X-ray Software Packages

Michael A. Nowak (MIT-Kavli Institute)
- with useful advice over the years from-
John Davis, John Houck, Dave Huenemoerder, Jörn Wilms
Outline

• Overview of Software Packages & Their Purposes
• Flexible Image Transport System (FITS)
  • Data Visualization: fv, prism, DS9
  • Reading & Writing FITS files
• Spacecraft Specific Software
  • CALDB
• Data Analysis Packages
• Your Responsibilities
  • How to Install Software, Send Bug Reports, and Write Software
Purpose of Software Packages

- Read, Write, Visualize Astrophysical Data - Primarily from Binary Files
- Standard Format - Flexible Image Transport System
- Other Formats: HDF5, ASCII, ... (but not universal)
- Data Reduction - Temporal/Spatial/Energy/Grade Filters, and Spacecraft Specific Procedures
- Create Images, Spectra, Lightcurves, Backgrounds, Responses, ...
- Data Analysis - Apply Models to Reduced Products
Spacecraft Specific or Not?

- Analysis usually begins “agnostic” - reading/writing, visualizing
- Proceeds through a spacecraft specific stage, especially for the creation of response files (and often backgrounds)
- Spacecraft determines which software package
- Analysis often then becomes agnostic again - spectra, lightcurves, and images from many different missions can be handled in similar manners.
3 Systems, +1 Independent

- High Energy Astrophysics Software: HEASOFT (Current Version 6.11)
- CFITSIO, FV, FTOOLS, XSELECT, XSPEC, XRONOS
- Chandra Interactive Analysis of Observations: CIAO (Current Version 4.3)
- Data Model (DM), Crates, Prism, ChIPs, Sherpa
- Scientific Analysis System: SAS (Current Version 11.0.0)
- DS9 (Current Version 6.2)

Scripts Tie Pieces Together
3 Systems, +1 Independent

- High Energy Astrophysics Software: HEASOFT (Current Version 6.11)
- CFITSIO, FV, FTOOLS, XSELECT, XSPEC, XRONOS
- Chandra Interactive Analysis of Observations: CIAO (Current Version 4.3)
  - Data Model (DM), Crates, Prism, ChIPS, Sherpa
- Scientific Analysis System: SAS (Current Version 11.0.0)
- DS9 (Current Version 6.2)

Scripts Tie Pieces Together
3 Systems, +1 Independent

- High Energy Astrophysics Software: HEASOFT (Current Version 6.11)
- CFITSIO, FV, FTOOLS, XSELECT, XSPEC, XRONOS
- Chandra Interactive Analysis of Observations: CIAO (Current Version 4.3)
- Data Model (DM), Crates, Prism, ChIPS, Sherpa
- Scientific Analysis System: SAS (Current Version 11.0.0)
- DS9 (Current Version 6.2)

Scripts Tie Pieces Together
Plus Additional High Level Analysis Systems

- Interactive Spectral Interpretation System (ISIS- Current Version 1.6.1-43)
- Spectral X-ray and UV modeling, analysis, and fitting (SPEX- Current Version 2.02.04)
- Interactive Data Language (IDL) + Packages
  - Spitzer Analysis System
  - PINTofALE (Package for Interactive Analysis of Line Emission)
Data Storage, Manipulation, Visualization

- CFITSIO/FTOOLS
- CIAO/DMTOOLS
- DS9
- fv
- prism

Spacecraft Specific Software

- Suzaku
- Swift
- RXTE
- Chandra
- SAS
- XMM
- XSELECT
- SCRIPTS
- fv
- prism

“High Level” Data Analysis

- XSPEC
- Sherpa
- ISIS
- SPEX
- XRONOS
- IDL
- Spectra
- Timing
- Custom Analysis
Spectra
Timing
Custom Analysis
Data Storage, Manipulation, Visualization

CFITSIO/FTOOLS
prism
fv

CIAO/DMTOOLS

DS9

XSELECT

Suzaku
Swift
RXTE...

Spacecraft Specific Software

SAS

XMM

XSPEC
Sherpa
ISIS
SPEX
XRONOS
IDL

“High Level” Data Analysis

Spectra
Timing
Custom Analysis

SCRIPTS

fv

prism
Data Storage, Manipulation, Visualization

- CFITSIO/FTOOLS
- CIAO/DMTOOLS
- DS9
- fv
- prism

Spacecraft Specific Software

- Suzaku
- Swift
- RXTE
- Chandra
- SAS
- XMM

“High Level” Data Analysis

- XSPEC
- Sherpa
- ISIS
- SPEX
- XRONOS
- IDL
- Spectra
- Timing
- Custom Analysis

SCRIPTS
Spectra
Timing
Custom Analysis

Data Storage, Manipulation, Visualization

CFITSIO/FTOOLS
CIAO/DMTOOLS
DS9
fv
prism

Spacecraft Specific Software

Suzaku
Swift
RXTE...

Chandra

SAS

XMM

“High Level” Data Analysis

XSPEC
Sherpa
ISIS
SPEX
XRONOS
IDL

Spectra
Timing
Custom Analysis
Custom Analysis

Data Storage, Manipulation, Visualization
- CFITSIO/FTOOLS
- CIAO/DMTOOLs
- DS9
- fv
- prism

Spacecraft Specific Software
- Suzaku
- Swift
- RXTE
- Chandra
- SAS
- XMM

“High Level” Data Analysis
- XSPEC
- Sherpa
- ISIS
- SPEX
- XRONOS
- IDL

 Scripts
- XSELECT
- fv
- prism
- XMM

Custom Analysis
Custom Analysis

Data Storage, Manipulation, Visualization

CFITSIO/FTOOLS

CIAO/DMTOOLS

DS9

"High Level" Data Analysis

Python

Python

S-lang

XSPEC

Sherpa

ISIS

Modules

Custom Analysis
FITS Format

(http://heasarc.nasa.gov/docs/heasarc/fits.html)

- Binary File Format that is “Self Documenting”
- Named Extensions, containing Headers and Data
- Extensions can be referred to by Name or Number
- Headers Describe Contents, Format, Processing History
- Keywords contain descriptive data
- Data are rows & columns containing values, arrays, images ...
FITS Format

(http://heasarc.nasa.gov/docs/heasarc/fits.html)

- Standard types of astrophysical files, e.g., spectra or Ancillary Response Files (ARF) have specific format requirements

- OGIP (Office of Guest Investigator Programs)
  http://heasarc.nasa.gov/docs/heasarc/ofwg/ofwg_intro.html

- Analysis Packages can be More or Less Tolerant of lack of “OGIP Compliance”

- Most missions strive to be consistent, others openly scoff (I’m looking at you INTEGRAL …)
unix%> fv acisf11044N002_evt2.fits.gz &
unix%> fv acisf11044N002_evt2.fits.gz &
unix%> fv acisf11044N002_evt2.fits.gz &
unix%> fv acisf11044N002_evt2.fits.gz &
unix%> fv acisf11044N002_evt2.fits.gz &
unix%> fv acisf11044N002_evt2.fits.gz &

<table>
<thead>
<tr>
<th></th>
<th>time</th>
<th>expno</th>
<th>tg_r</th>
<th>tg_d</th>
<th>chipx</th>
<th>chipy</th>
<th>tdx</th>
<th>tdy</th>
<th>detx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-3.691336E-01</td>
<td>-2.468872E-04</td>
<td>31</td>
<td>90</td>
<td>1864</td>
<td>1792</td>
<td>1.851630E+03</td>
</tr>
<tr>
<td>2</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-3.317697E-01</td>
<td>-4.458029E-04</td>
<td>265</td>
<td>108</td>
<td>2098</td>
<td>1810</td>
<td>2.084881E+03</td>
</tr>
<tr>
<td>3</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-3.069667E-01</td>
<td>6.948889E-05</td>
<td>420</td>
<td>124</td>
<td>2253</td>
<td>1826</td>
<td>2.39277E+03</td>
</tr>
<tr>
<td>4</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.263381E-01</td>
<td>-3.327907E-03</td>
<td>927</td>
<td>138</td>
<td>2760</td>
<td>1840</td>
<td>2.745742E+03</td>
</tr>
<tr>
<td>5</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.482646E-01</td>
<td>9.906186E-04</td>
<td>787</td>
<td>160</td>
<td>2620</td>
<td>1862</td>
<td>2.606196E+03</td>
</tr>
<tr>
<td>6</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>NULL</td>
<td>NULL</td>
<td>367</td>
<td>181</td>
<td>2200</td>
<td>1883</td>
<td>2.186948E+03</td>
</tr>
<tr>
<td>7</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.278911E-01</td>
<td>-4.480751E-04</td>
<td>916</td>
<td>410</td>
<td>2749</td>
<td>2112</td>
<td>2.734578E+03</td>
</tr>
<tr>
<td>8</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.217081E-01</td>
<td>1.207414E-04</td>
<td>955</td>
<td>410</td>
<td>2788</td>
<td>2112</td>
<td>2.773782E+03</td>
</tr>
<tr>
<td>9</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.268105E-01</td>
<td>-2.887621E-05</td>
<td>923</td>
<td>412</td>
<td>2756</td>
<td>2114</td>
<td>2.741561E+03</td>
</tr>
<tr>
<td>10</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.293616E-01</td>
<td>-1.036949E-04</td>
<td>907</td>
<td>413</td>
<td>2740</td>
<td>2115</td>
<td>2.725656E+03</td>
</tr>
<tr>
<td>11</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.287421E-01</td>
<td>1.131784E-04</td>
<td>911</td>
<td>414</td>
<td>2744</td>
<td>2116</td>
<td>2.729746E+03</td>
</tr>
<tr>
<td>12</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.375345E-01</td>
<td>1.037267E-04</td>
<td>856</td>
<td>419</td>
<td>2698</td>
<td>2121</td>
<td>2.679404E+03</td>
</tr>
<tr>
<td>13</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.451881E-01</td>
<td>-1.207220E-04</td>
<td>808</td>
<td>422</td>
<td>2641</td>
<td>2124</td>
<td>2.626757E+03</td>
</tr>
<tr>
<td>14</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.528708E-01</td>
<td>-2.809214E-05</td>
<td>760</td>
<td>427</td>
<td>2593</td>
<td>2129</td>
<td>2.578562E+03</td>
</tr>
<tr>
<td>15</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.567194E-01</td>
<td>9.748785E-05</td>
<td>736</td>
<td>430</td>
<td>2569</td>
<td>2132</td>
<td>2.555083E+03</td>
</tr>
<tr>
<td>16</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>NULL</td>
<td>NULL</td>
<td>189</td>
<td>434</td>
<td>2022</td>
<td>2136</td>
<td>2.008408E+03</td>
</tr>
<tr>
<td>17</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.735118E-01</td>
<td>1.515010E-04</td>
<td>631</td>
<td>440</td>
<td>2464</td>
<td>2142</td>
<td>2.449811E+03</td>
</tr>
<tr>
<td>18</td>
<td>3.798256101785E+08</td>
<td>239</td>
<td>-2.789313E-01</td>
<td>-2.731359E-05</td>
<td>597</td>
<td>442</td>
<td>2430</td>
<td>2144</td>
<td>2.416175E+03</td>
</tr>
</tbody>
</table>
unix%> fv acisf11044N002_pha2.fits.gz &
unix%> fv acisf11044N002_pha2.fits.gz &
unix%> prism acisf11044N002Evt2.fits.gz &
unix%> ds9 acisf11044N002_evt2.fits.gz &
unix%> ds9 acisf11044N002_evt2.fits.gz &
unix%> ds9 acisf11044N002_evt2.fits.gz &
unix%> ds9 acisf11044N002_evt2.fits.gz &
unix%> ds9 acisf11044N002_evt2.fits.gz &
READING & WRITING: HEASOFT—FITSIO & FTOOLS

(heasarc.gsfc.nasa.gov/lheasoft/)

- CFITSIO library: Suitable for compiled programs (Fortran, C, C++)
- Interfaces exist for scripting languages (S-lang, Python, Ruby, TCL, Perl, MATLAB, IDL)
- Overlapping functionality in script packages (IDL, Python via PyFITS)
- FTOOLS functions: Command line interfaces
- Suitable for use in Shell/Perl Scripts
HEASOFT – FTOOLS
(http://heasarc.gsfc.nasa.gov/docs/software/ftools)

- Tools have a variety of parameters (plist), that can be set before running (pset), or on the command line

- Command line interface is less “hidden”; therefore, less prone to error, cleaner for scripts

- Be careful to update default parameters every new HEASOFT release! (punlearn)

- Wide variety of tools - fdump (list contents), fcopy (copy contents), fextract (extract an extension), fstatistic (statistics of columns), ftcopy (copy with filters), fltime (filter file by time), ...
READING & WRITING: CIAO–CRATES & DM


• CRATES – A high level Python interface for Input/Output of data files
  • Used in ChIPS and Sherpa
  • Can be used with other Python packages
• Data Model functions: Command line interfaces
  • Suitable for use in Shell/Perl Scripts
CIAO – Data Model
(http://cxc.harvard.edu/ciao/ahelp/dm.html)

- Tools have a variety of parameters (plist), that can be set before running (pset), or on the command line
- Command line interface is less “hidden”; therefore, less prone to error, cleaner for scripts
- Be careful to update default parameters every new CIAO release! (punlearn)
- Wide variety of tools - dmlist (list contents), dmcopy (copy contents), dmextract (extract data), dmstat (statistics of columns), ...
Parameter Files

- Usually found in your home directory:
  - HEASOFT – /home/me/pfiles/*
  - CIAO – /home/me/cxcds_param4/*

- *Not All Parameters are Prompted for by Tools!*

- Unprompted parameters take on defaults; *often* are parameters that needn’t/shouldn’t be changed

- Check parameter files directly; read FITS history

- When in doubt, set all parameters explicitly

- For scripts, set PFILES environment to a dedicated directory (avoids jobs clobbering each other)
Spacecraft Specific Reduction

- Three Primary Choices for X-ray Astronomy:
  - CIAO – Chandra Observations
    - Includes (& Usually Requires) DS9
  - SAS – XMM-Newton Observations
    - Requires HEASOFT, DS9, & GRACE
  - HEASOFT – (Almost) Everything Else
    - RXTE, Suzaku, Swift, (some of) INTEGRAL (->OSA)
    - ASCA, CGRO, EXOSAT, ROSAT, Einstein, HEAO-1
Data Analysis Steps I.

- “Raw” Telemetry is converted to a FITS data file
- “Level 0” -> “Level 1”
  - Processing Pipeline step, usually not accessible/reproducible by an ordinary user
- “Level 1” files can still have many detector effects included in the file
  - False Events: flaring pixels, time tags, etc.
  - “Bad Grades”: Cosmic Ray hits
  - Uncorrected positions, times, gain (energy), ...
Data Analysis Steps II.

- Further Cleaning/Filtering Provided by Pipelines
- “Level 1” -> “Level 2”
- “Standard” Choices are Applied
- (Most) False Events Removed & Corrections Made
- You may have other choices and/or calibration may have improved, requiring reprocessing
- You likely will want to subdivide the data, e.g., remove times of high background, or select specific Space/Time/Energy cuts
Data Analysis Steps IIa.

- You Do Want to Change Spacecraft Specific Filters & Corrections, e.g., CTI Correction, Bad Pixels, etc.
- Spacecraft Specific Software needs to be used
  - `acis_process_events` -> Chandra data
  - `xispi` -> Suzaku data, ... etc.
- Software will properly update FITS headers to account for changes in integration Time or Area
Data Analysis Steps IIb.

- Applying “Generic” Time/Space/Energy Filters
- In Principle, Multiple Software Packages Work
  - CIAO: Data Model with filters
  - HEASOFT: Xselect (extractor run under the hood)
  - SAS: evselect or xmmselect (XMM only!)
- Software Updates Exposures, Areas, etc., in Headers
- Warning: Sometimes information needed later is only properly passed along by “principal” system, e.g., Suzaku “works best” under HEASOFT
Data Analysis Steps III.

- Extraction of “High Level” Data Products
  - Spectra
  - Lightcurves
  - Images – DS9 Often Used to Define Regions
- In Principle, Multiple Software Packages Work
  - CIAO: Data Model with filters
  - HEASOFT: Xselect (extractor run under the hood)
  - SAS: evselect or xmmselect (XMM only!)
XSELECT

- Attempts to Combine IIa, IIb, and III All in One Package
- Not Really a GUI, Not Really a Command Line System, Not Quite a Programmable Environment
- Runs extractor “under the hood”, but hides (obfuscates?) the details
- Can be worked into Shell Scripts, but given that XSELECT expects user interaction it is easier to incorporate into Perl/Python scripts
Data Analysis Steps IV.

• Create Associated Files to Aid Analysis

• Responses (RMF & ARF), Exposure Maps, Background Files (& their Responses [maybe])

• Spacecraft Specific Software Required!

• Follow the ABC Guides!

• Try to understand *why* a step is being done; don’t try to automate until you understand the pitfalls!
Helpful Starting Points

- Chandra Analysis: http://asc.harvard.edu/ciao/guides
- XMM Analysis: http://xmm.esa.int/sas
- Suzaku Analysis: http://heasarc.nasa.gov/docs/suzaku/analysis/abc
- Swift Analysis: http://heasarc.nasa.gov/docs/swift/analysis/start
• Calibration Database Required for Analyses

• Check Spacecraft Sites periodically for Updates & Caveats. Learn the files to apply in each situation

• Use Consistent Software & Calibration Database

• New Software with Old Calibration Files, or visa versa, sometimes can produce wrong results

• This is where having the correct Parameter Files can be crucial! (New Software+New CALDB+Old Parameters = Wrong Results!)
Analysis

- You’ve got your Spectra, Images, and Lightcurves, now what?
- Analyze with your favorite software package!
- Products are back to “Standard Forms” so packages work well on products from a wide range of spacecraft
- Timing – XRONOS is specifically designed for timing
  - I have never used it
  - I used IDL (<= 2001), & now ISIS (>2001)
  - Lots of custom code out there ...
Analysis

- Imaging Analysis –
  - Sherpa will convolve Chandra PSF with simple 2D models & fit a 2D histogram
  - Some fraction of image analysis reduces 2D -> 1D, and then any spectral package applies
  - My limited experience is simple 2D models applied to 2D histograms using ISIS
  - You’ll see examples from people far more experienced than me in this arena...
• Spectral Analysis –
  • Lots of choices, with the four major choices being XSPEC, Sherpa, ISIS, SPEX
  • XSPEC is the oldest & most established, so many models will be written with XSPEC in mind
  • Sherpa is ≈ the youngest; lectures on advanced use on Saturday
  • SPEX is in some ways a “specialty” package for high-resolution X-ray spectroscopy
  • I use ISIS for most of my mathematical analysis (spectra, timing, & other things)
## Comparison of Packages:

<table>
<thead>
<tr>
<th></th>
<th>XSPEC Models</th>
<th>XSPEC Local Models</th>
<th>Scripted Models</th>
<th>User Scripts</th>
<th>Data Product Access</th>
<th>Other Fit Kernels</th>
<th>User Fit Kernels</th>
<th>User Optim. Methods</th>
<th>User Fit Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XSPEC</strong></td>
<td>All</td>
<td>Yes</td>
<td>Limited</td>
<td>TCL, Python</td>
<td>Limited</td>
<td>Gain</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Sherpa</strong></td>
<td>Most</td>
<td>With Effort</td>
<td>Python</td>
<td>Python</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ISIS</strong></td>
<td>All</td>
<td>Yes</td>
<td>S-lang</td>
<td>S-lang</td>
<td>Yes</td>
<td>Gain, Pileup</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>SPEX</strong></td>
<td>Few</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Non-X-ray Data</th>
<th>Atomic Data Access</th>
<th>Model Caching</th>
<th>Multi-core Errors</th>
<th>Multi-core Fits</th>
<th>Multi-System Errors</th>
<th>Multi-System Models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XSPEC</strong></td>
<td>With Fake RMF, ARF</td>
<td>No</td>
<td>Limited</td>
<td>No*</td>
<td>No*</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Sherpa</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>ISIS</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>SPEX</strong></td>
<td>No</td>
<td>Yes</td>
<td>?</td>
<td>No*</td>
<td>No*</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

*Some internal code is, or can be, parallelized upon compilation*
Your Responsibilities:

- You will use & install a lot of software over your astrophysical career.
- You will probably write a lot of scripts, and maybe even spectral fit models, or even a software package.
- Undoubtedly, sometime, somewhere you will run into some problems.
- There are things you can do to help yourself, and things you can do to help us help you.
Installing Software:

- Don’t have anyone else to install software for you. *Know what’s going where on your machine!*

- Use a consistent set of compilers and libraries. E.g., all 32-bit built with gcc/gfortran/g++ 4.4.1. *Know what compilers and libraries you are using!*

  - `unix%> gcc --version`

  - **Builds:** `setenv CC ; setenv CXX ; setenv FC`

- **Linux:** package management systems, e.g., `apt-get` can be helpful in creating a consistent system

- **Apple:** it’s the Wild West, between Xcode, Fink, MacPorts, & Downloaded Packages

  - I use Fink/MacPorts sparingly, and hand-install compilers (sourceforge.net) and many libraries
Installing Software:

- Install software in sane, “standard” locations, for example, /usr/local or /opt

- Don’t default paths to access software automatically – use start up scripts when you need the program

- Our programs try to play nice with one another, but some mistakes happen ...

- It’s easy to end up with multiple copies of CFITSIO, PGPLOT, ... Using one program at a time decreases the chance of inconsistent libraries
Writing Bug Reports:

- If we’ve made a mistake, let us know! But give us the info we need to figure it out:
  - Clear, informative subject line on e-mail
  - Tell us what version of the software, caldb, compilers, and operating system you are using
  - Send us copies of any log files and the complete text of any error messages
  - Send us figures, with clear labels and detailed descriptions, that illustrate the issues. *Figure names* should make it obvious what you mean!
  - If possible, write us a script that reproduces the problem. Point us to a place where we can download the script and data that you used
If You are going to use existing code (e.g., Numerical Recipes), either:

- Compile it in via an existing library, or,
- Change the subroutine name in your code! (Fortran, especially, will happily use the first instance of a name it finds)

- Avoid generic names, e.g., directories called “data”. “agn_torus_data” would be a better name.

- Be descriptive in variable, subroutine, data, directory, and code names. And ...

- Comments, Comments, Comments ...
Writing Software:

- Put comments and contact info and dates in your code. Use version numbers! E.g., ISIS 1.6.1-35
- -35: minor changes (mostly bug fixes), .1: new functionality, but backward compatibility preserved, .6: significant changes, backward compatibility not guaranteed, 1: major changes!
- If you change *anything*, change the version number! I mean *anything, change the number!*
- Version control is your friend, for software, scripts, papers... I highly recommend git
Writing Software:

- The basics of git are straightforward:
  - `unix%> git init`
  - `unix%> git add .isisrc_plots`
  - `unix%> git commit -a`

- git lets you track changes:
  - `unix%> git show`
  - `unix%> git log`

- git lets you go back to earlier versions:
  - `unix%> git checkout <commit-hash>`
  - `unix%> git revert <commit-hash>`

- Your life will be much happier! For more, see tutorials at: [http://git-scm.com/documentation](http://git-scm.com/documentation)