Bayesian Data Analysis

and some other things

A. Caldwell
Max Planck Institute for Physics

1. Another example – fitting an energy spectrum
2. Frequentist intervals and Bayesian intervals for Poisson process
3. P-values; definitions and pitfalls
4. BAT

Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

MAX-PLANCK-GESELLSCHAFT

Example-energy spectrum

Suppose we make a measurement of an energy with a calorimeter. What can we say about the ‘true’ value ? If we assume a flat prior, we get

$$P(E_0|E) = P(E|E_0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(E_0 - E)^2}{2\sigma^2}}$$

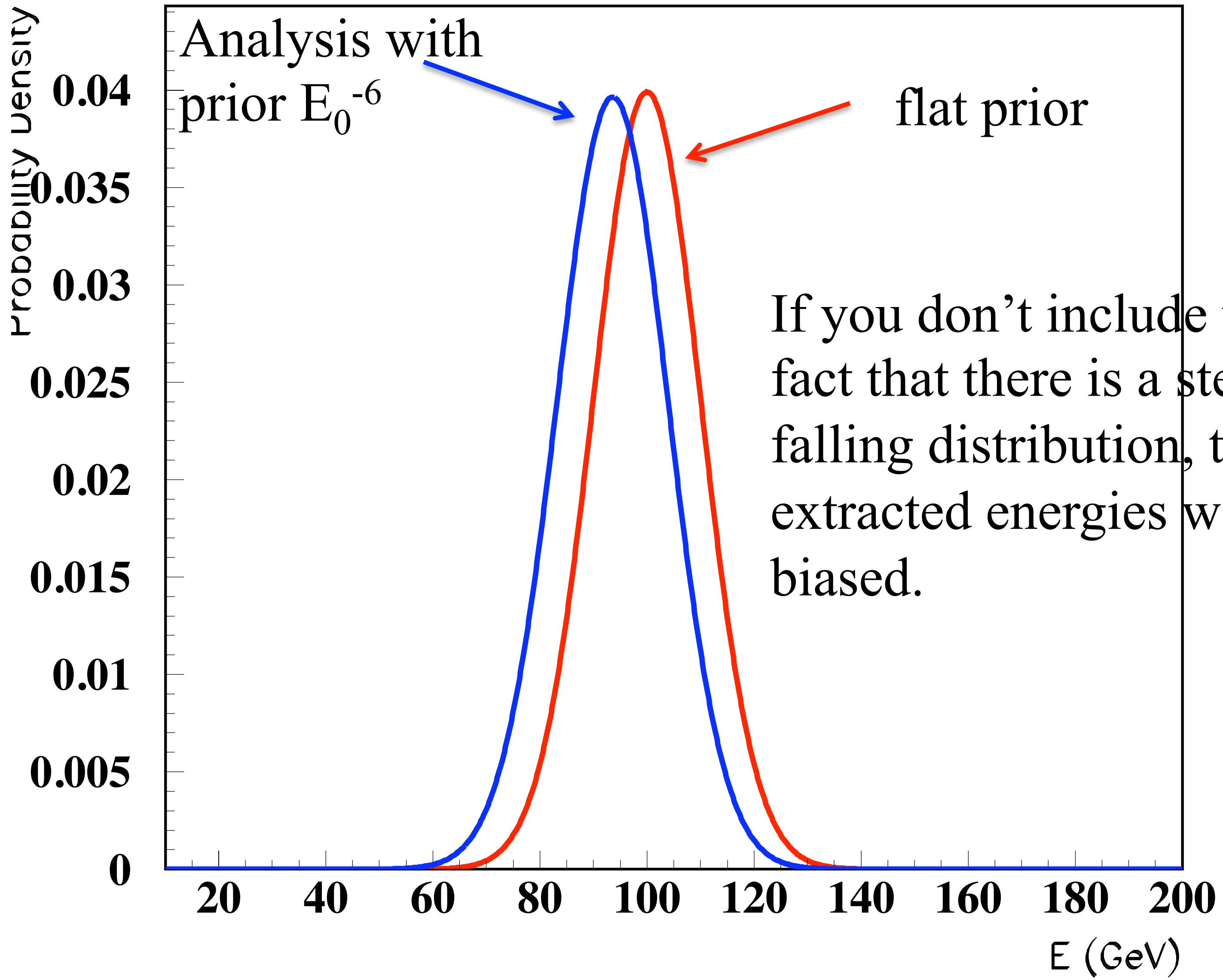
The probability distribution for the true energy is a Gaussian centered on the measured value. However, energy distributions often have a steep distribution. Suppose the starting distribution was

$$f(E_0) \propto E_0^{-6}$$

then

$$f(E) \propto \int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(E_0 - E)^2}{2\sigma^2}} E_0^{-6} dE_0$$

one measurement of the energy, resolution 10 GeV, measured 100 GeV



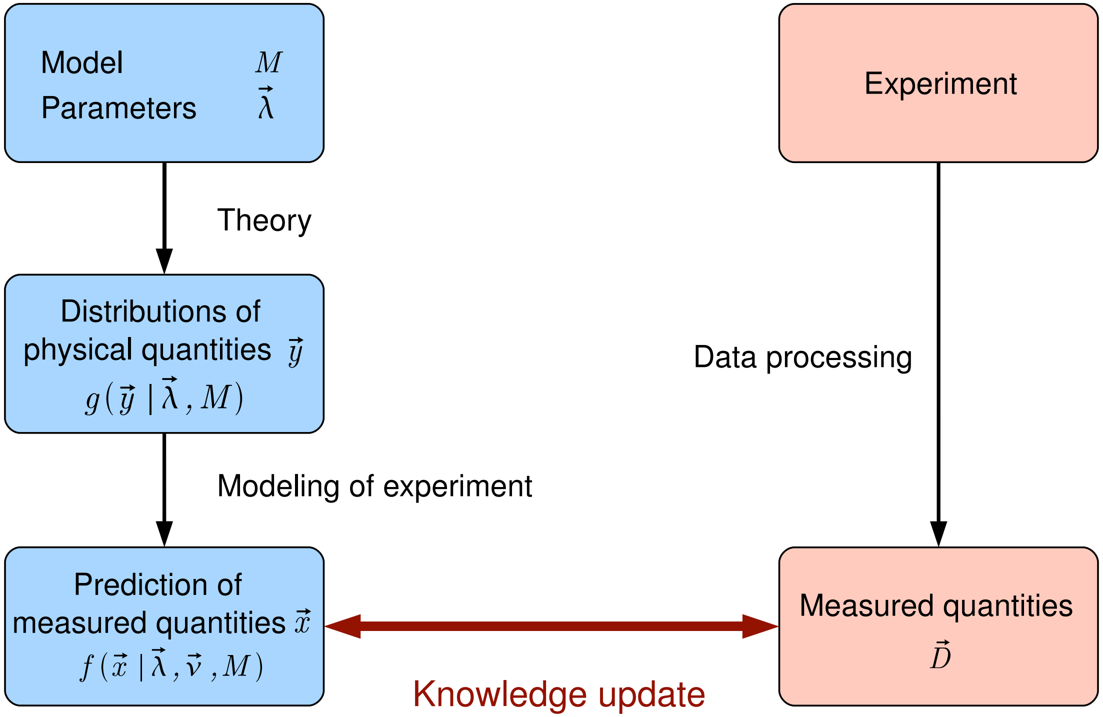
Analysis with prior E_0^{-6}

flat prior

If you don’t include the fact that there is a steeply falling distribution, the extracted energies will be biased.

Power for Energy Spectrum

Suppose what we are trying to extract is the power of the underlying energy distribution. How would we proceed ?


Model M
Parameters $\vec{\lambda}$
Theory
Distributions of physical quantities $\vec{y}$
$g(\vec{y}\,|\,\vec{\lambda}, M)$
Modeling of experiment
Prediction of measured quantities $\vec{x}$
$f(\vec{x}\,|\,\vec{\lambda}, \vec{\nu}, M)$
Experiment
Data processing
Measured quantities
$\vec{D}$
Knowledge update

In this case, assume $g(E_0|\lambda, M) \propto E_0^{-\lambda}$

Power example

We assume the measured values are related to the true as:

$$P(E|E_0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(E_0 - E)^2}{2\sigma^2}}$$

Now apply the 'law of total probability'

$$P(E|\lambda) = \int P(E|E_0) P(E_0|\lambda) dE_0$$

And Bayes' equation yields $P(\lambda|E) \propto \prod_i P(E_i|\lambda) P_0(\lambda)$

$$P(\lambda|E) \propto \left[\prod_i \int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(E_i - E_0)^2}{2\sigma^2}} E_0^{-\lambda} dE_0 \right] P_0(\lambda)$$

Power example

$$P(\lambda|E) \propto \left[\prod_i \int \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(E_i-E_0)^2}{2\sigma^2}} E_0^{-\lambda} dE_0\right] P_0(\lambda)$$

Need numerical approach.

1. Either integrate numerically many many times during parameter scan.

2. Make a histogram of expected number of entries in measured energy bins from your event simulation, then reweight the distribution for different values of λ and see how the agreement between expected and measured varies (Poisson statistics). Note that this does not use the equation above – in this case

$$P(E|\lambda) = \prod_{i=1}^{Nbins} \frac{e^{-\nu_i}\nu_i^{n_i}}{n_i!}$$

n_i Number of events in energy bin *i*

$\nu_i = \nu_i(\lambda)$ Expectation based on λ

Reweighting a Simulated Distribution

1. Generate events according to a reasonable pdf. In this case, interested in $f(E_0) \propto E_0^{-\lambda}$.

2. Smear the true energy to account for the apparatus resolution. Can also apply other constraints, e.g. lower thresholds on energy measurement, etc.

$$E = E_0 + \delta \qquad\qquad P(\delta|E_0) = \frac{1}{\sqrt{2\pi}\sigma(E_0)} e^{-\frac{1}{2}(\frac{\delta}{\sigma(E_0)})^2}$$

$$\sigma(E_0) = \sqrt{a^2 \cdot E_0 + b^2 \cdot E_0^2 + \sigma_n^2}$$

Reweighting Simulated Spectrum

Suppose now you wanted to simulate a distribution with a different power of λ. Can give the simulated events a weight

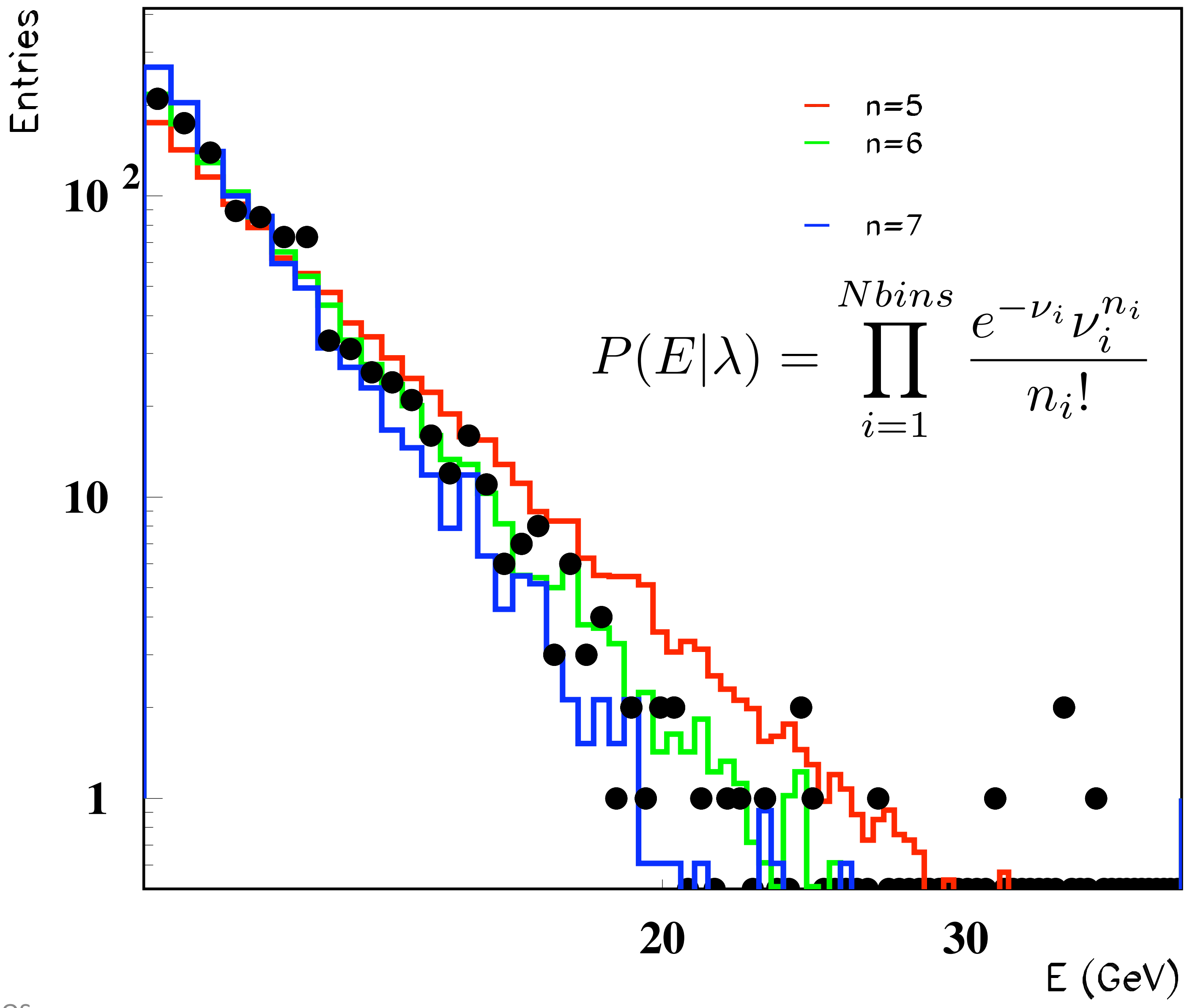
$$w(E_0) = \frac{f(E_0|\lambda')}{f(E_0|\lambda_{\rm gen})}$$

Statistical uncertainty (in the limit of a large number of events) behaves as

$$\sqrt{\sum_i w(E_{0i})^2}$$

Rule of thumb: avoid large weights (here, initial λ should not be too big) and make sure you have plenty of simulated events !

Power example



$$P(E|\lambda) = \prod_{i=1}^{Nbins} \frac{e^{-\nu_i}\nu_i^{n_i}}{n_i!}$$

Assumes no uncertainty on ν_i

Comparison of Bayesian Credible Intervals & Frequentist Confidence Level Intervals

Bayesian interval from cumulative of the Posterior pdf

Neymann Classical Interval – for each value of the parameter, find set of possible outcomes that contain at least 1-α probability. For the central interval and Poisson distribution:

$$n_1 = \sup_{n \in 0,...,\infty} \left\{ \sum_{i=0}^{n} P(i|\nu) \leq \alpha/2 \right\} + 1$$

$$P(n=0|\nu) > \alpha/2 \rightarrow n_1 = 0$$

$$n_2 = \inf_{n \in 0,...,\infty} \left\{ \sum_{i=n}^{\infty} P(i|\nu) \leq \alpha/2 \right\} - 1$$

$$\mathcal{O}^C_{1-\alpha} = \{n_1, \ldots, n_2\}$$

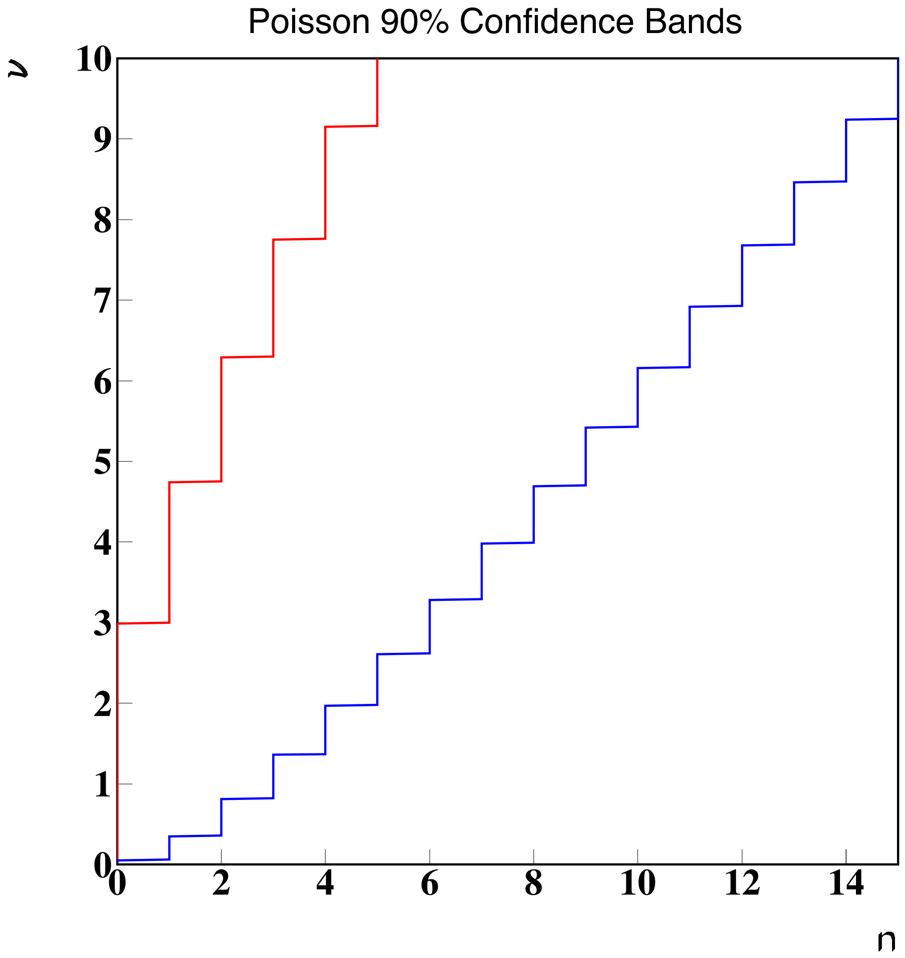
Poisson Example

Example for ν=10/3

n	$P(n\|\nu)$	$F(n\|\nu)$	R	$F_R(n\|\nu)$
0	0.0357	0.0357	7	0.9468
1	0.1189	0.1546	5	0.8431
2	0.1982	0.3528	2	0.4184
3	0.2202	0.5730	1	0.2202
4	0.1835	0.7565	3	0.6019
5	0.1223	0.8788	4	0.7242
6	0.0680	0.9468	6	0.9111
7	0.0324	0.9792	8	0.9792
8	0.0135	0.9927	9	0.9927
9	0.0050	0.9976	10	0.9976
10	0.0017	0.9993	11	0.9993
11	0.0005	0.9998	12	0.9998
12	0.0001	1.0000	13	1.0000

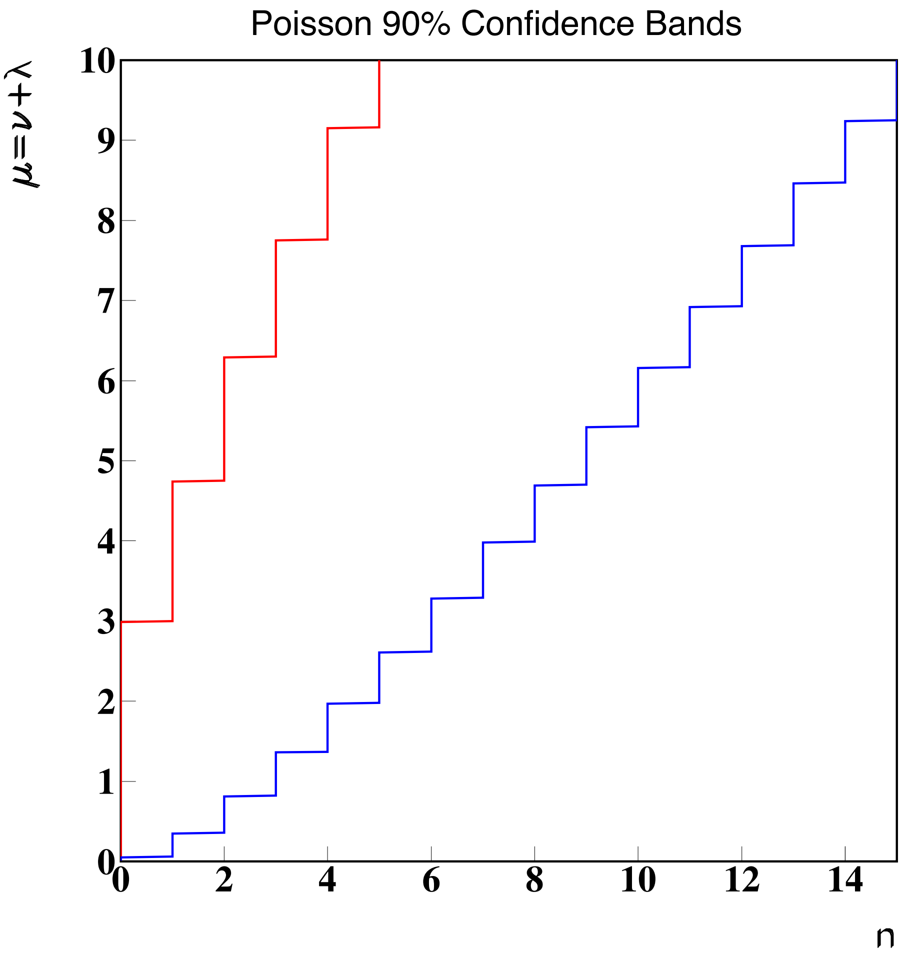
Confidence Level Calculation

We observe n events, and ask which values of ν are accepted with confidence level 1-α. For 1- α=0.9, central intervals:



Frequentist Statistics

Poisson distribution in the presence of background, with mean λ. Then we have the same curves as for signal only, but replace ν with (ν+λ).



- Traditional approach: find limit on μ, then subtract λ to get limit on ν

- limit for ν improves for a fixed n when we add background.

- can get negative limits ! For example, n=0, λ>3 gives ν<0.

Feldman-Cousins Confidence Levels

Imagine we have a Poisson process with known background expectation and unknown signal. If $\lambda \geq 3$ and $n = 0$ then the confidence interval for ν is empty (or includes unphysical values).

This has led to new definitions for the Confidence Intervals. The most popular (at least in particle physics) is the Feldman-Cousins construction, where a rank is assigned to possible outcomes based on

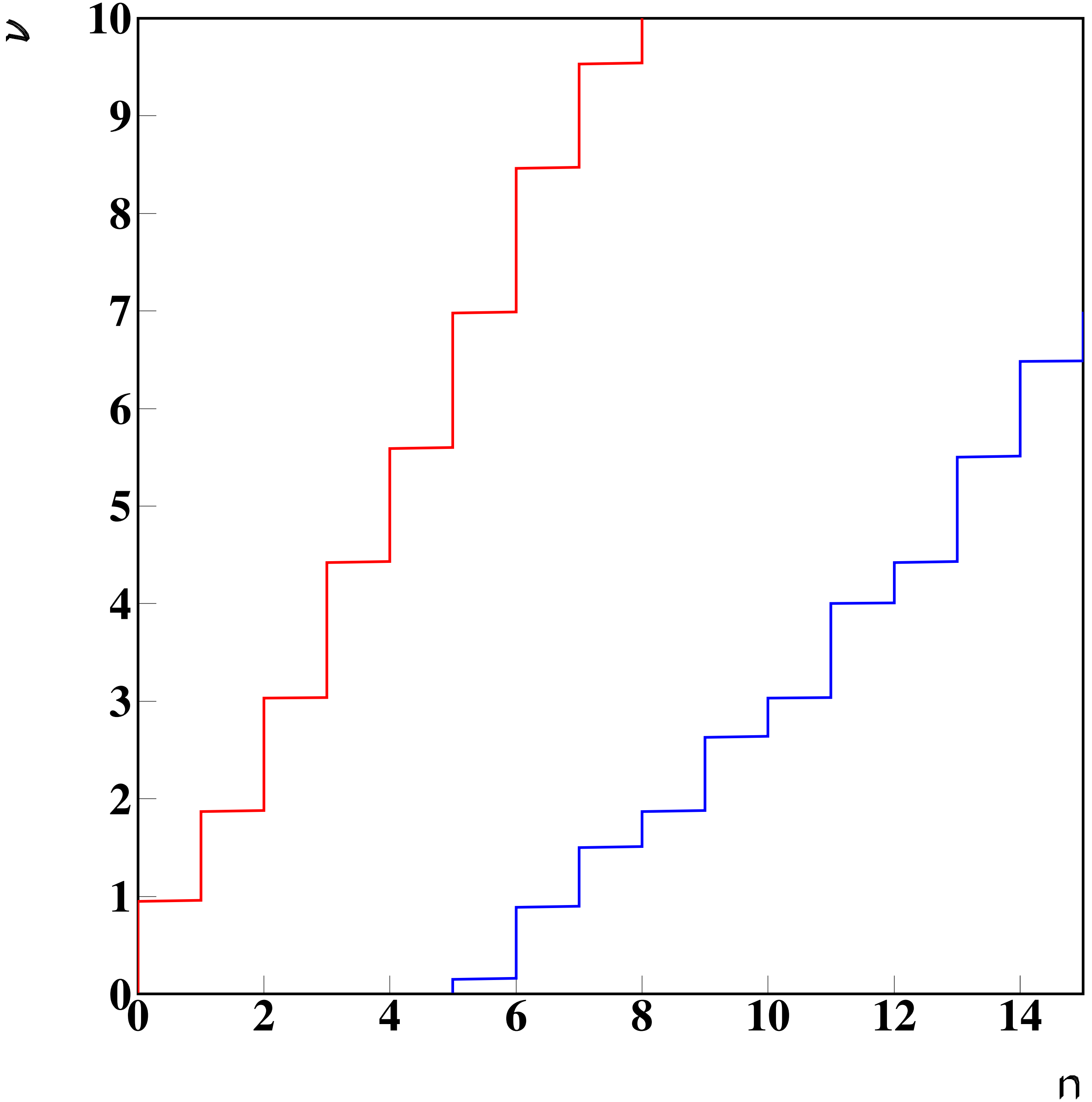
$$r = \frac{P(n|\mu = \lambda + \nu)}{P(n|\hat{\mu})}$$

Where $\hat{\mu}$ is the value of μ that maximizes $P(n|\mu)$ given the constraints.

Concrete example: $\lambda = 3.0 \quad \nu = 0.\bar{3}$

n	$P(n\|\nu)$	$\hat{\mu}$	$P(n\|\hat{\mu})$	r	Rank	$F_R(n\|\nu)$
0	0.0357	3.0	0.050	0.717	5	0.7565
1	0.1189	3.0	0.149	0.796	4	0.7208
2	0.1982	3.0	0.224	0.885	3	0.6091
3	0.2202	3.0	0.224	0.983	1	0.2202
4	0.1835	4.0	0.195	0.941	2	0.4037
5	0.1223	5.0	0.175	0.699	6	0.8788
6	0.0680	6.0	0.161	0.422	7	0.9468
7	0.0324	7.0	0.149	0.217	8	0.9792
8	0.0135	8.0	0.140	0.096	9	0.9927
9	0.0050	9.0	0.132	0.038	10	0.9976
10	0.0017	10.0	0.125	0.014	11	0.9993
11	0.0005	11.0	0.119	0.004	12	0.9998

Poisson 90% CL Bands a la Feldman-Cousins for λ=3.0



Comparing Feldman-Cousins with Bayesian Analysis with same background $\lambda = 3.0$ and a flat prior.

Recall: $P(\nu|n,\lambda) = \frac{e^{-\nu}(\lambda+\nu)^n}{n!\sum_{i=0}^{n}\frac{\lambda^i}{i!}}$

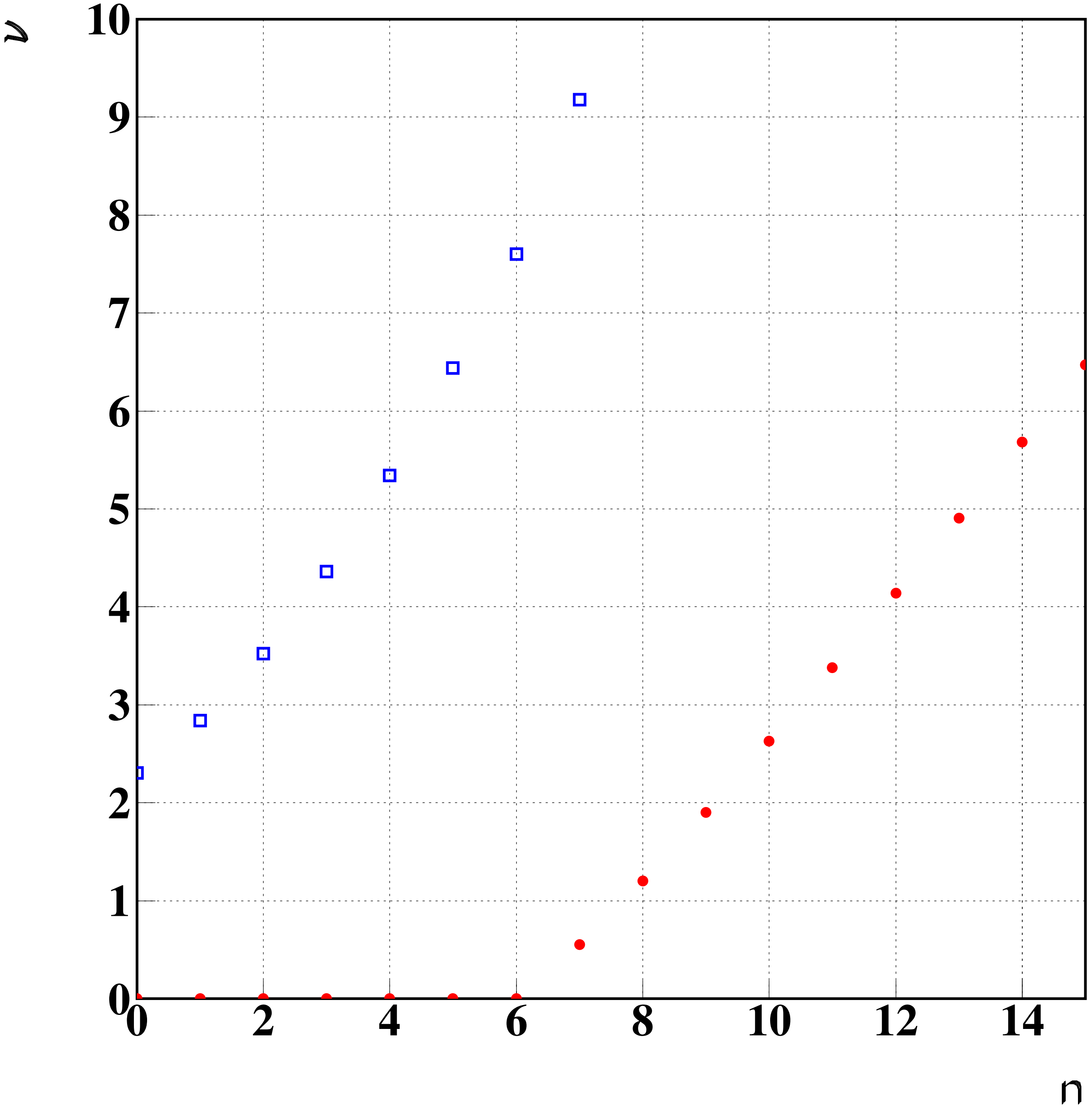
$$F(\nu|n,\lambda) = 1 - \frac{e^{-\nu}\sum_{i=0}^{n}\frac{(\lambda+\nu)^i}{i!}}{\sum_{i=0}^{n}\frac{\lambda^i}{i!}}$$

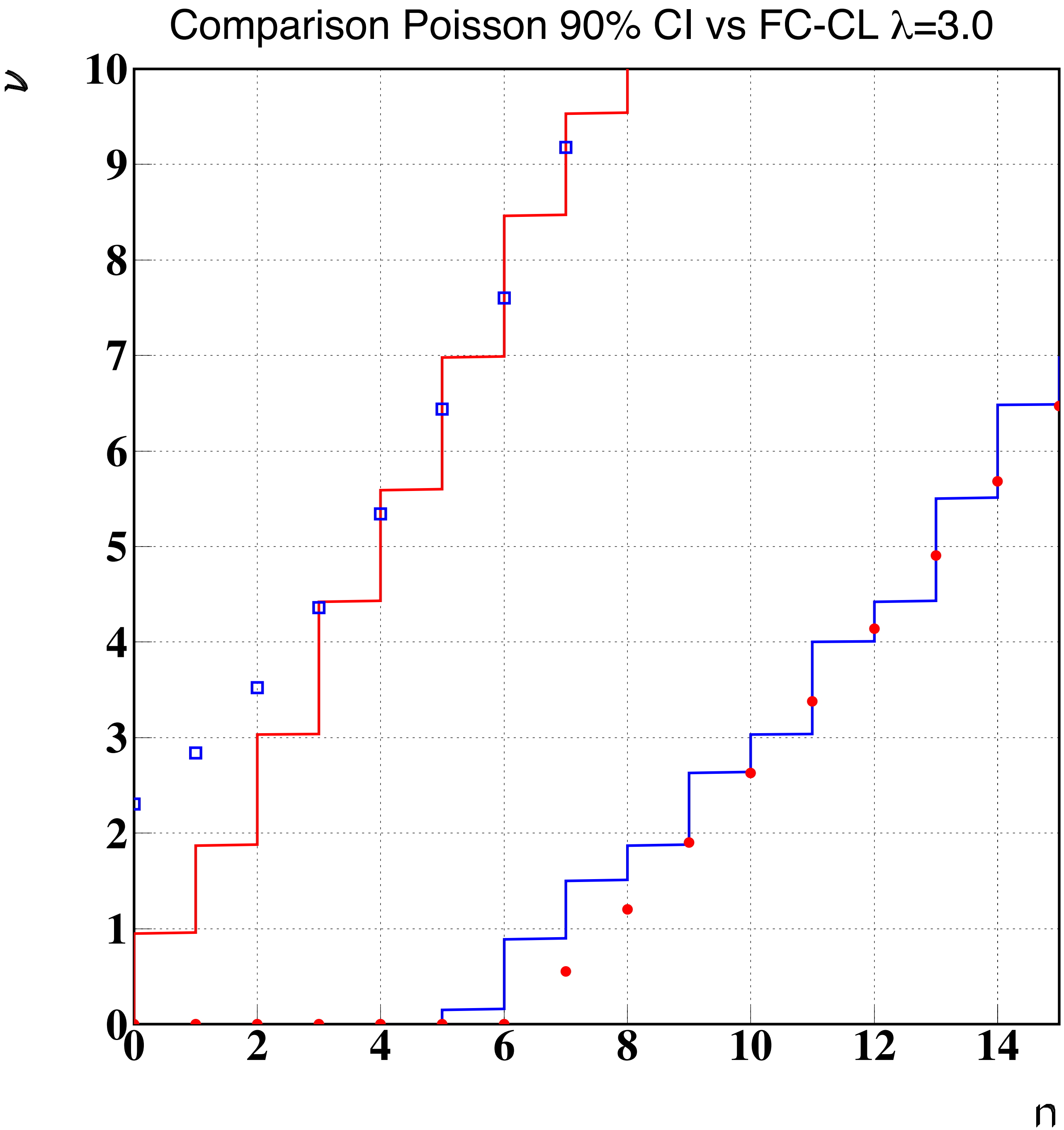
We will take the smallest interval with 90% credibility. I.e.,

$$\int_{P>C} P(\nu|n,\lambda)d\nu = 0.90$$

We find ν_{down} ν_{up} fulfilling this condition. Numerical integration.

Poisson 90% Credibility Intervals for λ=3.0




Comparison Poisson 90% CI vs FC-CL λ=3.0
ν
10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10
12
14
n

p-values and Goodness-of-fit

In general, we can think of quantities that summarize a 'distance' between the expectation and the observed. E.g., χ^2 is such a quantity. It is a test statistic (scalar function of the data, given the model).

$T(x|M,\lambda)$ Test statistic for possible data *x* given the model M and parameters λ

Create probability density for this quantity:

$$P(T(x|M,\lambda)) = P(x|M,\lambda)\frac{dx}{dT}$$

p-values and Goodness-of-fit

A p-value is a value of the cumulative pdf for the test statistic for some observed value of the data, D.

$$p = F(T(D)) = \int_{T_{\min}}^{T(D)} P(T)dT \qquad (= 1 - p)$$

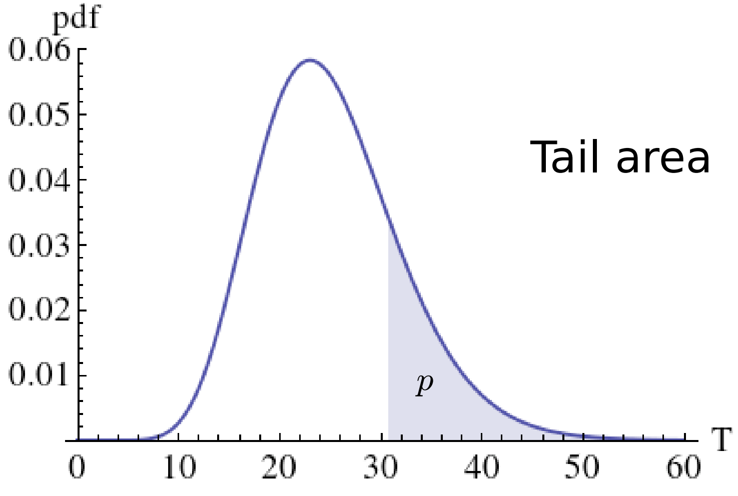
If the model is correct, we expect a flat distribution for p-values between (0,1).

$$P(F) = P(x)\frac{dx}{dF(T)} = \frac{P(x)}{d/dx\int P(T)dT} = \frac{P(x)}{d/dx\int P(x)dx} = 1$$

p-values and Goodness-of-fit

- Definition:

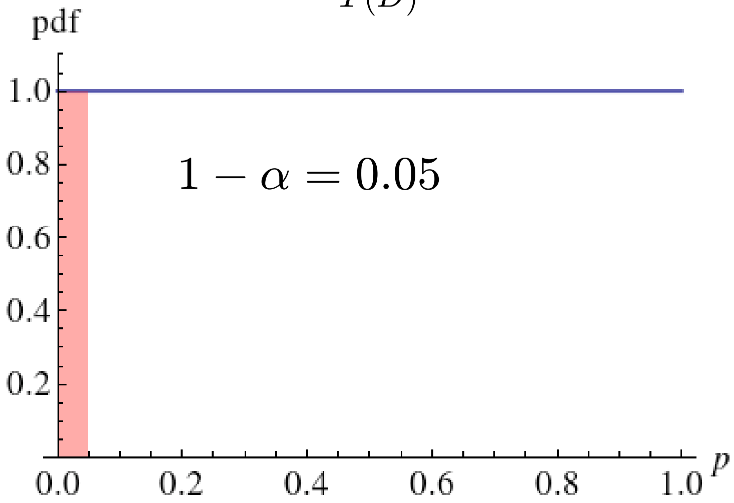
$$p \equiv P(T > T(D)|M)$$



- Assuming *M* and before data is taken: *p* uniform in [0,1]

- Confidence level α:

$$p < 1 - \alpha \Rightarrow \text{ reject model}$$



Why do we reject the small p-values if all are equally likely ?

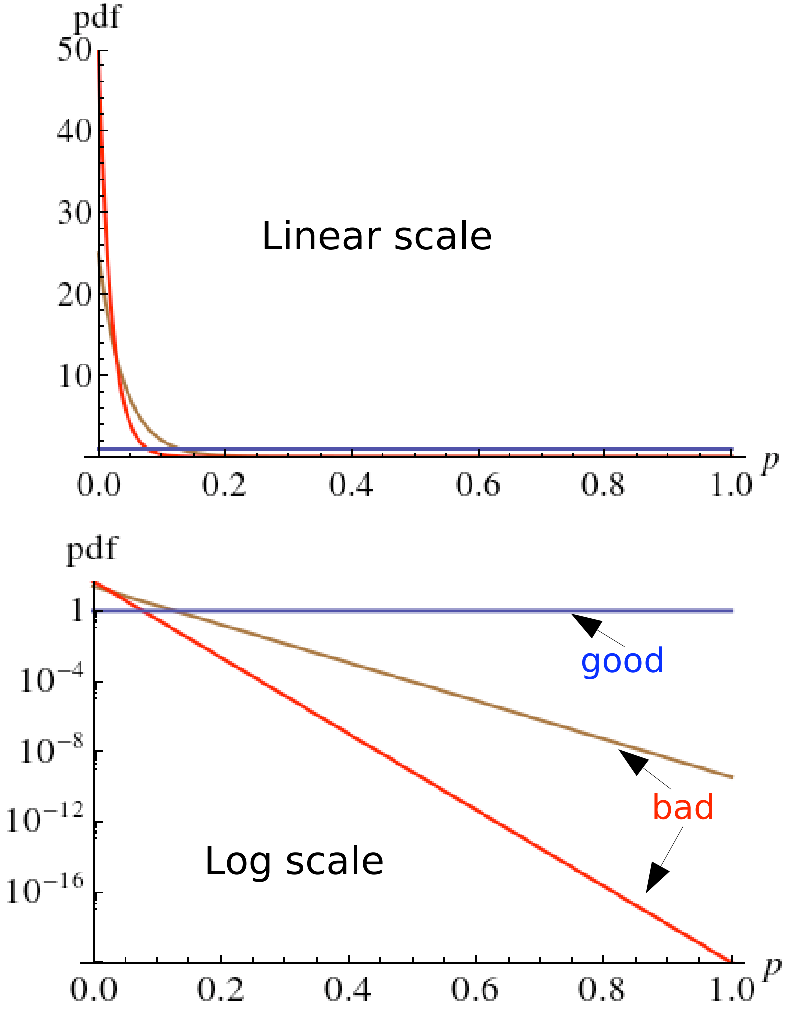
Comment on reasoning behind p-values

- Need prior knowledge about alternatives
- Good model: flat p-value

$$P(p|M_0) = 1$$

- Bad model: peak at *p*=0, sharply falling

$$P(p|M_i) \approx c_i e^{-c_i p}\ , \quad c_i \gg 1$$



Reasoning behind p-values

- Similar prior for all models $P(M_i) \approx P(M_j)$
- Bayes Theorem:

$$\begin{tikzcd}
 & P(M_0|p) \approx \dfrac{P(p|M_0)}{\sum_{i=0}^{K} P(p|M_i)} \arrow[dl, "\text{Small } p"'] \arrow[dr, "\text{Large } p"] & \\
P(M_0|p \approx 0) \approx \dfrac{1}{1+\sum_{i=1}^{K} c_i} \ll 1 & & P(M_0|p \approx 1) \approx 1
\end{tikzcd}$$

Bayes Theorem gives justification to p-values

Goodness of Fit

Use χ^2 as our test statistic. The probability distribution of χ^2 is known analytically. This is one of the main reasons why this test statistic is so popular. Strictly only applicable in limited cases (data follow Gaussian distribution from expectation, resolutions are not parameter dependent, if parameters fitted, then function needs to be linear in parameters, …).

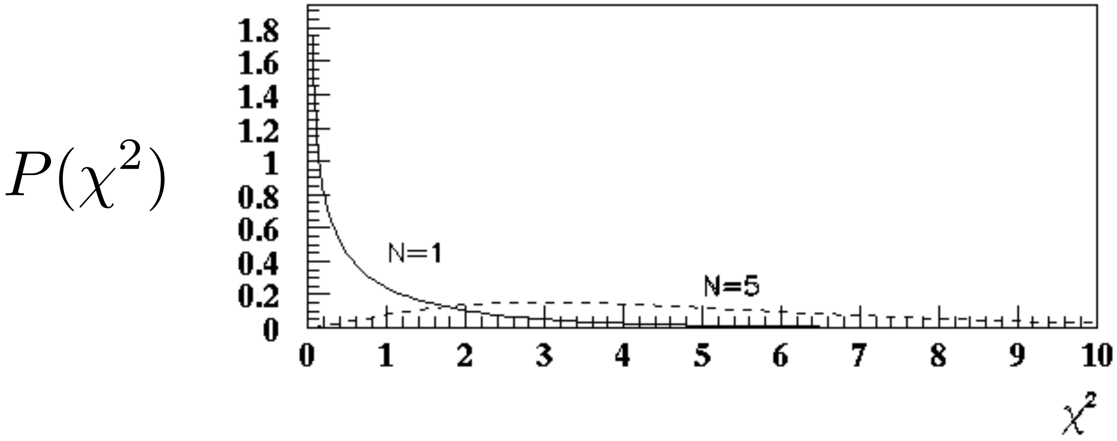
$$\boxed{P(\chi^2)d\chi^2 = \frac{1}{2^{N/2}\Gamma(N/2)}e^{-\chi^2/2}(\chi^2)^{(N/2)-1}d\chi^2}$$

$$\Gamma(n) = (n-1)! \qquad \text{n integer} > 0$$

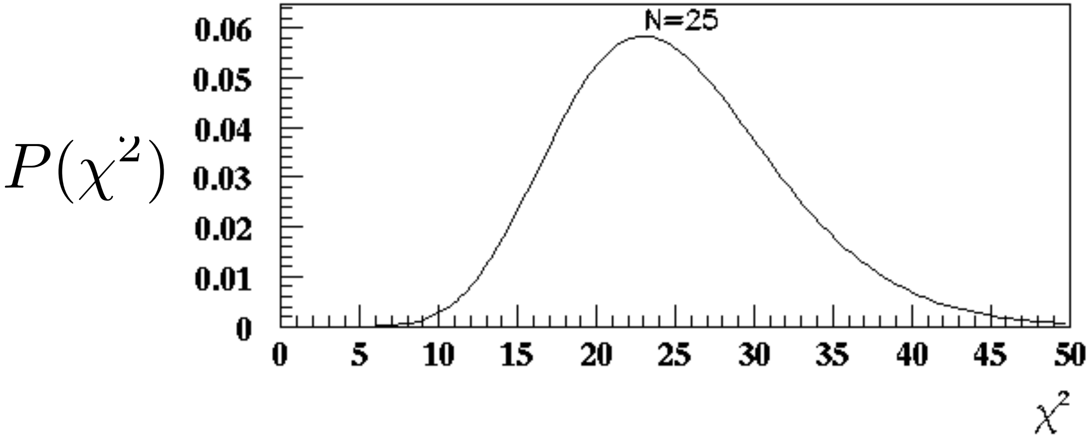
$$\Gamma(n+1/2) = \frac{(2n)!}{4^n n!}\sqrt{\pi} \qquad \text{n integer} \geq 0$$

Goodness of Fit

For a given (least-squares) fit to a set of data, a certain χ^2 value will be obtained. One can then look up in tables whether this value is reasonable by calculating, e.g.,



$$p = \int_{\chi_0^2}^{\infty} P(\chi^2) d\chi^2$$



Warning on p-values

p-values depend critically on how you have chosen the test statistic (or discrepancy variable). The same data set can have hugely varying p-values resulting from different choices of the test quantity.

E.g., consider a model where we assume an exponential decay law. We can define the following probabilities of the data:

Unbinned likelihood

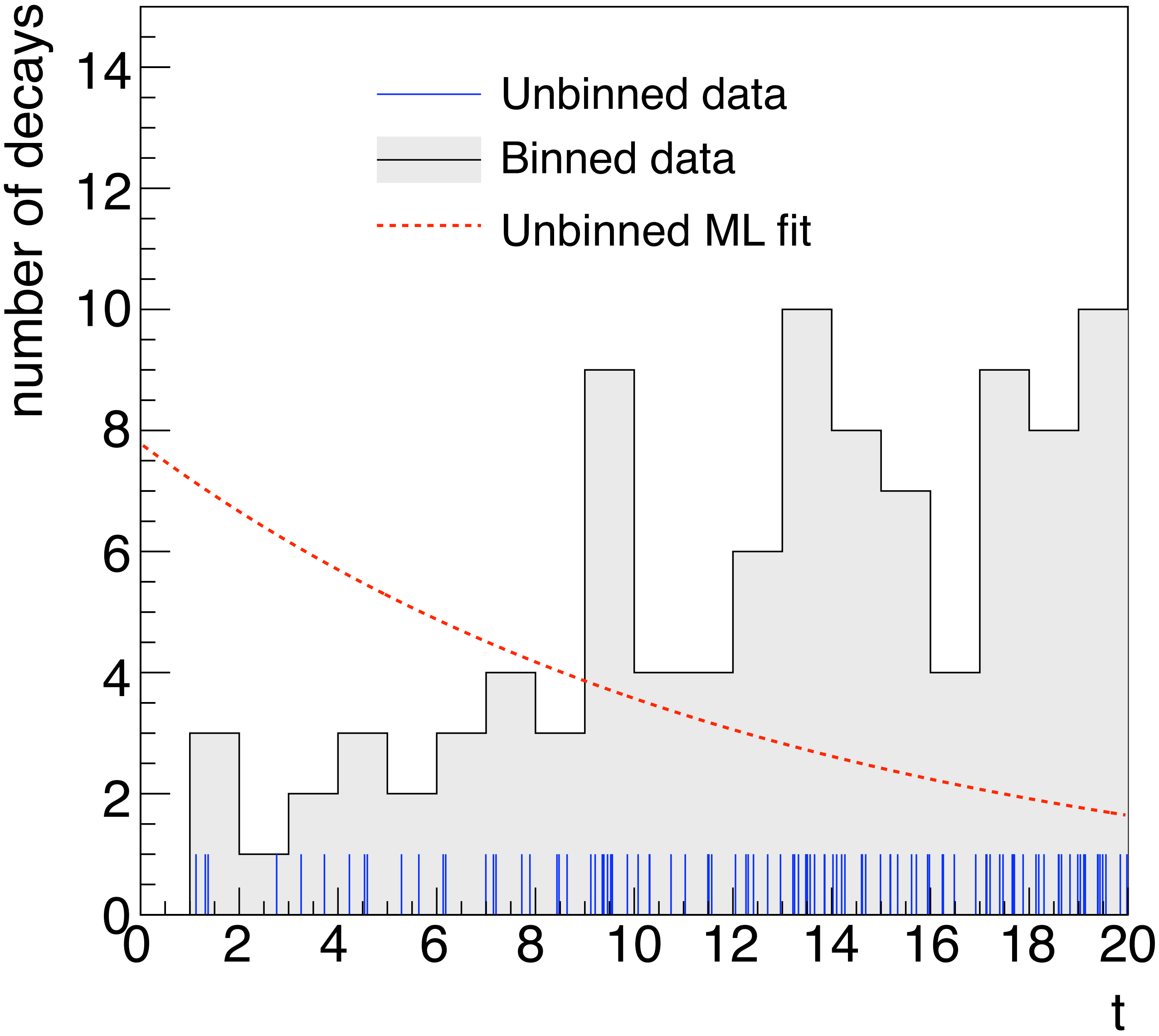
$$P(\vec{t}|\tau) = \prod_{i=1}^{N} \frac{1}{\tau} e^{-t_i/\tau}$$

Binned Poisson distribution

$$P(\vec{t}|\tau) = \prod_{j=1}^{M} \frac{e^{-\nu_j} \nu_j^{n_j}}{n_j!}$$

$$\nu_j = \text{expected events in bin j}$$
$$n_j = \text{observed events in bin j}$$

pitfalls



Assumed model is exponential. Data actually from linearly increasing function.

pitfalls

We take the best fit probability as our test statistic. For the unbinned fit

$$\tau^* = \frac{1}{N}\sum_{i=1}^{N} t_i$$

$$p = \int_{\sum t'_i > \xi} \mathrm{d}t'_1 \int \mathrm{d}t'_2 \ldots (\tau^*)^{-N} e^{-\sum t'_i/\tau^*} = 1 - P(N,N)$$

Regularized incomplete gamma function

$$P\left(s,x\right) = \frac{\gamma\left(s,x\right)}{\Gamma\left(s\right)} = \frac{\int_0^x t^{s-1}e^{-t}\mathrm{d}t}{\int_0^\infty t^{s-1}e^{-t}\mathrm{d}t}$$

Doesn't depend on the data ! In fact, for large *N,* $p \approx 0.5$

pitfalls

The p-value from the maximum likelihood is about 0.5 !

The p-value from the binned fit is 0

What happened ? The maximum likelihood quantity does not know anything about the distribution of the events, and the result only depends on

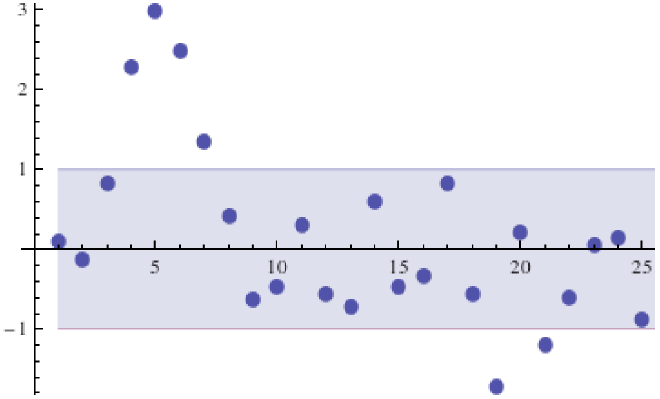
$$\tau^* = \frac{1}{N} \sum_{i=1}^{N} t_i$$

and the p-value only depends on N !

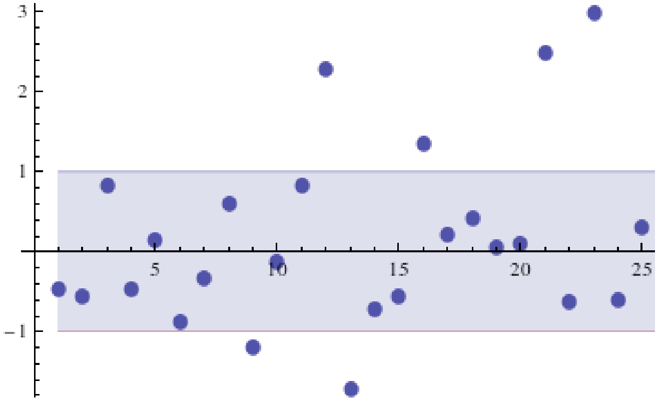
Lesson: make sure your test statistic is sensitive to what you want to test ! The fitting program may give you a high p-value and it could well be that the fit function looks nothing like the data.

χ^2

- Most statistics disrespect order of data, information wasted
- Human brain good for simple problems



$$\chi^2 = 32.1 \Rightarrow p = 0.16$$



Example:

- Series of *N*=25 datapoints
- Each Gaussian with mean = 0 and variance = 1

⇒ Can we combine information about **order** and **magnitude of deviation**?

F. Beaujean, A. Caldwell, Jour. Stat. Plan. & Infer. 141 (2011) 3437.

Bayesians and Frequentists

Frequentists make statements of the kind:

‘Assuming the model is correct, this result will occur in XX% of the experiments’

The model is **assumed true**, and estimators for the true parameters in the model are produced from the data.

In the ‘classical’ approach, this is then converted to ‘assuming the model, the bounds [a,b] will contain the true value in XX% of experiments performed’ (confidence levels). Does not imply that the true value is in the range [a,b] with probability XX !

The decision on whether to then believe the model/parameters is left to the individual (subjective). *The inductive part of the reasoning is left out of the analysis.*

Bayesians and Frequentists

Bayesians make statements of the kind:

'the degree-of-belief in model A is XX (between 0,1)'

Given the new data, the degree-of-belief is updated using the frequencies of possible outcomes in the context of the models (full set)

Credible regions are then defined: with XX% credibility, the parameter is in the interval [a,b]. **Note – very different from a CL.**

The inductive part of the reasoning is built in to the analysis, and the connection between prior beliefs and posterior beliefs is made clear.

Subjective, but the subjective element is made explicit.

Bayesians and Frequentists

In both approaches, work with models and frequencies of outcomes within the model.

Many elements are the same: modeling; picking the most sensitive variables to test the theory, …

There is no right and wrong approach, but you have to understand what you get out of each type of analysis. E.g., don't confuse confidence levels with probabilities, p-values with support for a model, …

BAT → Software package for solving data analysis problems

Code structured on Bayes' formula for parameter estimation

$$P(\vec{\lambda}, M|\vec{D}) = \frac{P(\vec{D}|\vec{\lambda}, M)P(\vec{\lambda}, M)}{P(\vec{D})}$$

- **The idea behind BAT**
- Merge common parts of every Bayesian analysis into a software package
- Provide flexible environment to phrase arbitrary problems
- Provide a set of well tested/tuned numerical algorithms and tools
- C++ based framework (flexible, modular)
- Interfaces to ROOT, Cuba, Minuit, user defined, ..
- can be downloaded from: http://www.mppmu.mpg.de/bat

Parameter Estimation

The posterior pdf gives the full probability distribution for all parameters, including all correlations – no approximations. If interested in subset of parameters, then marginalize. E.g., for one parameter:

$$P(\lambda_i|\vec{D},M) = \int P(\vec{\lambda}|\vec{D},M)d\vec{\lambda}_{\mathrm{J}\neq i}$$

Can calculate what you need from the posterior pdf. E.g.,

$\text{Mode} \quad \overset{\lambda_i}{\max}\left\{P(\lambda_i|D,M)\right\}$ + probability intervals, …

$\text{Mean of } \lambda_i \quad <\lambda_i>=\int P(\lambda_i|\vec{D},M)\lambda_i d\lambda_i$

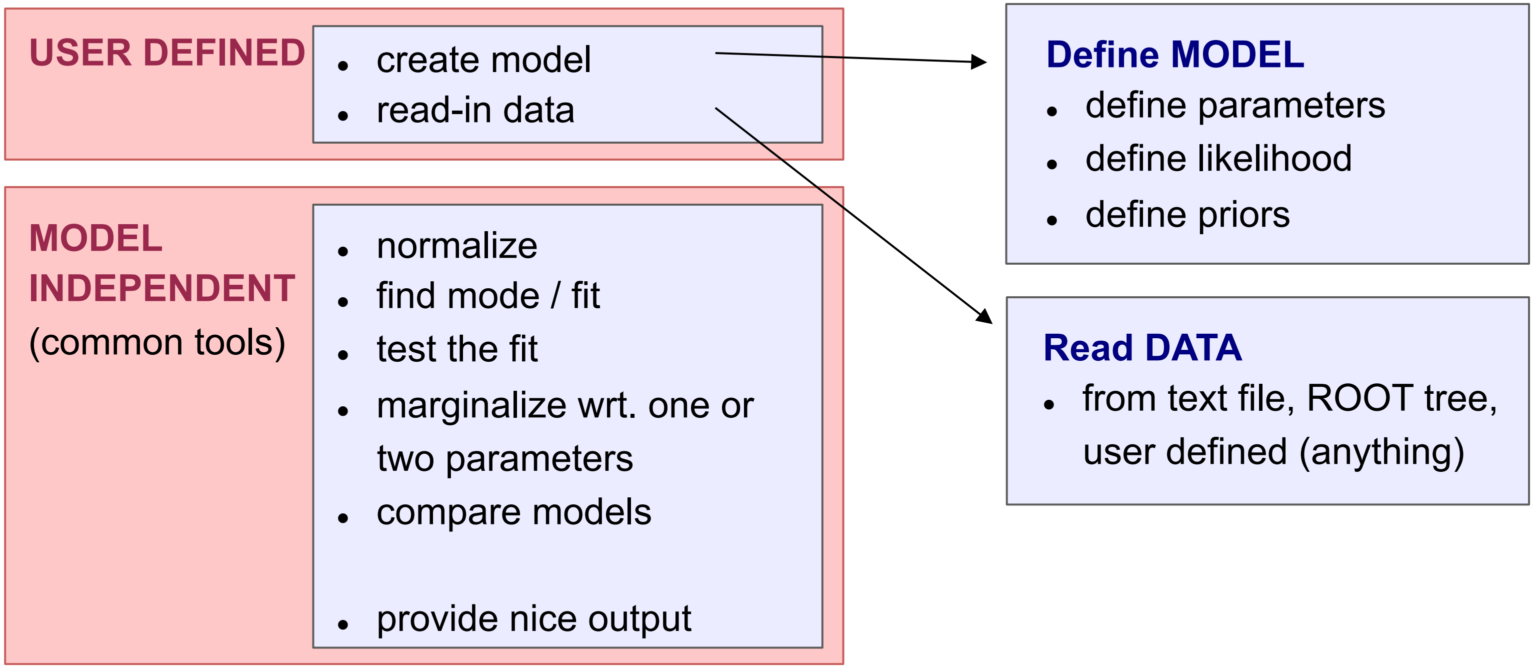
$\text{Median} \quad \int_{\lambda_{min}}^{\lambda_{med}} P(\lambda_i|\vec{D},M)d\lambda_i = 0.5$

Can also perform uncertainty propagation w/o approximations

The idea

Separate the common parts from the rest

- case specific: the model and the data
- common tools: all the rest



USER DEFINED

- create model
- read-in data

MODEL INDEPENDENT (common tools)

- normalize
- find mode / fit
- test the fit
- marginalize wrt. one or two parameters
- compare models
- provide nice output

Define MODEL

- define parameters
- define likelihood
- define priors

Read DATA

- from text file, ROOT tree, user defined (anything)

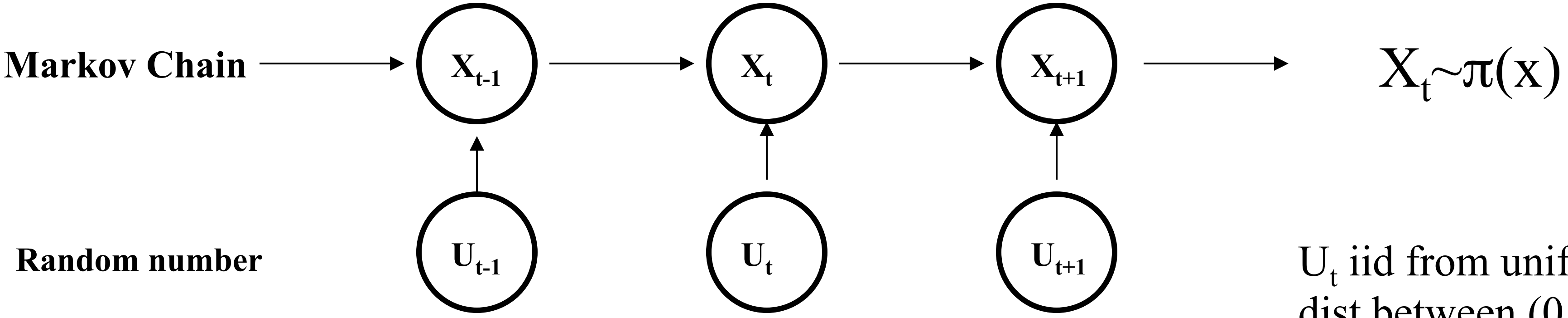
Markov Chain Monte Carlo (MCMC)

- generally it is very difficult to obtain the full posterior PDF
  - number of parameters can be large
  - different input data will result in a different posterior
- also the visualization of the PDF in more than 3 dimensions is rather impractical and hard to understand
- usually one looks at marginalized posterior wrt. one, two or three parameters
  - a projection of the posterior onto one (two, three) parameter
  - integrating all the other parameters out
  - still numerically difficult

- the Markov Chain Monte Carlo revolutionized the area of Bayesian analysis

Markov Chain Monte Carlo

Goal of MCMC is to find a chain with $(\pi_i)_{i=0}^{\infty}$=pdf of interest. Sampling according to the Markov Chain will then correspond to sampling from the desired pdf.



U_t iid from uniform dist between (0,1)

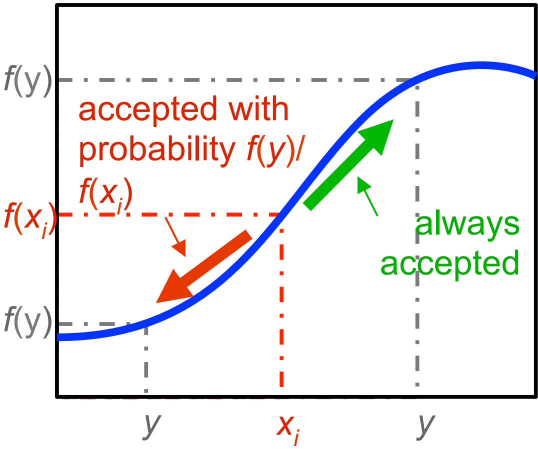
Markov Chain Monte Carlo is any method producing an ergodic Markov chain X_t whose stationary distribution in the distribution of interest.

The original algorithm is due to Metropolis. Later generalized by Hastings.

Metropolis algorithm

- In BAT implemented Metropolis algorithm
- Map positive function ***f*(*x*)** by random walk towards higher probabilities
- Algorithm:

  - Start at some randomly chosen x_i
  - Randomly generate y around x_i
  - If $f(y) \geq f(x_i)$, set $x_{i+1} = y$
  - If $f(y) < f(x_i)$, set $x_{i+1} = y$ with probability $f(y)/f(x_i)$
  - If y not accepted, stay where you are, i.e., set $x_{i+1} = x_i$
  - Generate new y , repeat



- For each step fill the histogram with x_{i+1}
- For infinite number of steps the distribution in the histogram converges to ***f(x)***

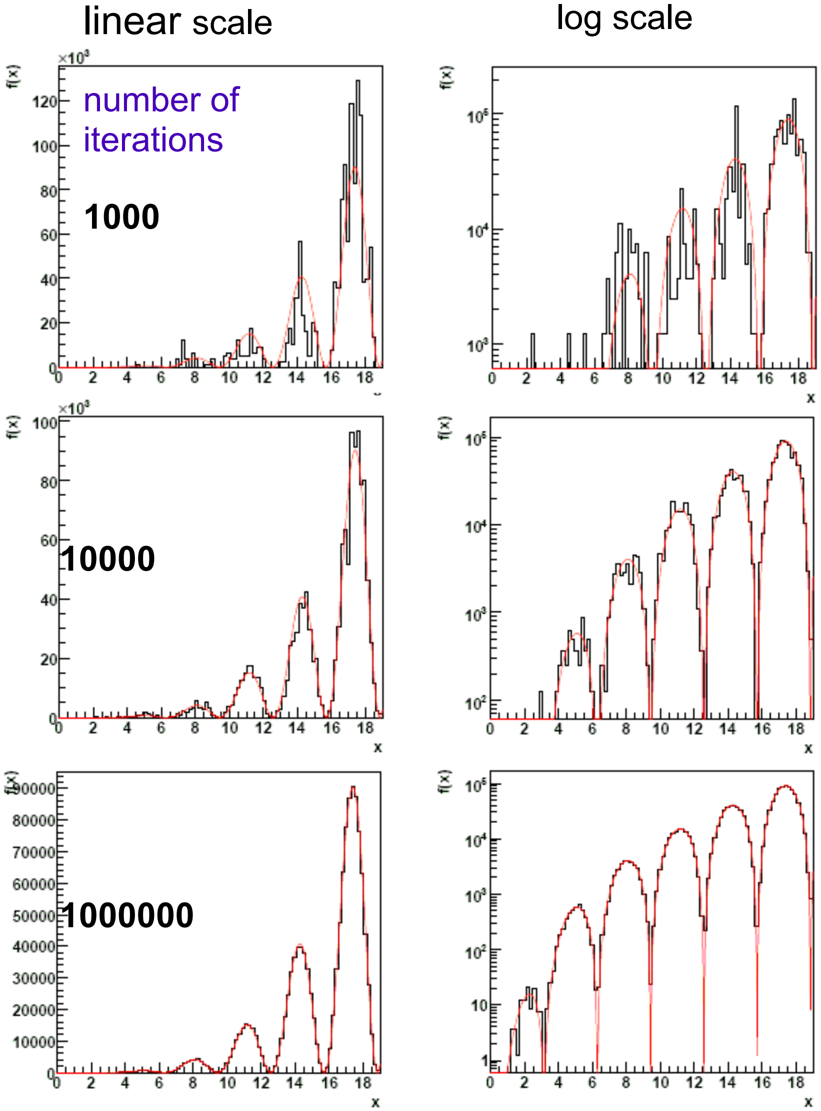
Exercise: try out the Metropolis algorithm to generate a Gaussian distribution from flat rn [0,1]

MCMC: an example

- mapping an arbitrary function:

$$\text{e.g.} \quad f(x) = x^4 \sin^2 x$$

- distribution sampled by MCMC in this case quickly converges towards the underlying distribution
- **mapping of complicated shapes with multiple minima and maxima**

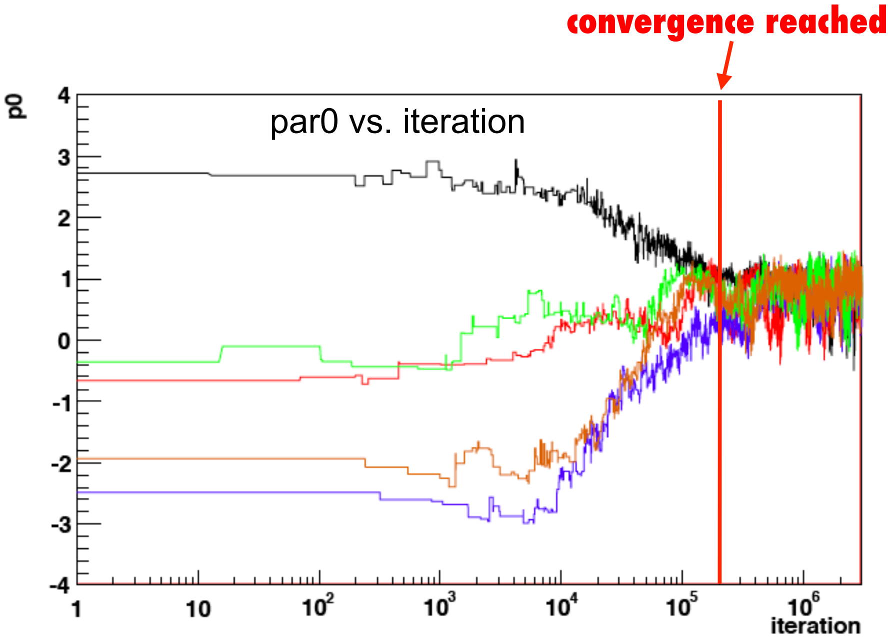


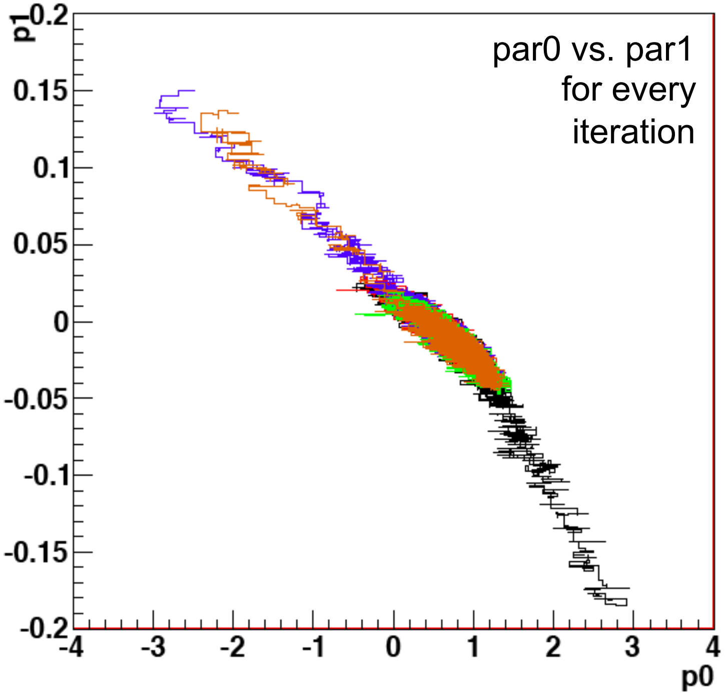
Note:

- MCMC has to become stationary to sample from underlying distribution
- in general the convergence is a non-trivial problem

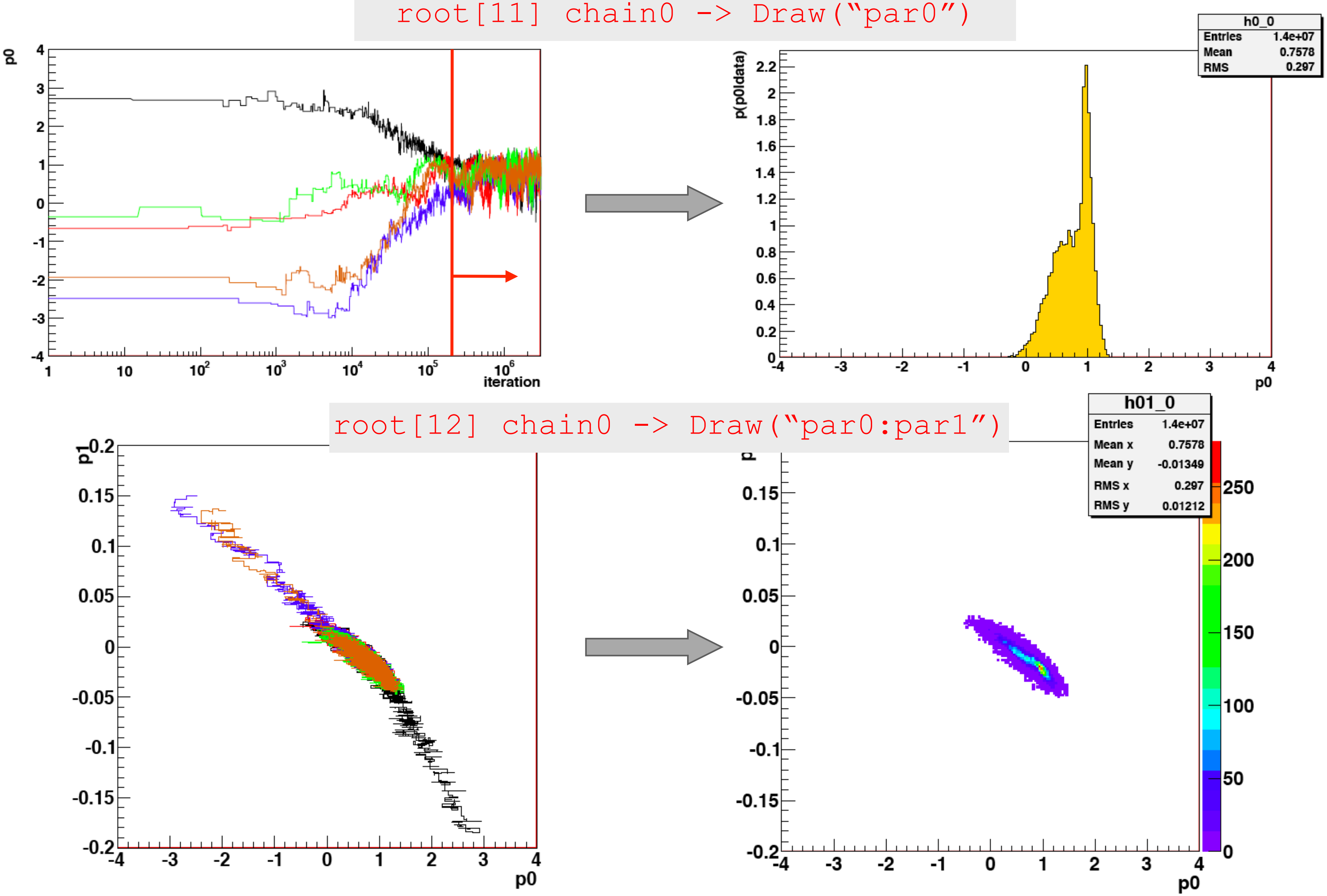
Analysis of Markov Chain

- the full chain(s) can be stored for further analysis and parameter tuning as ROOT TTree(s)
  - allows direct usage of standard ROOT tools for analysis
- Markov Chain contains the complete information about the posterior (except for the normalization)





Obtaining marginalized distributions from TTree



```
root[11] chain0 -> Draw("par0")
```

```
root[12] chain0 -> Draw("par0:par1")
```

Using the Markov Chain

Once you have the chain, it is simple to calculate quantities of interest.

Chain is $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}_i \quad i = 1, N$

E.g., pdf for one parameter: just plot $\{\lambda_j\}_i$ joint $\{\lambda_j, \lambda_k\}_i$

Expectation value of a function $E[f(\vec{\lambda})] = \frac{1}{N} \sum_{i=1}^{N} f(\lambda_{1i}, \ldots, \lambda_{ni})$

Probability distribution of your function: just plot $\{f(\lambda_1, \ldots, \lambda_n)\}_i$