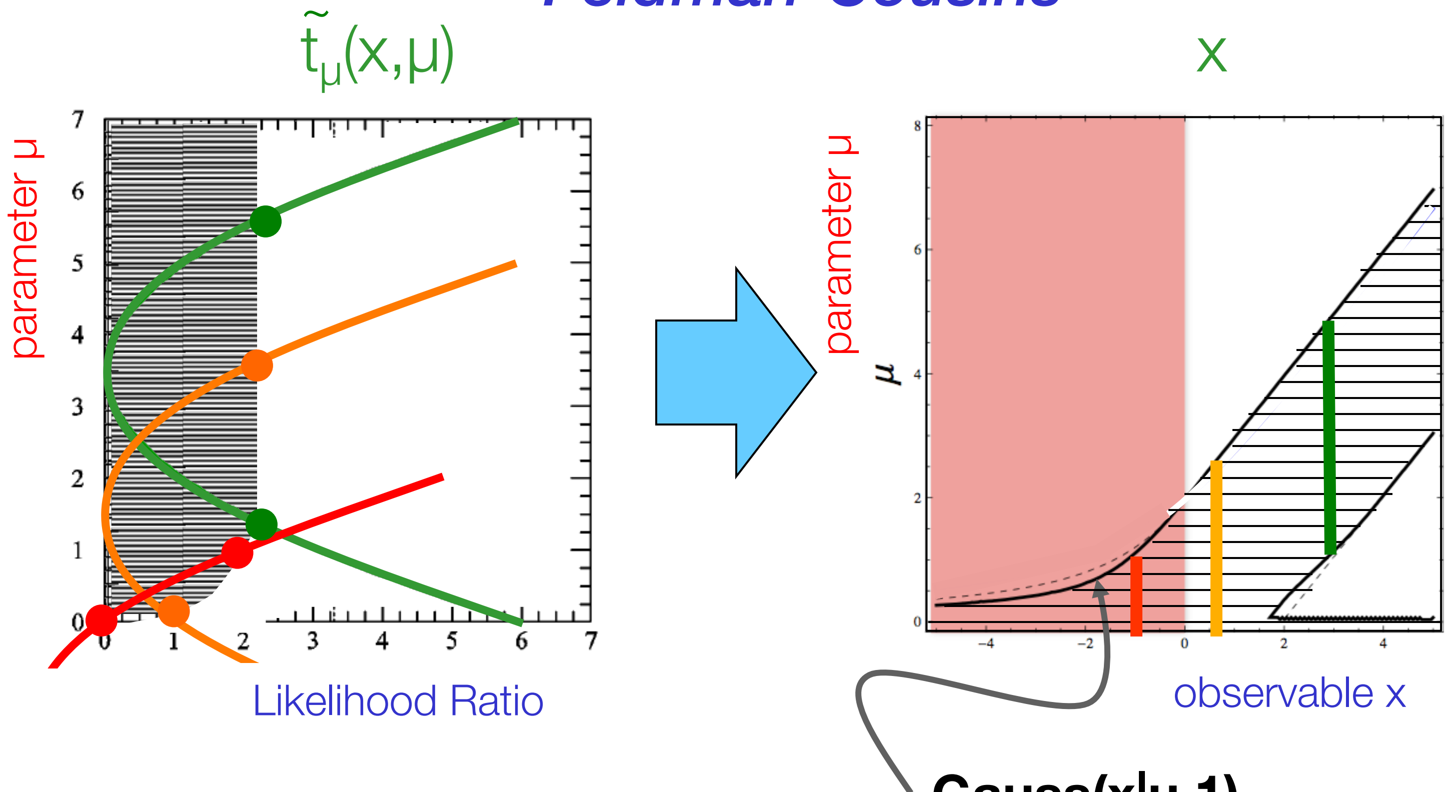
$$L = Gauss(x \mid \mu, 1)$$



Physical boundaries frequentist confidence intervals

- First map back horizontal axis of confidence belt from $t_\mu(x) \rightarrow x$

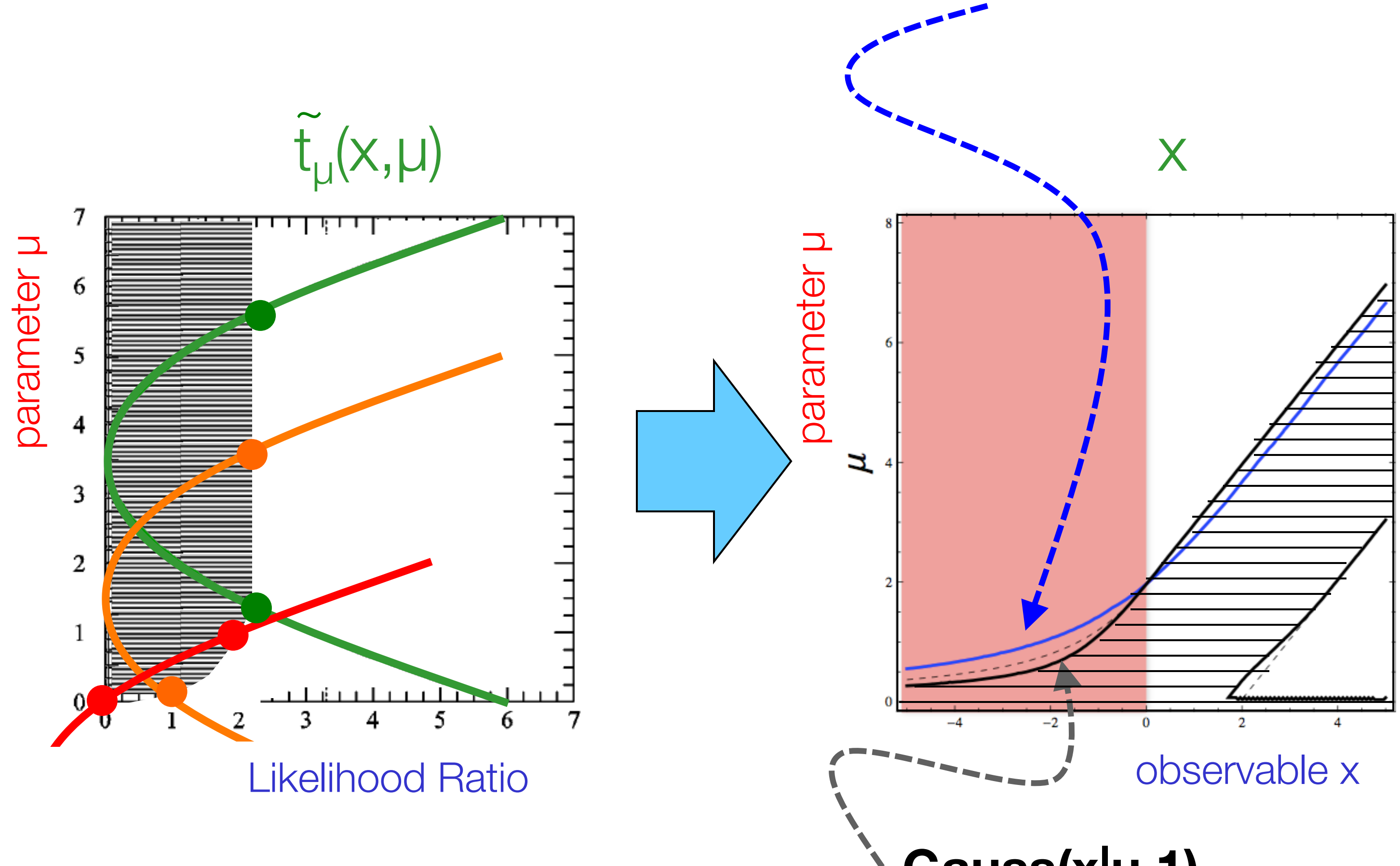
"Feldman-Cousins"



Gauss(x|μ,1)
95% Confidence belt in (x,μ)
defined by cut on $\tilde{t}_\mu$

Comparison of Bayesian and Frequentist limit treatment

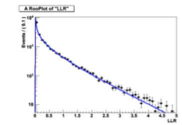
- Bayesian 95% credible upper-limit interval with flat prior μ>0



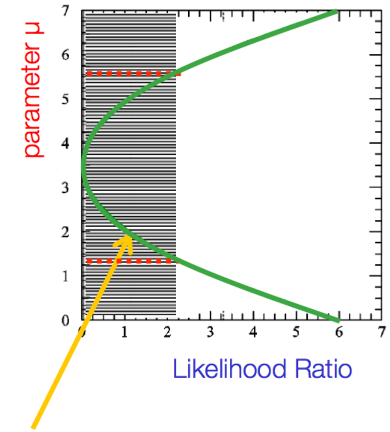
Gauss(x|μ,1)
95% Confidence belt in (x,μ)
defined by cut on t_μ for

Recap on test statistics

- The ‘default’ frequentist test statistic is the likelihood ratio t_μ
  - Confident belt (t_μ vs μ) is asymptotically a box
  - Observed value t_μ depends on μ
  - Confidence intervals as reported by MINOS
  - No notion of boundaries in parameters
- The ‘modified’ frequentist test statistics is likelihood ration t_μ
  - Confident belt will pinch near boundary in μ
  - Observed value t_μ identical to t_μ in the physical region
  - Reported interval will by construction be contained in the physical region
  - Built-in procedure that changes from 2-sided to 1-sided interval with increasing signal yield
  - Best known as ‘Feldman-Cousins’



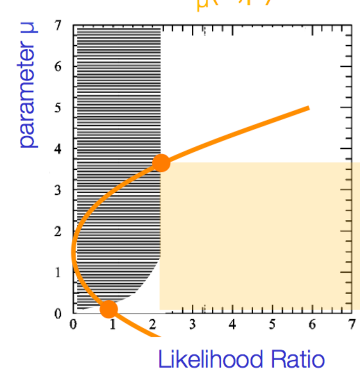
$$t_\mu(x) = -2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})}$$



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

$$\tilde{t}_\mu(x) = \begin{cases} -2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall \hat{\mu} \geq 0 \\ -2\log\frac{L(x\,|\,\mu)}{L(x\,|\,0)} & \forall \hat{\mu} < 0 \end{cases}$$

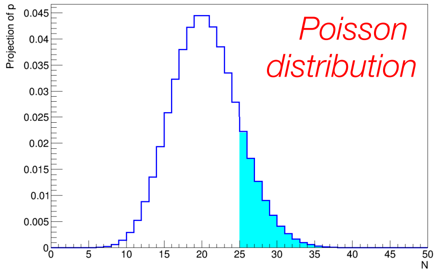
$\tilde{t}_\mu(x,\mu)$



← Small $\mu>0$ ‘upper limit’ interval

The order of things

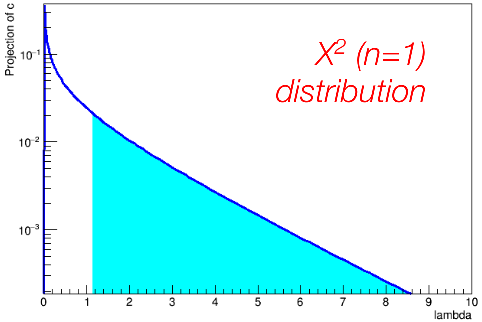
- The goal of the ‘ordering’ is to sort potential observations by signal extremity. Let’s reexamine discovery counting experiment
- For a Poisson counting distribution this is was trivial
  - Larger observed event count → more extreme



Example: B=20, N_{obs}=25

$$p_0 = \sum_{i=N_{obs}}^{\infty} Poisson(i \mid S+B) = 0.156$$

- A Likelihood-Ratio test statistic generalizes this concept to measurement of any type, but note that it quantifies the (incompatibility) of the data with a fixed hypothesis



Example: B=20, N_{obs}=25

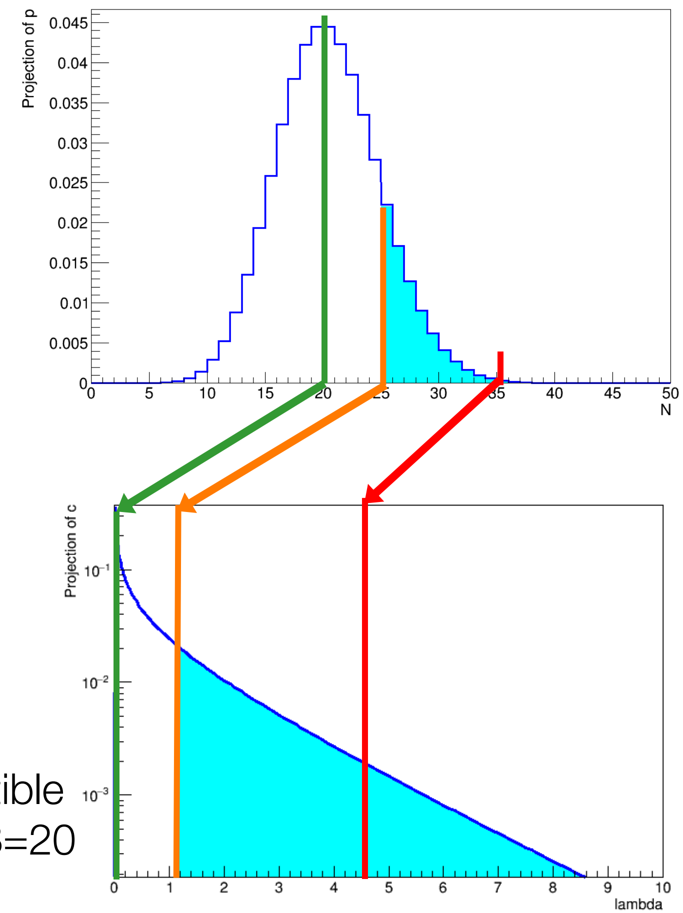
$$t_\mu = -2\log\left(\frac{Poisson(N \mid S+20)}{Poisson(N \mid \hat{S}+20)}\right) = 1.14$$

$$p_0 = \int_{t_\mu^{obs}}^{\infty} f_{\chi^2}(t_\mu)\,dt_\mu = 0.28$$

The order of things

- Why do we get a different answer?
- Because in the Likelihood Ratio test for discovery we order observations by compatibility with the hypothesis B=20

For upward fluctuations

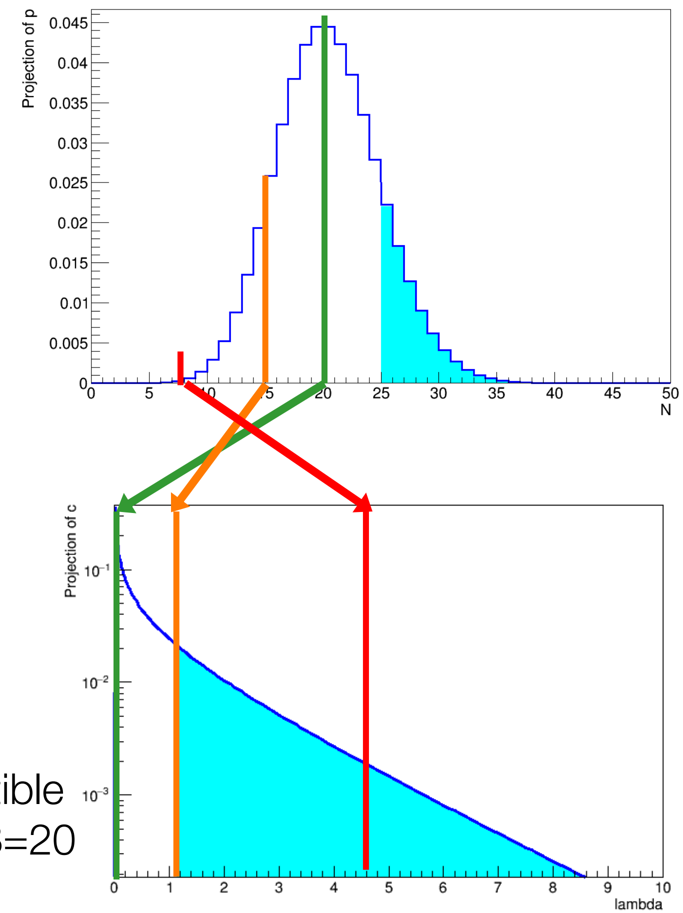


Compatible with B=20

Incompatible with B=20

The order of things

- Why do we get a different answer?
- Because in the Likelihood Ratio test for discovery we order observations by compatibility with the hypothesis B=20



For upward fluctuations

But also for downward fluctuations!

Compatible with B=20

Incompatible with B=20

This is clearly not what we intended for a discovery test!

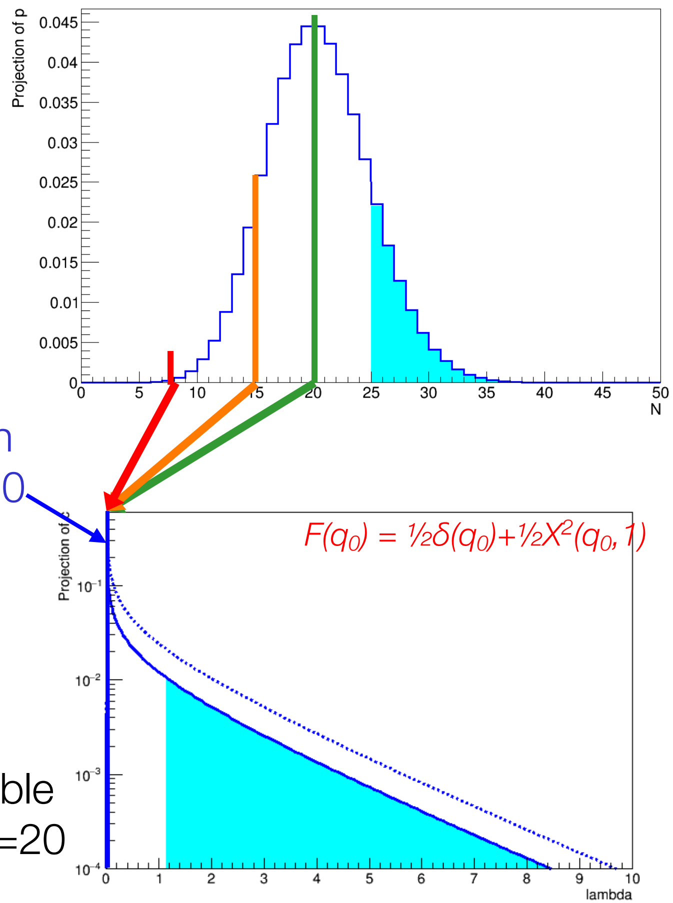
LR test measures consistency with *boundary* of a confidence interval
→ but inside boundary has different meaning than outside boundary…

Formulating a test statistic for discovery

- We can formulate a new test statistic q_0 which all negative fluctuations are considered to be maximally compatible with the background

$$q_0(x) = \begin{cases} -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall \hat{\mu} \geq 0 \\ 0 & \forall \hat{\mu} < 0 \end{cases}$$

δ-function at q_0=0



Compatible with B=20

Asymptotically half of fluctuations around null hypothesis are negative

(for small N, actual distribution may deviate from asymptotic)

Now very close to Poisson result (0.156)

(remaining difference due to discreteness of Poisson distribution)

Example: B=20, N_{obs}=25

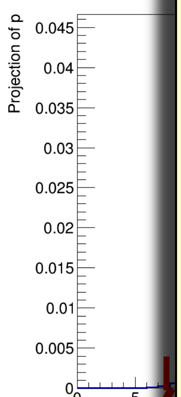
$$t_\mu = -2\log\left(\frac{Poisson(N\,|\,S+20)}{Poisson(N\,|\,\hat{S}+20)}\right) = 1.14$$

$$p_0 = \int_{t_\mu^{obs}}^{\infty} \tfrac{1}{2}\delta(t_\mu) + \tfrac{1}{2} f_{\chi^2}(t_\mu)\,dt_\mu = \int_{t_\mu^{obs}}^{\infty} \tfrac{1}{2} f_{\chi^2}(t_\mu)\,dt_\mu = 0.145$$

Formulating a test statistic for discovery

- We can formulate a new test statistic q_0 which all negative fluctuations are considered to be maximally compatible with th

$$q_0(x) = \begin{cases} -2\log\dfrac{L(x|\mu)}{L(x|\hat{\mu})} & \forall \hat{\mu} \geq 0 \\ \quad \forall \hat{\mu} < 0 \end{cases}$$



Note that q_0 is in fact *not* a new test statistic, but rather a special case of the Feldman-Cousins test statistic $\tilde{t}_\mu$!

$$q_0(x) = \begin{cases} -2\log\dfrac{L(x|\mu)}{L(x|\hat{\mu})} & \forall \hat{\mu} \geq 0 \\ 0 & \forall \hat{\mu} < 0 \end{cases}$$

$$\tilde{t}_\mu(x) = \begin{cases} -2\log\dfrac{L(x|\mu)}{L(x|\hat{\mu})} & \forall \hat{\mu} \geq 0 \\ -2\log\dfrac{L(x|\mu)}{L(x|0)} & \forall \hat{\mu} < 0 \end{cases}$$

=0 for μ=0

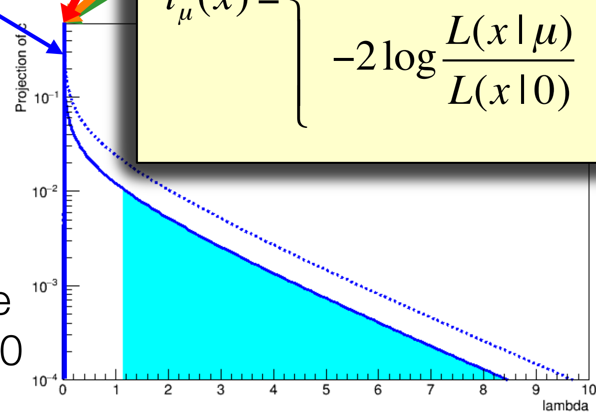
$$q_0 = \tilde{t}_0$$

d

sult (0.156)

sson distribution)

δ-function at q_0=0



Compatible with B=20

$$Poisson(N \mid \hat{S}+20)$$

$$p_0 = \int_{t_\mu^{obs}}^{\infty} \tfrac{1}{2}\delta(t_\mu) + \tfrac{1}{2} f_{\chi^2}(t_\mu)\,dt_\mu = \int_{t_\mu^{obs}}^{\infty} \tfrac{1}{2} f_{\chi^2}(t_\mu)\,dt_\mu = 0.145$$

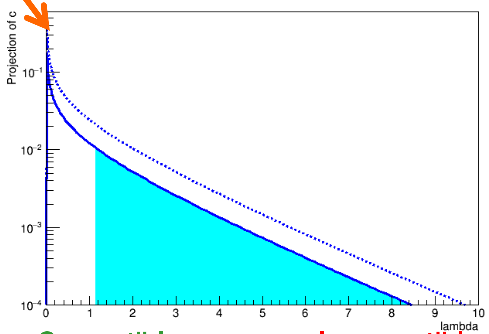
But wait… there is more

- A similar problem of dilution of sensitivity applies when considering results in the form of upper limits

Discovery

$p(\mu=0) = \ldots$

Incompatible with H_0 (N<bkg)



Compatible with H_0

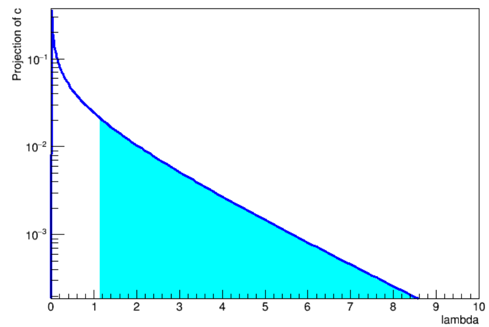
Incompatible with H_0 (N>bkg)

$$q_0(x) = \begin{cases} -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall\hat{\mu} \geq 0 \\ 0 & \forall\hat{\mu} < 0 \end{cases}$$

When considering discovery fluctuations *below* H_0 are ***not*** counted against hypothesis

Measurement

$\mu_{low} < \mu < \mu_{high}$ (68% C.L.)



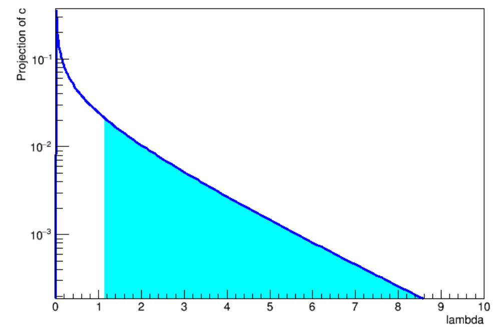
Compatible with $\hat{\mu}$

Incompatible with $\hat{\mu}$ (both dir.)

$$t_\mu(x) = 2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})}$$

Exclusion limit

$\mu < X$ (95% C.L.)



Compatible with μ(limit)

Incompatible with μ(limit) both directions

$$t_\mu(x) = 2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})}$$

When considering limit $\mu<X$ fluctuations above H_μ ***are*** counted against hypothesis

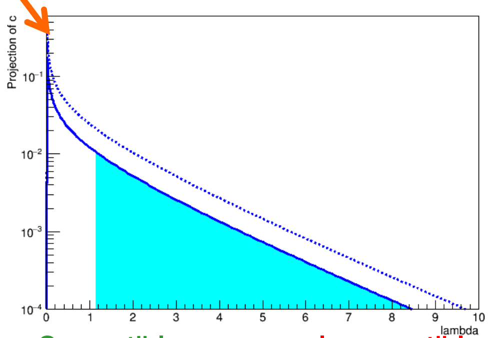
But wait… there is more

- A similar problem of dilution of sensitivity applies when considering results in the form of upper limits

Discovery

$p(\mu=0) = \ldots$

Incompatible with H_0 (N<bkg)



Compatible with H_0

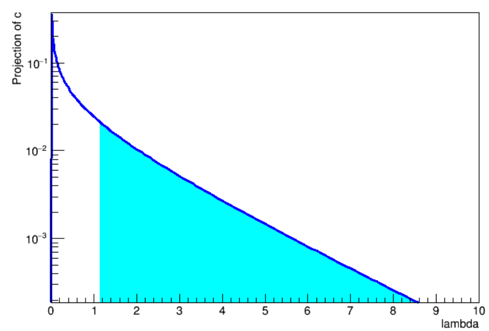
Incompatible with H_0 (N>bkg)

$$q_0(x) = \begin{cases} -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall \hat{\mu} \geq 0 \\ 0 & \forall \hat{\mu} < 0 \end{cases}$$

When considering discovery fluctuations *below* H_0 are ***not*** counted against hypothesis

Measurement

$\mu_{low} < \mu < \mu_{high}$ (68% C.L.)



Compatible with $\hat{\mu}$

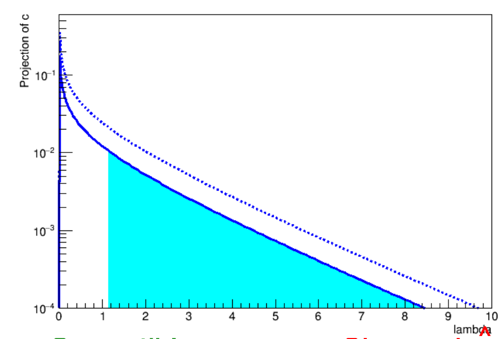
Incompatible with $\hat{\mu}$ (both dir.)

$$t_\mu(x) = 2\log\frac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})}$$

Exclusion limit

$\mu < X$ (95% C.L.)

Incompatible with H_μ ($\hat{\mu} > \mu$)



Compatible with μ(limit)

Observed $\hat{\mu}$ below μ(limit)

$$q_\mu(x) = \begin{cases} -2\log\dfrac{L(x\,|\,\mu)}{L(x\,|\,\hat{\mu})} & \forall \hat{\mu} \leq \mu \\ 0 & \forall \hat{\mu} > \mu \end{cases}$$

When considering limit μ<X fluctuations above H_μ are ***not*** counted against hypothesis

Summary of likelihood ratio test statistics

- All LR test statistics have a calibrated coverage
  - 'Size of the test' – generalization of concept of fixed 'false positive rate'

- The power of the LR test statistics depends on underlying question
  - Discovery (exclusion of H_0) → Use q_0
  - Signal exclusion (exclusion of H_μ) → Use q_μ
  - Measurement (Conf. Interval on μ) → Use t_μ

  These suppress influence of fluctuations in the 'wrong' direction

  For maximum sensitivity choose the correct one

- The discovery statistic q_0 is a special case of the 'Feldman-Cousins' test statistic t_μ
  - Bonus of feature of FC is that it automatically transitions from the optimal formulation for discovery q_0 to the optimal formulation for measurement (t_μ) as the signal strength increases (without spoiling coverage)
  - Note that while FC deals with downward fluctuations, it does not deal with upward fluctuations like q_μ → limit setting power with FC ($\tilde{t}_\mu$) is weaker than q_μ!

Summary of likelihood ratio test statistics

- All LR test statistics have a calibrated coverage
  - 'Size of the test' – generalization of concept of fixed 'false positive rate'

- The power of the LR test statistics depends on underlying question
  - Discovery (exclusion of H_0) → Use q_0
  - Signal exclusion (exclusion of H_μ) → Use q_μ
  - Measurement (Conf. Interval on μ) → Use t_μ

These suppress influence of fluctuations in the 'wrong' direction

For maximum sensitivity choose the correct one

- Features of FC and q_μ can be combined into a new test statistic q_μ:

$$\tilde{q}_\mu = \begin{cases} 0 & \hat{\mu} < 0 \\ -2\log\frac{L(\mu)}{L(\hat{\mu})} & 0 < \hat{\mu} < \mu \\ 0 & \hat{\mu} > \mu \end{cases}$$

Improved limit setting power
(upward fluctuations not counted against hypothesis μ that is being excluded)

Exclusion limit is guaranteed to be >0
(avoid all signal strengths being excluded on fluctuation below bkg-only level)

Summary of likelihood ratio test statistics

- All LR test statistics have a calibrated coverage
  - 'Size of the test' – generalization of concept of fixed 'false positive rate'

- The power of the LR test statistics depends on underlying question
  - Discovery (exclusion of H_0) → Use q_0
  - Signal exclusion (exclusion of H_μ) → Use q_μ
  - Measurement (Conf. Interval on μ) → Use t_μ

  These suppress influence of fluctuations in the 'wrong' direction

  For maximum sensitivity choose the correct one for your purpose!

- A popular (but less formal) approach to ensuring that exclusion limits do not report an empty interval in case of a fluctuation below the background-only expectation is the so-called CL_S technique

  Essence: instead of setting limit at 95% C.L. using test statistic q_μ, one aims for the 95% target in a ratio of p-values

$$CL_S(\mu) = \frac{p(\mu)}{1 - p(0)}$$

← p-value for $\hat{\mu}<\mu$

← p-value for $\hat{\mu}<0$ (since p(0) is p-value for $\hat{\mu}>0$)

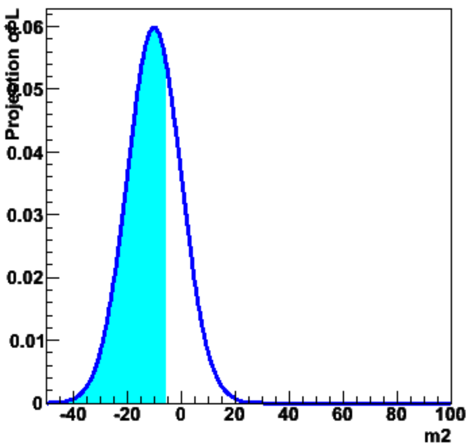
Idea: if a (negative) fluctuation is as improbable under H(0) as under H(μ) it is considered to carry no information on H(μ) that value of μ is not excluded

'ex14_roostats_cls_limit.C'

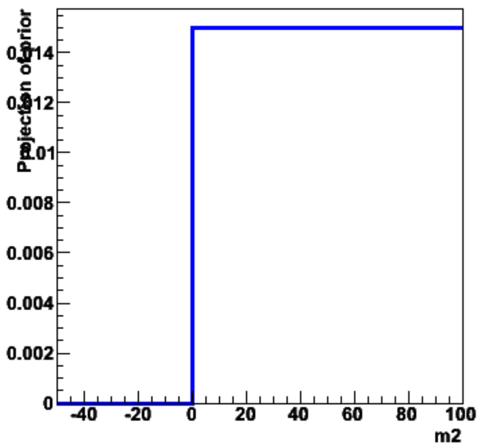
Bayesian intervals using priors to exclude unphysical regions

- Priors provide simple method to exclude unphysical regions
- Simplified example situations for a measurement of m_ν^2
  1. Central value comes out negative (= unphysical).
  2. Even upper limit (68%) may come out negative, e.g. $m^2<-5.3$,
  3. What is inference on neutrino mass, given that is must be >0?

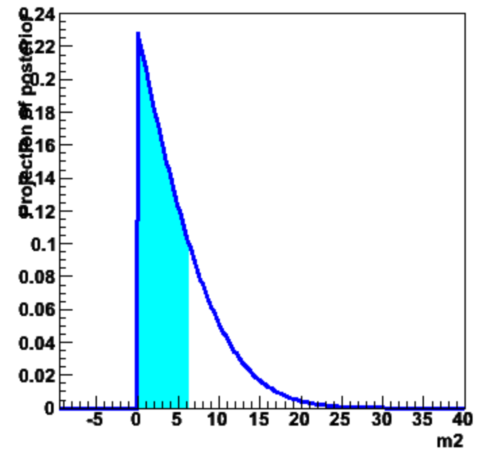
$p(\mu|x_0)$ with flat prior



$p'(\mu)$



$p(\mu|x_0)$ with $p'(\mu)$

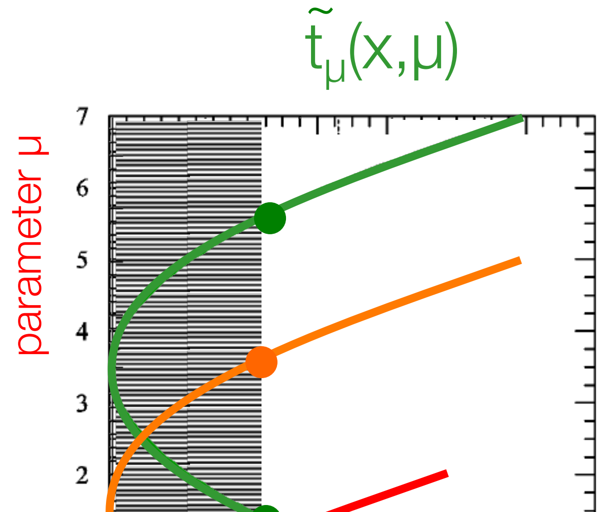


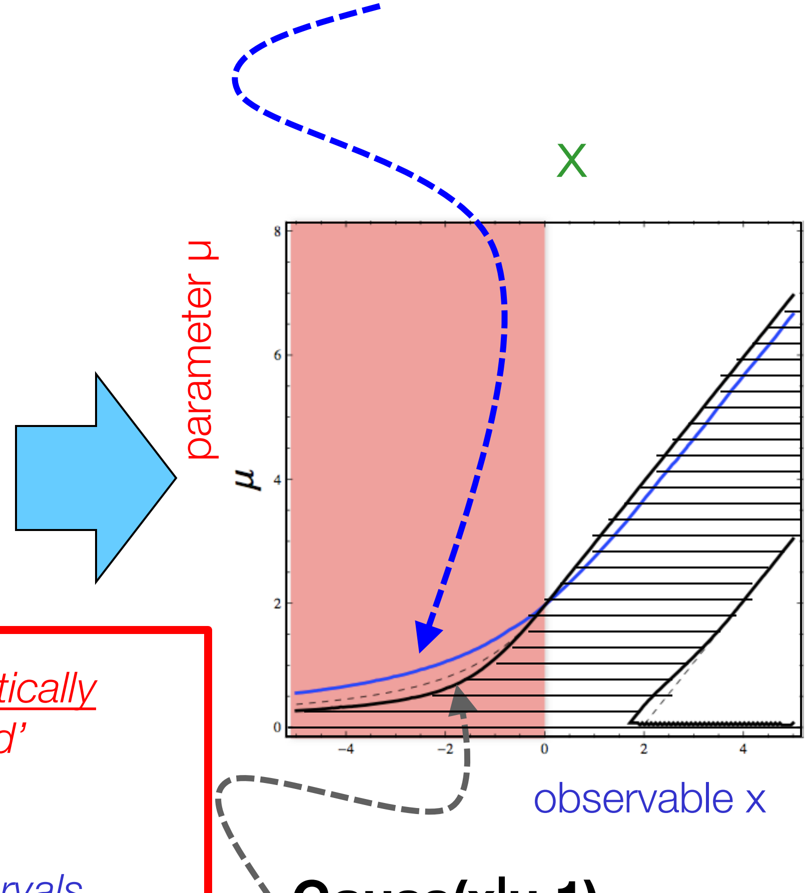
  – Introducing prior that excludes unphysical region ensure limit in physical range of observable ($m^2<6.4$)

- Beware of apparent simplicity – strong entanglement with ill-defined concept of 'flat prior'!

Numeric comparison Bayes/FC limit results for Gaussian measurement

- Bayesian 95% credible upper-limit interval with flat prior μ>0





Note that $\tilde{t}_\mu$ / Feldman-Cousins <u>automatically</u> switches from 'upper limit' to 'two-sided'
→ "unified procedure"

Note that Bayesian and Frequentist intervals at x>2 would agree exactly for Gaussian example if both would be taken as 'two-sided'

Gauss(x|μ,1)
95% Confidence belt in (x,μ) defined by cut on t_μ for

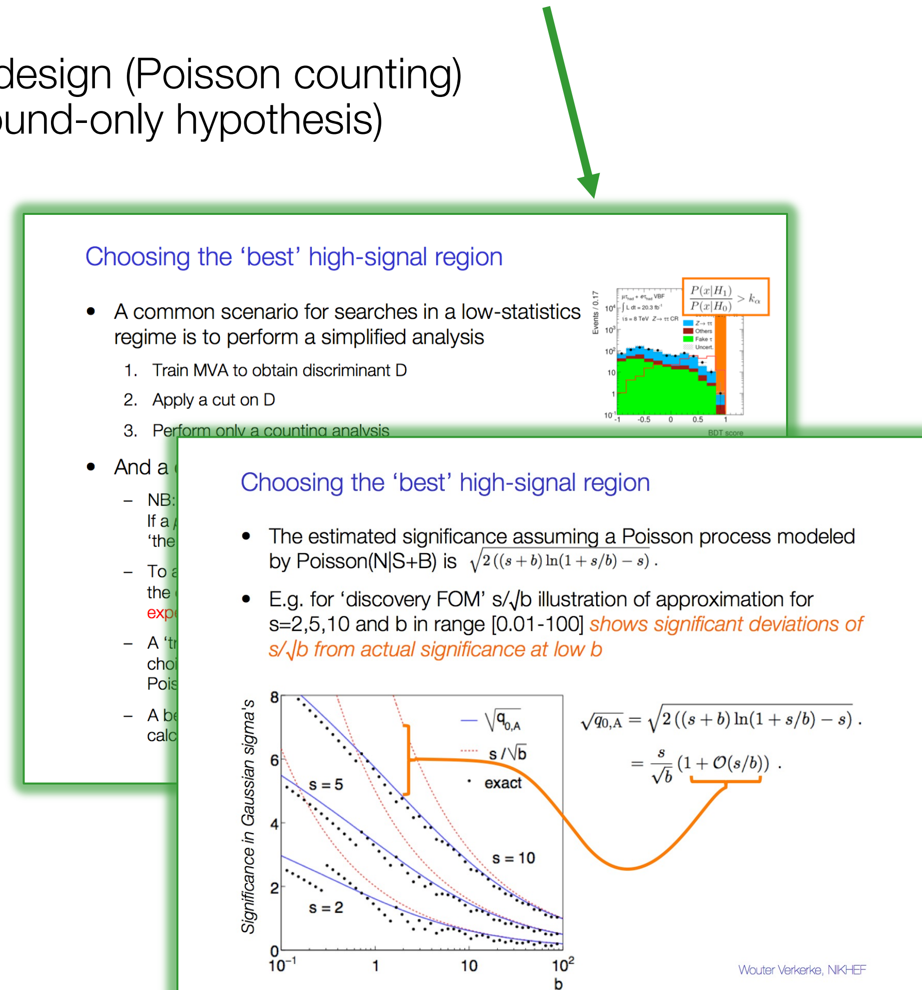
Using priors to exclude unphysical regions

- Do you want publish (only) results restricted to the physical region?
  - It depends very much to what further analysis and/or combinations is needed…
- An interval / parameter estimate that in includes unphysical still represents the best estimate of *this* measurement
  - Straightforward to combined with future measurements, new combined result might be physical (and more precise)
  - You need to decide between 'reporting outcome of this measurement' vs 'updating belief in physics parameter'
- Procedures exist to guarantee that procedures result in non-empty intervals in physics domain
  - Frequentist confidence intervals → Modified test statistics
  - Bayesian credible intervals → Priors that exclude unphysical regions
- When reporting results constrained to physical region always aim to also report unconstrained results
  - Unconstrained results carry more information for future combination/interpretation

Expected results

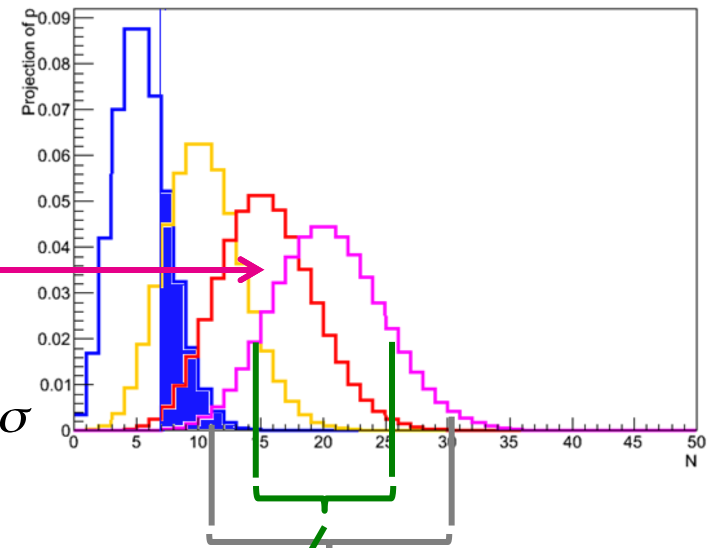
- An important part of experimental design is being able to quantify the expected sensitivity of your proposed analysis
  - Briefly touched on this already when discussing connection between LR and optimal event selection
  - Only considered simplest analysis design (Poisson counting) and one metric (p-value of background-only hypothesis)
- Will now generalize in 2 ways

1. Type of statistical models: calculate sensitivity for any type of statistical model
   - Via a LR test statistic
2. Types of output statement
   - Discovery (p0), Signal Exclusion, and Measurement
   - In addition to median expectation (of p0 etc) also calculate uncertainty interval due to expected statistical fluctuations



Expected sensitivity distributions - Poisson

- Given a Poisson counting experiment P(N|S+B) with B=5 events
- Q: What is the *median* expected p-value for a hypothetical signal S=15?



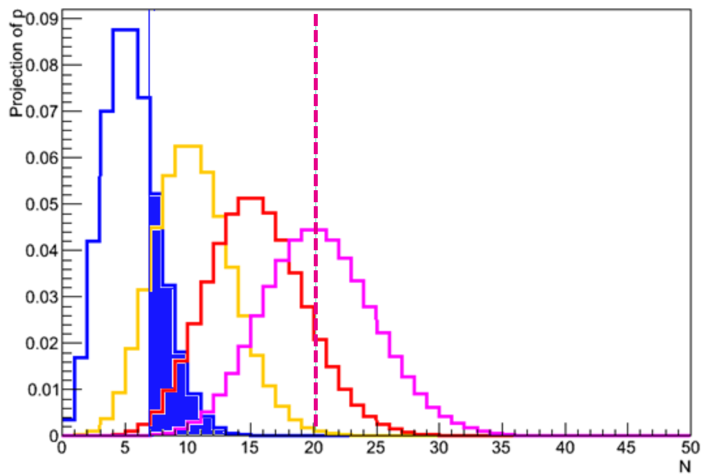
- A: $p_0 = \sum_{i=20}^{\infty} Poisson(i\,|\,5) = 2.11\cdot10^{-5} \rightarrow Z = 5.0\sigma$
- Q: What is spread in p-values for a hypothetical signal S=15?
- A: To obtain 68% (95%) intervals for p-values, **map 68%(95%) intervals of observable distribution (N) to p/Z-value intervals**

  68% interval p-values: [$6.09\ 10^{-5}$ – $8.07\ 10^{-10}$], Z [3.8-6.0]

  95% interval p-values: [$1.37\ 10^{-2}$ – $1.70\ 10^{-13}$], Z [2.2-7.2]

Expected sensitivity – comparison with Likelihood Ratio

- Compare distributions of counting experiment, direct vs LR

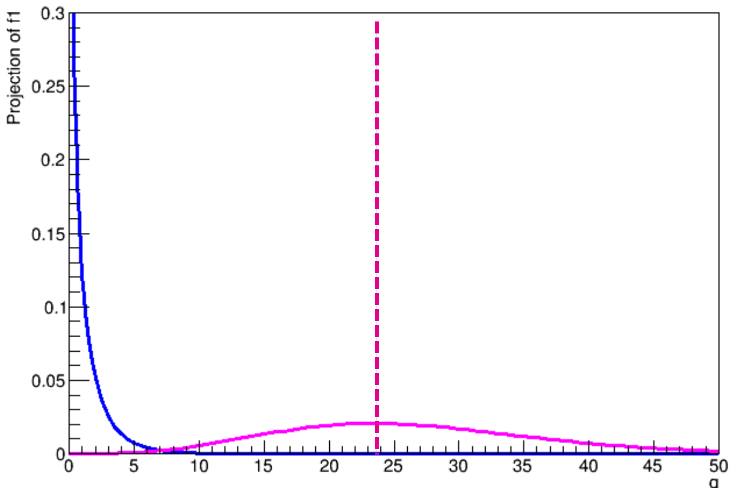


f=Poisson

Expression for Poisson distributions

$F_0(N)$=Poisson(N|0+5)

$F_{15}(N)$=Poisson(N|15+5)



$f=q_0$

Expression for discovery test statistic q_0 <u>asymptotic</u> distributions

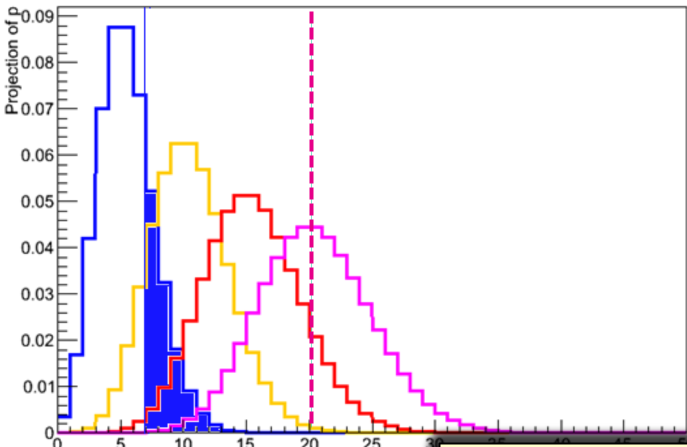
$F_0(q_0) = \quad 0.5\delta(q_0) + 0.5f_{X2}(q_0,1)$

$F_{15}(q_0) = (1-\Phi(\Lambda_{15}))\delta(q_0) + 0.5f_{NCX2}(q_0,1,\Lambda_{15})$

$\Lambda_{15} = q_0(15)$

Expected sensitivity – comparison with Likelihood Ratio

- Compare distributions of counting experiment, direct vs LR



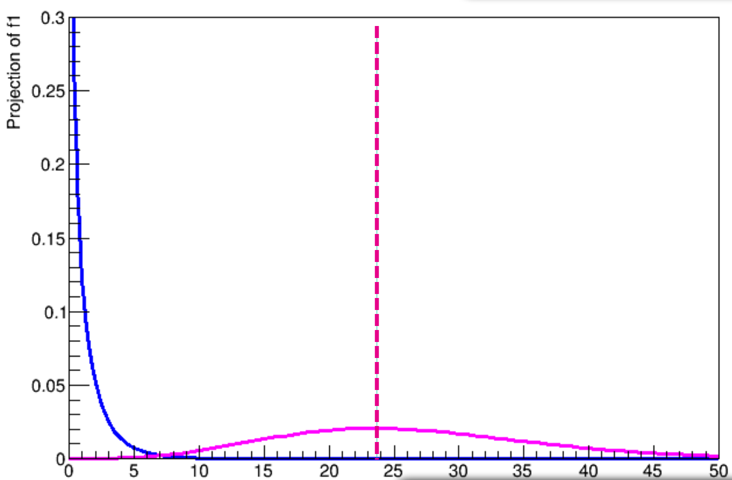
Expression for Poisson distributions

$F_0(N)=\text{Poisson}(N|0+5)$

$f_{NCX2}(x,k,\Lambda)$ = non-central X^2 distribution for k d.o.f. with impact parameter Λ

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

$f_{X2}(x,k)$ = X^2 distribution for k d.o.f.



Expression for discovery test statistic q_0 asymptotic distributions

$F_0(q_0) = \quad 0.5\delta(q_0) + 0.5 f_{X2}(q_0,1)$

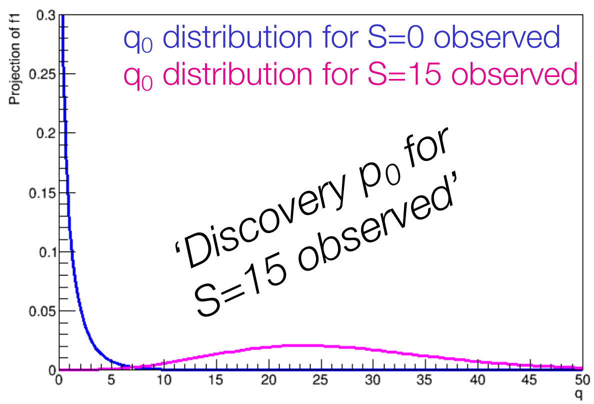
$F_{15}(q_0)= (1-\Phi(\Lambda_{15}))\delta(q_0) + 0.5 f_{NCX2}(q_0,1,\Lambda_{15})$

$\Lambda_{15} = q_0(15)$

$\Phi(x)$ = Cumulative of unit Gaussian

Expected sensitivity – Poisson Likelihood Ratio asymptotics

- If you have sufficient statistics in your measurement asymptotic expressions for distributions of $q_0(0)$ and $q_0(\mu)$ allow for *direct calculation of median significance and its statistical uncertainty*

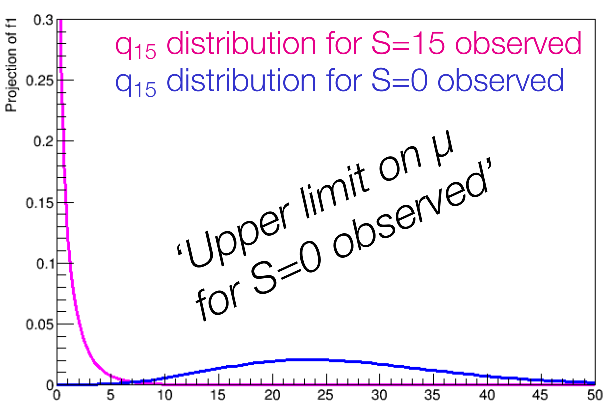


$\text{Median}[q_{0,15}] = q_0(15)$

$\text{Median}[Z_0(15)] = \sqrt{\text{Med}[q_{0,15}]} =$ **5.0σ**

68% interval = [$\sqrt{\text{Med}[q_{0,15}]}$**-1**, $\sqrt{\text{Med}[q_{0,15}]}$**+1**] = **[4.0, 6.0]**
95% interval = [$\sqrt{\text{Med}[q_{0,15}]}$**-2**, $\sqrt{\text{Med}[q_{0,15}]}$**+2**] = **[3.0, 7.0]**

- Direct calculation of median upper limit and it's statistical uncertainty

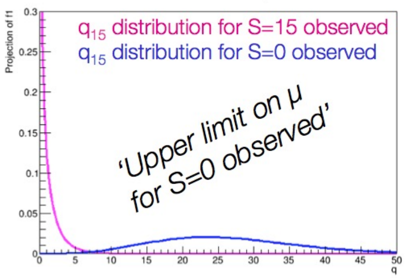


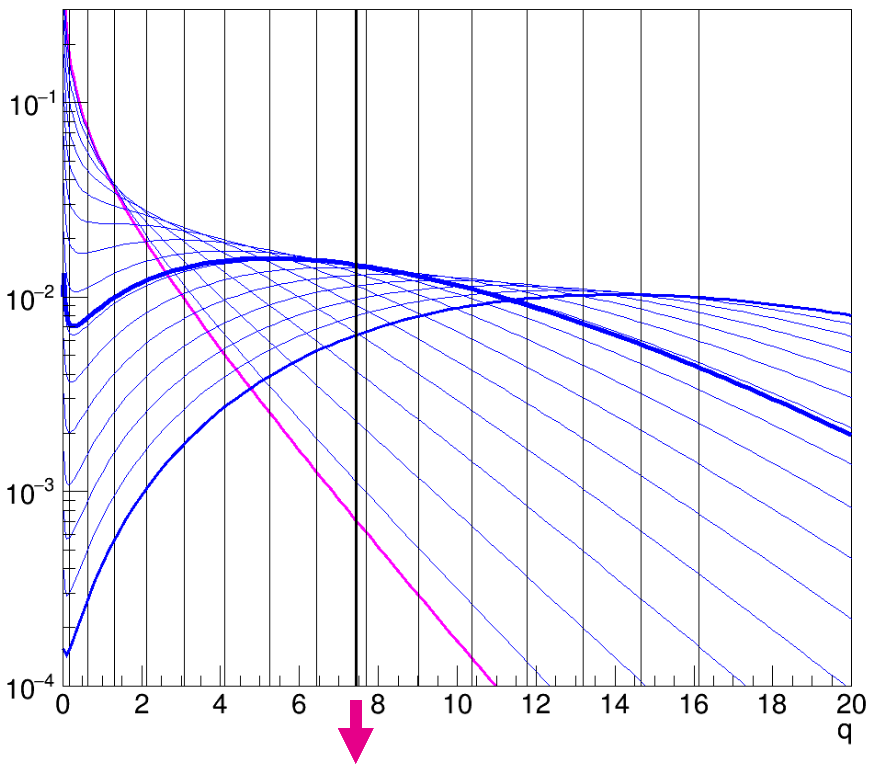
To obtain 95% excl. limit on S, find value of X that for which a test statistic $q_{\mu=X}$ for S=0 observed yields 0.05

→ No analytical solution → must scan $q_{\mu=X}$ for X=0…15

Expected sensitivity – Asymptotic upper limits

- Visualization of scanning process





$F(q_\mu|1)$ → $Med[q_\mu|1]=0.18$
$F(q_\mu|2)$ → $Med[q_\mu|2]=0.63$
…
$F(q_\mu|8.8)$ → $Med[q_\mu|8.8]=2.7$
…
$F(q_\mu|15)$ → $Med[q_\mu|1]=16.0$

Result s<8.8 at 95% C.L.

Asymptotically:

$\mu_{UL95\%}=\sigma^*\Phi^{-1}(0.95)$ → $\sigma=\mu_{UL95\%}/1.67=5.27$

$\mu_{UL95\%\pm N\sigma}=\sigma^*(\Phi^{-1}(0.95)\pm N)$

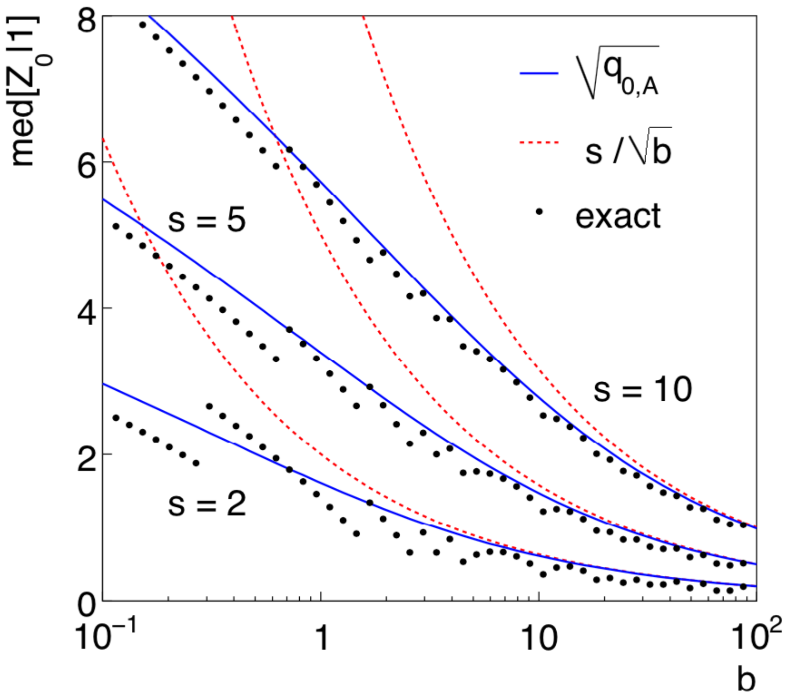
1σ band = [3.5,14.1]
2σ band = [-1.8,19.4]

p-value = 0.05 for q_μ>2.7 (defined by $f(q_\mu|\mu)$)

Expected sensitivity – Asymptotic vs Toys

- Demonstrated asymptotic formulas for
  expected discovery p0 and
  expected signal exclusions
  along with N sigma uncertainty bands for Poisson counting model

- Use of asymptotic formulas only valid in limit of sufficient statistics!



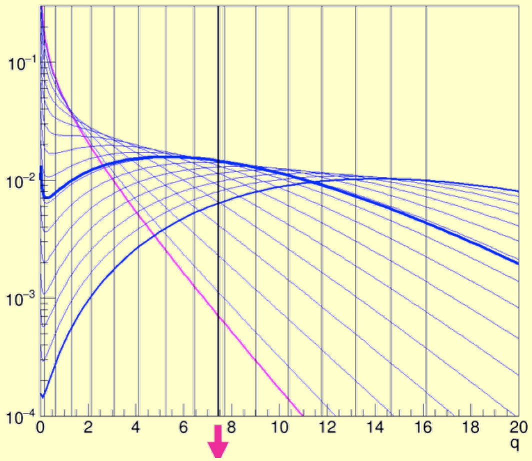
Easy to verify numerically
for counting experiments

Decent results already for N>=10!

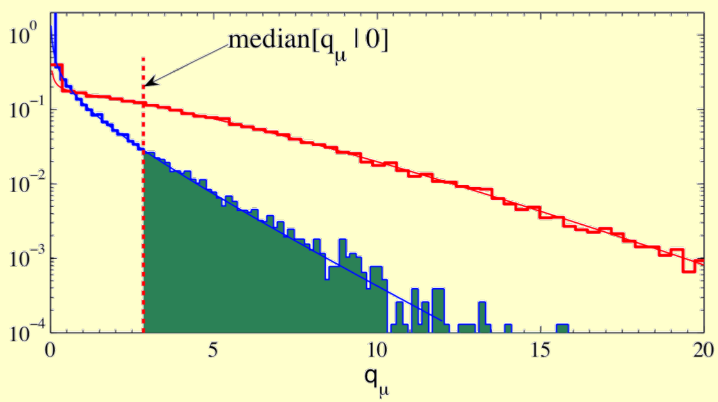
If outside validity regime
→ obtain $f(q_\mu|\mu')$ from simulation
→ very CPU intensive because
  * For 5σ discovery need, $O(10^9)$ toys
  to model tail of $f(q_0|0)$ far out
  * For 95% limits need *repeatedly* generate
  $O(10^4)$ toys to remodel distribution $f(q_\mu|\mu')$
  at *every* scan point of μ'

Expected sensitivity – Asymptotic vs Toys

Numeric limit scan:
For *every* line in this plot



Make a toy MC run to make a histogram



'ex14_roostats_cls_limit_toys.C'

ulas for

bands for Poisson counting model

y valid in limit of sufficient statistics!

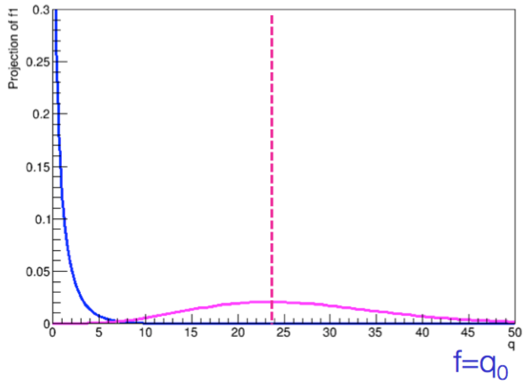
Easy to verify numerically
for counting experiments

Decent results already for N>=10!

If outside validity regime
→ obtain $f(q_\mu|\mu')$ from simulation
→ very CPU intensive because
* For 5σ discovery need, $O(10^9)$ toys
to model tail of $f(q_0|0)$ far out
* For 95% limits need *repeatedly* generate
$O(10^4)$ toys to remodel distribution $f(q_\mu|\mu')$
at *every* scan point of μ'

Expected sensitivity – Beyond counting experiments

- NB: Asymptotic formulas make use of concept '*expectation value data*' sets



Expression for discovery test statistic q_0 <u>asymptotic</u> distributions

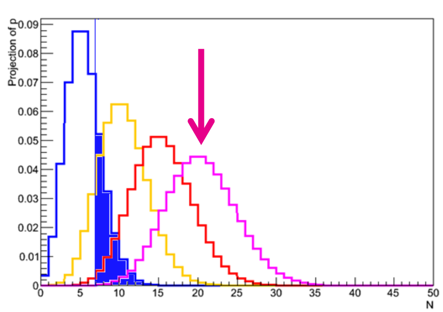
$F_0(q_0) = \quad 0.5\delta(q_0) + 0.5f_{\chi 2}(q_0,1)$

$F_{15}(q_0) = (1-\Phi(\Lambda_{15}))\delta(q_0) + 0.5f_{NC\chi 2}(q_0,1,\Lambda_{15})$

$\Lambda_{15} = q_0(15)$

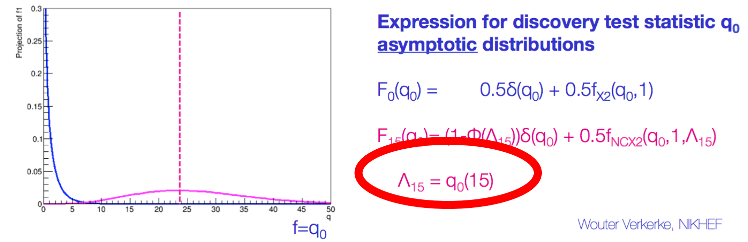
Wouter Verkerke, NIKHEF

- For counting experiments this trivial, e.g. dataset N=20, represent exactly expectation value of Poisson(N|20)



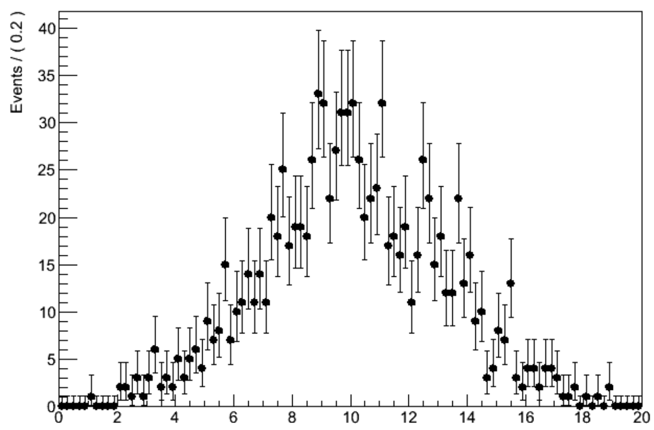
Expected sensitivity – Beyond counting experiments

- NB: Asymptotic formulas make use of concept '*expectation value data*' sets

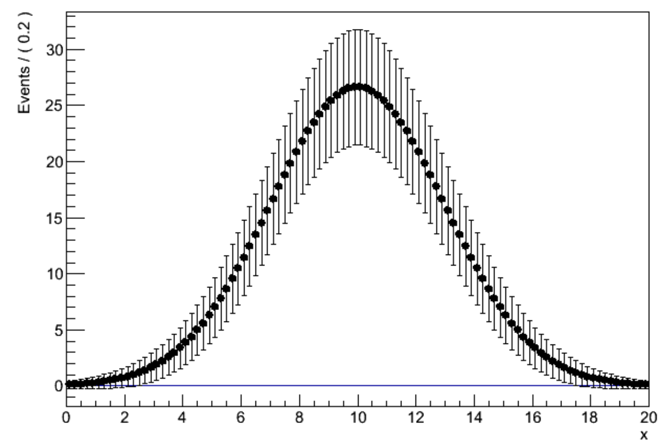


- For generic data (e.g. with distributions) an analogous concept can defined – the 'so-called Asimov dataset'
  - For example for Gaussian distribution in an observable x, the Asimov dataset is a dataset without any statistical fluctuations

'regular' sampled dataset



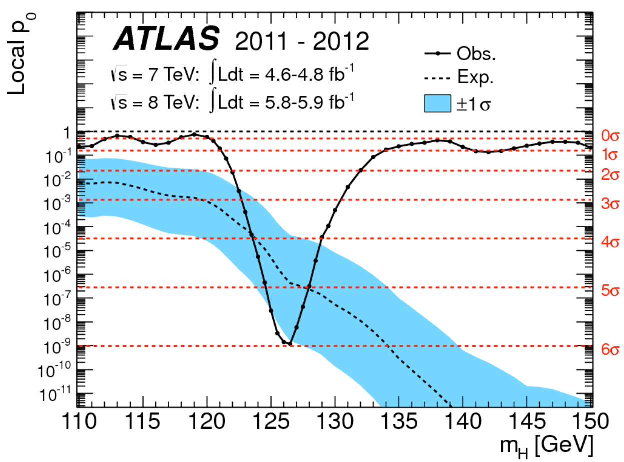
'Asimov' dataset



- Asymptotic formulas can thus be used for measurements of any shape and form (given enough statistics)

Expected results

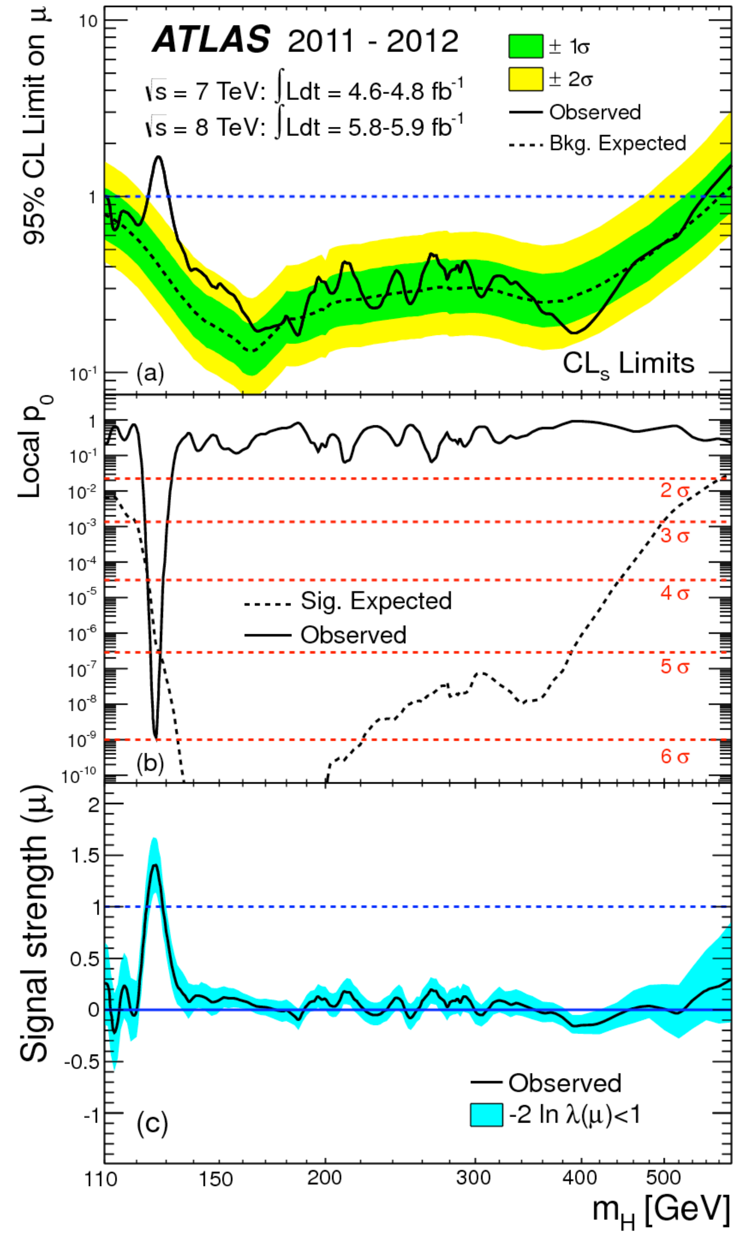
- Example plot from Higgs boson discovery



Limit

Discovery

Measurement



Software tools 2

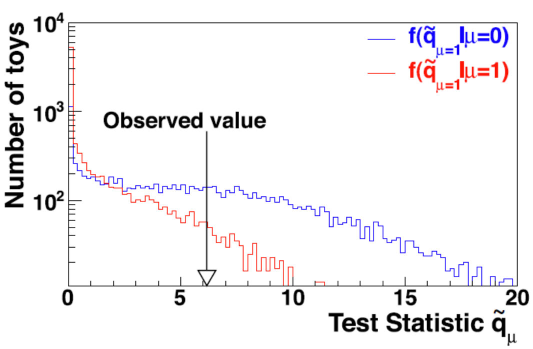
RooStats and its interface to RooFit

Wouter Verkerke, NIKHEF

Everything starts with the likelihood

Frequentist statistics

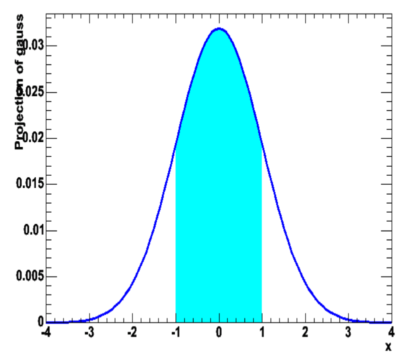
$$\lambda_\mu(\vec{N}_{obs}) = \frac{L(\vec{N}\,|\,\mu)}{L(\vec{N}\,|\,\hat{\mu})}$$



Confidence interval or p-value

Bayesian statistics

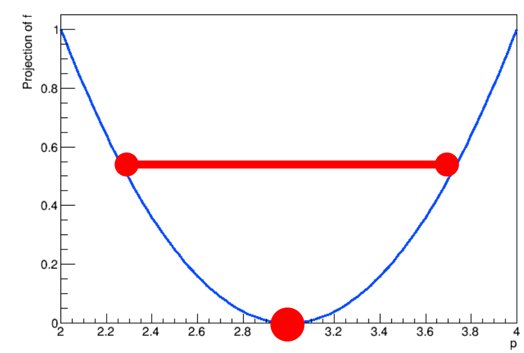
$$P(\mu) \propto L(x\,|\,\mu) \cdot \pi(\mu)$$



Posterior on s or Bayes factor

Maximum Likelihood

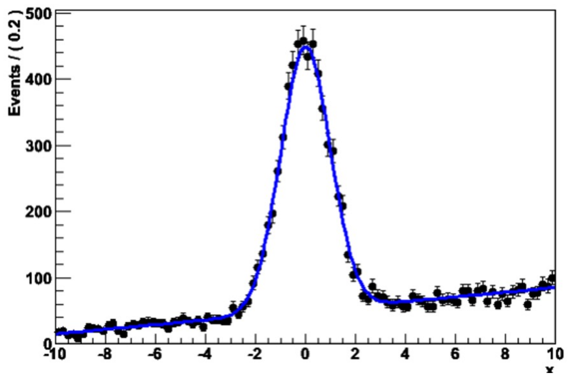
$$\left.\frac{d \ln L(\vec{p})}{d\vec{p}}\right|_{p_i=\hat{p}_i} = 0$$



s = x ± y

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

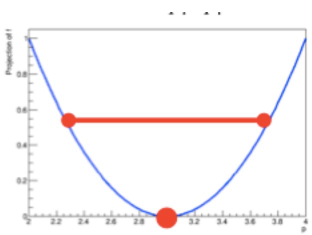


ROOT TH1

ROOT TF1

$$L(\vec{N} \mid \mu, \vec{\theta}) = \prod_i Poisson(N_i \mid f(x_i, \mu, \vec{\theta}))$$

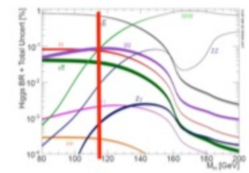
"inside ROOT"

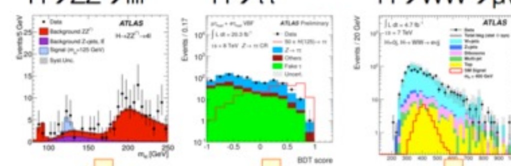


ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

$\mu = 5.3 \pm 1.7$

Higgs combination model

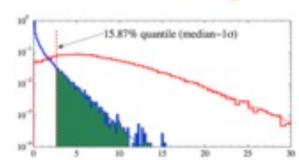




$L(\vec{N}_{ZZ} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{ZZ}, \ldots)$ $L(\vec{N}_{\tau\tau} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{\tau\tau}, \ldots)$ $L(\vec{N}_{WW} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{WW}, \ldots)$

$$L(\vec{N}_{ZZ}, \vec{N}_{\tau\tau}, \vec{N}_{WW} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{ZZ}, \ldots) \cdot \prod Poisson(N^i_{\tau\tau}, \ldots) \cdot \prod Poisson(N^i_{WW}, \ldots) \cdot \ldots$$

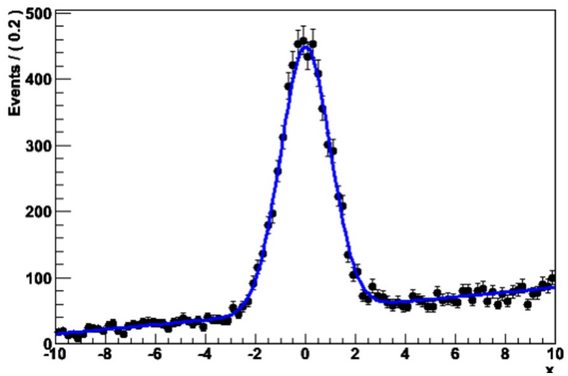
$$\lambda_\mu(\vec{N}_{ZZ}, \vec{N}_{WW}, \vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ}, \vec{N}_{WW}, \vec{N}_{\tau\tau} \mid \mu, \hat{\hat{\theta}})}{L(\vec{N}_{ZZ}, \vec{N}_{WW}, \vec{N}_{\tau\tau} \mid \hat{\mu}, \hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda \mid H_\mu)\, d\lambda = \ldots$$

How is Higgs discovery different from a simple fit?

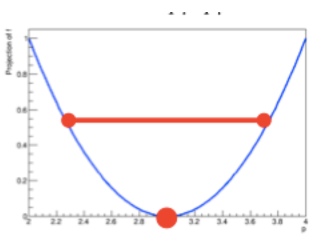
Gaussian + polynomial



ROOT TH1 ROOT TF1

$$L(\vec{N} | \mu,\vec{\theta}) = \prod_i Poisson(N_i | f(x_i,\mu,\vec{\theta}))$$

"inside ROOT"



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

$\mu = 5.3 \pm 1.7$

Likelihood Model orders of magnitude more complicated. Describes
- **O(100) signal distributions**
- **O(100) control sample distr.**
- **O(1000) parameters representing syst. uncertainties**

$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau},\vec{N}_{WW} | \mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)\cdot\prod Poisson(N^i_{\tau\tau},...)\cdot\prod Poisson(N^i_{WW},...)\cdot...$$

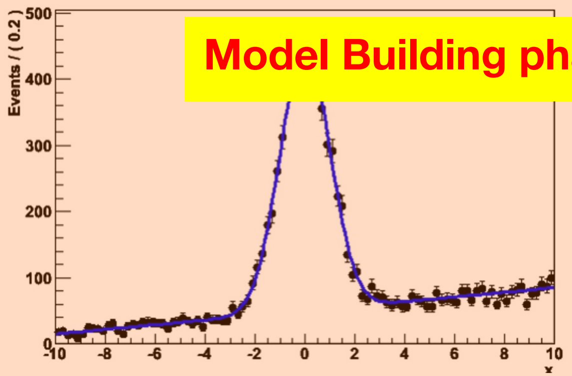
Frequentist confidence interval construction and/or p-value calculation not available as 'ready-to-run' algorithm in ROOT

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

Higgs combination model

Model Building phase (formulation of L(x|H)

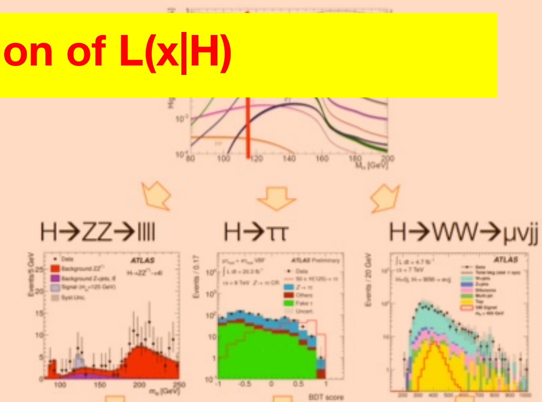


ROOT TH1

ROOT TF1

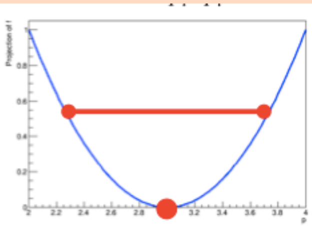
$$L(\vec{N} \mid \mu, \vec{\theta}) = \prod_i Poisson(N_i \mid f(x_i, \mu, \vec{\theta}))$$

"inside ROOT"



$L(\vec{N}_{ZZ} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{ZZ}, ...)$ $L(\vec{N}_{\tau\tau} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{\tau\tau}, ...)$ $L(\vec{N}_{WW} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{WW}, ...)$

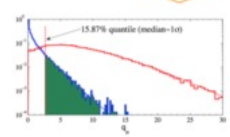
$$L(\vec{N}_{ZZ}, \vec{N}_{\tau\tau}, \vec{N}_{WW} \mid \mu, \vec{\theta}) = \prod Poisson(N^i_{ZZ}, ...) \cdot \prod Poisson(N^i_{\tau\tau}, ...) \cdot \prod Poisson(N^i_{WW}, ...) \cdot ...$$



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

$\mu = 5.3 \pm 1.7$

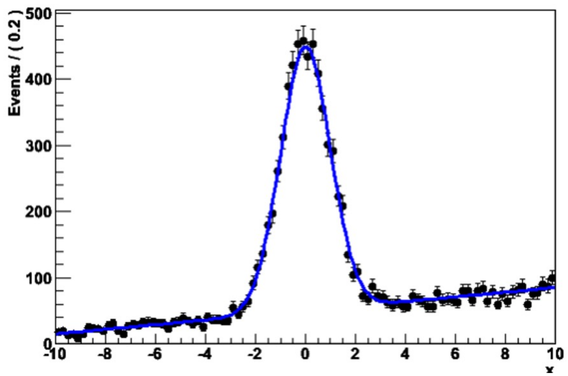
$$\lambda_\mu(\vec{N}_{ZZ}, \vec{N}_{WW}, \vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ}, \vec{N}_{WW}, \vec{N}_{\tau\tau} \mid \mu, \hat{\hat{\theta}})}{L(\vec{N}_{ZZ}, \vec{N}_{WW}, \vec{N}_{\tau\tau} \mid \hat{\mu}, \hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda \mid H_\mu) d\lambda = ...$$

How is Higgs discovery different from a simple fit?

Gaussian + polynomial



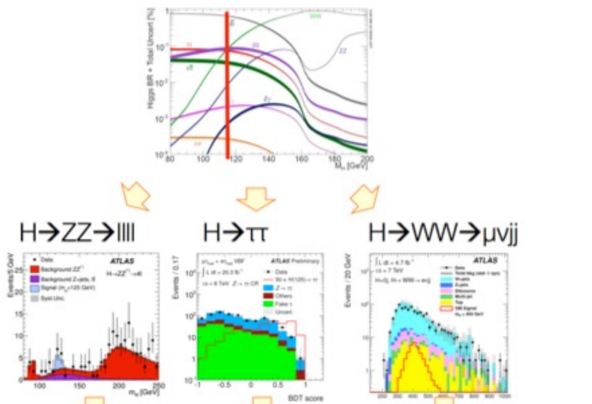
ROOT TH1

ROOT TF1

$$L(\vec{N} \mid \mu, \vec{\theta}) = \prod_i Poisson(N_i \mid f(x_i, \mu, \vec{\theta})$$

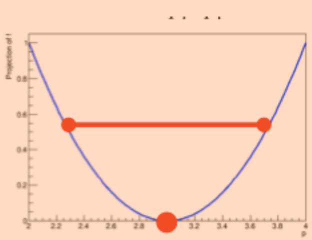
"inside ROOT"

Higgs combination model



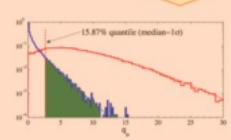
$L(\vec{N}_{ZZ} \mid \mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)$ $L(\vec{N}_{\tau\tau} \mid \mu,\vec{\theta}) = \prod Poisson(N^i_{\tau\tau},...)$ $L(\vec{N}_{WW} \mid \mu,\vec{\theta}) = \prod Poisson(N^i_{WW},...)$

$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau},\vec{N}_{WW} \mid \mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...) \cdot \prod Poisson(N^i_{\tau\tau},...) \cdot \prod Poisson(N^i_{WW},...) \cdot ...$$



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

$$\lambda_\mu(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau} \mid \mu,\hat{\hat{\theta}})}{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau} \mid \hat{\mu},\hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda \mid H_\mu) d\lambda = ...$$

Model Usage phase (use L(x|H) to make statement on H)

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

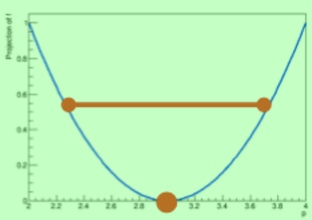
Higgs combination model

Design goal:
Separate **building of Likelihood model** as much as possible from statistical analysis **using the Likelihood model**

→ More modular software design
→ 'Plug-and-play with statistical techniques
→ Factorizes work in collaborative effort

RC

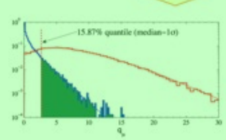
“in



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

μ = 5.3 ± 1.7

$$\lambda_\mu(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}\,|\,\mu,\hat{\hat{\theta}})}{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}\,|\,\hat{\mu},\hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda\,|\,H_\mu)\,d\lambda = \ldots$$

The idea behind the design of RooFit/RooStats/HistFactory

- Modularity, Generality and flexibility
- **Step 1 – Construct the likelihood function L(x|p)**

RooFit, or RooFit+HistFactory

- **Step 2 – Statistical tests on parameter of interest p**

  Procedure can be Bayesian, Frequentist, or Hybrid),
  but always based on L(x|p)

RooStats

- Steps 1 and 2 are conceptually separated,
  and in Roo* suit also implemented separately.

The idea behind the design of RooFit/RooStats/HistFactory

- Steps 1 and 2 can be 'physically' separated (in time, or user)
- **Step 1** – Construct the likelihood function *L(x|p)*

RooFit, or RooFit+HistFactory

RooWorkspace

Complete description of likelihood model, persistable in ROOT file (RooFit pdf function)

Allows full introspection and a-posteriori editing

- **Step 2** – Statistical tests on parameter of interest *p*

RooStats

The benefits of modularity

- Perform different statistical test on exactly the same model

RooFit, or RooFit+HistFactory

↓

RooWorkspace

↓ ↓ ↓ ↓

"Simple fit" (ML Fit with HESSE or MINOS)	**RooStats (Frequentist with toys)**	**RooStats (Frequentist asymptotic)**	**RooStats Bayesian MCMC**

Running RooStats interval calculations 'out-of-the-box'

- Confidence intervals calculated with model
  - 'Simple Fit'

```
RooAbsReal* nll = myModel->createNLL(data) ;
RooMinuit m(*nll) ;
m.migrad() ;
m.hesse() ;
```

  - Feldman Cousins (Frequentist Confidence Interval)

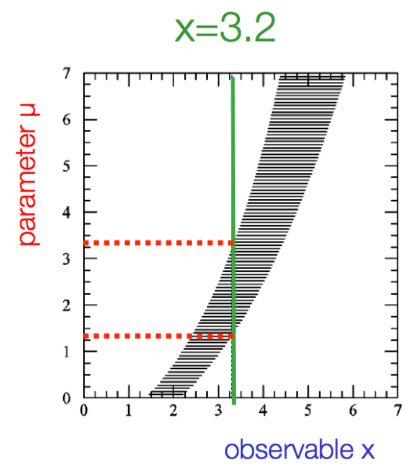
```
FeldmanCousins fc;
fc.SetPdf(myModel);
fc.SetData(data); fc.SetParameters(myPOU);
fc.UseAdaptiveSampling(true);
fc.FluctuateNumDataEntries(false);
fc.SetNBins(100); // number of points to test per parameter
fc.SetTestSize(.1);
ConfInterval* fcint = fc.GetInterval();
```

  - Bayesian (MCMC)

```
UniformProposal up;
MCMCCalculator mc;
mc.SetPdf(w::PC);
mc.SetData(data);  mc.SetParameters(s);
mc.SetProposalFunction(up);
mc.SetNumIters(100000); // steps in the chain
mc.SetTestSize(.1); // 90% CL
mc.SetNumBins(50); // used in posterior histogram
mc.SetNumBurnInSteps(40);
ConfInterval* mcmcint = mc.GetInterval();
```

But you can also look ‘in the box’ and build your own

High-level tool that constructs the confidence belt




```
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

// configure  ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();

ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

HypoTestInverter calc(*fc);
calc.UseCLs(true);

// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);

HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```

Offset advanced control over details of statistical procedure (use of CLS, choice of test statistic, boundaries…)

But you can also look 'in the box' and build your own

```
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

// configure  ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();

ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

HypoTestInverter calc(*fc);
calc.UseCLs(true);

// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);

HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```

$f(q_\mu \mid \mu')$

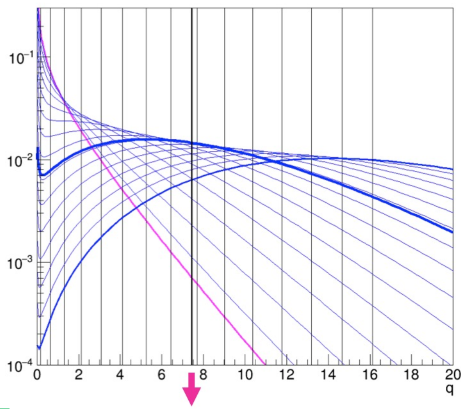
Tool to construct
test statistic distribution

$q_\mu(\mu')$

The test statistic
to be used for
the calculation
of p-values

Offset advanced control over details of statistical procedure (use of CLS, choice of test statistic, boundaries…)

But you can also look 'in the box' and build your own



Tool to scan over values of μ to find a q_μ that results in a p-value of 0.05 (for 95% C.L.)

```
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

// configure  ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();

ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

HypoTestInverter calc(*fc);
calc.UseCLs(true);

// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);

HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```

Offset advanced control over details of statistical procedure (use of CLS, choice of test statistic, boundaries…)

But you can also look 'in the box' and build your own

```
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

// configure  ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();

ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

HypoTestInverter calc(*fc);
calc.UseCLs(true);

// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);

HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```

Optionally choose a technique to avoid *spurious exclusions* (*all* at 95% C.L. signal excluded due to low fluctuation)

Options are
1) FC-style test stat q_μ
2) CLS: calculate p-value from q_μ divide by p-value of bkg hypothesis in scan for 95% point.

Offset advanced control over details of statistical procedure (use of CLS, choice of test statistic, boundaries…)

But you can also look ‘in the box’ and build your own

```
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

// configure  ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();

ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

HypoTestInverter calc(*fc);
calc.UseCLs(true);
// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);

HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```

Run calculation

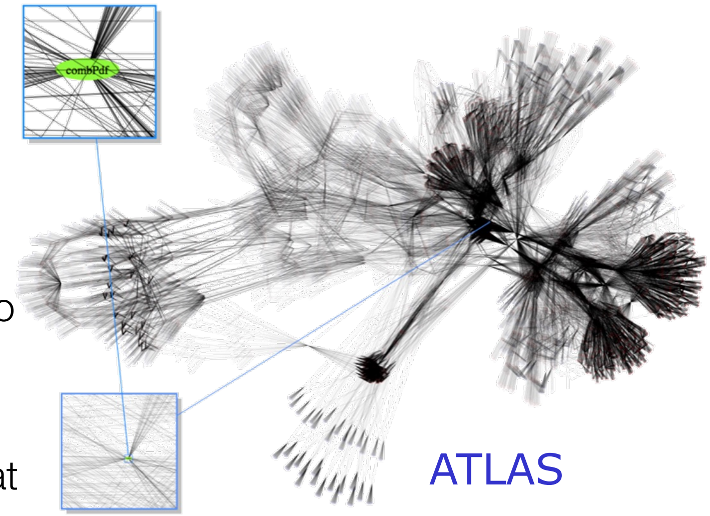
Extract result

Make optional plot

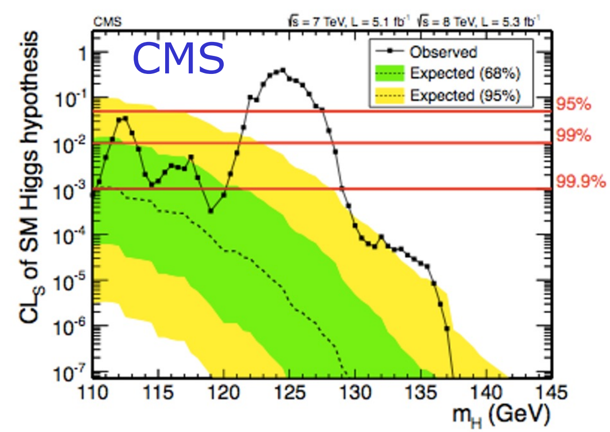
Offset advanced control over details of statistical procedure (use of CLS, choice of test statistic, boundaries…)

Summary

- RooFit and RooStats allow you to perform advanced statistical data analysis
  - LHC Higgs results a prominent example



- RooFit provides (almost) limitless model building facilities
  - Concept of persistable model workspace allows to separate model building and model interpretation
  - HistFactory package introduces structured model building for binned likelihood template models that are common in LHC analyses
- Concept of RooFit Workspace has completely restructured HEP analysis workflow with ‘collaborative modeling’
- RooStats provide a wide set of statistical tests that can be performed on RooFit models
  - Bayesian, Frequentist and Likelihood-based test concepts

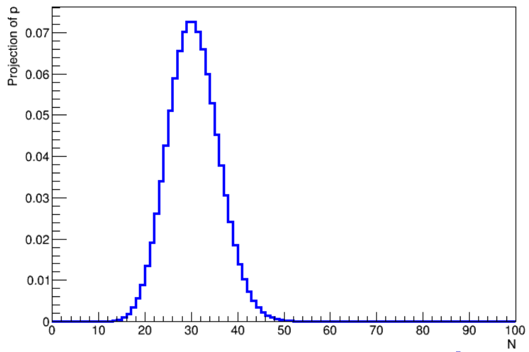


'ex09_build_Poisson.C'

Full demo of RooFit/RooStats calculation

- Phase 1 – Build model (here just a Poisson), **prepare for use**

```
RooWorkspace w("w") ;

// Construct a single Poisson model P(N|mu*S+B)
w.factory("Poisson::model('mu*S+B',mu[1,-1,10],S[10],B[20])") ;
w.factory("expr::Nexp( (Nobs[0,100],Nexp)") ;

// Construct a dataset containing N=25
RooDataSet d("d","d",*w.var("Nobs")) ;
w.var("Nobs")->setVal(25) ;
d.add(*w.var("Nobs")) ;
w.import(d,RooFit::Rename("observed_data")) ;

// Construct interpretatation of model used by RooStats
RooStats::ModelConfig mc("ModelConfig",&w);

// Define the pdf, the parameter of interest and the observables
mc(*w.pdf("model"));
mc.SetParametersOfInterest(*w.var("mu"));
mc.SetObservables.SetPdf (*w.var("Nobs"));

// Define the current value mu (1) as an hypothesis
mc.SetSnapshot(*w.var("mu"));

// import model in the workspace
w.import(mc);
w.writeToFile("model.root") ;
```



Poisson::model(N_{obs}|μS+B)

$f(N|\mu)$ = model
POI=μ
obs=N_{obs}

H_1=model(μ=1)
H_0=model(μ=0) [implicit]

Full demo of RooFit/RooStats calculation

- Phase 2 – Perform limit calculation

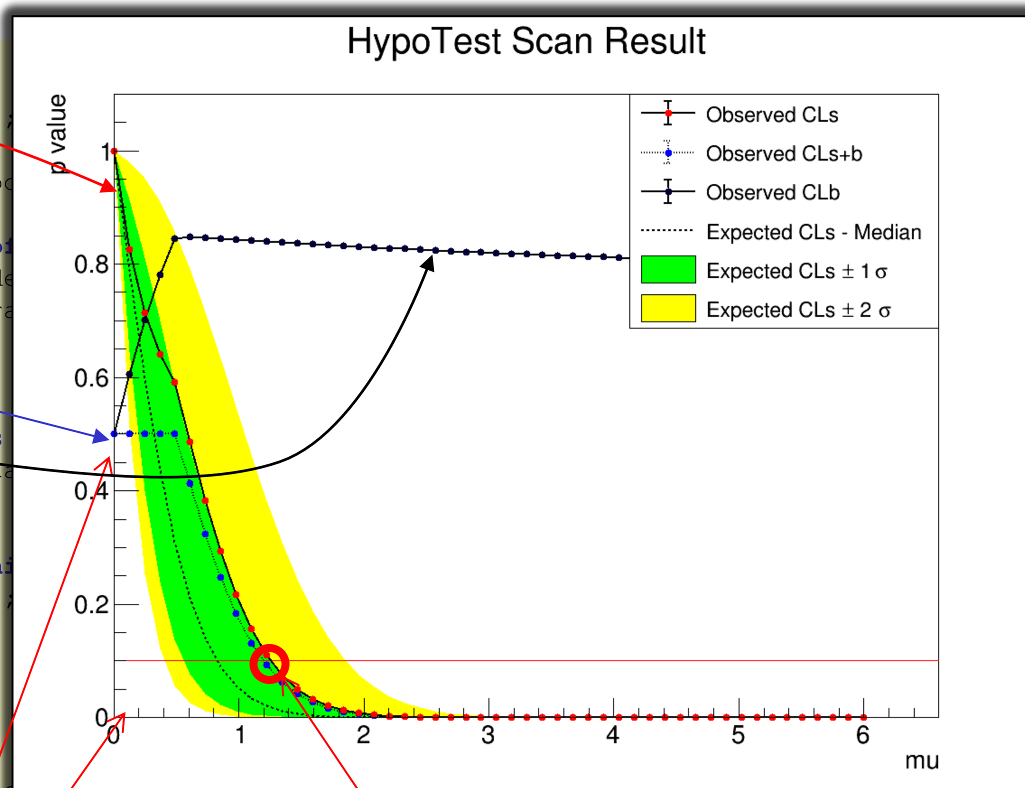
'ex10_roostats_plr_interval.C'
'ex10_roostats_bayes_interval.C'
'ex14_roostats_cls_limit.C'
'ex14_roostats_cls_limit_toys.C'

```
// Retrieve components
RooWorkspace* w = (RooWorkspace*) f->Get("w") ;
RooAbsData* data = w->data("observed_data") ;
RooStats::ModelConfig* sbModel = (RooStats::ModelConfig*) w->obj("ModelConfig") ;

// Construct B-only model (for CLS) as clone of P(N|muS+B) with B=0
RooStats::ModelConfig* bModel = (RooStats::ModelConfig*) sbModel->Clone("BonlyModel") ;
RooRealVar* poi = (RooRealVar*) bModel->GetParametersOfInterest()->first();
poi->setVal(0) ;
bModel->SetSnapshot( *poi  );

// Use calculator based on asymptotic formulas
RooStats::AsymptoticCalculator  asympCalc(*data, *bModel, *sbModel);
asympCalc.SetOneSided(true);

// Request 90% C.L. upper limit with CLS technique enabled
RooStats::HypoTestInverter inverter(asympCalc);
inverter.SetConfidenceLevel(0.90);
inverter.UseCLs(true);

// Run interval calculation
inverter.SetVerbose(false);
inverter.SetFixedScan(50,0.0,6.0); // set number of points , xmin and xmax
RooStats::HypoTestInverterResult* result =  inverter.GetInterval();

// Report results
cout << 100*inverter.ConfidenceLevel() << "%  upper limit : " << result->UpperLimit() << endl;
std::cout << "Expected upper limits, using the B (alternate) model : " << std::endl;
std::cout << " expected limit (median) " << result->GetExpectedUpperLimit(0) << std::endl;
std::cout << " expected limit (-1 sig) " << result->GetExpectedUpperLimit(-1) << std::endl;
std::cout << " expected limit (+1 sig) " << result->GetExpectedUpperLimit(1) << std::endl;
```

Full demo of RooFit/RooStats calculation

- Phase 2 – Perform limit calculation

```
// Retrieve components
RooWorkspace* w = (RooWorkspace*) f->Get("w") ;
RooAbsData* data = w->data("observed_data") ;
RooStats::ModelConfig* sbModel = (RooStats::Mo

// Construct B-only model (for CLS) as clone of
RooStats::ModelConfig* bModel = (RooStats::Mode
RooRealVar* poi =                         >GetPara
poi->setVal(0) ;
bModel->SetSnapsh

// Use calculator                         ormulas
RooStats::AsymptoticCalculator  asympCalc(*data
asympCalc.SetOneSided(true);

// Request 90% C.L. upper limit with CLS techni
RooStats::HypoTestInverter inverter(asympCalc);
inverter.SetConfidenceLevel(0.90);
inverter.UseCLs(true);

// Run interval calculation
inverter.SetVerbose(false);
inverter.SetFixedScan(50,0.0,6.0); // set number of points , xmin and xmax
RooStats::HypoTestInverterResult* result =  inverter.GetInterval();

// Report resu
cout << 100*in                            %  upper limi
std::cout << "                            the B (altern
std::cout << "                            result->GetE
std::cout << "                            result->GetE
std::cout << " expected limit (+1 sig) " << result->GetE
```



CL_S ratio divides **p(s+b)** by **p(b)**

AsymptoticCalculator calculates **p-values** for given hypothesis **μ**

Hypothesis inverter finds **intersection** of CLS with target p-value (0.10) for 90% C.L.