Statistics in X-ray Data Analysis

Aneta Siemiginowska
Harvard-Smithsonian Center for Astrophysics

OUTLINE

- Introduction and general consideration
    - Motivation: why do we need statistics?
    - Probabilities/Distributions
    - Frequentist vs. Bayes
- Statistics in X-ray Analysis - use Sherpa CIAO
    - Poisson Likelihood
    - Parameter Estimation
    - Statistical Issues
    - Hypothesis Testing
- References and Summary

Why do we need Statistics?

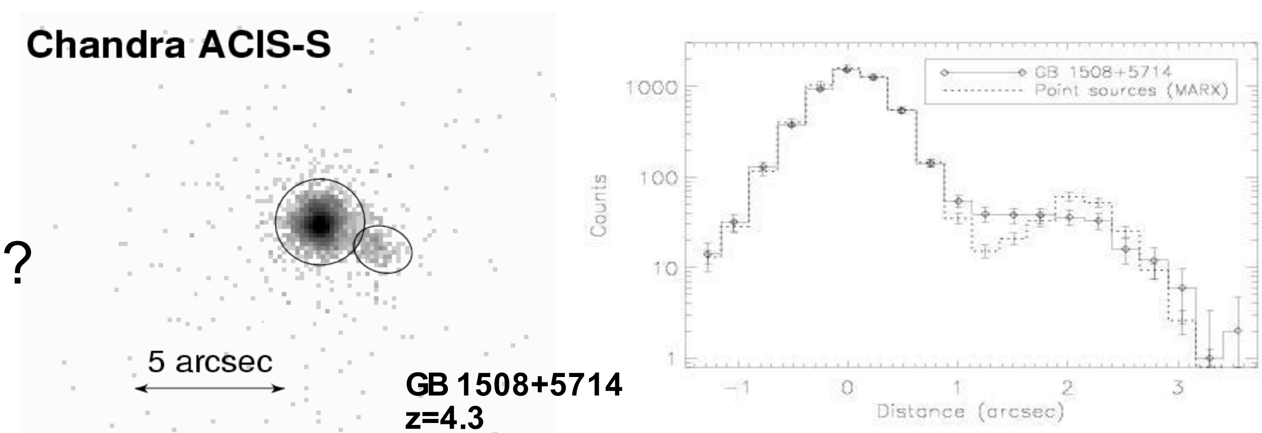
- How do we take decisions in Science?

  Tools: instruments, data collections, reduction, classifications – tools and techniques

  Decisions: is this hypothesis correct? Why not? Are these data consistent with other data? Do we get an answer to our question? Do we need more data?

- Comparison to decide:
  - Describe properties of an object or sample:

Example:
Is a faint extension a jet or a point source?



Siemiginowska et al (2003)

Stages in Astronomy Experiments

Stage	How	Example	Considerations
OBSERVE	Carefully	Experiment design, exposure time (S)	What? Number of objects, Type? (S)
REDUCE	Algorithms	calibration files QE,RMF,ARF,PSF (S)	data quality Signal-to-Noise (S)
ANALYSE	Parameter Estimation, Hypothesis testing (S)	Intensity, positions (S)	Frequentist Bayesian? (S)
CONCLUDE	Hypothesis testing (S)	Distribution tests, Correlations (S)	Belivable, Repeatable, Understandable? (S)
REFLECT	Carefully	Mission achieved? A better way? We need more data! (S)	The next Observations (S)

Wall & Jenkins (2003)

Statistic is a quantity that summarizes data

Statistics are combinations of data that do not depend on unknown parameters:
Mean, averages from multiple experiments etc.

=> Astronomers cannot avoid Statistics

Definitions

- ***Random variable:*** a variable which can take on different numerical values, corresponding to different experimental outcomes.
  - Example: binned data D_i , which can have different values even when an experiment is repeated exactly.
- ***Statistic:*** a function of random variables.
  - Example: data D_i , or a population mean

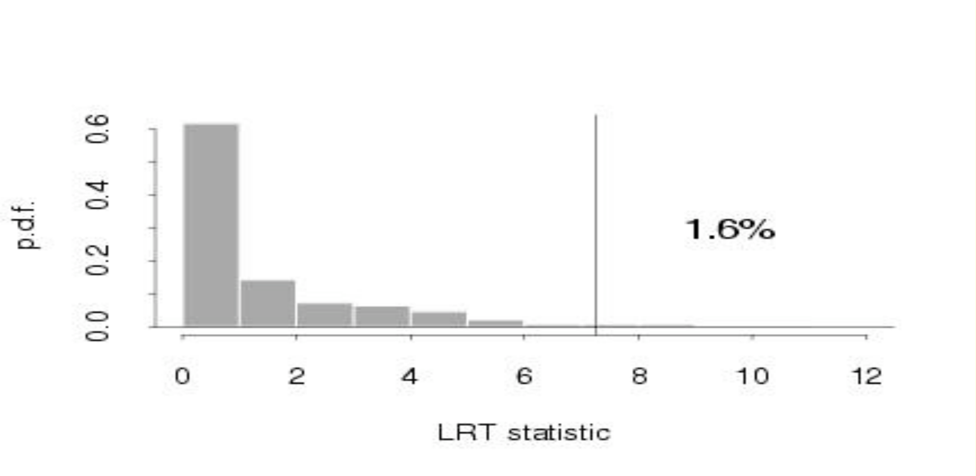
$$(\mu = \left[\sum_{i=1}^{N} D_i\right] / N)$$

- ***Probability sampling distribution:*** the normalized distribution from which a statistic is sampled. Such a distribution is commonly denoted $p(X\ |Y\ )$, "the probability of outcome X given condition(s) Y," or sometimes just $p\ (X\ )$. Note that in the special case of the Gaussian (or normal) distribution, $p\ (X\ )$ may be written as $N(\mu,\sigma^2)$, where μ is the Gaussian mean, and σ^2 is its variance.

Probability Distributions

Probability is crucial in decision process:

Example:



Limited data yields only partial idea about the line width in the spectrum. We can only assign the probability to the range of the line width roughly matching this parameter. We decide on the presence of the line by calculating the probability.

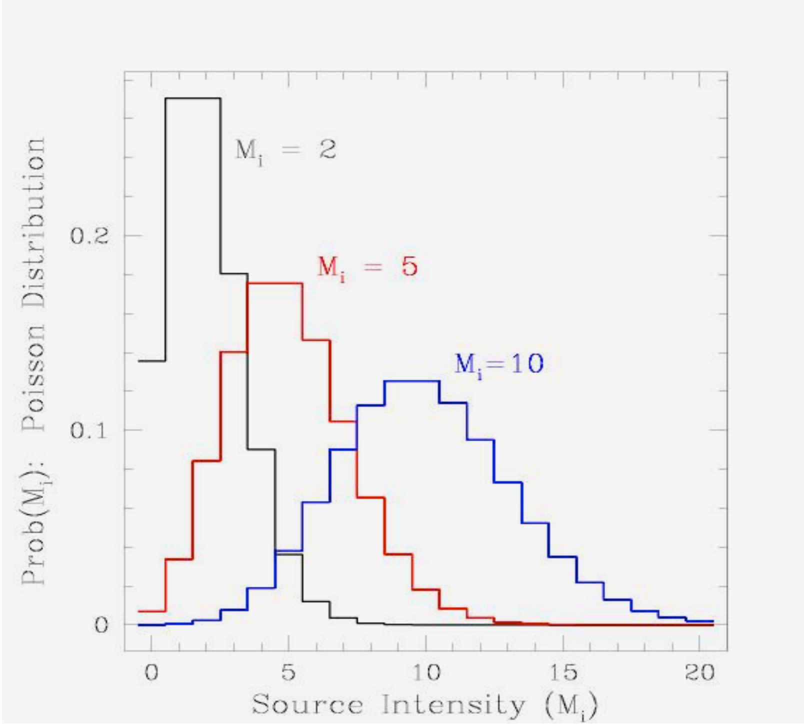
The Poisson Distribution

Collecting X-ray data => Counting individual photons
=> Sampling from Poisson distribution

The discrete Poisson distribution:

prob(D_i)= $$p(D_i \mid M_i) = \frac{M_i^{D_i}}{D_i!} e^{-M_i}$$

probability of finding D_i events (*counts*) in bin *i* (*energy rage*) of dataset *D* (*spectrum*) in a given length of time (exposure time), if the events occur independently at a constant rate M_i *(source intensity).*



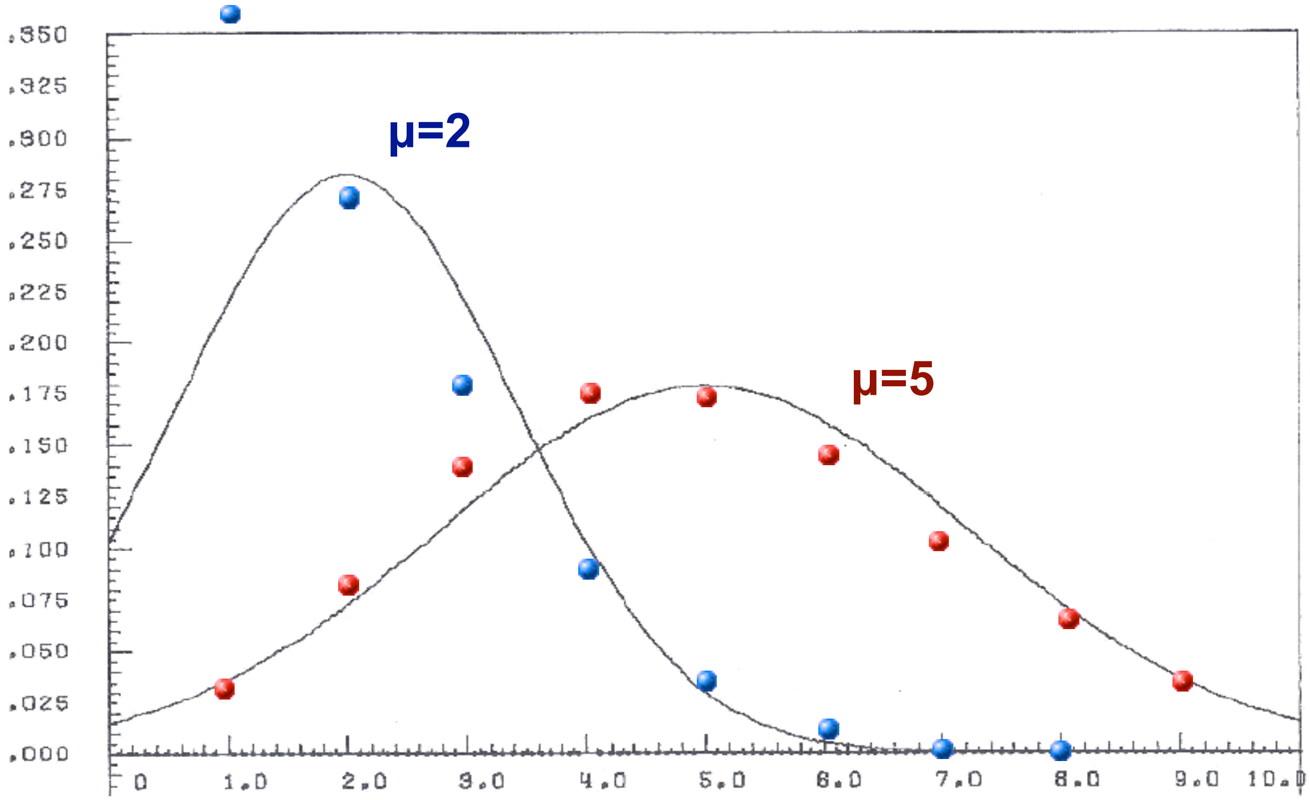
Things to remember:

- *Mean* $\mu = E[D_i] = M_i$
- Variance: $V[D_i] = M_i$
- $\mathrm{cov}[D_{i_1}, D_{i_2}] = 0$ => independent
- the sum of *n* Poisson-distributed variables is itself Poisson-distributed with variance:

$$\sum_{i=1}^{n} M_i$$

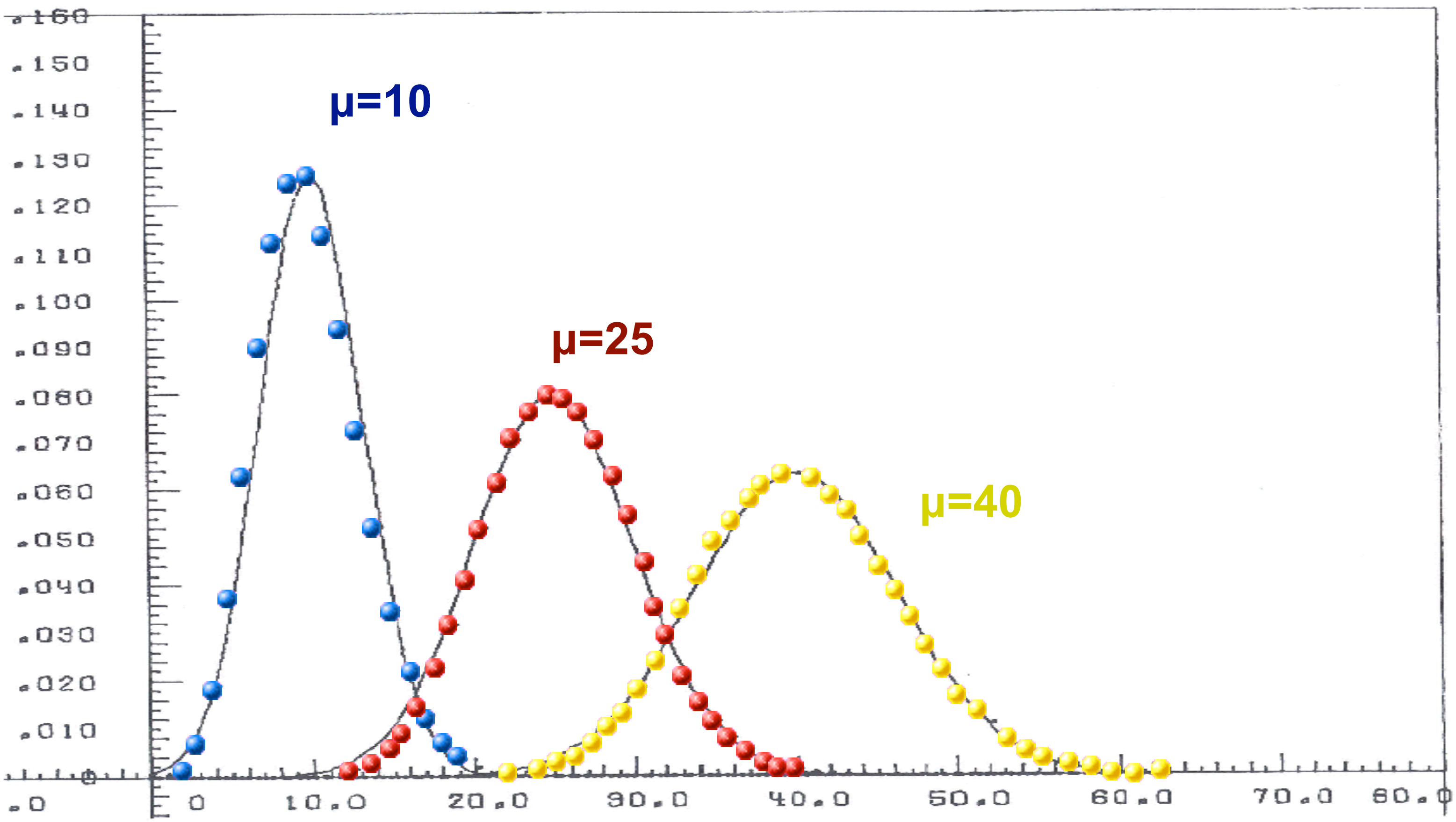
As $M_i \Rightarrow \propto$ Poisson distribution converges to Gaussian distribution $N(\mu = M_i \,;\, \sigma^2 = M_i)$

Poisson vs. Gaussian Distributions – Low Number of Counts



Comparison of Poisson distributions (dotted) of mean μ = 2 and 5 with normal distributions of the same mean and variance (Eadie *et al.* 1971, p. 50).

Poisson vs. Gaussian Distributions



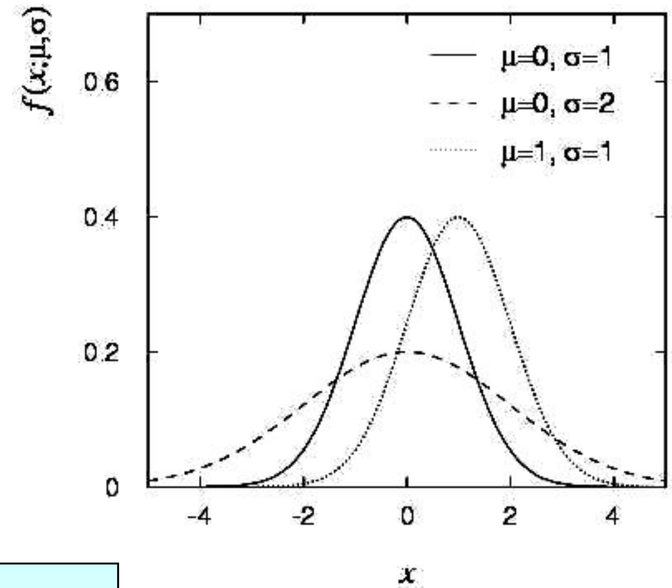
Comparison of Poisson distributions (dotted) of mean $\mu = 10, 25$ and 40 with normal distributions of the same mean and variance (Eadie *et al.* 1971, p. 50).

Gaussian Distribution

For large $\mu \to \infty$ Poisson (and the Binomial, large T) distributions converge to Gaussian (normal) distributions.

$$\mathrm{prob}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp[-(x-\mu)^2/2\sigma^2]$$

Mean - μ
Variance - σ^2



Note: Importance of the Tails!

±2σ range covers *95.45%* of the area, so *2σ* result has less than *5%* chance of occurring by chance, but because of the error estimates this is not the acceptable result. Usually *3σ* or *10σ* have to be quoted and the convergence to Gaussian is faster in the center than in the tails!

Central Limit Theorem

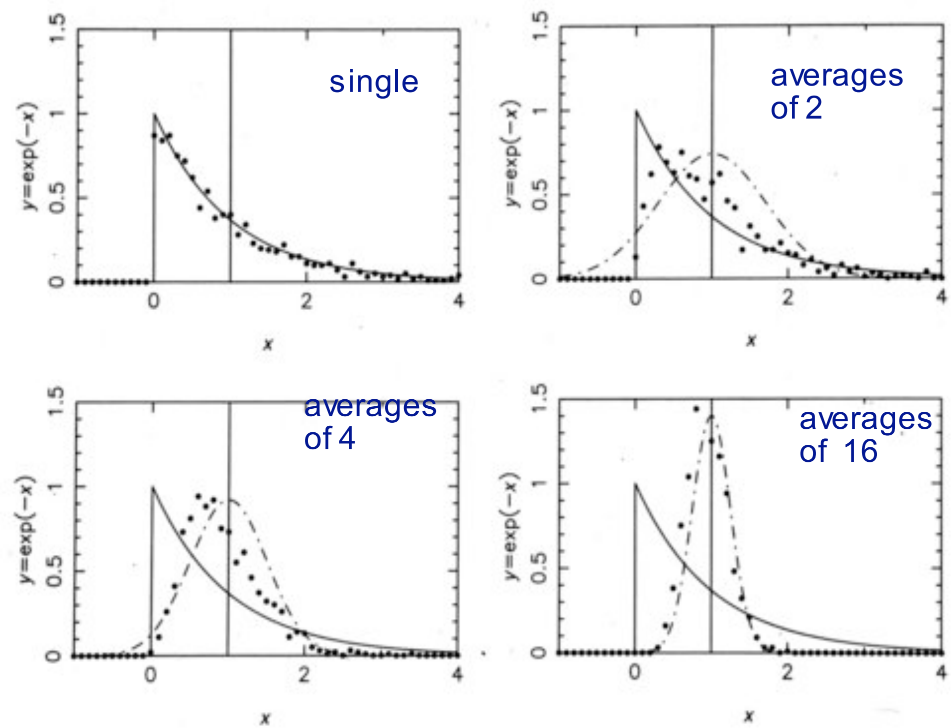
The true importance of the Gaussian distribution: Regardless of the original distribution **- an averaging will produce a Gaussian distribution.**

Form averages M_n from repeated drawing of n samples from a population with finite mean μ and variance σ^2

$$\frac{(M_n - \mu)}{\sigma/\sqrt{n}}$$ => *Gaussian Distribution*

as $n \rightarrow \infty$

$\mu=0,\ \sigma^2=1$



200 y values drawn from exp(-x) function

Bayesian vs. Classical

Example:

D = 8.5∓0.1 Mpc

Does not describe probability that a true value is between 8.4 and 8.6.

We **assume** that a Gaussian distribution applies and knowing the distribution of errors we can make probabilistic statements.

Classical Approach:

Assuming the true distance D_0 then D is normally distributed around D_0 with a standard deviation of 0.1. Repeating measurement will yield many estimates of distance D which all scatter around true D_0.

Assume the thing (distance) we want to know and tell us how the data will behave.

Bayesian Approach:

Deduce directly the probability distribution of D_0 from the data. *Assumes the data and tell us the thing we want to know. No repetition of experiment.*

X-ray Analysis

Main Steps in Analysis

- Data:
  - Write proposal, win and obtain new data
- Models:
  - model library that can describe the physical process in the source
  - typical functional forms or tables, derived more complex models - plasma emission models etc.
  - parameterized approach - models have parameters
- Optimization Methods:
  - to apply model to the data and adjust model parameters
  - obtain the model description of your data
  - constrain model parameters etc. search of the parameter space
- Statistics:
  - a measure of the model deviations from the data

What do we do in X-rays?

Example:

I've observed my source, reduce the data and finally got my X-ray spectrum – what do I do now? How can I find out what does the spectrum tell me about the physics of my source?

Run **XSPEC** or ***Sherpa*!** But what do those programs really do?

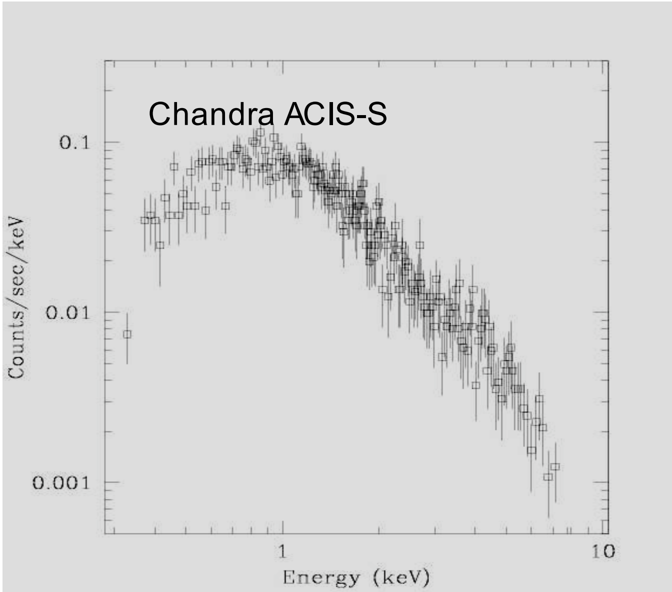
Fit the data => $\mathbf{C(h)=\int R(E,h)\ A(E)\ M(E,\theta)dE}$

Counts
Response
Effective Area
Model

h- detector channels
E- Energy
θ- model parameters

Assume a model and look for the best model parameters which describes the observed spectrum.

Need a Parameter Estimator - Statistics



Modeling: Models

- Parameterized models: $f(E,\Theta_i)$ or $f(x_i,\Theta_i)$

  absorption - N_H

  photon index of a power law function - Γ

  blackbody temperature kT

- Composite models:

  combined individual models in the library into a model that describes the observation

```
set_model("xsphabs.abs1*powlaw1d.p1")
set_model("const2d.c0+gauss2d.g2")
```

- Source models, Background models:

```
set_source(2,"bbody.bb+powlaw1d.pl+gauss1d.line1+gauss1d.line2")
set_bkg_model(2,"const1d.bkg2")
```

Modeling: Model Library

- Model Library in Sherpa includes standard XSPEC models and other 1D and 2D functions

```
sherpa-11> list_models()
['atten',
 'bbody',
 'bbodyfreq',
 'beta1d',
 'beta2d',
 'box1d',…
```

Example Function myline:

```
def myline(pars, x):
   return pars[0] * x + pars[1]
```

In sherpa:

```
from myline import *

load_data(1, "foo.dat")
load_user_model(myline, "myl")
add_user_pars("myl", ["m","b"])
set_model(myl)
myl.m=30
myl.b=20
```

- User Models:
  – Python or Slang Functions
  load_user_model, add_user_pars

  – Python and Slang interface to
  C/C++ or Fortran code/functions

Modeling: Parameters

```
sherpa-21> set_model(xsphabs.abs1*xszphabs.zabs1*powlaw1d.p1)
sherpa-22> abs1.nH = 0.041
sherpa-23> freeze(abs1.nH)
sherpa-24> zabs1.redshift=0.312

sherpa-25> show_model()
Model: 1
apply_rmf(apply_arf((106080.244442 * ((xsphabs.abs1 * xszphabs.zabs1)*powlaw1d.p1))))
   Param        Type          Value          Min          Max      Units
   -----        ----          -----          ---          ---      -----
   abs1.nh         frozen       0.041        0       100000 10^22 atoms / cm^2
   zabs1.nh        thawed           1        0       100000 10^22 atoms / cm^2
   zabs1.redshift frozen        0.312        0           10
   p1.gamma        thawed           1      -10           10
   p1.ref          frozen           1     -3.40282e+38  3.40282e+38
   p1.ampl         thawed           1          0  3.40282e+38
```

Parameter Estimators: Statistics

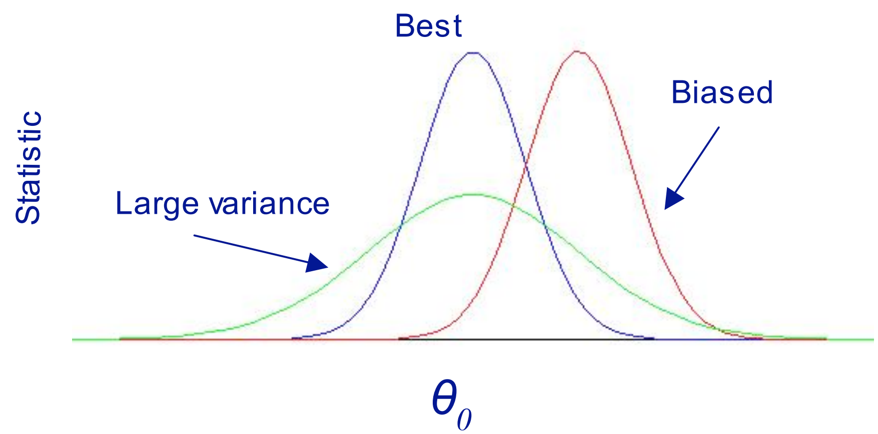
Requirements on Statistics:

- **Unbiased**
  - converge to true value with repeated measurements

- **Robust**
  – less affected by outliers

- **Consistent**
  – true value for a large sample size (Example: rms and Gaussian distribution)

- **Closeness**
  - smallest variations from the truth



Maximum Likelihood: Assessing the Quality of Fit

One can use the Poisson distribution to assess the probability of sampling data $\boldsymbol{D_i}$ given a predicted (convolved) model amplitude M_i. Thus to assess the quality of a fit, it is natural to maximize the product of Poisson probabilities in each data bin, *i.e.*, to maximize the Poisson likelihood:

$$\mathsf{L} = \prod_i^N \mathsf{L}_i = \prod_i^N \frac{M_i^{D_i}}{D_i!} \exp(-M_i) = \prod_i^N p(D_i \mid M_i)$$

In practice, what is often maximized is the log-likelihood,

$\boldsymbol{L} = \log\mathcal{L}$. A well-known statistic in X-ray astronomy which is related to L is the so-called "Cash statistic":

$$C \equiv 2\sum_i^N [M_i - D_i \log M_i] \propto -2L,$$

Likelihood Function

Likelihood

Observed Counts

Probability Distribution

Model parameters

$$
\begin{aligned}
\mathcal{L}(\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_N) &= \mathcal{P}(\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_N \,|\, \Theta) \\
&= \mathcal{P}(\mathcal{X}_1 \,|\, \Theta)\, \mathcal{P}(\mathcal{X}_2 \,|\, \Theta) \ldots \mathcal{P}(\mathcal{X}_N \,|\, \Theta) \\
&= \prod \mathcal{P}(\mathcal{X}_i \,|\, \Theta)
\end{aligned}
$$

$\mathcal{P}$ - Poisson Probability Distribution for X-ray data

$\mathcal{X}_1, \ldots \mathcal{X}_N$ - X-ray data - independent

Θ - model parameters

Likelihood Function: X-rays Example

- X-ray spectra modeled by a power law function:

$$f(E) = A * E^{-\Gamma}$$

E - energy; A, Γ - model parameters: a normalization and a slope

Predicted number of counts:

$$M_i = \int R(E,i) * A(E)\, AE^{-\Gamma}\, dE$$

For A = 0.001 ph/cm2/sec, Γ=2 than in channels i= (10, 100, 200)

Predicted counts: M_i = (10.7, 508.9, 75.5)

Observed X_i = (15, 520, 74)

Calculate individual probabilities:

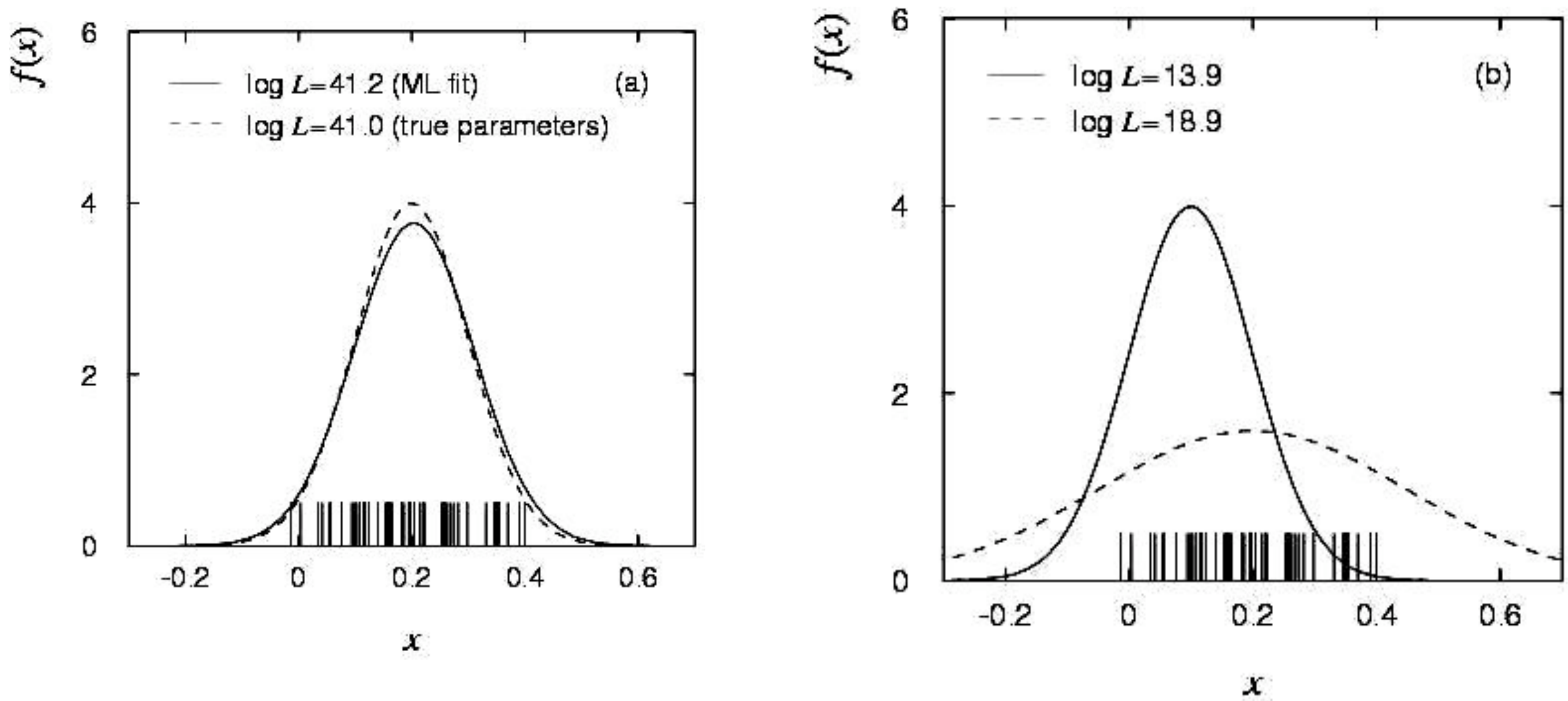
Use Incomplete Gamma Function

$\Gamma(X_i, M_i)$

$$\mathcal{L}(X_i) = \prod^{N} \mathcal{P}(X_i \mid M_i(A, \Gamma))$$
$$= \mathcal{P}(15 \mid 10.7)\mathcal{P}(520 \mid 508.9)\mathcal{P}(74 \mid 75.5)$$
$$= 0.116$$

- Finding the maximum likelihood means finding the set of model parameters that maximize the likelihood function

If the hypothesized θ is close to the true value, then we expect a high probability to get data like that which we actually found.



(Non-) Use of the Poisson Likelihood

In model fits, the Poisson likelihood is not as commonly used as it should be. Some reasons why include:

- a historical aversion to computing factorials;
- the fact the likelihood cannot be used to fit "background subtracted" spectra;
- the fact that negative amplitudes are not allowed (not a bad thing physics abhors negative fluxes!);
- the fact that there is no "goodness of fit" criterion, i.e. there is no easy way to interpret $\mathcal{L}_{max}$ (however, *cf*. the **CSTAT** statistic); and
- the fact that there is an alternative in the Gaussian limit: the χ^2 statistic.

χ^2–Statistic

Definition: $\chi^2 = \sum_i (D_i - M_i)^2 / \sigma_i^2$

The χ^2 statistics is **minimized** in the fitting the data, varying the model parameters until the best-fit model parameters are found for the minimum value of the χ^2 statistic

Degrees-of-freedom = **k-1- N**

N – number of parameters
K – number of spectral bins

“Versions” of the χ^2 Statistic

Generally, the χ^2 statistic is written as:

$$\chi^2 \equiv \sum_i^N \frac{(D_i - M_i)^2}{\sigma_i^2} ,$$

where σ_i^2 represents the (unknown!) variance of the Poisson distribution from which D_i is sampled.

χ^2 **Statistic**	σ_i^2
Data Variance	D_i
Model Variance	M_i
Gehrels	$[1+(D_i+0.75)^{1/2}]^2$
Primini	M_i from previous best-fit
Churazov	based on *smoothed* data D
“Parent”	$\frac{\sum_{i=1}^N D_i}{N}$
Least Squares	1

Note that some X-ray data analysis routines may estimate σ_i for you during data reduction. In PHA files, such estimates are recorded in the **STAT_ERR** column.

Statistics in Sherpa

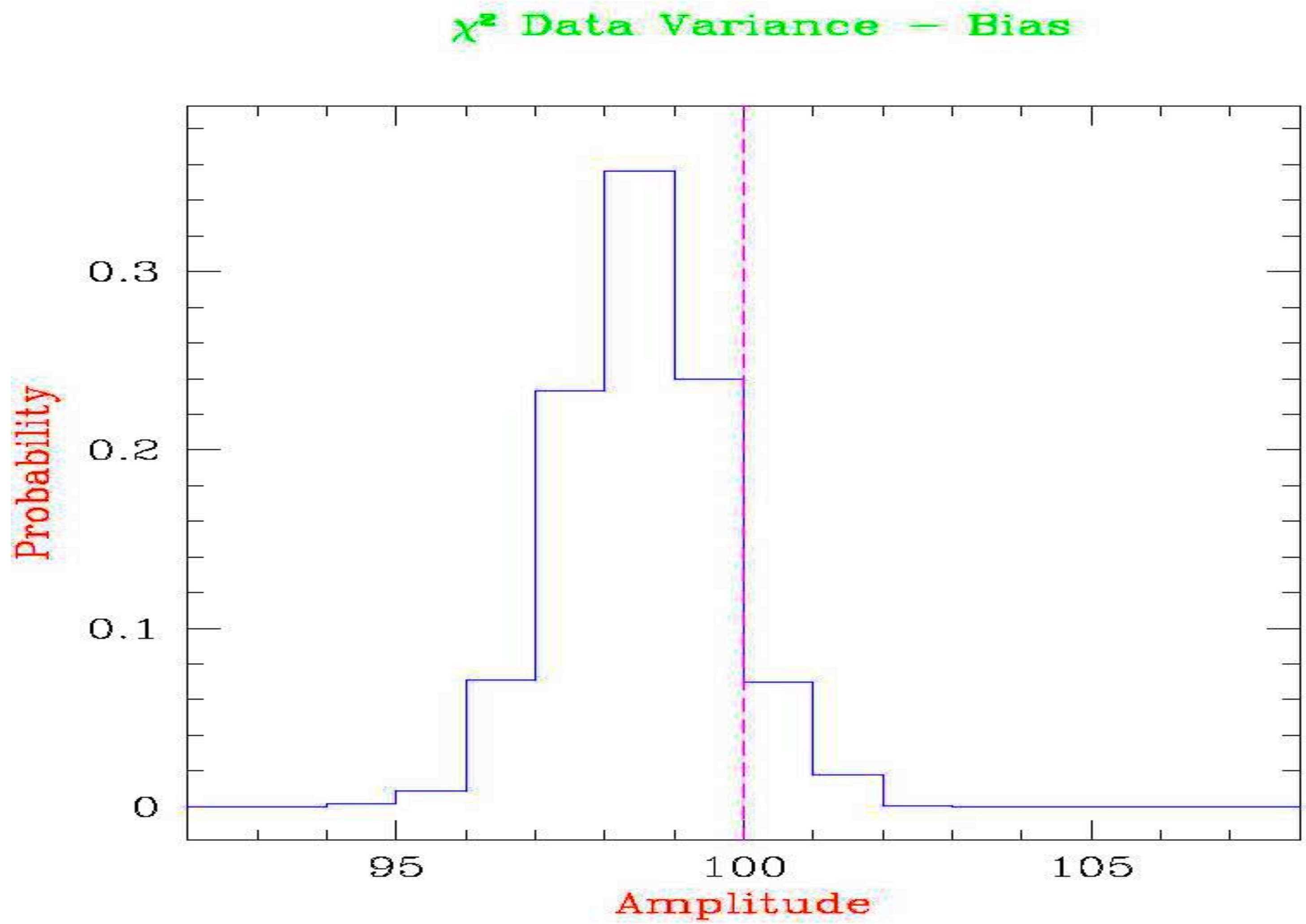
- χ^2 statistics with different weights
- Cash and Cstat based on Poisson likelihood

```
sherpa-12> list_stats()
['leastsq',
 'chi2constvar',
 'chi2modvar',
 'cash',
 'chi2gehrels',
 'chi2datavar',
 'chi2xspecvar',
 'cstat']
sherpa-13> set_stat("chi2datavar")
sherpa-14> set_stat("cstat")
```

Statistical Issues: Bias

- Sampling distributions:
  - For model $M(\theta_0)$ simulate (sample) a large number of data sets
  - fit the model (in respect to θ)
  - Collect distributions for each parameter
- Statistics (*e.g.*, χ^2) is biased if the mean of these distributions (E[θ_k]) differs from the true values θ_o.
- The Poisson likelihood is an unbiased estimator.
- The χ^2 statistic ***can*** be biased, depending upon the choice of σ:
  - Using the *Sherpa* **FAKEIT**, we simulated 500 datasets from a constant model with amplitude 100 counts.
  - We then fit each dataset with a constant model, recording the inferred amplitude.

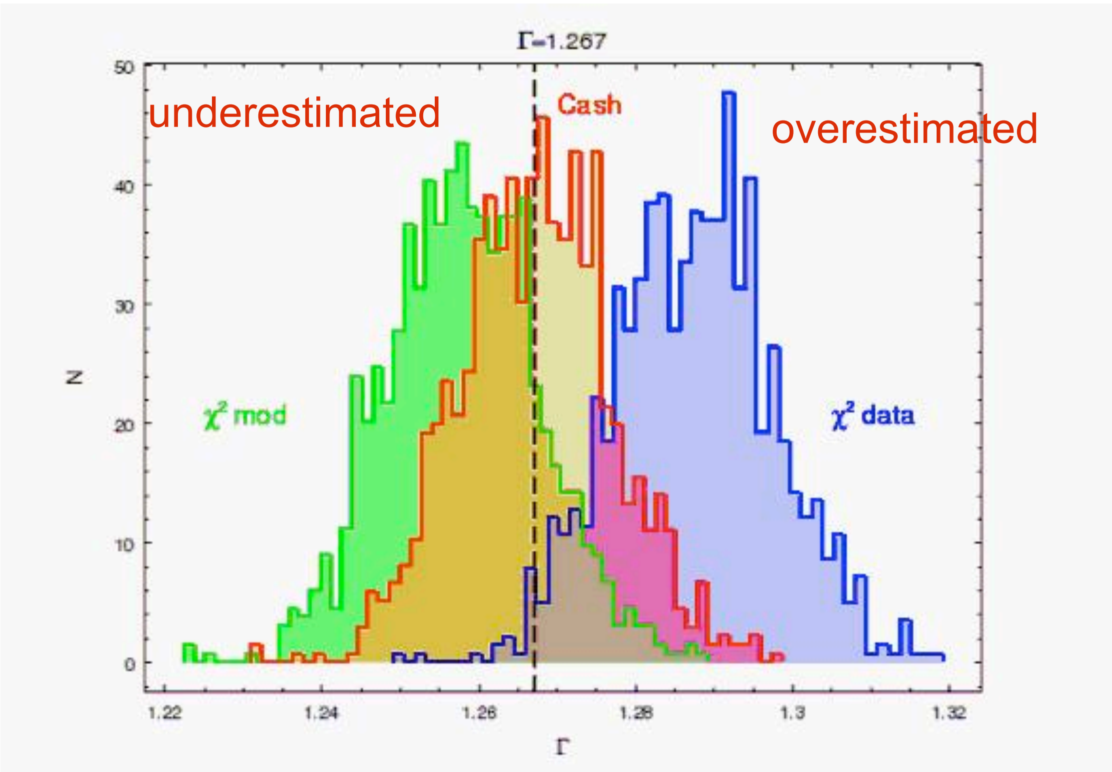
Statistic	Mean Amplitude
Gehrels	99.05
Data Variance	99.02
Model Variance	100.47
“Parent”	99.94
Primini	99.94
Cash	99.98



A demonstration of bias. Five hundred datasets are sampled from a constant model with amplitude 100 and then are fit with the same constant amplitude model, using χ^2 with data variance. The mean of the distribution of fit amplitude values is not 100, as it would be if the statistic were an unbiased estimator.

Statistics: Example of Bias

Simulation: ~60,000 counts



1/ Assume model parameters

2/ Simulate a spectrum - use fakeit - fold through response and add a Poisson noise.

3/ Fit each simulated spectrum assuming different weighting of χ^2 and Cash statistics

4/ Plot results

Line shows the assumed parameter value in the simulated model.

In High S/N data!

Fitting: Search the Parameter Space

```
sherpa-28> fit()
Dataset              = 1
Method               = levmar
Statistic            = chi2datavar
Initial fit statistic = 644.136
Final fit statistic   = 632.106 at function evaluation 13
Data points          = 460
Degrees of freedom     = 457
Probability [Q-value] = 9.71144e-08
Reduced statistic     = 1.38316
Change in statistic   = 12.0305
   zabs1.nh       0.0960949
   p1.gamma       1.29086
   p1.ampl        0.000707365
```

```
sherpa-29> print get_fit_results()
datasets   = (1,)
methodname = levmar
statname   = chi2datavar
succeeded  = True
parnames   = ('zabs1.nh', 'p1.gamma', 'p1.ampl')
parvals    = (0.0960948525609, 1.29085977295, 0.000707365006941)
covarerr   = None
statval    = 632.10587995
istatval   = 644.136341045
dstatval   = 12.0304610958
numpoints  = 460
dof        = 457
qval       = 9.71144259004e-08
rstat      = 1.38316385109
message    = both actual and predicted relative reductions in the sum of squares are at most
ftol=1.19209e-07
nfev       = 13
```

Fitting: Optimization Methods

- Optimization - finding a minimum (or maximum) of a function:

  "A general function f(x) may have many isolated local minima, non-isolated minimum hypersurfaces, or even more complicated topologies. No finite minimization routine can guarantee to locate the unique, global, minimum of f(x) without being fed intimate knowledge about the function by the user."

- Therefore:

1. Never accept the result using a single optimization run; always test the minimum using a different method.
2. Check that the result of the minimization does not have parameter values at the edges of the parameter space. If this happens, then the fit must be disregarded since the minimum lies outside the space that has been searched, or the minimization missed the minimum.
3. Get a feel for the range of values of the fit statistic, and the stability of the solution, by starting the minimization from several different parameter values.
4. Always check that the minimum "looks right" using a plotting tool.

Fitting: Optimization Methods

- "Single - shot" routines, e.g, Simplex and Levenberg-Marquardt in Sherpa

  start from a guessed set of parameters, and then search to improve the parameters in a continuous fashion:

  - Very Quick
  - Depend critically on the initial parameter values
  - Investigate a local behaviour of the statistics near the guessed parameters, and then make another guess at the best direction and distance to move to find a better minimum.
  - Continue until all directions result in increase of the statistics or a number of steps has been reached

- "Scatter-shot" routines, e.g. Monte Carlo in Sherpa

  examines parameters over the entire permitted parameter space to see if there are better minima than near the starting guessed set of parameters.

Statistical Issues

- Bias
- Goodness of Fit
- Background Subtraction
- Rebinning
- Errors

Statistical Issues: Background Subtraction

- A typical "dataset" may contain multiple spectra, e.g source and "background" counts, and one or more with only "background" counts.
  - The "background" may contain cosmic and particle contributions, *etc.*, but we'll ignore this complication and drop the quote marks.
- If possible, one should model background data:
  - ⇒ Simultaneously fit a background model M_B to the background dataset(s) B_j , and a source plus background model $M_S + M_B$ to the raw dataset *D*.
  - ⇒ The background model parameters must have the same values in both fits, *i.e.*, do not fit the background data first, separately.
  - ⇒ Maximize L_b x $L_{\mathrm{S+B}}$ or minimize
- However, many X-ray astronomers continue to subtract the background data from the raw data:

$$D_i' = D_i - \beta_D t_D \left[\frac{\sum_{j=1}^{n} B_{i,j}}{\sum_{j=1}^{n} \beta_{B_j} t_{B_j}} \right].$$

n is the number of background datasets, *t* is the observation time, and b is the "backscale" (given by the BACKSCAL header keyword value in a **PHA** file), typically defined as the ratio of data extraction area to total detector area.

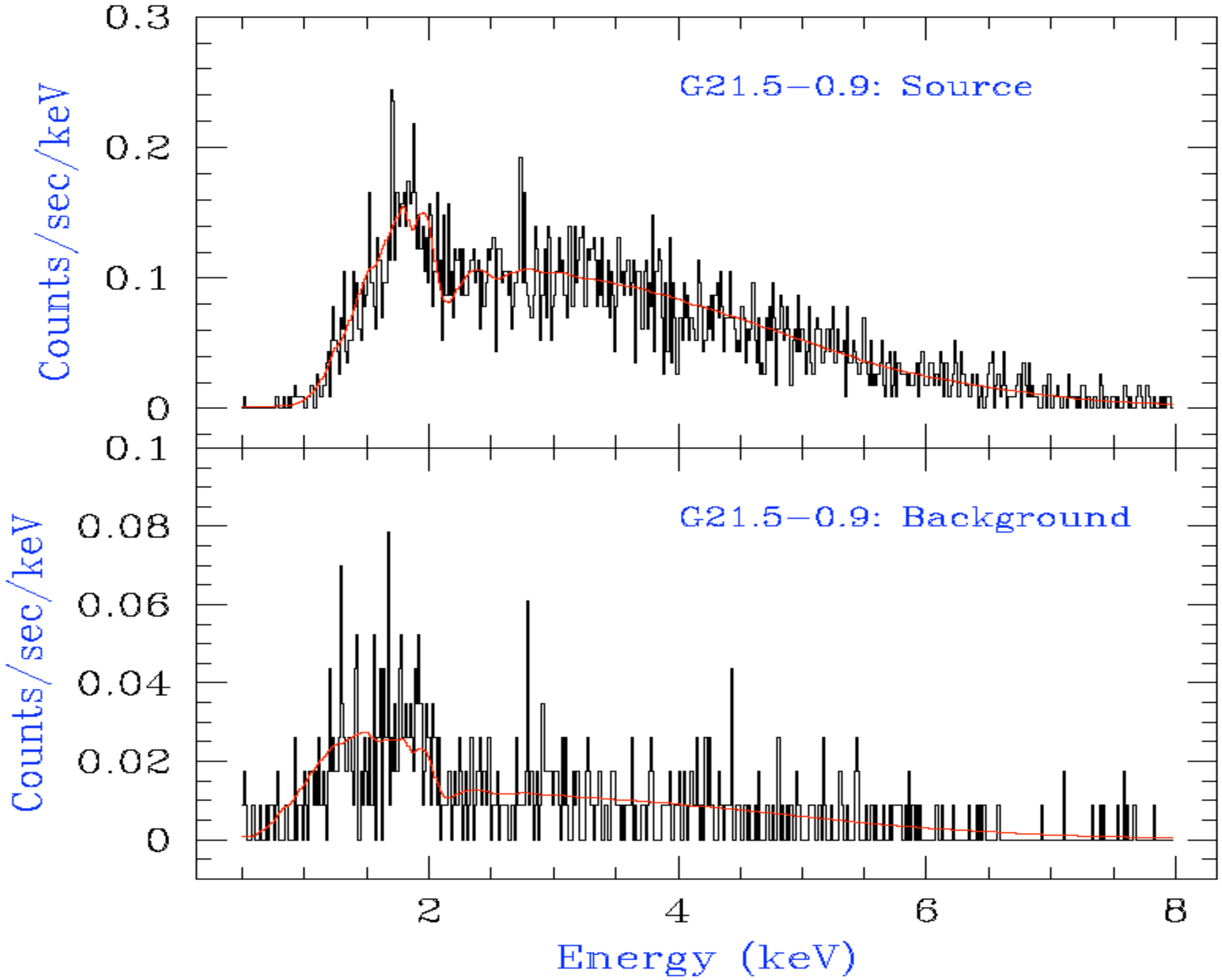


Figure 8: *Top:* Best-fit of a power-law times galactic absorption model to the source spectrum of supernova remnant G21.5-0.9. *Bottom:* Best-fit of a separate power-law times galactic absorption model to the background spectrum extracted for the same source.

Statistical Issues: Background Subtraction

- Why subtract the background?
  - It may be difficult to select an appropriate model shape for the background.
  - Analysis proceeds faster, since background datasets are not fit.
  - "It won't make any difference to the final results."
- Why not subtract the background?
  - The data D_i' are not Poisson-distributed -- one cannot fit them with the Poisson likelihood. (Variances are estimated via *error propagation*:

$$
\begin{aligned}
V[f\{X_1,...,X_m\}] &\approx \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{\partial f}{\partial \mu_i}\frac{\partial f}{\partial \mu_j}\mathrm{cov}(X_i, X_j) \\
&\approx \sum_{i=1}^{m}\left(\frac{\partial f}{\partial \mu_i}\right)^2 V[X_i] \\
\Rightarrow V[D_i'] &\approx V[D_i] + \sum_{j=1}^{n}\left(\frac{\beta_D t_D}{\beta_{B_j} t_{B_j}}\right)^2 V[B_{i,j}]) \,.
\end{aligned}
$$

  - It may well make a difference to the final results:
    - ⁎ Subtraction reduces the amount of statistical information in the analysis quantitative accuracy is thus reduced.
    - ⁎ Fluctuations can have an adverse effect, in, *e.g.*, *line detection*.

Statistical Issues: Rebinning

- *Rebinning data invariably leads to a loss of statistical information!*
- Rebinning is not necessary if one uses the Poisson likelihood to make statistical inferences.
- However, the rebinning of data may be necessary to use χ^2 statistics, if the number of counts in any bin is <= 5. In X-ray astronomy, rebinning (or *grouping*) of data may be accomplished with:
  - **grppha** in *FTOOLS*
  - **dmgroup**, in *CIAO*.
  - Interactive grouping with **group_** functions in Sherpa

One common criterion is to sum the data in adjacent bins until the sum equals five (or more).

- Caveat: always estimate the errors in rebinned spectra using the new data in each new bin (since these data are still Poisson-distributed), rather than propagating the errors in each old bin.

  ⇒ For example, if three bins with numbers of counts 1, 3, and 1 are grouped to make one bin with 5 counts, one should estimate $V[D' = 5]$ and *not* $V[D'] = V[D_1 = 1] + V[D_2 = 3] + V[D_3 = 1]$. The propagated errors may overestimate the true errors.

Statistical Issues: Systematic Errors

- There are two types of errors: statistical and systematic errors.
- Systematic errors are uncertainties in instrumental calibration:
- Assuming: exposure time t, perfect energy resolution, an effective area A_i with the uncertainty $\sigma_{A,i}$.
  - the expected number of counts in bin i $D_i = D_{g,i}(\Delta E)tA_i$.
  - the uncertainty in D_i

$$\sigma_{D_i} = D_{g,i}(\Delta E)t\sigma_{A,I} = D_{g,i}(\Delta E)tf_iA_i = f_iD_i$$

- f_iD_i - the systematic error ; in PHA files, the quantity f_i is recorded in the SYS_ERR column.
- Systematic errors are added in quadrature with statistical errors in χ^2 fitting then $\sigma_i = (D_i+D_if_i)^{1/2}$
- To account for systematic errors in a Poisson likelihood fit, one must incorporate them into the model, as opposed to simply adjusting the estimated errors.

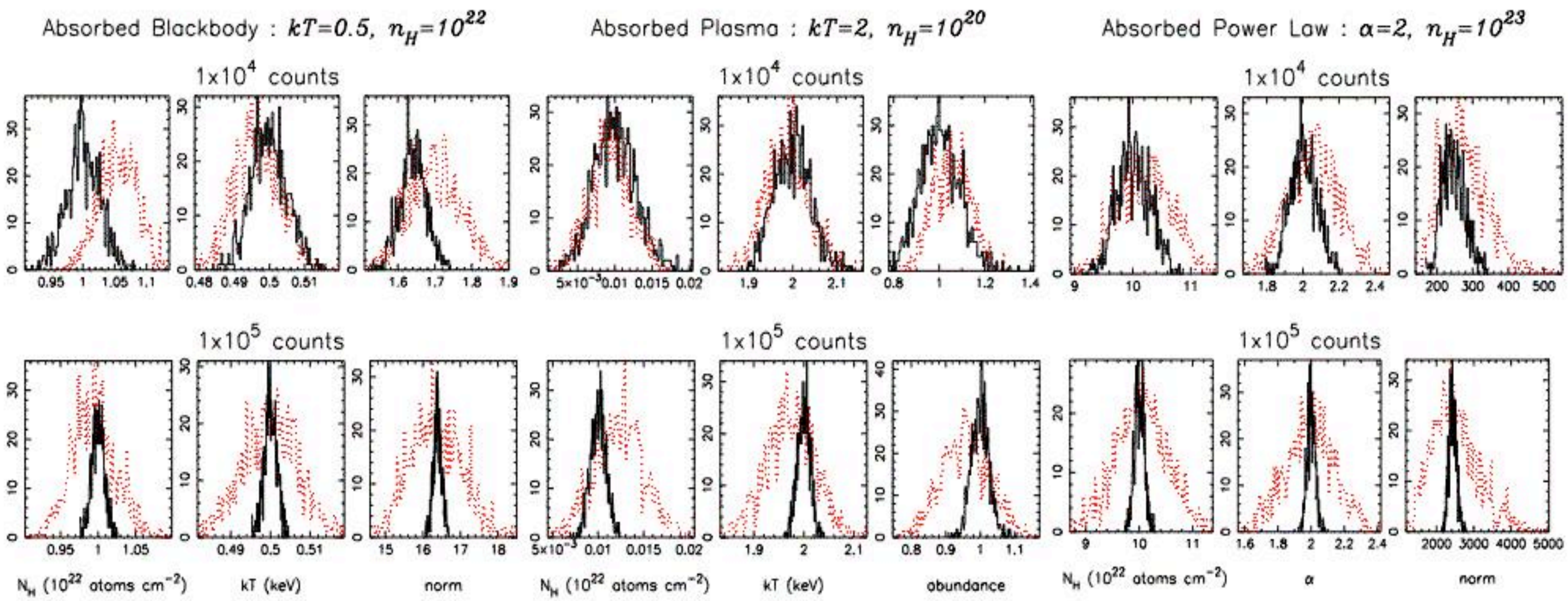
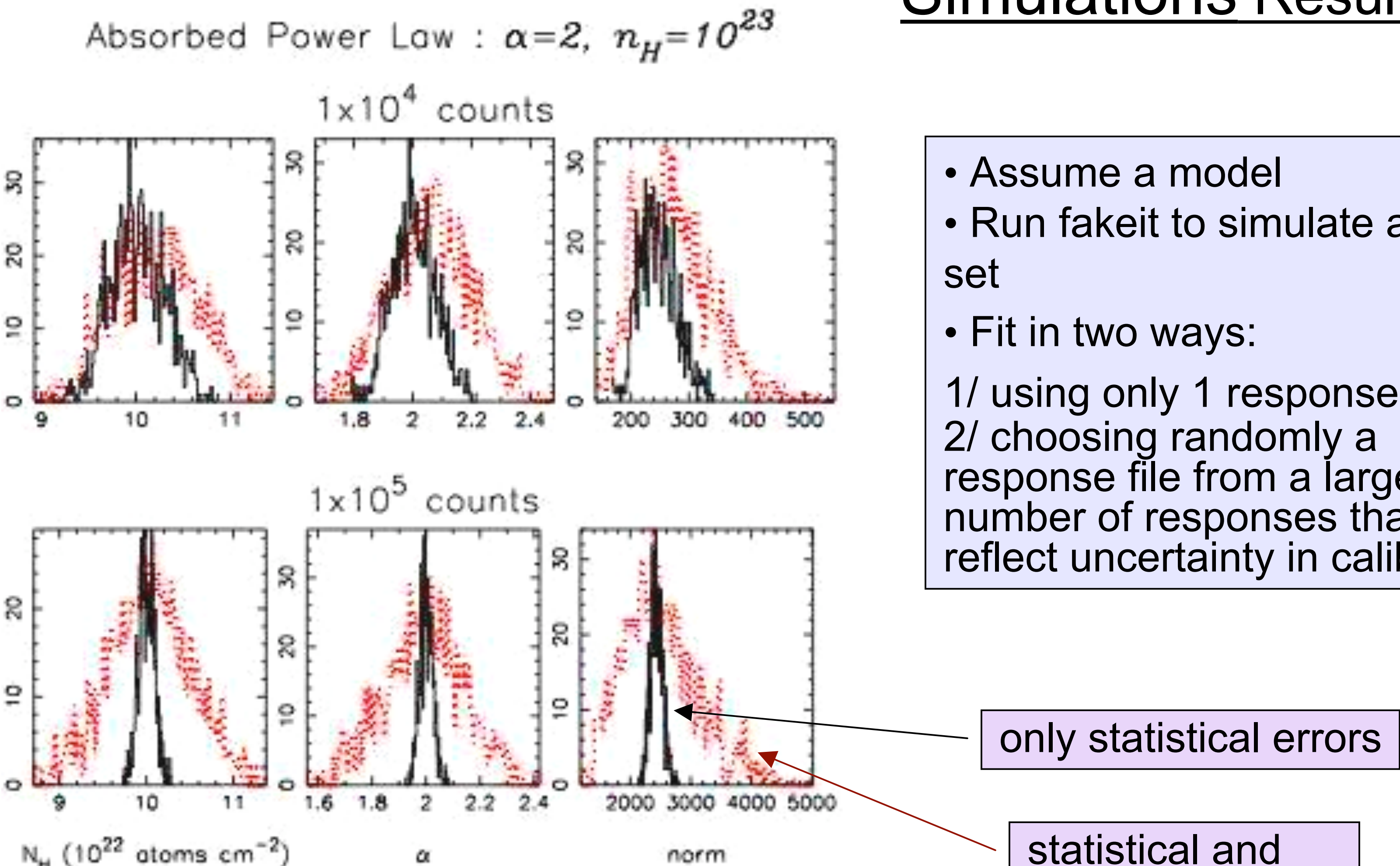


Figure 5. Frequency distributions of best-fit parameters obtained for typical blackbody, thermal plasma and powerlaw models from XSPEC for synthetic data sets containing 10^4 (upper panels) and 10^5 (lower panels). Black histograms are distributions resulting from 1000 Monte Carlo samplings of the synthetic data allowing Poisson noise variations alone. Red histograms are the distributions of parameters resulting from fits to a single synthetic data set using 1000 Monte Carlo-generated effective areas and response matrices.

Drake et al 2006, SPIE meeting

Simulations Results



- Assume a model
- Run fakeit to simulate a data set
- Fit in two ways:

1/ using only 1 response
2/ choosing randomly a response file from a large number of responses that reflect uncertainty in calibration

Systematic errors may dominate!

Final Analysis Steps

- How well are the model parameters constrained by the data?
- Is this a correct model?
- Is this the only model?
- Do we have definite results?
- What have we learned, discovered?
- How our source compares to the other sources?
- Do we need to obtain a new observation?

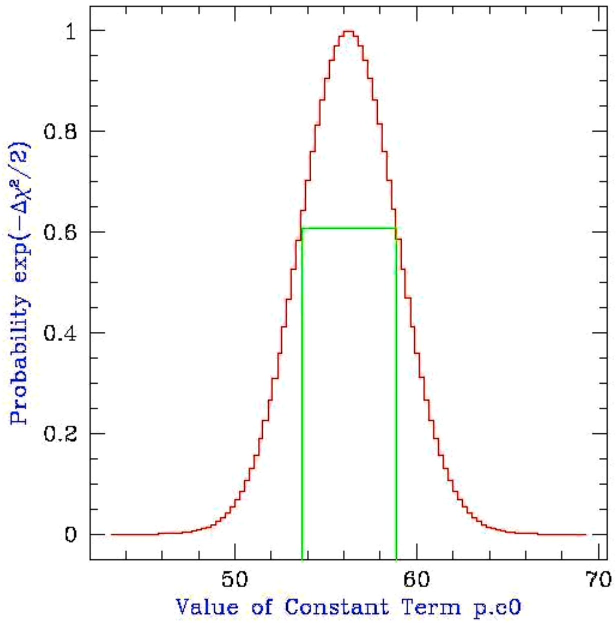
Confidence Limits

Essential issue = after the best-fit parameters are found estimate the confidence limits for them. The region of confidence is given by (Avni 1976):

$$\chi^2_{\alpha} = \chi^2_{min} + \Delta(\nu,\alpha)$$

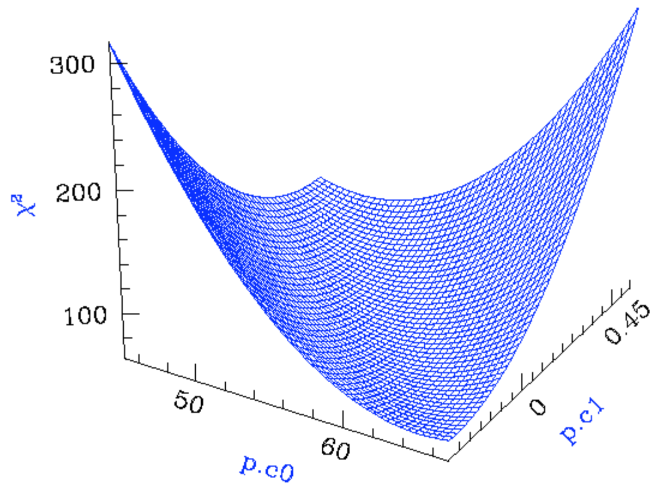
ν - degrees of freedom
α - significance
χ^2_{min} - minimum

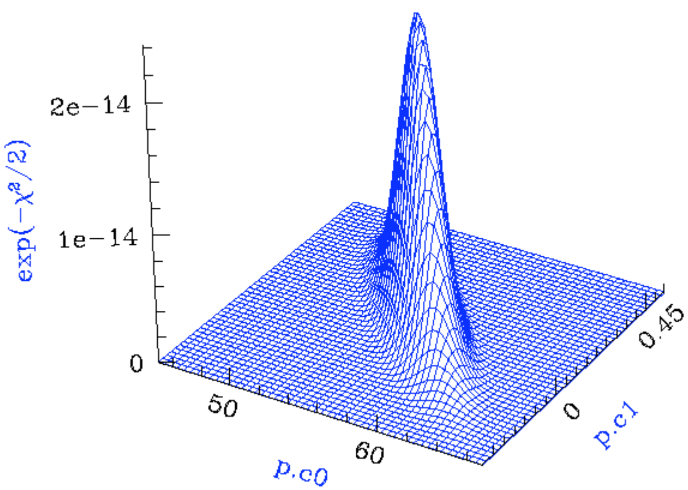
Δ depends only on the number of parameters involved nor on goodness of fit



Significance α	Number of parameters 1	2	3
0.68	1.00	2.30	3.50
0.90	2.71	4.61	6.25
0.99	6.63	9.21	11.30

CalculatingConfidence Limits means Exploring the Parameter Space - Statistical Surface





Example of a "well-behaved" statistical surface in parameter space, viewed as

1/ a multi-dimensional paraboloid (χ^2, *top*),
2/ a multi-dimensional Gaussian (exp(-χ^2 /2) ≈ $\mathcal{L}$, *bottom*).

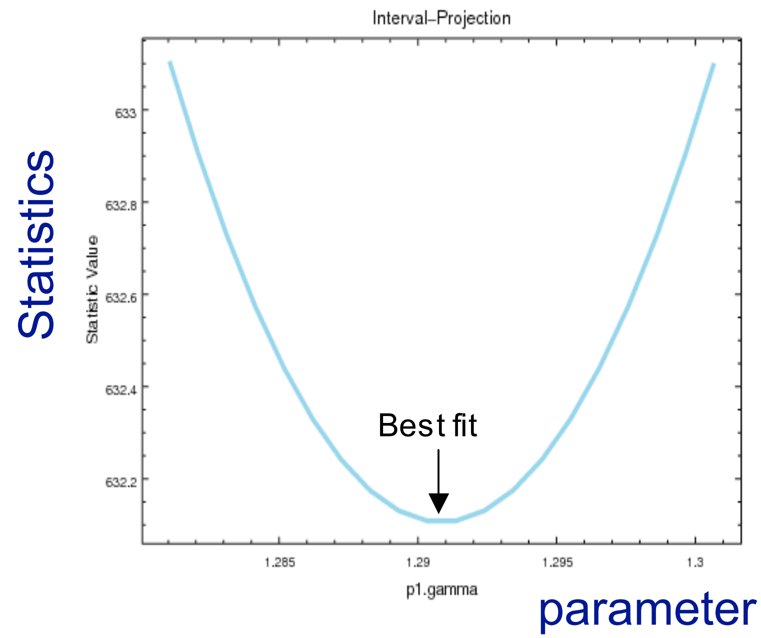
Confidence Intervals

sherpa-39> **covariance()**

```
Dataset              = 1
Confidence Method    = covariance
Fitting Method       = levmar
Statistic            = chi2datavar
covariance 1-sigma (68.2689%) bounds:
   Param            Best-Fit  Lower Bound  Upper Bound
   -----            --------  -----------  -----------
   zabs1.nh        0.0960949  -0.00436915   0.00436915
   p1.gamma          1.29086  -0.00981129   0.00981129
   p1.ampl       0.000707365 -6.70421e-06  6.70421e-06
```

sherpa-40> **projection()**

```
Dataset              = 1
Confidence Method    = projection
Fitting Method       = levmar
Statistic            = chi2datavar
projection 1-sigma (68.2689%) bounds:
   Param            Best-Fit  Lower Bound  Upper Bound
   -----            --------  -----------  -----------
   zabs1.nh        0.0960949  -0.00435835   0.00439259
   p1.gamma          1.29086  -0.00981461   0.00983253
   p1.ampl       0.000707365 -6.68862e-06    6.7351e-06
```

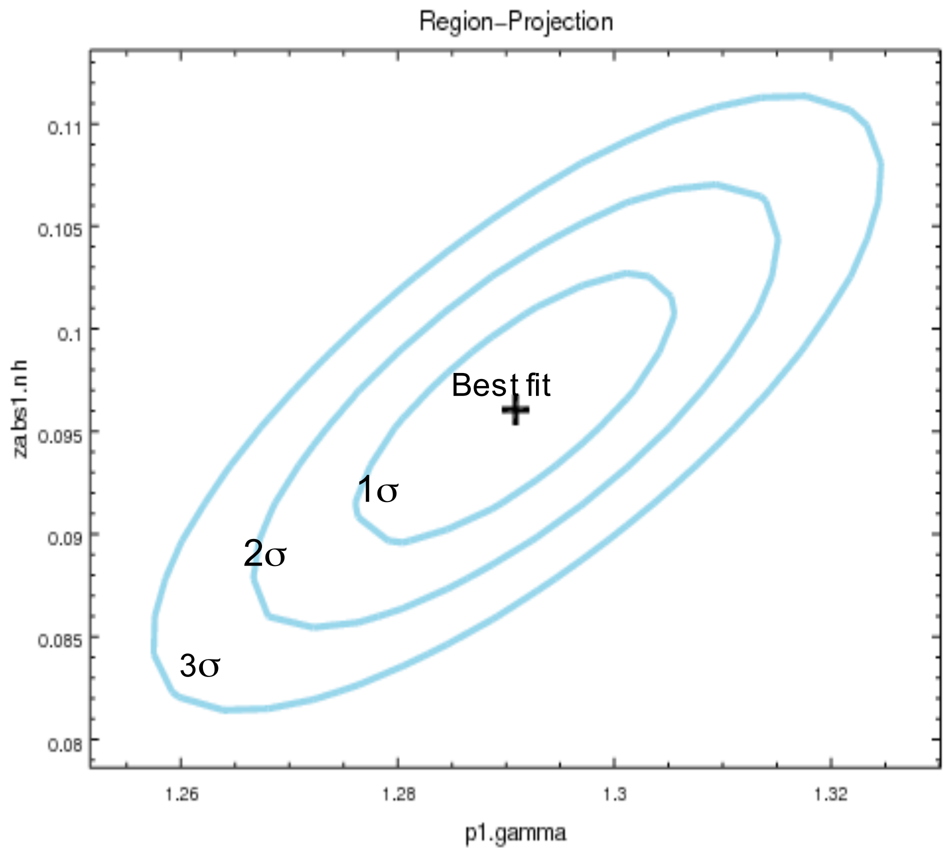


sherpa-48> **print get_proj_results()**

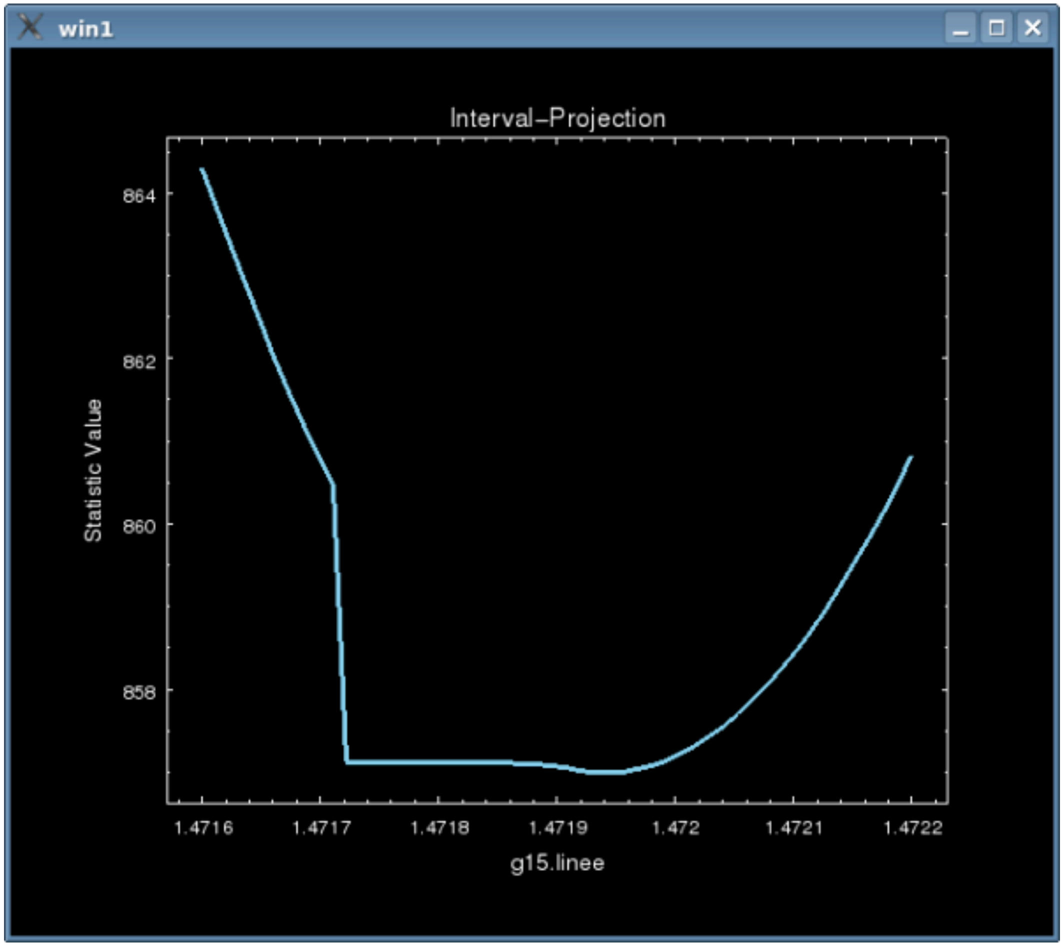
```
datasets   = (1,)
methodname = projection
fitname    = levmar
statname   = chi2datavar
sigma      = 1
percent    = 68.2689492137
parnames   = ('zabs1.nh', 'p1.gamma', 'p1.ampl')
parvals    = (0.0960948525609, 1.29085977295, 0.000707365006941)
parmins    = (-0.00435834667074, -0.00981460960484, -6.68861977704e-06)
parmaxes   = (0.0043925901652, 0.00983253275984, 6.73510303179e-06)
nfits      = 46
```

Confidence Regions


```
sherpa-61> reg_proj(p1.gamma,zabs1.nh,nloop=[20,20])
sherpa-62> print get_reg_proj()
min     = [ 1.2516146   0.07861824]
max     = [ 1.33010494  0.11357147]
nloop   = [20, 20]
fac     = 4
delv    = None
log     = [False False]
sigma   = (1, 2, 3)
parval0 = 1.29085977295
parval1 = 0.0960948525609
levels  = [ 634.40162888  638.28595426
```



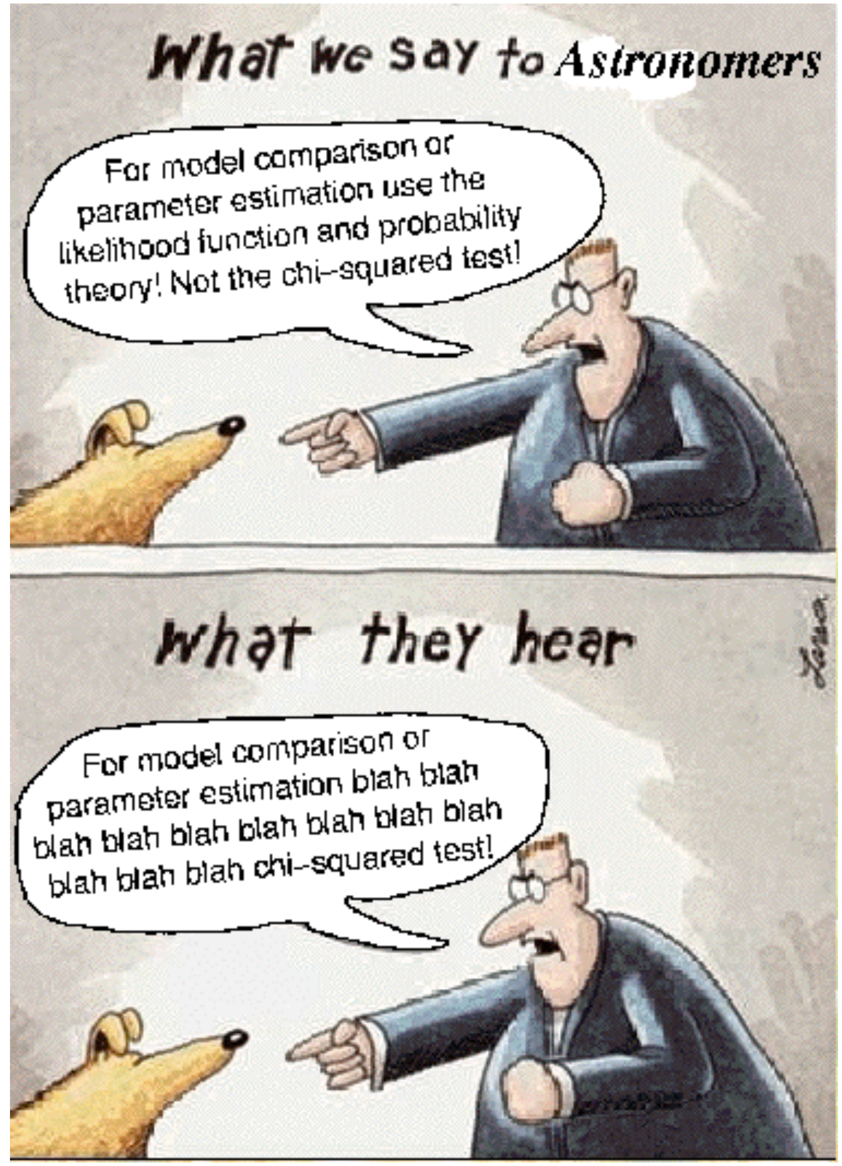
Not well-behaved Surface



Non-Gaussian Shape

Why do we talk about statistics?

Why do we talk about statistics?



Hierarchical Models, Hyperparameters and Hyperpriors

- Hierarchical models => models of models
- Hyperparameters => parameters of the hierarchical models

Example: an absorbed power law model fit to an X-ray spectrum

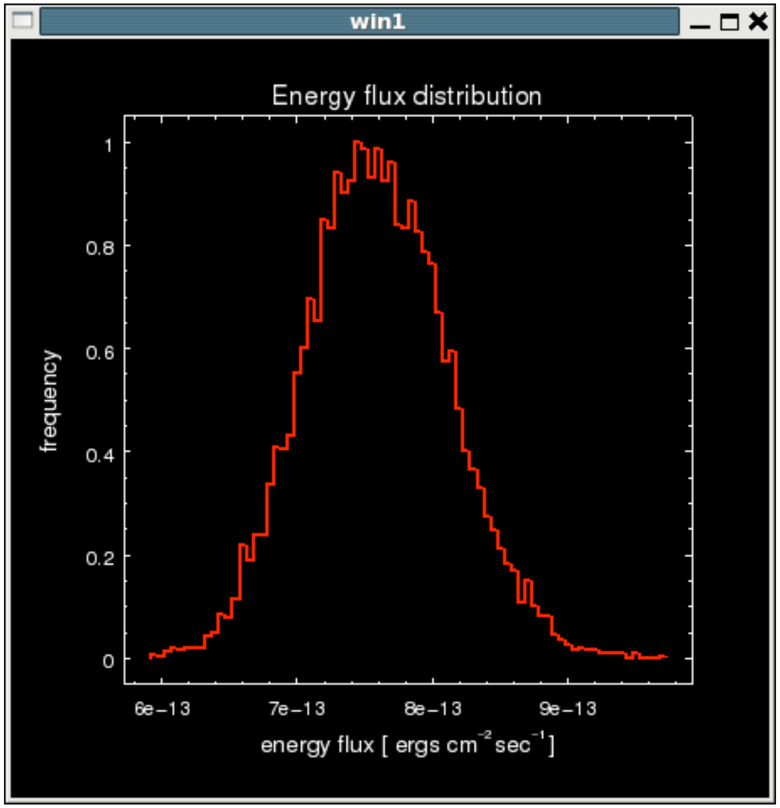
$(N_H, \Gamma, \text{Norm})$

Flux links all 3 parameters

- Hyperpriors => priors on the hyperparameter, e.g. on flux

Flux Uncertainties

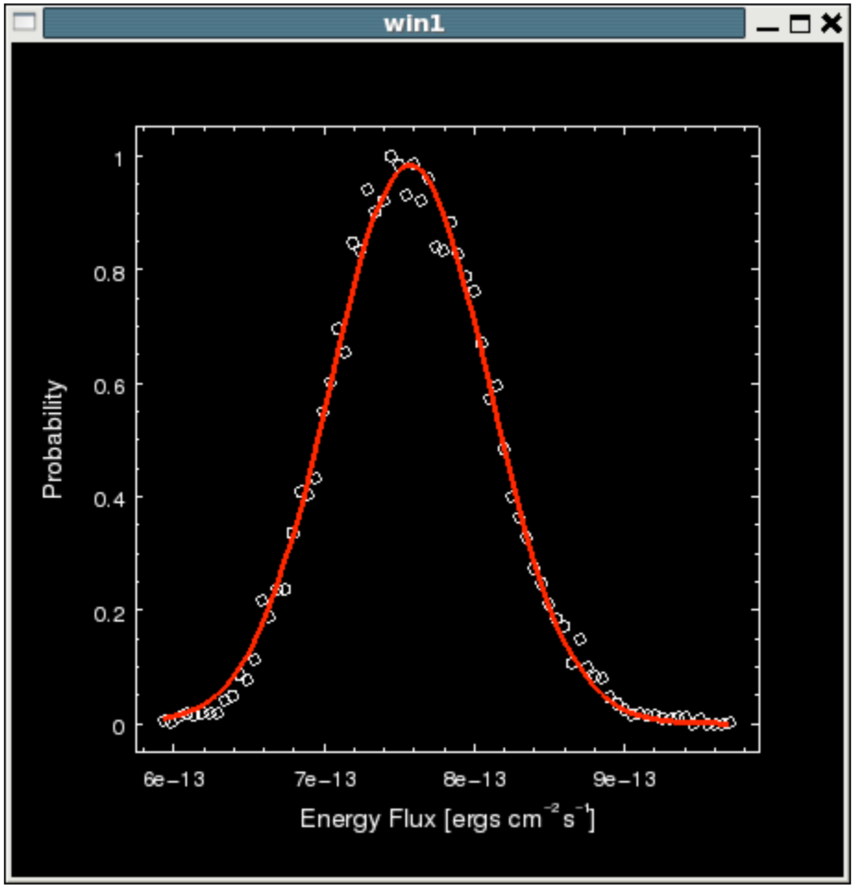
- Run simulations: sample model parameters from their distributions, e.g. Gaussians, or mulit-variate Gaussians
- Calculated flux for each data set - create flux distribution
- Derive the properties of the distribution - fit shape, find mode, mean…
- Is it normal?
- Need to determine a required confidence level from the distribution - calculate 68% or 99% bounds



Flux uncertainties: Sherpa Example

- Use Python contributed module in Sherpa: contrib_sherpa.flux_dist
- Check thread on the web

```
# simulate fluxes based on model for a sample size
of 100

flux100=sample_energy_flux(0.5,7.,num=100)
# run 10e4 flux simulations and create a histogram
from the results

hist10000=get_energy_flux_hist(0.5,7.,num=10000)
plot_energy_flux(0.5,7.,recalc=False)

fluxes = flux100[:,0]

numpy.mean(fluxes)

numpy.median(fluxes)

numpy.std(fluxes)

# determine 95% and 50% quantiles using numpy
array sorting

sf=numpy.sort(fluxes)

sf[0.95*len(fluxes)-1]
sf[0.5*len(fluxes)-1]
```



How to choose between different models?

Hypothesis Testing

Steps in the X-ray Data Analysis:

1/ Obtain the data (observe or archive)

2/ Reduce Data => standard processing or reprocessed, extract an image or a spectrum, include appropriate calibration files

3/ Analysis – fit the data with assumed model (choice of model - our prior knowledge)

4/ Conclude: which model describe the data best?

Hypothesis Testing!

5/ Reflect - what did we learn? Do we need more observations? What type?

A model *M* has been fit to dataset *D* :

- the maximum of the likelihood function $\mathcal{L}_{max}$,
- the minimum of the χ^2 statistic χ^2_{min},
- or the mode of the posterior distribution

Model Comparison. The determination of which of a suite of models (*e.g.,* blackbody, power-law, *etc.*) best represents the data.

Parameter Estimation. The characterization of the sampling distribution for each best-fit model parameter (*e.g.,* blackbody temperature and normalization), which allows the errors (*i.e.,* standard deviations) of each parameter to be determined.

Steps in Hypothesis Testing

1/ Set up 2 possible exclusive hypotheses - two models:

M0 – null hypothesis – formulated to be rejected

M1 – an alternative hypothesis, research hypothesis

each has associated terminal action

2/ Specify a priori the significance level α

choose a test which:
- approximates the conditions
- finds what is needed to obtain the sampling distribution and the region of rejection, whose area is a fraction of the total area in the sampling distribution

3/ Run test: reject ***M0*** if the test yields a value of the statistics whose probability of occurance under ***M0*** is $< \alpha$

4/ Carry on terminal action

STEPS AGAIN

Two models, M_0 and M_1 **,** have been fit to *D.* M_0 **,** the “simpler” of the two models (generally speaking, the model with fewer free parameters) is the *null hypothesis.*

Frequentists compare these models by:

- constructing a test statistic *T* from the best-fit statistics of each fit (*e.g.*, $\Delta\chi^2 = \chi^2_0 - \chi^2_1$);
- determining each sampling distributions for T, $p(T\,|\,M_0)$ and $p(T\,|\,M_1)$;
- determining the *significance*, or Type I error, the probability of selecting M_1 when M_0 is correct:

$$\alpha = \int_{T_{obs}} p(T|M_0)$$

- and determing the *power*, or Type II error, which is related to the probability β of selecting M_0 when M_1 is correct:

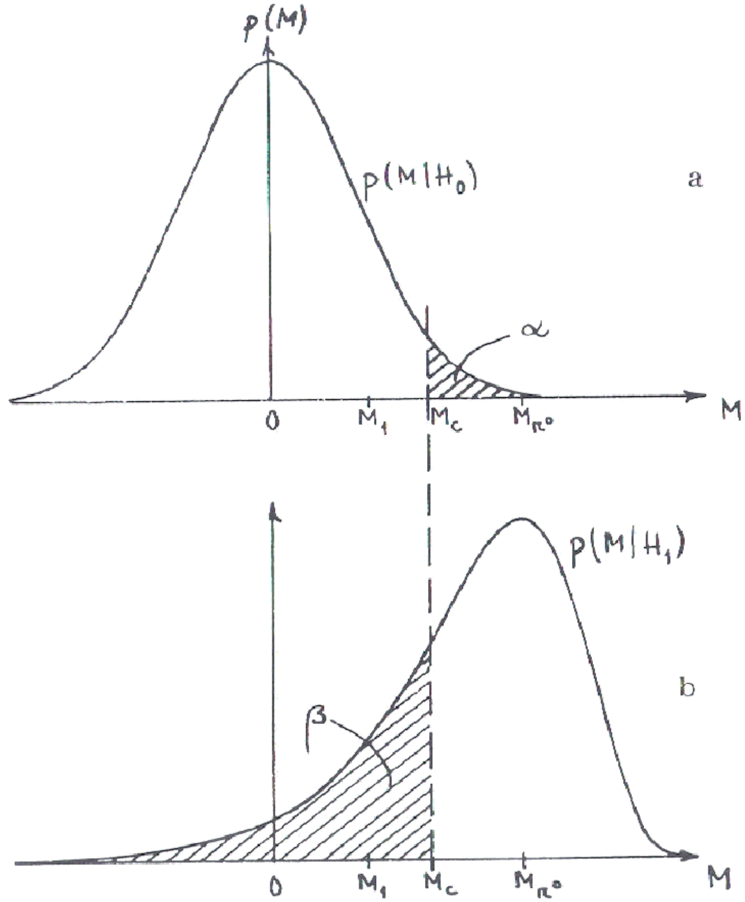
$$1-\beta = \int_{T_{obs}} p(T|M_1)$$

⇒ If α is smaller than a pre-defined threshold (≤ 0.05, or ≤ 10^{-4}, *etc.*, with smaller thresholds used for more controversial alternative models), then the frequentist rejects the null hypothesis - M_0 model.

⇒ If there are several model comparison tests to choose from, the frequentist uses the most powerful one!

α-significance

1-β – power of test



Comparison of distributions $p(T \mid M_0)$ (from which one determines the significance α) and $p(T \mid M_1)$ (from which one determines the power of the model comparison test $1 - \beta$) (Eadie *et al.* 1971, p.217)

Statistical Issues: Goodness-of-Fit

- The χ^2 goodness-of-fit is derived by computing

$$\alpha_{\chi^2} = \int_{\chi^2_{\rm obs}}^{\infty} d\chi^2\, p(\chi^2 \mid N-P)$$

$$= \frac{1}{2\Gamma\left(\frac{N-P}{2}\right)} \int_{\chi^2_{\rm obs}}^{\infty} d\chi^2 \left(\frac{\chi^2}{2}\right)^{\frac{N-P}{2}-1} e^{-\frac{\chi^2}{2}} \ .$$

  This can be computed numerically using, *e.g.*, the **GAMMQ** routine of *Numerical Recipes*.

- A typical criterion for rejecting a model is $\alpha < 0.05$ (the "95% criterion"). However, using this criterion blindly *is not recommended*!
- A quick'n'dirty approach to building intuition about how well your model fits the data is to use the *reduced* χ^2, *i.e.,* $\chi^2_{\rm obs,r} = \chi^2_{\rm obs} / (N-P)$:
  - A "good" fit has $\chi^2_{\rm obs,r} \approx 1$
  - If $\chi^2_{\rm obs} \rightarrow 0$ the fit is "too good" -- which means (1) the error bars are too large, (2) $\chi^2_{\rm obs,r}$ is *not* sampled from the χ^2 distribution, and/or (3) the data have been fudged.

The reduced χ^2 should never be used in any mathematical computation if you are using it, you are probably doing something wrong!

Frequentist Model Comparison

Standard frequentist model comparison tests include:

The χ^2 Goodness-of-Fit (GoF) test:

$$\alpha_{\chi^2} = \int_{\chi^2_{\min,0}}^{\infty} d\chi^2\, p(\chi^2 \mid N - P_0) = \frac{1}{2\Gamma(\frac{N-P_0}{2})} \int_{\chi^2_{\min,0}}^{\infty} d\chi^2 \left(\frac{\chi^2}{2}\right)^{\frac{N-P_0}{2}-1} e^{-\frac{\chi^2}{2}} .$$

The Maximum Likelihood Ratio (MLR) test:

t:

$$\alpha_{\chi^2 MLR} = \int_{\Delta\chi^2}^{\infty} d\chi^2\, p(\Delta\chi^2 \mid \Delta P),$$

where ΔP is the number of additional freely varying model parameters in model M_1

The F-test:

$$F = \frac{\Delta\chi^2}{\Delta P} \Big/ \frac{\chi_1^2}{(N - P_1)} .$$

where P_1 is the total number of thawed parameters in model M_1

These are standard tests because they allow estimation of the significance without time-consuming simulations!

Bayesian Model Comparison

Bayes' theorem can also be applied to model comparison:

$$p(M \mid D) = p(M)\,\frac{p(D \mid M)}{p(D)}.$$

- $p(M)$ is the prior probability for M;
- $p(D)$ is an ignorable normalization constant; and
- $p(D \mid M)$ is the average, or global, likelihood:

$$\begin{aligned} p(D \mid M) &= \int d\theta\, p(\theta \mid M)\, p(D \mid M, \theta) \\ &= \int d\theta\, p(\theta \mid M)\, \mathcal{L}(M, \theta). \end{aligned}$$

In other words, it is the (normalized) integral of the posterior distribution over all parameter space. Note that this integral may be computed numerically

Bayesian Model Comparison

To compare two models, a Bayesian computes the odds, or odd ratio:

$$O_{10} = \frac{p(M_1 | D)}{p(M_0 | D)}$$

$$= \frac{p(M_1) p(D | M_1)}{p(M_0) p(D | M_0)}$$

$$= \frac{p(M_1)}{p(M_0)} B_{10} \; ,$$

where B_{10} is the *Bayes factor*. When there is no *a priori* preference for either model, B_{10} = 1 of one indicates that each model is equally likely to be correct, while $B_{10} \geq 10$ may be considered sufficient to accept the alternative model (although that number should be greater if the alternative model is controversial).

Model Comparison Tests:

Notes and caveats regarding these standard tests:

- The GoF test is an "alternative-free" test, as it does not take into account the alternative model M_1. It is consequently a *weak* (*i.e.,* not powerful) model comparison test and should not be used!
- Only the version of *F*-test which generally has the greatest power is shown above: in principle, one can construct three *F* statistics out of $\Delta\chi^2$ χ^2_0 χ^2_1
- The MLR ratio test is generally the most powerful for detecting emission and absorption lines in spectra.

But the most important caveat of all is that…

The *F* and ***MLR*** tests are often misused by astronomers!

There are two important conditions that must be met so that an estimated derived value α is actually correct, *i.e.,* so that it is an accurate approximation of the tail of the sampling distribution (Protassov *et al.* 2001):

- *M_0 must be nested within M_1*, *i.e.,* one can obtain M_0 by setting the extra ΔP parameters of M_1 to default values, often zero; and
- *those default values may not be on a parameter space boundary.*

The second condition may not be met, *e.g.,* when one is attempting to detect an emission line, whose default amplitude is zero and whose minimum amplitude is zero. Protassov *et al.* recommend Bayesian posterior predictive probability values as an alternative,

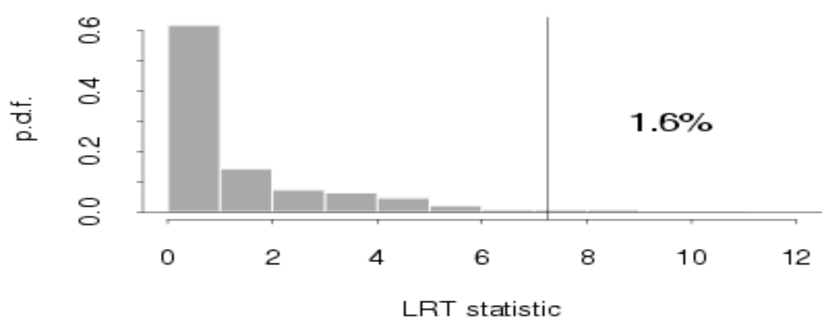
---

If the conditions for using these tests are not met, then they can still be used, but the significance must be computed via Monte Carlo simulations.

Monte Carlo Simulations

- Simulations to test for more complex models, e.g. addition of an emission line
- Steps:
  - Fit the observed data with both models, M_0, M_1
  - Obtain distributions for parameters
  - Assume a simpler model M_0 for simulations
  - Simulate/Sample data from the assumed simpler model
  - Fit the simulated data with simple and complex model
  - Calculate statistics for each fit
  - Build the probability density for assumed comparison statistics, e.g. LRT and calculate p-value

Example:
Visualization, here accept more complex model, p-value 1.6%



Observations: Chandra Data and more…

- X-ray Spectra

  typically PHA files with the RMF/ARF calibration files
- X-ray Images

  FITS images, exposure maps, PSF files
- Lightcurves

  FITS tables, ASCII files
- Derived functional description of the source:
  - Radial profile
  - Temperatures of stars
  - Source fluxes
- Concepts of Source and Background data

Observations: Data I/O in Sherpa

- Load functions (PyCrates) to input the data:
  - data: load_data, load_pha, load_arrays, load_ascii
  - calibration: load_arf, load_rmf load_multi_arfs, load_multi_rmfs
  - background: load_bkg, load_bkg_arf , load_bkg_rmf
  - 2D image: load_image, load_psf
  - General type: load_table, load_table_model, load_user_model

Help file:
load_data([id=1], filename, [options])
load_image([id=1], filename|IMAGECrate,[coord="logical"])

Examples:
load_data("src", "data.txt", ncols=3)

load_data("rprofile_mid.fits[cols RMID,SUR_BRI,SUR_BRI_ERR]")
load_data("image.fits")
load_image("image.fits", coord="world"))

- Multiple Datasets - data id

Default data id =1
load_data(2, "data2.dat", ncols=3)

- Filtering of the data
  - load_data expressions
  - notice/ignore commands in Sherpa

Examples:
notice(0.3,8)
notice2d("circle(275,275,50)")

A Simple Problem
Fit Chandra 2D Image data in Sherpa using Command Line Interface in Python

- Read the data
- Choose statistics and optimization method
- Define the model
- Minimize to find the best fit parameters for the model
- Evaluate the best fit - display model, residuals, calculate uncertainties

A Simple Problem

List of Sherpa Commands

Read Image data and Display in ds9

```
load_image("image2.fits")
image_data()
set_coord("physical")
notice2d("circle(4065.5,4250.5,78.8)")
image_data()
```

Set Statistics and Optimization Method

```
set_stat("cash")
set_method("neldermead")
```

Define Model and Set Model parameters

```
set_model(gauss2d.g1+const2d.bgnd)
g1.ampl = 20
g1.fwhm = 20
g1.xpos = 4065.5
g1.ypos = 4250.5
freeze(g1.ellip)
freeze(g1.theta)

bgnd.c0 = 0.2
```

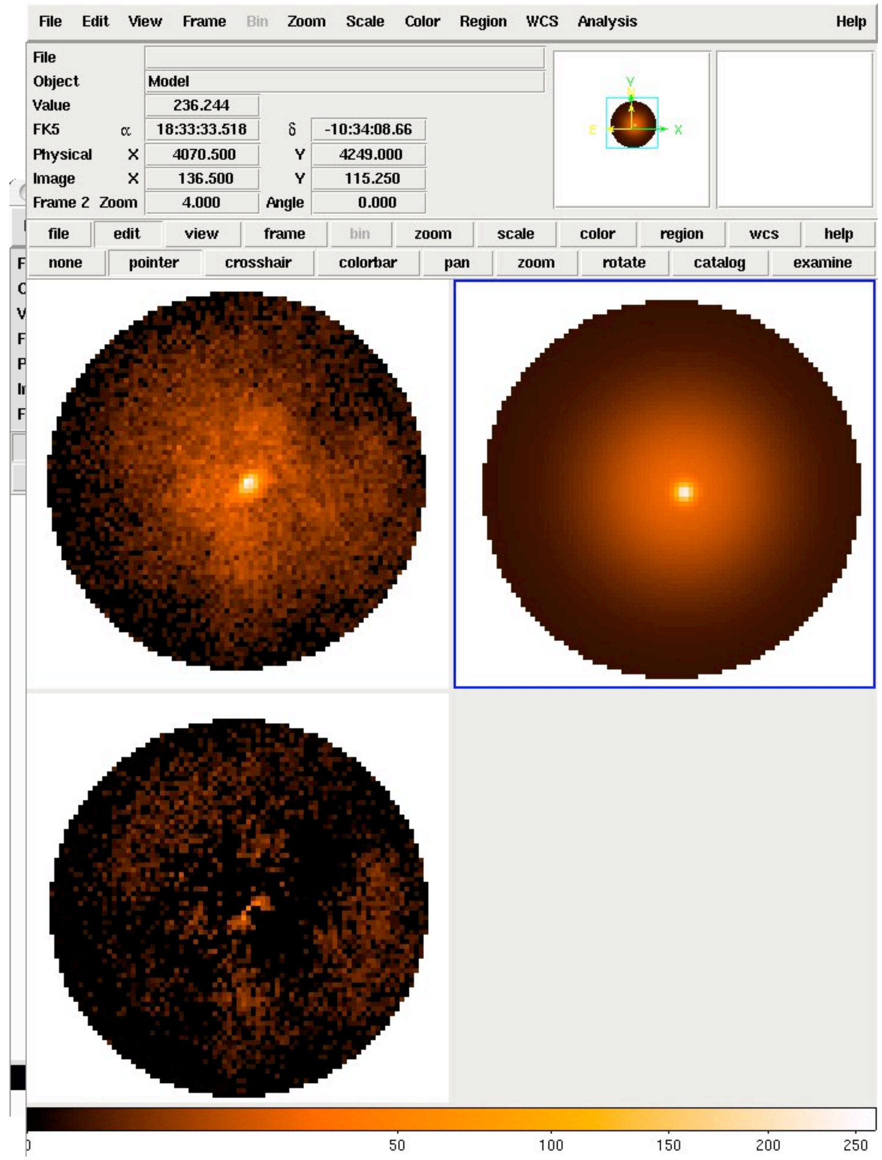
Fit, Display Get Confidence Range

```
fit()
image_fit()

projection()
```

A Simple Problem

List of Sherpa Commands

```
load_image("image2.fits")
image_data()
set_coord("physical")
notice2d("circle(4065.5,4250.5,78.8)")
image_data()

set_stat("cash")
set_method("neldermead")

set_model(gauss2d.g1+const2d.bgnd)
g1.ampl = 20
g1.fwhm = 20
g1.xpos = 4065.5
g1.ypos = 4250.5
freeze(g1.ellip)
freeze(g1.theta)

bgnd.c0 = 0.2

fit()
image_fit()

projection()
```



List of Sherpa Commands

```
load_image("image2.fits")
image_data()
set_coord("physical")
notice2d("circle(4065.5,4250.5,78.8)")
image_data()

set_stat("cash")
set_method("neldermead")

set_model(gauss2d.g1+const2d.bgnd)
g1.ampl = 20
g1.fwhm = 20
g1.xpos = 4065.5
g1.ypos = 4250.5
freeze(g1.ellip)
freeze(g1.theta)

bgnd.c0 = 0.2

fit()
image_fit

projectio
```

xterm

```
mac% sherpa fit.script.py

--------------------------------------------------------
Welcome to Sherpa: CXC's Modeling and Fitting Package
--------------------------------------------------------
Version: CIAO 4.0

Statistic value = -47094 at function evaluation 361
Data points = 4881
Degrees of freedom = 4876
   g1.fwhm        20.0002
   g1.xpos        4068.76
```

Command Line View

xterm

```
mac% sherpa

--------------------------------------------------------
Welcome to Sherpa: CXC's Modeling and Fitting Package
--------------------------------------------------------
Version: CIAO 4.0

sherpa-1> load_image("image2.fits")
sherpa-2> image_data()
sherpa-3> set_coord("physical")
sherpa-4> notice2d("circle(4065.5,4250.5,78.8)")
sherpa-5> image_data()
sherpa-6>
sherpa-6> set_stat("cash")
sherpa-7> set_method("neldermead")
sherpa-8>
sherpa-8> set_model(gauss2d.g1+const2d.bgnd)
sherpa-9> g1.ampl = 20
sherpa-10> g1.fwhm = 20
sherpa-11> g1.xpos = 4065.5
sherpa-12> g1.ypos = 4250.5
sherpa-13> freeze(g1.ellip)
sherpa-14> freeze(g1.theta)
sherpa-15>
sherpa-15> bgnd.c0 = 0.2
sherpa-16>
sherpa-16> fit()
Statistic value = -47094 at function evaluation 361
Data points = 4881
Degrees of freedom = 4876
   g1.fwhm        20.0002
   g1.xpos        4068.76
   g1.ypos        4249.38
   g1.ampl        71.674
   bgnd.c0        3.14686
sherpa-17> image_fit()
sherpa-18>
```

Setup Environment

Set the System

Import Sherpa and Chips

Define directories

```
#!/usr/bin/env python

import sys
import re
import os
from glob import glob


    from sherpa.astro.ui import *
    import pychips






def main():
    dataids = []
    dataid = 0
    dataids = {}
    obsdirs = glob('data/obs*')
```

Model Parameters

Loops

```
# set_method('powell')
set_method('neldermead')

# Define model components for the (y,z) center of thermal expansion of
# ACIS fid lights when detector housing temperature varies
create_model_component('polynom1d', 'yc')
create_model_component('polynom1d', 'zc')
set_par('yc.c0', -65.0)
set_par('zc.c0', 1394.0)
freeze('yc.c0')
freeze('zc.c0')

for iobs, obsdir in enumerate(obsdirs[:2]):
    # Create gridmodel component that has the temperature change (from the default
    # -60 C) as a function of dt.  This dataset is required to have the exact
    # same gridding as the data sets.
    obs_dtemp = 'obs%s_dtemp' % iobs
    if ciao4:
        load_table_model(obs_dtemp, os.path.join(obsdir, 'delta_temp.dat'))
        norm_par = '.ampl'
    else:
        create_model_component('gridmodel', obs_dtemp)
        set_par(obs_dtemp + '.file', os.path.join(obsdir, 'delta_temp.dat'))
        norm_par = '.norm'

    if (iobs == 0):
        set_par(obs_dtemp + norm_par, 1.6e-5)
        freeze(obs_dtemp + norm_par)
    else:
        link(obs_dtemp + norm_par, 'obs0_dtemp' + norm_par)

    obs_dataids = []
    for axis in ('y', 'z'):
        # Make the model component for the SIM dy and dz motion during the observation.
        # This is common to the three fid slots.  This model tracks only the motion
        # and not the constant per-slot offset
```

A Complex Example
Fit Chandra and HST Spectra with Python script

- Setup the environment
- Define model functions
- Run script and save results in nice format.
- Evaluate results - do plots, check uncertainties, derive data and do analysis of the derived data.

Setup

X-ray spectra

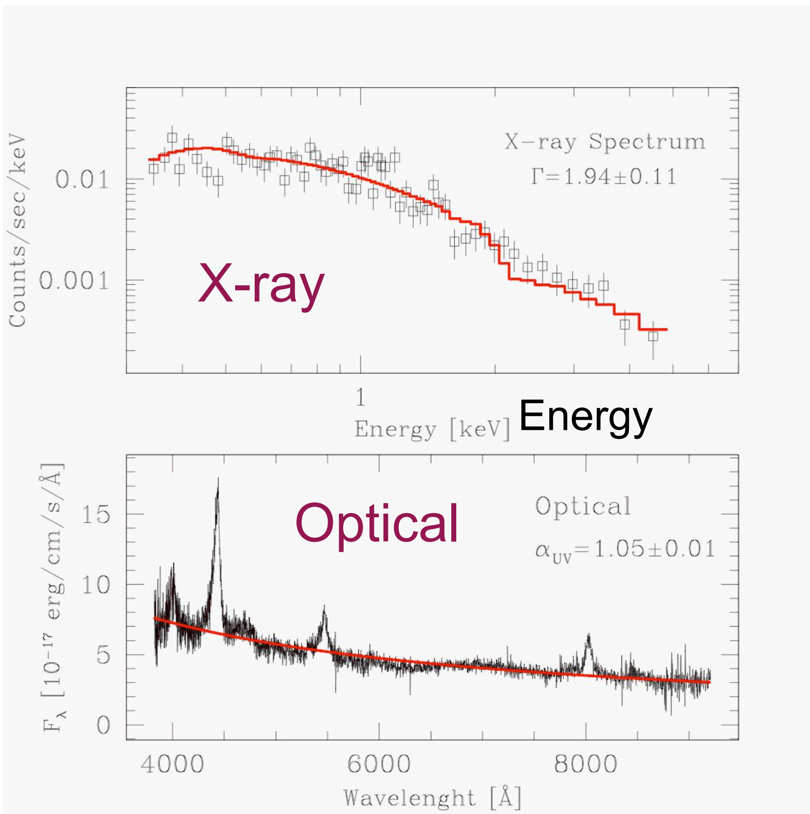
Optical spectra

Units Conversion

```
from sherpa.astro.ui import *
from sherpa.utils import rebin
import numpy

set_stats("chi2datavar")
set_method("neldermead")

load_pha(1,"acis_grp15.pha")

xray=get_data(1)
notice_id(1,0.3,7.0)
set_model(1, xswabs.abs1 * powlaw1d.pl1)
print(get_model(1))

load_data(2,"q1701_test.dat",3)
opt=get_data(2)
notice_id(2,6000,9200.)
plot_data(2)
set_model(2, powlaw1d.pl2)
print(get_model(2))


fit(1,2)

# Change Wave to Freq and nuFnu in Log
x = get_data(2).x
y = get_data(2).y

cspeed=3e10
freq=cspeed*1.e8/x
fnu=x*y*1.e-17
lfreq=numpy.log10(freq)
lnfnu=numpy.log10(fnu)

# Change X-ray Units:
# first get the flux in photons/cm2/s/keV
(counts, staterr,syserr) = get_data(1).to_fit(get_stat().calc_staterror)
```

Fit Results

X-ray data with RMF/ARF and Optical Spectra in ASCII

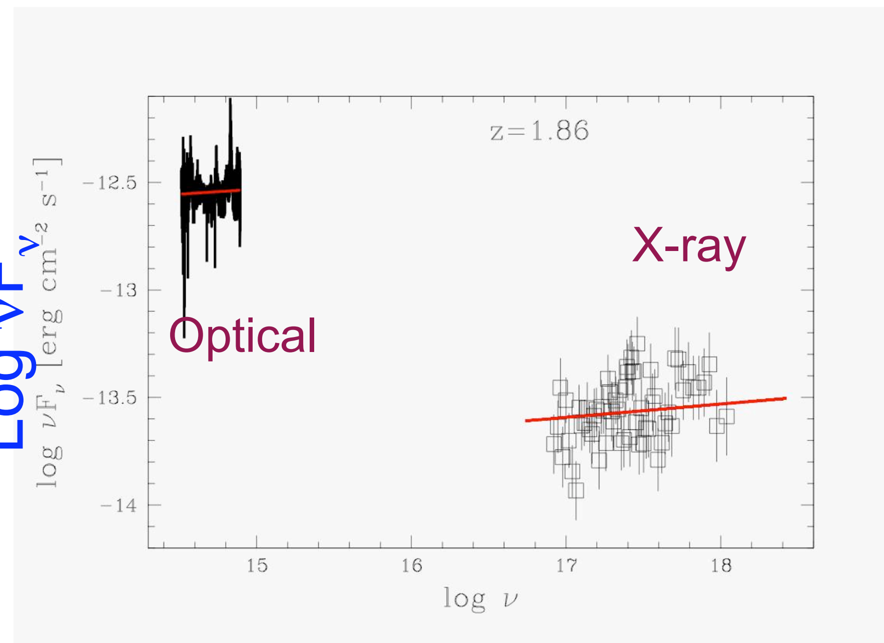


Flux

Energy

Wavelength

Quasar SED



Log νF_ν

Log ν

Learn more on Sherpa Web Pages

Last modified: 15 May 2008

Sherpa β

Sherpa 3.4 Homepage
Sherpa 4.0 Beta Homepage
Choosing Python or S-Lang
About the Sherpa Beta Release
Analysis Threads
Sherpa
Science (CIAO)
ChIPS
ChaRT
Help Pages (AHELP)
Alphabetical
By context
Using ahelp
Models, Statistics, and Methods
Models
Statistics
Documentation
Bug List
CXC Links
CIAO (Data Analysis)
ChIPS (Plotting)
Astrostatistics Collaboration

News
Previous Items

15 May 2008

The Fitting FITS Image Data thread (S-Lang or Python) has been updated for Sherpa Beta.

Sherpa β

CIAO's modeling and fitting package

WHAT'S NEW | WATCH OUT
Analysis Threads | Ahelp | Download CIAO || CIAO | ChIPS | ChaRT

The CIAO 4 release features an experimental (beta) version of the new Sherpa, the CIAO modeling and fitting package. Sherpa enables the user to construct complex models from simple definitions and fit those models to data, using a variety of statistics and optimization methods.

Sherpa is designed for use in a variety of modes: as a user-interactive application and in batch mode. Sherpa is an importable module for scripting languages (Python or S-Lang) and is available as a C/C++ library for software developers. In addition, users may write their own Python and S-Lang scripts for use in Sherpa. Refer to the Should I Use Sherpa in Python or S-Lang? page for help in deciding which language to use.

Since this is the initial phase of the Sherpa redesign, not all of the functionality of the CIAO 3.4 version is implemented yet. The About the Sherpa Beta Release page outlines new features and provides a summary of missing items. Please send feedback and questions on Sherpa Beta to the Helpdesk.

Citing *Sherpa* in a Publication

If you are writing a paper and would like to cite *Sherpa*, we recommend the following paper:

Sherpa: a mission-independent data analysis application (ADS)
P. E. Freeman, S. Doe, A. Siemiginowska
SPIE Proceedings, Vol. 4477, p.76, 2001

```
\bibitem[Freeman et al.(2001)]{2001SPIE.4477...76F} Freeman, P., Doe, S.,
\& Siemiginowska, A.\ 2001, \procspie, 4477, 76
```

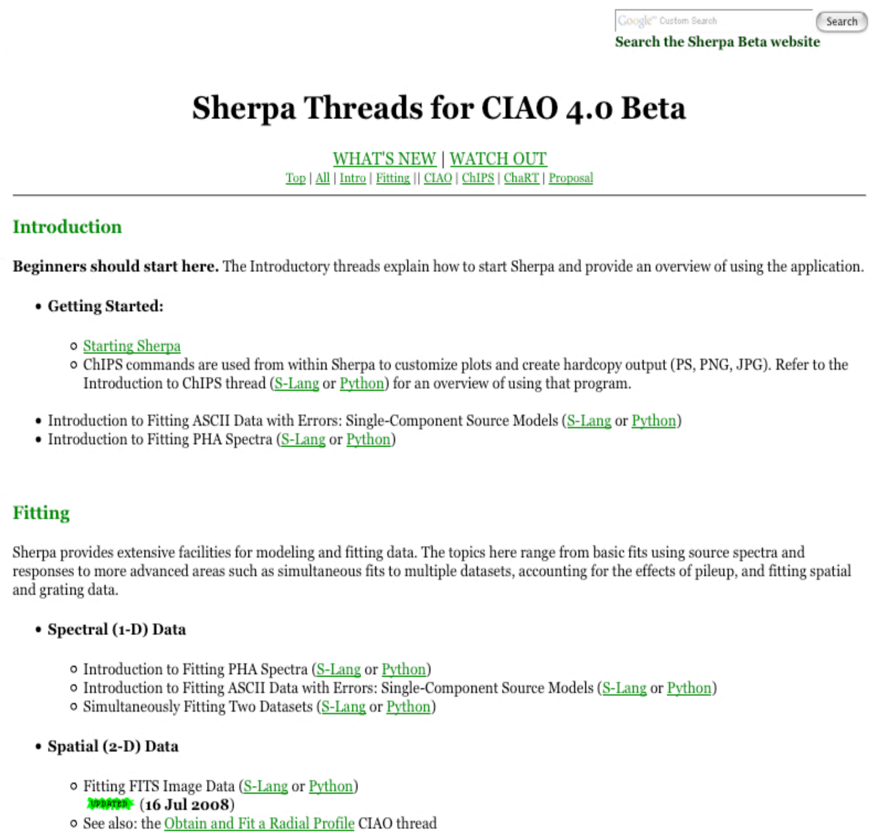
The specific version of CIAO and CALDB (if applicable) used for the analysis should be mentioned as well.

A reference for the Python interface to *Sherpa* is also available:

Developing Sherpa with Python (ADS)
S. Doe, et al.
Astronomical Data Analysis Software and Systems XVI, 376, 543

```
\bibitem[Doe et al.(2007)]{2007ASPC..376..543D} Doe, S., et al.\ 2007,
Astronomical Data Analysis Software and Systems XVI, 376, 543
```

Further guidelines are available from the Acknowledgment of Use of Chandra Resources.



Search
Search the Sherpa Beta website

Sherpa Threads for CIAO 4.0 Beta

WHAT'S NEW | WATCH OUT
Top | All | Intro | Fitting || CIAO | ChIPS | ChaRT | Proposal

Introduction

Beginners should start here. The Introductory threads explain how to start Sherpa and provide an overview of using the application.

- **Getting Started:**
  - Starting Sherpa
  - ChIPS commands are used from within Sherpa to customize plots and create hardcopy output (PS, PNG, JPG). Refer to the Introduction to ChIPS thread (S-Lang or Python) for an overview of using that program.
- Introduction to Fitting ASCII Data with Errors: Single-Component Source Models (S-Lang or Python)
- Introduction to Fitting PHA Spectra (S-Lang or Python)

Fitting

Sherpa provides extensive facilities for modeling and fitting data. The topics here range from basic fits using source spectra and responses to more advanced areas such as simultaneous fits to multiple datasets, accounting for the effects of pileup, and fitting spatial and grating data.

- **Spectral (1-D) Data**
  - Introduction to Fitting PHA Spectra (S-Lang or Python)
  - Introduction to Fitting ASCII Data with Errors: Single-Component Source Models (S-Lang or Python)
  - Simultaneously Fitting Two Datasets (S-Lang or Python)
- **Spatial (2-D) Data**
  - Fitting FITS Image Data (S-Lang or Python) UPDATED (16 Jul 2008)
  - See also: the Obtain and Fit a Radial Profile CIAO thread

Summary

- Motivation: why do we need statistics?
- Probabilities/Distributions
- Poisson Likelihood
- Parameter Estimation
- Statistical Issues
- Statistical Tests

Conclusions

Statistics is the main tool for any astronomer who need to do data analysis and need to decide about the physics presented in the observations.

References:

Peter Freeman's Lectures from the Past X-ray Astronomy School

"Practical Statistics for Astronomers", Wall & Jenkins, 2003
Cambridge University Press

Eadie et al 1976, "Statistical Methods in Experimental Physics"

Selected References

- General statistics:
  - Babu, G. J., Feigelson, E. D. 1996, Astrostatistics (London: Chapman & Hall)
  - Eadie, W. T., Drijard, D., James, F. E.,Roos, M., & Sadoulet, B. 1971, Statistical Methods in Experimental Physics (Amsterdam: North-Holland)
  - Press, W. H., Teukolsky, S. A., Vetterling, W. T.,& Flannery, B. P. 1992, Numerical Recipes (Cambridge: Cambridge Univ. Press)

- Introduction to Bayesian Statistics:
  - Loredo, T. J. 1992, in Statistical Challenges in Modern Astronomy,ed. E. Feigelson & G. Babu (New York: Springer-Verlag), 275

- Modified $\mathcal{L}$ and χ^2 statistics:
  - Cash, W. 1979, ApJ 228, 939\item Churazov, E., et al. 1996, ApJ 471, 673
  - Gehrels, N. 1986, ApJ 303, 336
  - Kearns, K., Primini, F., & Alexander, D. 1995, in Astronomical Data Analysis Software and Systems IV,eds. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 331

- Issues in Fitting:
  - Freeman, P. E., et al. 1999, ApJ 524, 753 (and references therein)

- *Sherpa* and *XSPEC*:
  - Freeman, P. E., Doe, S., & Siemiginowska, A. 2001, astro-ph/0108426
  - http://asc.harvard.edu/ciao/download/doc/sherpa_html_manual/index.html
  - Arnaud, K. A. 1996, in Astronomical Data Analysis Software and Systems V, eds. G. H. Jacoby & J. Barnes (San Francisco: ASP), 17
  - http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec/manual/manual.html

Example

From a paper by Martinez-Sansigre et al published in Aug 4, 2005 issue of *Nature*

What is the fraction of the unobscured quasars?

Use IR Spitzer observations

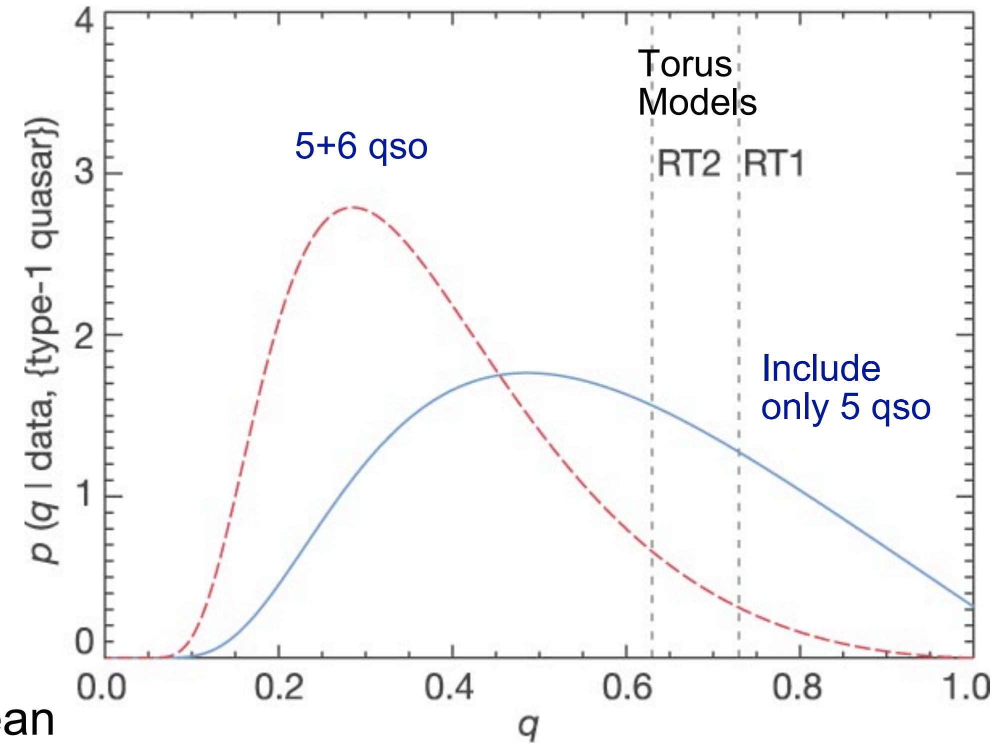
q – quasar fraction

$$q = \frac{\text{Type-1 quasars}}{\text{Type-1 + Type-2}} = \frac{N1}{N1+N2}$$

<N1> - number of Type-1 qso
<N2> - number of Type-2 qso

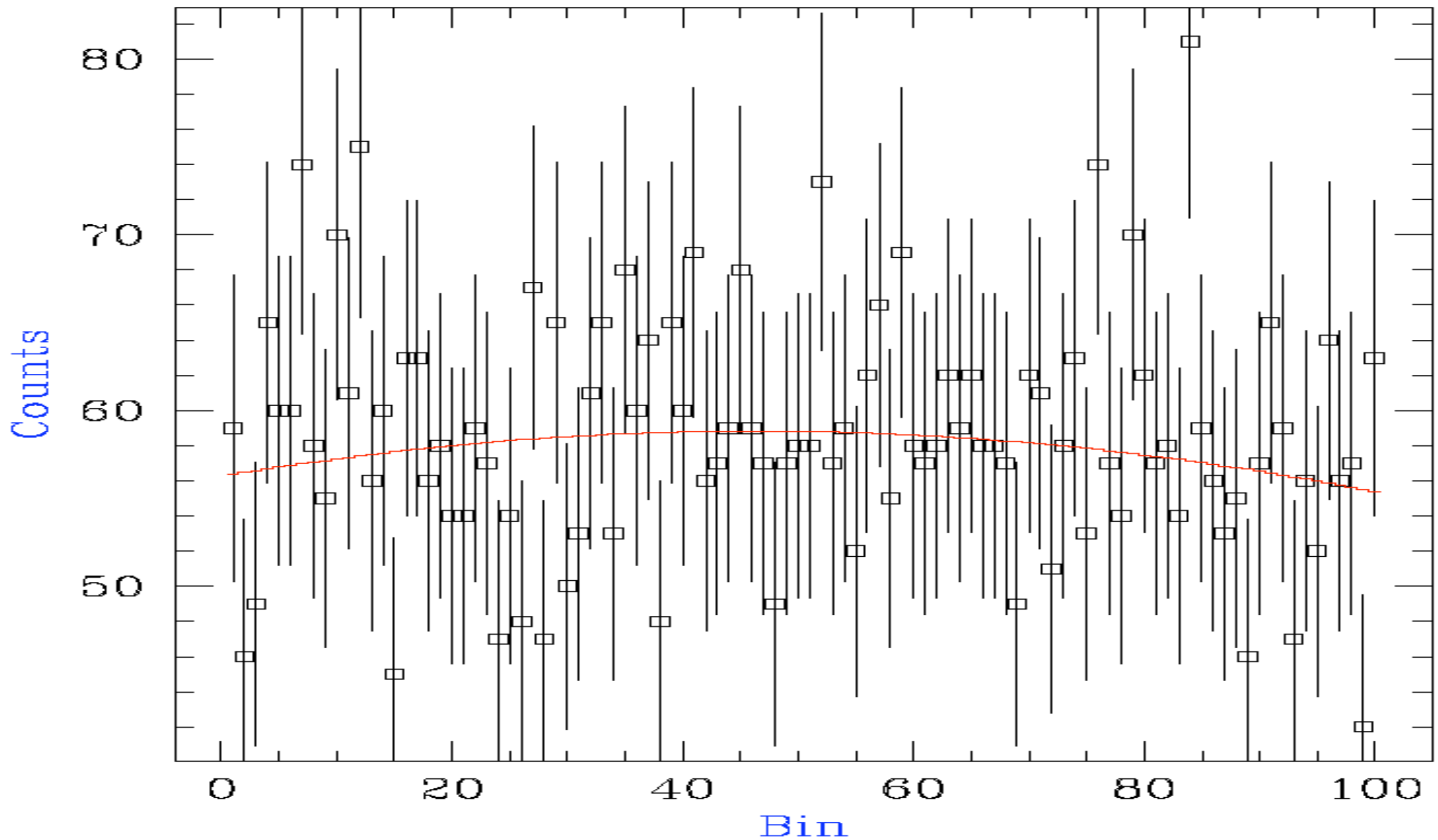
1/ take Poisson likelihood with the mean <N2> = (1-q)<N1>/q
2/ evaluate likelihood at each q and N1
3/ integrate P(N1|q)P(N1) over N1

Posterior Probability distribution for the quasar fraction



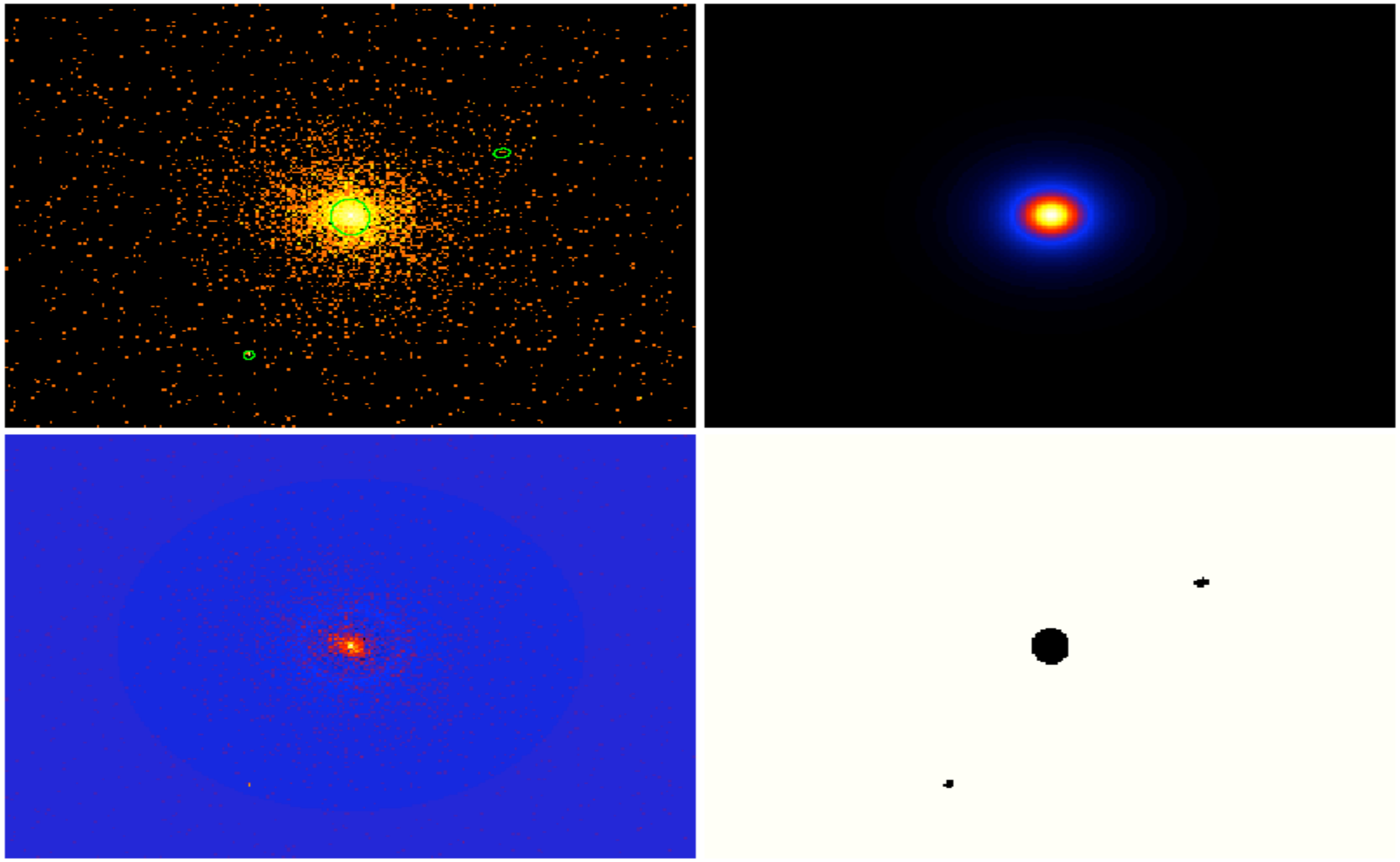
p(q|data,{type-1 qso}) = p(data|q,{type-1 qso})

Example:



Integer counts spectrum sampled from a constant amplitude model with mean μ = 60 counts, and fit with a parabolic model.

Example2



Example of a two-dimensional integer counts spectrum. *Top Left: Chandra* ACIS-S data of X-ray cluster MS 2137.3-2353, with *ds9* source regions superimposed.
Top Right: Best-fit of a two-dimensional beta model to the filtered data.
Bottom Left: Residuals (in units of σ) of the best fit.
Bottom Right: The applied filter; the data within the ovals were excluded from the fit.

Probability

Numerical formalization of our degree of belief. => $\dfrac{\text{Number of favorable events}}{\text{Total number of events}}$

Laplace principle of indifference:

All events have equal probability

Example 1:
1/6 is the probability of throwing a 6 with 1 roll of the dice **BUT** the dice can be biased!
=> need to calculate the probability of each face

Example 2:
Use data to calculate probability, thus the probability of a cloudy observing run:

$\dfrac{\text{number of cloudy nights last year}}{\text{365 days}}$

Issues:
- limited data
- not all nights are *equally likely* to be cloudy

Conditionality and Independence

A and B events are **independent** if the probability of one is unaffected by what we know about the other:

$$\text{prob}(A \text{ and } B)=\text{prob}(A)\text{prob}(B)$$

If the probability of A depends on what we know about B
A given B => **conditional probability**

$$\text{prob}(A|B)=\frac{\text{prob}(A \text{ and } B)}{\text{prob}(B)}$$

If A and B are independent => prob(A|B)=prob(A)

If there are several possibilities for event B (B_1, B_2....)

$$\text{prob}(A) = \sum \text{prob}(A|B_i)\ \text{prob}(B_i)$$

A – parameter of interest
B_i – not of interest, instrumental parameters, background
prob(B_i) - if known we can sum (or integrate) - *Marginalize*

Bayes' Theorem

Bayes' Theorem is derived by equating:

prob(A and B) = prob (B and A)

$$\text{prob}(B|A) = \frac{\text{prob}(A|B)\,\text{prob}(B)}{\text{prob}(A)}$$

Gives the Rule for induction:

the data, the event A, are succeeding B, our knowledge preceeding the experiment.

prob(B) – **prior probability** which will be modified by experience

prob(A|B) – **likelihood**

prob(B|A) **– posterior probability** – the knowledge after the data have been analyzed

prob(A) – normalization

Example

A box with colored balls:
what is the content of the box?

prob(content of the box | data) ∝ prob(data | content of the box)

Experiment:
N red balls
M white balls
N+M = 10 total, known
Draw 5 times (putting back) (T) and get 3 red balls (R)
How many red balls are in the box?

Model (our hypothesis) => $\text{prob}(R) = \dfrac{N}{N+M}$

Likelihood = $\binom{T}{R}\, \text{prob}(R)^R\, \text{prob}(M)^{T-R}$