STATISTICAL METHODS FOR PARTICLE AND NUCLEAR PHYSICS

A short tour

Fabrizio Napolitano, INFN-LNF fabrizio.napolitano@lnf.infn.it

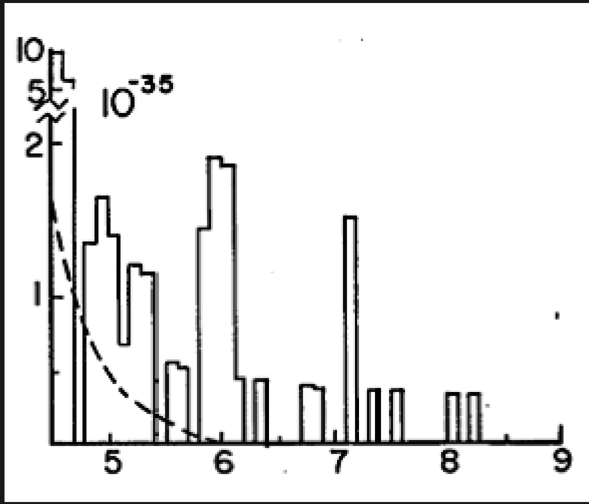
EXOTICO: EXOTIc atoms meet nuclear COllisions for a new frontier precision era in low-energy strangeness nuclear physics

20th October 2022, ECT* Trento, Italy

Istituto Nazionale di Fisica Nucleare
LABORATORI NAZIONALI DI FRASCATI

WHY STATISTICS IS IMPORTANT

do you see the particle at 6 GeV?

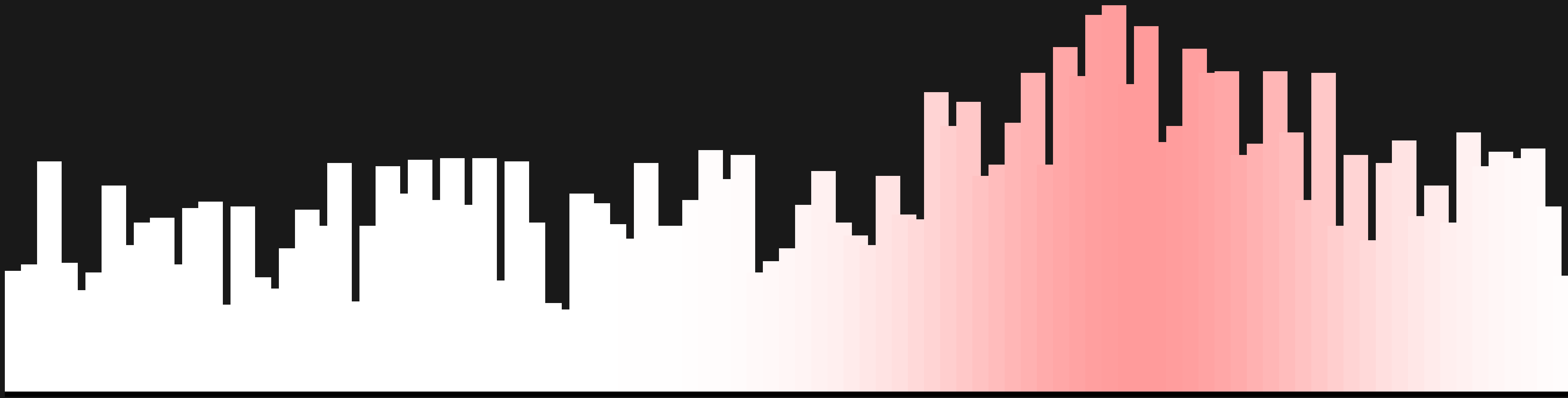


Oops-Leon Particle (Leon Lederman, Υ)

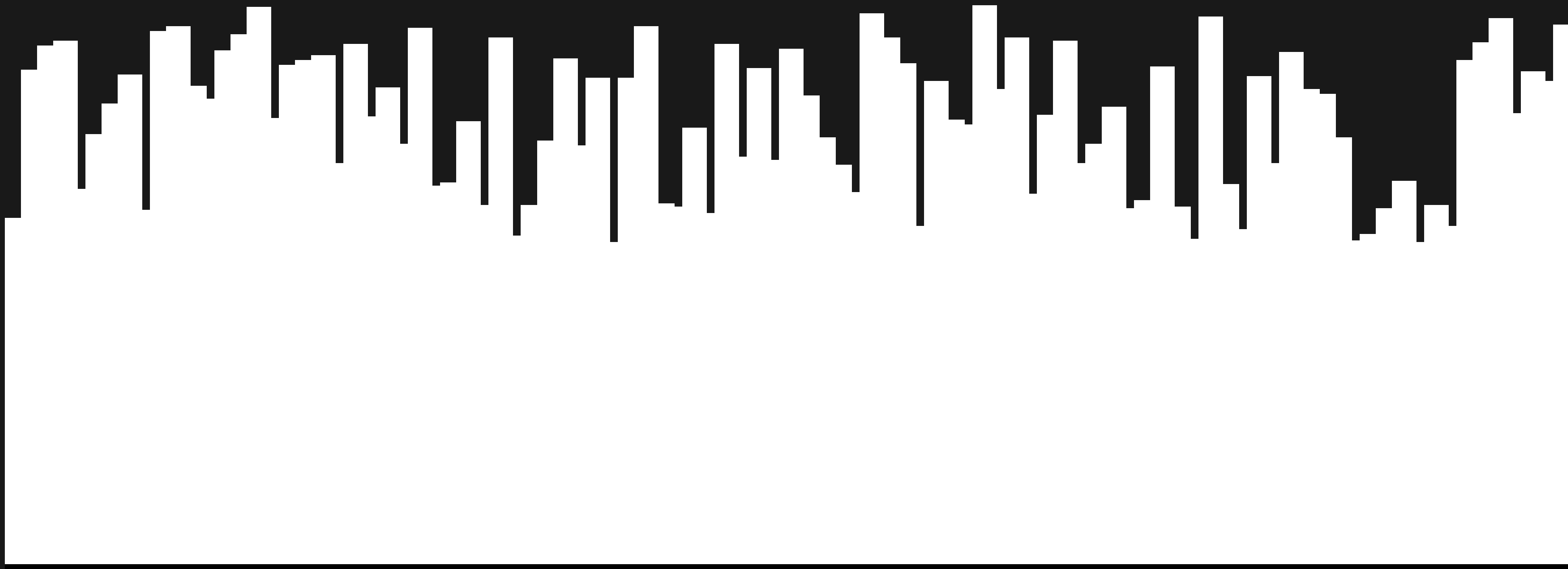
https://doi.org/10.1103/PhysRevLett.36.1236

TAKING DATA..

..WILL THE EXCESS STAY..



..OR DISAPPEAR WITH MORE DATA?



OUTLINE OF THE TALK

1. Likelihoods
2. Discoveries
3. Exclusions
4. Confidence Intervals
5. Bayesian statistics

Disclaimer! based on talks from N. Berger, W. Verkerke, G. Cowan

HYPOTHESIS TESTING

Null Hypothesis H_0: only background is present

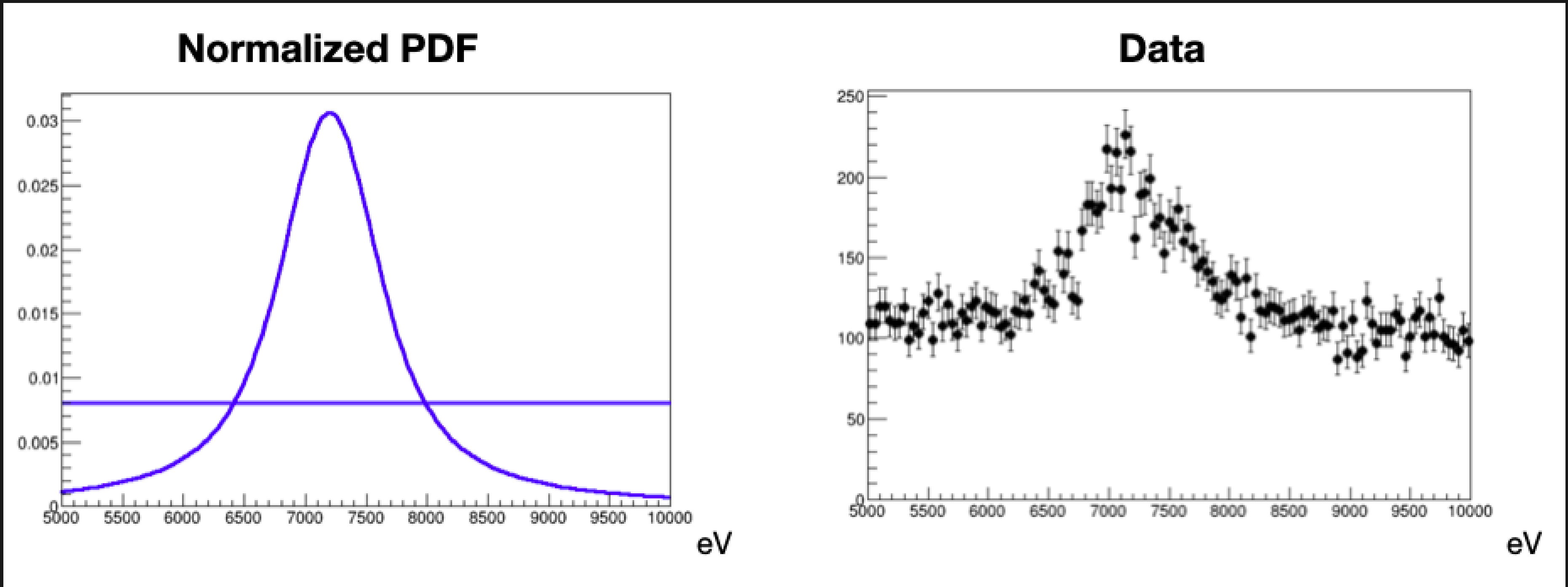
p-value or significance: how strong is the rejection

	Data Disfavors H_0	Data Favors H_0
H_0 is false	Discovery	Missed Discovery
H_0 is true	False Claim	No new physics, nothing found

We want to avoid False Claims and Missed Discoveries

LIKELIHOOD

Probability Density Function (PDF) for the data $p(\text{data}|\mu)$ encodes the entire process



We want to learn something about μ from the data:
Likelihood $\mathcal{L}(\mu) = P(\text{data}|\mu)$

LIKELIHOOD

Want e.g. to estimate the parameter (centroid etc.)
use the Maximum-Likelihood Estimator (MLE)
$\hat{\mu}$ is the parameter value which maximizes $\mathcal{L}(\mu)$

Want to test hypothesis: Neyman-Pearson Lemma
Null hypothesis H_0, alternate hypothesis H_1
The most potent discriminator is the likelihood ratio

$$\frac{\mathcal{L}(\mu = \mu_1)}{\mathcal{L}(\mu = \mu_0)}$$

Minimizes missed discoveries, need alternate hypothesis

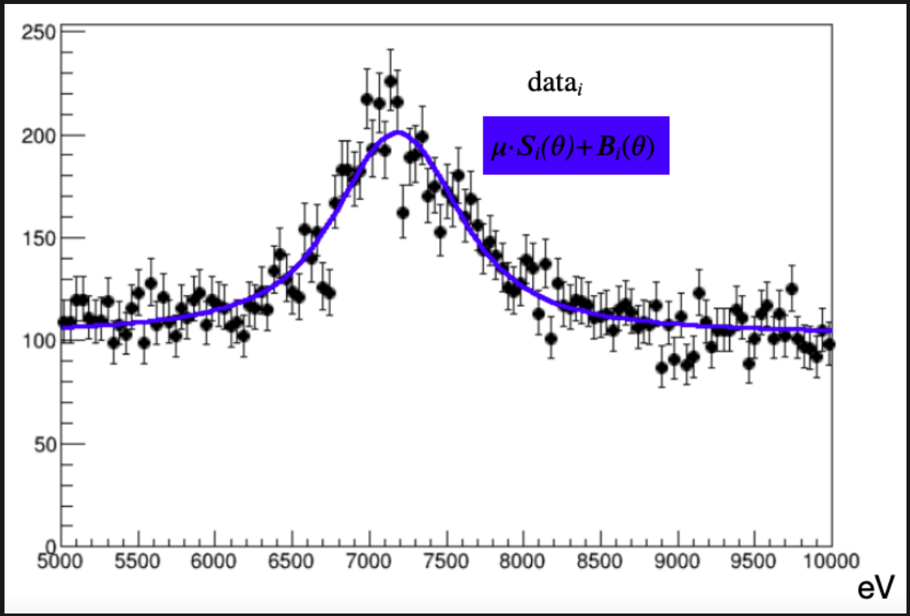
HOW TO BUILD A (BINNED) LIKELIHOOD

Likelihood is the core description of the statistical processes

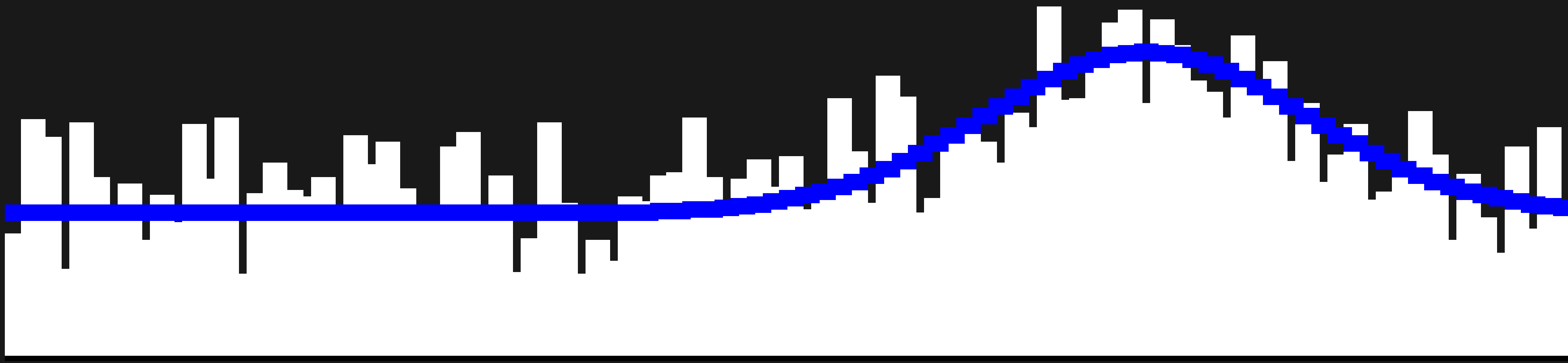
$$\mathcal{L}(\text{data}|\mu, \theta) = \prod_{i \in bins} \mathcal{P}(\text{data}_i|\mu \cdot S_i(\theta) + B_i(\theta))$$

$\mu \cdot S_i(\theta) + B_i(\theta)$ is the prediction of the events in the bin i

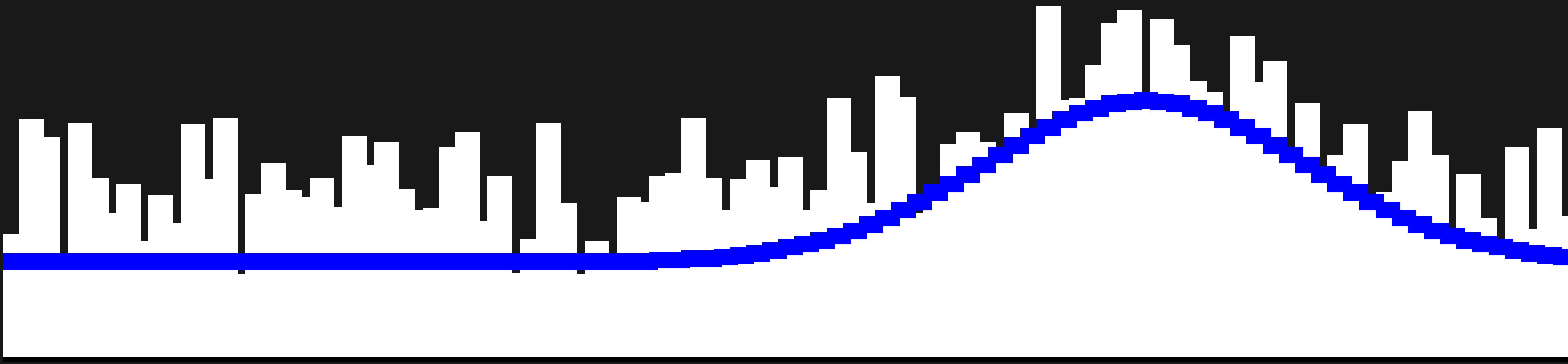
data_i is the number of observed events in the bin i



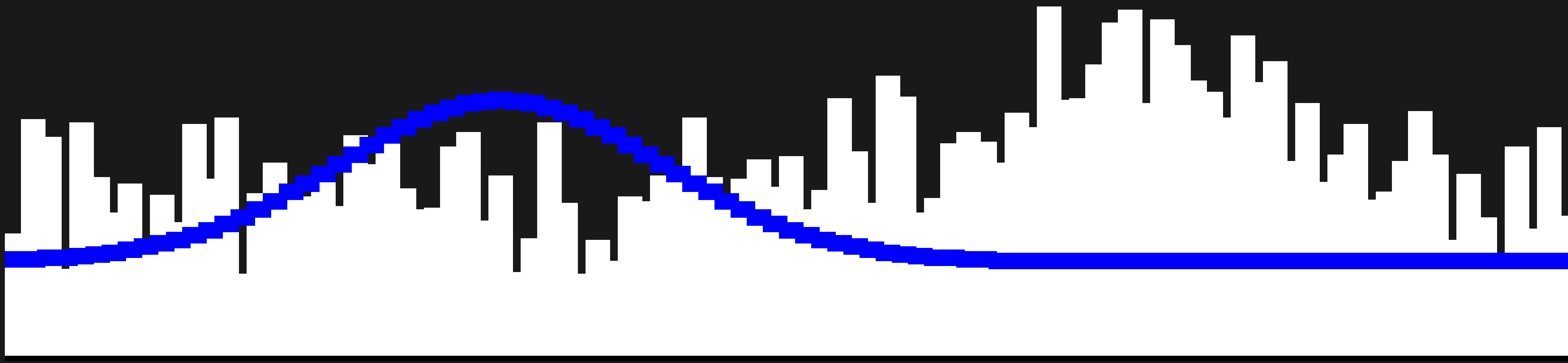
LIKELIHOOD EXAMPLE



LIKELIHOOD EXAMPLE



LIKELIHOOD EXAMPLE



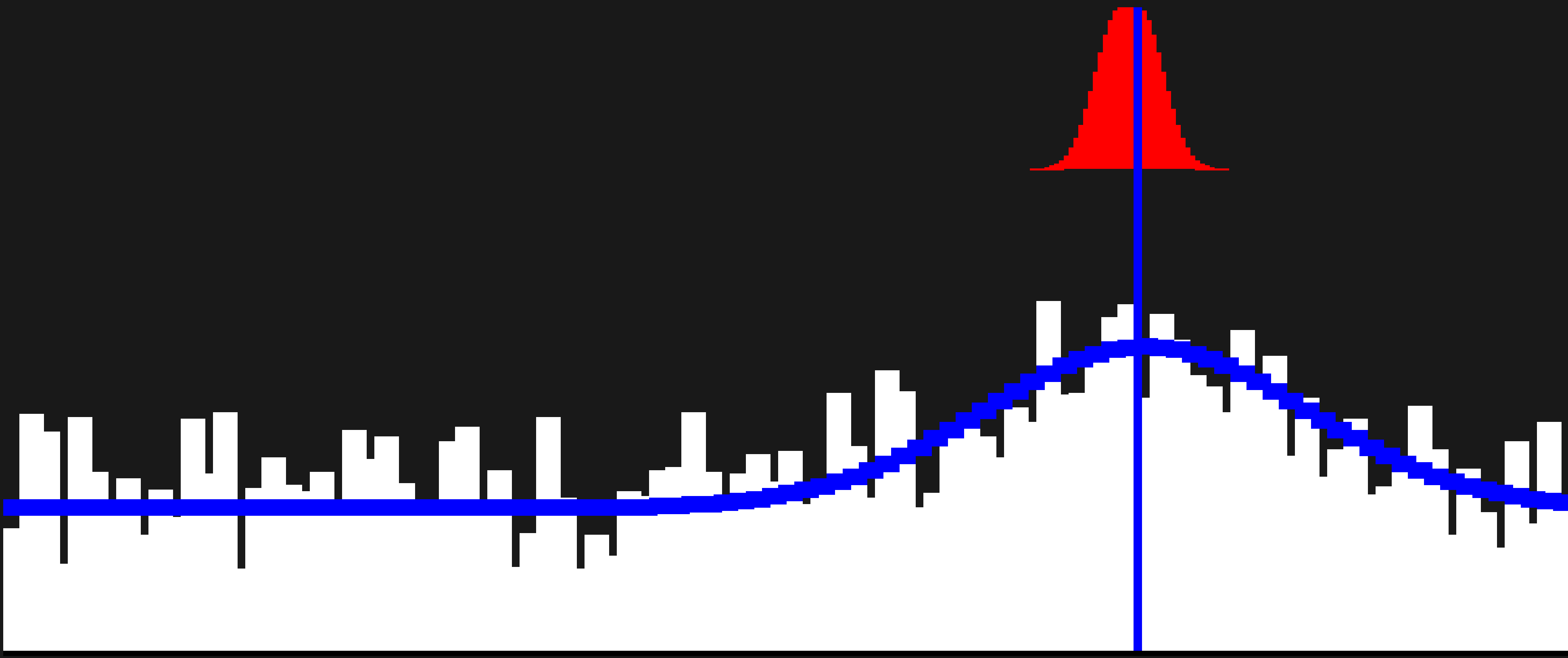
NUISANCE AND SYSTEMATICS

Parameter of interest (POI), e.g. signal yield, peak position etc
Nuisance Parameters (NP): all other parameters, representing properties of the data
If they represent systematic (e.g. energy scale) they need additional terms in the likelihood

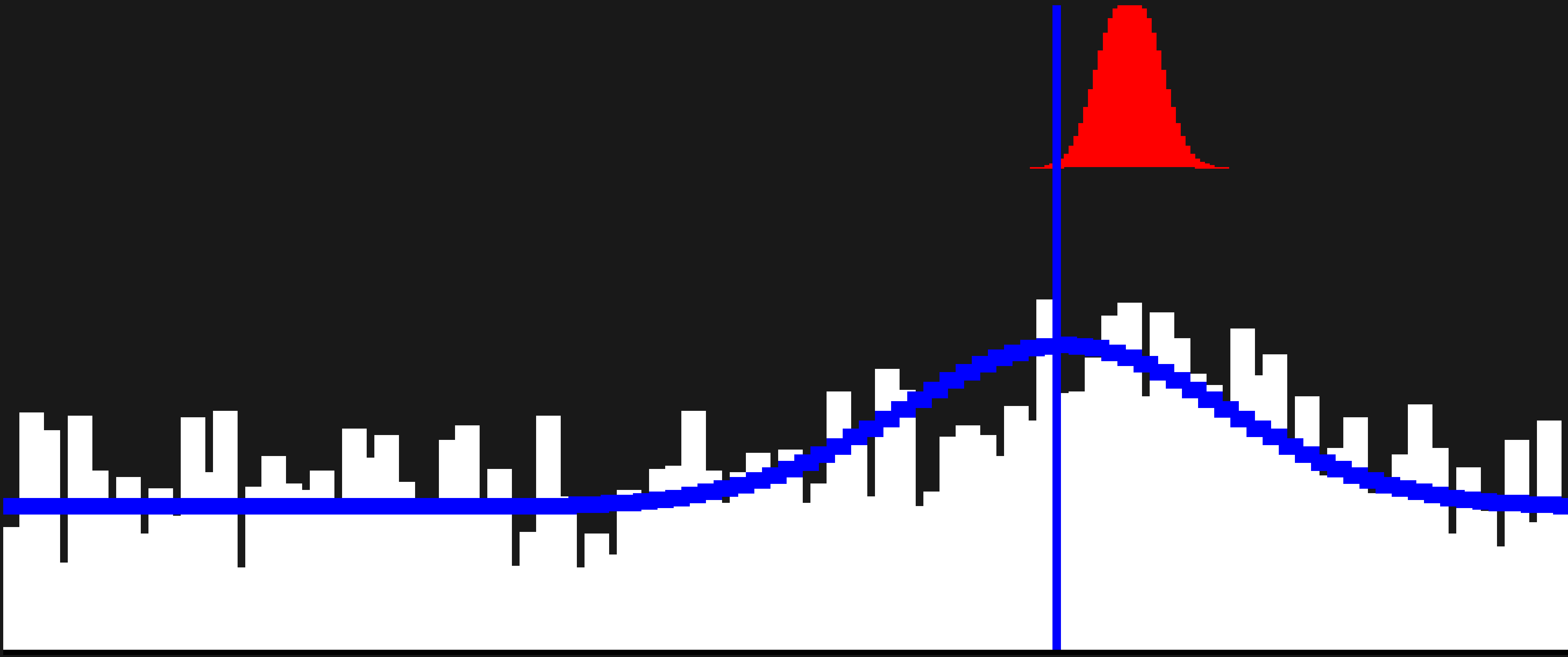
$$\mathcal{L}(\mu, \theta) = \mathcal{L}(\text{data}|\mu, \theta) \times C(\text{aux. data}|\theta)$$

Constraint C is implemented directly in the combined likelihood
Takes care of experimental uncertainties
Gaussian often used $C(\text{aux. data}|\theta) = G(\theta^{obs}, \sigma_{syst}|\theta))$

NUISANCE EXAMPLE



NUISANCE EXAMPLE



CATEGORIES

Can divide the analysis in multiple regions

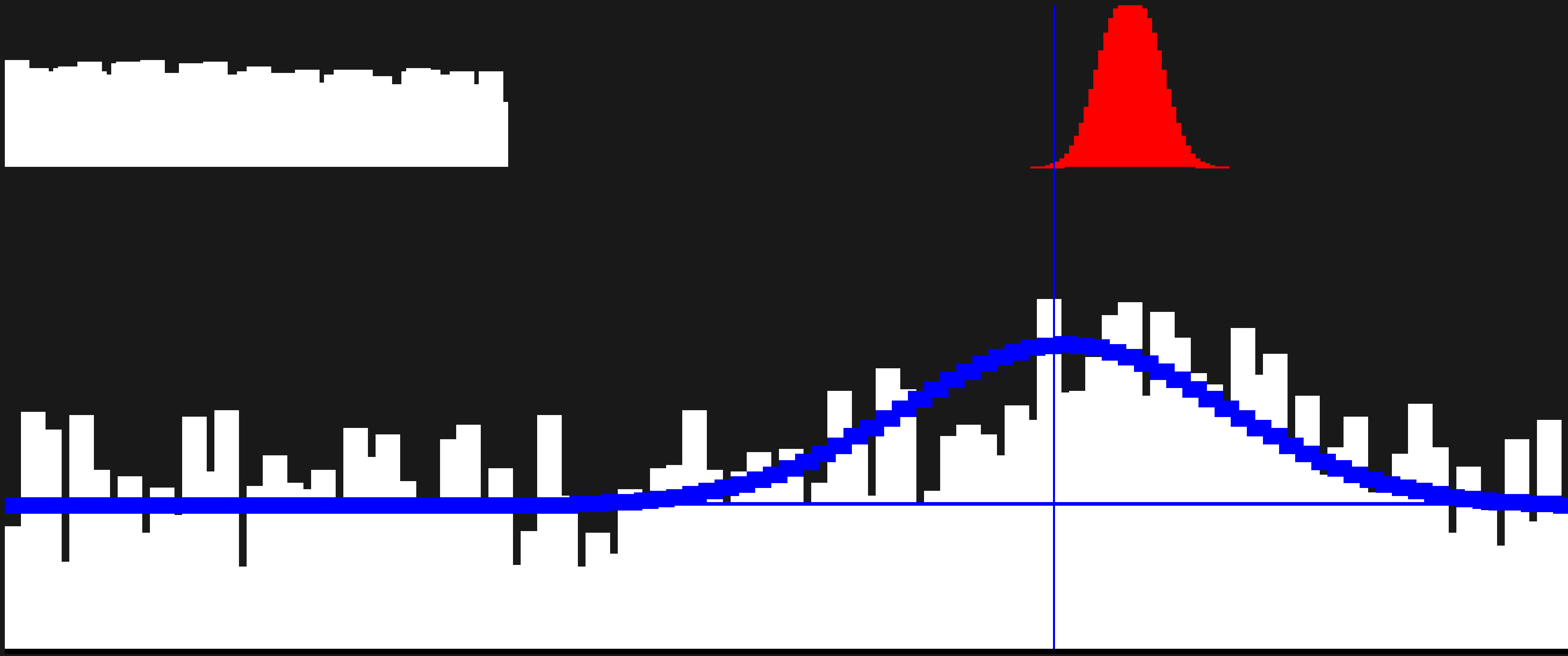
Usually Signal Regions (SR) and Control Regions (CR)

Defined in a way to have control over the NPs, and have better sensitivity

$$\mathcal{L}(\mathrm{data}|\mu, \theta) = \prod_{i \in \mathrm{cat}} \mathcal{L}_i(\mathrm{data}_i|\mu, \theta)$$

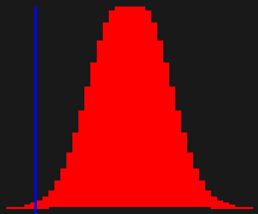
CATEGORIES EXAMPLE

Control Region



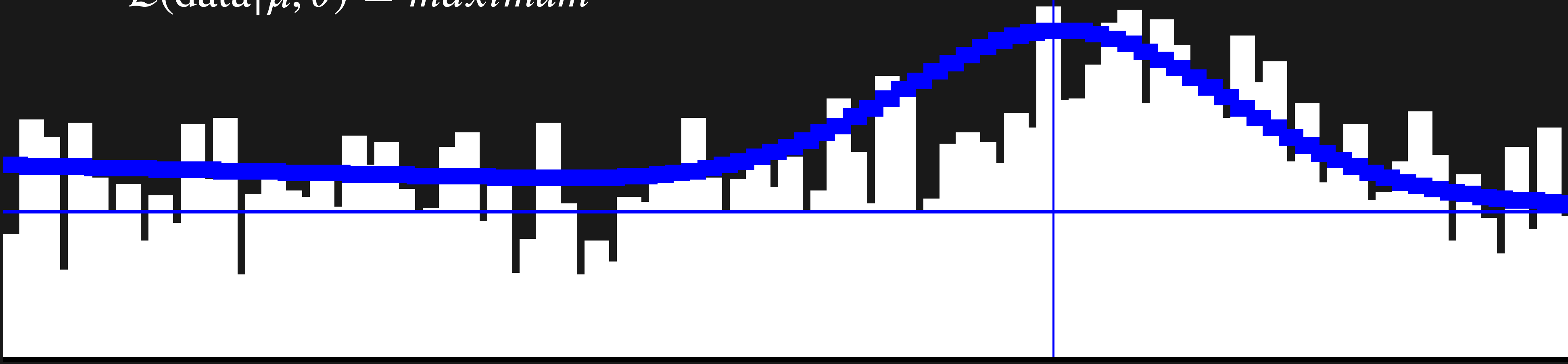
CATEGORIES EXAMPLE

Control Region



Multidimensional problem; $\hat{\mu}, \hat{\theta}$ such that

$$\mathcal{L}(\text{data}|\hat{\mu}, \hat{\theta}) = maximum$$



TEST STATISTICS

We want to construct a way to discriminate against the null hypothesis H_0 (background-only $\mu = 0$) and the signal hypothesis H_1 $(\mu = 1)$
Reminder: the Neyman-Pearson Lemma is the tool to use.

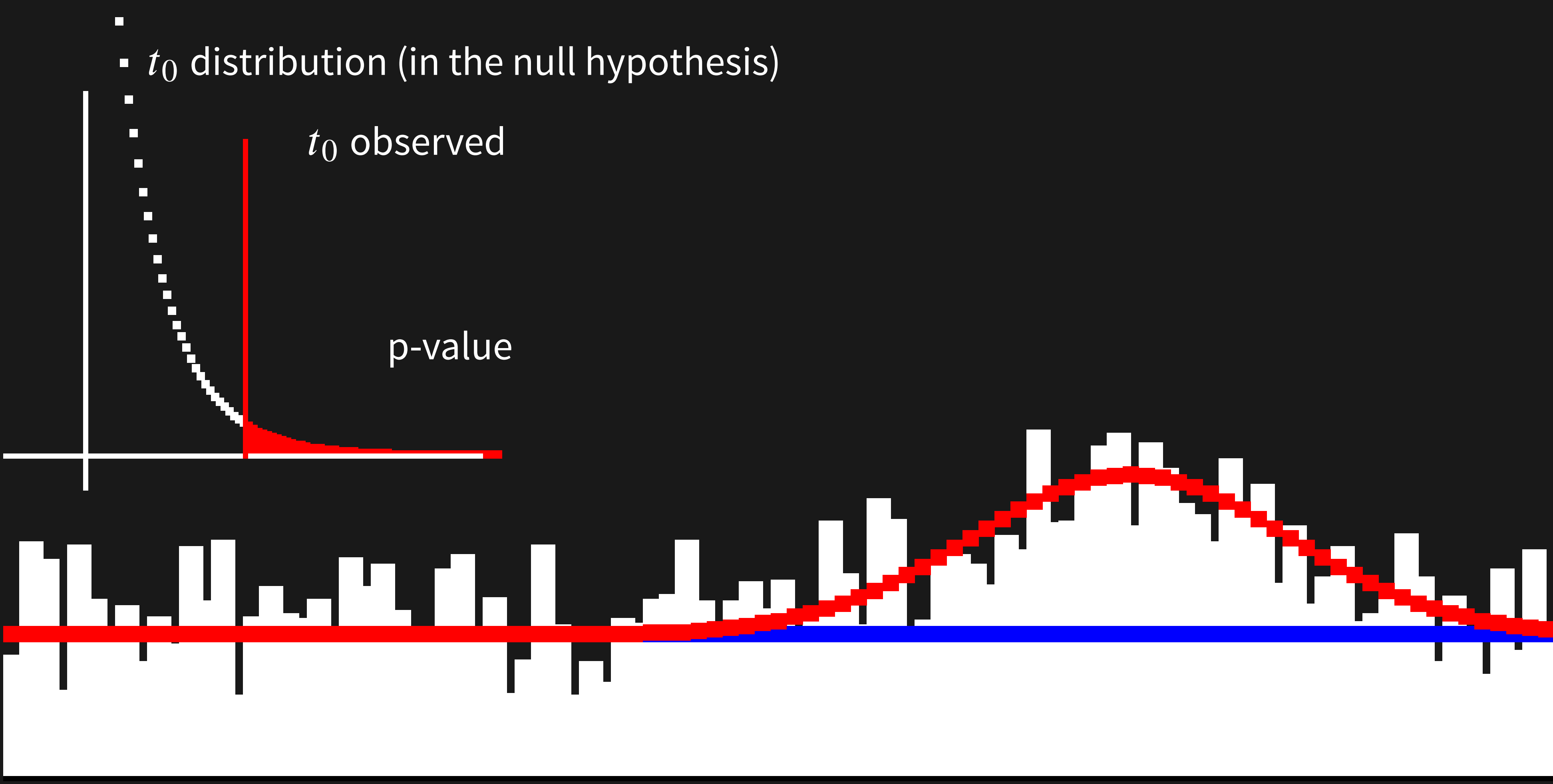
Use as test-statistics

$$t_0 = -2\log\frac{\mathcal{L}(\mu = 0)}{\mathcal{L}(\hat{\mu})} = -2\log\lambda(\hat{\mu})$$

How is t_0 distributed?
Estimate with toy MC or asymptotically

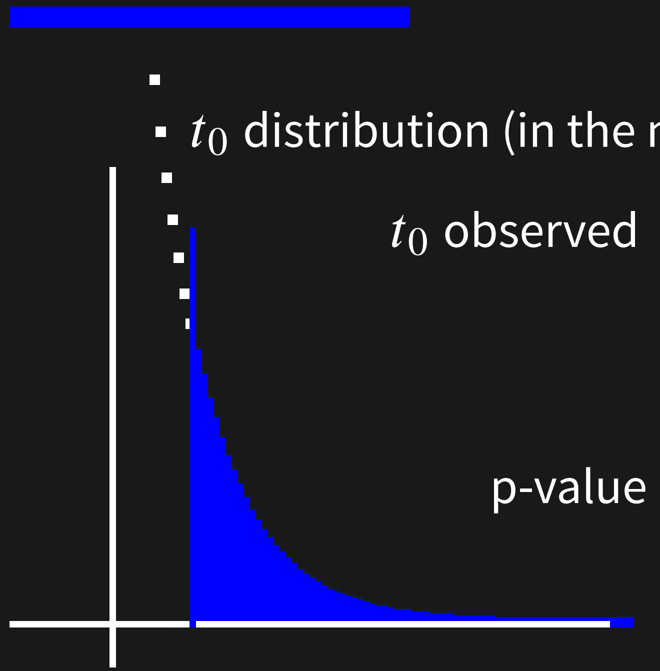
TEST STATISTICS EXAMPLE

$\mathcal{L}(\mu = 0)$ small, $\mathcal{L}(\hat{\mu})$ large; $-2\ln(\frac{\mathcal{L}(\mu=0)}{\mathcal{L}(\hat{\mu})})$ large

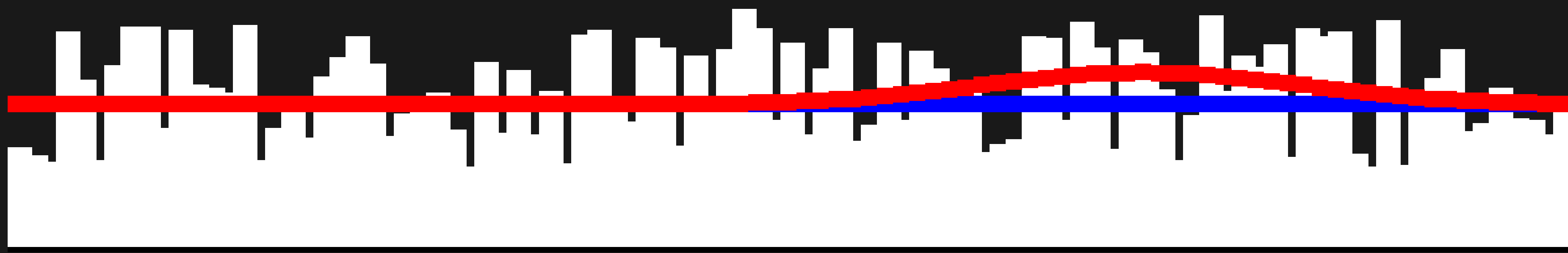


TEST STATISTICS EXAMPLE

$\mathcal{L}(\mu = 0)$ similar to $\mathcal{L}(\hat{\mu})$; $-2\ln(\frac{\mathcal{L}(\mu=0)}{\mathcal{L}(\hat{\mu})})$ to zero



t_0 distributed like a χ^2 (if μ Gaussian)
approximation not valid for few events (<20)
Need to use MC toys in that case



PROFILE LIKELIHOOD RATIO (PLR)

Likelihood with NP (systematic uncertainties and data parameters)
Which NP values to use when testing hypothesis (H_0, H_1)?
-> Use best fit values

Profile Likelihood Ratio (PLR)

$$t_{\mu_0} = -2 \log \frac{\mathcal{L}(\mu = \mu_0, \hat{\hat{\theta}}_{\mu_0})}{\mathcal{L}(\hat{\mu}, \hat{\theta})}$$

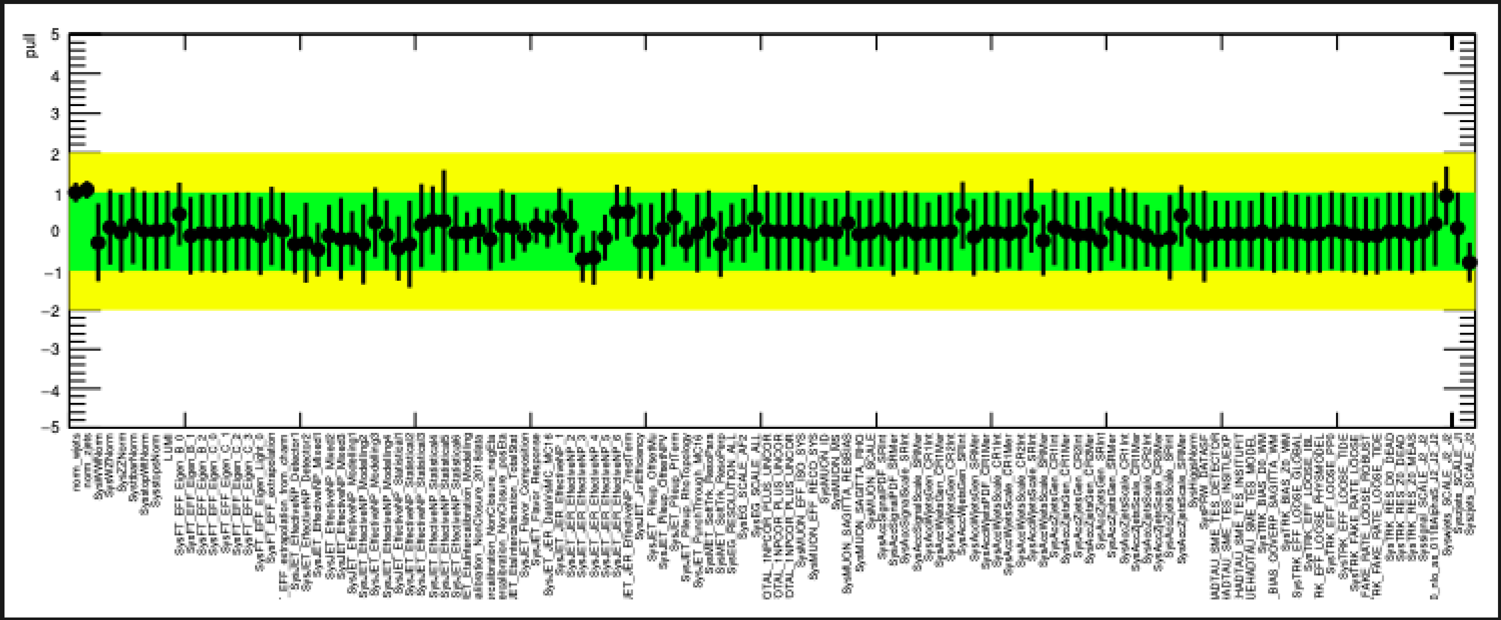
$\hat{\hat{\theta}}_{\mu_0}$, values that maximize $\mathcal{L}$ for fixed μ_0 (conditional),
$\hat{\theta}$, values that maximize the $\mathcal{L}$ overall (unconditional).

Wilks' theorem: PLR follows a χ^2 distribution as well.

NP PULL

Need to see how the NPs behave after the fit
Should be:
- within their central value: $\hat{\theta} \simeq \theta_{obs}$
- with their uncertainty: $\sigma_{\hat{\theta}} \simeq \sigma_{\theta_{sys}}$

if not, need to investigate



TEST STATISTICS FOR DISCOVERY

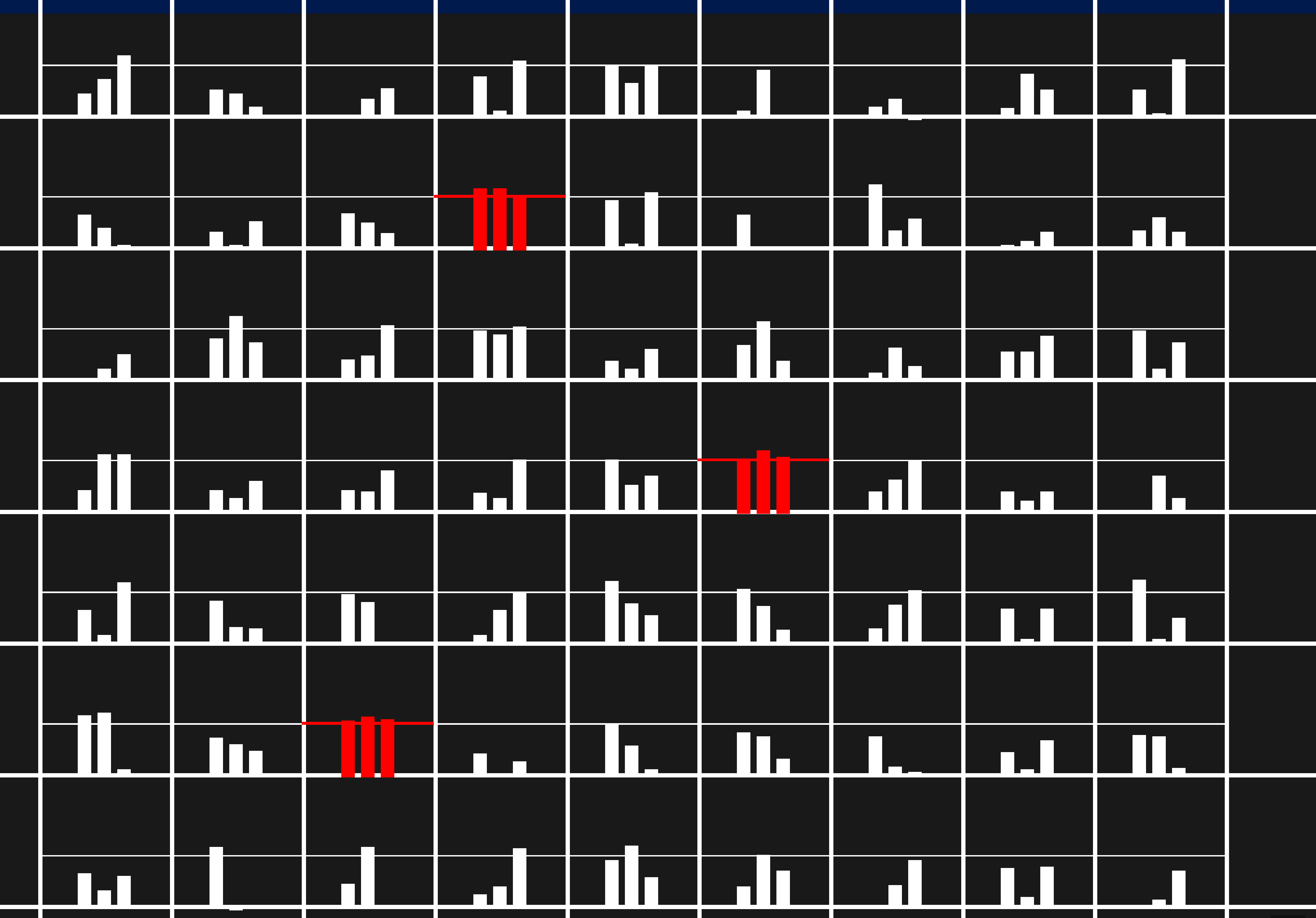
Discovery: when null hypothesis H_0 is rejected

The test must be one-sided: for $\hat{\mu} < 0$, perfect agreement with the null hypothesis

$$q_0 = \begin{cases} -2\log \frac{\mathcal{L}(\mu=\mu_0,\hat{\hat{\theta}}_{\mu_0})}{\mathcal{L}(\hat{\mu},\hat{\theta})} & \text{if } \hat{\mu} \geq 0 \\ 0 & \text{if } \hat{\mu} < 0 \end{cases}$$

Significance $Z = \Phi^{-1}(1 - p_0)$,

Φ: Gaussian CDF



LOOK ELSEWHERE EFFECT

when performing scan over unknown parameter (mass, shift etc)
when having multiple signal regions
it is more likely to find an excess simply by chance
take this into account with the Look Elsewhere Effect (LEE)
Two way to deal with that
- Brute force: MC toys.
- Asymptotic approximation of the trial factors

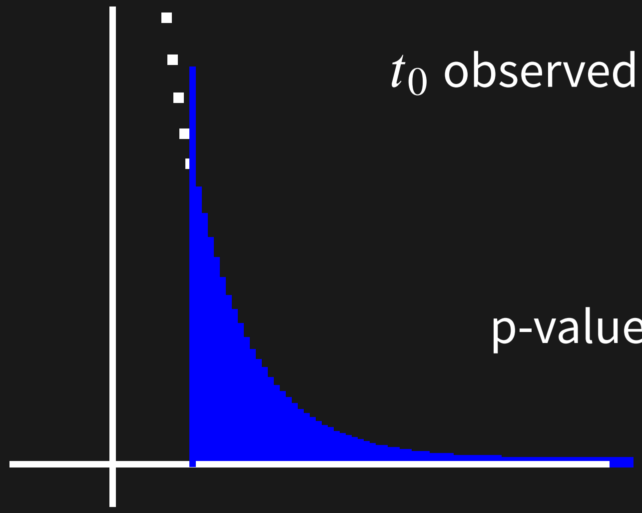
$$p_{global} = p_{local} \cdot \left(1 + \frac{1}{p_{local}} \langle N_{bump}(Z_{test}) \rangle \cdot e^{\frac{Z^2_{local} - Z^2_{test}}{2}}\right)$$

$\langle N_{bump}(Z_{test}) \rangle$ is average number of excesses with $Z > Z_{test}$

UPPER LIMITS

- t_0 distribution (in the null hypothesis)

t_0 observed

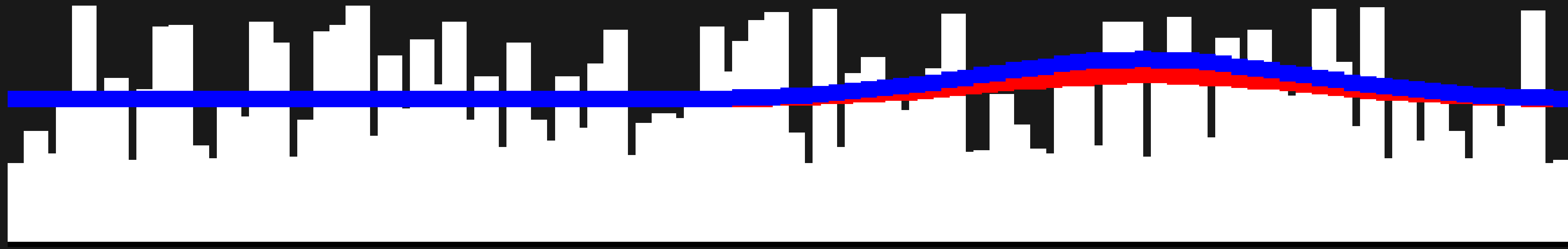


p-value < 5%

Use $-2\log\frac{\mathcal{L}(\mu=\mu_0)}{\mathcal{L}(\hat{\mu})}$

Use μ_0 to be excluded as null hypothesis H_0 and $\hat{\mu}$ as alternative hypothesis H_1 Raise μ_0 until p-value=5% (95% CL)

μ_1 -> No exclusion

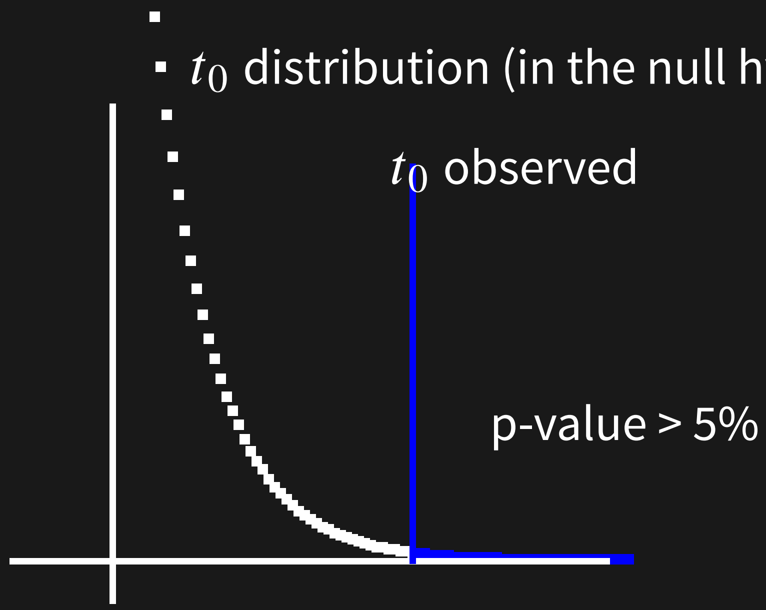


UPPER LIMITS

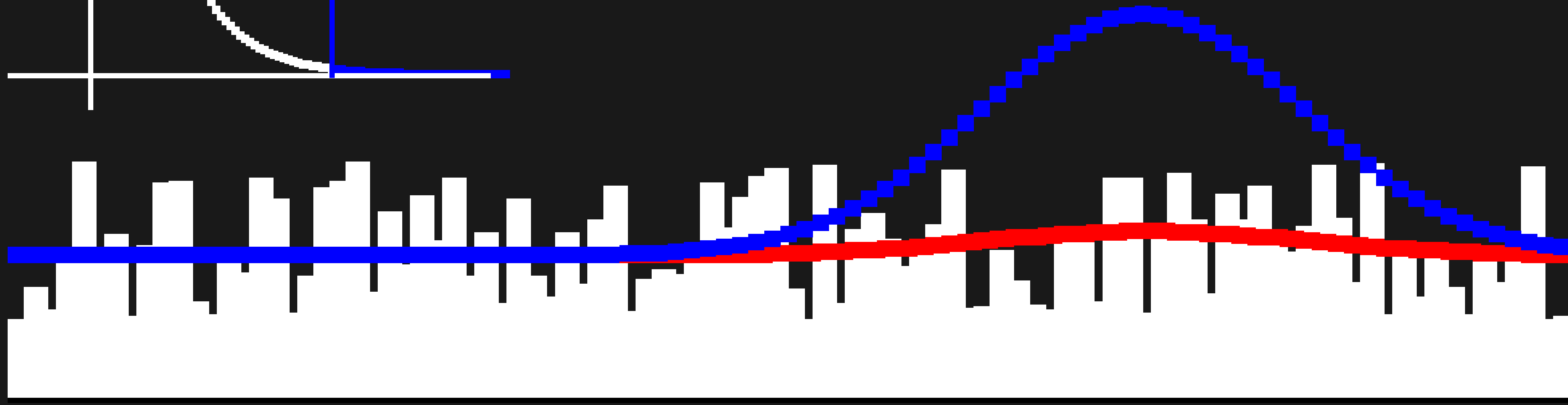
t_0 distribution (in the null hypothesis)

Use $-2\log\frac{\mathcal{L}(\mu=\mu_0)}{\mathcal{L}(\hat{\mu})}$

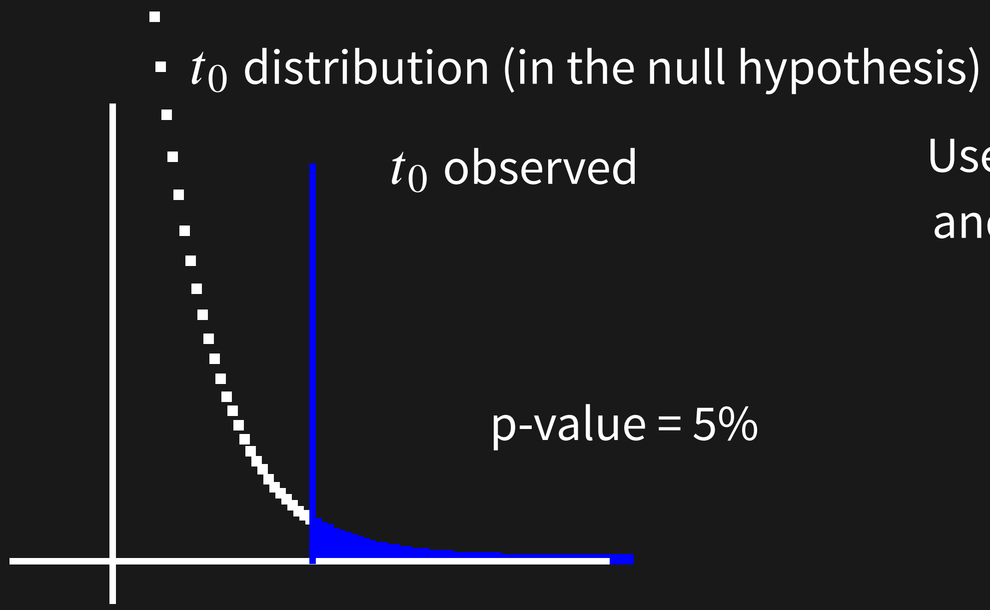
Use μ_0 to be excluded as null hypothesis H_0 and $\hat{\mu}$ as alternative hypothesis H_1 Raise μ_0 until p-value=5% (95% CL)



μ_2 -> Too strong exclusion



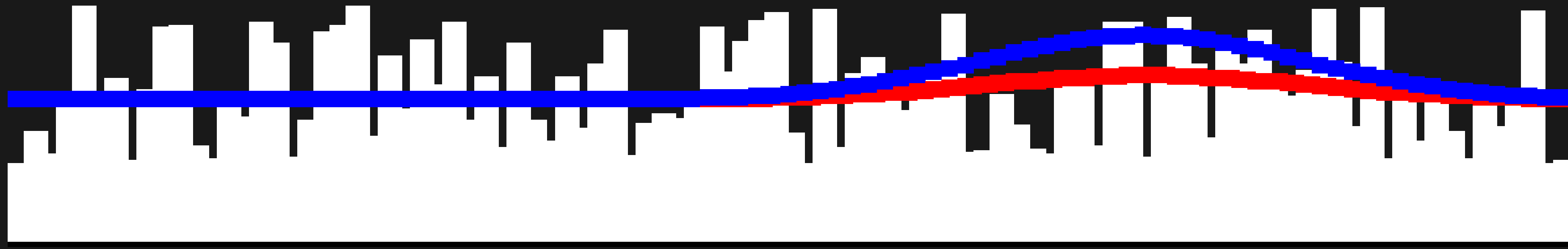
UPPER LIMITS



Use $-2\log\frac{\mathcal{L}(\mu=\mu_0)}{\mathcal{L}(\hat{\mu})}$

Use μ_0 to be excluded as null hypothesis H_0 and $\hat{\mu}$ as alternative hypothesis H_1 Raise μ_0 until p-value=5% (95% CL)

μ_3 -> exclusion

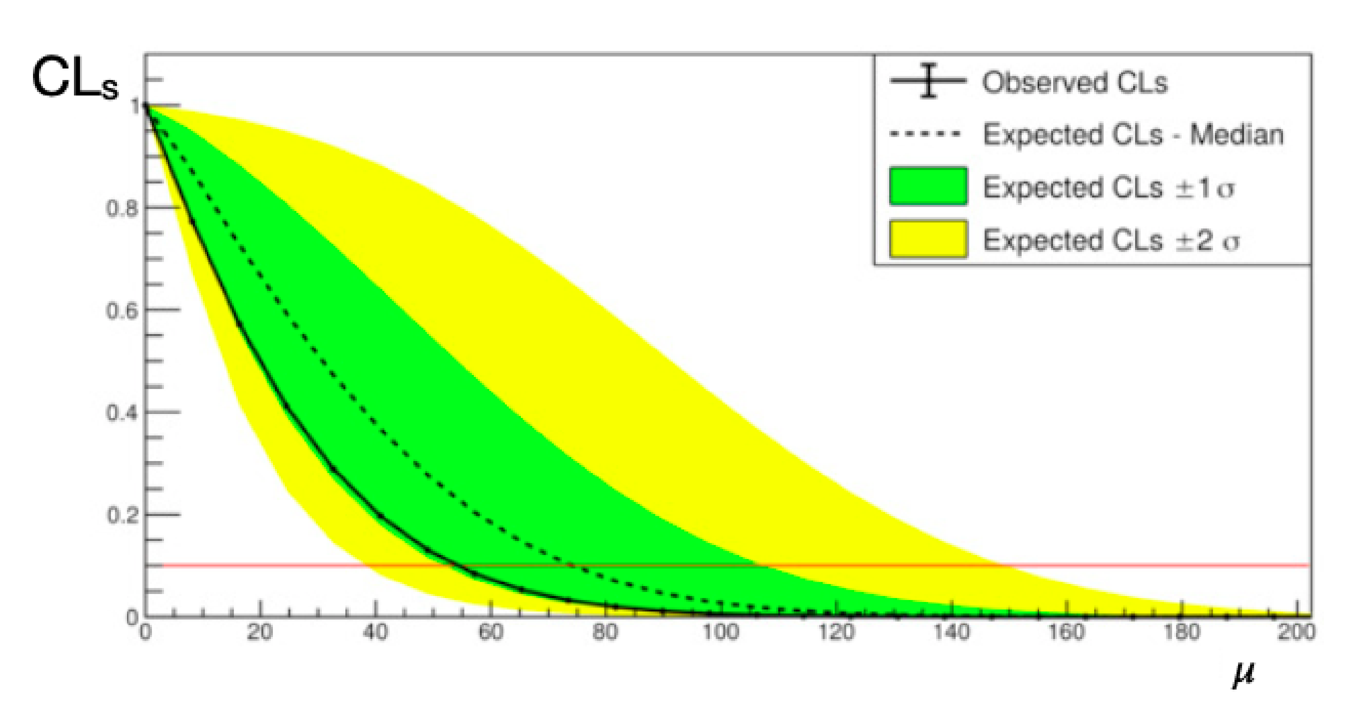


CL_s LIMITS

Problem: using p_{μ_0} can get strong limits for underfluctuations
Have to modify the p-value to avoid setting limits without sensitivity

$$CL_s = p_{\mu_0}/p_{\mu=0} = CL_{s+b}/CL_b$$

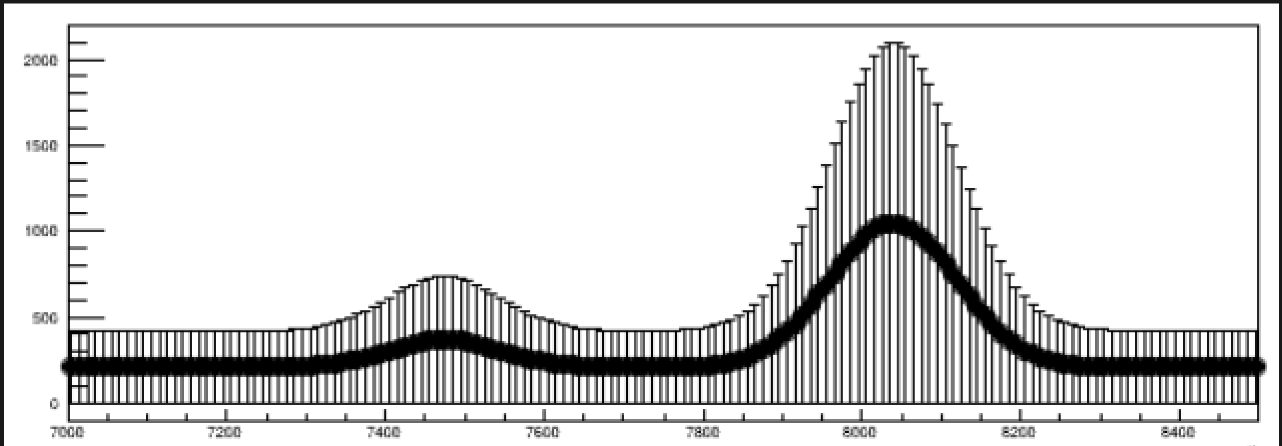
Slightly more conservative, HEP standard



CL_s LIMITS, EXPECTED

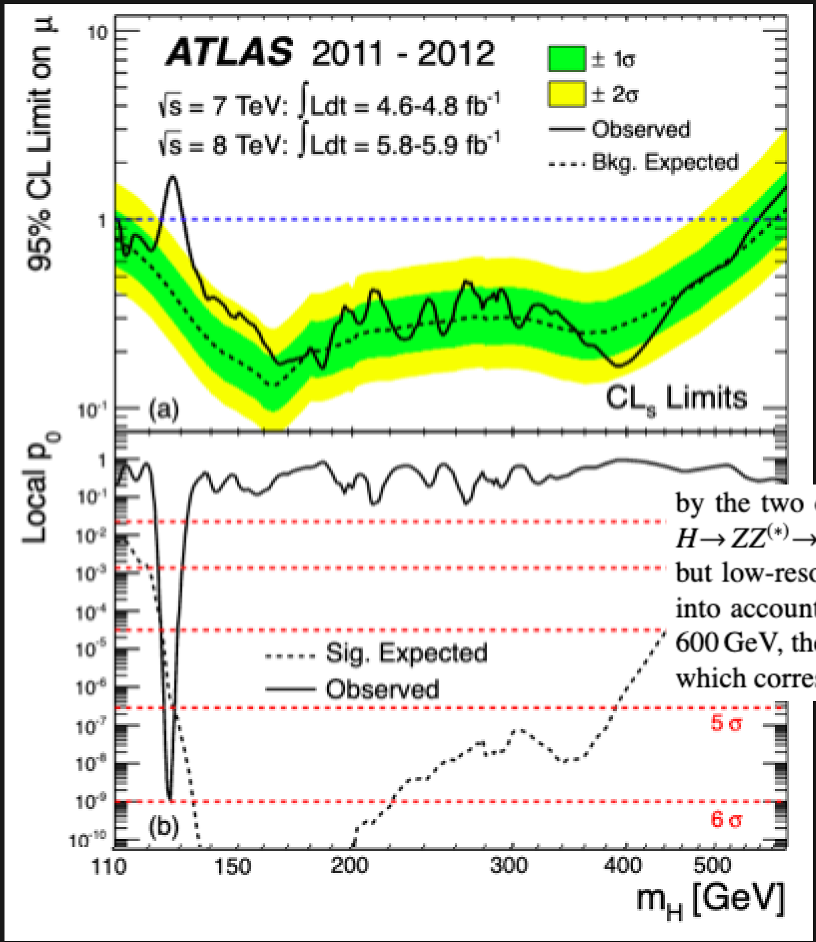
How to compare observed limits (from the data) to the background-only hypothesis? computed using:

- MC: generate toys in the H_0 hypothesis, use median and std. dev.
- "Asimov dataset" with no fluctuations; the $\mathcal{L}$ maximizes in the H_0 hypothesis



HIGGS DISCOVERY

For each m_H point,
e 95% CL_s is computed
as in the slide before



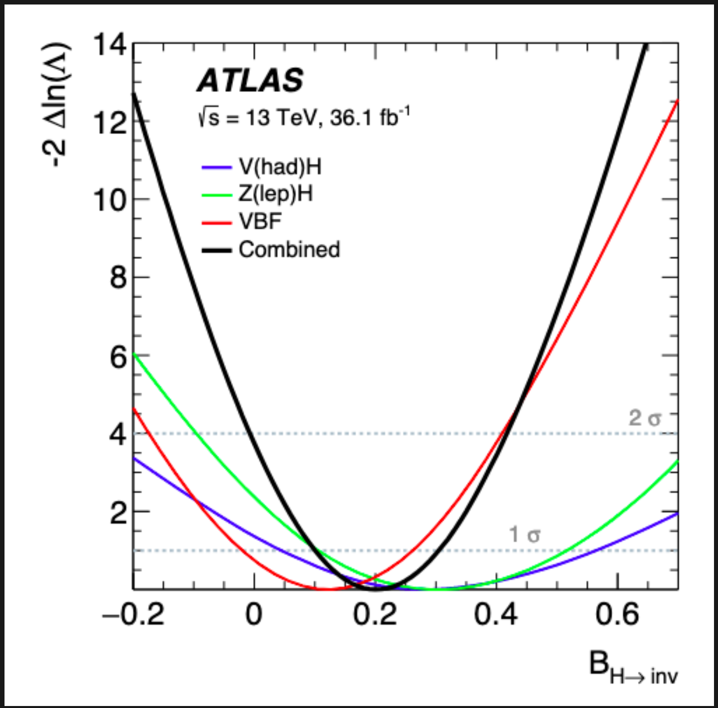
by the two channels with the highest mass resolution, $H \to ZZ^{(*)} \to 4\ell$ and $H \to \gamma\gamma$, and the equally sensitive but low-resolution $H \to WW^{(*)} \to \ell\nu\ell\nu$ channel. Taking into account the entire mass range of the search, 110–600 GeV, the global significance of the excess is 5.1 σ, which corresponds to $p_0 = 1.7 \times 10^{-7}$.

CONFIDENCE INTERVALS

Use as test statistics

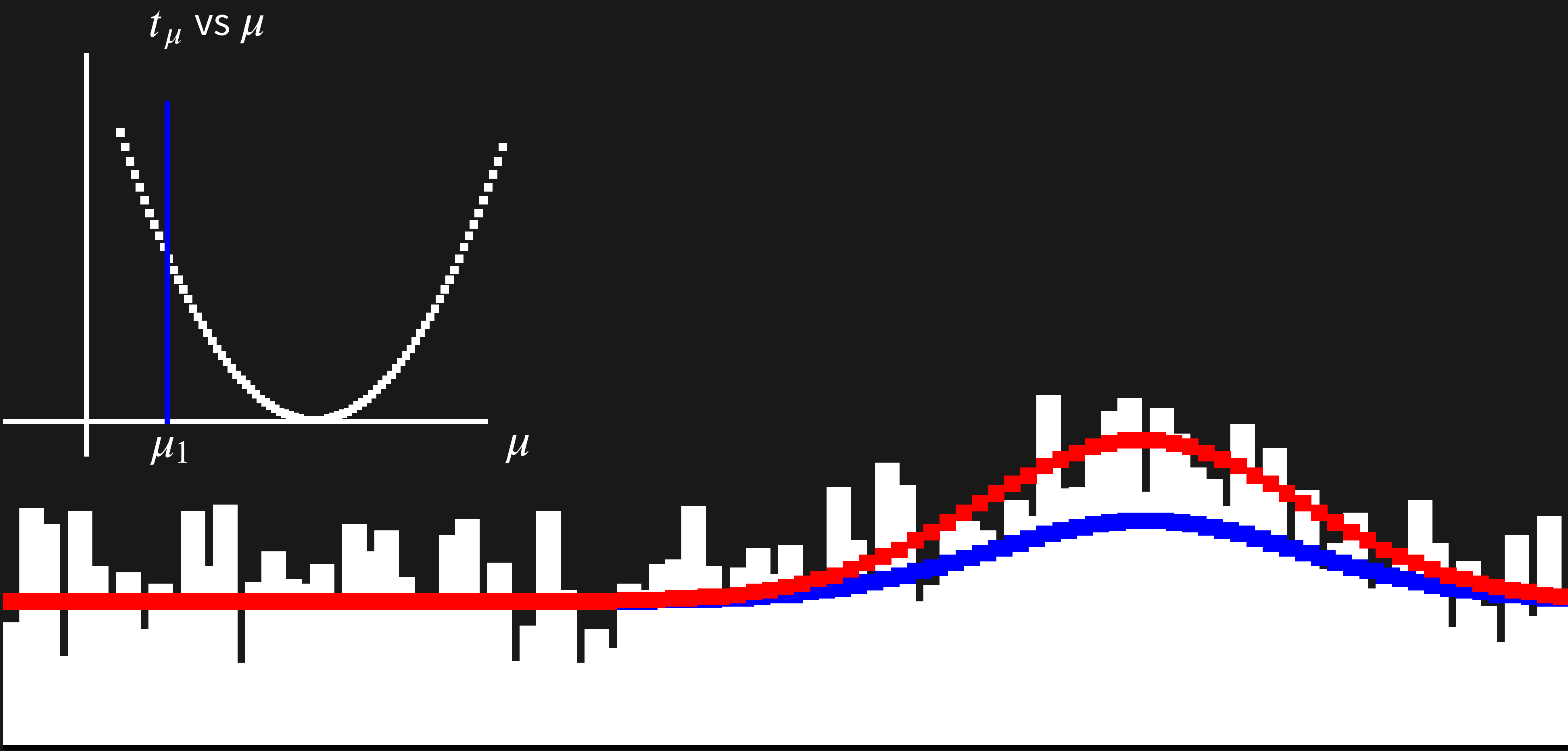
$$t_{\mu_0} = -2\log\frac{\mathcal{L}(\mu=\mu_0,\hat{\hat{\theta}}_{\mu_0})}{\mathcal{L}(\hat{\mu},\hat{\theta}_{\mu_0})}$$

to exclude μ values outside the interval
Example: Higgs to invisible



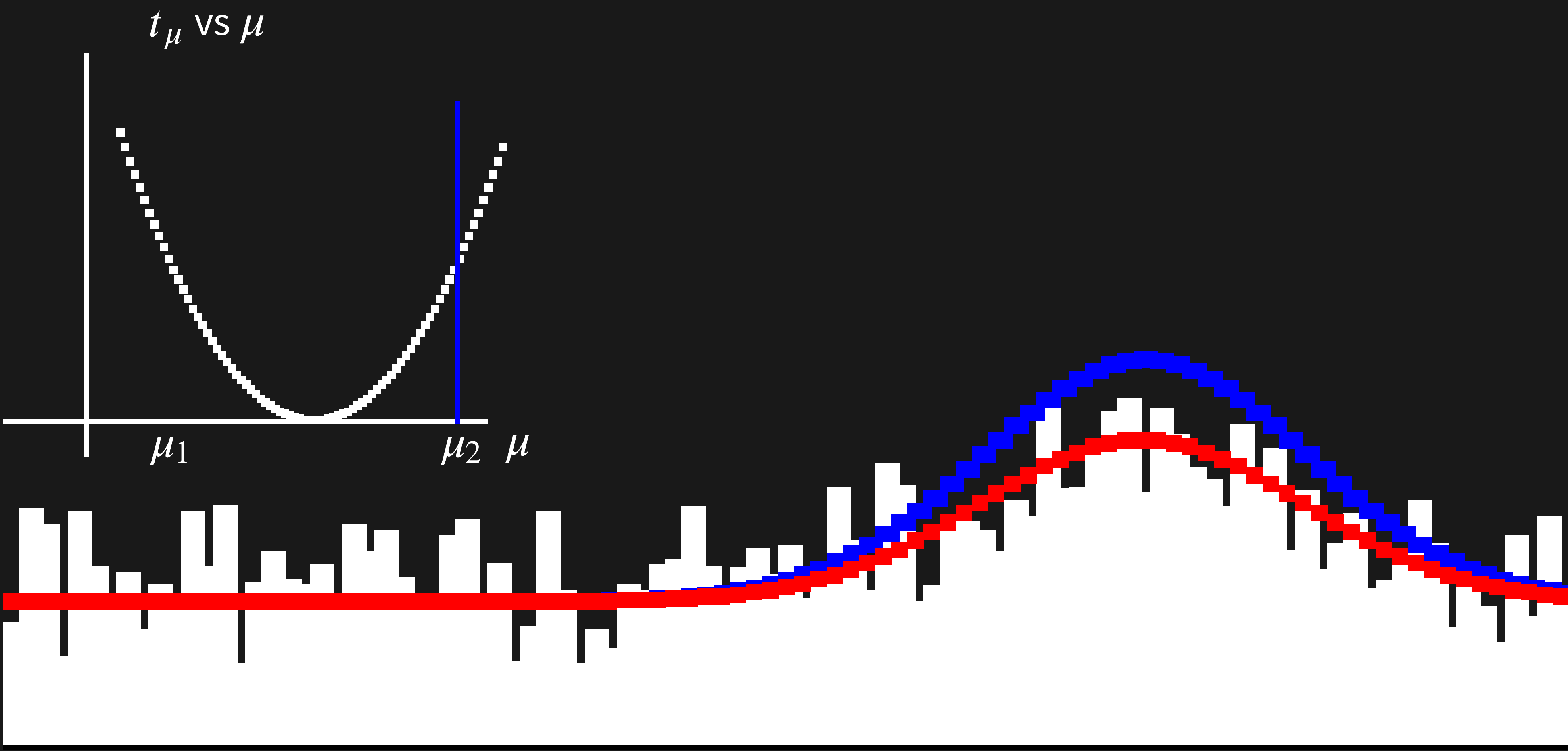
CONFIDENCE INTERVALS EXAMPLE

$\mathcal{L}(\mu = \mu_1)$ small, $\mathcal{L}(\hat{\mu})$ large; $-2\ln(\frac{\mathcal{L}(\mu=\mu_1)}{\mathcal{L}(\hat{\mu})})$ large



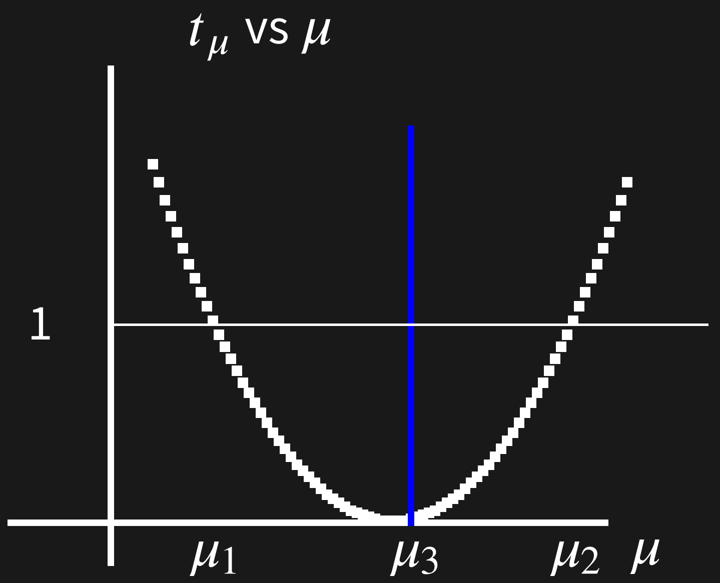
CONFIDENCE INTERVALS EXAMPLE

$\mathcal{L}(\mu = \mu_2)$ small, $\mathcal{L}(\hat{\mu})$ large; $-2\ln\left(\frac{\mathcal{L}(\mu=\mu_2)}{\mathcal{L}(\hat{\mu})}\right)$ large

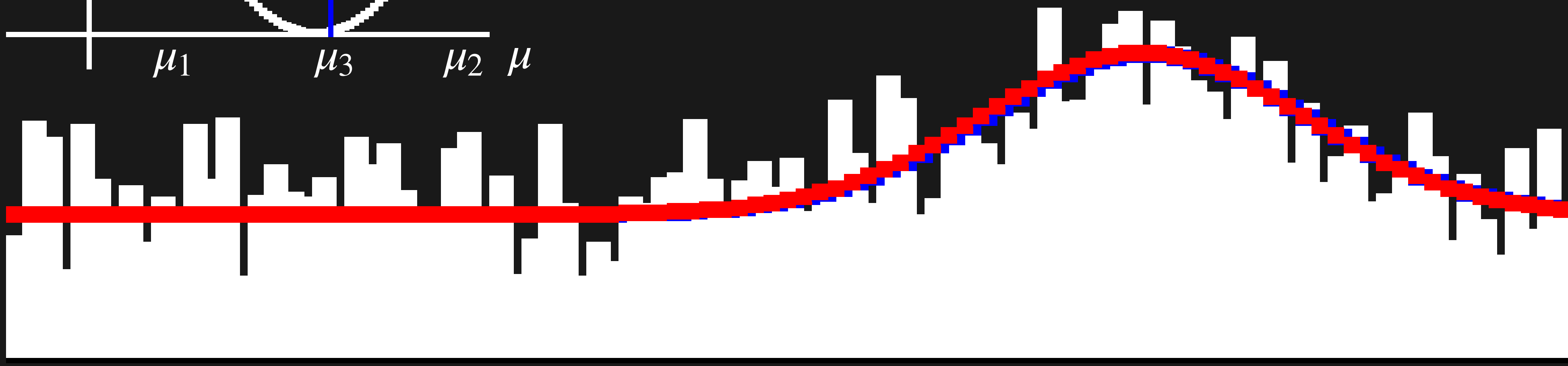


CONFIDENCE INTERVALS EXAMPLE

$\mathcal{L}(\mu = \mu_3)$ similar to $\mathcal{L}(\hat{\mu})$; $-2\ln(\frac{\mathcal{L}(\mu=\mu_3)}{\mathcal{L}(\hat{\mu})})$ small



crossing $t_\mu = 1$ gives the $\pm 1\sigma$



pyhf Tutorial

Welcome!

HOW TO DO IT: PYHF

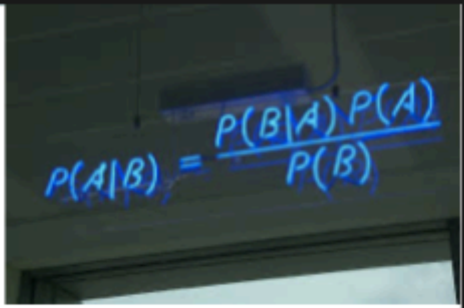
Python implementation of HistFactory

https://pyhf.github.io/pyhf-tutorial/introduction.html

Welcome to the `pyhf` tutorial! We'll first point you towards our documentation website (pyhf.readthedocs.io/) and recommend that you visit it for much more detailed explanations and examples. Let's dive right in.

We won't review the full pedagogy of `HistFactory`, so instead we'll point you to the `pyhf` talk at SciPy 2020.

BAYESIAN STATISTICS



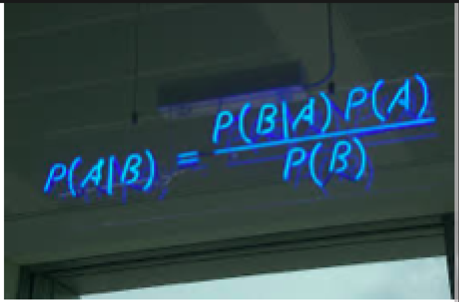
$$p(\boldsymbol{\theta}|\mathcal{D}, M) = \frac{p(\mathcal{D}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)}{\int \mathrm{d}\boldsymbol{\theta}\, p(\mathcal{D}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)}$$

Free params

Dataset

Model

BAYESIAN STATISTICS



Posterior probability density

Likelihood

Prior

$$p(\boldsymbol{\theta}|\mathcal{D}, M) = \frac{p(\mathcal{D}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)}{\int \mathrm{d}\boldsymbol{\theta}\, p(\mathcal{D}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)}$$

Free params

Dataset

Model

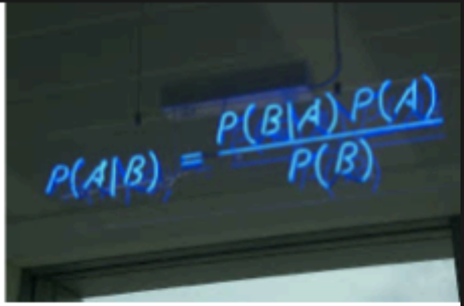
Evidence (Z)

$$Z = P(\mathcal{D}|M) = \int \mathrm{d}\boldsymbol{\theta}\, p(\mathcal{D}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)$$

Marginalized Probability density

$$p(\theta_i|\mathcal{D}, M) = \int p(\theta|\mathcal{D}, M) \prod_{i \neq j} \mathrm{d}\theta_j$$

BAYESIAN STATISTICS



Posterior probability density — *Likelihood* — *Prior*

Can be many — *Non trivial*

$$p(\boxed{\theta}|\mathcal{D}, M) = \frac{\boxed{p(\mathcal{D}|\theta, M)}\, p(\theta|M)}{\int \mathrm{d}\theta\, p(\mathcal{D}|\theta, M) p(\theta|M)}$$

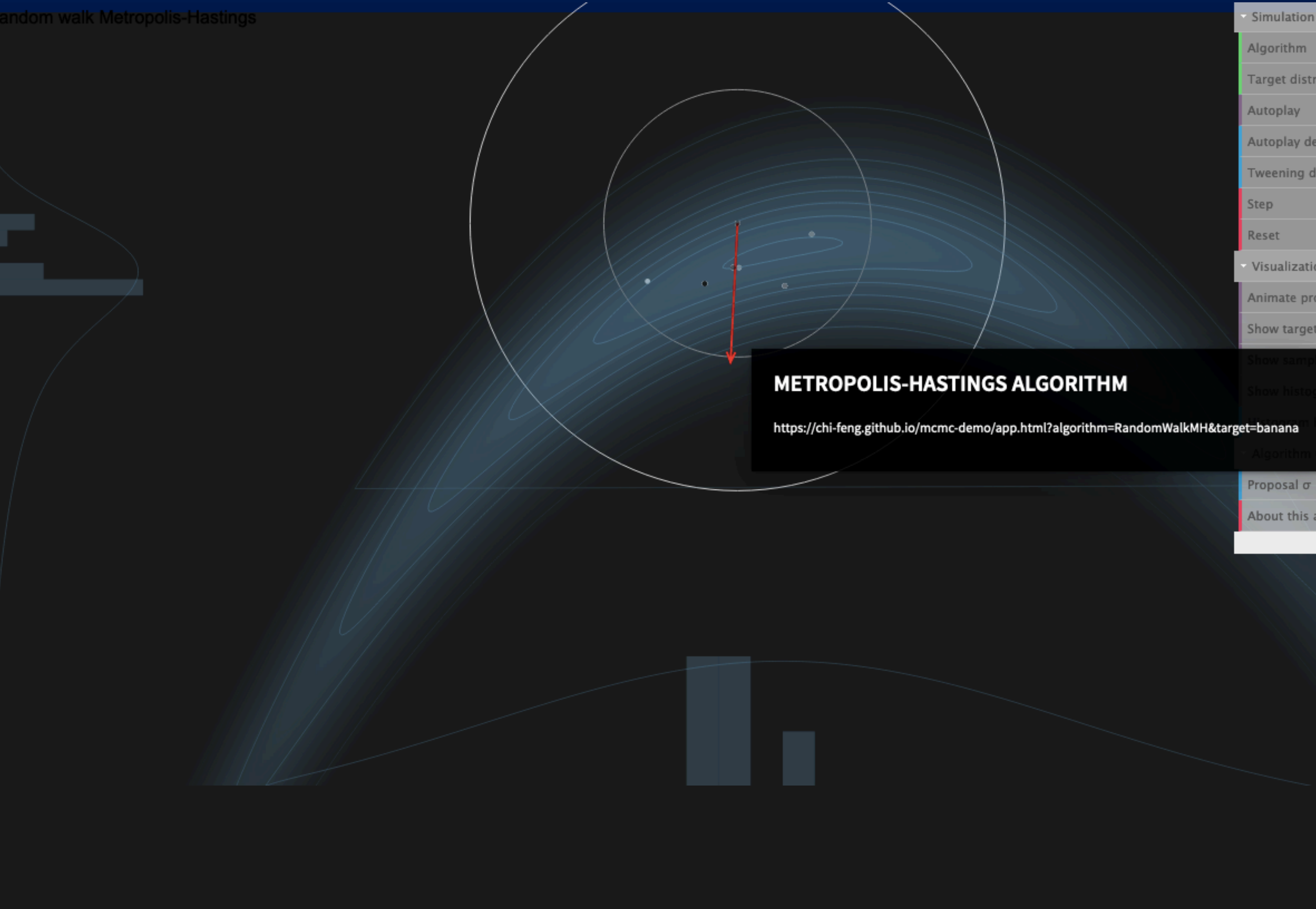
Free params — *Dataset* — *Model* — *Evidence (Z)*

$$Z = P(\mathcal{D}|M) = \boxed{\int \mathrm{d}\theta\, p(\mathcal{D}|\theta, M) p(\theta|M)}$$

Numerically intensive

Marginalized probability density

$$p(\theta_i|\mathcal{D}, M) = \boxed{\int p(\theta|\mathcal{D}, M) \prod_{i \neq j} \mathrm{d}\theta_j}$$



METROPOLIS-HASTINGS ALGORITHM

https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=banana

BAT.jl
Bayesian Analysis Toolkit

BAT

Search docs

lome

Table of contents
Citing BAT.jl
Learning (more about) Julia
Acknowledgements

nstallation

utorial

PI Documentation

lotting

xperimental Features

nternal API

Home

Edit on GitHub

BAT.jl Documentation

BAT.jl is a Bayesian Analysis Toolkit in Julia. It is a high-performance tool box for Bayesian inference with statistical models expressed in a general-purpose programming language instead of a domain-specific language.

Typical applications for this package are parameter inference given a model (in the form of a likelihood function and prior), the comparison of different models in the light of a given data set, and the test of the validity of a model to represent the data set at hand. BAT.jl provides access to the full Bayesian posterior distribution to enable parameter estimation, limit setting and uncertainty propagation. BAT.jl also provides supporting functionality like plotting recipes and reporting functions.

BAT.jl is implemented in pure Julia and allows for a flexible definition of mathematical mo enabling the user to code for the performance required for computationally expensive numerical operations. BAT.jl provides implementations (internally and via other Julia packages) of algorithms for sampling, optimization and integration. BAT's main focus is on the analysis of complex custom models. It is designed to enable parallel code execution at various levels (running multiple MCMC chains in parallel is provided out-of-the-box).

BAYESIAN ANALYSIS TOOLKIT
https://bat.github.io/BAT.jl/stable/

It's possible to use BAT.jl with likelihood functions implemented in languages other than Julia: Julia allows for calling code in C and Fortran, C++, Python and several other languages directly. In addition, BAT.jl provides (as an experimental feature) a very lightweight binary RPC protocol that is easy to implement, to call non-Julia likelihood functions written in another language and running in separate processes.

BAT.jl originated as a rewrite/redesign of BAT, the Bayesian Analysis Toolkit in C++. BAT.jl now offer a different set of functionality and a wider variety of algorithms than it's C++ predecessor.

CONCLUSIONS

- Statistics important for precise claims
- Can use likelihood to build arbitrarily complex analyses
- Can build in systematics uncertainties
- Test statistics can be used for: Discovery, Limit setting and param estimation
- Bayesian methods: no need test statistics, need priors

THANKS FOR YOUR ATTENTION!