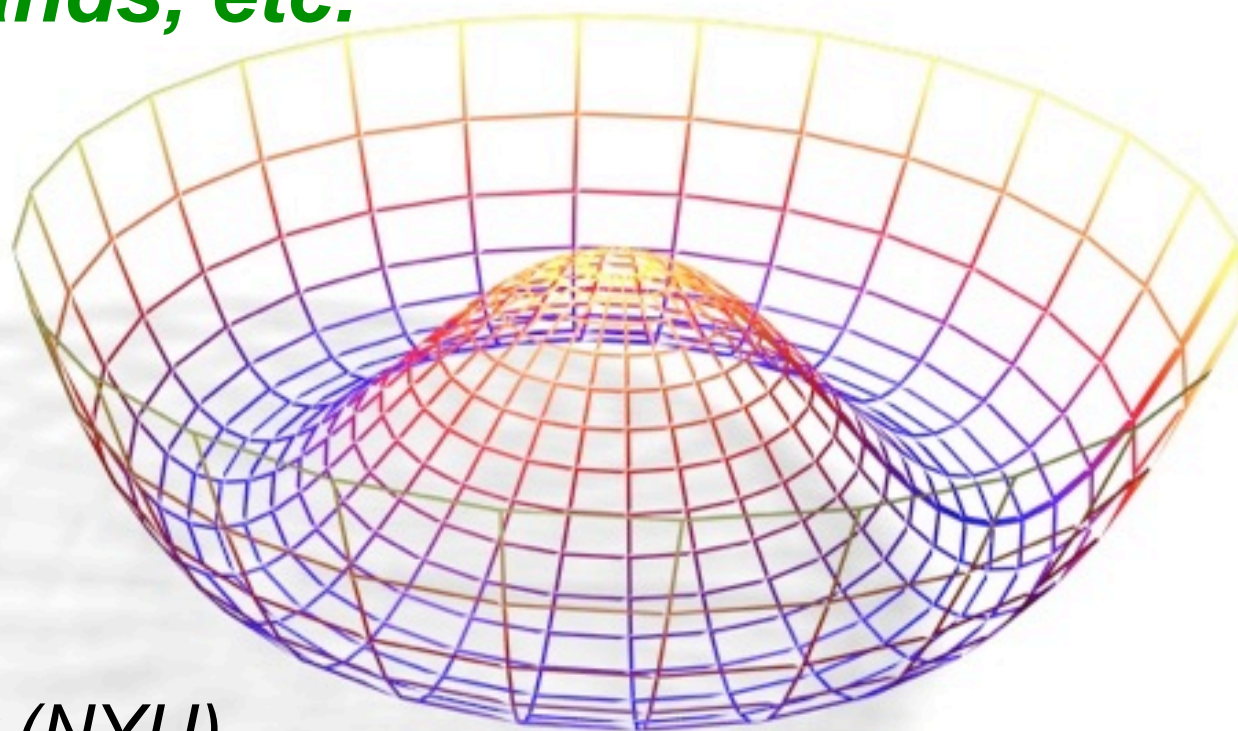




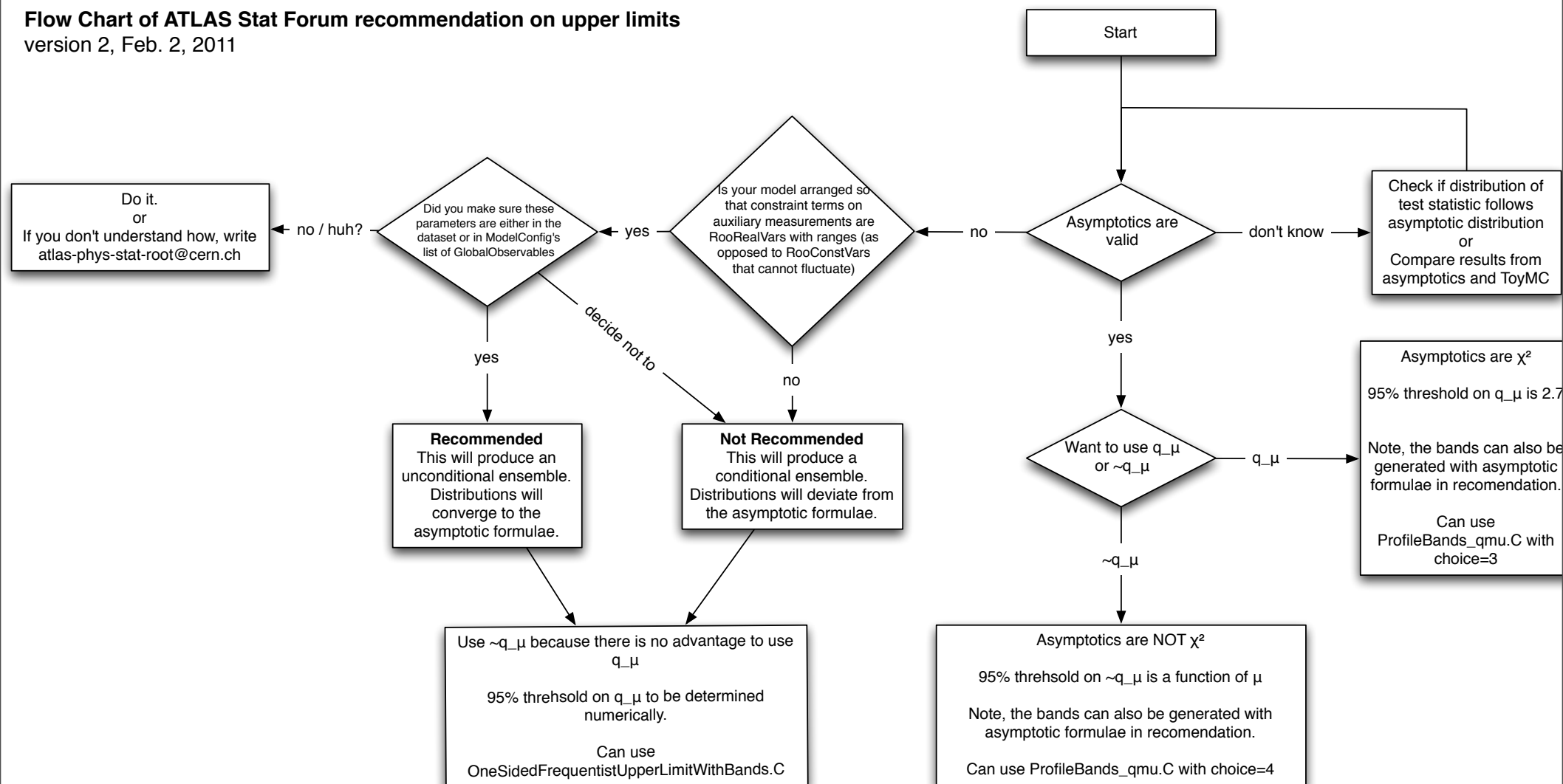
RooStats updates based on recent experience: Bands, etc.



Kyle Cranmer (NYU)



Flow Chart of ATLAS Stat Forum recommendation on upper limits version 2, Feb. 2, 2011





Putting changes described here into patches release:

- <http://root.cern.ch/drupal/content/root-version-v5-28-00-patch-release-notes>

Draft 1 of HistFactory Users' Guide:

- <https://twiki.cern.ch/twiki/pub/RooStats/WebHome/HistFactoryLikelihood.pdf>

Should be built as 5.28.a in the next week or so

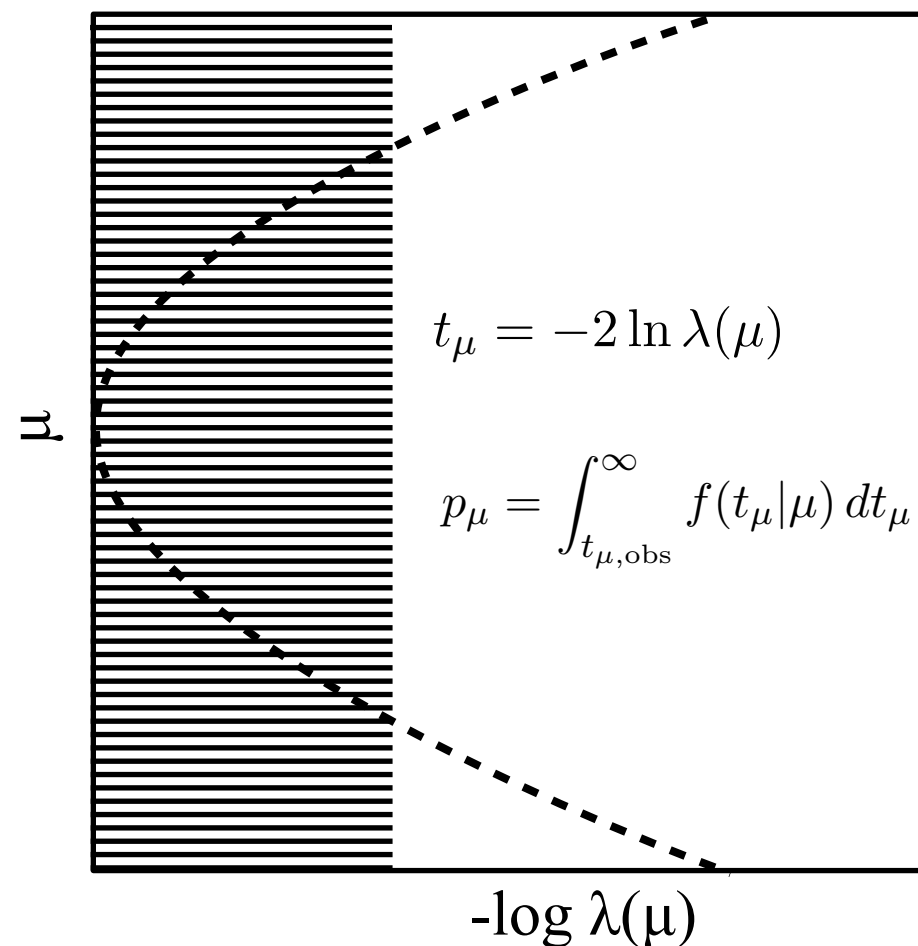
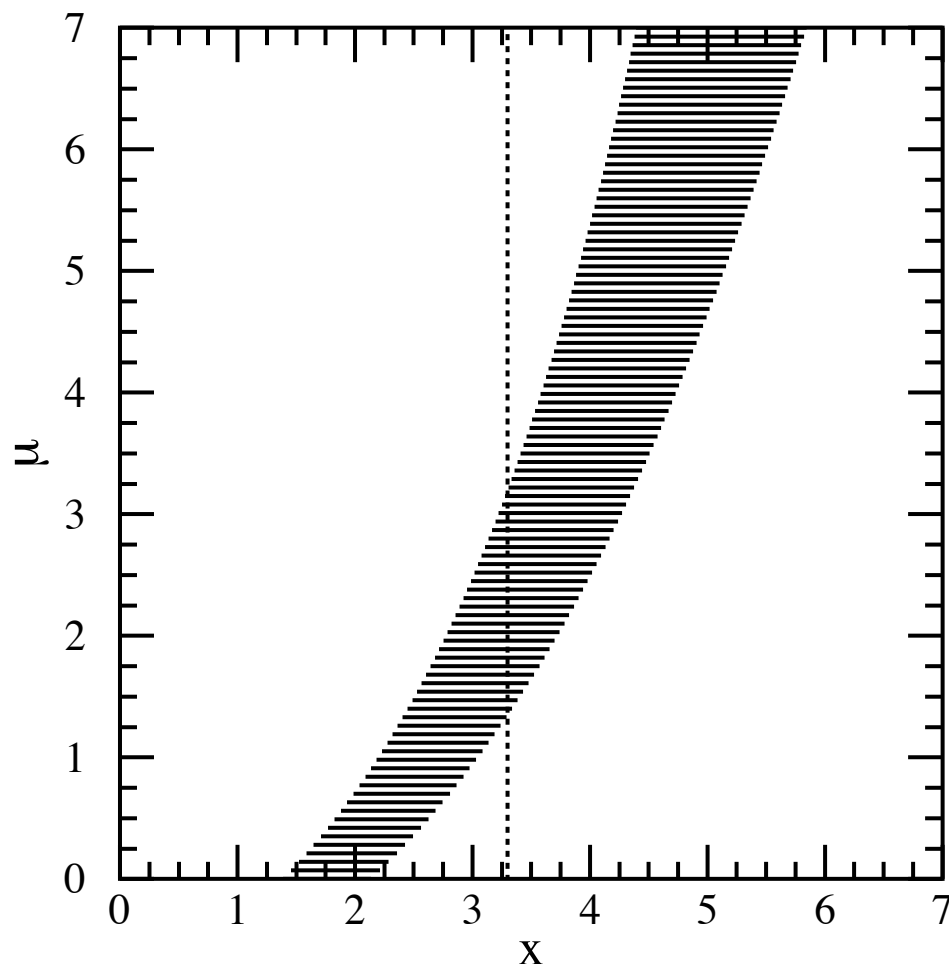
A different way to picture Feldman-Cousins

Most people think of plot on left when thinking of Feldman-Cousins

- bars are regions “ordered by” $R = P(n|\mu)/P(n|\mu_{\text{best}})$, with $\int_{x_1}^{x_2} P(x|\mu)dx = \alpha$.

But this picture doesn't generalize well to many measured quantities.

- Instead, just use R as the test statistic... and R is $\lambda(\mu)$

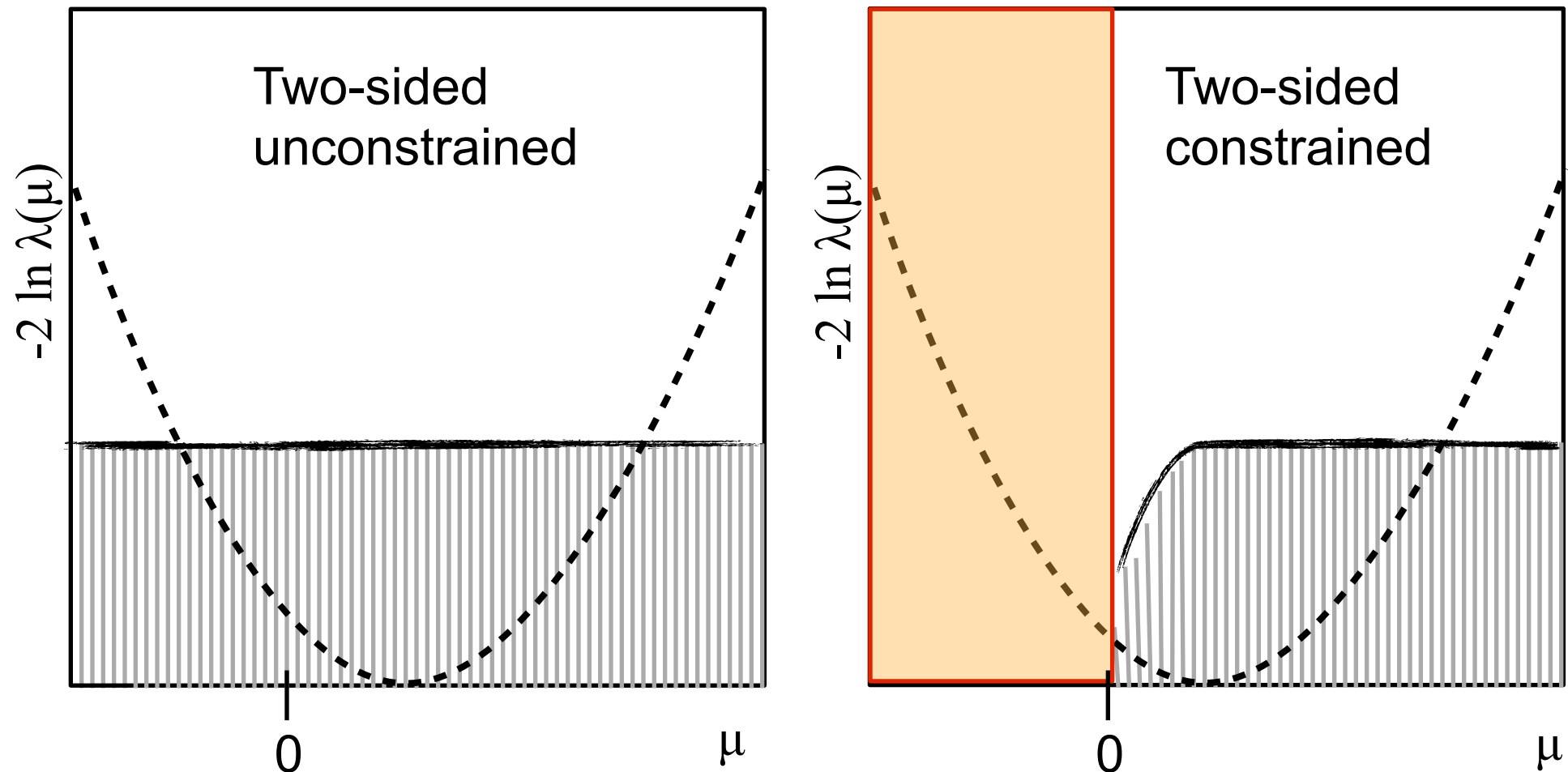




With a physical constraint ($\mu > 0$) the confidence band changes, but conceptually the same. Do not get empty intervals.

$$t_\mu = -2 \ln \lambda(\mu)$$

$$\tilde{t}_\mu = -2 \ln \tilde{\lambda}(\mu) = \begin{cases} -2 \ln \frac{L(\mu, \hat{\hat{\theta}}(\mu))}{L(0, \hat{\hat{\theta}}(0))} & \hat{\mu} < 0, \\ -2 \ln \frac{L(\mu, \hat{\hat{\theta}}(\mu))}{L(\hat{\mu}, \hat{\hat{\theta}})} & \hat{\mu} \geq 0. \end{cases}$$



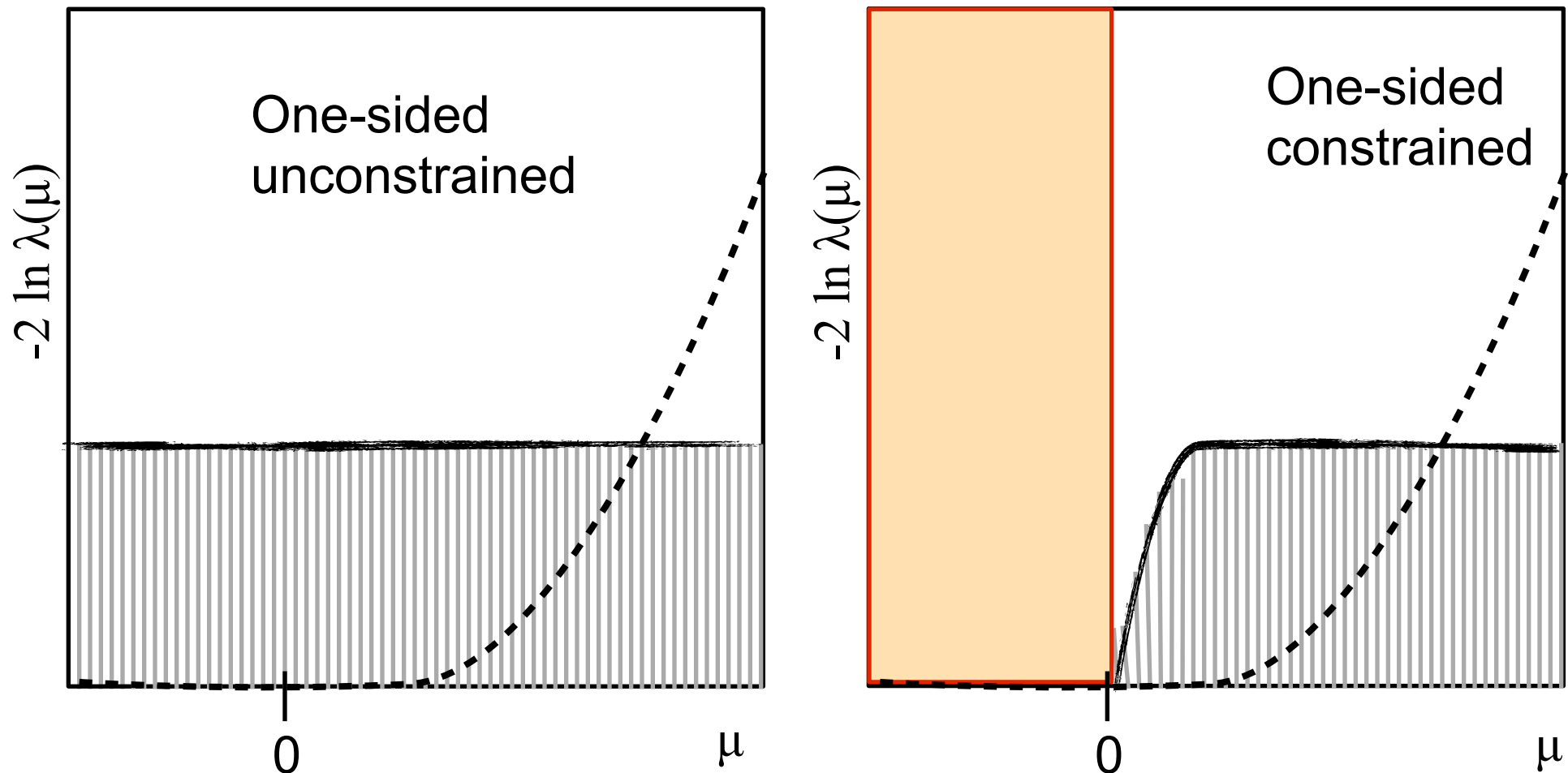
Modified test statistic for 1-sided upper limits



For 1-sided upper-limit one constructs a test that is more powerful for all $\mu > 0$ (but has no power for $\mu = 0$) simply by discarding “upward fluctuations”

$$q_\mu = \begin{cases} -2 \ln \lambda(\mu) & \hat{\mu} \leq \mu, \\ 0 & \hat{\mu} > \mu, \end{cases}$$

$$\tilde{q}_\mu = \begin{cases} -2 \ln \frac{L(\mu, \hat{\theta}(\mu))}{L(0, \hat{\theta}(0))} & \hat{\mu} < 0 \\ -2 \ln \frac{L(\mu, \hat{\theta}(\mu))}{L(\hat{\mu}, \hat{\theta})} & 0 \leq \hat{\mu} \leq \mu \\ 0 & \hat{\mu} > \mu. \end{cases}$$

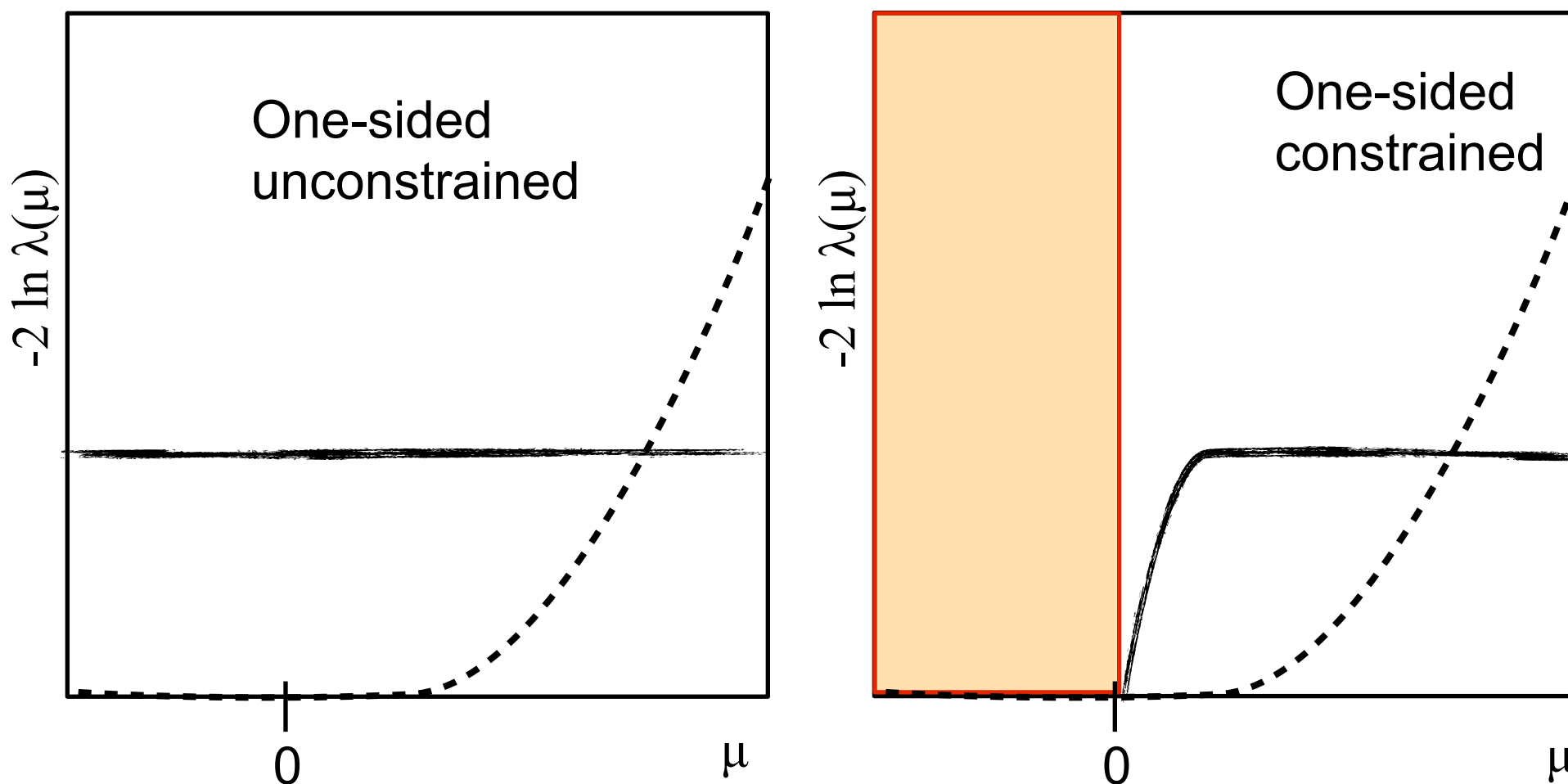




The ProfileLikelihoodCalculator in RooStats currently is for the 2-sided t_μ and it does not take into account effects of boundaries

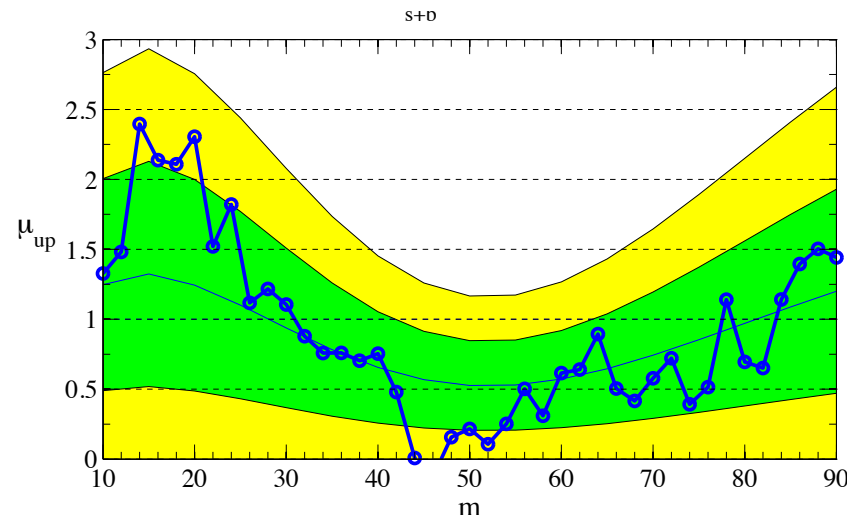
- But, we know [arXiv:1007.1727] the asymptotic distribution for all 4 cases

Will modify tool in future, but for now provided a script to adjust threshold





The asymptotic formulas can also be used to calculate the median and bands on the expected limit (based on Wald's theorem)



This is not yet implemented, but only a day's work.

Current script ProfileBands_qmu.C uses background-only toy experiments and finds upper limit (based on asymptotics) for each background-only toy.

Several channels do not have enough events for asymptotic regime to be valid.

- In this case we use pseudo-experiments

Two main tools:

- HybridCalculator (and invert the test)
 - this approach randomizes nuisance parameters --> Bayesian
- NeymanConstruction and FeldmanCousins tool
 - fully frequentist.
 - conceptually connected to the standard 'asymptotic theory' in statistics.

Both can be slow.



ROOT now has a new executable in \$ROOTSYS/bin called **hist2workspace**

- command line: **hist2workspace myAnalysis.xml**
- Can drive parameter settings, constraints, etc. via XML
- Supports Gaussian, LogNormal, Gamma constraints

$$L(\sigma_{sig}, \mathcal{L}, \alpha_j) = \prod_{l \in \{ee, \mu\mu, e\mu\}} \left\{ \prod_{i \in bins} \left[Pois(N_i^{obs} | N_{i,tot}^{exp}) Gaus(\tilde{\mathcal{L}} | \mathcal{L}, \sigma_{\mathcal{L}}) \prod_{j \in syst} Gaus(0 | \alpha_j, 1) \right] \right\}$$

```
<!DOCTYPE Channel SYSTEM 'Config.dtd'>

<Channel Name="channel1" InputFile="./data/example.root" HistoName="" >
  <!--Data Name="data" InputFile="" HistoPath="" HistoName="" /-->
  <Sample Name="signal" HistoPath="" HistoName="signal">
    <OverallSys Name="syst1" High="1.05" Low="0.95"/>
    <NormFactor Name="SigXsecOverSM" Val="1" Low="0.5" High="1.8" Const="True" />
  </Sample>
  <Sample Name="background1" HistoPath="" NormalizeByTheory="True" HistoName="background1">
    <OverallSys Name="syst2" Low="0.95" High="1.05"/>
  </Sample>
  <Sample Name="background2" HistoPath="" NormalizeByTheory="True" HistoName="background2">
    <OverallSys Name="syst3" Low="0.95" High="1.05"/>
    <!-- <HistoSys Name="syst4" HistoPathHigh="" HistoPathLow="histForSyst4"/-->
  </Sample>
</Channel>
```

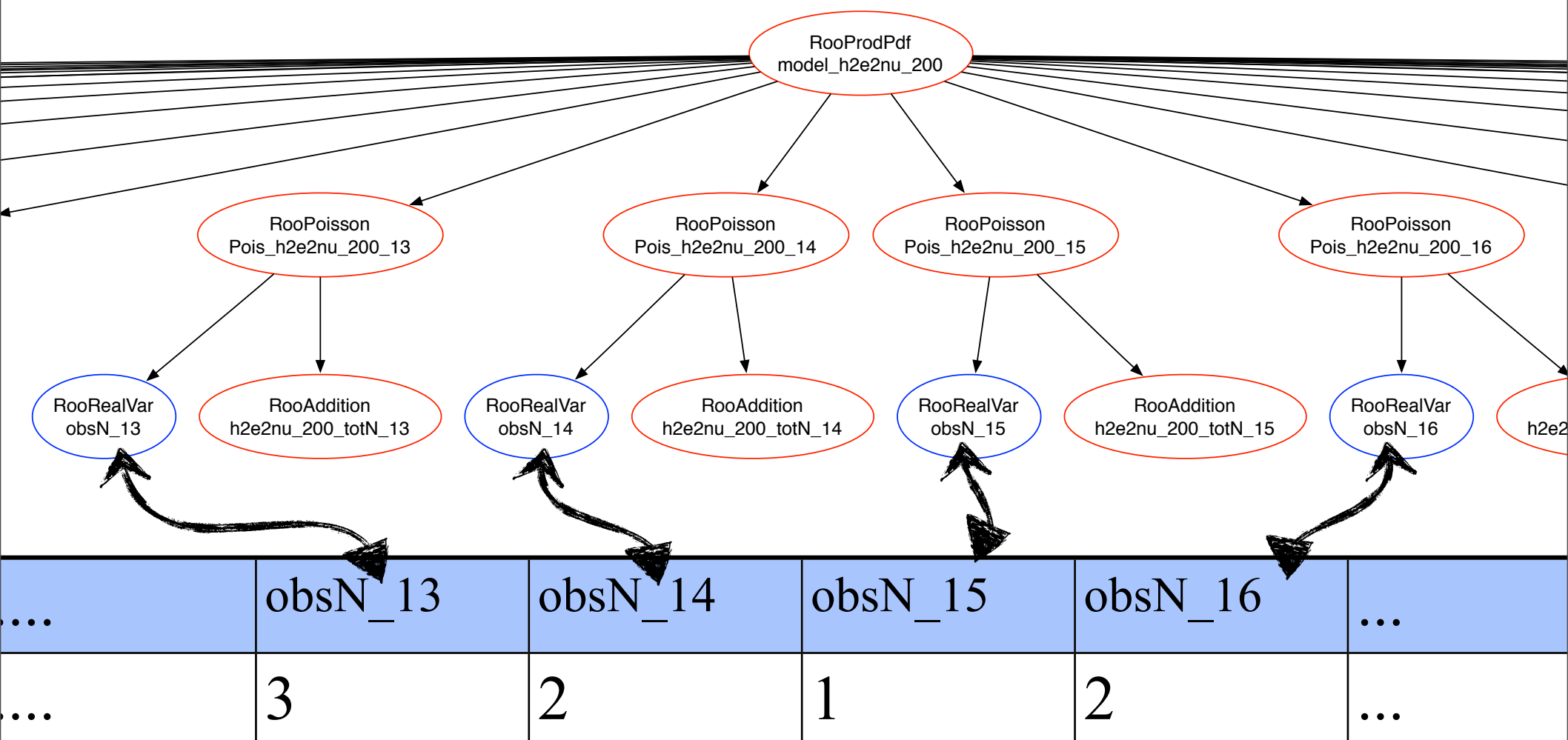
I did some profiling of the code identify why it was being so slow

Before: spending all the time in coupling PDF to dataset (RooNLLVar constructor)... that's bad

```
1958 ProfileLikelihoodTestStatModified::Evaluate(RooAbsData&, RooArgSet&)
1634 RooAbsPdf::createNLL(RooAbsData&, RooCmdArg const&, RooCmdArg const&, RooCmdArg const&, RooCmdArg const&, RooCmdArg c
1634 RooAbsPdf::createNLL(RooAbsData&, RooLinkedList const&)
1495 RooNLLVar::RooNLLVar(char const*, char const*, RooAbsPdf&, RooAbsData&, RooArgSet const&, bool, char const*, char const*, int, bo
1495 RooAbsOptTestStatistic::RooAbsOptTestStatistic(char const*, char const*, RooAbsReal&, RooAbsData&, RooArgSet const&, char cons
792 RooAbsOptTestStatistic::optimizeCaching()
```

For each pseudo-experiment, one must couple the PDF with the dataset

- with this structure, it can require navigating a graph with >4000 nodes
- generation and evaluation were pretty fast, but this coupling was slow
- also related to huge memory consumption (several GB)

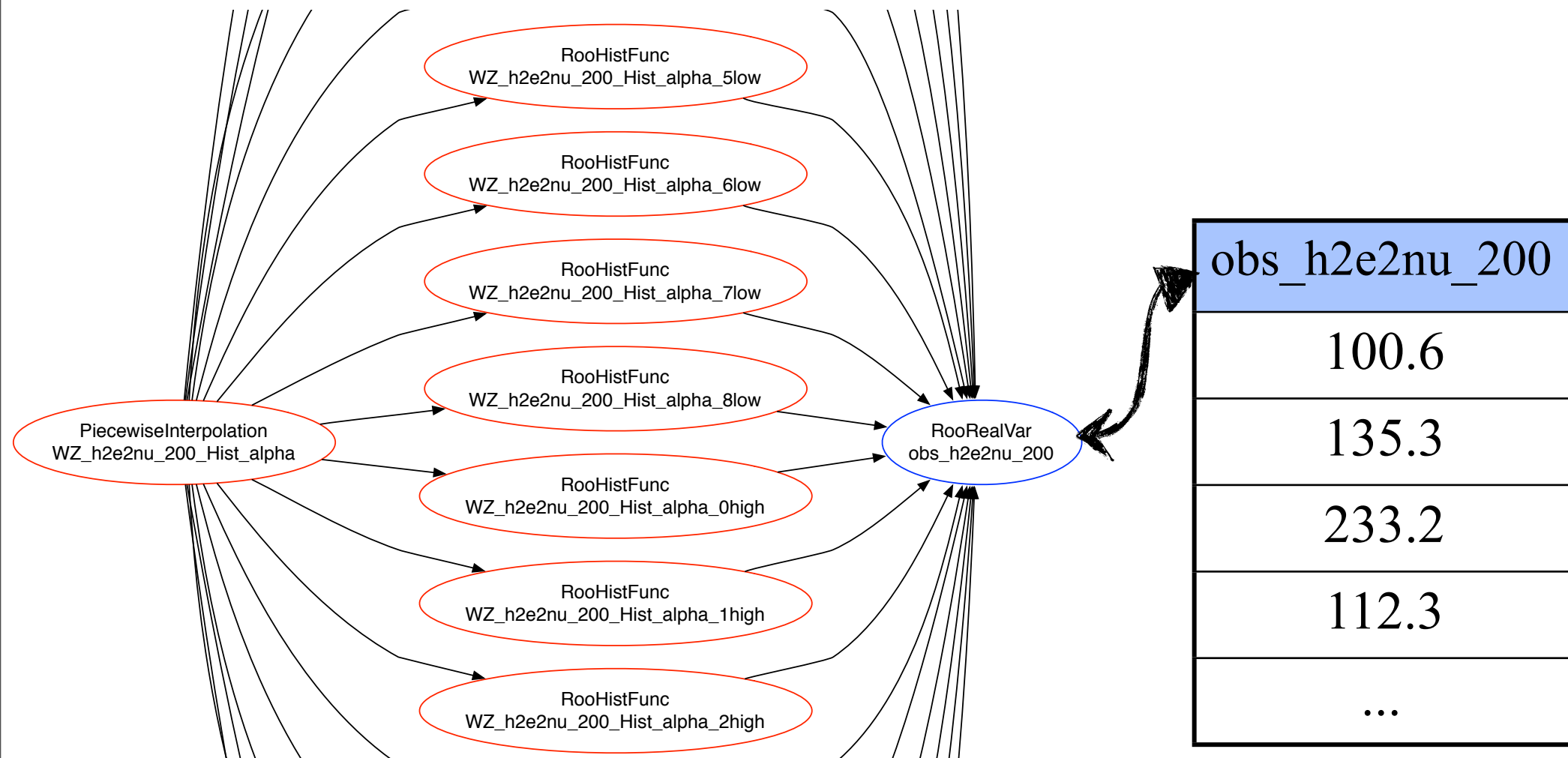




In the new structure, there is just one observable (corresponding to the x-axis of the histograms), so the coupling is fast

- the dataset has many entries corresponding to different histogram bins

The shape is an interpolation between nominal and variational histograms



I did some profiling of the code identify why it was being so slow

Before: spending all the time in coupling PDF to dataset (RooNLLVar constructor)... that's bad

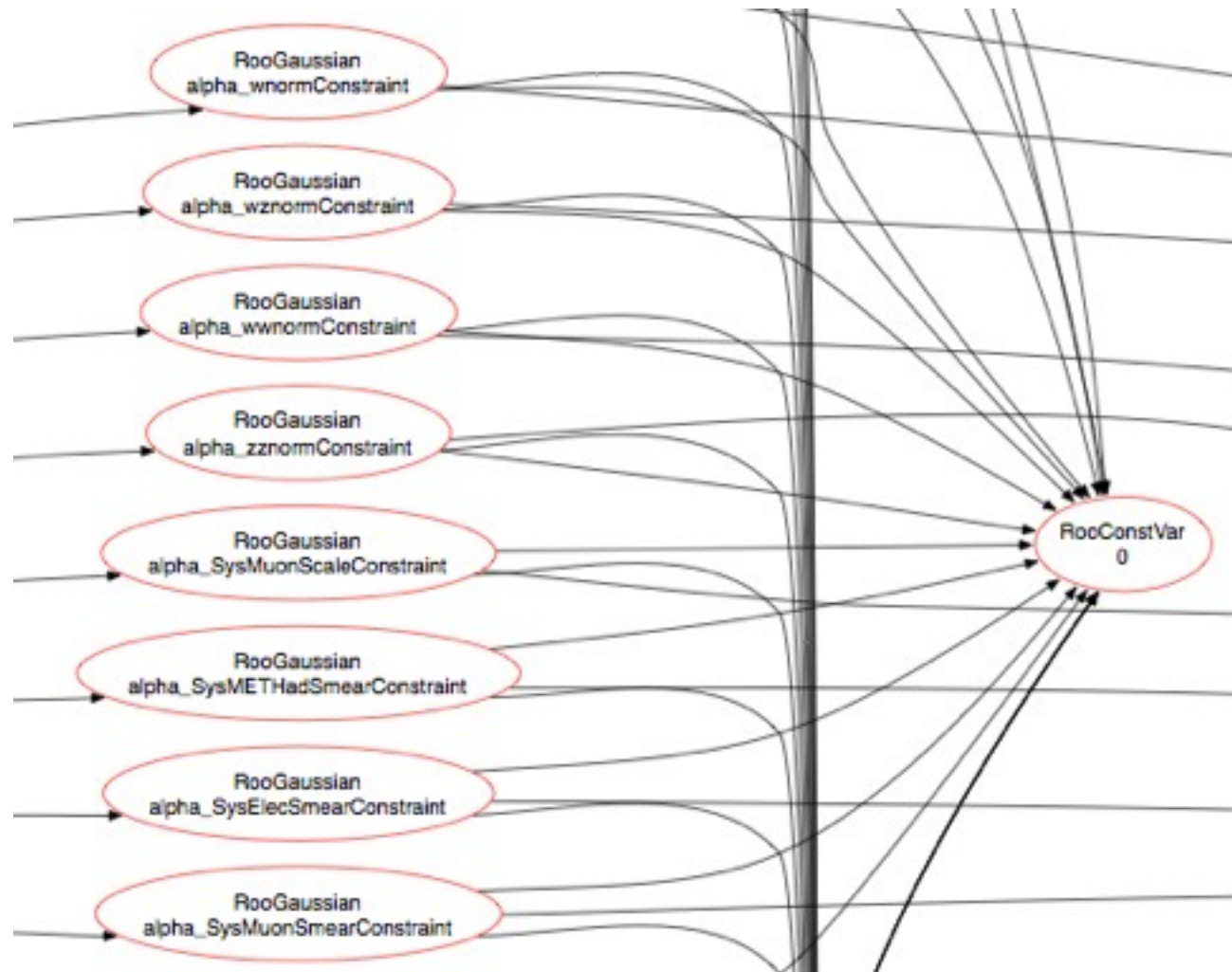
```
1958 ProfileLikelihoodTestStatModified::Evaluate(RooAbsData&, RooArgSet&)
1634 RooAbsPdf::createNLL(RooAbsData&, RooCmdArg const&, RooCmdArg const&, RooCmdArg const&, RooCmdArg const&, RooCmdArg c
1634 RooAbsPdf::createNLL(RooAbsData&, RooLinkedList const&)
1495 RooNLLVar::RooNLLVar(char const*, char const*, RooAbsPdf&, RooAbsData&, RooArgSet const&, bool, char const*, char const*, int, bo
1495 RooAbsOptTestStatistic::RooAbsOptTestStatistic(char const*, char const*, RooAbsReal&, RooAbsData&, RooArgSet const&, char cons
792 RooAbsOptTestStatistic::optimizeCaching()
```

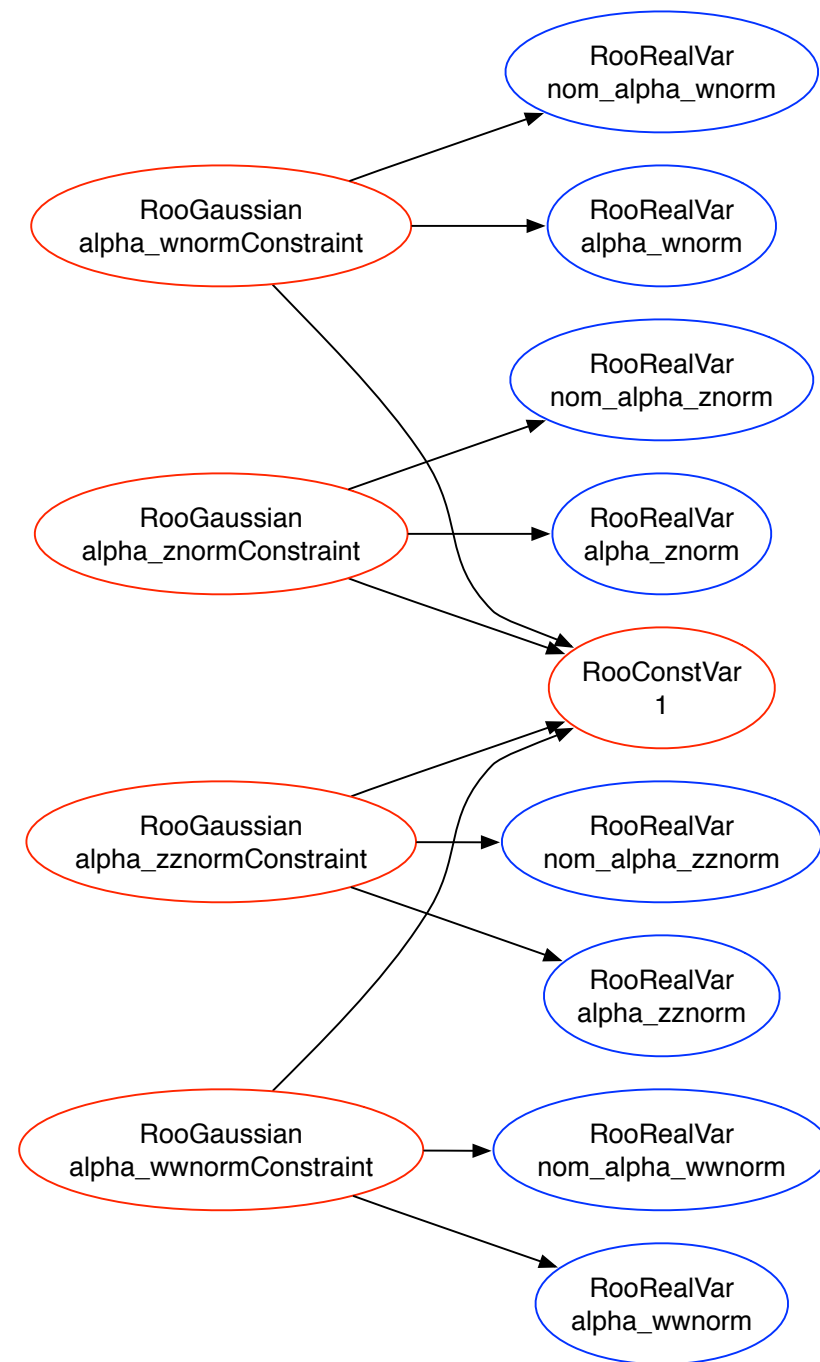
After the changes to the model, overhead is gone...

▸ **x5-15 speed increase for problems with 50 bins and 30 nuisance**

After: spending all the time in minuit... that's good.

```
1821 ProfileLikelihoodTestStatModified::Evaluate(RooAbsData&, RooArgSet&)
1741 RooAbsReal::getVal(RooArgSet const*) const
1741 RooAbsReal::traceEval(RooArgSet const*) const
1741 RooProfileLL::evaluate() const
1444 RooProfileLL::validateAbsMin() const
1444 RooMinuit::migrad()
1444 TFitter::ExecuteCommand(char const*, double*, int)
1444 TMinuit::mnexcm(char const*, double*, int, int&)
1444 TMinuit::mnmigr()
1307 TMinuit::mnderi()
1306 TMinuit::Eval(int, double*, double&, double*, int)
1306 RooMinuitGlue(int&, double*, double&, double*, int)
1303 RooAbsReal::getVal(RooArgSet const*) const
1303 RooAbsReal::traceEval(RooArgSet const*) const
1303 RooAddition::evaluate() const
1303 RooAbsReal::getVal(RooArgSet const*) const
1303 RooAbsReal::traceEval(RooArgSet const*) const
1302 RooAbsTestStatistic::evaluate() const
1302 RooNLLVar::evaluatePartition(int, int, int) const
```

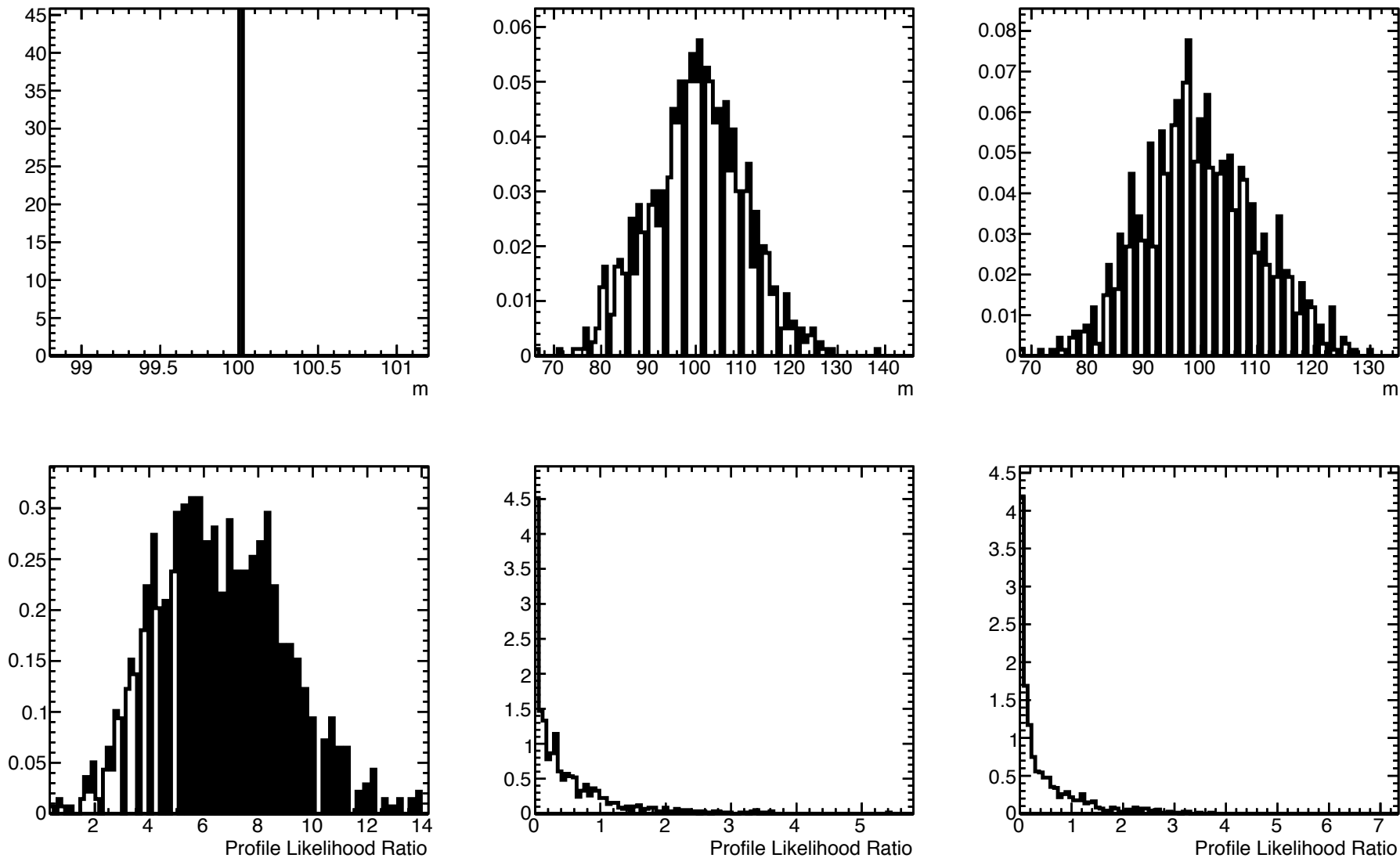




Conditional vs. Unconditional Ensemble



Note, the asymptotic results (eg. χ^2 distributions) correspond to unconditional ensembles.





The HistFactory takes input XML that organizes nominal and variational histograms and produces a workspace with a ModelConfig object that provides all the meta-information information need for the statistical tools

- **Note, the list of “Global Observables” associated to auxiliary measurements that constrain nuisance parameters (global b/c one per experiment, not one per event)**

```
root [7] modelConfig->Print()
```

```
=== Using the following for ModelConfig ===
```

```
Observables:      RooArgSet:: = (obs_h2e2nu_200)
```

```
Parameters of Interest: RooArgSet:: = (SigXsecOverSM)
```

```
Nuisance Parameters:  RooArgSet:: =
```

```
(Lumi,alpha_SysBtagEff,alpha_SysElecScale,alpha_SysElecSmear,alpha_SysJetScale,alpha_SysJetSmear,alpha_SysMETHadScale,alpha_SysMETHadSmear,alpha_SysMuonScale,alpha_SysMuonSmear,alpha_dieleceff,alpha_mjet2enorm,alpha_signorm,alpha_topnorm,alpha_wnorm,alpha_wnorm,alpha_wznorm,alpha_znorm,alpha_zznorm)
```

```
Global Observables:   RooArgSet:: =
```

```
(nominalLumi,nom_alpha_dieleceff,nom_alpha_signorm,nom_SysMuonScale,nom_SysMETHadSmear,nom_SysElecSmear,nom_SysMuonSmear,nom_SysJetSmear,nom_SysBtagEff,nom_SysJetScale,nom_SysMETHadScale,nom_SysElecScale,nom_alpha_topnorm,nom_alpha_wnorm,nom_alpha_wznorm,nom_alpha_zznorm,nom_alpha_wnorm,nom_alpha_znorm,nom_alpha_mjet2enorm)
```

```
PDF:      RooProdPdf::model_h2e2nu_200[ lumiConstraint * alpha_dieleceffConstraint *  
alpha_signormConstraint * alpha_SysMuonScaleConstraint * alpha_SysMETHadSmearConstraint *  
alpha_SysElecSmearConstraint * alpha_SysMuonSmearConstraint * alpha_SysJetSmearConstraint *  
alpha_SysBtagEffConstraint * alpha_SysJetScaleConstraint * alpha_SysMETHadScaleConstraint *  
alpha_SysElecScaleConstraint * alpha_topnormConstraint * alpha_wnormConstraint * alpha_wznormConstraint *  
alpha_zznormConstraint * alpha_wnormConstraint * alpha_znormConstraint * alpha_mjet2enormConstraint *  
h2e2nu_200_model ] = 0
```

Conditional vs. Unconditional

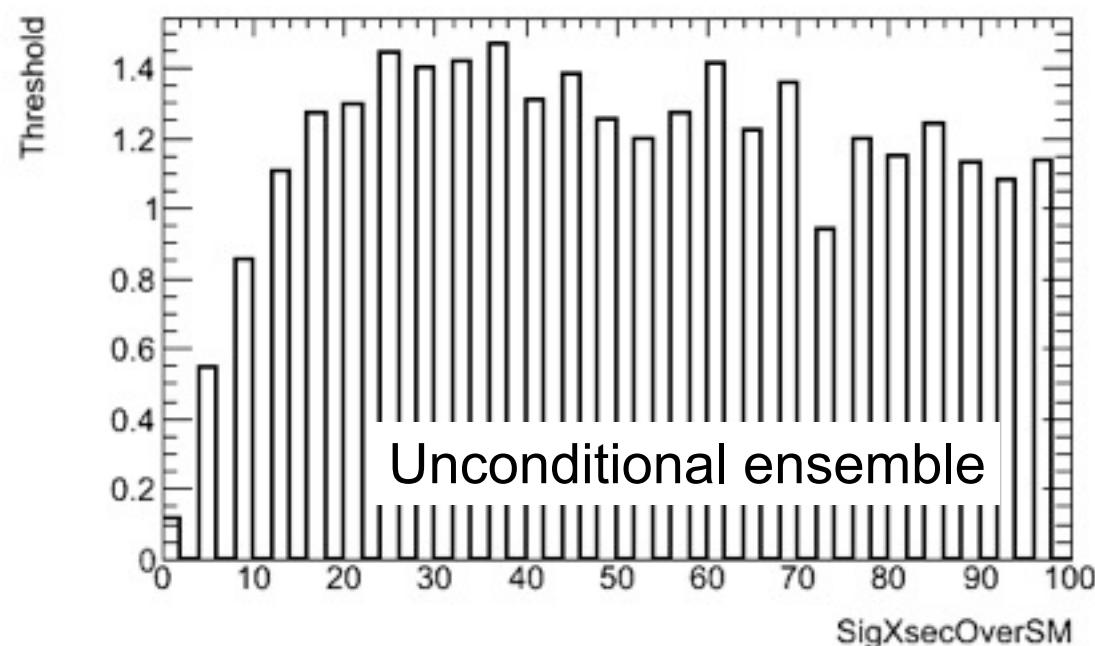
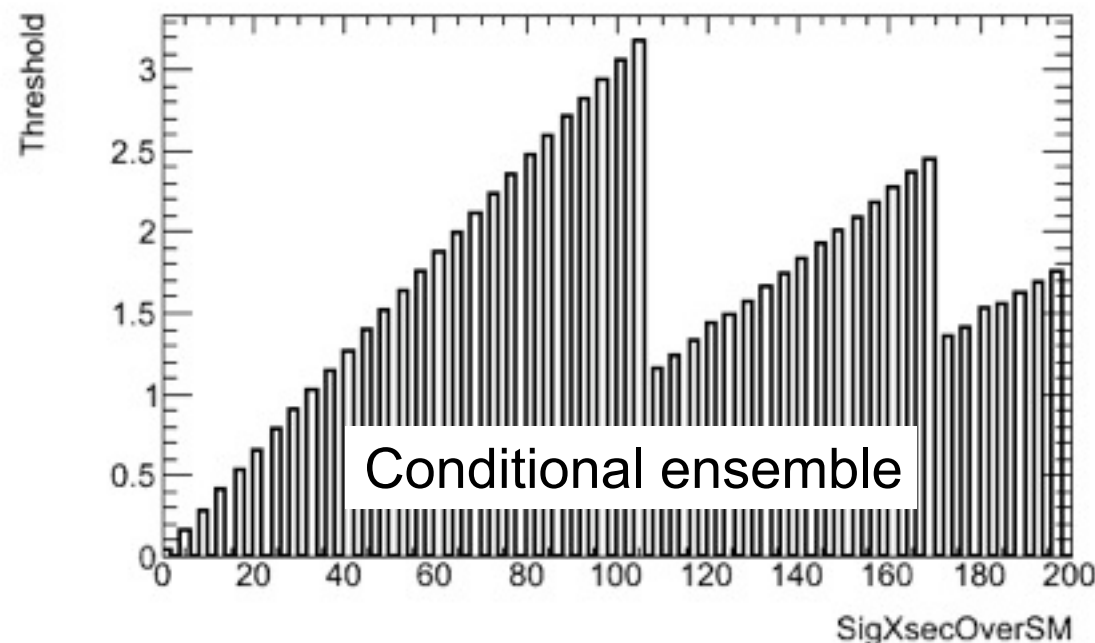


This is for an analysis with several bins but $b \ll 1$.

- so most toy experiments have $N=0$

In the Conditional ensemble the global observables / auxiliary measurements are always the same, so the test statistic has a specific discrete value

In the Unconditional ensemble the global observables / auxiliary measurements fluctuate “smearing out” the value of the test statistic.



Script constructs confidence belt

- uses b-only toys to get bands and CLb
- `OneSidedFrequentistUpperLimitWithBands.C`
- Being used in ATLAS with real-life complicated models

