SHERPA

CIAO's Modeling and Fitting Application

Aneta Siemiginowska

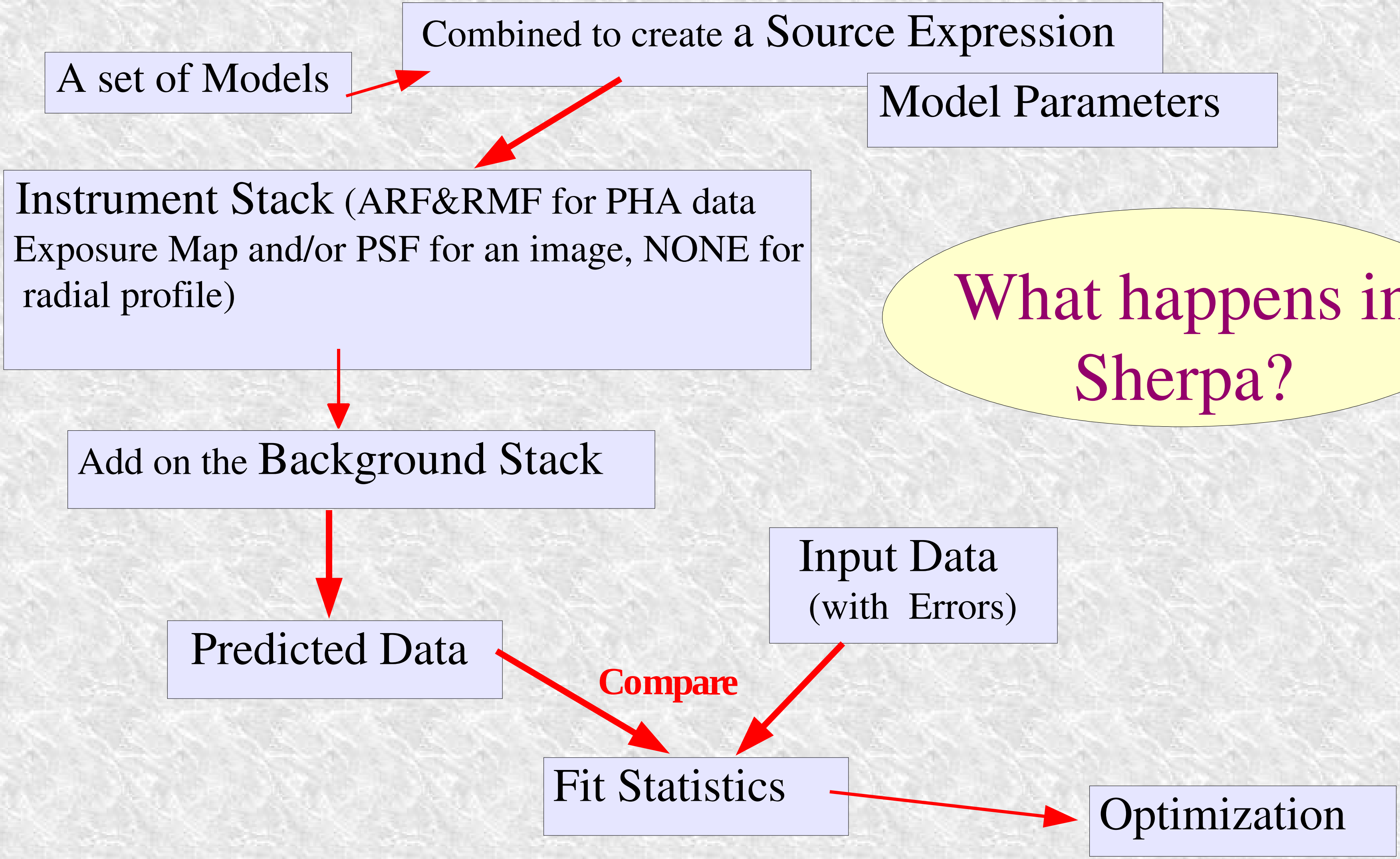
Modeling and Fitting Software

- **XSPEC** - analysis of 1D X-ray data (imaging + grating)
- **ISIS** and **Pint of Ale**- primarily for analysis of high-resolution (ie grating) X-ray data
- **Sherpa** - generalised multi-dimensional fitting package
- All programs use the technique of **forward fitting**:
  - a model is evaluated, compared to the actual data, and then the parameters are changed to improve the match. This is repeated until convergence occurs.

What can you do in Sherpa?

- Standard PHA based analysis.
- Model data in many spectral bands simultaneously, e.g., optical/ X-rays.
- Access ATOMDB and GUIDE/ISIS for grating data analysis.
- Fit radial profiles.
- Simulate 1D data.
- Model 2D image data, e.g., fit surface brightness of the extended source.
- Get normalization of your PSF, while fitting the data with 1D/2D PSF.
- Use the PSF as a convolution kernel in the 2D image analysis(FFT or sliding cell).
- Convolution using the TCD library kernel.
- Use of exposure maps in the image analysis.
- Joint-mode data: spatial-spectral, spatial-timing
- Use scripts based on Sherpa only commands.
- Use S-lang on command line and in S-lang based scripts. S-lang allows you to access directly the internal information about the data, models, statistics.
- Use your own models with User Models and S-lang user models.

What happens in Sherpa?



A set of Models

Combined to create a Source Expression

Model Parameters

Instrument Stack (ARF&RMF for PHA data Exposure Map and/or PSF for an image, NONE for radial profile)

Add on the Background Stack

Predicted Data

Input Data (with Errors)

Compare

Fit Statistics

Optimization

Main SHERPA Components

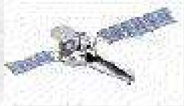
- Data Input/Output.
- Visualization through ChIPS and ds9
- Model library and model language.
- Statistics and Error Analysis.
- Optimization Methods.
- Access to the internals through S-lang.

Data Input/Output

- General use of data type and dimensionality.
- Supported types of files: ASCII, FITS binary tables and Images,PHA types I \& II, IRAF IMH and QPOE files
- Sherpa:
  - groups the data if appropriate;
  - treats integer, float or double precision data;
  - supports data of arbitrary dimensionality
- I/O interface through Data Model and Varmm
- Filtering while reading the data.
- Input data on the command line in two ways.

MODELS

- Three main type of models:
  - Source
  - Background
  - Instrument
- Model library consists of several models (plus XSPEC v.11) which can be used to define a source or background model
- There are different types of instrument models to support both 1D and 2D analysis.
- Instrument models are convolved with Source and Background models before the model predicted data is compared with the observed data.
- Instrument and Background models are NOT required. Source models have to be defined for fitting.



Fit Statistics in Sherpa:

Sherpa has a large array of statistics appropriate for analyzing Poisson-distributed (*i.e.* counts) data.

- Statistics based on χ^2 :
  - CHI GEHRELS
  - CHI DVAR
  - CHI MVAR
  - CHI PARENT
  - CHI PRIMINI
- Statistics based on the Poisson likelihood:
  - CASH
  - BAYES

If the data are not Poisson-distributed (*i.e.* fluxes), then alternatives include:

● least-squares fitting: setting all variances to one

● providing errors in an input file.

Optimization in Sherpa

Optimization => minimizing the statistics (^{2} or log $\mathcal{L}$) by varying the thawed parameters of the model.

- **Find a local minimum:**

  LEVENBERG-MARQUARDT
  POWELL
  SIMPLEX

Fast, but not appropriate for finding the global minimum of a complex statistical space when starting from a random point

- **Attempt to find the global minimum:**

  GRID
  GRID-POWELL
  MONTECARLO
  MONTE-LM
  MONTE-POWELL
  SMULATED ANNEALING

Computationaly intensive algorithms designed to search comlicated statistical surfaces.

- **Optimize/Reject/Filter:**

  SIGMA-REJECTION outliers are filtered from the data.

Confidence Intervals

- Vary a parameter's value, while holding the values of all the parameters to their best-fit values, until the fit statistic increases by some preset amount from its minimum value (^{2} = 1 for 1).
  - Uncertainty
  - Projection
- Calculate **Covariance** matrix:

  1 confidence intervals are given by $\sqrt{C_{i,i}}$

  where $C_{j,i} = I^{-1}_{i,j}$

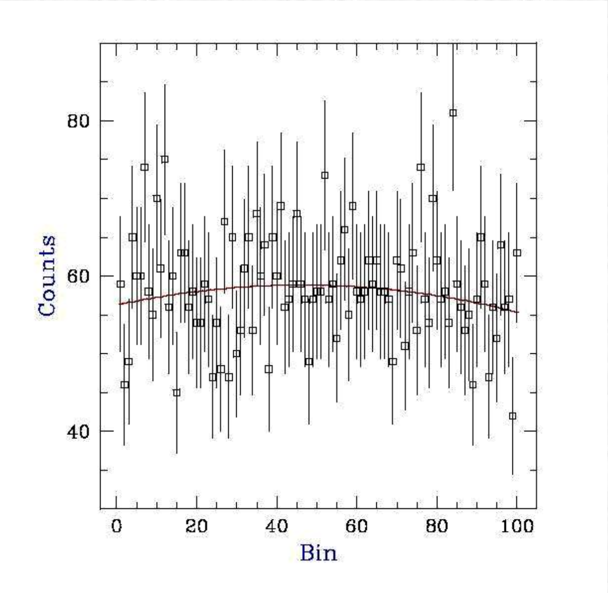
  and $I_{i,j}$ - the information matrix computed at the best-fit point:

$$I_{i,j} = \frac{\partial^2 \quad ^2}{\partial p_i \partial p_j}$$

or any other statistics

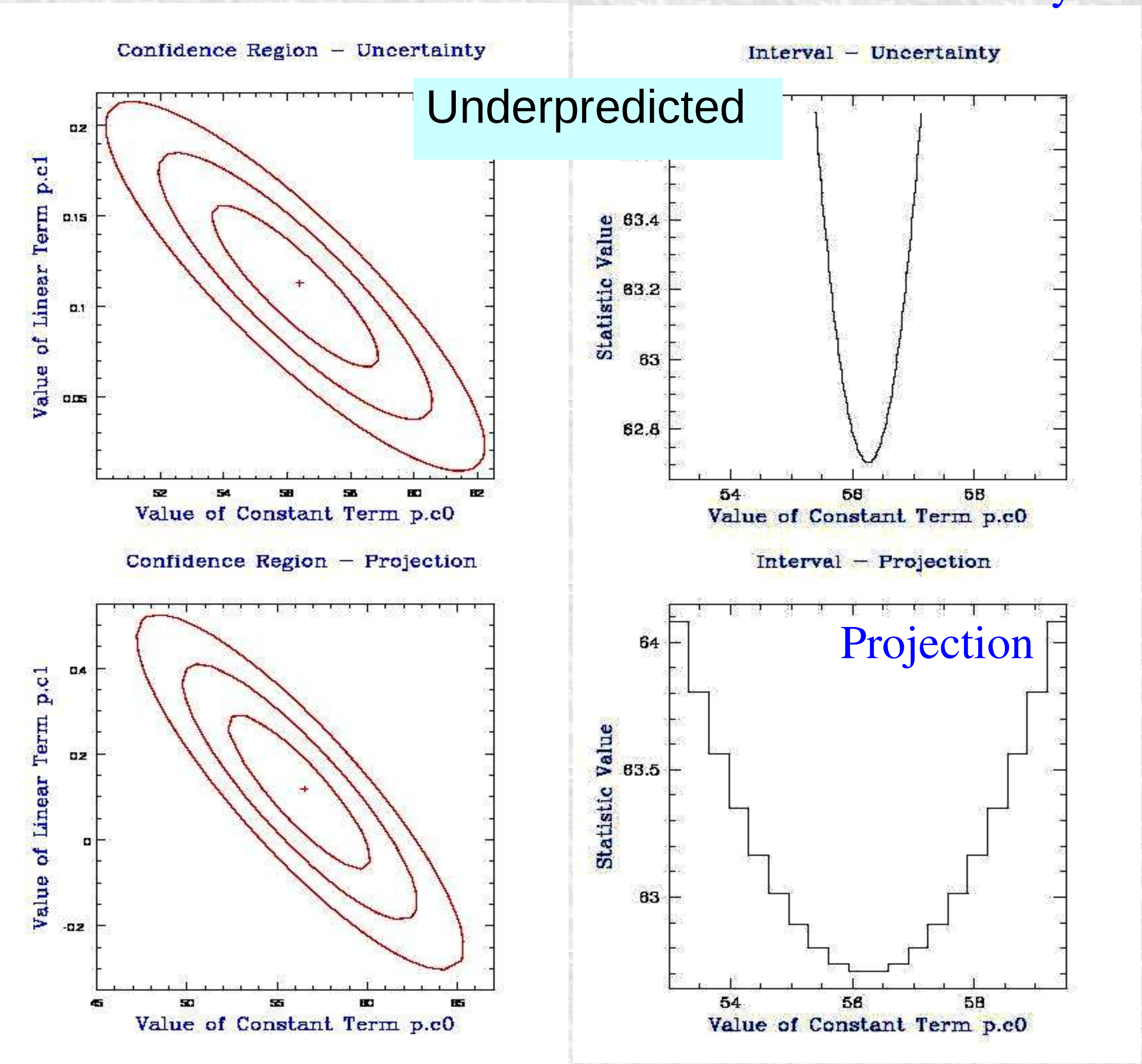
Visualize Confidence Levels

Data and the Best Fit Model



Well behave parameter space!

Uncertainty



Underpredicted

Projection

Customize Sherpa

- Sherpa State Object (e.g. Configuration file) – S-lang variable initialized at the start of the Sherpa session:

```
sherpa> print(sherpa)
plot             =  sherpa_Plot_State
dataplot          =  sherpa_Plot_State
fitplot          =  sherpa_FitPlot_State
resplot          =  sherpa_Plot_State
multiplot         =  sherpa_Draw_State
output           =  sherpa_Output_State
regproj          =  sherpa_VisParEst_State
regunc           =  sherpa_VisParEst_State
intproj          =  sherpa_VisParEst_State
intunc           =  sherpa_VisParEst_State
proj             =  sherpa_Proj_State
cov              =  sherpa_Cov_State
unc              =  sherpa_Unc_State
con_levs           =  NULL
modeloverride      =  0
multiback          =  0
deleteframes        =  1
clobber            =  0
```

Customize Plotting

Customize Confidence Levels

```
sherpa> print(sherpa.regproj)
fast             =  1
expfac           =  3
arange          =  1
min              =  Double_Type[2]
max              =  Double_Type[2]
log              =  Integer_Type[2]
nloop            =  Integer_Type[2]
sigma            =  Double_Type[3]
```

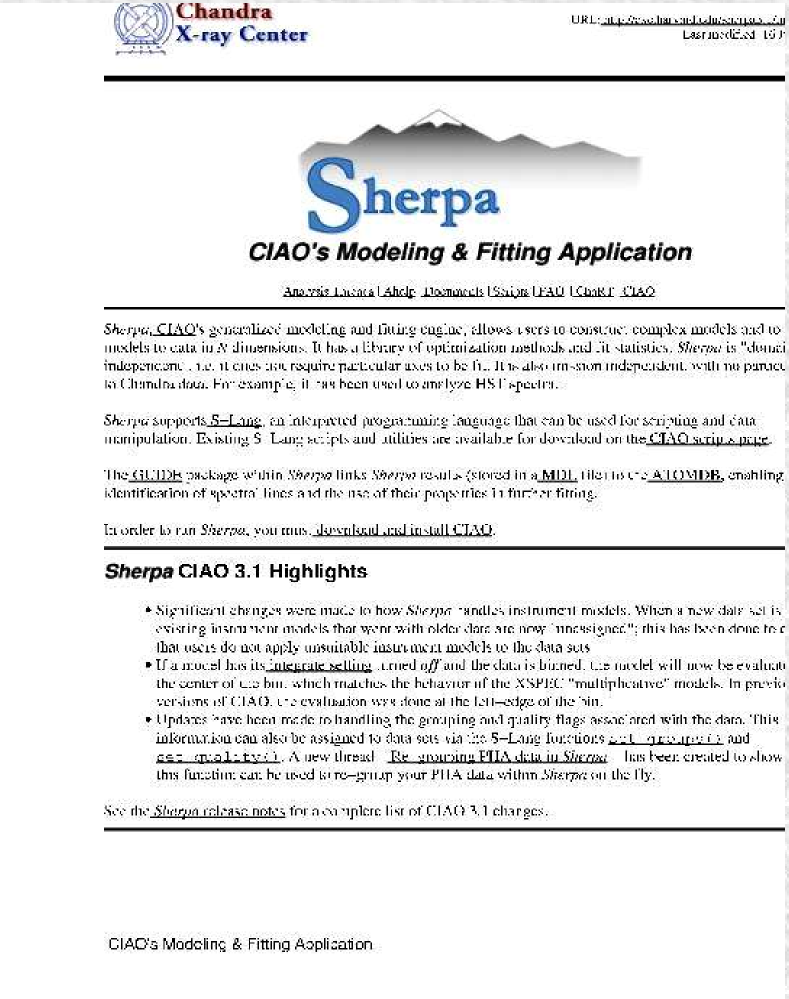
Customize Sherpa

- Sherpa Resource File:
  - a text file with Sherpa/Chips/S-lang commands
- Access:
  - Environment variable SHERPARC
  - File .sherparc in current directory $PWD
  - File .sherparc in HOME directory $HOME
- Example:

```
unix% more .sherparc
# Example Sherpa resource file
message("Starting to process sherparc")
paramprompt off
method simplex
define q () { () = sherpa_eval("quit"); }
message("Finished processing .sherparc")
```

Learn More on Sherpa Web Page

http://cxc.harvard.edu/sherpa/



Chandra X-ray Center

URL: http://cxc.harvard.edu/sherpa/in
Last modified: 16 J

Sherpa

CIAO's Modeling & Fitting Application

Analysis Threads | Ahelp | Documents | Scripts | FAQ | ChaRT | CIAO

Sherpa, CIAO's generalized modeling and fitting engine, allows users to construct complex models and to models to data in *N* dimensions. It has a library of optimization methods and fit statistics. *Sherpa* is "domai independent", i.e. it does not require particular axes to be fit. It is also mission independent, with no particu to Chandra data. For example, it has been used to analyze HST spectra.

Sherpa supports S–Lang, an interpreted programming language that can be used for scripting and data manipulation. Existing S–Lang scripts and utilities are available for download on the CIAO scripts page.

The GUIDE package within *Sherpa* links *Sherpa* results (stored in a MDL file) to the ATOMDB, enabling identification of spectral lines and the use of their properties in further fitting.

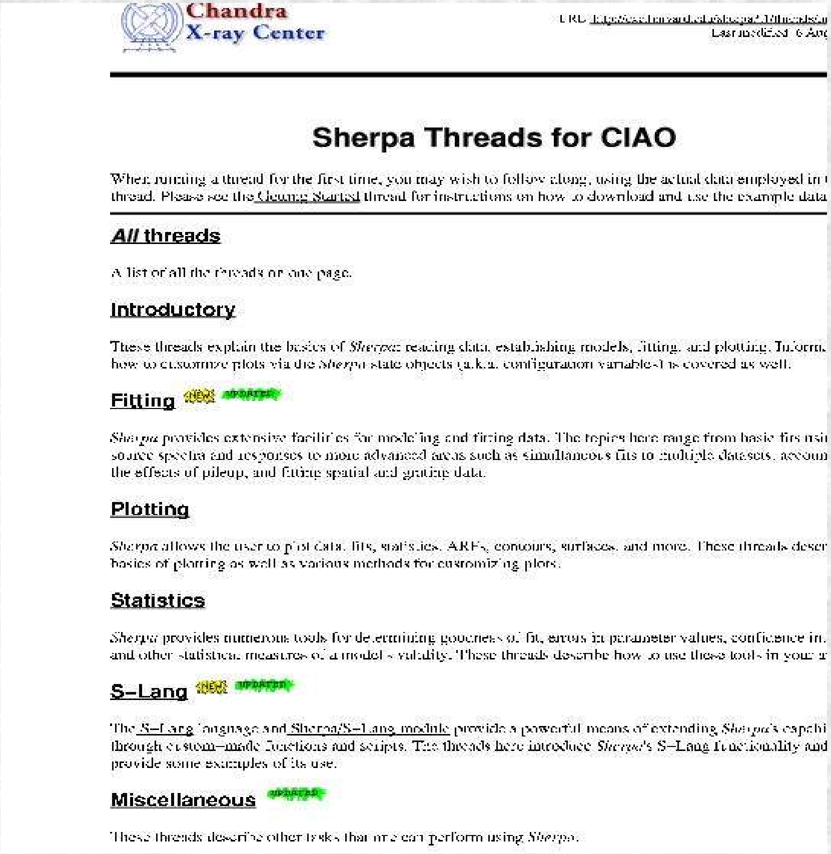
In order to run *Sherpa*, you must download and install CIAO.

Sherpa CIAO 3.1 Highlights

- Significant changes were made to how *Sherpa* handles instrument models. When a new data set is existing instrument models that went with older data are now "unassigned"; this has been done to e that users do not apply unsuitable instrument models to the data sets
- If a model has its integrate setting turned *off* and the data is binned, the model will now be evaluat the center of the bin, which matches the behavior of the XSPEC "multiplicative" models. In previo versions of CIAO, the evaluation was done at the left–edge of the bin.
- Updates have been made to handling the grouping and quality flags associated with the data. This information can also be assigned to data sets via the S–Lang functions set_grouping() and set_quality(). A new thread "Re–grouping PHA data in *Sherpa*" has been created to show this function can be used to re–group your PHA data within *Sherpa* on the fly.

See the *Sherpa* release notes for a complete list of CIAO 3.1 changes.

CIAO's Modeling & Fitting Application



Chandra X-ray Center

URL: http://cxc.harvard.edu/sherpa/threads/in
Last modified: 6 Aug

Sherpa Threads for CIAO

When running a thread for the first time, you may wish to follow along, using the actual data employed in t thread. Please see the Getting Started thread for instructions on how to download and use the example data

All threads

A list of all the threads on one page.

Introductory

These threads explain the basics of *Sherpa*: reading data, establishing models, fitting, and plotting. Informa how to customize plots via the *Sherpa* state objects (a.k.a. configuration variables) is covered as well.

Fitting

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

Plotting

Sherpa allows the user to plot data, fits, statistics, ARFs, contours, surfaces, and more. These threads descr basics of plotting as well as various methods for customizing plots.

Statistics

Sherpa provides numerous tools for determining goodness of fit, errors in parameter values, confidence in and other statistical measures of a model's validity. These threads describe how to use these tools in your a

S–Lang

The S–Lang language and Sherpa/S–Lang module provide a powerful means of extending *Sherpa*'s capabi through custom–made functions and scripts. The threads here introduce *Sherpa*'s S–Lang functionality and provide some examples of its use.

Miscellaneous

These threads describe other tasks that one can perform using *Sherpa*.