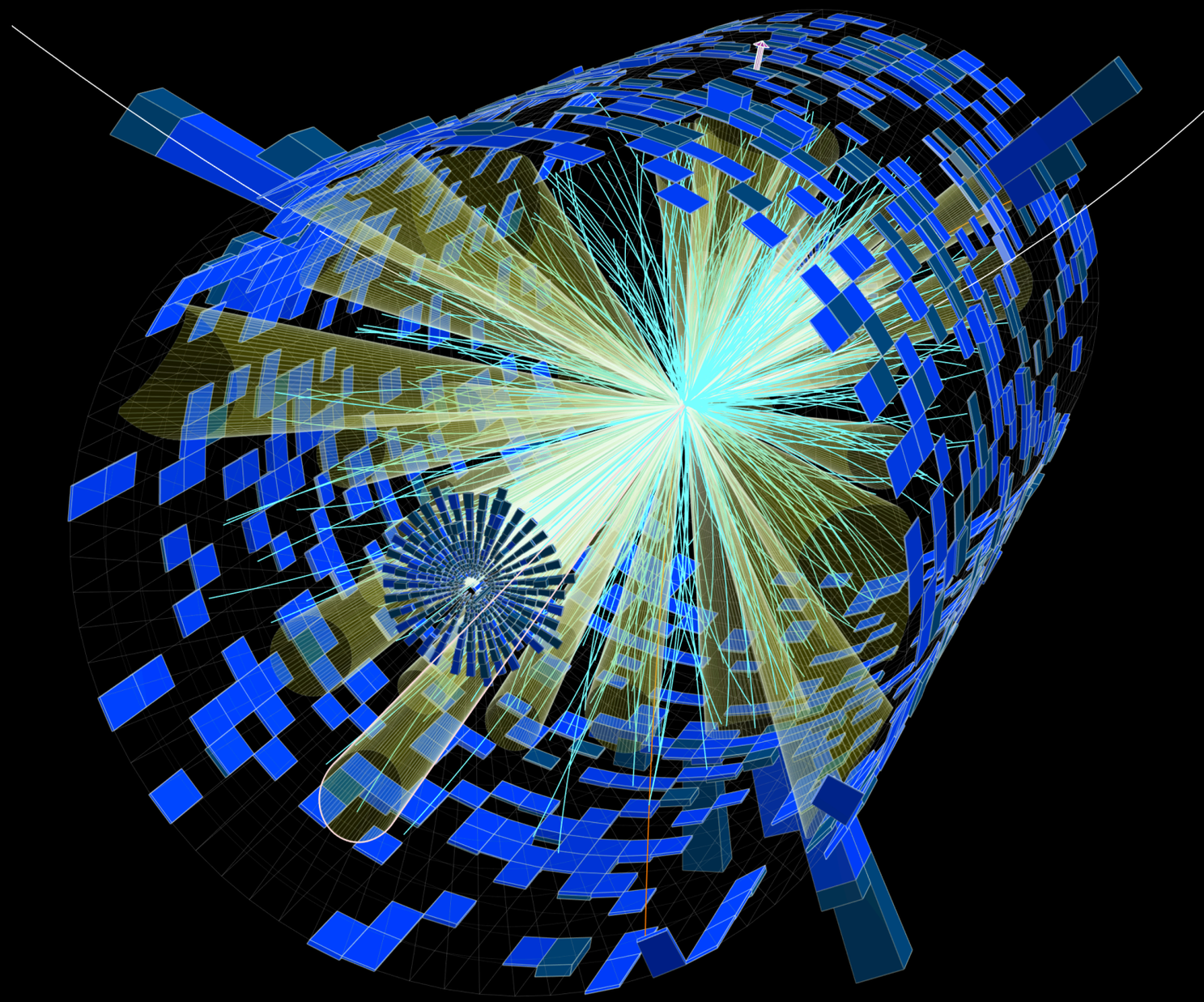
NYU Center for Data Science

Center for Cosmology and particle physics

STATISTICS



@KyleCranmer
New York University
Department of Physics
Center for Data Science
CILVR Lab

INTRODUCTION

Statistics plays a vital role in science, it is the way that we:

- quantify our knowledge and uncertainty
- communicate results of experiments

Big questions:

- how do we make discoveries, measure or exclude theoretical parameters, ...
- how do we get the most out of our data
- how do we incorporate uncertainties
- how do we make decisions

In these talks I will try to:

- **explain** some fundamental ideas & prove a few things
- **enrich** what you already know
- **expose** you to some new ideas

LECTURE NOTES

Practical Statistics for the LHC

Kyle Cranmer
Center for Cosmology and Particle Physics, Physics Department, New York University, USA

Abstract
This document is a pedagogical introduction to statistics for particle physics. Emphasis is placed on the terminology, concepts, and methods being used at the Large Hadron Collider. The document addresses both the statistical tests applied to a model of the data and the modeling itself . I expect to release updated versions of this document in the future.

Links:
On Authorea
arxiv:1503.07622

Contents

Probability & Statistics

Terminology & Definitions

Terms

The next lectures will rely on a clear understanding of these terms:

- Random variables / “observables” x
- Probability mass and probability density function (pdf) $p(x)$ or $f(x)$
- Parametrized Family of pdfs / “model” $p(x|\alpha)$
- Parameter α
- Likelihood $L(\alpha)$
- Estimate (of a parameter) $\hat{\alpha}(x)$

Probability Mass Functions

When dealing with discrete random variables, define a **Probability Mass Function** as probability for ith possibility

$$P(x_i) = p_i$$

Defined as limit of long term frequency

- probability of rolling a 3 := limit $_{\#trials \to \infty}$ (# rolls with 3 / # trials)
  - you don't need an infinite sample for definition to be useful

And it is normalized

$$\sum_i P(x_i) = 1$$

Probability Density Functions

When dealing with continuous random variables, need to introduce the notion of a **Probability Density Function**

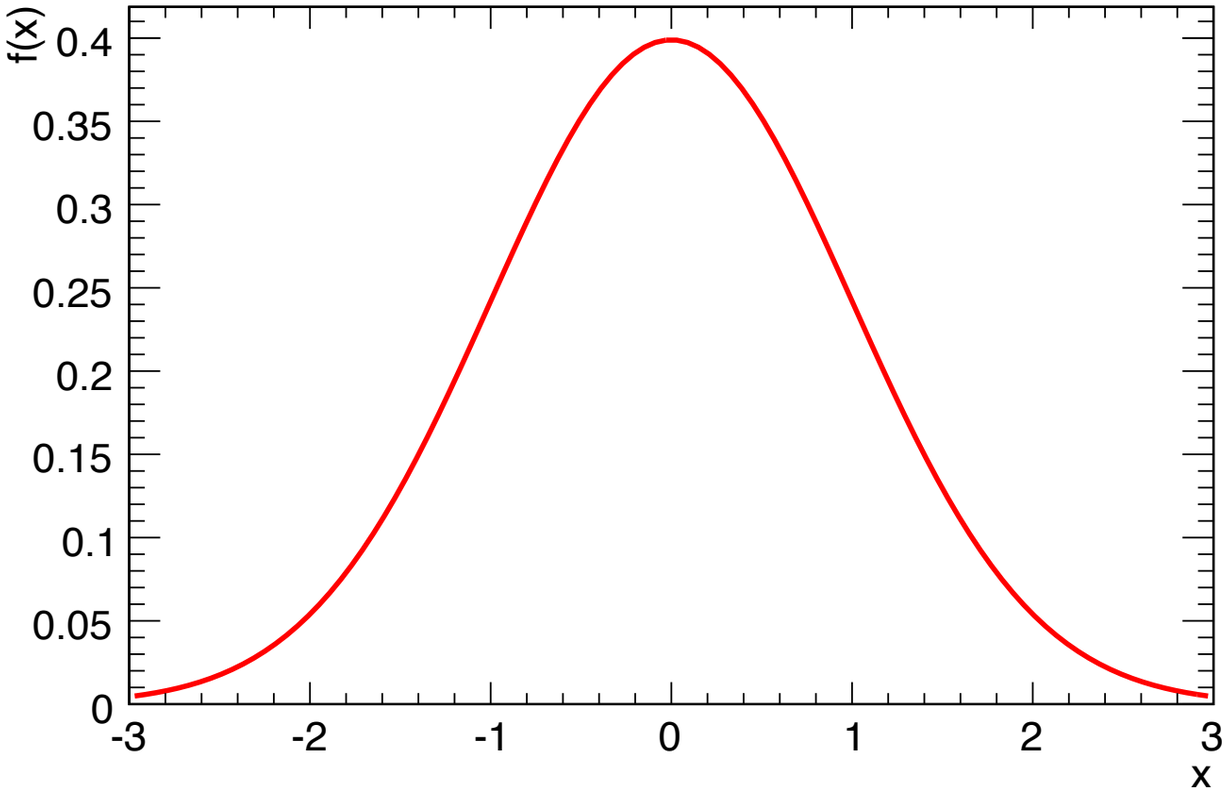
$$P(x \in [x, x + dx]) = f(x)dx$$

Note, $f(x)$ is NOT a probability

PDFs are always normalized

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

(ok for $f(x) > 1$, it's a density)

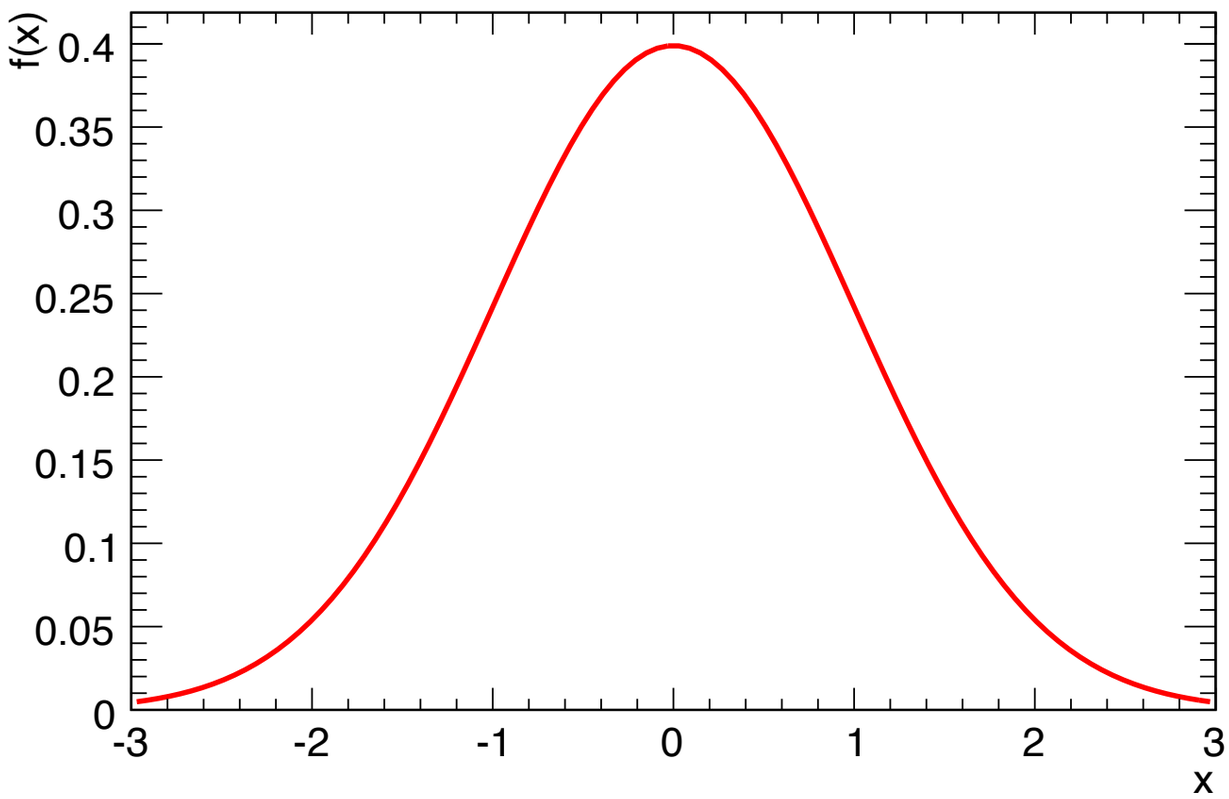


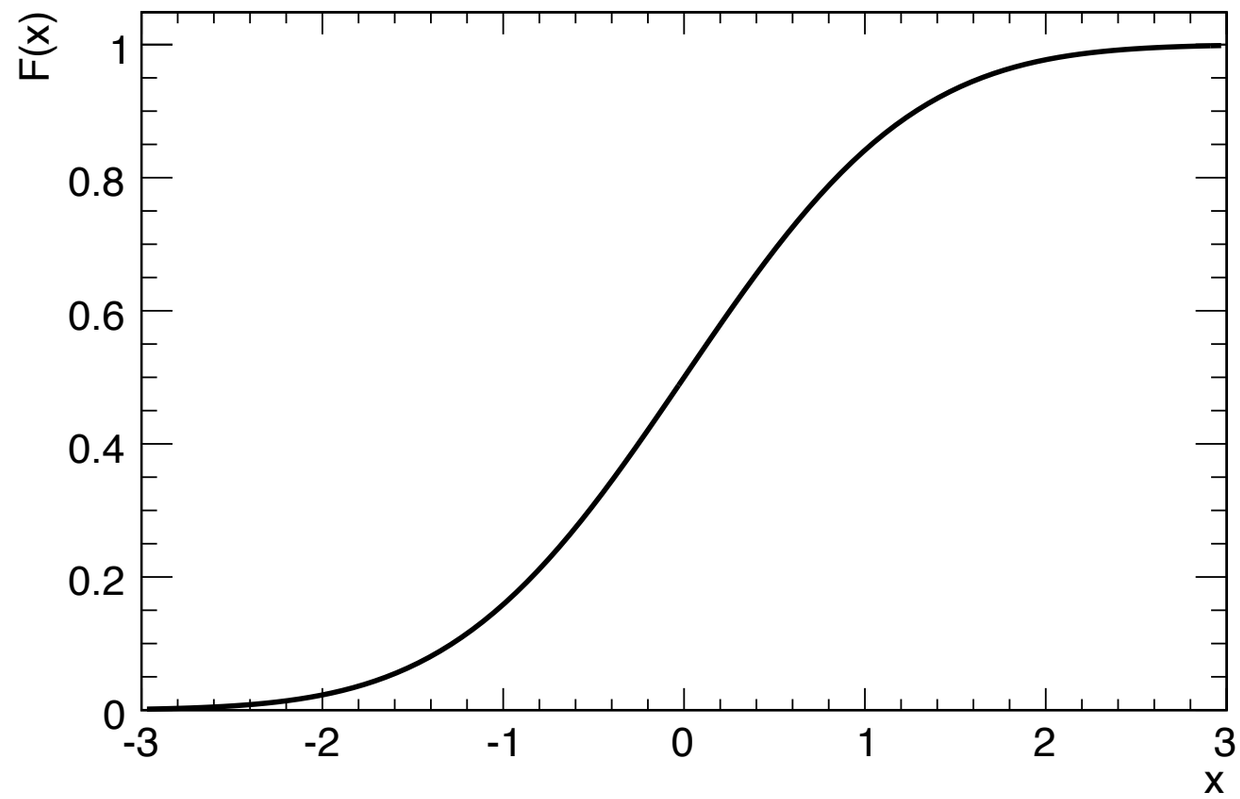
Cumulative Density Functions

Often useful to use a cumulative distribution:

- in 1-dimension:

$$\int_{-\infty}^{x} f(x')dx' = F(x)$$



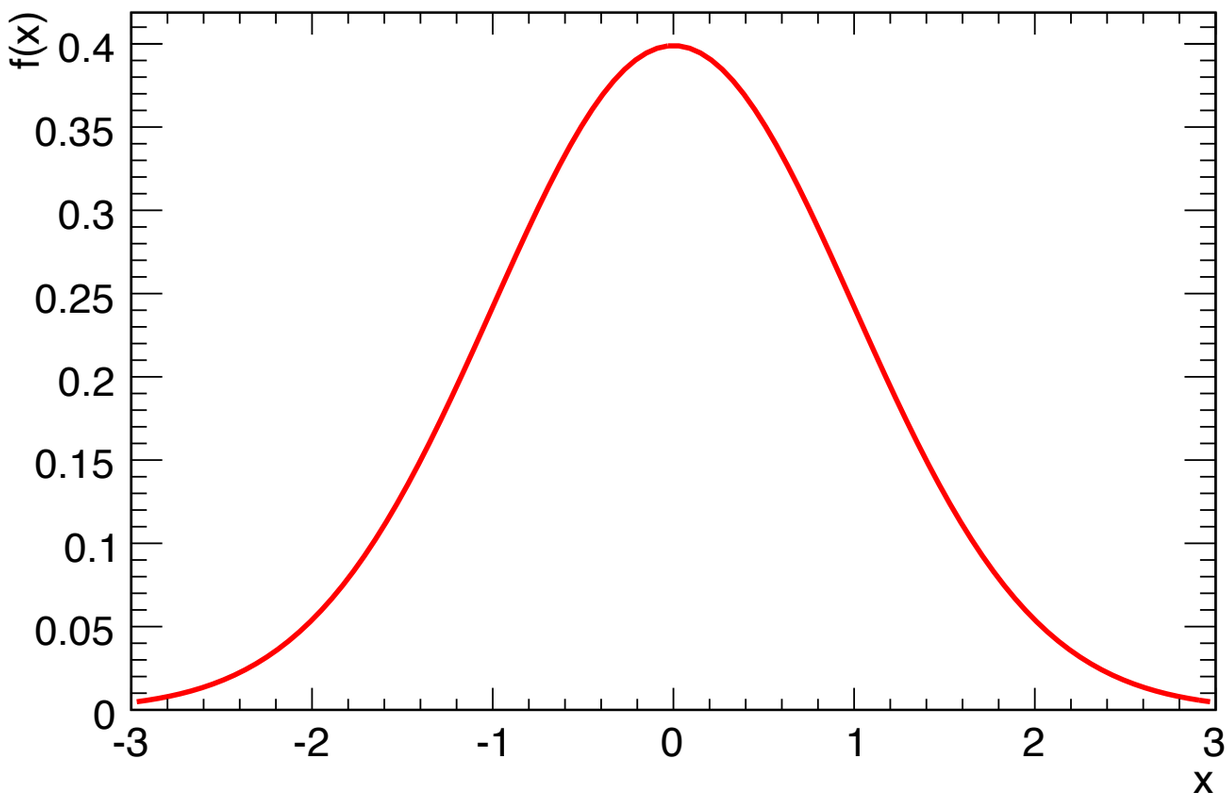


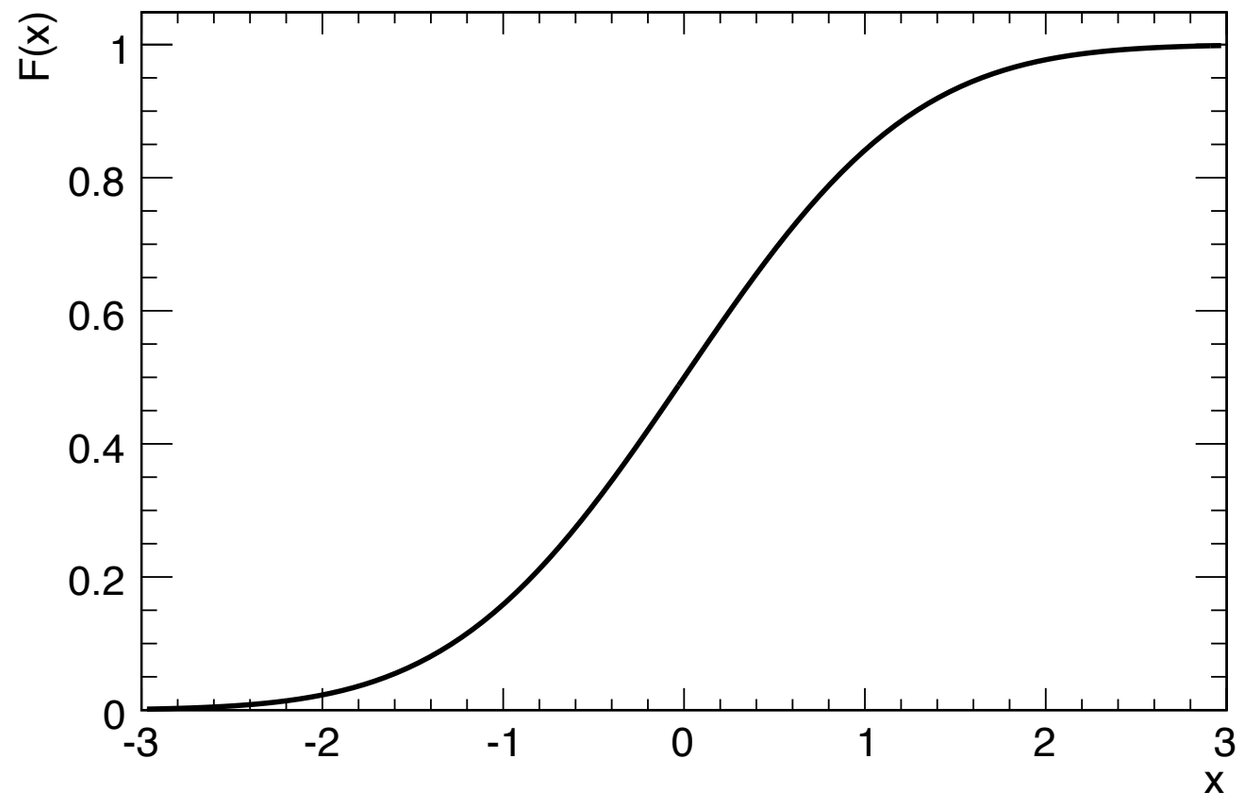
Cumulative Density Functions

Often useful to use a cumulative distribution:

- in 1-dimension:

$$\int_{-\infty}^{x} f(x')dx' = F(x)$$





- alternatively, define density as partial of cumulative:

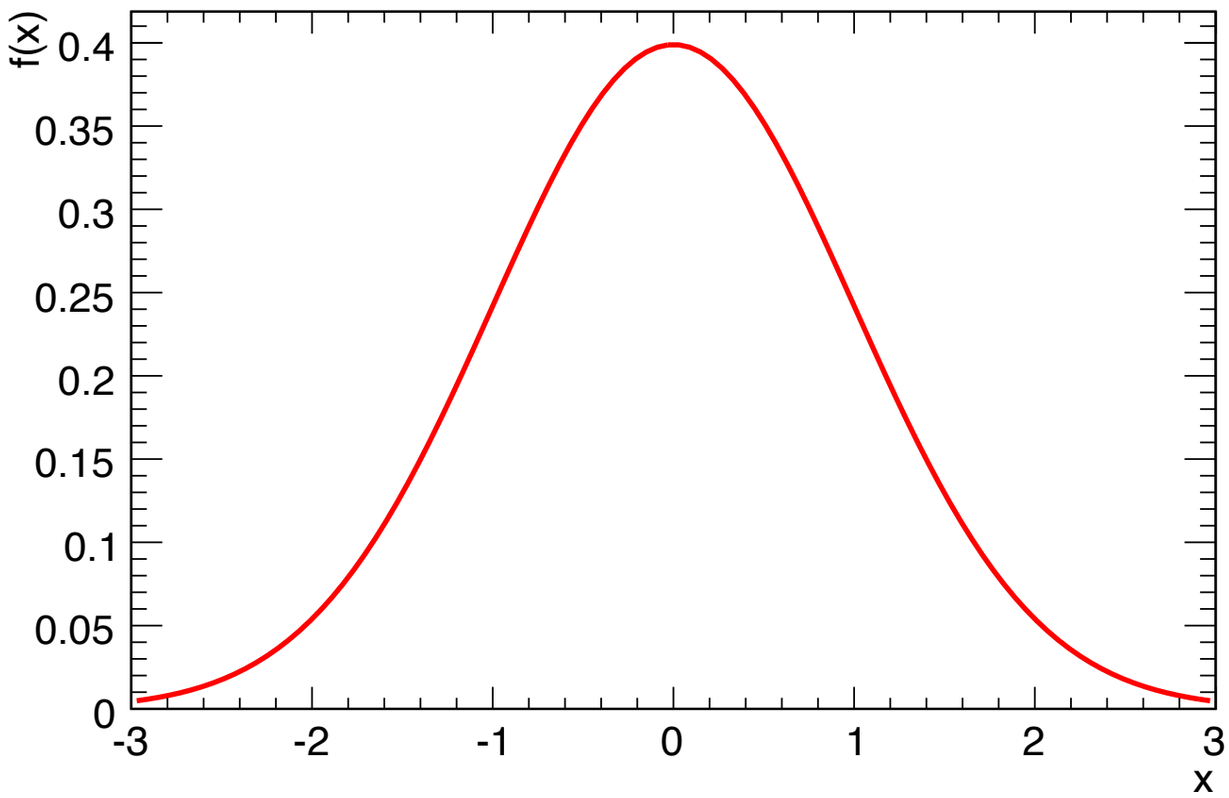
$$f(x) = \frac{\partial F(x)}{\partial x}$$

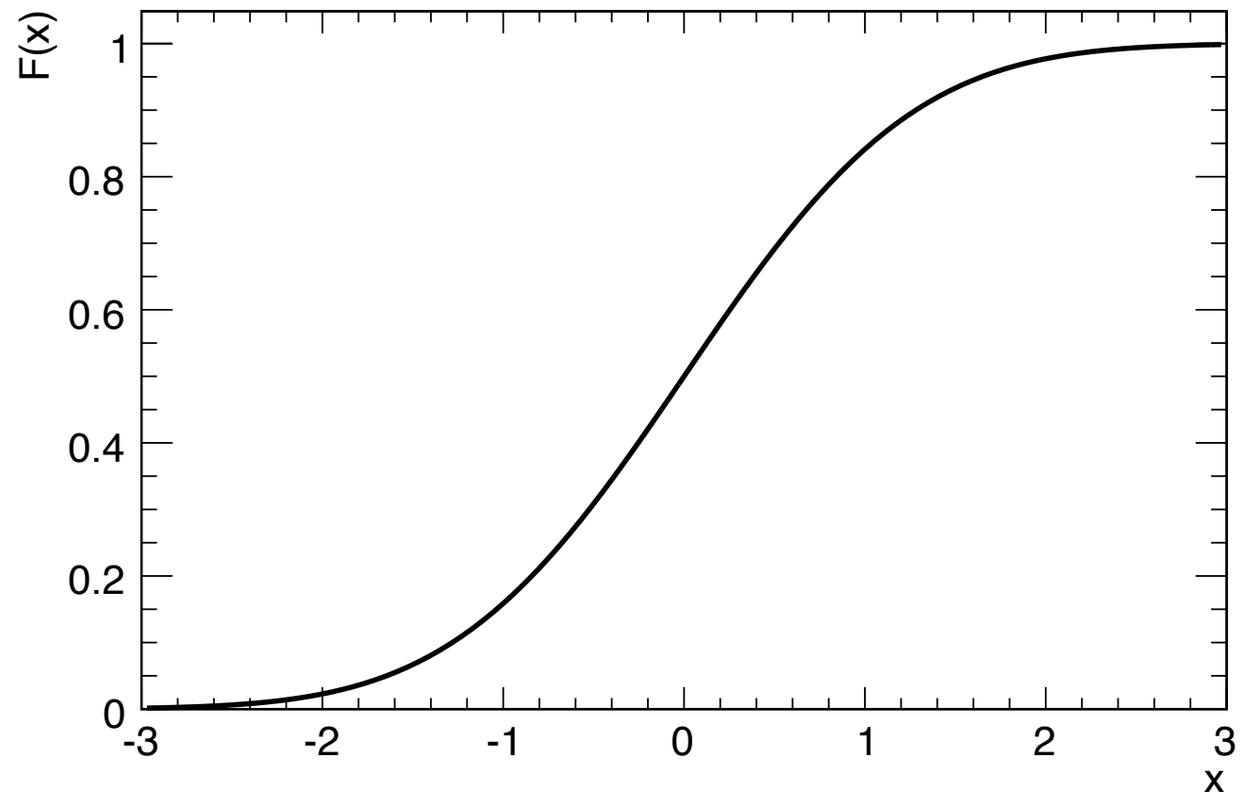
Cumulative Density Functions

Often useful to use a cumulative distribution:

- in 1-dimension:

$$\int_{-\infty}^{x} f(x')dx' = F(x)$$





- alternatively, define density as partial of cumulative:

$$f(x) = \frac{\partial F(x)}{\partial x}$$

- same relationship as total and differential cross section:

$$f(E) = \frac{1}{\sigma}\frac{\partial \sigma}{\partial E}$$

HISTOGRAM $\{X_I\} \rightarrow F(X)$

Given a set of observations $\{x_i\}$ we can approximate the pdf with a histogram.

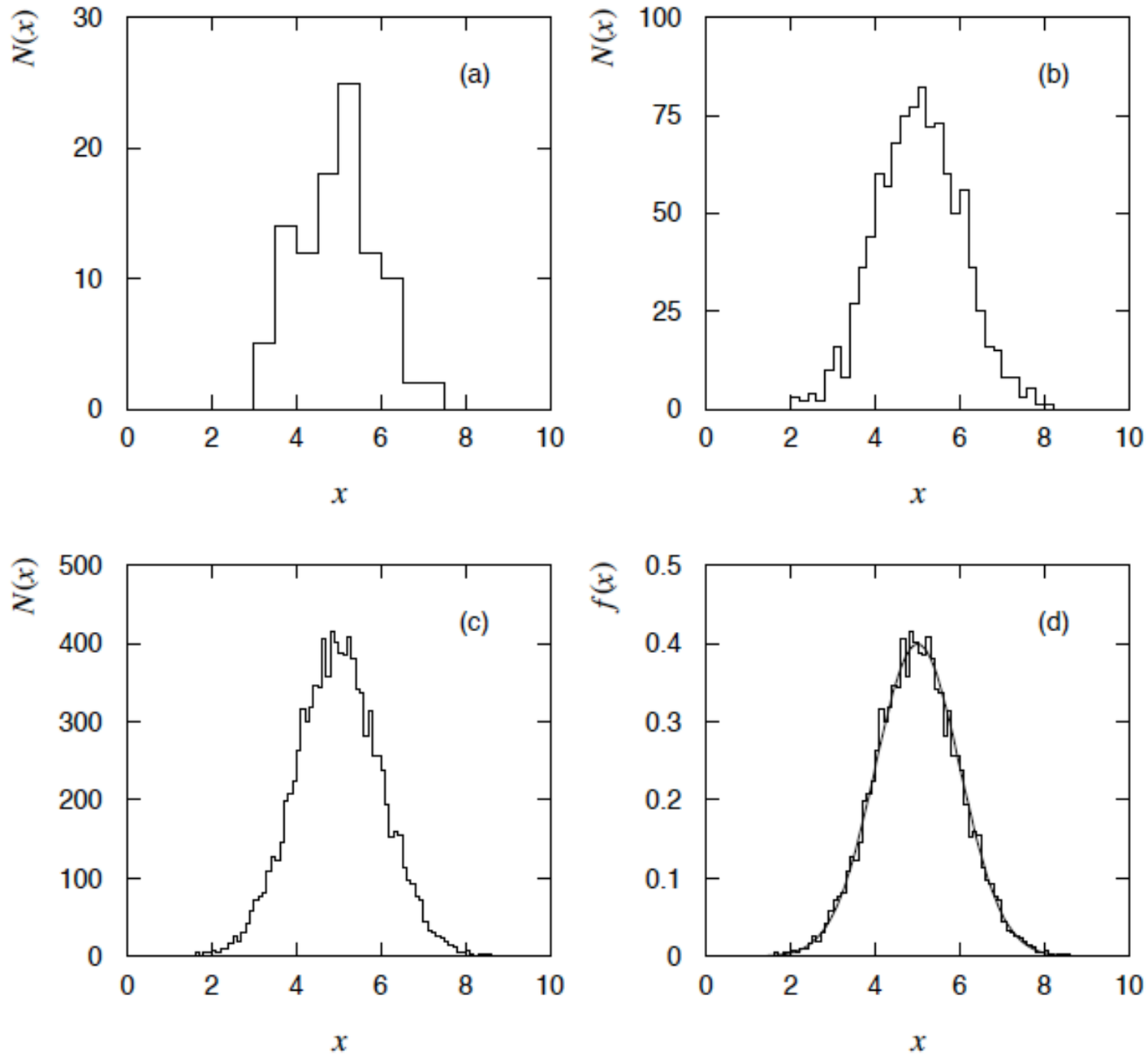
Think of a pdf as a histogram with:

infinite data sample,
zero bin width,
normalized to unit area.

$$f(x) = \frac{N(x)}{n\Delta x}$$

$n =$ number of entries

$\Delta x =$ bin width



[G. Cowan]

Parametrized families / models

Often we are interested in a parametried family of pdfs

- We will write these as: $f(x|\alpha)$ said "f of x given α"
  - where α are the parameters of the "model" (written in greek characters)

A discrete example:

- The Poisson distribution is a probability mass function for n, the number of events one observes, when one expects μ events

$$Pois(n|\mu) = \mu^n \frac{e^{-\mu}}{n!}$$

A continuous example

- The Gaussian distribution is a probability density function for a continuous variable x characterized by a mean μ and standard deviation σ

$$G(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The Likelihood Function

Consider the Poisson distribution describes a discrete event count n for a real-valued mean μ.

$$Pois(n|\mu) = \mu^n \frac{e^{-\mu}}{n!}$$

The **likelihood** of μ given n is the same equation evaluated as a function of μ

- Now it's a continuous function
- But it is not a pdf!

$$L(\mu) = Pois(n|\mu)$$

Common to plot the -ln *L* (or -2 ln *L*)

- helps avoid thinking of it as a PDF
- connection to χ^2 distribution

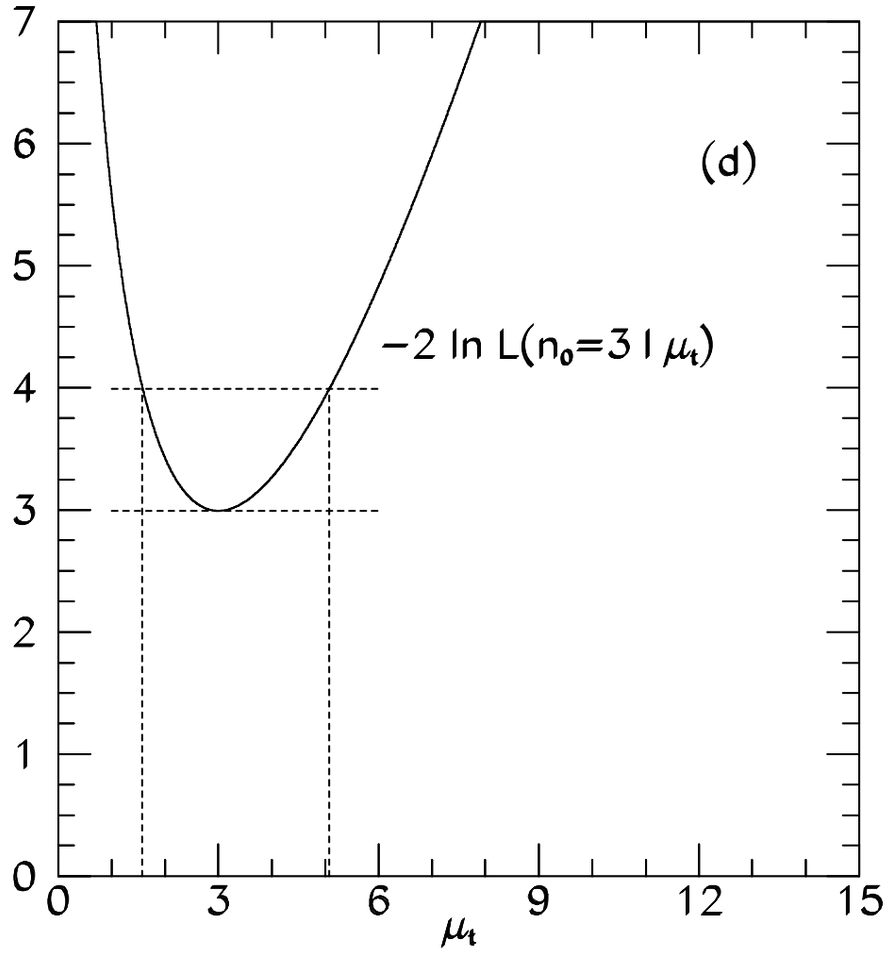


Figure from R. Cousins, Am. J. Phys. 63 398 (1995)

Repeated observations

In particle physics we are usually able to perform repeated observations of x that are **independent & identically distributed**

- These repeated observations are written $\{x_i\}$
- and the likelihood in that case is

$$L(\alpha) = \prod_i f(x_i|\alpha)$$

- and the log-likelihood is

$$\log L(\alpha) = \sum_i \log f(x_i|\alpha)$$

Marked Poisson Process

Given a subset of the data defined by some selection requirements

- eg. all events with 4 electrons with energy > 10 GeV
- n: number of events observed in the channel
- ν: number of events expected in the channel

Discriminating variable: a property of those events that can be measured and which helps discriminate the signal from background

- eg. the invariant mass of two particles
- $f(x)$: the p.d.f. of the discriminating variable x

$$\mathcal{D} = \{x_1, \ldots, x_n\}$$

Marked Poisson Process / Extended Likelihood:

$$\mathbf{f}(\mathcal{D}|\nu) = \mathrm{Pois}(n|\nu) \prod_{e=1}^{n} f(x_e)$$

PARAMETRIZING THE MODEL $\boldsymbol{\alpha} = (\mu, \boldsymbol{\theta})$

Parameters of interest (*μ*): parameters of the theory that modify the rates and shapes of the distributions, eg.

- the mass of a hypothesized particle
- the "signal strength" μ=0 no signal, μ=1 predicted signal rate

Nuisance parameters (*θ* or α_p): associated to uncertainty in:

- response of the detector (calibration)
- phenomenological model of interaction in non-perturbative regime

Lead to a parametrized model: $\nu \to \nu(\boldsymbol{\alpha}), f(x) \to f(x|\boldsymbol{\alpha})$

$$\mathbf{f}(\mathcal{D}|\boldsymbol{\alpha}) = \mathrm{Pois}(n|\nu(\boldsymbol{\alpha})) \prod_{e=1}^{n} f(x_e|\boldsymbol{\alpha})$$

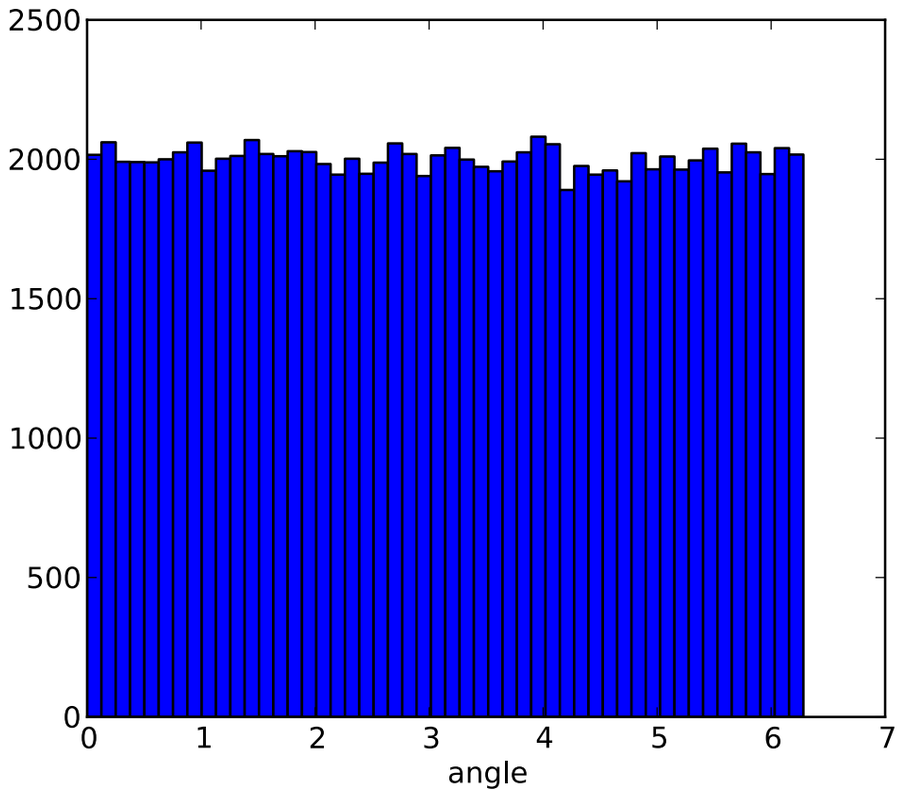
TRANSFORMATION PROPERTIES:

PDF VS. LIKELIHOOD

CHANGE OF VARIABLES

What happens with x→ cos(x)

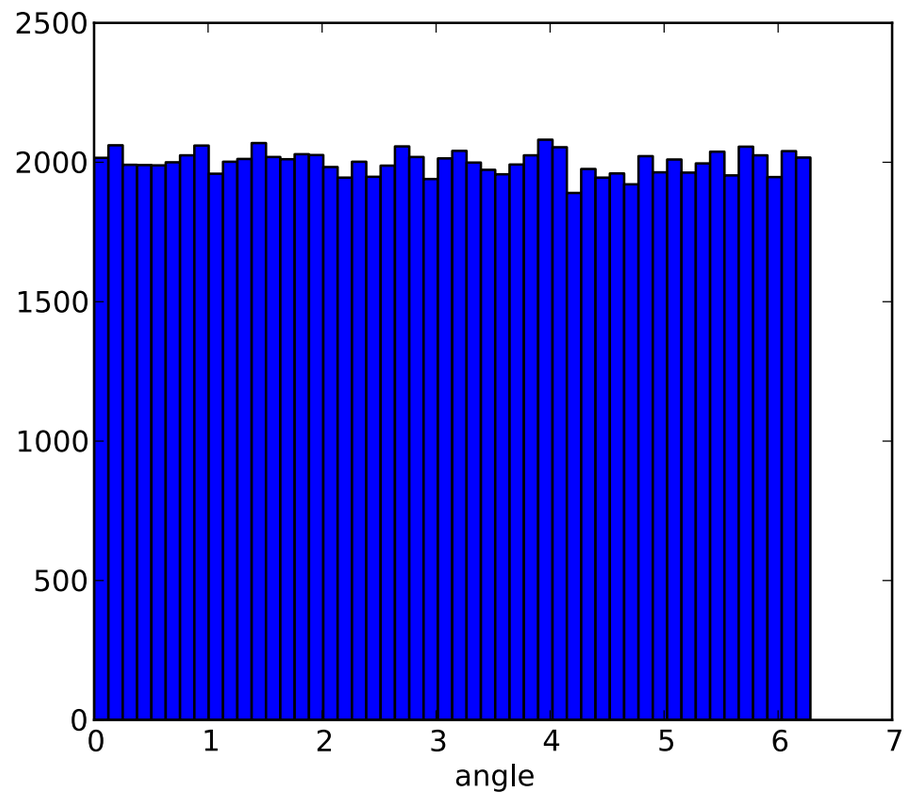
```python
import numpy as np
import matplotlib.pyplot as plt

N_MC=100000  # number of Monte Carlo Experiments
nBins = 50 # number of bins for Histograms

data_x, data_y = [],[]  #lists that will hold x and y

# do experiments
for i in range(N_MC):
    # generate observation for x
    x = np.random.uniform(0,2*np.pi)

    y = np.cos(x)
    data_x.append(x)
    data_y.append(y)

#setup figures
fig = plt.figure(figsize=(13,5))
fig_x = fig.add_subplot(1,2,1)
fig_y = fig.add_subplot(1,2,2)

fig_x.hist(data_x,nBins)
fig_x.set_xlabel('angle')

fig_y.hist(data_y,nBins)
fig_y.set_xlabel('cos(angle)')

plt.show()
```

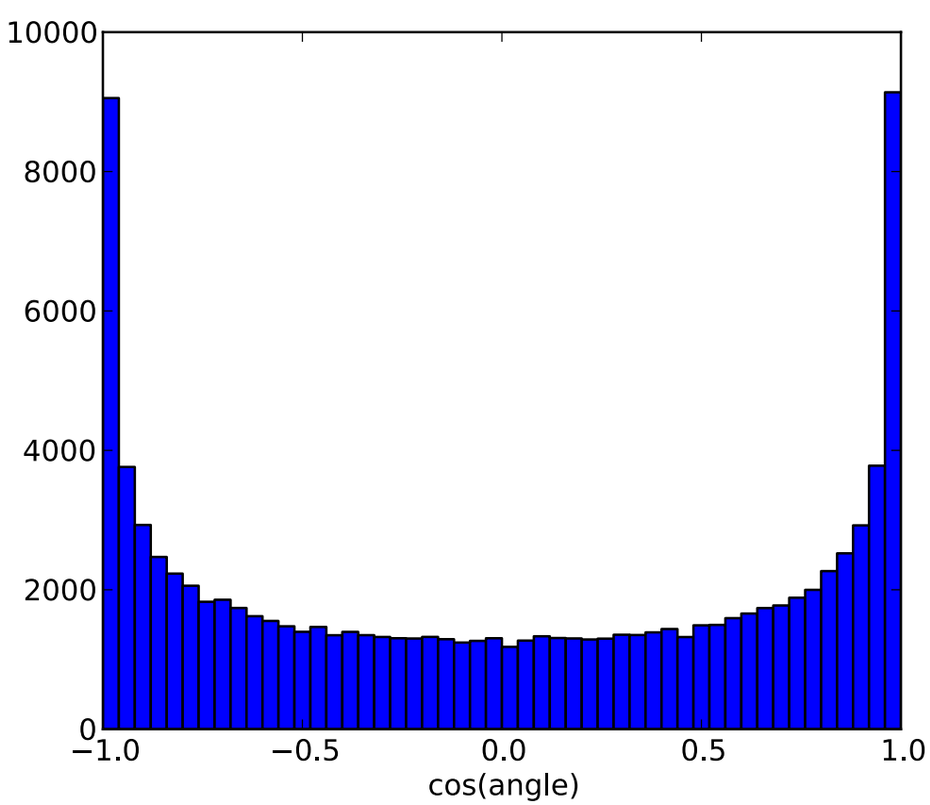


CHANGE OF VARIABLES

What happens with x→ cos(x)

```
1   import numpy as np
2   import matplotlib.pyplot as plt
3
4   N_MC=100000  # number of Monte Carlo Experiments
5   nBins = 50 # number of bins for Histograms
6
7   data_x, data_y = [],[]  #lists that will hold x and y
8
9   # do experiments
10  for i in range(N_MC):
11      # generate observation for x
12      x = np.random.uniform(0,2*np.pi)
13
14      y = np.cos(x)
15      data_x.append(x)
16      data_y.append(y)
17
18  #setup figures
19  fig = plt.figure(figsize=(13,5))
20  fig_x = fig.add_subplot(1,2,1)
21  fig_y = fig.add_subplot(1,2,2)
22
23  fig_x.hist(data_x,nBins)
24  fig_x.set_xlabel('angle')
25
26  fig_y.hist(data_y,nBins)
27  fig_y.set_xlabel('cos(angle)')
28
29  plt.show()
```





CHANGE OF VARIABLES

If *f(x)* is the pdf for *x* and *y(x)* is a change of variables, then the pdf *g(y)* must satisfy

$$P(x_a < x < x_b) \equiv \int_{x_a}^{x_b} f(x)dx = \int_{y(x_a)}^{y(x_b)} g(y)dy \equiv P(y(x_a) < y < y(x_b))$$

We can rewrite the integral on the right

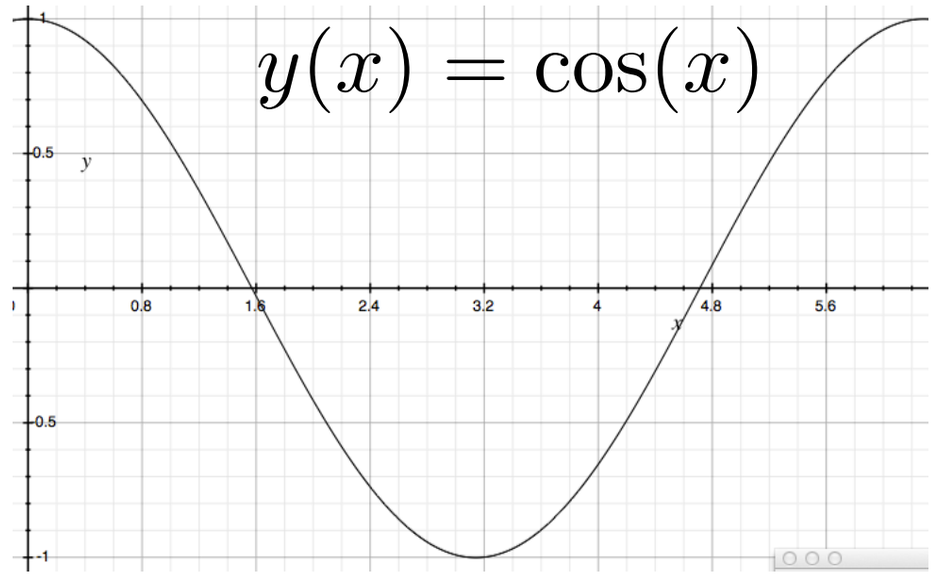
$$\int_{y(x_a)}^{y(x_b)} g(y)dy = \int_{x_a}^{x_b} g(y(x)) \left|\frac{dy}{dx}\right| dx$$

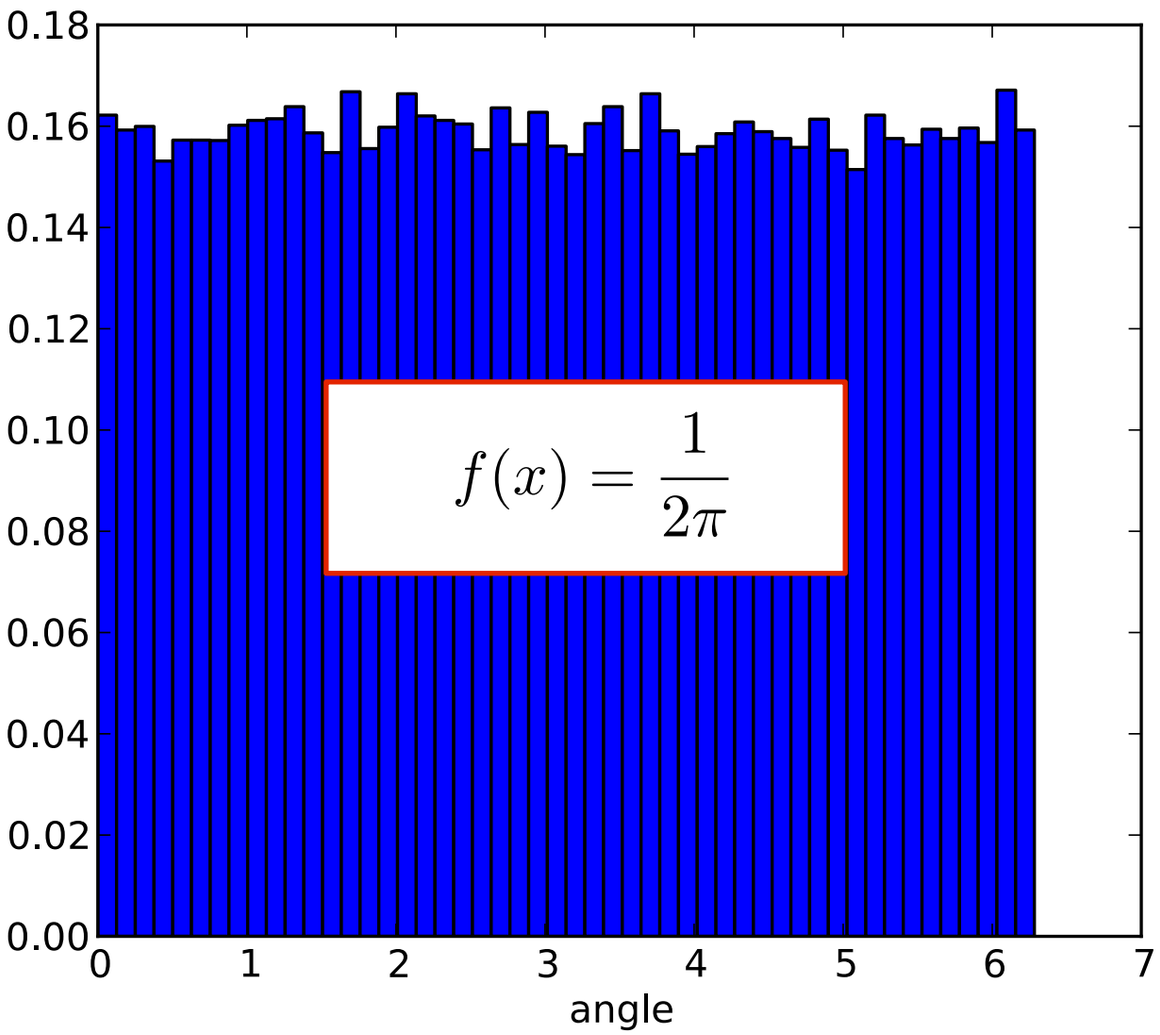
therefore, the two pdfs are related by a Jacobian factor

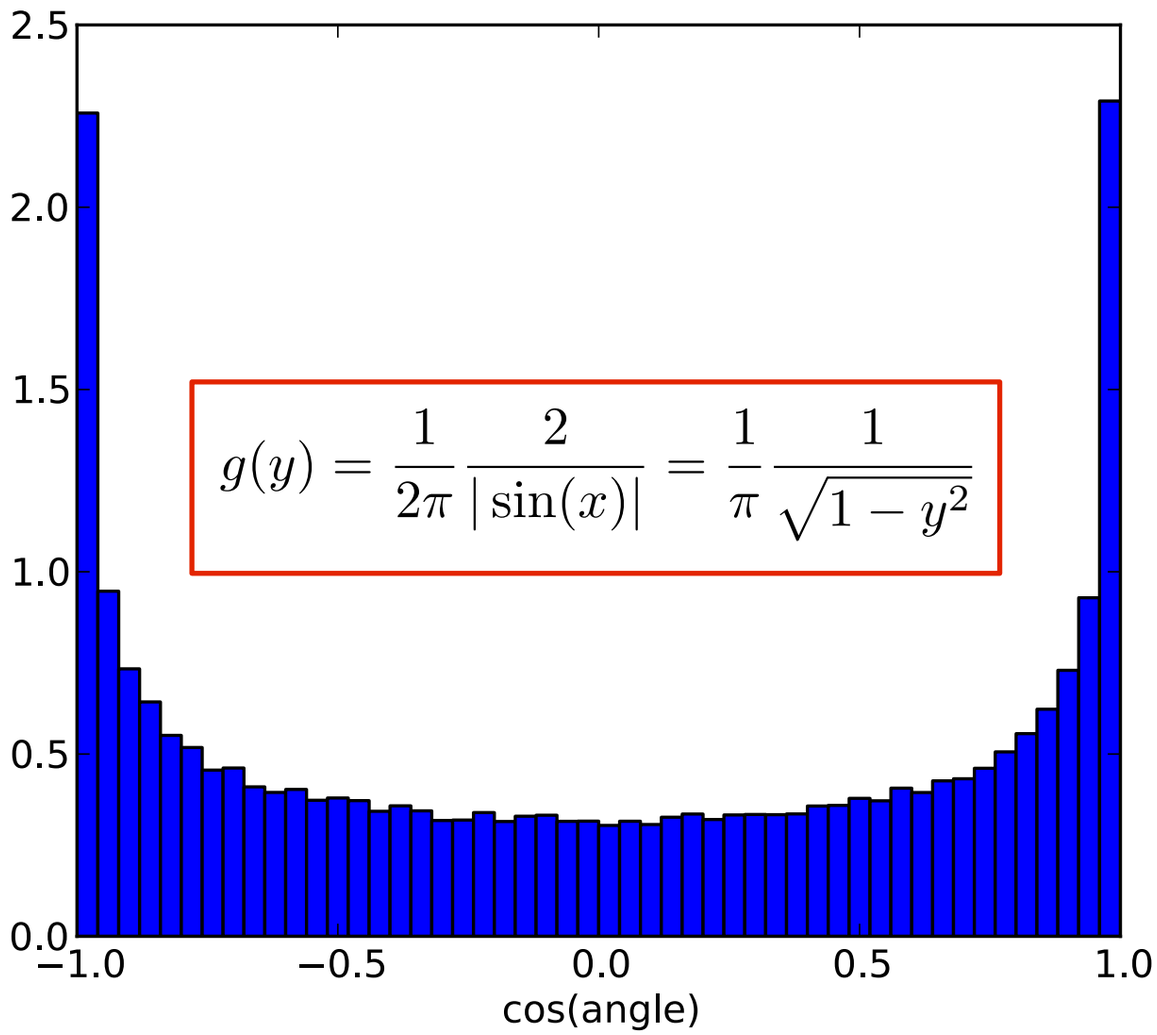
$$f(x) = g(y) \left|\frac{dy}{dx}\right|$$

An example

$$f(x) = g(y)\left|\frac{dy}{dx}\right|$$

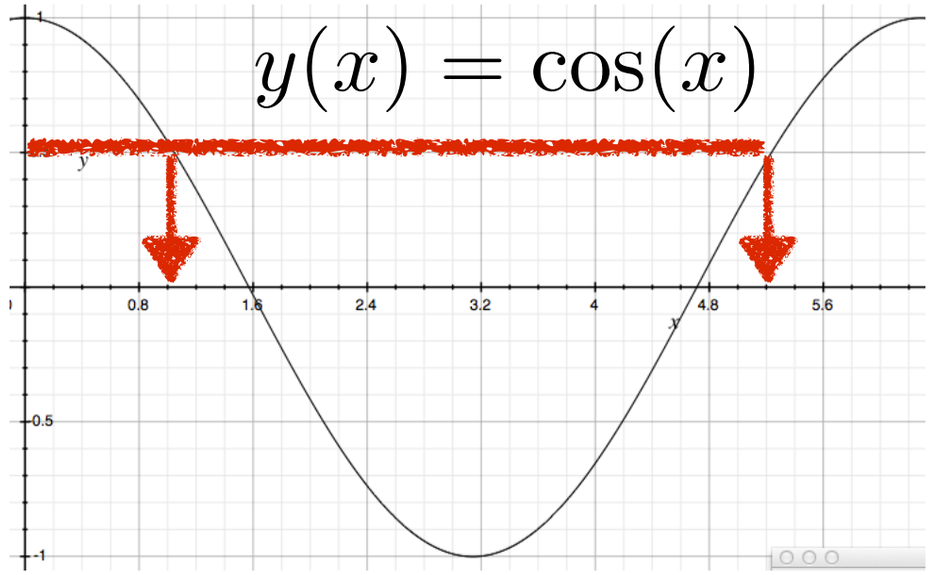


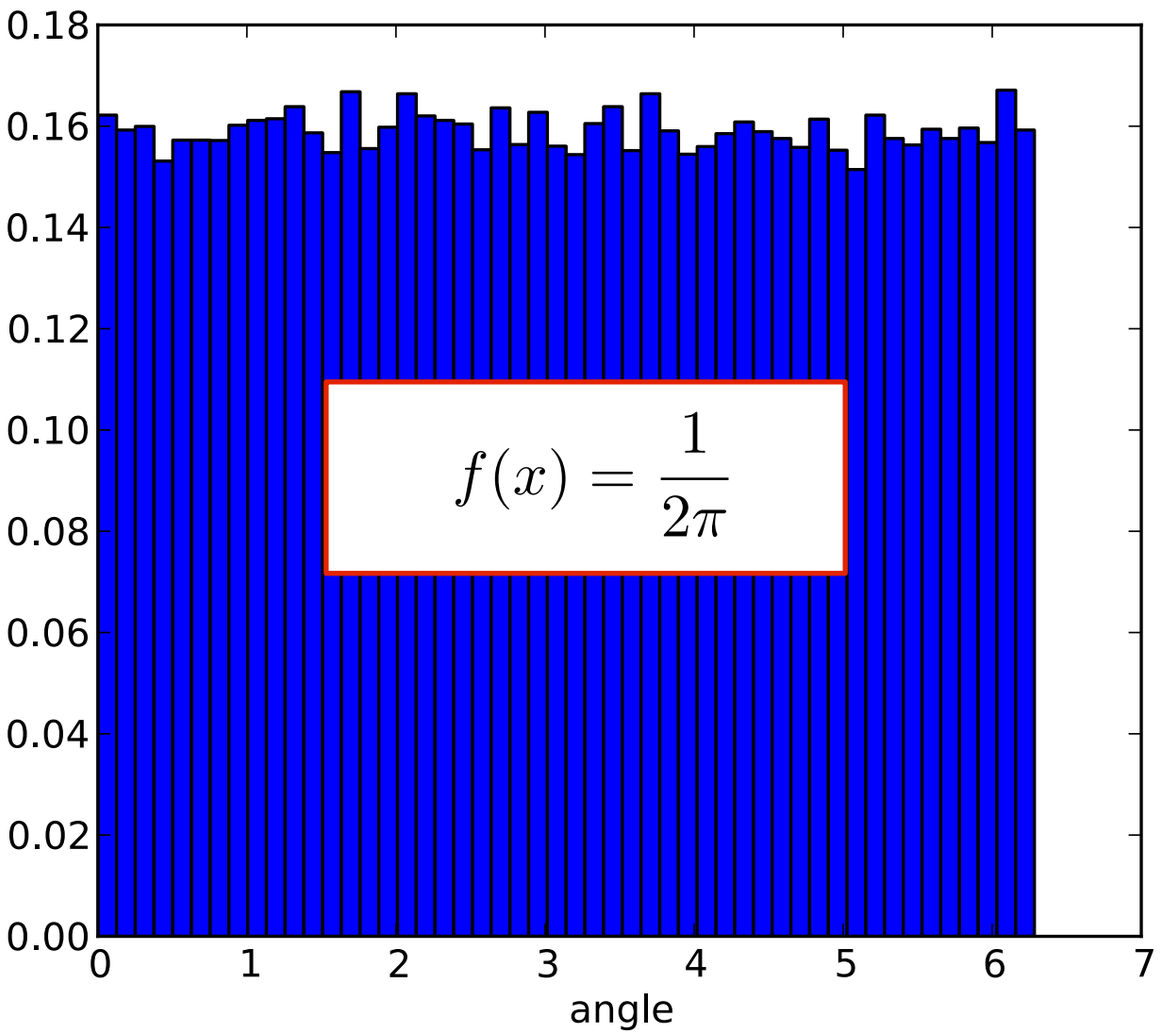


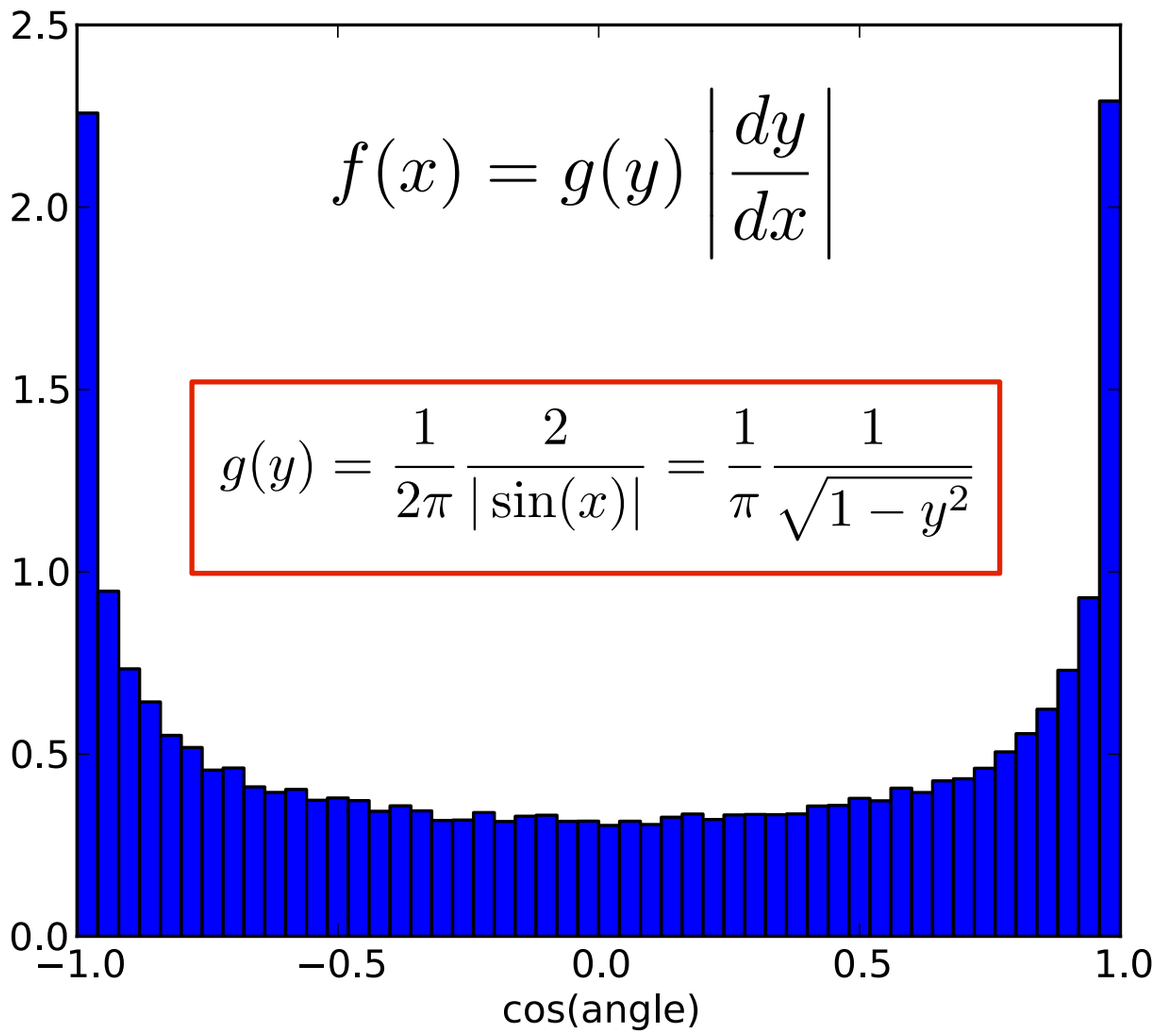


An example

I am glossing over the fact that the map is not 1-to-1. Different values of x, map into same value of y. We will need to sum/integrate over them. Here it is easy, but in general this may become intractable… need inverse map







Change of variable x, change of parameter θ

- **For pdf p(x|θ) and change of variable from x to y(x):**

  p(y(x)|θ) = p(x|θ) / |dy/dx|.

  Jacobian modifies probability *density*, guaranties that

  P($y(x_1) < y < y(x_2)$) = P($x_1 < x < x_2$), i.e., that

  ***Probabilities* are invariant under change of variable x.**

  - **Mode of probability *density* is *not* invariant (so, e.g., criterion of maximum probability density is ill-defined).**
  - **Likelihood *ratio* is invariant under change of variable x. (Jacobian in denominator cancels that in numerator).**

- **For likelihood $\mathcal{L}(\theta)$ and reparametrization from θ to u(θ):**

  $\mathcal{L}(\theta) = \mathcal{L}(u(\theta))$ (!).

  - **Likelihood $\mathcal{L}(\theta)$ is invariant under reparametrization of parameter θ (reinforcing fact that $\mathcal{L}$ is *not* a pdf in θ).**

The Likelihood Function

Consider the Poisson distribution describes a discrete event count n for a real-valued mean μ.

$$Pois(n|\mu) = \mu^n \frac{e^{-\mu}}{n!}$$

The **likelihood** of μ given n is the same equation evaluated as a function of μ

- Now it's a continuous function
- But it is not a pdf!

$$L(\mu) = Pois(n|\mu)$$

Common to plot the -ln *L* (or -2 ln *L*)

- helps avoid thinking of it as a PDF
- connection to χ^2 distribution

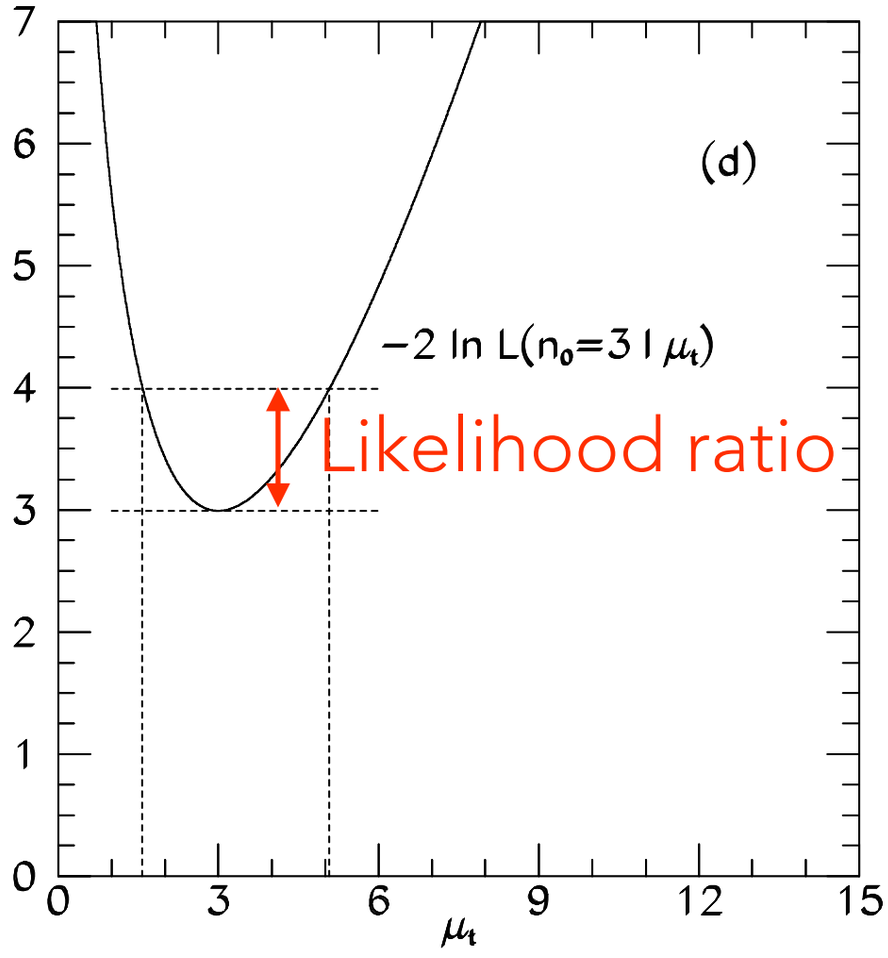


Figure from R. Cousins, Am. J. Phys. 63 398 (1995)

Probability Integral Transform

Consider a specific change of variables related to the cumulative for some arbitrary *f(x)*

$$y(x) = \int_{-\infty}^{x} f(x')dx'$$

Using our general change of variables formula:

$$f(x) = g(y)\left|\frac{dy}{dx}\right|$$

We find for this case the Jacobian factor is

$$\left|\frac{dy}{dx}\right| = f(x)$$

Thus $g(y) = 1$

Probability Integral Transform

"…seems likely to be one of the most fruitful conceptions introduced into statistical theory in the last few years"
– Egon Pearson (1938)

Given continuous x ∈ (a,b), and its pdf p(x), let

$$y(x) = \int_a^x p(x')\,dx' \ .$$

Then y ∈ (0,1) and p(y) = 1 (uniform) for all y. (!)

So there always exists a metric in which the pdf is uniform.

***Many* issues become more clear (or trivial) after this transformation*. (If x is discrete, some complications.)**

The specification of a Bayesian prior pdf p(μ) for parameter μ is equivalent to the choice of the metric f(μ) in which the pdf is uniform. This is a *deep* issue, not always recognized as such by users of flat prior pdf's in HEP!

***And the inverse transformation provides for efficient M.C. generation of p(x) starting from RAN().**

BAYES THEOREM

Bayes' Theorem

Bayes' theorem relates the conditional and marginal probabilities of events A & B

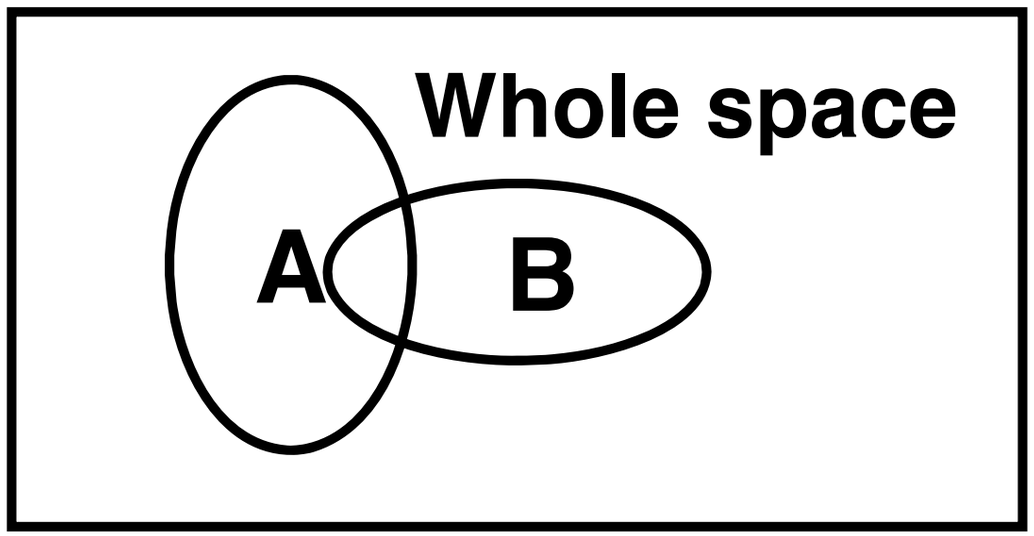
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

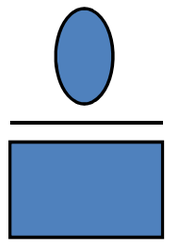
- **P(*A*)** is the prior probability. It is "prior" in the sense that it does not take into account any information about *B*.
- **P(*A* | *B*)** is the conditional probability of *A*, given *B*. It is also called the posterior probability because it is derived from or depends upon the specified value of *B*.
- **P(*B* | *A*)** is the conditional probability of *B* given *A*.
- **P(*B*)** is the prior or marginal probability of *B*, and acts as a normalizing constant.

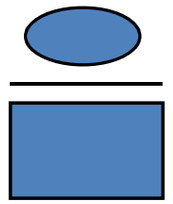
$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\mathcal{N}} \propto L(\theta)\pi(\theta)$$

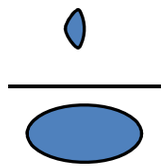
… IN PICTURES (FROM BOB COUSINS)

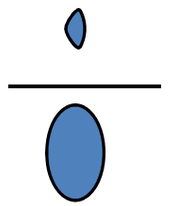
P, Conditional P, and Derivation of Bayes' Theorem in Pictures

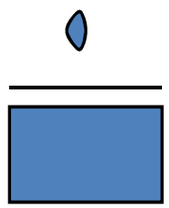


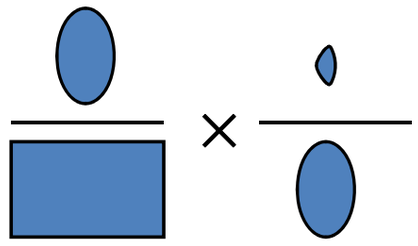
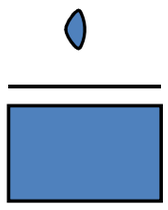
$P(A) =$ 

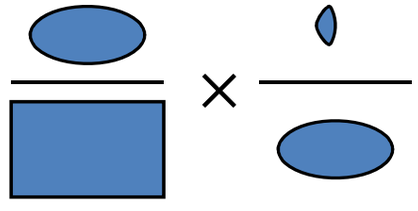
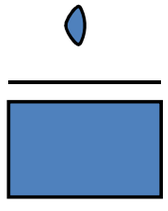
$P(B) =$ 

$P(A|B) =$ 

$P(B|A) =$ 

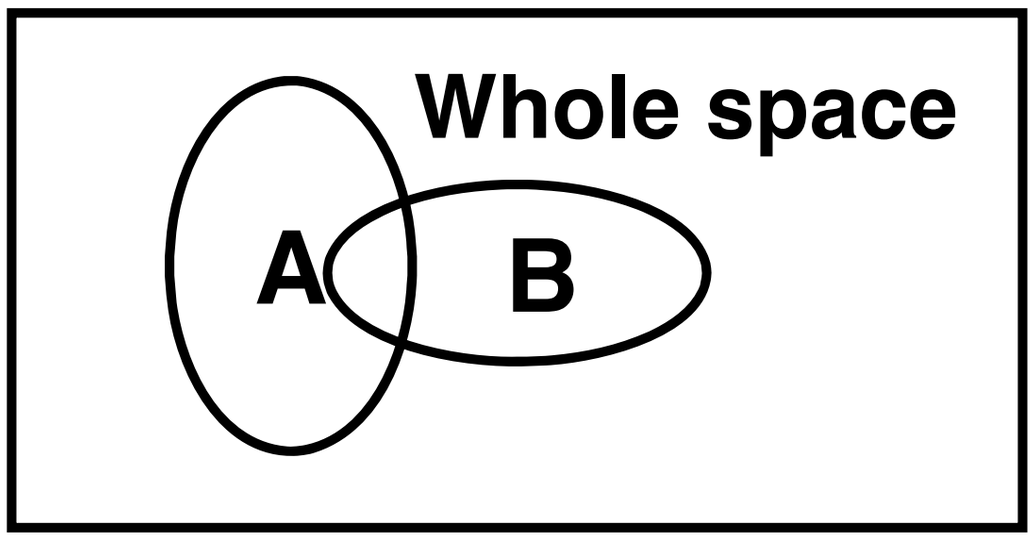
$P(A \cap B) =$ 

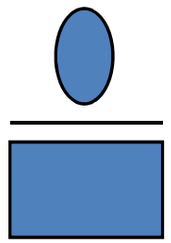
$P(A) \times P(B|A) =$  $=$  $= P(A \cap B)$

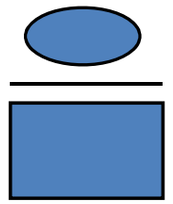
$P(B) \times P(A|B) =$  $=$  $= P(A \cap B)$

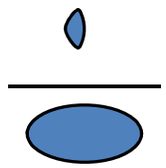
$\Rightarrow P(B|A) = P(A|B) \times P(B) / P(A)$

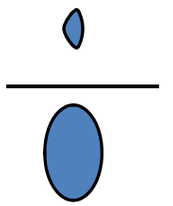
P, Conditional P, and Derivation of Bayes' Theorem in Pictures

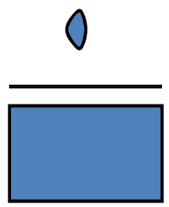


$P(A) =$ 

$P(B) =$ 

$P(A|B) =$ 

$P(B|A) =$ 

$P(A \cap B) =$ 

Don't forget about "Whole space" Ω I will drop it from the notation typically, but occasionally it is important.

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

Likelihood vs. Posterior

$$P(\text{Data};\text{Theory}) \neq P(\text{Theory};\text{Data})$$

Theory = Professional Basketball Player or not

Data = Short or Tall

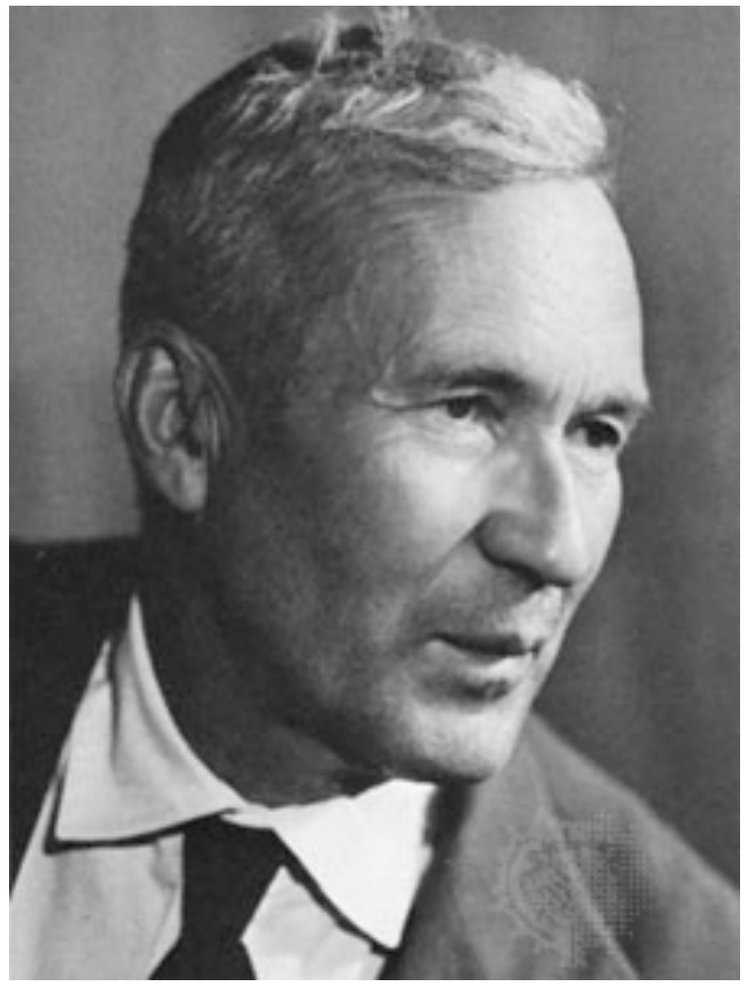
p(Tall | Professional Basketball Player) ~ 1

p(Professional Basketball Player | Tall) << 1

Axioms of Probability

These Axioms are a mathematical starting point for probability and statistics

1. probability for every element, E, is non-negative $P(E) \geq 0 \qquad \forall E \subseteq \mathcal{F} = 2^{\Omega}$
2. probability for the entire space of possibilities is 1 $P(\Omega) = 1.$
3. if elements E_i are disjoint, probability is additive $P(E_1 \cup E_2 \cup \cdots) = \sum_i P(E_i).$



Kolmogorov axioms (1933)

Consequences:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(\Omega \setminus E) = 1 - P(E)$$

Different definitions of Probability

Frequentist

- defined as limit of long term frequency
- probability of rolling a 3 := limit of (# rolls with 3 / # trials)
  - you don't need an infinite sample for definition to be useful
  - sometimes ensemble doesn't exist
    - eg. P(Higgs mass = 125 GeV), P(it will snow tomorrow)
- Intuitive if you are familiar with Monte Carlo methods
- compatible with orthodox interpretation of probability in Quantum Mechanics.

  Probability to measure spin projected on x-axis if spin of beam is polarized along +z

$$|\langle \rightarrow | \uparrow \rangle|^2 = \frac{1}{2}$$

Subjective Bayesian

- Probability is a degree of belief (personal, subjective)
  - can be made quantitative based on betting odds
  - most people's subjective probabilities are not ***coherent*** and do not obey laws of probability

http://plato.stanford.edu/archives/sum2003/entries/probability-interpret/#3.1

COMPUTATIONAL TOOLS FOR BAYESIAN ANALYSIS

Challenge of Bayesian inference is that one often needs to marginalize / integrate in high dimensions

- Eg. To get the normalizing constant or to get the marginal distribution for specific quantities
- This is almost never done with an explicit numerical integral

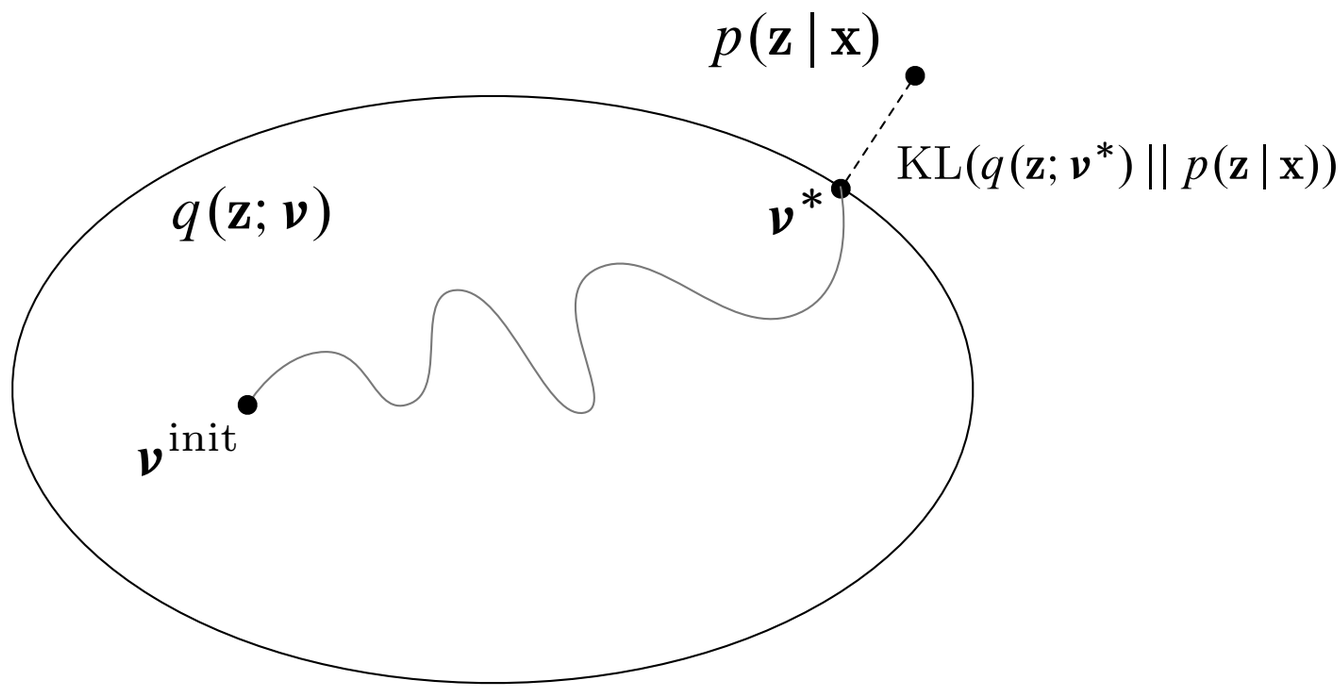
Markov Chain Monte Carlo

- A procedure to explore the parameter space that produces samples according to the posterior (eg. You can histogram the posterior samples).
- Many variants, including Hamiltonian Monte Carlo that scales well to high dimensions and posteriors that may be very narrow in some directions

Variational Inference

- alternatively, you can make a variational ansatz $q_{\varphi}(\theta)$ for the posterior $\pi(\theta|x)$ and try to optimize the parameters φ to minimize "distance" between the two
- Usually the Kullback-Leibler divergent $KL[q_{\varphi}(\theta) \,||\, \pi(\theta|x)]$ (which is like the free energy in stat mech). One can use stochastic gradient descent as in ML to optimize φ
- Relies on optimization instead of integration

Variational Inference: Foundations and Modern Methods



VI approximates difficult quantities from complex models.

With **stochastic optimization** we can

- scale up VI to massive data
- enable VI on a wide class of difficult models
- enable VI with elaborate and flexible families of approximations

Variational Inference:
Foundations and Modern Methods

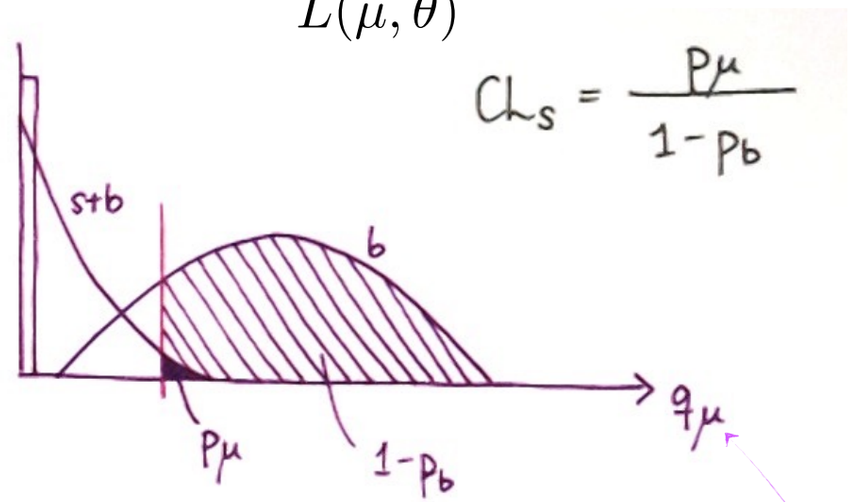
David Blei, Rajesh Ranganath, Shakir Mohamed

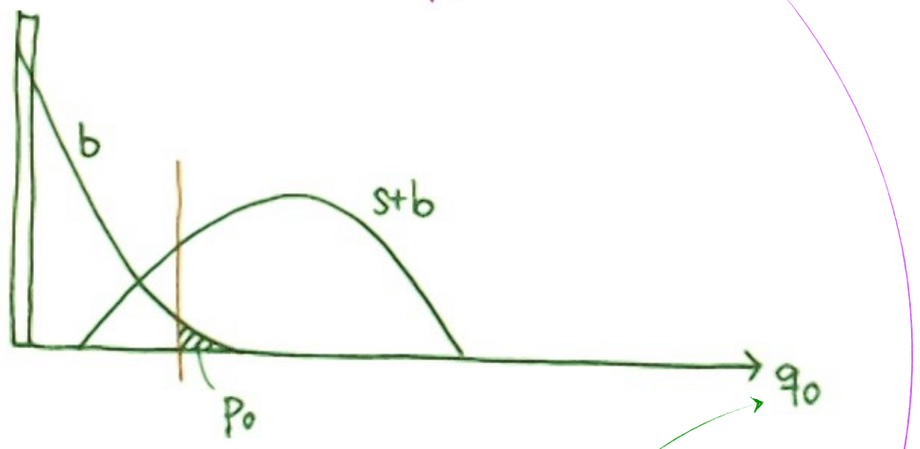
NIPS 2016 Tutorial · December 5, 2016

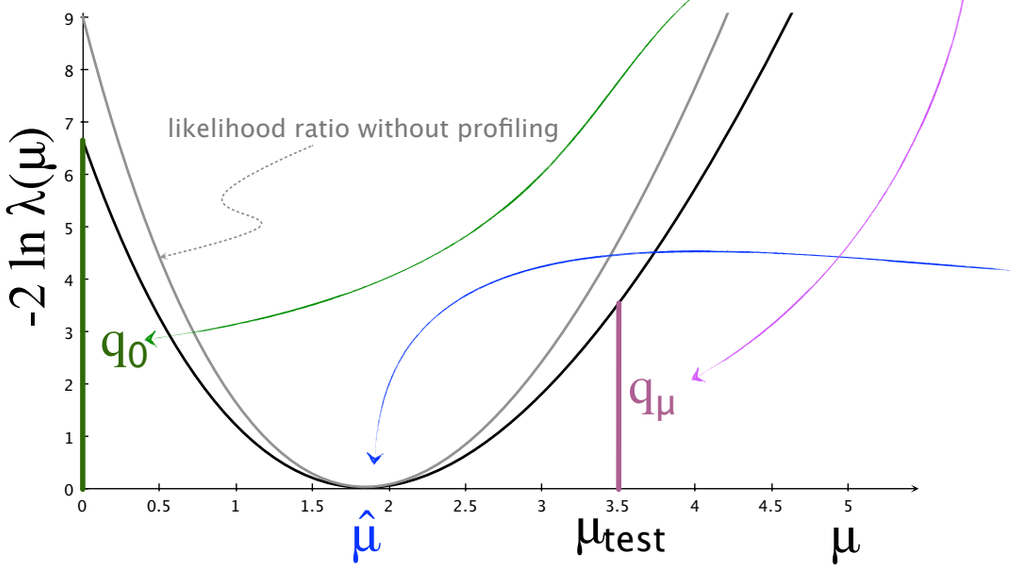
COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK
PRINCETON UNIVERSITY
DeepMind

THUMBNAIL OF THE STATISTICAL PROCEDURE

$$\lambda(\mu) = \frac{L(\mu, \hat{\hat{\theta}}(\mu))}{L(\hat{\mu}, \hat{\theta})}$$





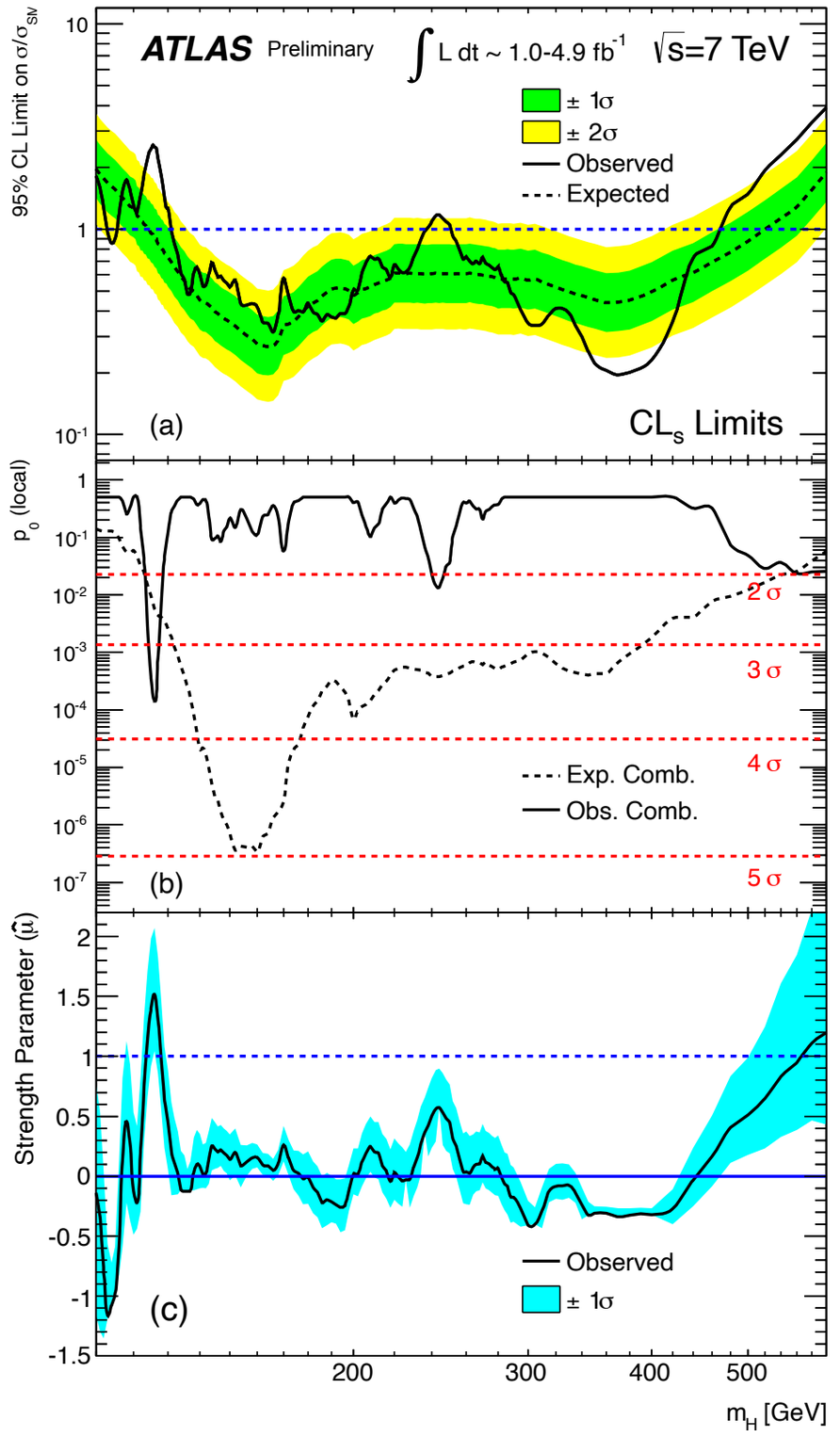


Follow LHC-HCG Combination Procedures

CL_s to test signal hypothesis

p_0 to test background hypothesis

$\hat{\mu}$ to estimate signal strength



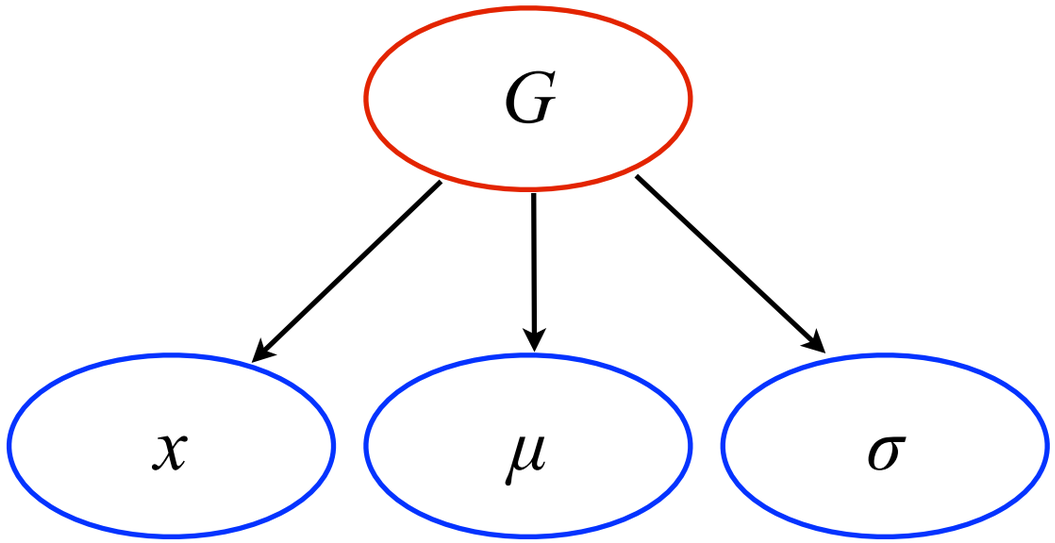
Extras

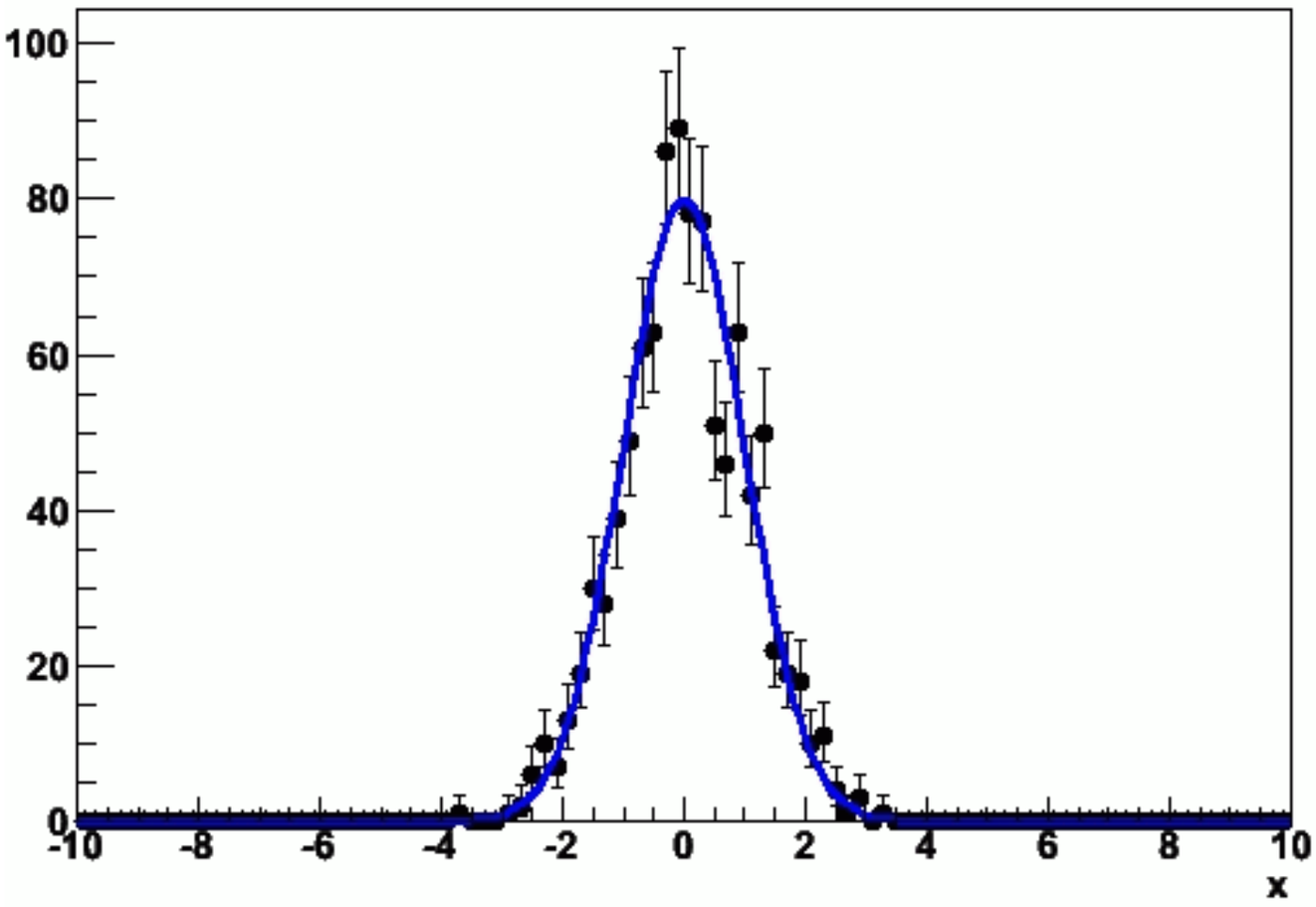
A GENERAL PURPOSE STATISTICAL MODEL

VISUALIZING PROBABILITY MODELS

I will represent PDFs graphically as below (directed acyclic graph)

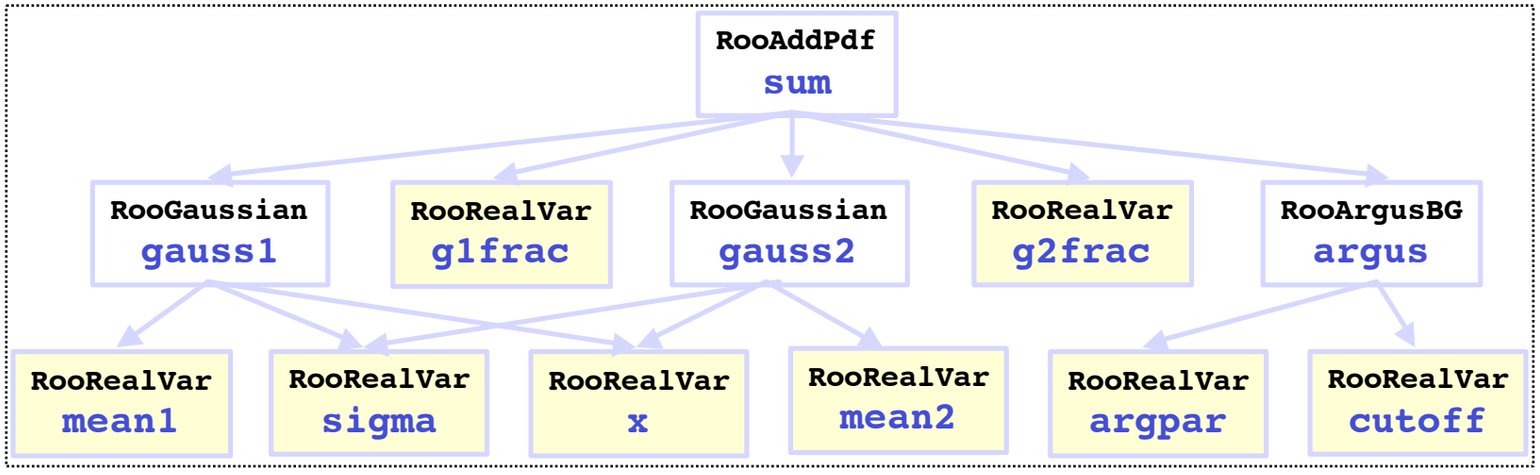
- eg. a Gaussian $G(x|\mu,\sigma)$ is parametrized by (μ,σ)
- every node is a real-valued function of the nodes below



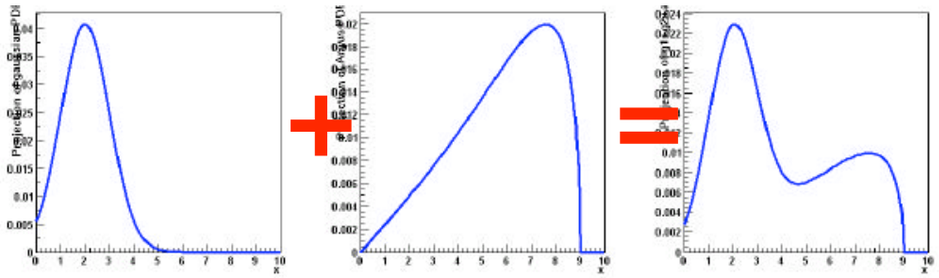


RooFit: A data modeling toolkit

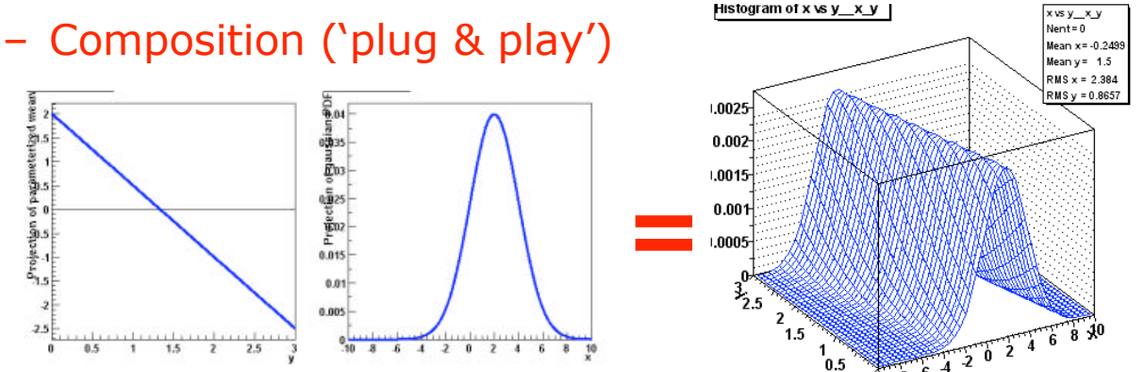
RooFit is a major tool developed at BaBar for data modeling.
RooStats provides higher-level statistical tools based on these PDFs.



– Addition



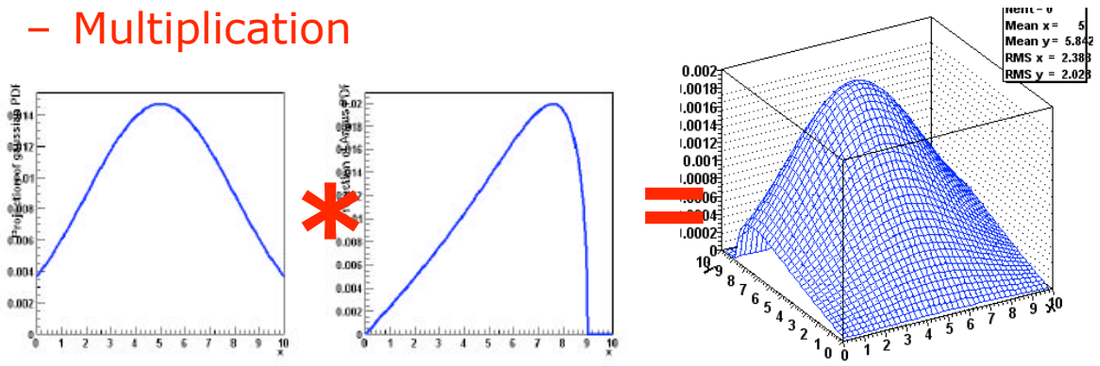
– Composition ('plug & play')



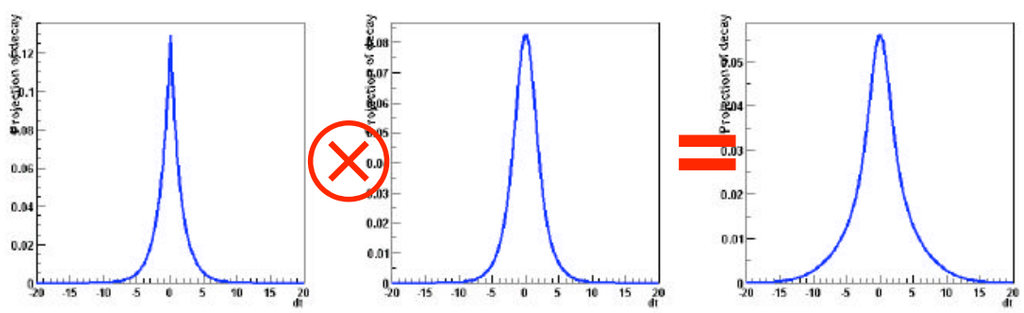
$m(y;a_0,a_1)$ $g(x;m,s)$ $g(x,y;a0,a1,s)$

Possible in *any* PDF
No explicit support in PDF code needed

– Multiplication



– Convolution



Marked Poisson Process

Channel: a subset of the data defined by some selection requirements.

- eg. all events with 4 electrons with energy > 10 GeV
- n: number of events observed in the channel
- ν: number of events expected in the channel

Discriminating variable: a property of those events that can be measured and which helps discriminate the signal from background

- eg. the invariant mass of two particles
- $f(x)$: the p.d.f. of the discriminating variable x

$$\mathcal{D} = \{x_1, \ldots, x_n\}$$

Marked Poisson Process / Extended Likelihood:

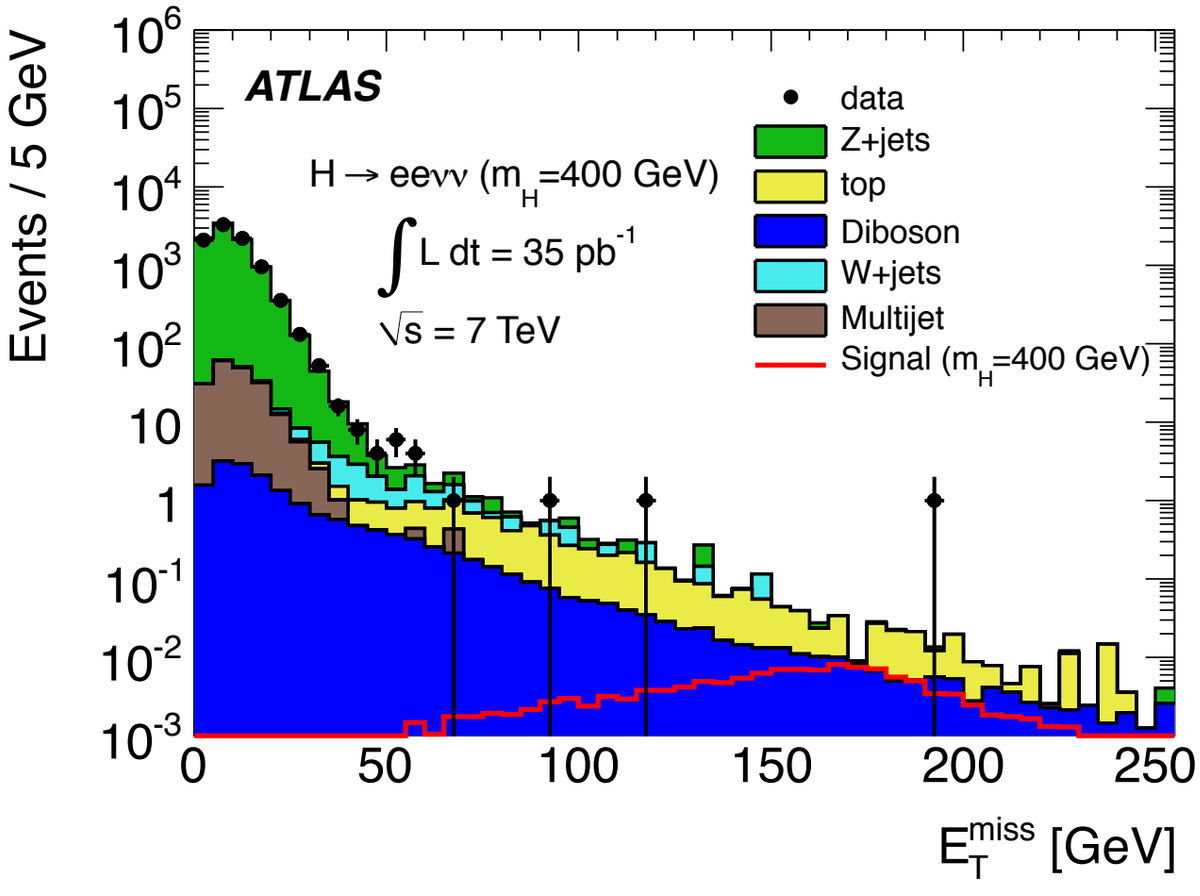
$$\mathbf{f}(\mathcal{D}|\nu) = \mathrm{Pois}(n|\nu) \prod_{e=1}^{n} f(x_e)$$

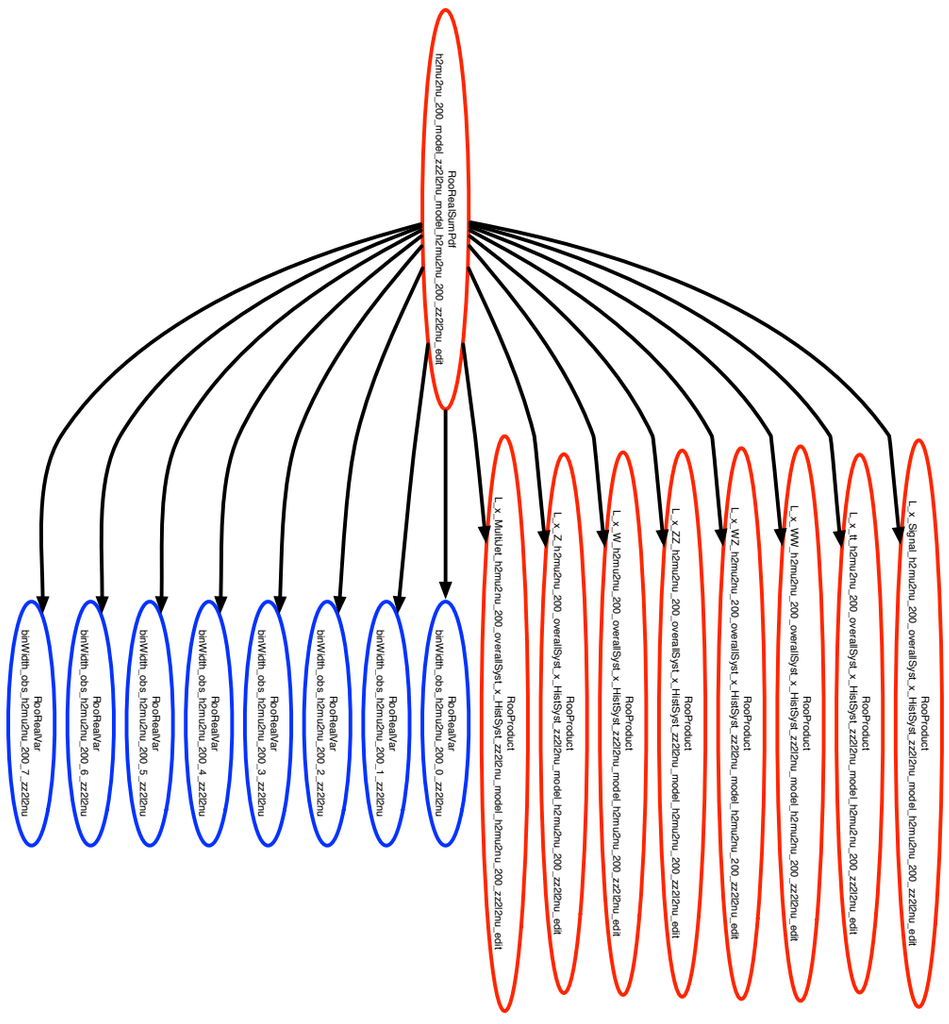
MIXTURE MODEL

Sample: a sample of simulated events corresponding to particular type interaction that populates the channel.

- statisticians call this a mixture model

$$f(x) = \frac{1}{\nu_{\text{tot}}} \sum_{s \in \text{samples}} \nu_s f_s(x) \ , \qquad \nu_{\text{tot}} = \sum_{s \in \text{samples}} \nu_s$$





PARAMETRIZING THE MODEL $\boldsymbol{\alpha} = (\mu, \boldsymbol{\theta})$

Parameters of interest (*μ*): parameters of the theory that modify the rates and shapes of the distributions, eg.

- the mass of a hypothesized particle
- the "signal strength" μ=0 no signal, μ=1 predicted signal rate

Nuisance parameters (*θ* or α_p): associated to uncertainty in:

- response of the detector (calibration)
- phenomenological model of interaction in non-perturbative regime

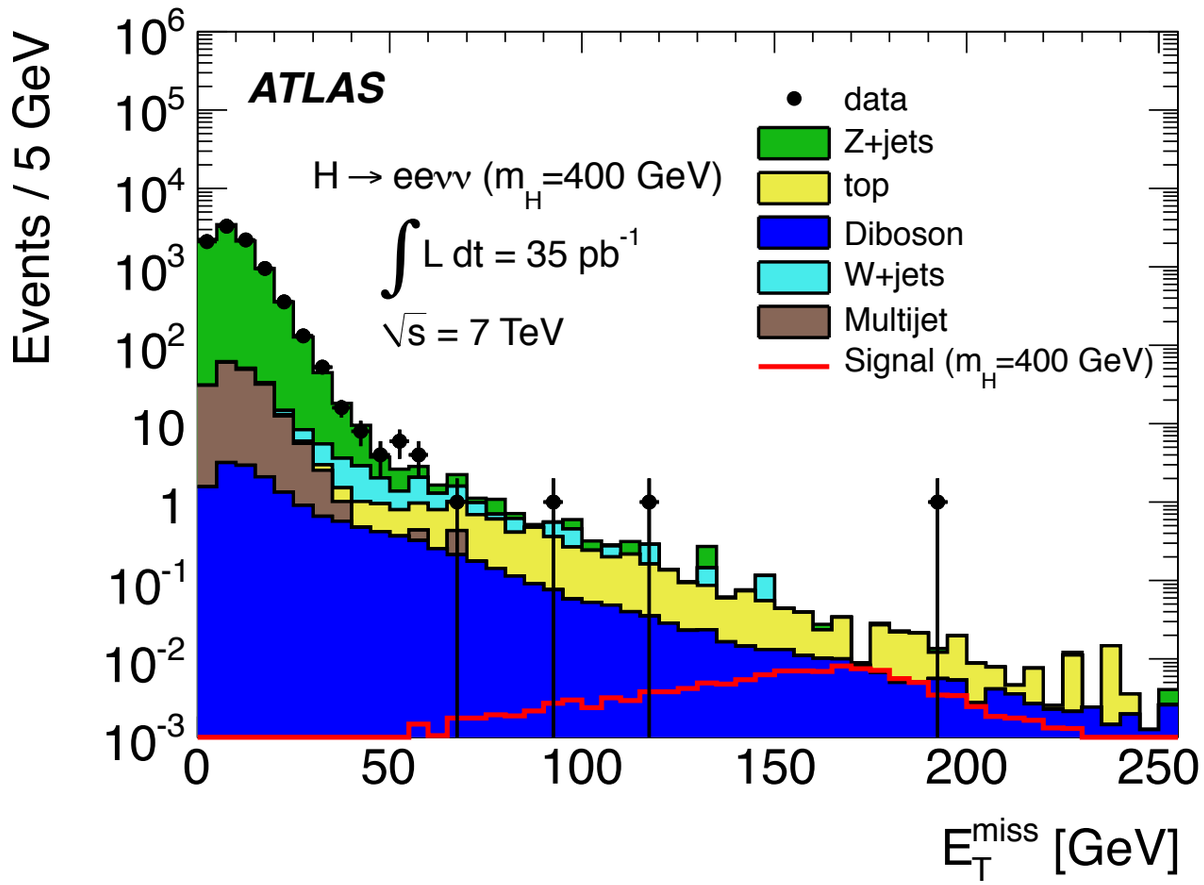
Lead to a parametrized model: $\nu \to \nu(\boldsymbol{\alpha}), f(x) \to f(x|\boldsymbol{\alpha})$

$$\mathbf{f}(\mathcal{D}|\boldsymbol{\alpha}) = \mathrm{Pois}(n|\nu(\boldsymbol{\alpha})) \prod_{e=1}^{n} f(x_e|\boldsymbol{\alpha})$$

Incorporating Systematic Effects

Tabulate effect of individual variations of sources of systematic uncertainty

- typically one at a time evaluated at nominal and "± 1 σ"
- use some form of interpolation to parametrize p^{th} variation in terms of **nuisance parameter** α_p



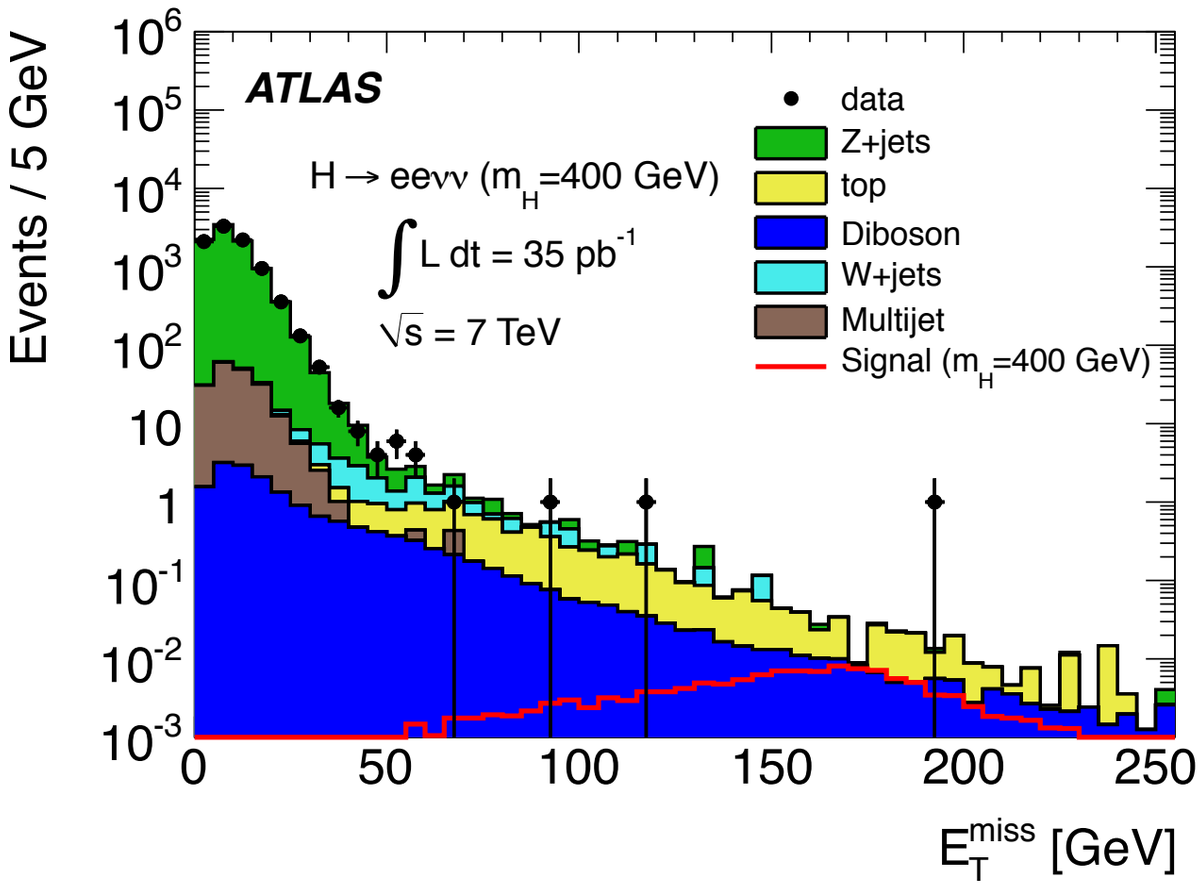
	Z+jets	top	Diboson	...
syst 1				
syst 2				
...				

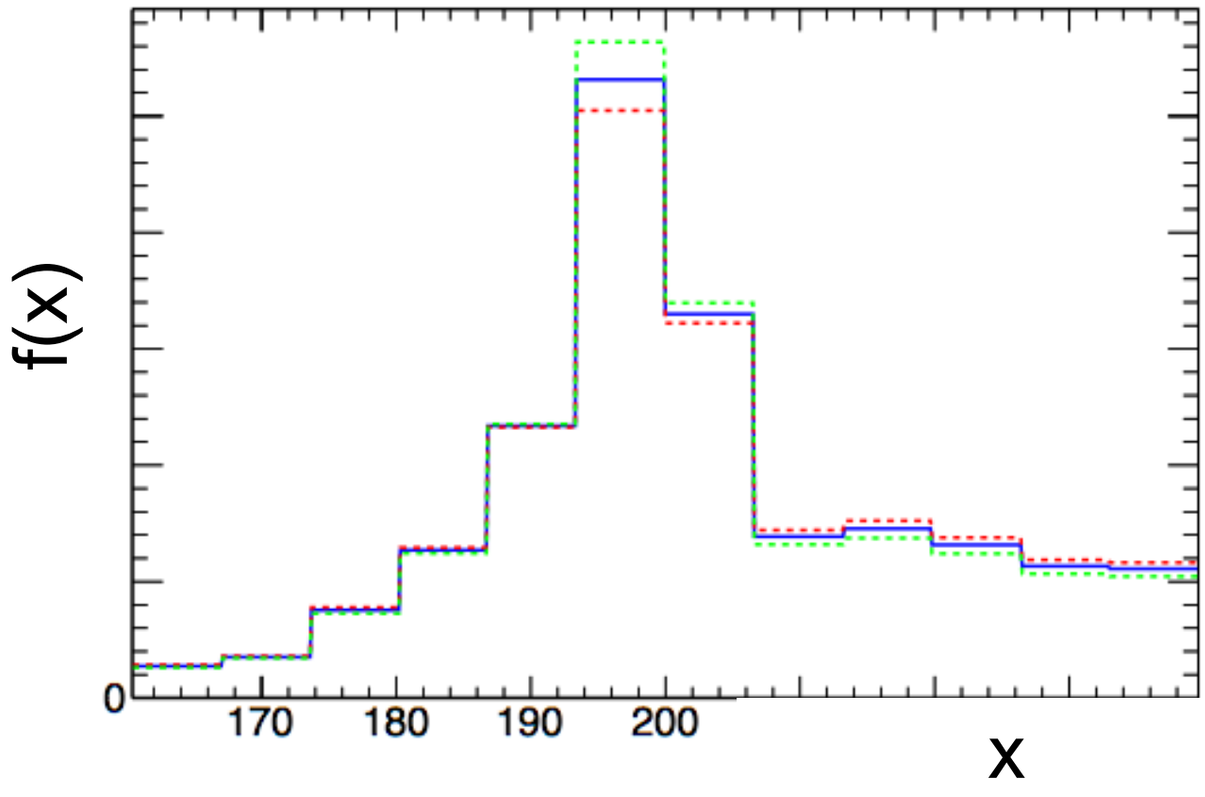
$$\mathbf{f}(\mathcal{D}|\boldsymbol{\alpha}) = \mathrm{Pois}(n|\nu(\boldsymbol{\alpha})) \prod_{e=1}^{n} f(x_e|\boldsymbol{\alpha})$$

Incorporating Systematic Effects

Tabulate effect of individual variations of sources of systematic uncertainty

- typically one at a time evaluated at nominal and "± 1 σ"
- use some form of interpolation to parametrize p^{th} variation in terms of **nuisance parameter** α_p



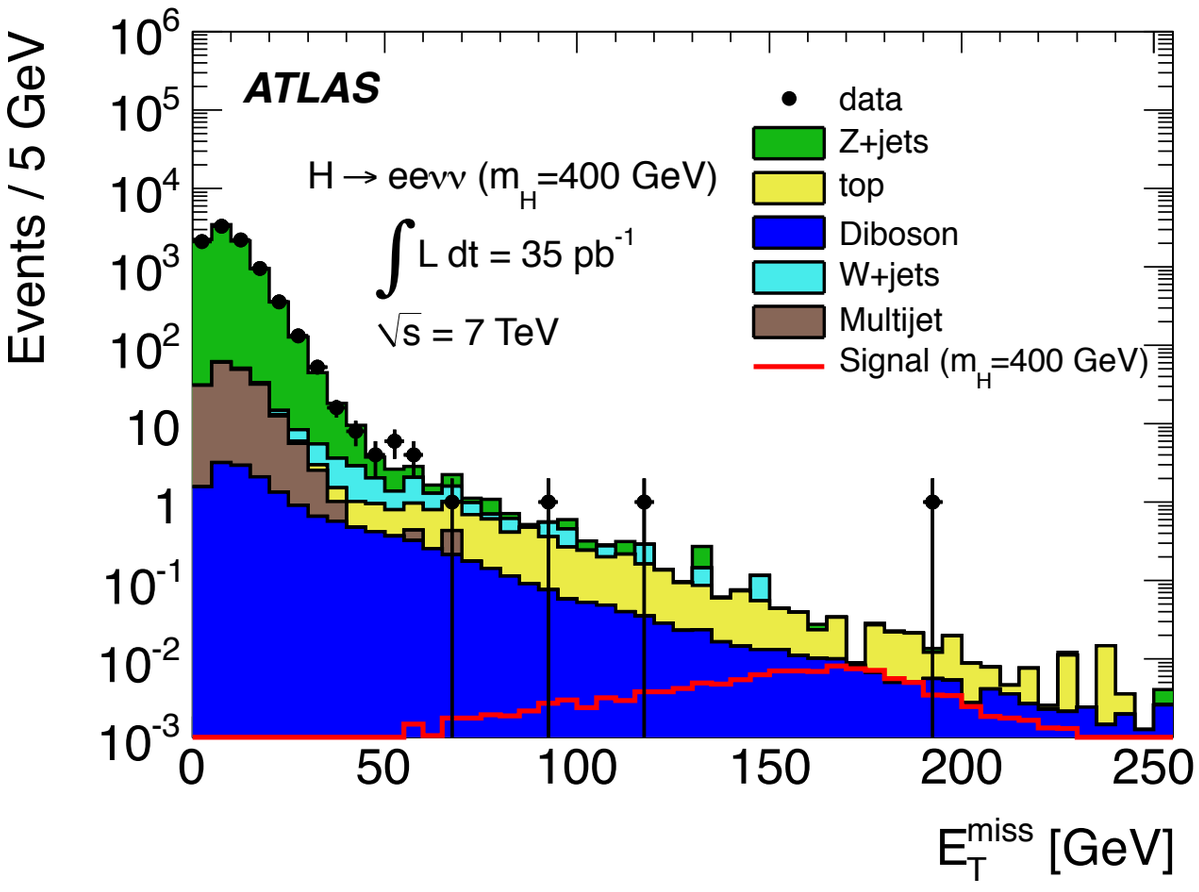


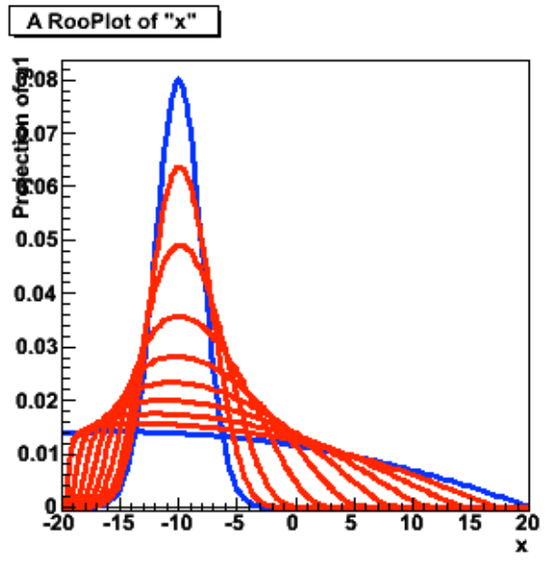
$$\mathbf{f}(\mathcal{D}|\boldsymbol{\alpha}) = \mathrm{Pois}(n|\nu(\boldsymbol{\alpha})) \prod_{e=1}^{n} f(x_e|\boldsymbol{\alpha})$$

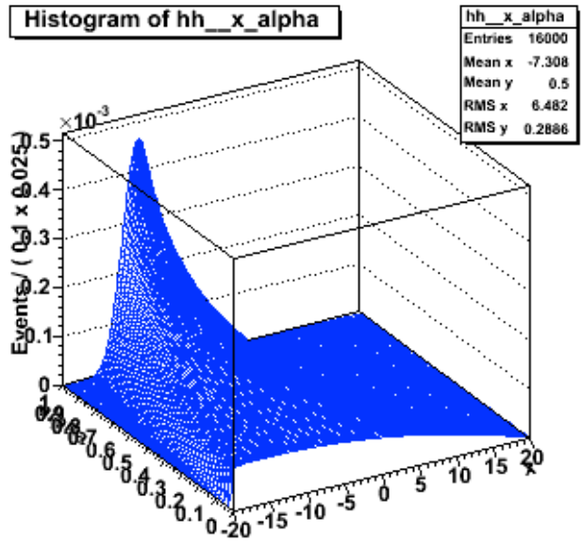
Incorporating Systematic Effects

Tabulate effect of individual variations of sources of systematic uncertainty

- typically one at a time evaluated at nominal and "± 1 σ"
- use some form of interpolation to parametrize p^{th} variation in terms of **nuisance parameter** α_p

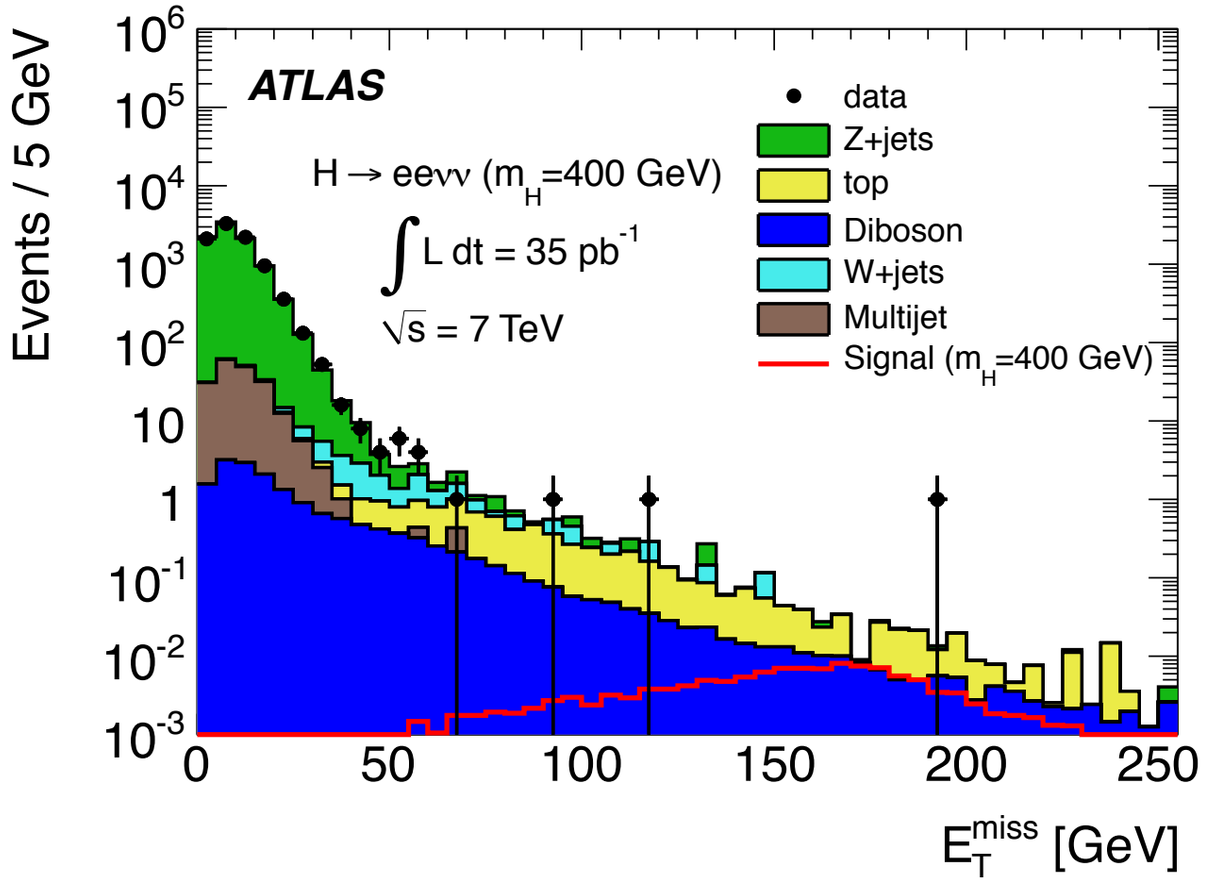


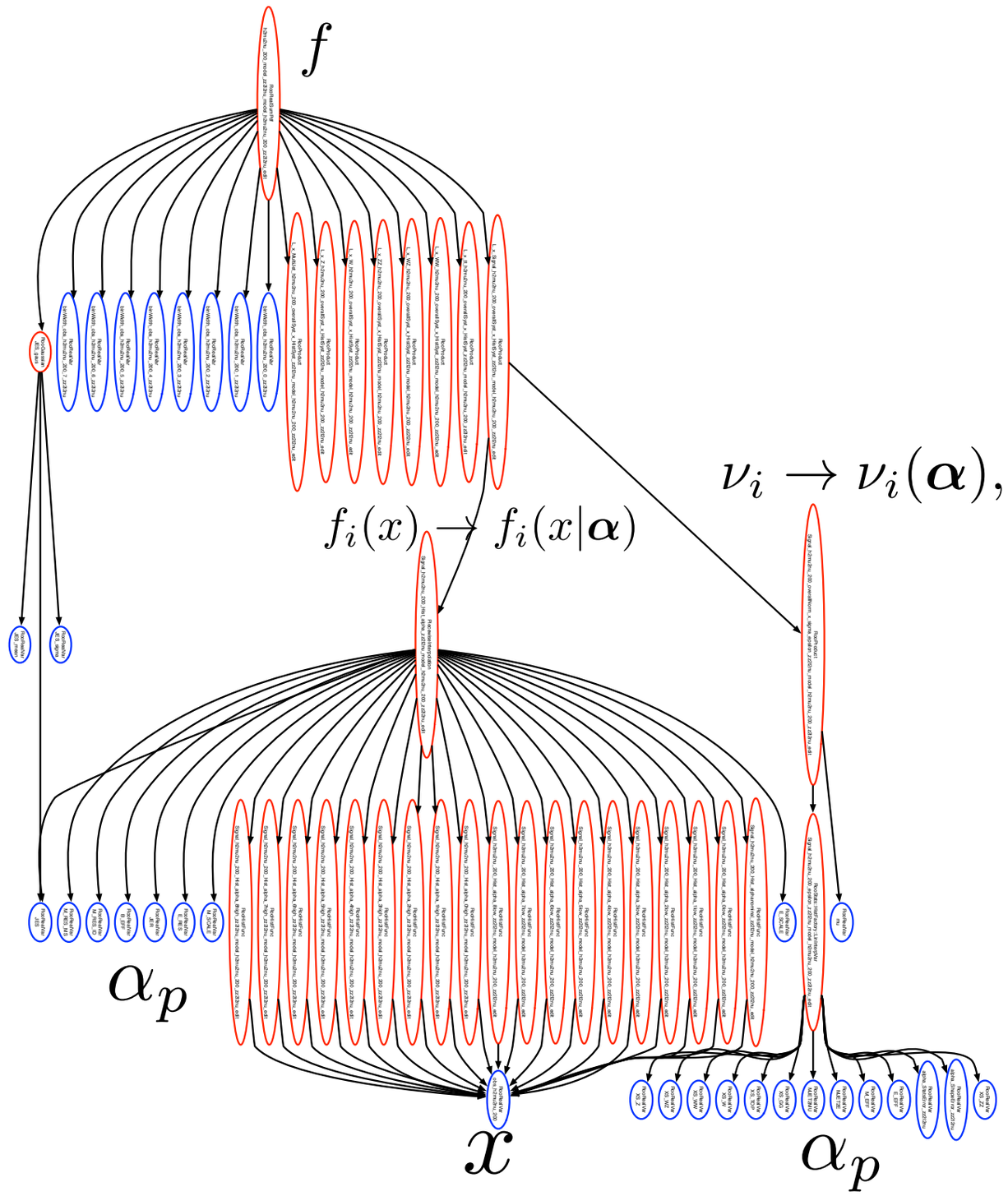




$$\mathbf{f}(\mathcal{D}|\boldsymbol{\alpha}) = \mathrm{Pois}(n|\nu(\boldsymbol{\alpha})) \prod_{e=1}^{n} f(x_e|\boldsymbol{\alpha})$$

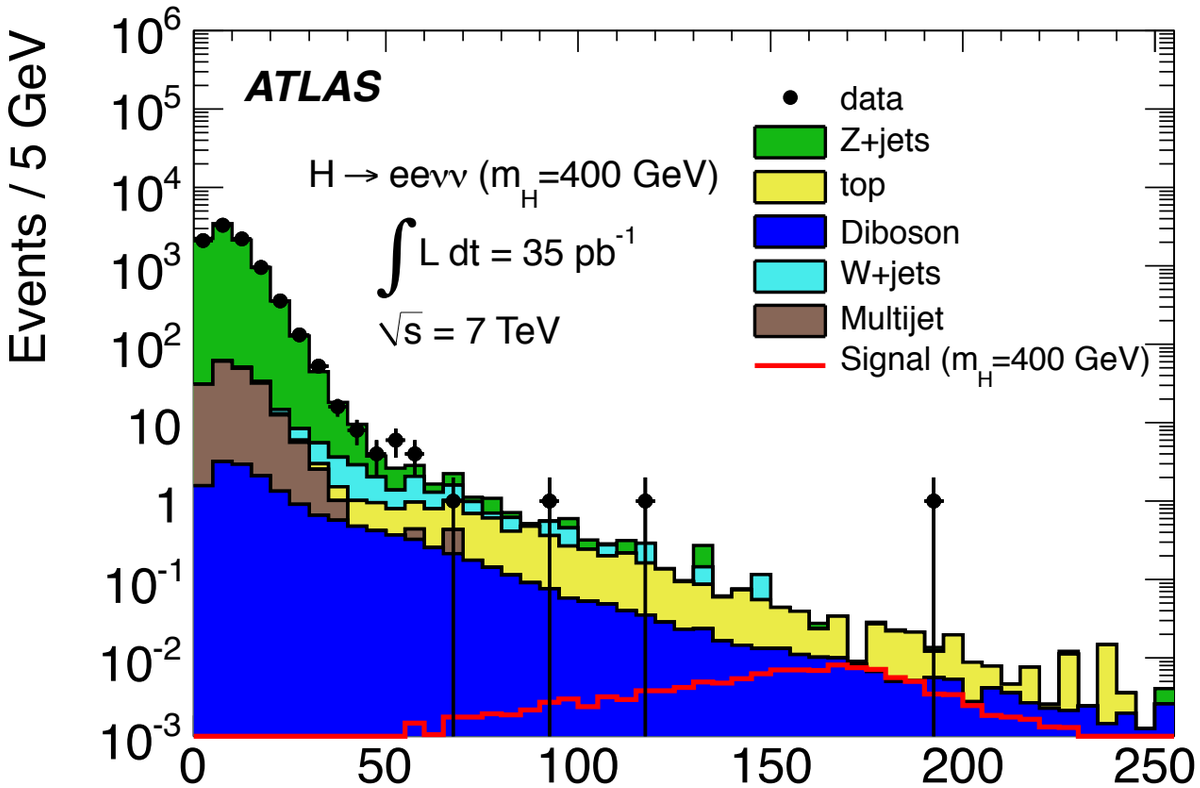
VISUALIZING THE MODEL FOR ONE CHANNEL

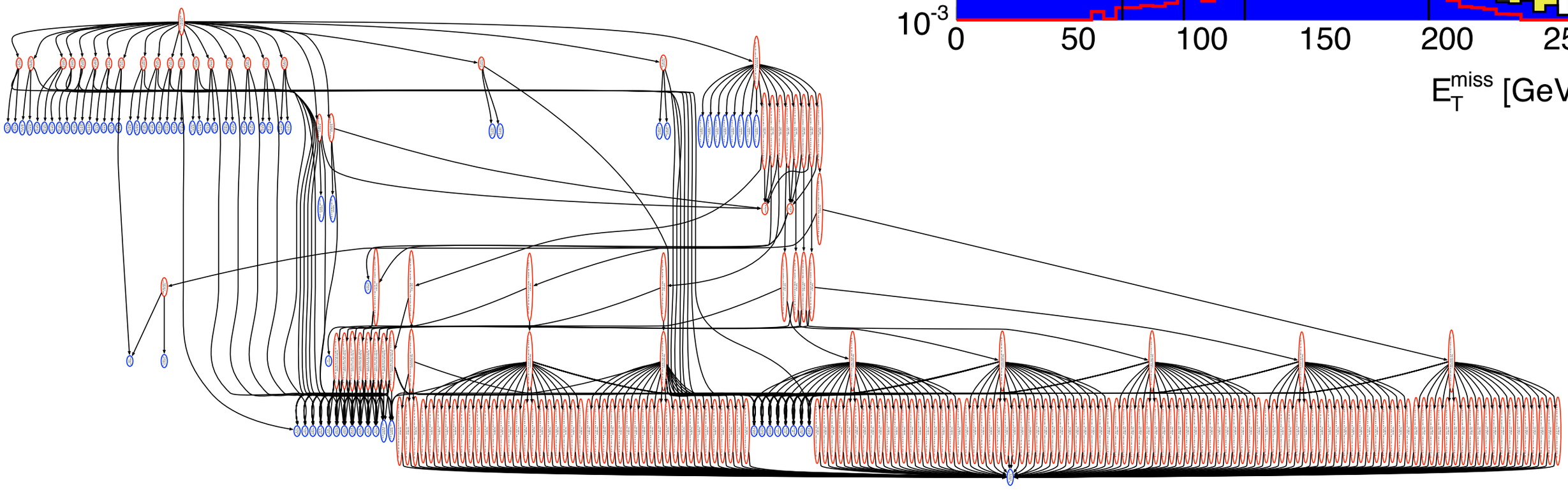




VISUALIZING THE MODEL FOR ONE CHANNEL

After parametrizing each component of the mixture model, the pdf for a single channel might look like this





SIMULTANEOUS MULTI-CHANNEL MODEL

Simultaneous Multi-Channel Model: Several disjoint regions of the data are modeled simultaneously. Identification of common parameters across many channels requires coordination between groups such that meaning of the parameters are really the same.

$$\mathbf{f}_{\text{sim}}(\mathcal{D}_{\text{sim}}|\boldsymbol{\alpha}) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right]$$

where $\mathcal{D}_{\text{sim}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_{c_{\max}}\}$

Control Regions: Some channels are not populated by signal processes, but are used to constrain the nuisance parameters

- attempt to describe systematics in a statistical language
- Prototypical Example: "on/off" problem with unknown ν_b

$$\mathbf{f}(n, m|\mu, \nu_b) = \underbrace{\text{Pois}(n|\mu + \nu_b)}_{\text{signal region}} \cdot \underbrace{\text{Pois}(m|\tau\nu_b)}_{\text{control region}}$$

Constraint terms

Often detailed statistical model for auxiliary measurements that measure certain nuisance parameters are not available.

- one typically has MLE for α_p, denoted a_p and standard error

Constraint Terms: are idealized pdfs for the MLE.

$$f_p(a_p|\alpha_p) \quad \text{for} \quad p \in \mathbb{S}$$

- common choices are Gaussian, Poisson, and log-normal
- New: careful to write constraint term a frequentist way
- Previously: $\pi(\alpha_p|a_p) = f_p(a_p|\alpha_p)\eta(\alpha_p)$ with uniform η

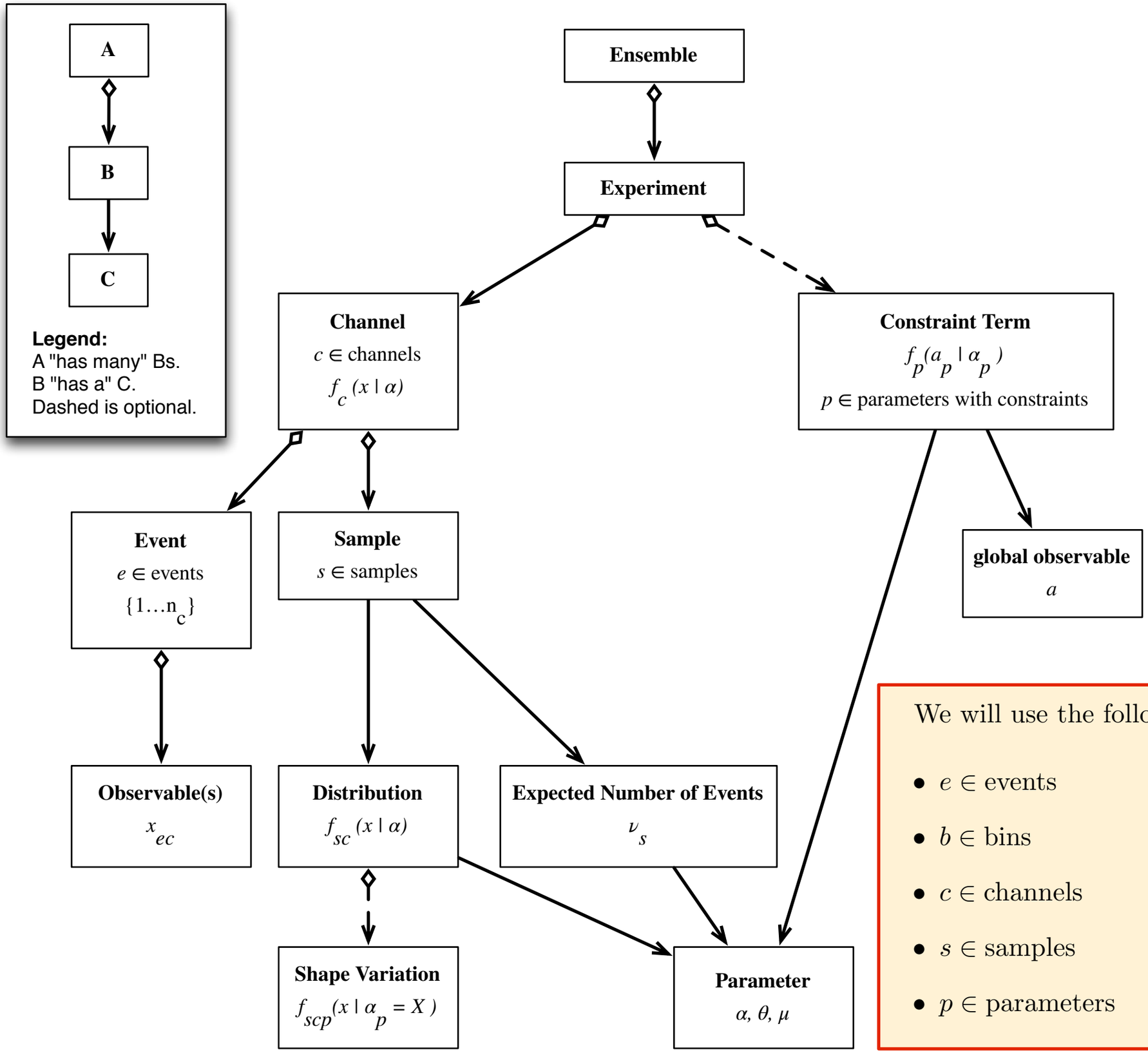
Simultaneous Multi-Channel Model with constraints:

$$\mathbf{f}_{\text{tot}}(\mathcal{D}_{\text{sim}}, \mathcal{G}|\boldsymbol{\alpha}) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right] \cdot \prod_{p \in \mathbb{S}} f_p(a_p|\alpha_p)$$

where

$$\mathcal{D}_{\text{sim}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_{c_{\max}}\}, \quad \mathcal{G} = \{a_p\} \quad \text{for} \quad p \in \mathbb{S}$$

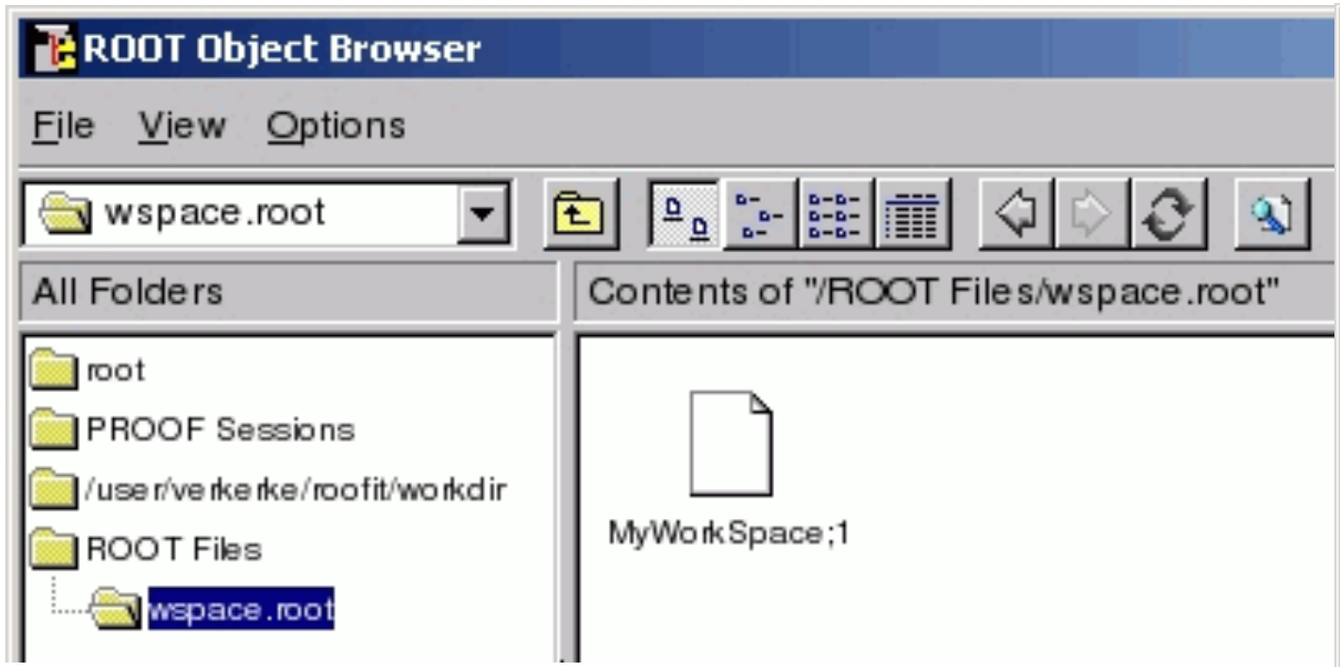
Conceptual building blocks



We will use the following mnemonic index conventions:

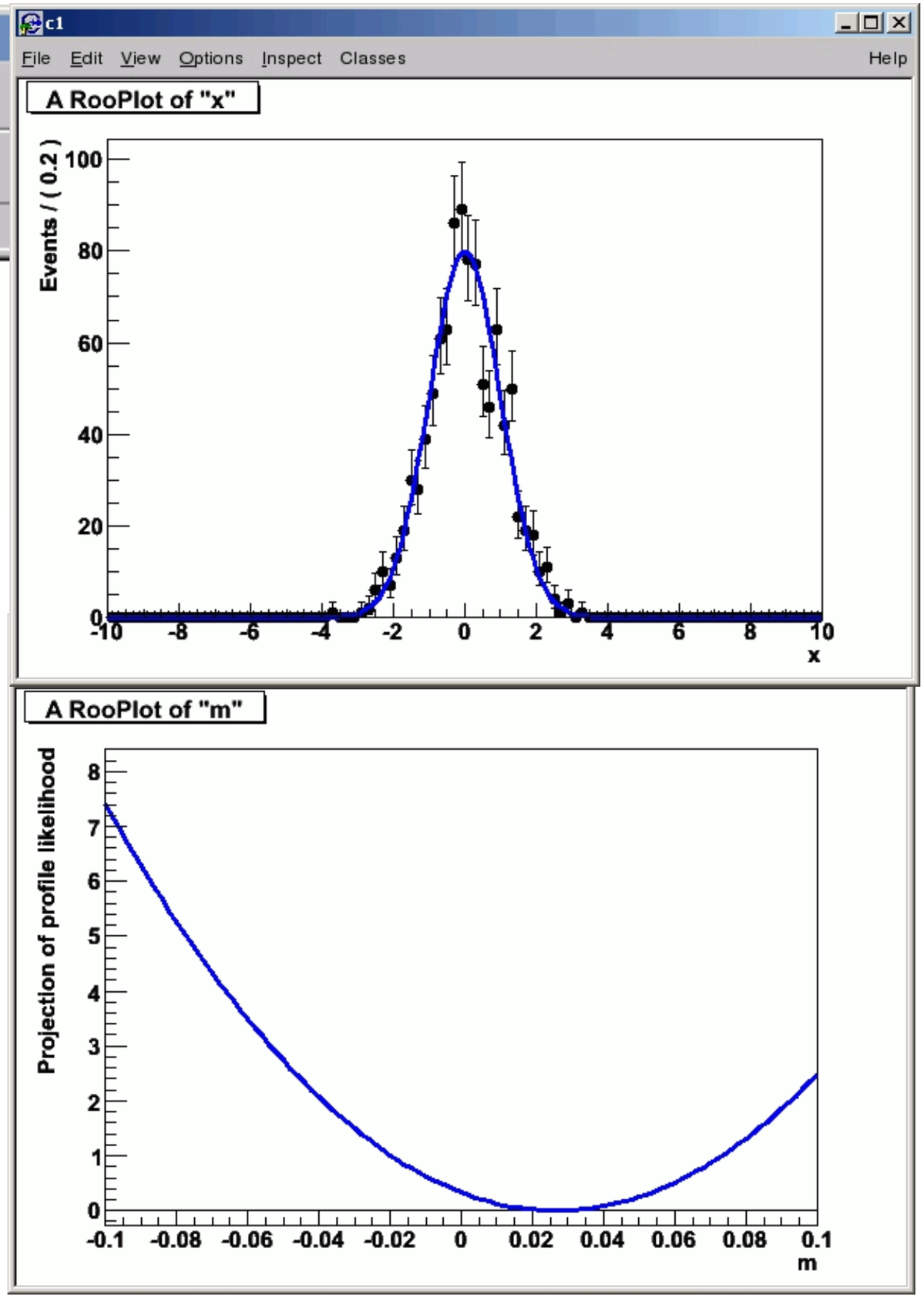
- $e \in$ events
- $b \in$ bins
- $c \in$ channels
- $s \in$ samples
- $p \in$ parameters

Example of Digital Publishing



RooFit's Workspace now provides the ability to save in a ROOT file the full likelihood model, any priors you might want, and the minimal data necessary to reproduce likelihood function.

Need this for combinations, as p-value is not sufficient information for a proper combination.



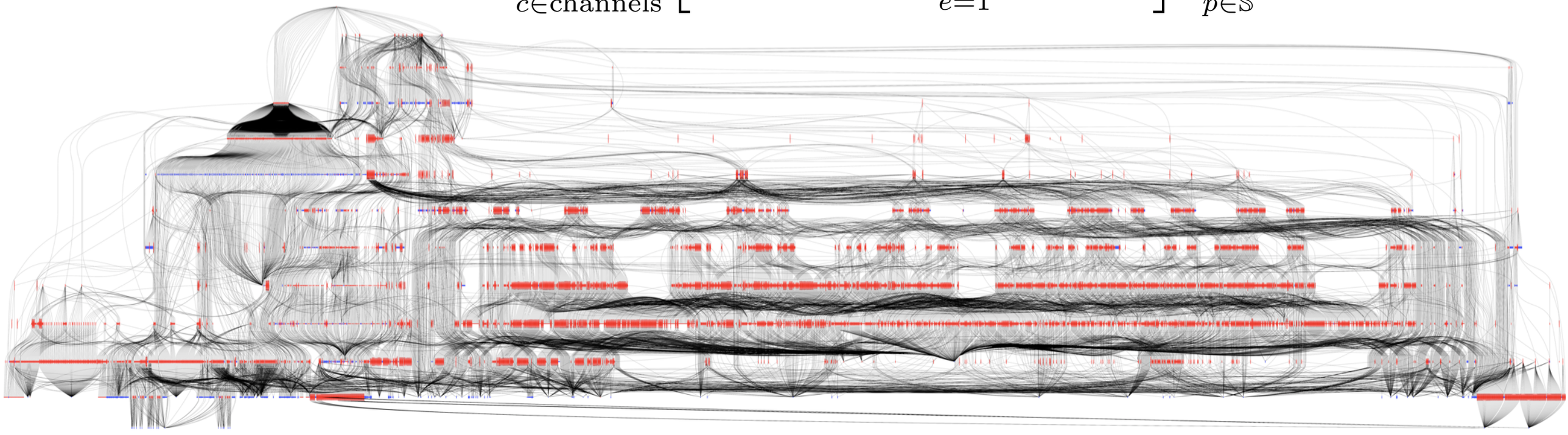
VISUALIZING THE COMBINED MODEL

State of the art: At the time of the discovery, the combined Higgs search included 100 disjoint channels and >500 nuisance parameters

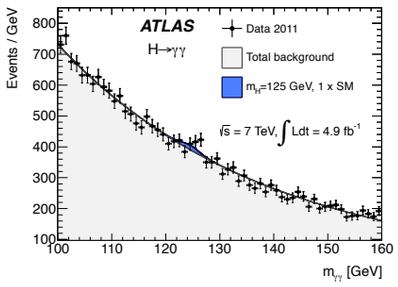
RooFit / RooStats: is the modeling language (C++) which provides technologies for collaborative modeling

- provides technology to publish likelihood functions digitally
- and more, it's the full model so we can also generate pseudo-data

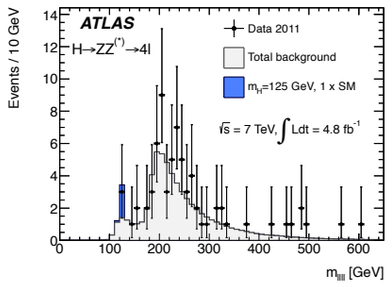
$$\mathbf{f}_{\text{tot}}(\mathcal{D}_{\text{sim}}, \mathcal{G}|\boldsymbol{\alpha}) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right] \cdot \prod_{p \in \mathbb{S}} f_p(a_p|\alpha_p)$$



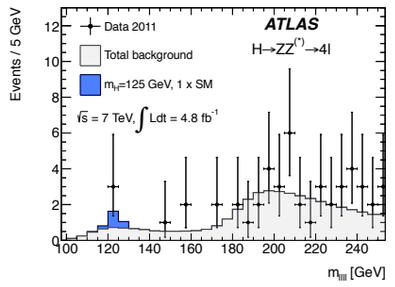
Evolution of Model Complexity



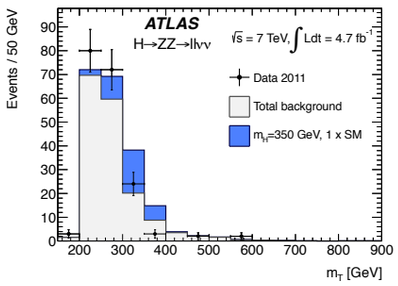
(a)



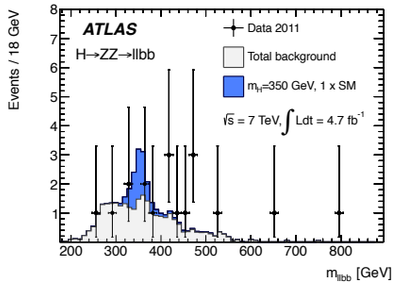
(b)



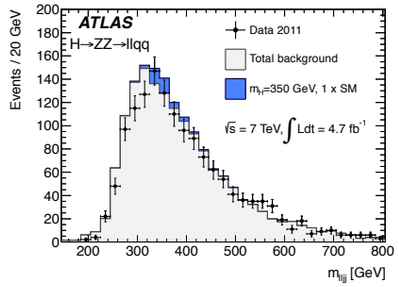
(c)



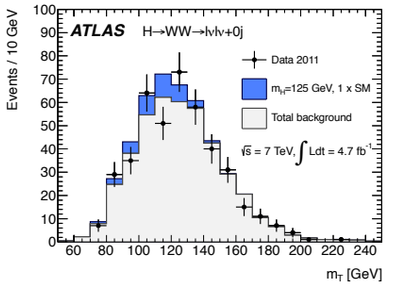
(d)



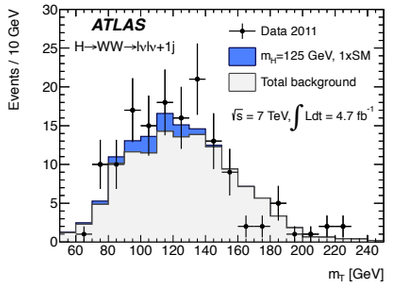
(e)



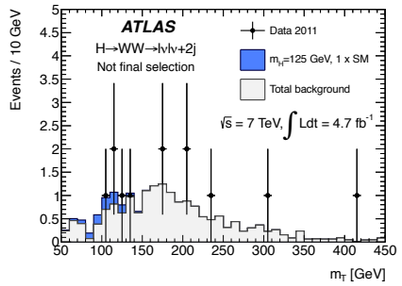
(f)



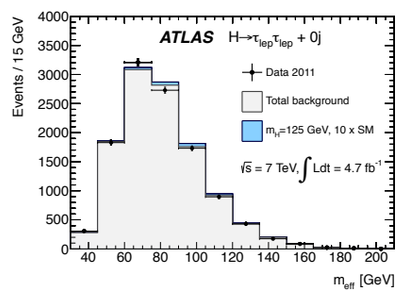
(g)



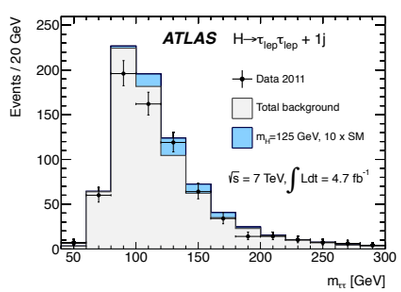
(h)



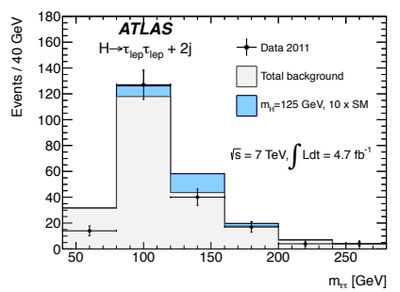
(i)



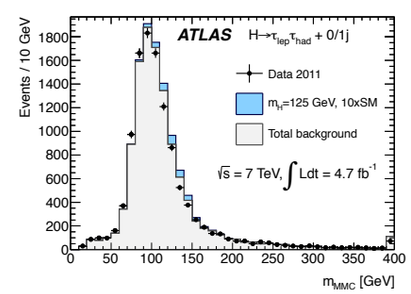
(a)



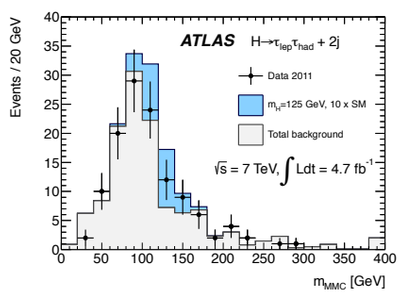
(b)



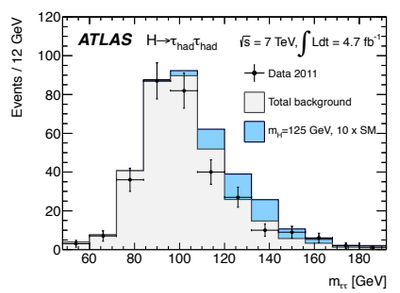
(c)



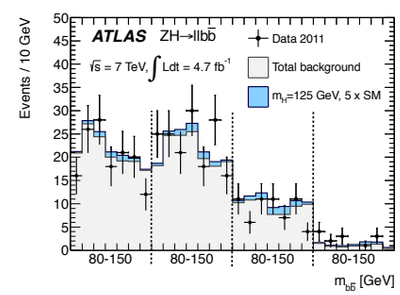
(d)



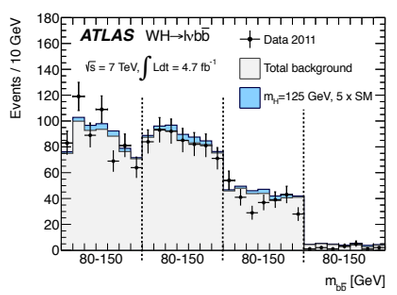
(e)



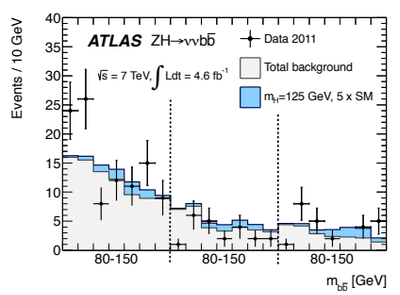
(f)



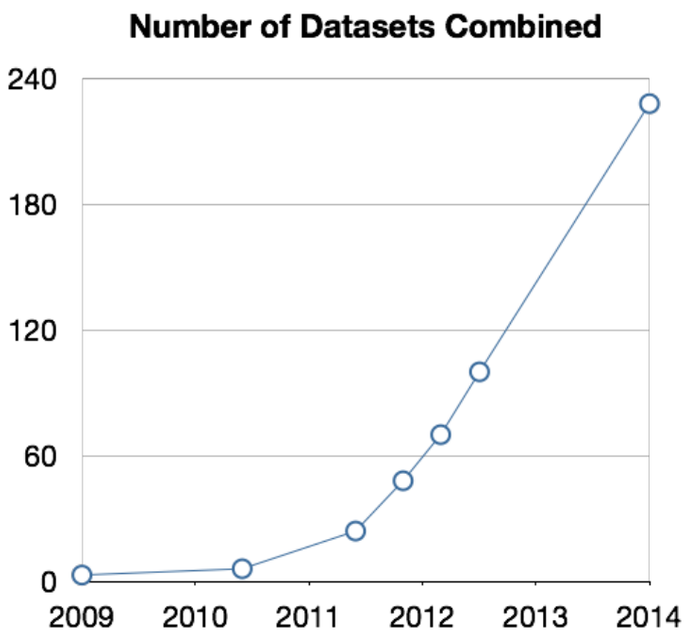
(g)

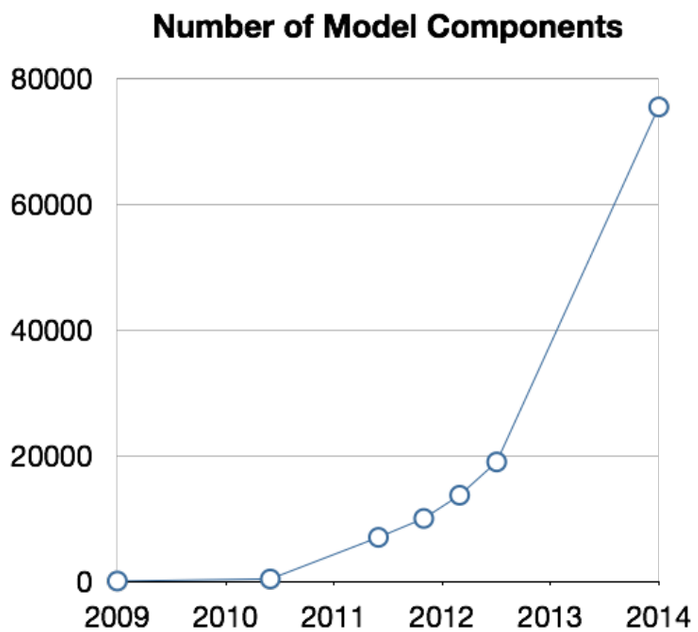


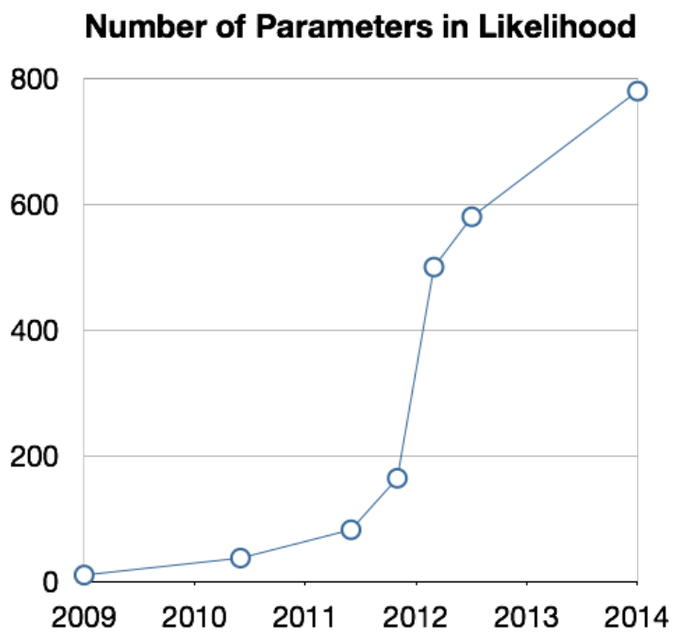
(h)



(i)

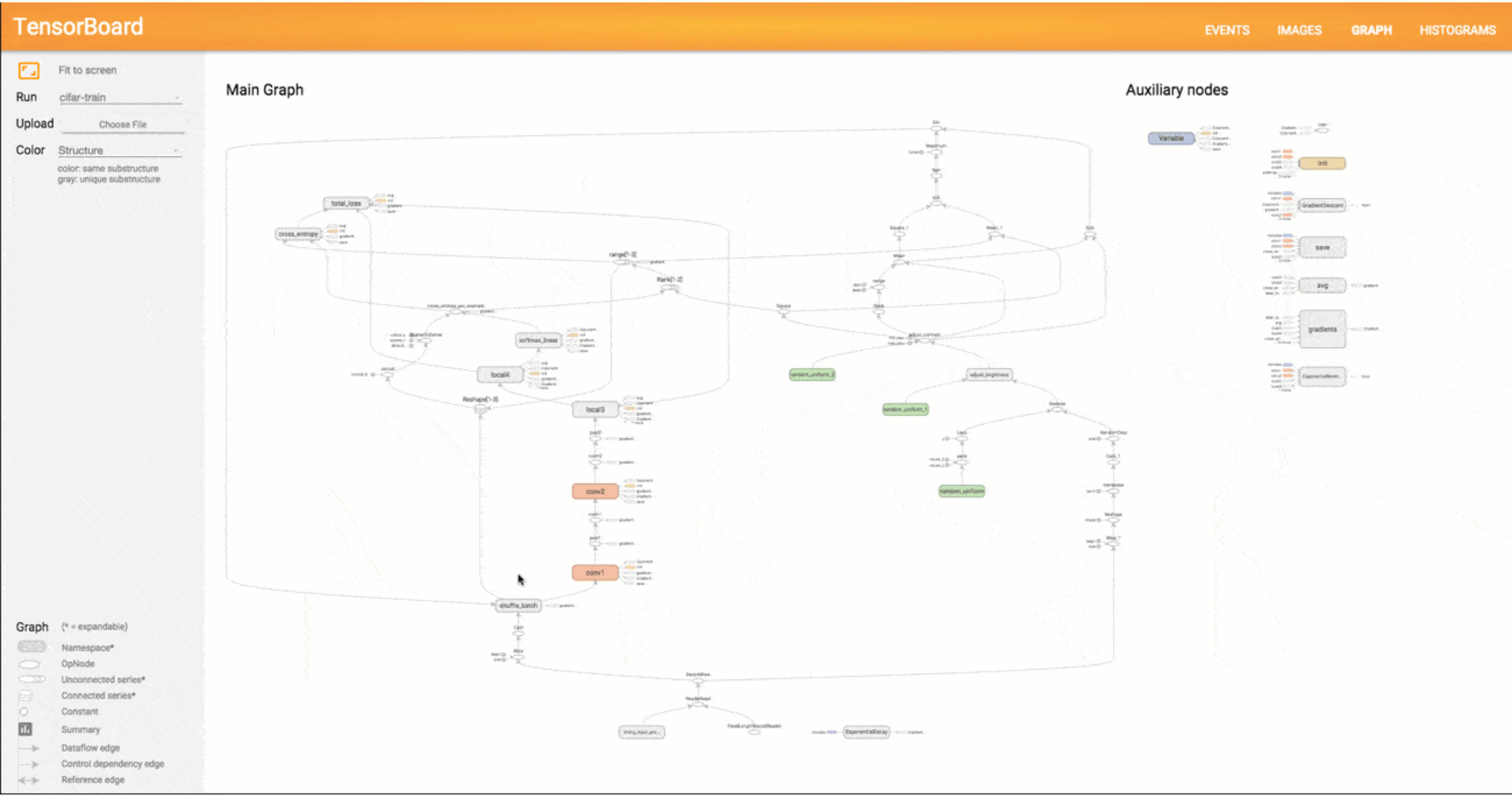






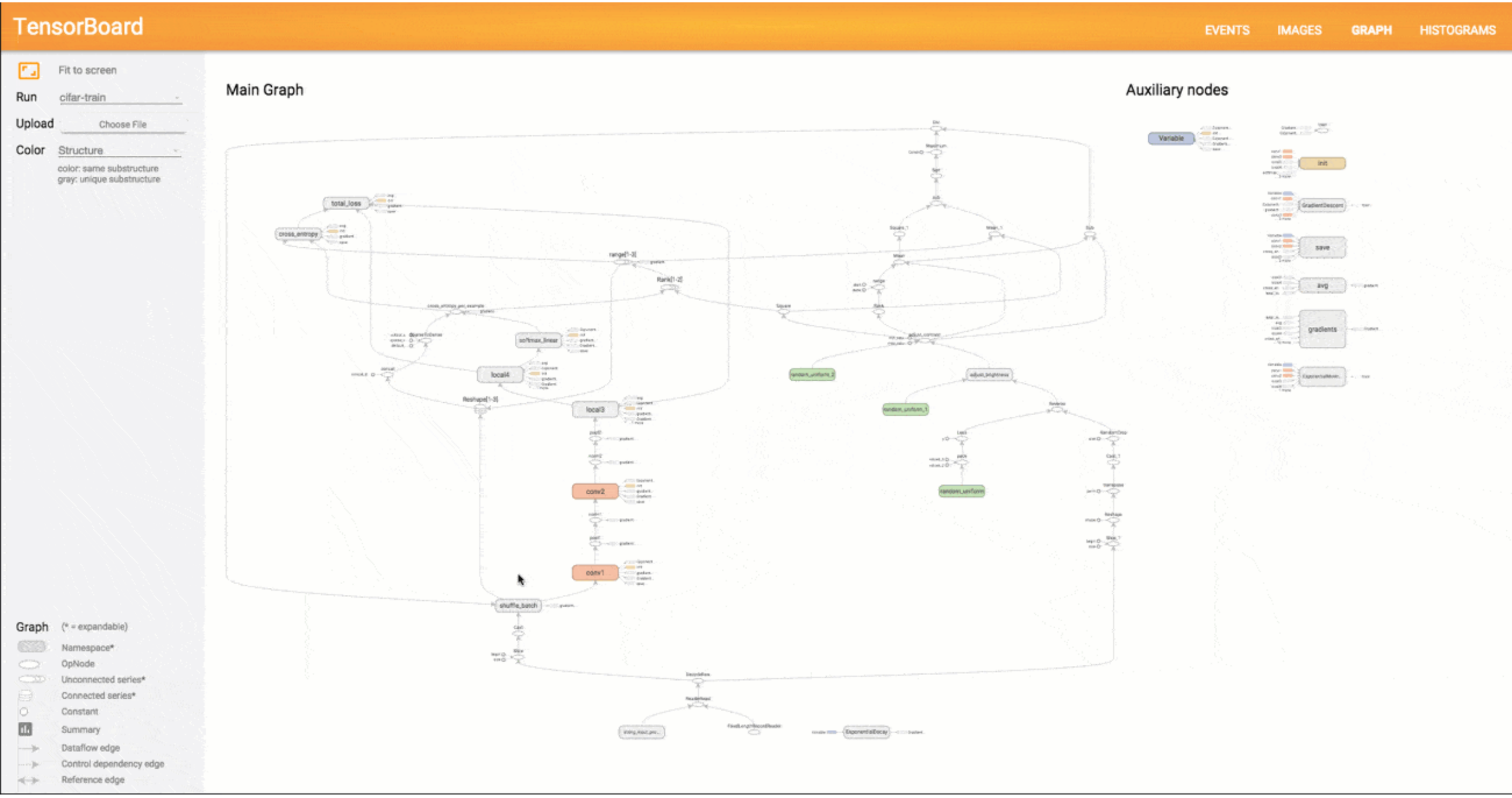
TENSORBOARD

Modern Machine Learning tools like TensorFlow express the model in a similar way as a Directed Acyclic Graph (DAG)

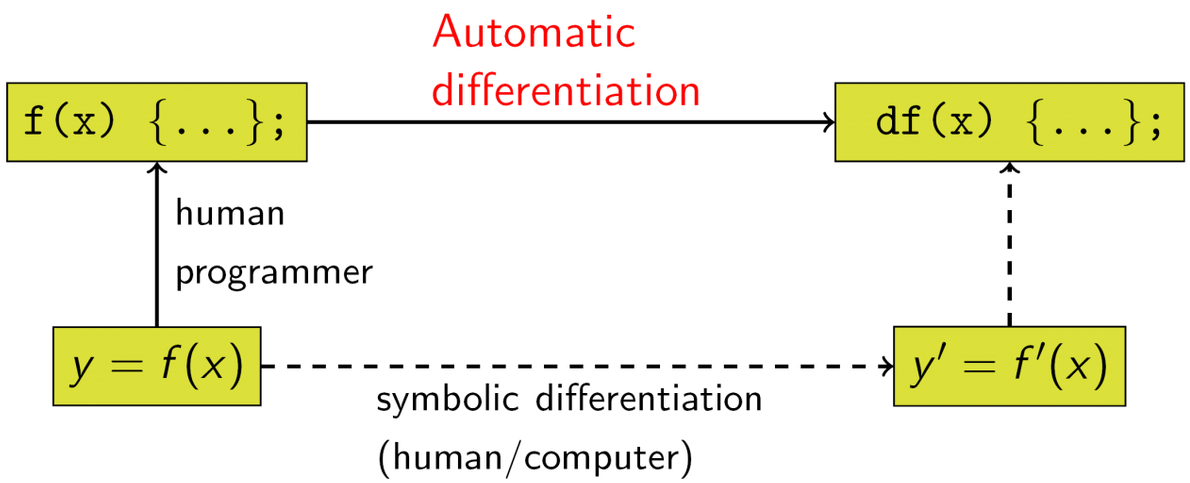


TENSORBOARD

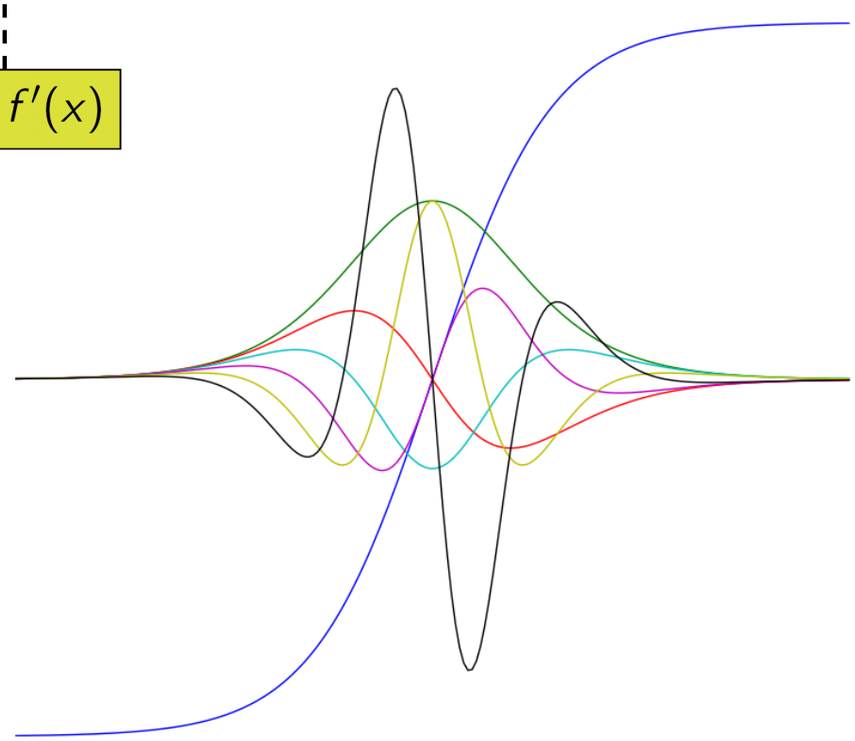
Modern Machine Learning tools like TensorFlow express the model in a similar way as a Directed Acyclic Graph (DAG)



AUTOMATIC DIFFERENTIATION




```
>>> import autograd.numpy as np  # Thinly-wrapped numpy
>>> from autograd import grad    # The only autograd function you may ever need
>>>
>>> def tanh(x):                 # Define a function
...     y = np.exp(-2.0 * x)
...     return (1.0 - y) / (1.0 + y)
...
>>> grad_tanh = grad(tanh)       # Obtain its gradient function
>>> grad_tanh(1.0)               # Evaluate the gradient at x = 1.0
0.41997434161402603
>>> (tanh(1.0001) - tanh(0.9999)) / 0.0002  # Compare to finite differences
0.41997434264973155
```

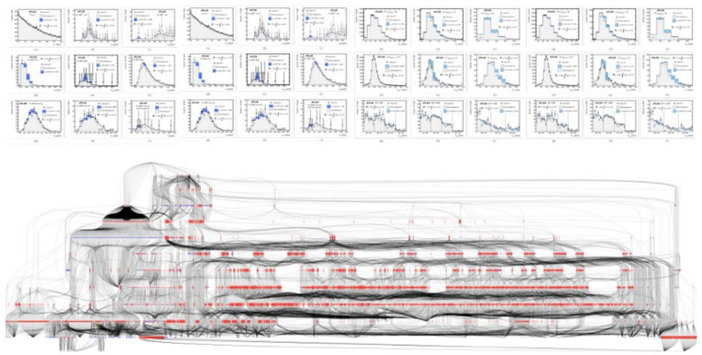


We can continue to differentiate as many times as we like, and use numpy's vectorization of scalar-valued functions across many different input values:

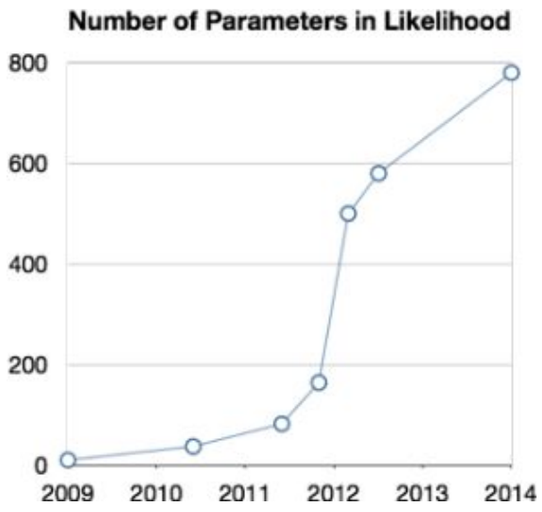
```
>>> from autograd import elementwise_grad as egrad  # for functions that vectorize over inputs
>>> import matplotlib.pyplot as plt
>>> x = np.linspace(-7, 7, 200)
>>> plt.plot(x, tanh(x),
...          x, egrad(tanh)(x),                                 # first  derivative
...          x, egrad(egrad(tanh))(x),                          # second derivative
...          x, egrad(egrad(egrad(tanh)))(x),                   # third  derivative
...          x, egrad(egrad(egrad(egrad(tanh))))(x),            # fourth derivative
...          x, egrad(egrad(egrad(egrad(egrad(tanh)))))(x),     # fifth  derivative
...          x, egrad(egrad(egrad(egrad(egrad(egrad(tanh))))))(x))  # sixth  derivative
>>> plt.show()
```

https://en.wikipedia.org/wiki/Automatic_differentiation

Probabilistic programming frameworks



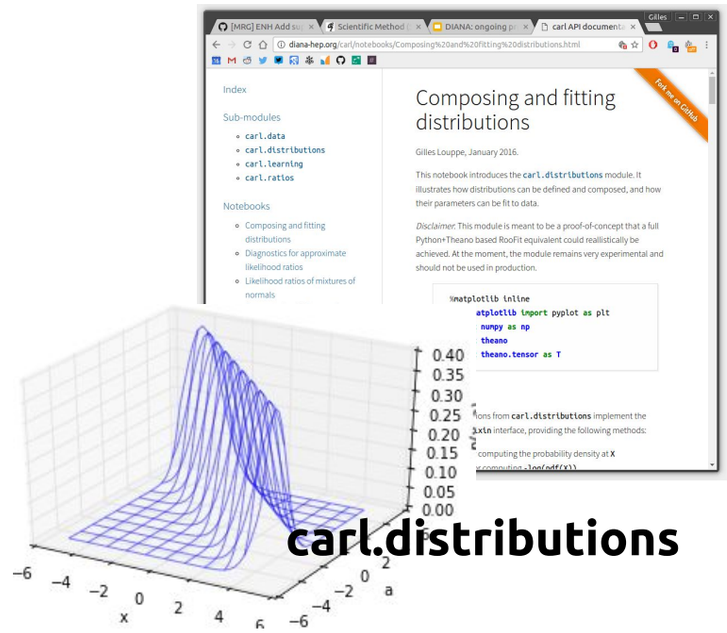
$$\mathbf{f}_{\text{tot}}(\mathcal{D}_{\text{sim}}, \mathcal{G}|\boldsymbol{\alpha}) = \prod_{c \in \text{channels}} \left[\text{Pois}(n_c|\nu_c(\boldsymbol{\alpha})) \prod_{e=1}^{n_c} f_c(x_{ce}|\boldsymbol{\alpha}) \right] \cdot \prod_{p \in \mathbb{S}} f_p(a_p|\alpha_p)$$

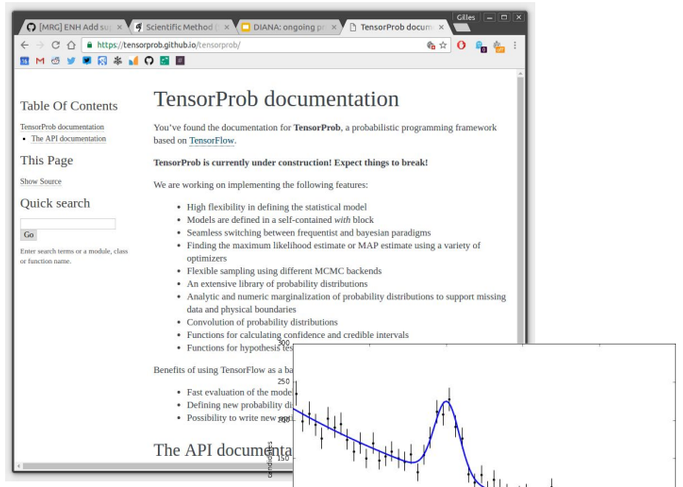


RooFit serves us well, but shows limits in terms of **scalability**.

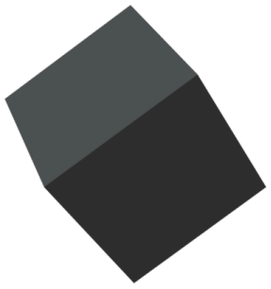
Using a data flow graph framework, RooFit would be **distributed**, **GPU-enabled** and automatically **differentiable**.

Feasibility? Certainly **within reach!** As illustrated by our tentative proof-of-concepts carl.distributions [Gilles Louppe] and tensorprob [Igor Babuschkin, now at DeepMind]. See also Edward.





Edward



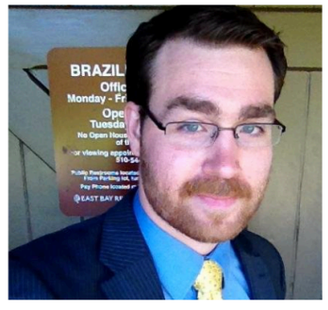
A library for probabilistic modeling, inference, and criticism.

Edward is a Python library for probabilistic modeling, inference, and criticism. It is a testbed for fast experimentation and research with probabilistic models, ranging from classical hierarchical models on small data sets to complex deep probabilistic models on large data sets. Edward fuses three fields: Bayesian statistics and machine learning, deep learning, and probabilistic programming.

It supports **modeling** with

Dustin Tran

Ph.D. Student
Columbia University
dustin@cs.columbia.edu (@dustinvtran, http://dustintran.com



Matthew Feickert

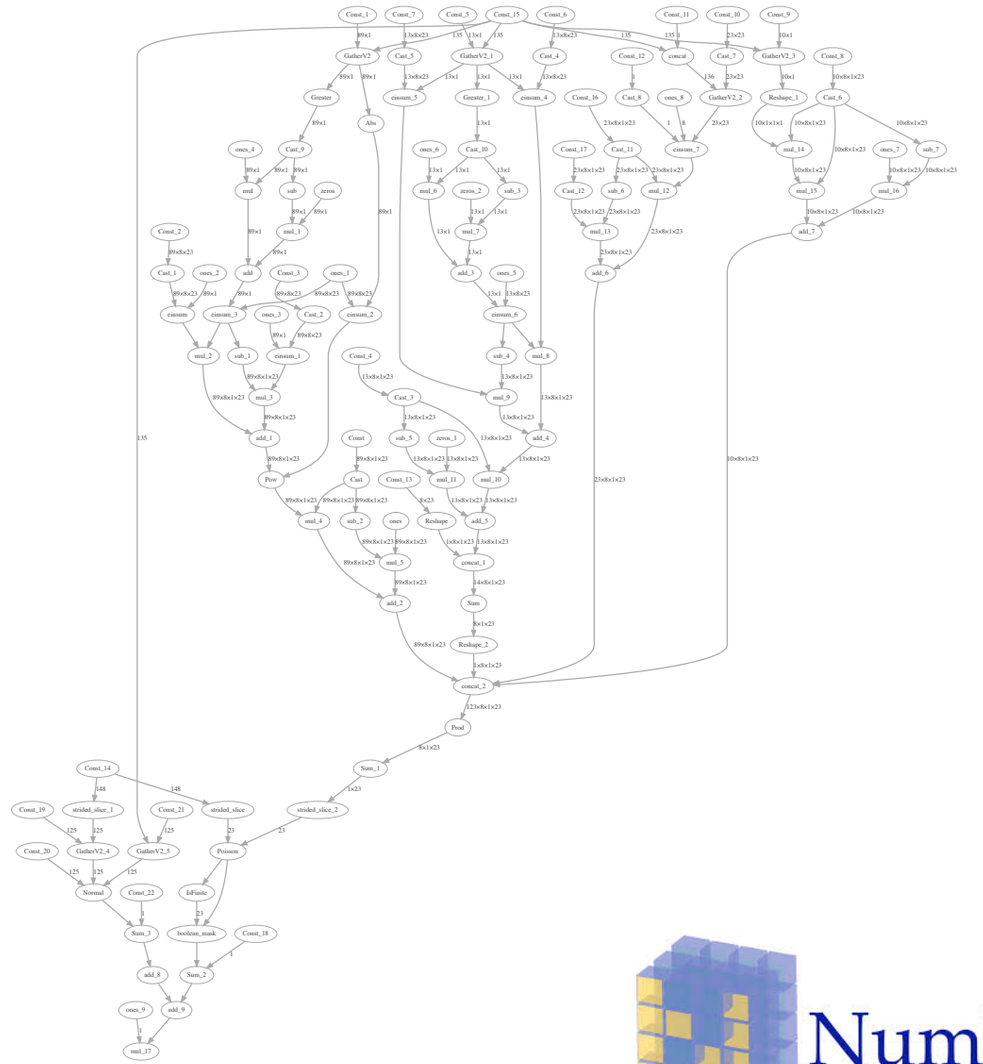
High Energy Physics Ph.D. Candidate
Southern Methodist University
matthew.feickert@cern.ch or mfeickert@smu.edu
GitHub: matthewfeickert @HEPfeickert

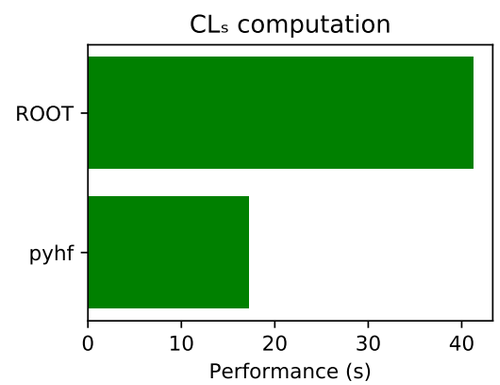
pyhf: modern implementation of HEP likelihood computations

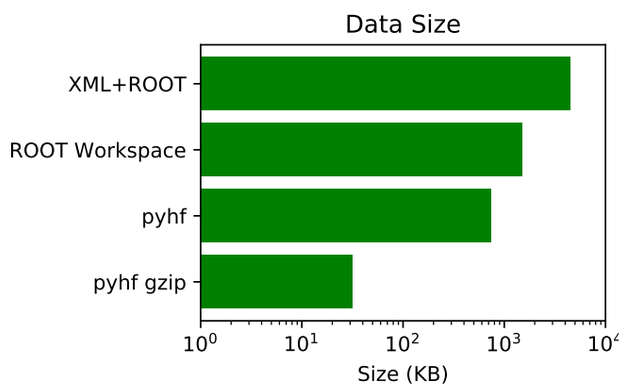
→ easy-to-publish likelihoods
→ make likelihoods fast to compute

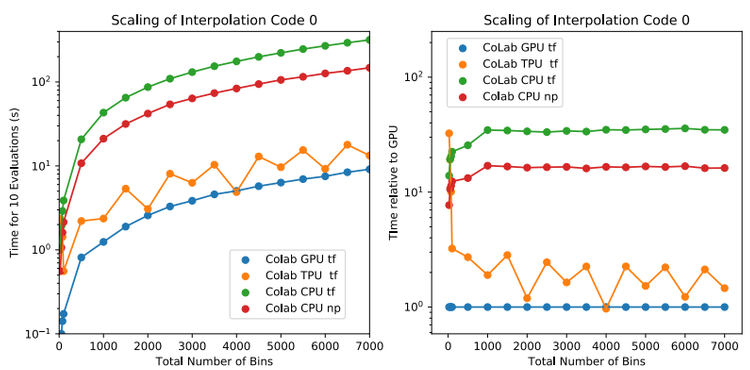
Likelihoods as computational graphs of array computations

- automatic gradients
- compute on specialized hardware (GPU / TPU)
- graph structure allows distribution across machines — stat. combinations









NumPy

PYTORCH

DASK

TensorFlow

Gaussian Processes

NEW PAPER

https://arxiv.org/abs/1709.05681

Information | References (44) | Citations (0) | Files | Plots

Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes

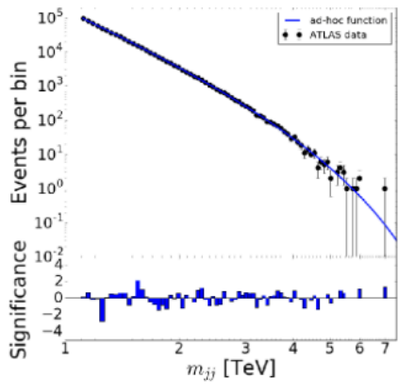
Meghan Frate, Kyle Cranmer, Saarik Kalia, Alexander Vandenberg-Rodes, Daniel Whiteson

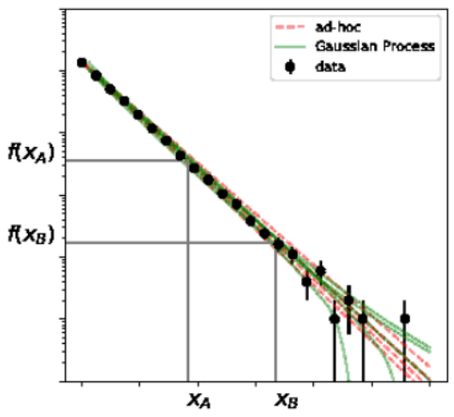
Sep 17, 2017 - 14 pages

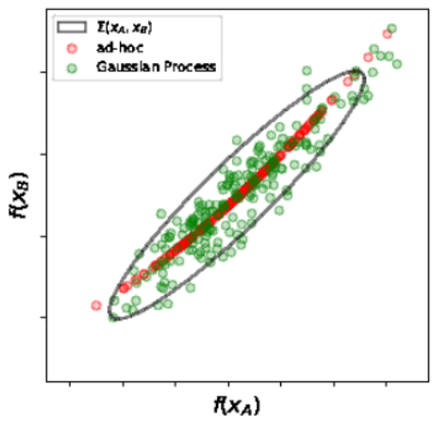
e-Print: arXiv:1709.05681 [physics.data-an] | PDF

Abstract (arXiv)
We describe a procedure for constructing a model of a smooth data spectrum using Gaussian processes rather than the historical parametric description. This approach considers a fuller space of possible functions, is robust at increasing luminosity, and allows us to incorporate our understanding of the underlying physics. We demonstrate the application of this approach to modeling the background to searches for dijet resonances at the Large Hadron Collider and describe how the approach can be used in the search for generic localized signals.

Note: *Temporary entry*
Note: 14 pages, 16 figures
Keyword(s): INSPIRE: background | CERN LHC Coll | dijet | resonance | data analysis method | Gauss model | statistics | statistical analysis







Show more plots

Record added 2017-09-19, last modified 2017-10-07

diana hep

Collaborative Analyses

Establish infrastructure for a higher-level of collaborative analysis, building on the successful patterns used for the Higgs boson discovery and enabling a deeper communication between the theoretical community and the experimental community

Reproducible Analyses

Streamline efforts associated to reproducibility, analysis preservation, and data preservation by making these native concepts in the tools

Interoperability

Improve the interoperability of HEP tools with the larger scientific software ecosystem, incorporating best practices and algorithms from other disciplines into HEP

Faster Processing

Increase the CPU and IO performance needed to reduce the iteration time so crucial to exploring new ideas

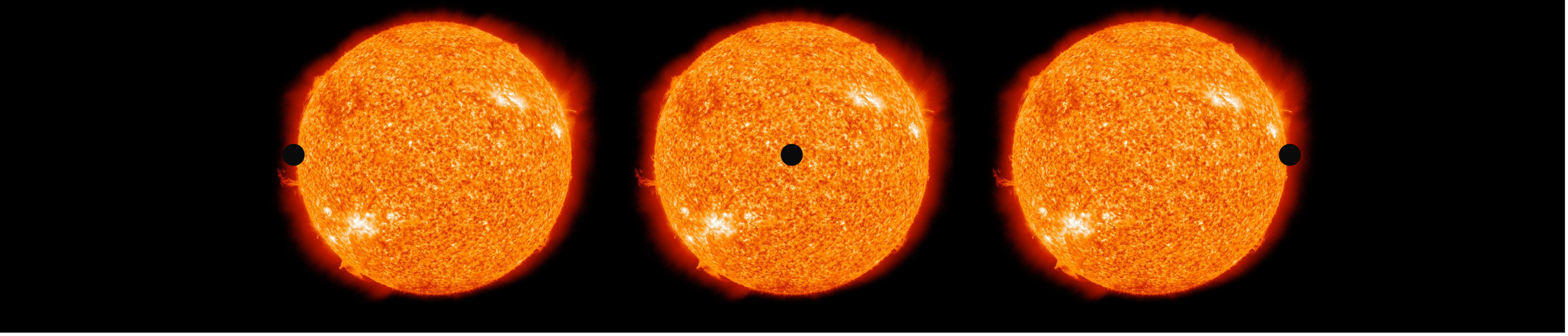
Better Software

Develop software to effectively exploit emerging many- and multi-core hardware.
Promote the concept of software as a research product.

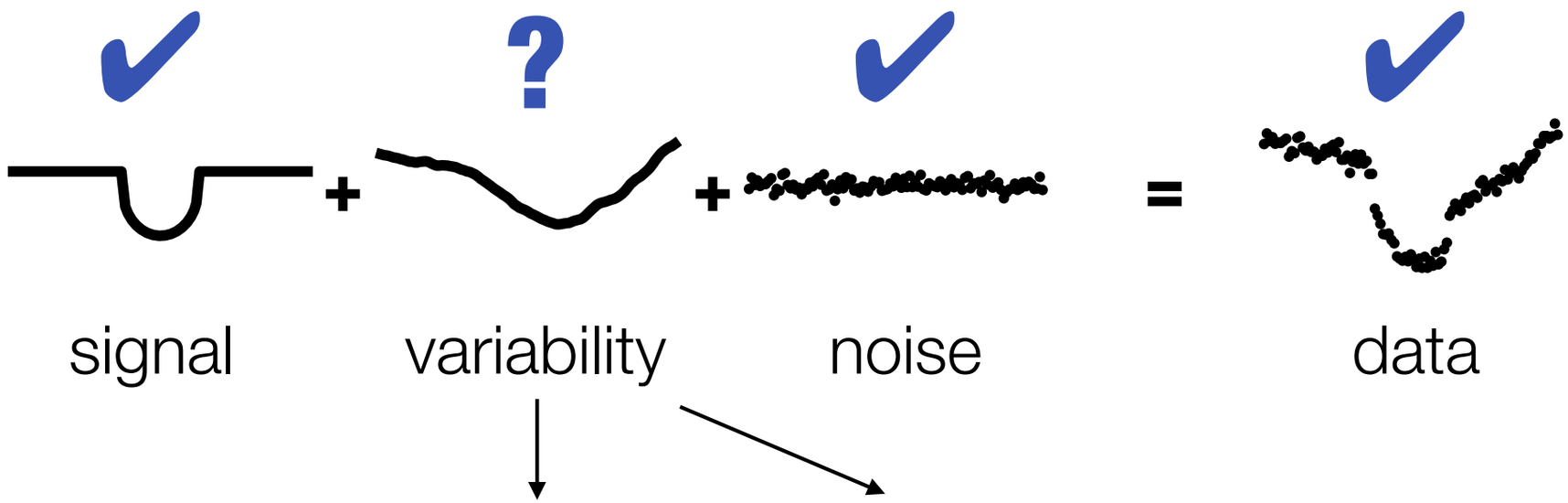
Training

Provide training for students in all of our core research topics.

http://diana-hep.org

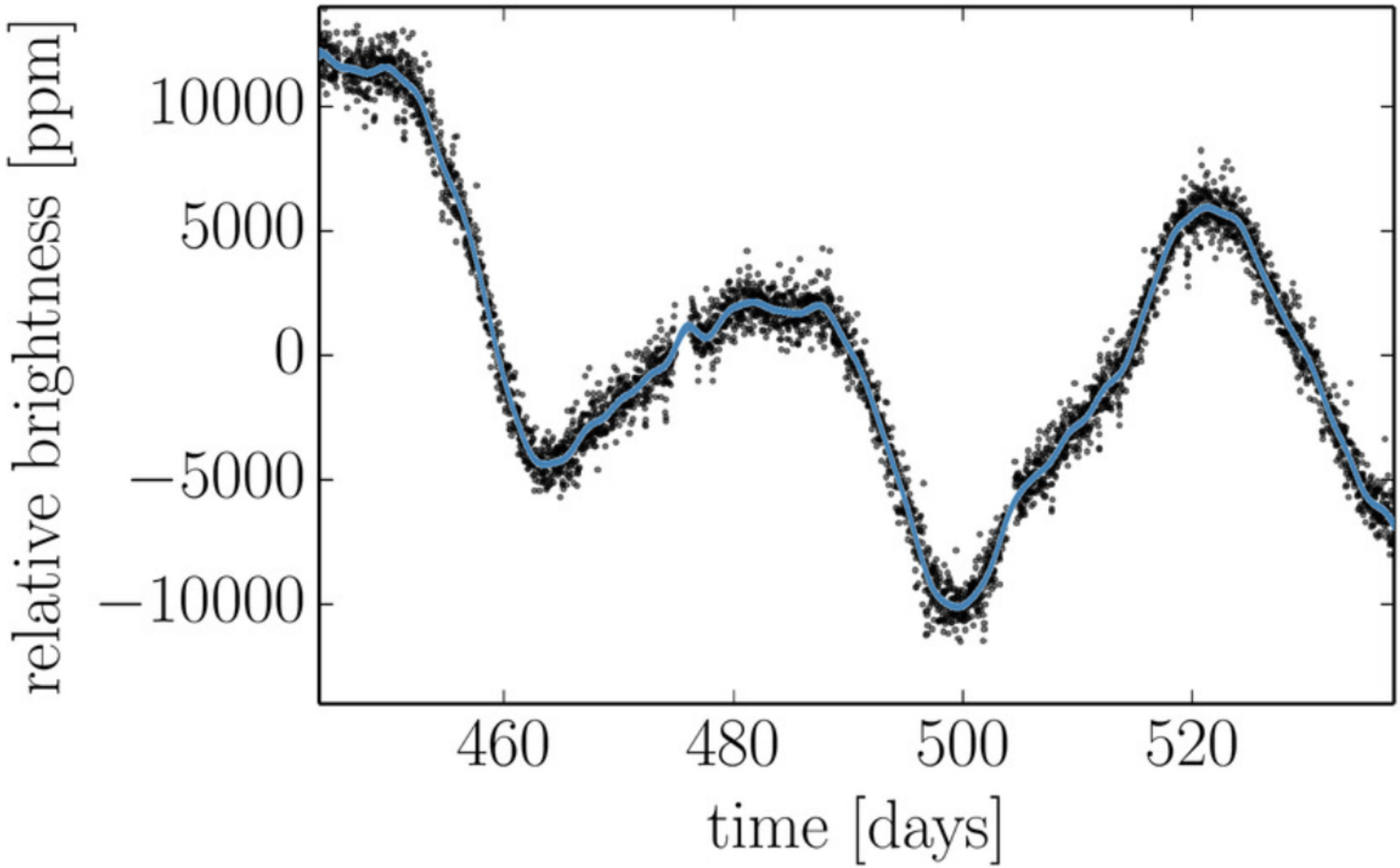


The anatomy of a transit observation



[Slides by Dan Foreman-Mackey]

AN EXOPLANET EXAMPLE



the data are drawn from one

HUGE*

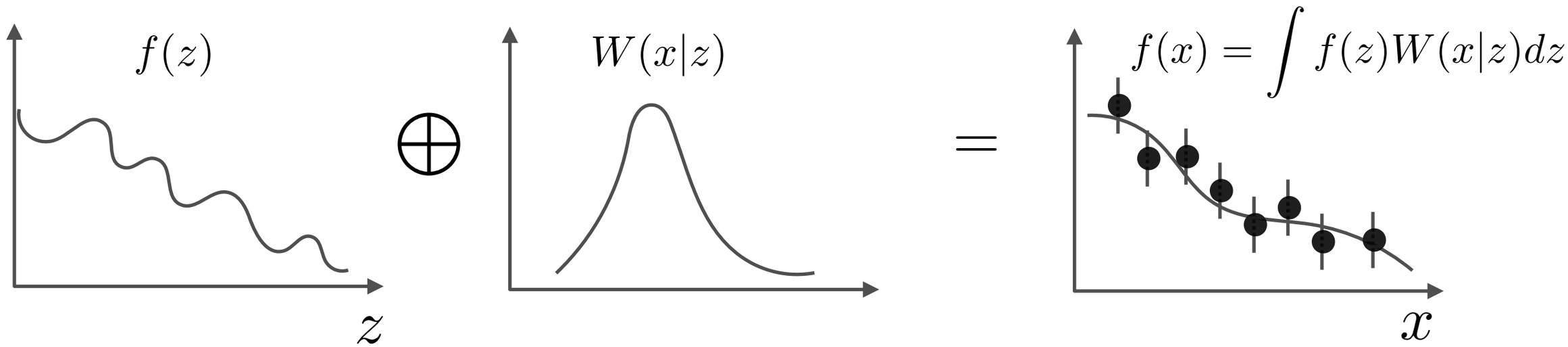
Gaussian

* the dimension is the number of data points.

A PARTICLE PHYSICS EXAMPLE OF A GAUSSIAN PROCESS

Consider unfolding when the detector response / "folding matrix" is known exactly (eg. no systematic uncertainty in detector response).

- the bin counts of observed distribution are uncorrelated Poisson fluctuations.



truth level

folding / smearing detector response

folded distribution & poisson fluctuations

The unfolding process gives us a best estimate for unfolded distribution $f(z_i)$ and covariance matrix (eg. $f(z_i)$ and $f(z_{i+1})$ are usually highly correlated)

- the result of unfolding can be considered a Gaussian Process (GP).
- Gaussian Processes can be generalized to continuous z (unbinned distribution)

GAUSSIAN PROCESSES

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{f_\theta}(\boldsymbol{x}),\, K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right)$$

where

$$\left[K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right]_{ij} = {\sigma_i}^2\, \delta_{ij} + \underbrace{k_{\boldsymbol{\alpha}}(x_i,\, x_j)}_{\textbf{kernel function}}$$

(where the magic happens)

see: gaussianprocess.org/**gpml** github.com/**dfm/george**

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta},\, \boldsymbol{\alpha}) = -\frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right]^{\mathrm{T}} K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})^{-1}\left[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right] - \frac{1}{2}\log\det K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma}) - \frac{N}{2}\log 2\,\pi$$

where

$$\left[K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right]_{ij} = {\sigma_i}^2\,\delta_{ij} + \underbrace{k_{\boldsymbol{\alpha}}(x_i,\, x_j)}_{\textbf{kernel function}}$$

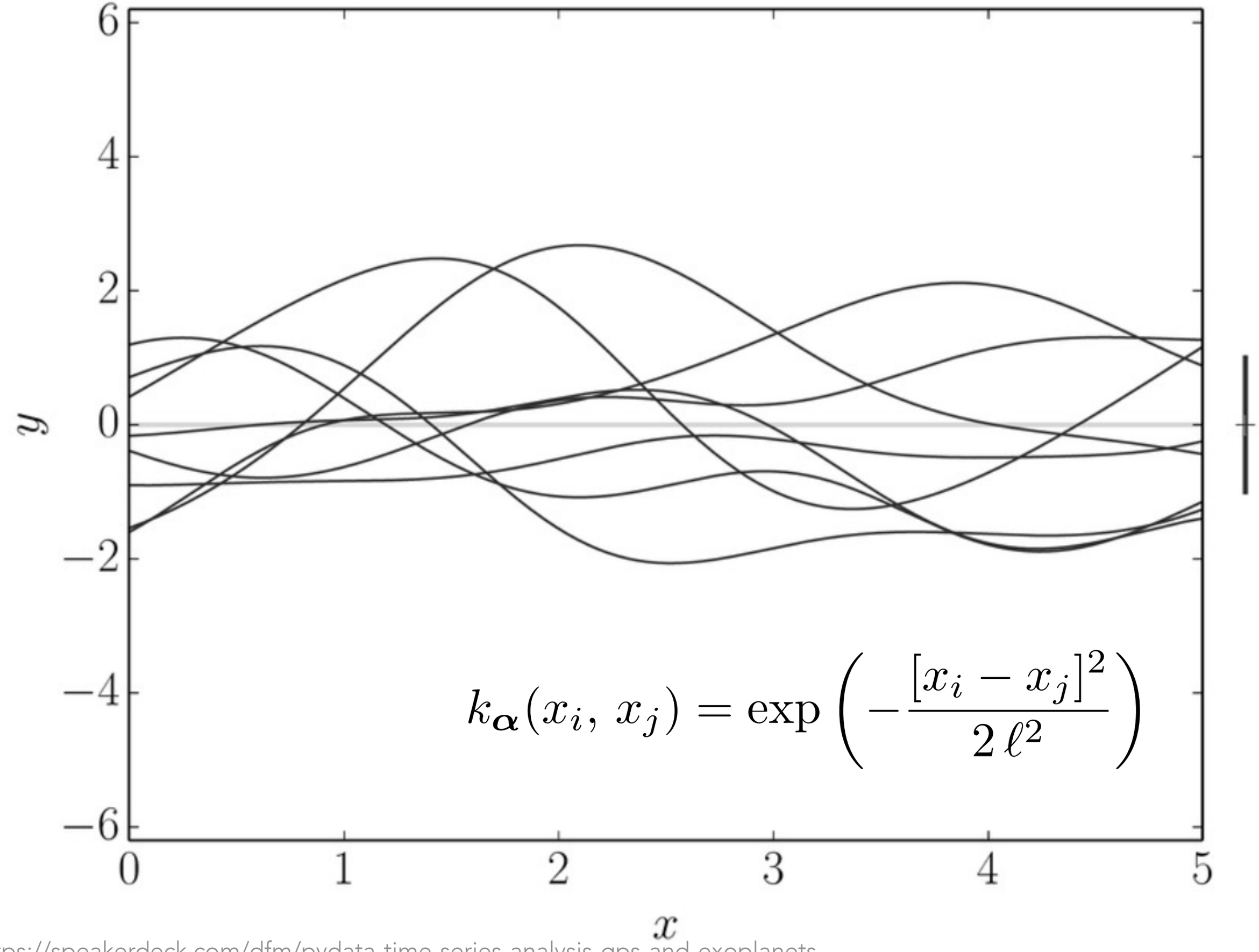
(where the magic happens)

see: gaussianprocess.org/**gpml** github.com/**dfm/george**

Gaussian Processes

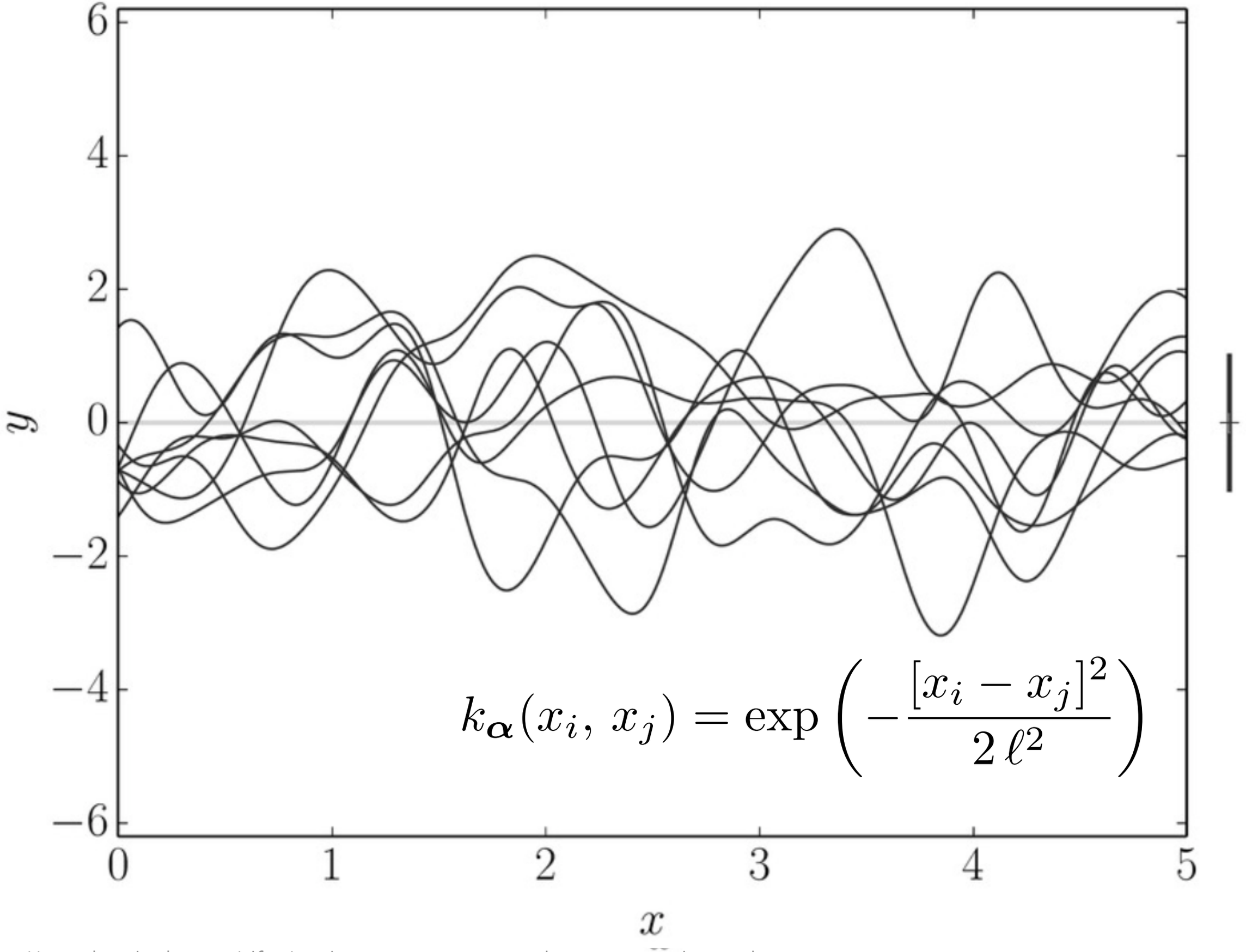
$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \exp\left(-\frac{[x_i - x_j]^2}{2\,\ell^2}\right)$$

Gaussian Processes

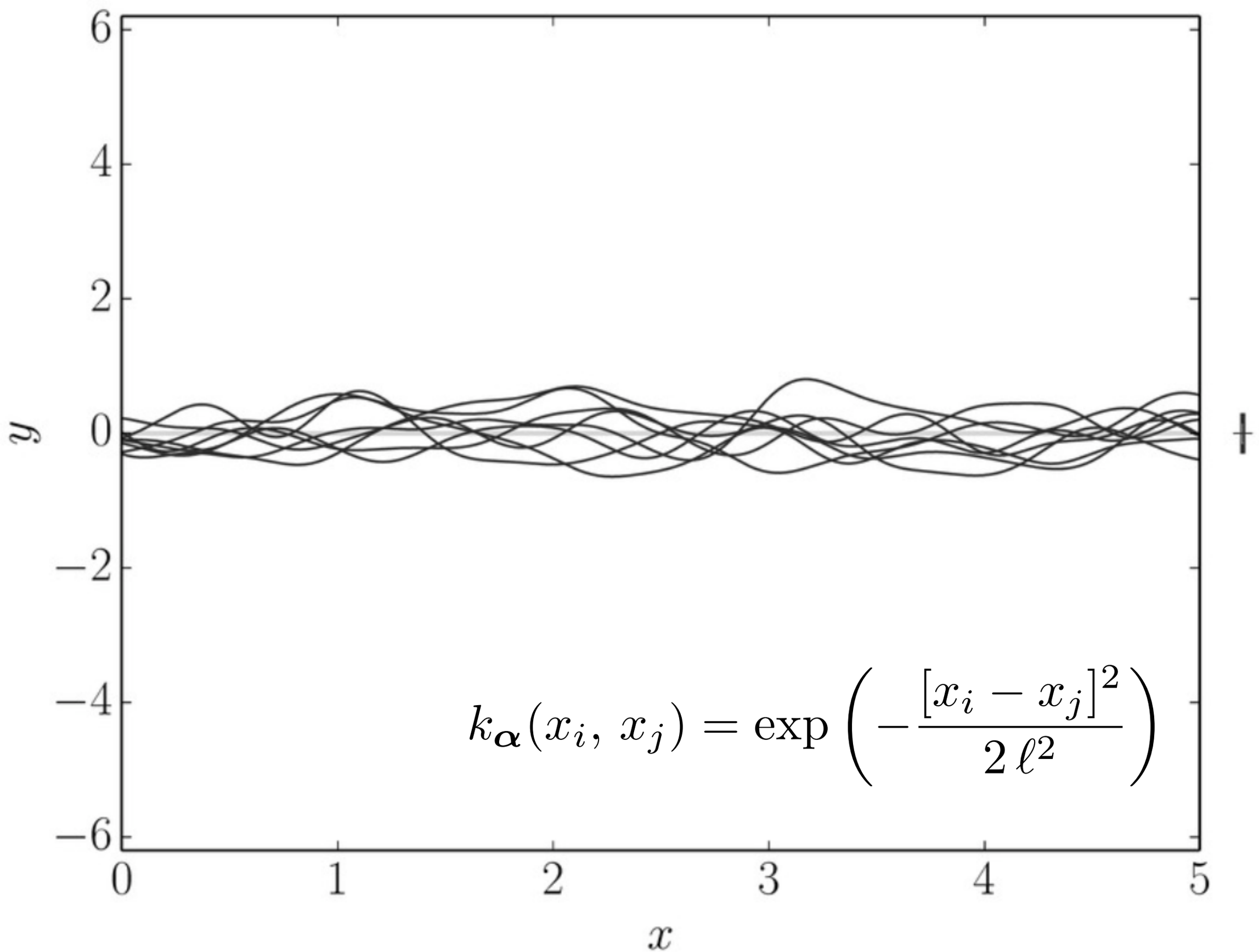


https://speakerdeck.com/dfm/pydata-time-series-analysis-gps-and-exoplanets

Gaussian Processes



Gaussian Processes



https://speakerdeck.com/dfm/pydata-time-series-analysis-gps-and-exoplanets

LEARN MORE

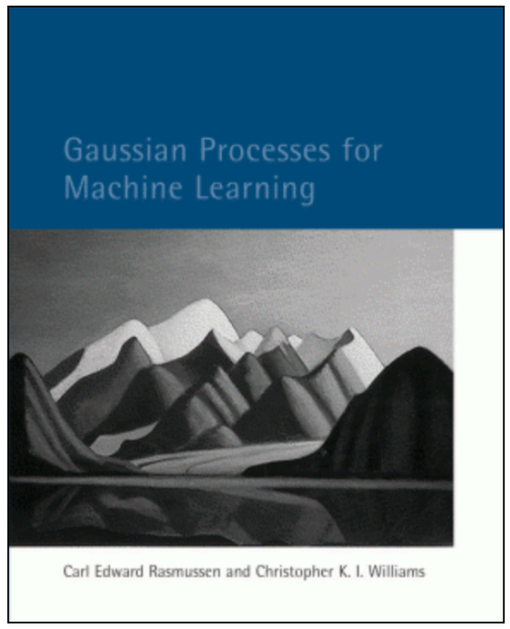
gaussianprocess.org

DiscoveryLinks | Higgs | RooStats | ALEPH | Apple | News | Life Stuff | Wikipedia, | ATLAS | inSpire | dblp | Theory&Practice | HCG | Evernote | Web Equation | job | CERN | inSpire

Division of P... | for Jet Physi... | isaachenrion... | learning-qcd... | Submit Test I... | cweniger/sw... | Statistics For... | Gaussian Pro...

Gaussian Processes for Machine Learning

Carl Edward Rasmussen and Christopher K. I. Williams
The MIT Press, 2006. ISBN 0-262-18253-X.



[Contents | Software | Datasets | Errata | Authors | Order]

Gaussian processes (GPs) provide a principled, practical, probabilistic approach to learning in kernel machines. GPs have received increased attention in the machine-learning community over the past decade, and this book provides a long-needed systematic and unified treatment of theoretical and practical aspects of GPs in machine learning. The treatment is comprehensive and self-contained, targeted at researchers and students in machine learning and applied statistics.

The book deals with the supervised-learning problem for both regression and classification, and includes detailed algorithms. A wide variety of covariance (kernel) functions are presented and their properties discussed. Model selection is discussed both from a Bayesian and a classical perspective. Many connections to other well-known techniques from machine learning and statistics are discussed, including support-vector machines, neural networks, splines, regularization networks, relevance vector machines and others. Theoretical issues including learning curves and the PAC-Bayesian framework are treated, and several approximation methods for learning with large datasets are discussed. The book contains illustrative examples and exercises, and code and datasets are available on the Web. Appendixes provide mathematical background and a discussion of Gaussian Markov processes.

The book is available for download in electronic format.

http://www.gaussianprocess.org/gpml/

Parametrized Function

vs.

Gaussian Process

PARAMETRIC FUNCTION VS. GP

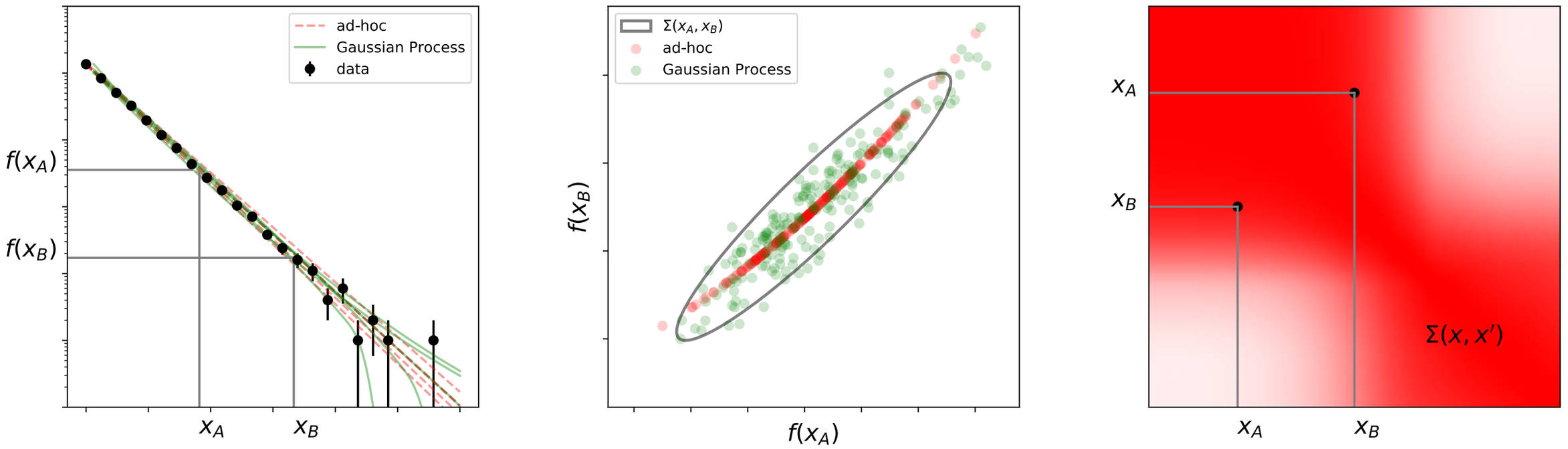


FIG. 2: Schematic of the relationship between an ad-hoc function and the GP. An example toy dataset is shown (left) with samples from the posterior for an ad-hoc 1-parameter function (red) and a GP (green). Each posterior sample is an entire curve $f(x)$, which corresponds to a particular point in the (center) plane of $f(x_A)$ vs. $f(x_B)$. The red dots for the ad-hoc 1-parameter function trace out a 1-dimensional curve, which reveals how the function is overly-rigid. In contrast, the green dots from the GP relax the assumptions and fill a correlated multivariate Gaussian (with covariance indicated by the black ellipse). The covariance kernel $\Sigma(x, x')$ for the GP is shown (right) with $\Sigma(x_A, x_B)$ corresponding to the black ellipse of the center panel.

MOAR DATA!

GP fits the background well, and continues to as we add more data. Parametric function no longer fits well

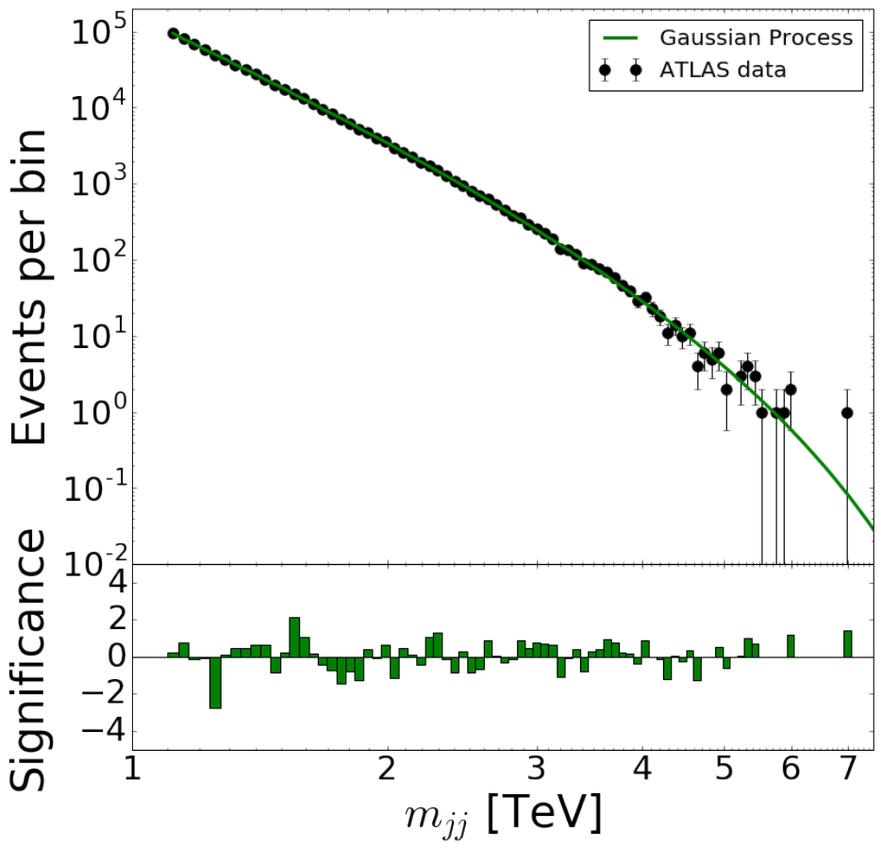
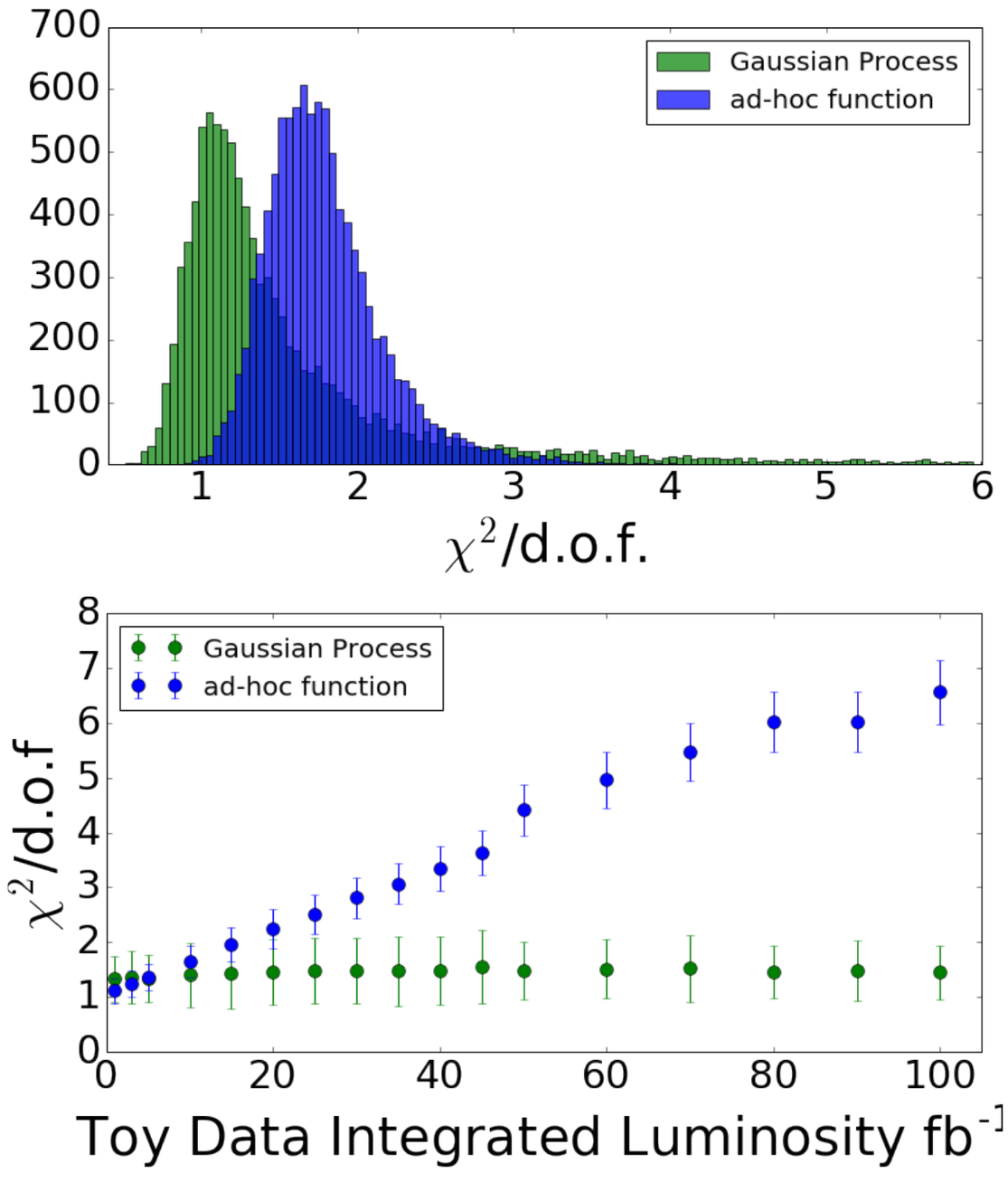


FIG. 5: Invariant mass of dijet pairs reported by ATLAS [15] in proton-proton collisions at $\sqrt{s} = 13$ TeV with integrated luminosity of 3.6 fb^{-1}. The green line shows the resulting Gaussian process background model. The bottom pane shows the significance of the residual between the data and the GP model.

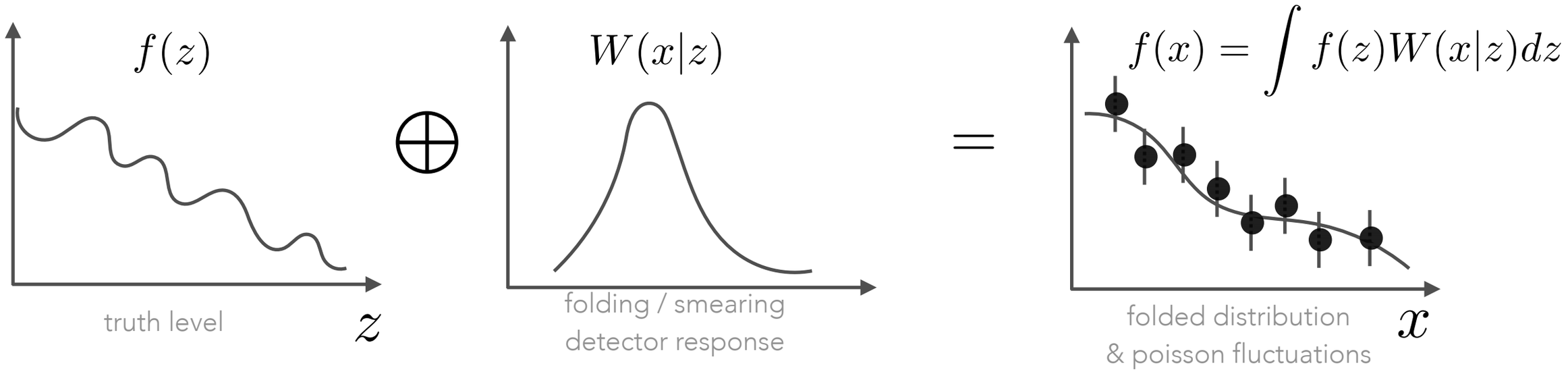


Physically Motivated Kernels

CONNECTION TO UNFOLDING

If the truth level distribution f(z) is a Gaussian Process with kernel $\Sigma(z, z')$, then the reconstructed distribution f(x) is also a Gaussian process with $\Sigma(x,x')$

$$\Sigma(x, x') = \int\int dz dz' \Sigma(z, z') W(x, z) W(x', z') \qquad (8)$$



If we are making predictions with Monte Carlo, truth level distribution f(z) is usually known exactly.

To think of f(z) as a Gaussian Process, we need some notion of uncertainty (eg. parton density functions, higher-order corrections, renormalization/factorization scales)

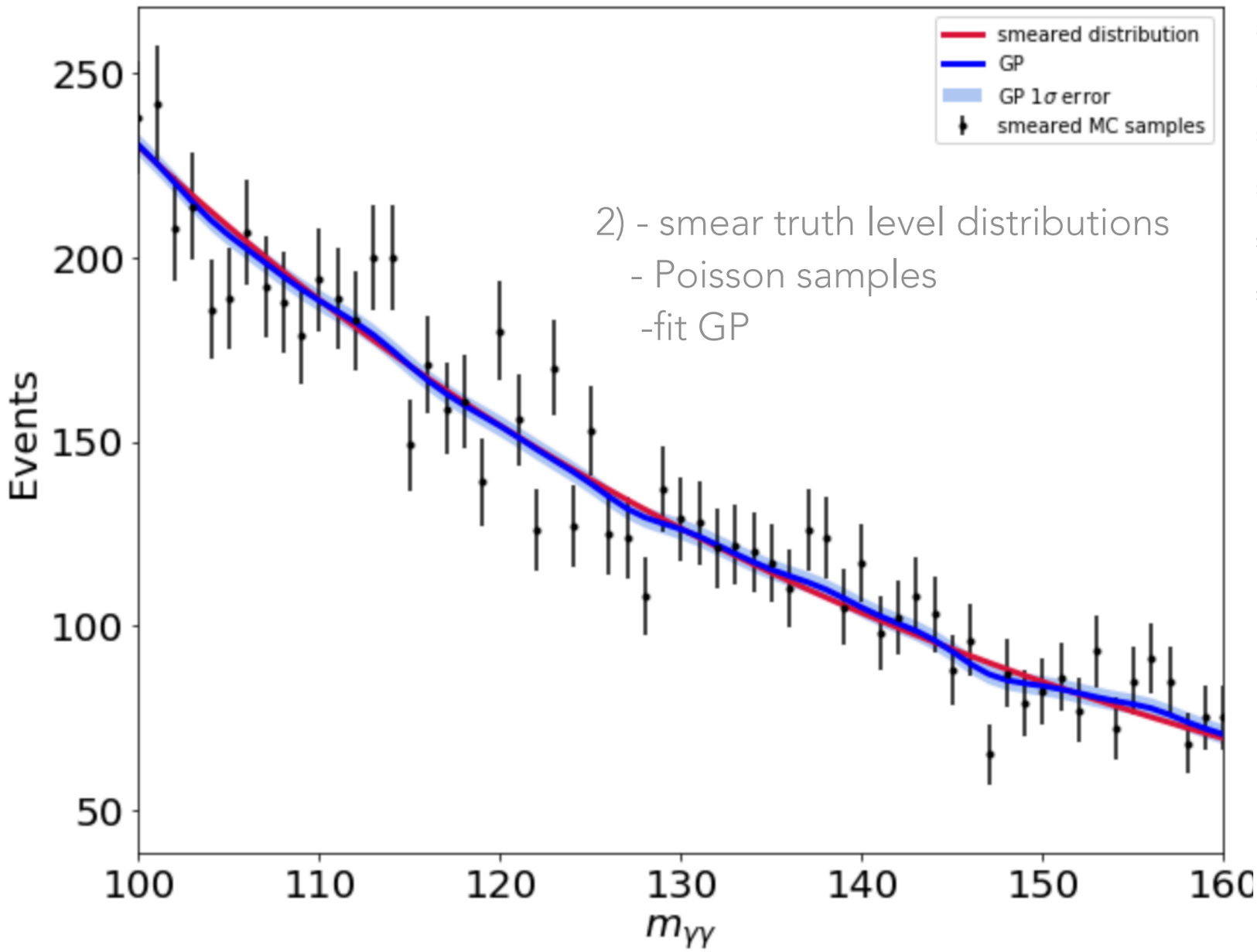
In unfolding, we often don't want to make assumptions about f(z)… it could be anything. But regularization in unfolding is equivalent to choosing a kernel for f(z).

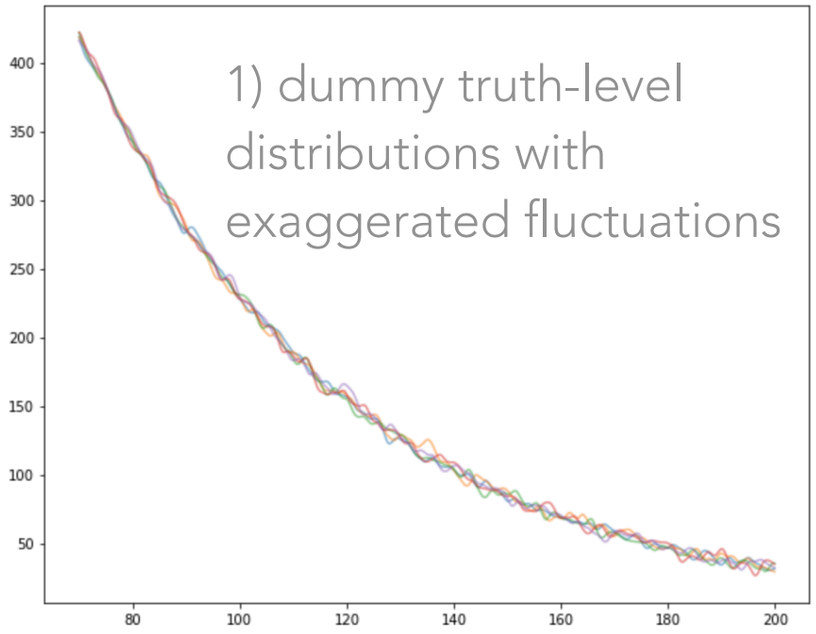
Even in extreme case where we assume no smoothness in f(z), f(x) has to be smooth due to detector resolution.

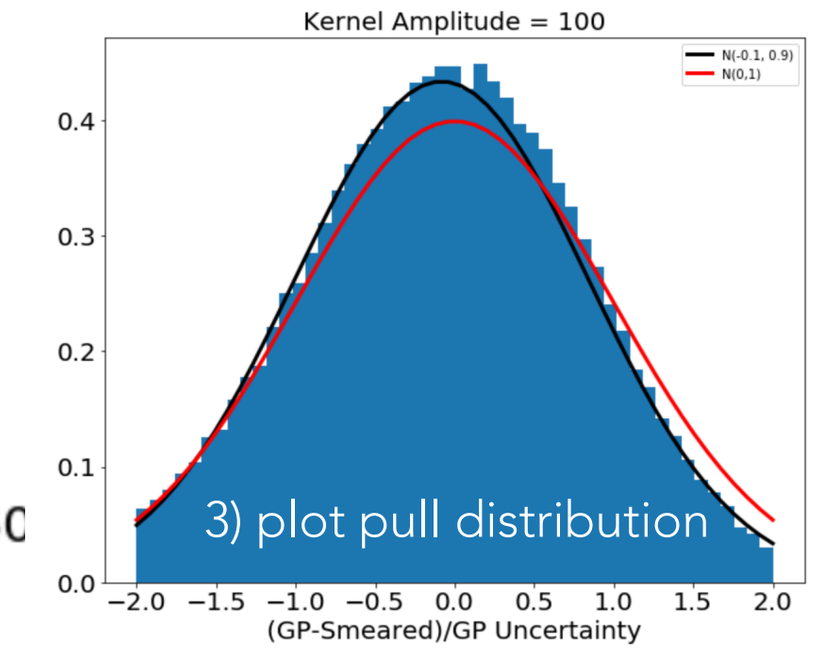
MC-TEMPLATE SMOOTHER

In H→γγ, we have used functional forms, like Bernstein polynomial. We "trust" the Monte Carlo, and assign "spurious signal" to account for differences between MC and functional form, but MC Stat error is a limiting factor for spurious signal etc.

Alternate idea: fit GP to MC histogram. No functional form assumed. Here only assume length scale must be > √2 mass resolution.

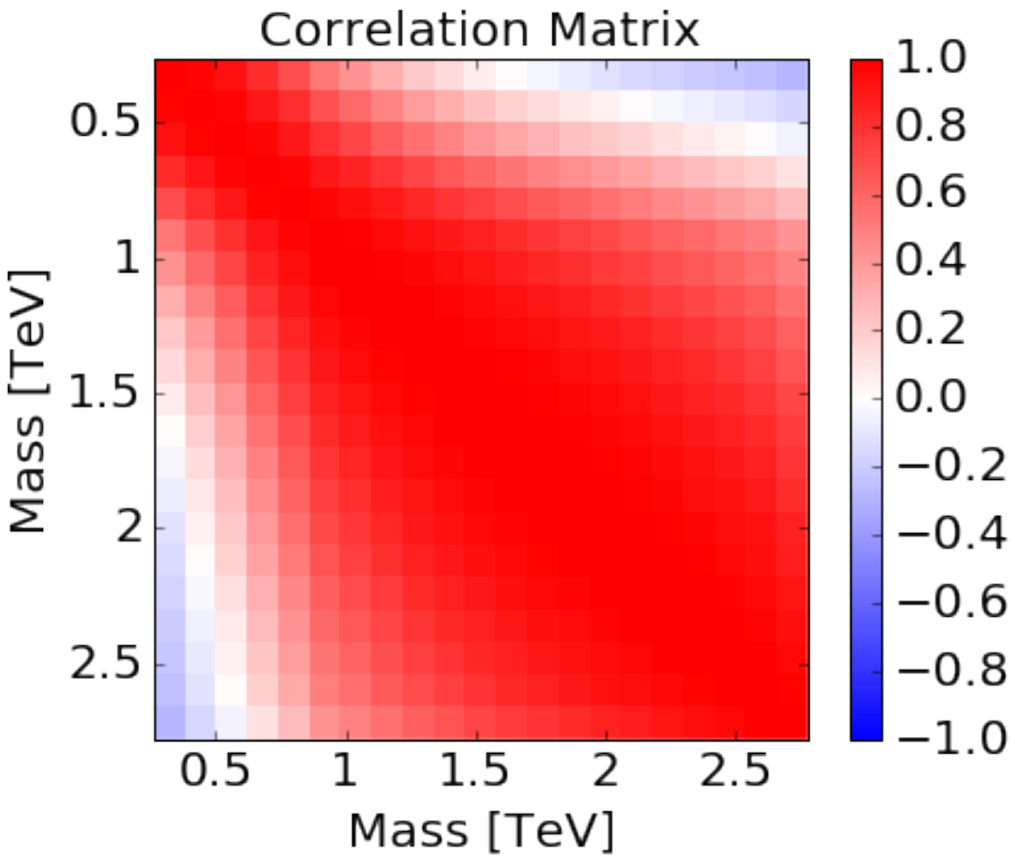




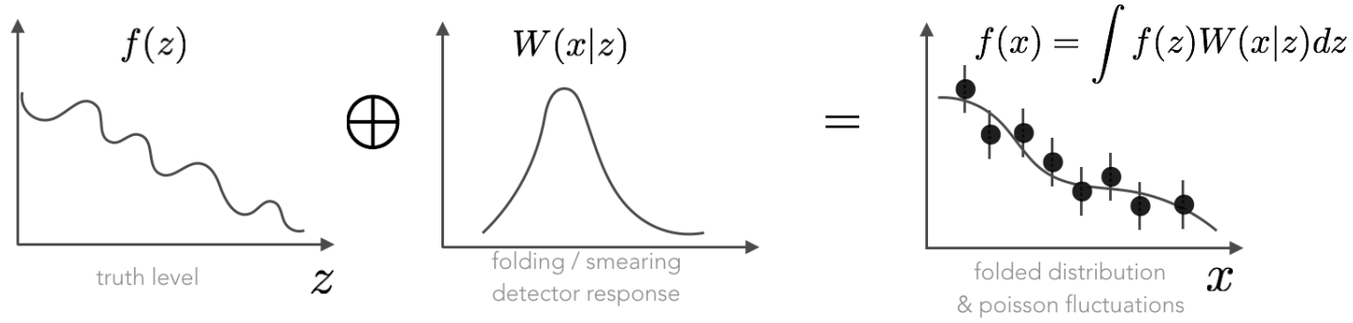


https://github.com/mfrate28/gp-mc-template-smoother/blob/master/GPsmearedFitting.ipynb

EXAMPLE: PDF UNCERTAINTIES



$$\Sigma(x,x') = \int\int dz dz' \Sigma(z,z') W(x,z) W(x',z') \qquad (8)$$

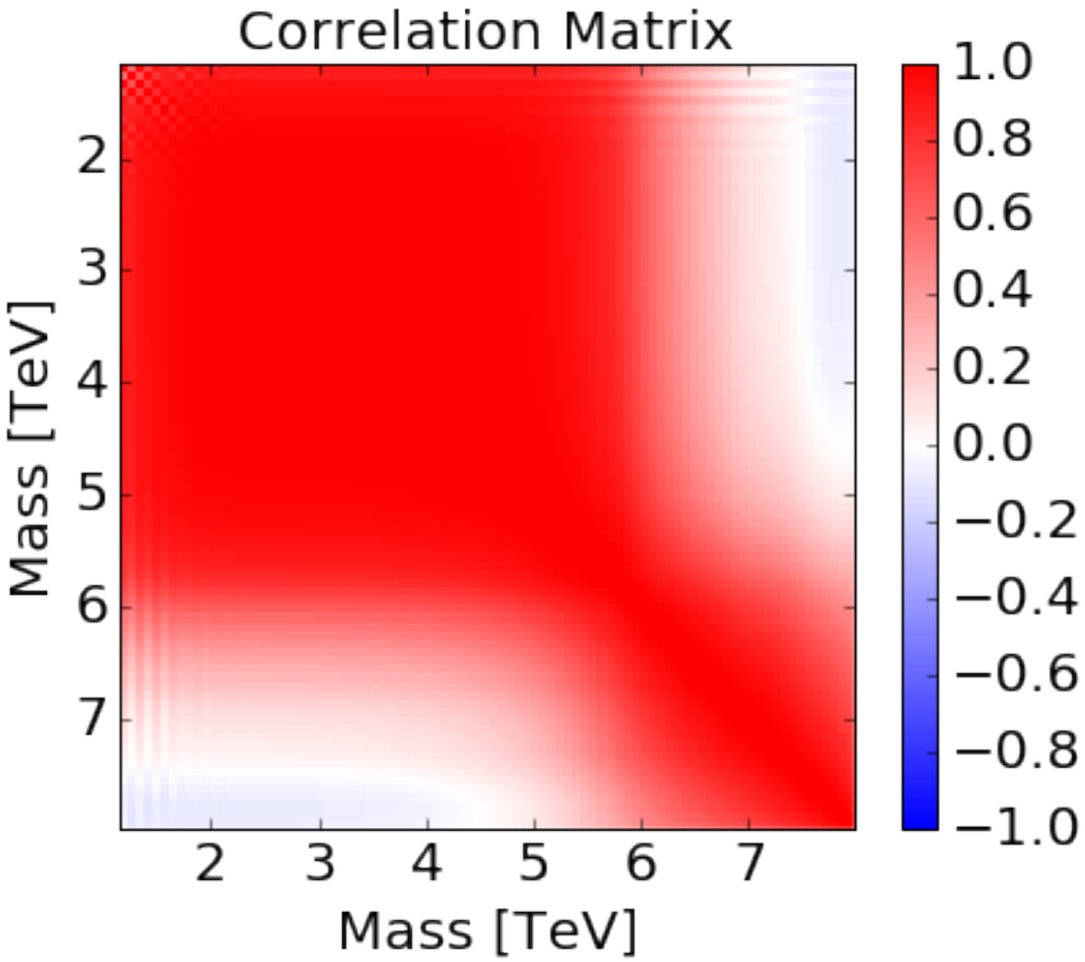


Here we focused on truth level distribution f(z).

Used dijet spectrum predicted at NLO with POWHEG-BOX and look into PDF uncertainties from NNPDF3

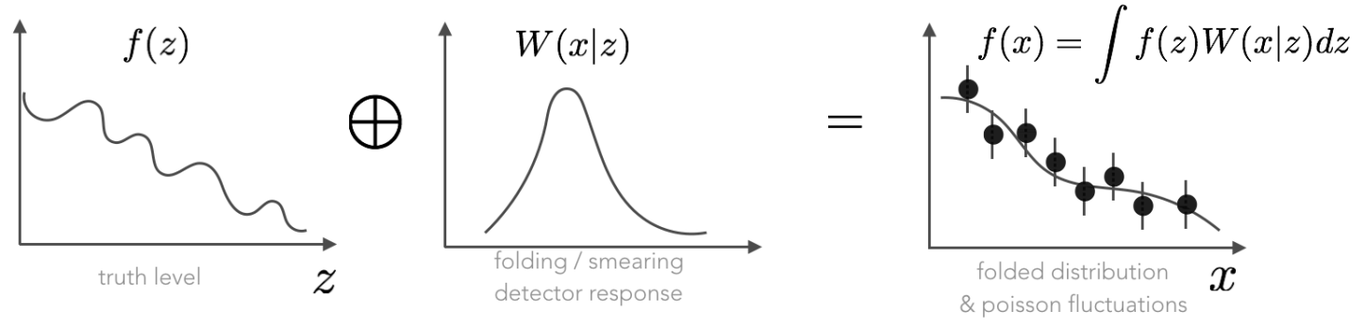
This is the PDF uncertainty in the truth distribution expressed as a Gaussian Process Kernel.

EXAMPLE: JET ENERGY SCALE



Take for example, the jet energy scale (JES) uncertainty. As described in Refs. [17, 27] the ATLAS JES uncertainty is only a few percent for jets with p_{T} of around 1 TeV where data are plentiful, while the the limited size of observed examples for higher-p_{T} jets requires an alternate approach to estimating the JES. The resulting JES uncertainty therefore grows rapidly with m_{jj} and has an impact of at most 15% [27]. To illustrate the covariance due to the JES uncertainty, consider a simplified two-parameter model for the impact on the m_{jj} distribution: $\mathrm{J}(z,\theta) = 1 + 15\%\,\theta_1 z^4 + 5\%\,\theta_2(1-z)$, where z is the true dijet invariant mass and $z_{\mathrm{max}} = 7$ TeV. We use the best fit 3-parameter fit as a proxy for $f(z)$ and fold in the smearing $W(x|z,\theta) = \mathrm{Gaus}(x|z\,\mathrm{J}(z/z_{\mathrm{max}},\theta),\sigma_x)$, where $\sigma_x = 2\%z$ is the dijet invariant mass resolution [17]. By assuming a uniform prior and an appropriate scaling for θ, we sample from the posterior $\mathrm{Gaus}(\theta_1|0,1)\mathrm{Gaus}(\theta_2|0,1)$ and propagate the uncertainty in θ through to the predicted bin counts $\bar{f}(\mathbf{x}|\theta)$ as in Eqs. 4 and 5. This allows us to explicitly build the covariance matrix Σ using the simulation shown in Fig. 3. As expected, we see a roughly block-diagonal structure defined by low and high mass regions.

$$\Sigma(x,x') = \int\int dz dz' \Sigma(z,z') W(x,z) W(x',z') \qquad (8)$$

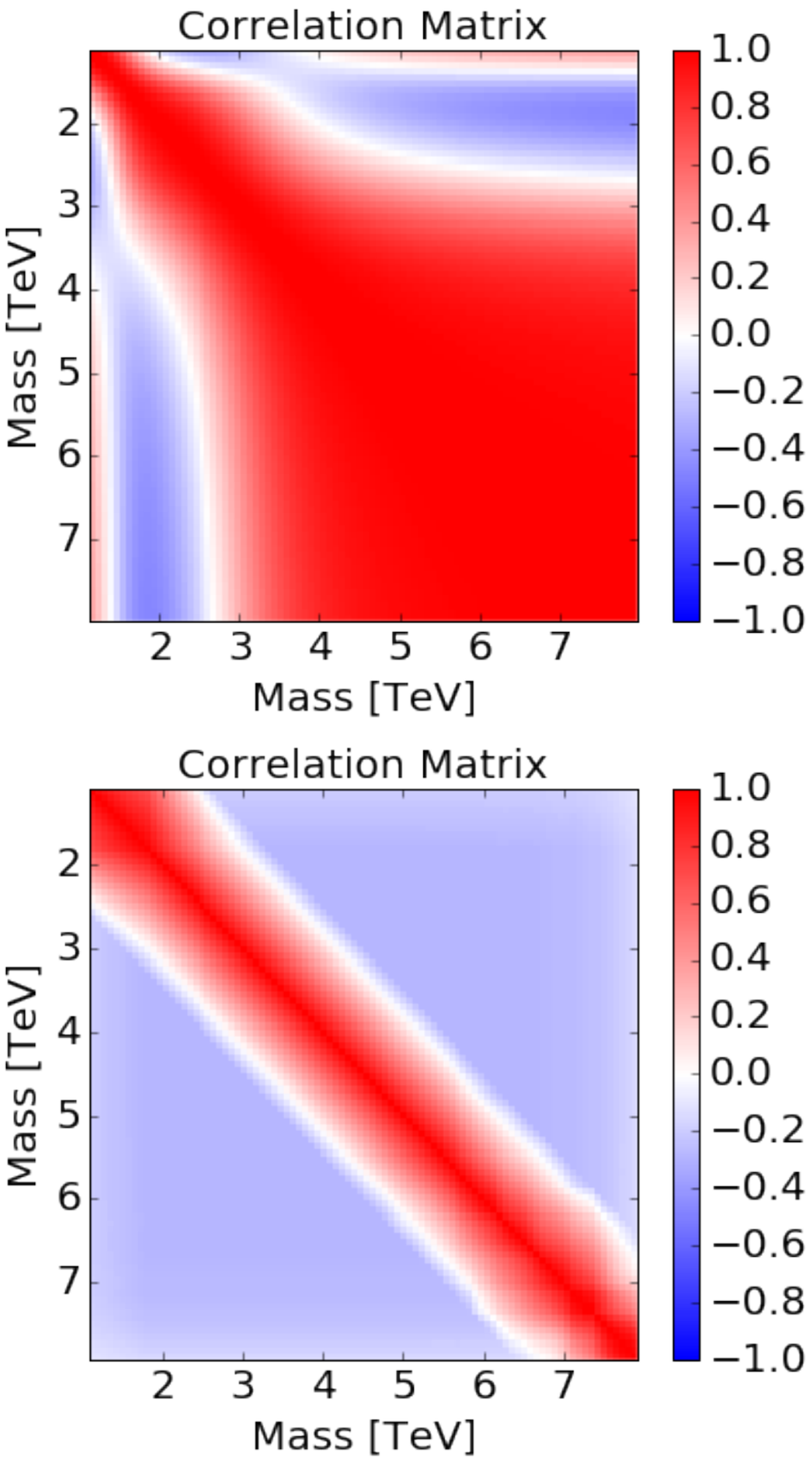


Even if the truth-level distribution is known, the folding matrix may not be known exactly.

Example: consider a jet energy scale with 2 nuisance parameters, where one parameter dominantly affects low-pT jets (in situ) and the other high-pT jets (limited stats for in situ).

Propagate uncertainty in jet energy scale to reconstructed mjj spectrum, obtain covariance kernel.

EXAMPLE: TRADITIONAL DIJET

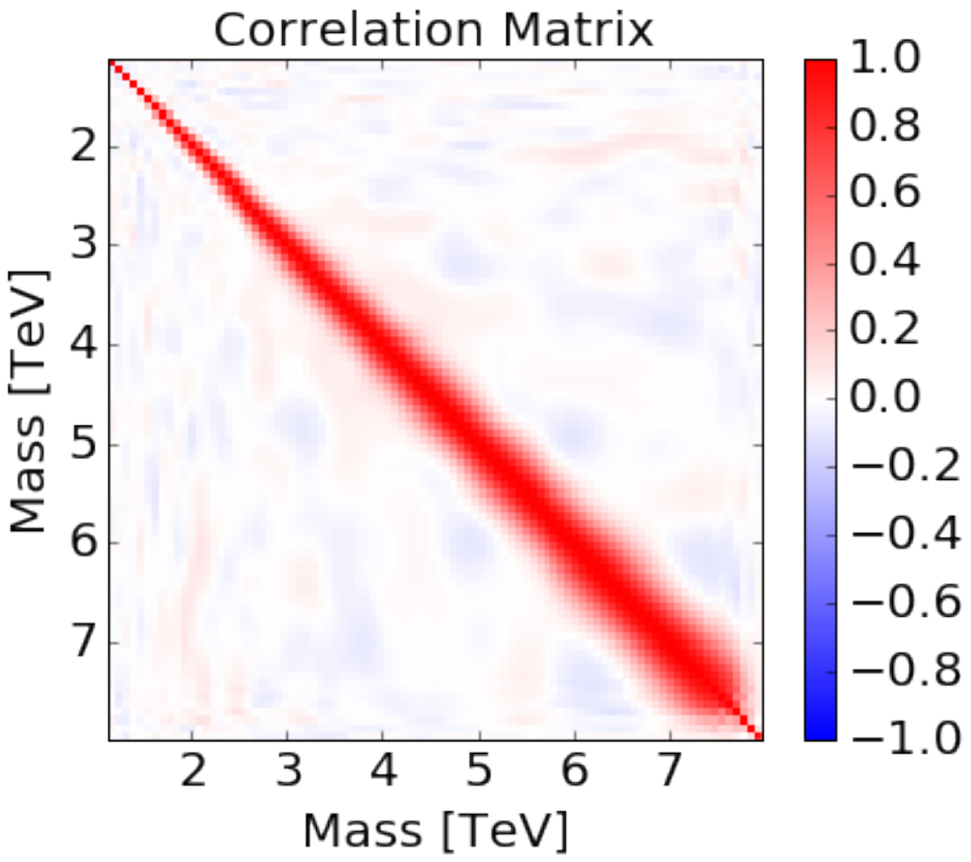


We can also think of the covariance structure for current fitting strategies.

- top: 3-parameter dijet function
- bottom: sliding window (SWIFT)

These are post-fit covariance plots.

POST-FIT PARAMETRIZED DIJET KERNEL



$$\Sigma(x,x') = Ae^{\frac{d-(x+x')}{2a}}\sqrt{\frac{2l(x)l(x')}{l(x)^2+l(x')^2}}e^{\frac{-(x-x')^2}{l(x)^2+l(x')^2}}$$

In addition to kernels constructed bottom-up from first-principles, we can also construct parametrized kernels using some intuition.

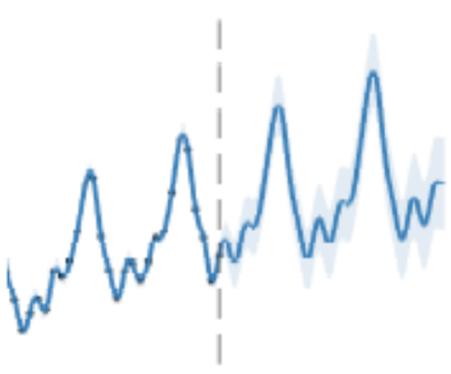
GPs adapt to the data very well, so even simple exponential-squared kernels often work fine.

For our dijet studies, we used a "Gibbs kernel", which has length scale $l(x)$ and amplitude vary with x

- plot shows post-fit covariance kernel

Vocabulary of kernels + grammar for composition

- physics goes into the construction of a “Kernel” that describes covariance of data

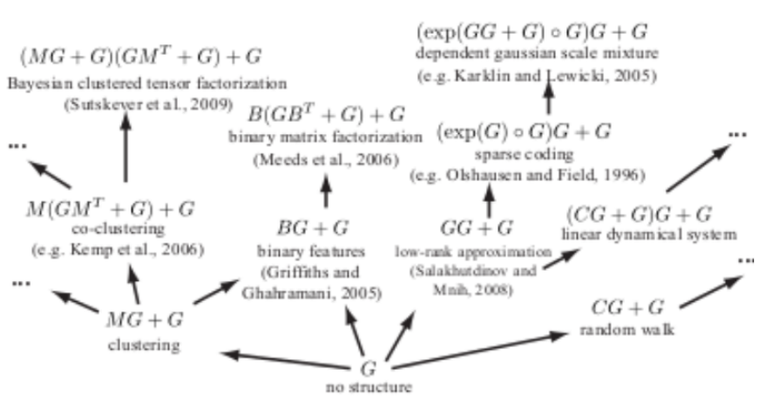


Structure Discovery in Nonparametric Regression through Compositional Kernel Search

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani

International Conference on Machine Learning, 2013

pdf | code | poster | bibtex

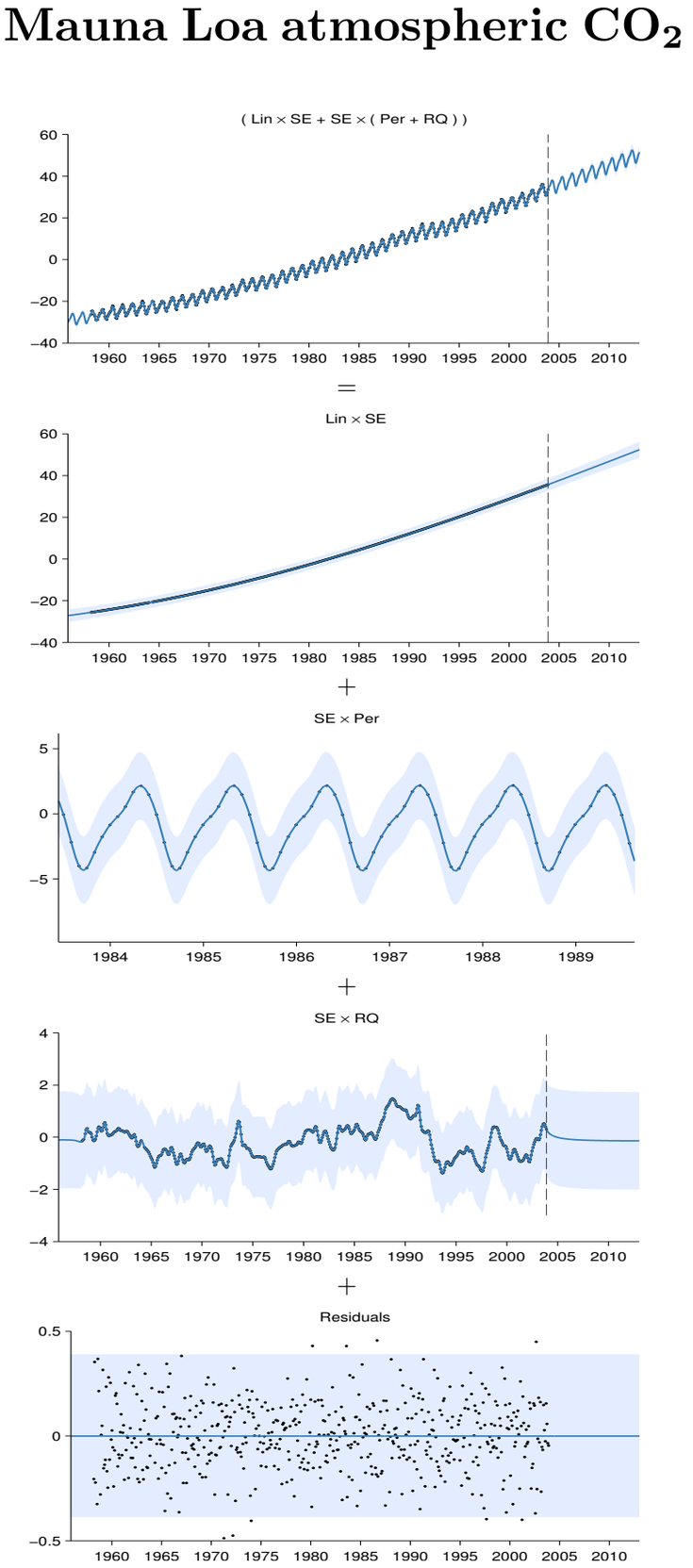


Exploiting compositionality to explore a large space of model structures

Roger Grosse, Ruslan Salakhutdinov, William T. Freeman, Joshua B. Tenenbaum

Conference on Uncertainty in Artificial Intelligence, 2012

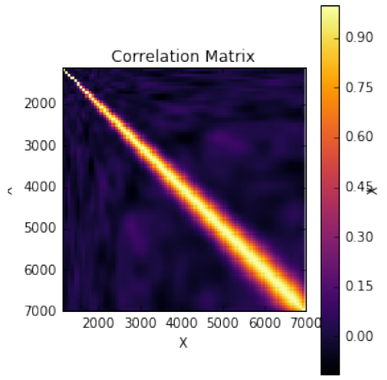
pdf | code | bibtex



FUTURE DIRECTIONS

with Meghan Frate
& Daniel Whiteson

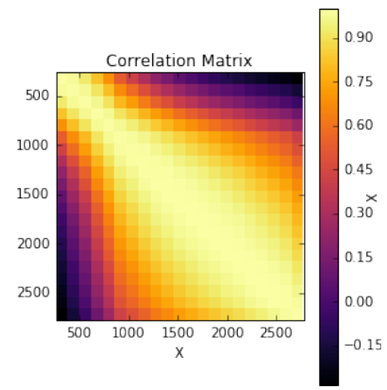
Instead of fitting the dijet spectrum with an ad hoc 3-5 parameter function, use GP with kernel motivated from physics



Final Kernel =

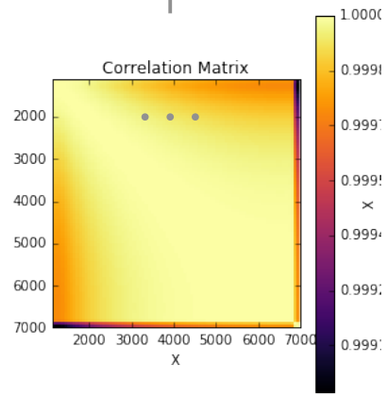
Poisson stats
+ Mass Resolution

=



+ Parton Density Functions

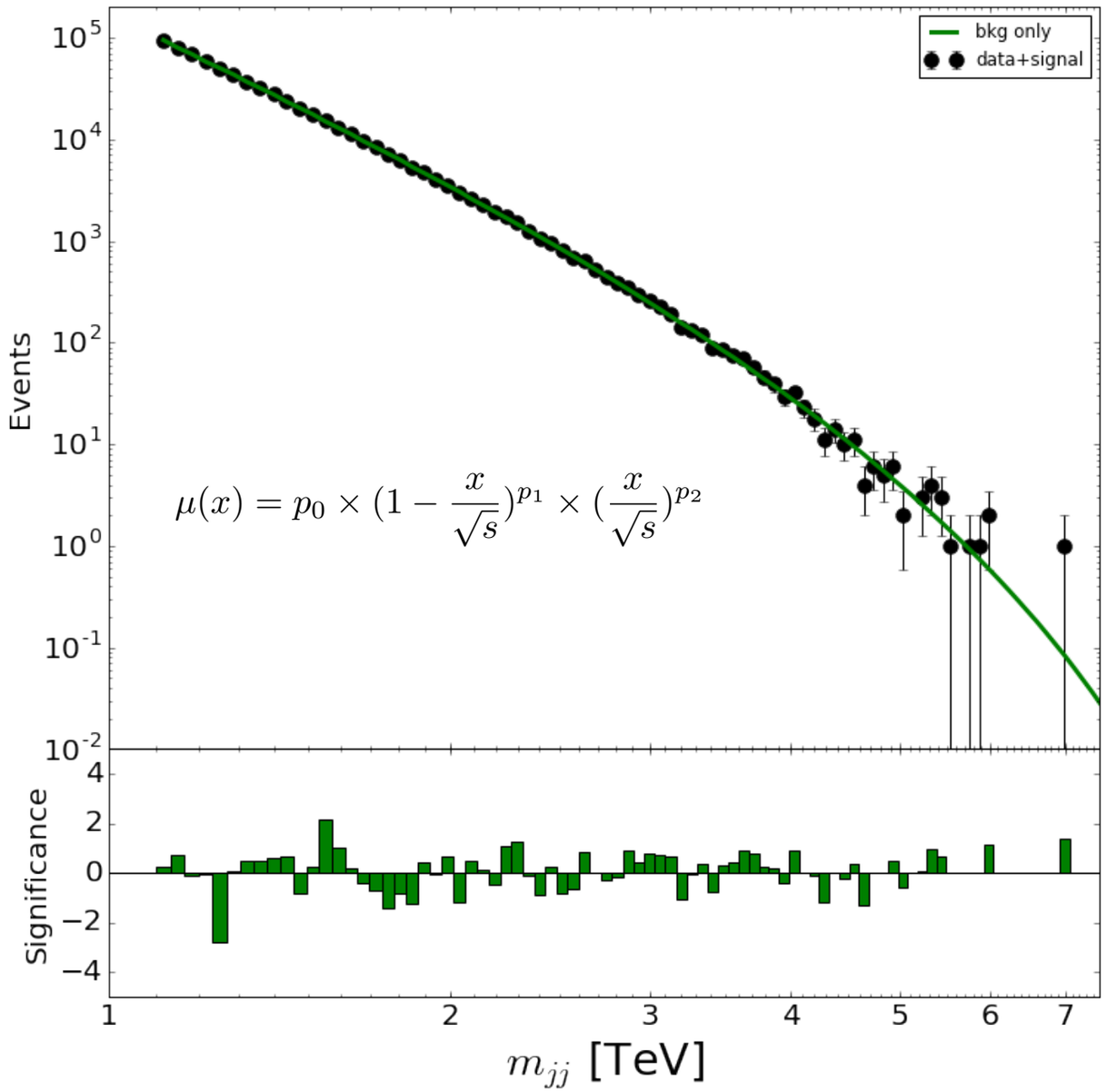
+



+ Jet Energy Scale

+

. . .



Integration into our Statistical Procedures

BAYES VS. FREQUENTIST

The statistical interpretation of GPs can be a bit subtle. Specifically Bayesian vs. Frequentist issues

- Most GP literature is presented in a Bayesian formalism
- the GP is usually thought of as a prior over functions, and the result of the fit is posterior given observations
- then usually fit “hyperparameters” of kernel using marginal likelihood

However, also consistent to think of the GP likelihood where the kernel represents auxiliary measurements / constraint terms.

A third interpretation is that the kernel represents the penalty term in “penalized maximum likelihood” in the spirit of regularization in unfolding

INTEGRATION INTO OUR STATISTICAL PROCEDURES

Integration of GPs into our statistical procedures can be done in a few ways.

- start with our typical extended maximum likelihood for a statistical model parametrized by θ

$$p(\mathcal{D}, \mathbf{a}|\theta) = \text{Pois}(N|\nu(\theta)) \prod_{e=1}^{N} p(x_e|\theta) \cdot p_{\text{constr.}}(\mathbf{a}|\theta) \ .$$

- If we are using a binned distribution in a high-statistics regime, and we approximate the **effect** of the constraint terms on the bin counts as a Gaussian, then we can approximate this as

$$\begin{aligned} p(\mathbf{y}, \mathbf{a}|\theta) = & \quad \prod_{i=1}^{n} \text{Pois}(y_i|\bar{f}(x_i|\theta)) \cdot p_{\text{constr.}}(\mathbf{a}|\theta) \\ \approx & \ \text{Gaus}(\mathbf{y}|\bar{f}(\mathbf{x}|\theta), \sigma^2) \cdot \text{Gaus}(\bar{f}(\mathbf{x}|\theta)|\mu, \mathbf{\Sigma}) \ , \end{aligned}$$

- The Poisson mean $\bar{f}(\mathbf{x}|\theta)$ can be a parametrized signal + a Gaussian Processs for the background.

INTEGRATION INTO OUR STATISTICAL PROCEDURES

Integration of GPs into our statistical procedures can be done in a few ways.

1. Fully Bayesian analysis using Poisson fluctuations about a GP mean. This is called a Cox process. Cumbersome to implement because it is "doubly stochastic"

2. Fit the total model (parametrized signal + background GP) to the data (assuming stat errors are Gaussian), use result as the mean $\mu(\mathbf{x}_*|\mathbf{y}) = \mu(\mathbf{x}_*)+\Sigma(\mathbf{x}_*,\mathbf{x})[\Sigma(\mathbf{x},\mathbf{x})+\sigma^2(\mathbf{x})\mathbf{I}]^{-1}(\mathbf{y}-\mu(\mathbf{x}))$ in standard likelihood Poisson likelihood

3. Fit the GP, use the posterior mean and covariance of the GP as a simple Gaussian likelihood / chi-square with covariance matrix.

We used option 2., most consistent with our existing statistical procedures

HYPOTHESIS TESTING

Here true hypothesis has no signal, but is neither the ad-hoc function nor the GP, so we don't expect it to be a chi-square exactly.

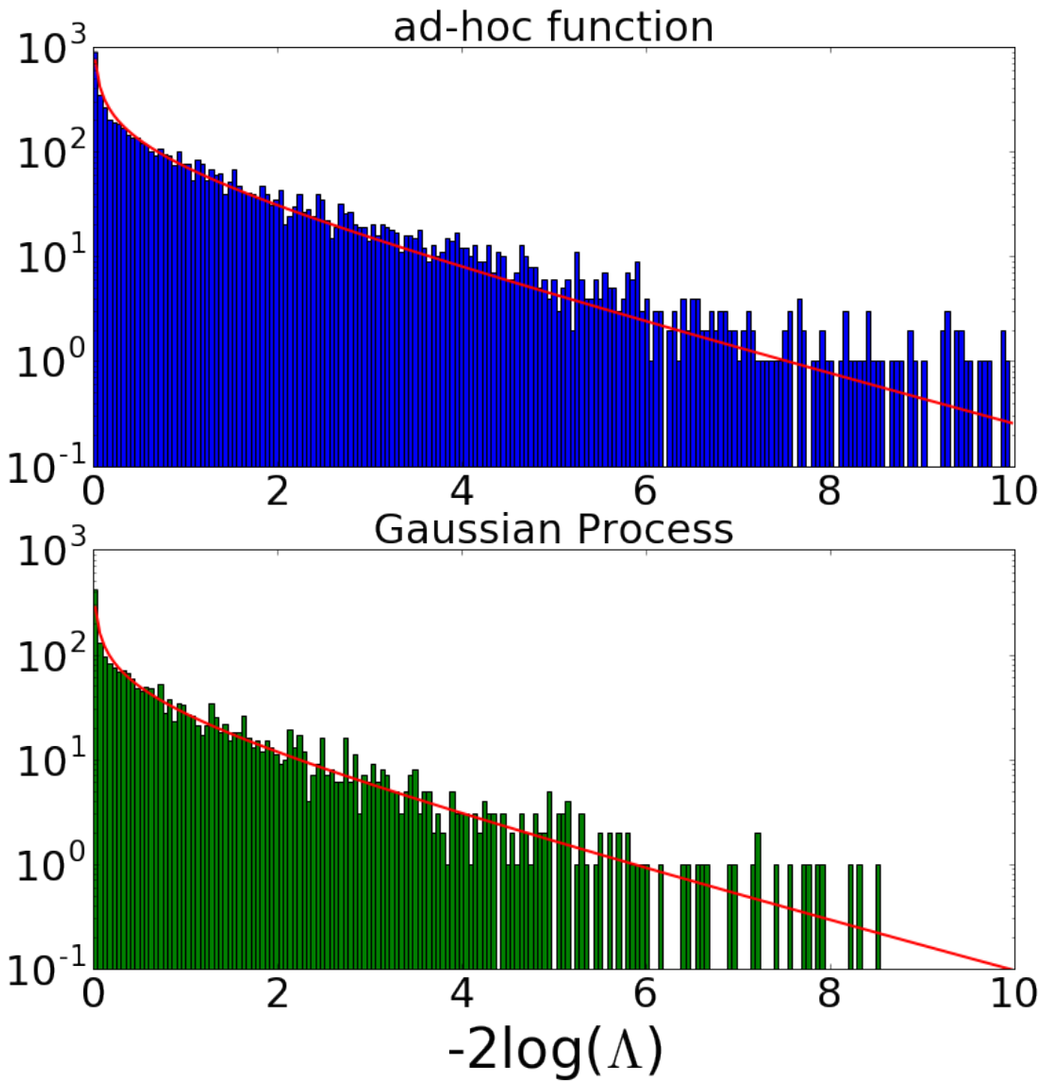
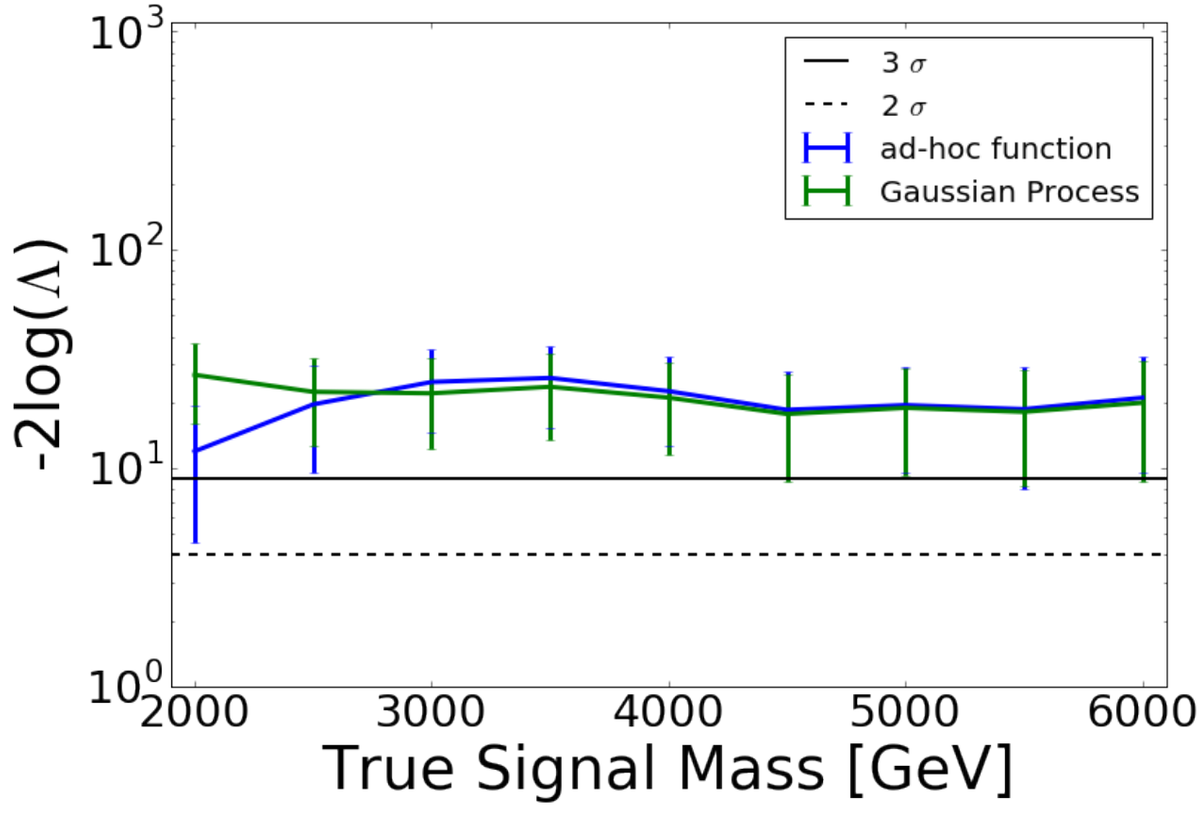


FIG. 11: Distribution of $-2\log(\Lambda)$, where Λ is the likelihood ratio between the background-only and the background-plus-signal hypotheses, for toy data with no signal present, shown for both the ad-hoc fit (top) and the Gaussian process background model (bottom). Overlaid in red is a χ^2 distribution with one degree of freedom.

Worry is GP might be too flexible.

So need to check expected significance (power) by injecting signal.

(Result depends on kernel used)



Modeling Generic Localized Signals

(related to spurious signal)

BUMPHUNTER

In many exotics searches, we don't want to assume a specific signal model.

- difficult to do likelihood-ratio based tests using shape information, since we don't know the signal's shape

- Instead, typically use **BumpHunter** and look for a localized signal in some mass window.

- difficulties here because BumpHunter needs a global background estimate to do background-only toys to correct for look elsewhere effect

- If we are fitting background from data, this is circular do we do this?

AN ALTERNATIVE

An alternative is to use a **GP for the signal**

- Use a kernel that looks for an excess only in a localized excess in a window around mass m with width t (keeping length scale l for smoothness)

$$\Sigma(x, x') = Ae^{-\frac{1}{2}(x-x')^2/l^2} e^{-\frac{1}{2}((x-m)^2+(x'-m)^2)/t^2}, \quad (14)$$

- Now we have a signal shape, so we can do likelihood-ratio tests between signal and background

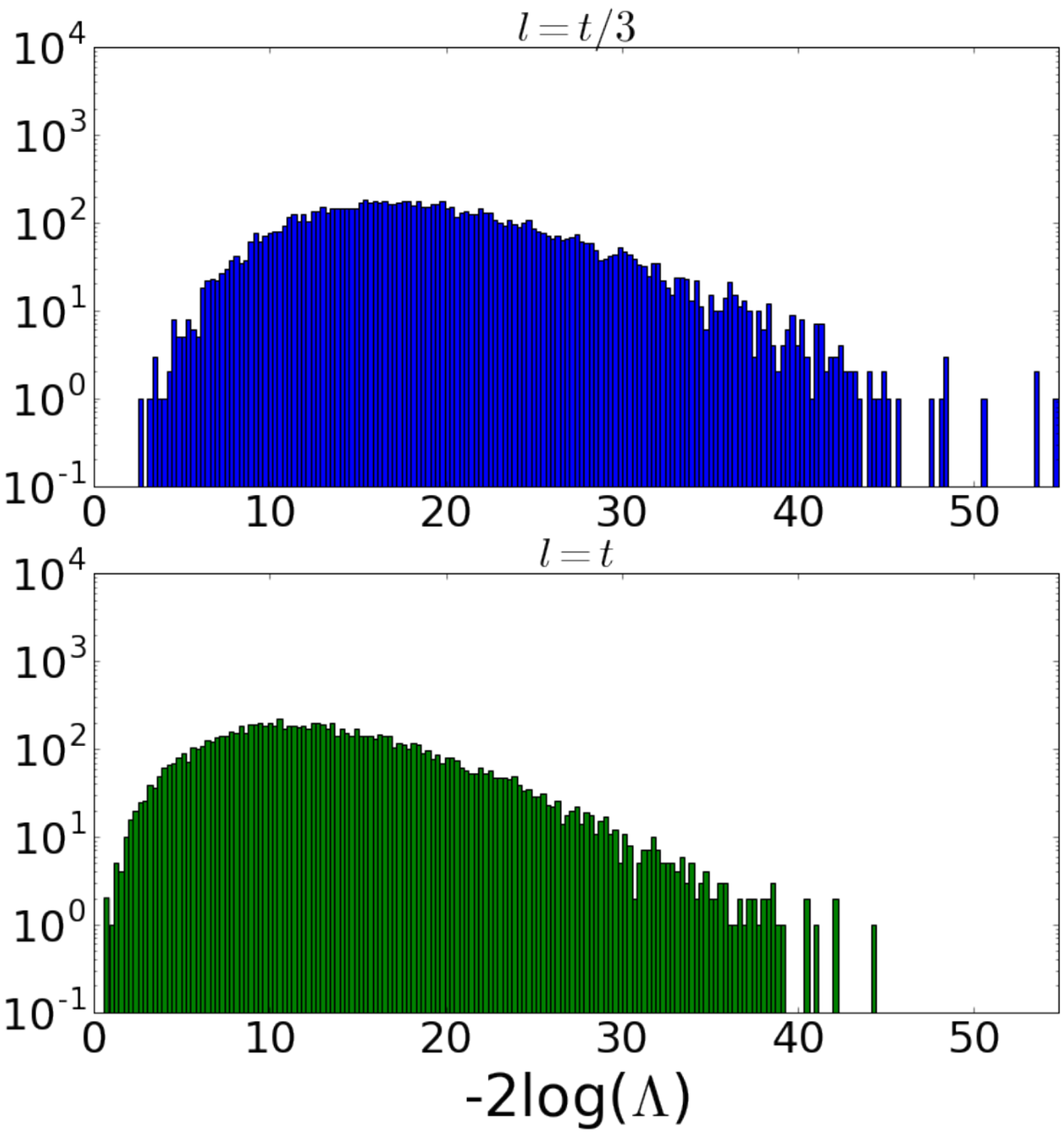
The issue now is that the signal has many free parameters, so these tests will have a look-elsewhere effect.

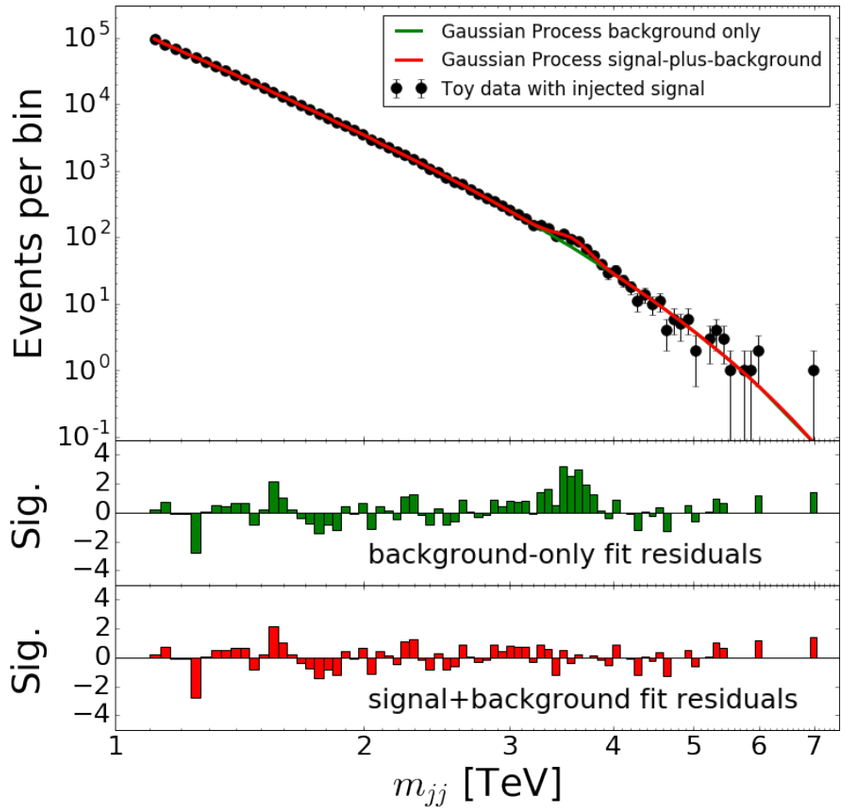
- this isn't a problem though, we still do background-only fits to get the "global p-value"

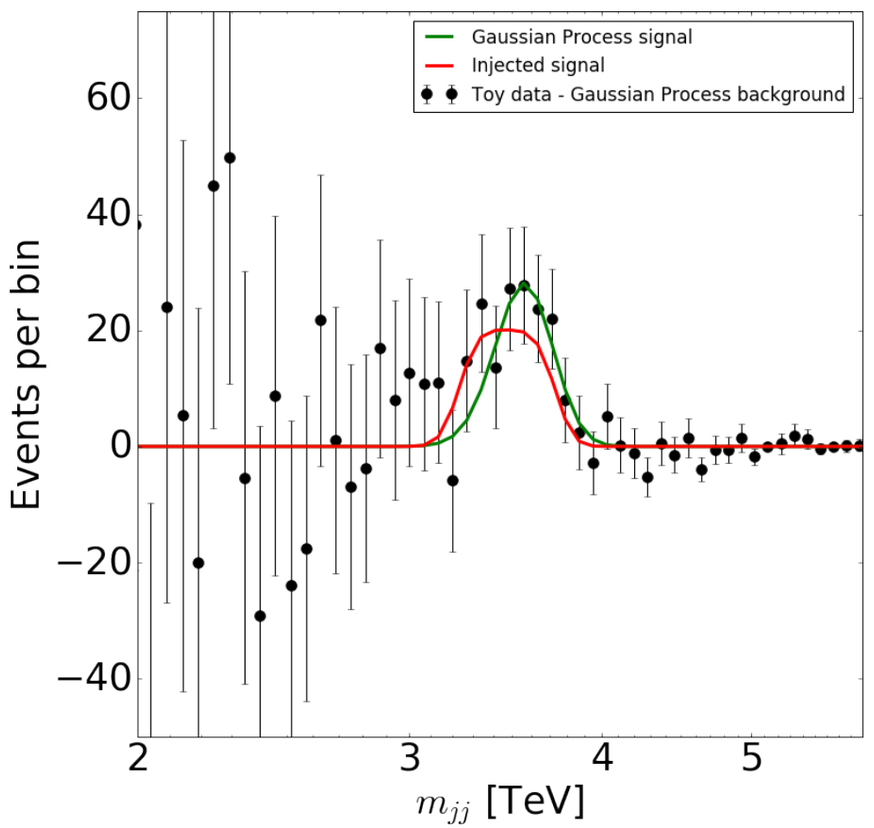
LOOK-ELSEWHERE EFFECT

The plot below shows 2logΛ(μ=0) for the background-only.Use this for global p-value.

(depends on kernel hyper parameters)







Software & Examples

SOFTWARE

You don’t need to do this yourself, there’s many Gaussian Process packages that do this for you

- See github.com/mfrate28/ComparingGPpackages for a comparison of GP packages

Meghan worked on some tutorials that help with common HEP use cases

- https://github.com/mfrate28/GP_Tutorial

We are investigating a RooFit interface.