

Miscellaneous AFNI Utilities

- Reminder: all command line utilities have a [-help](#) option that provides a reminder about their usage; for most programs, the output of [-help](#) is the most up-to-date documentation
- [3dinfo](#) — print out information from a dataset .HEAD file

◇ [3dinfo astrip+orig](#) OR Define Datamode→Misc→Anat Info

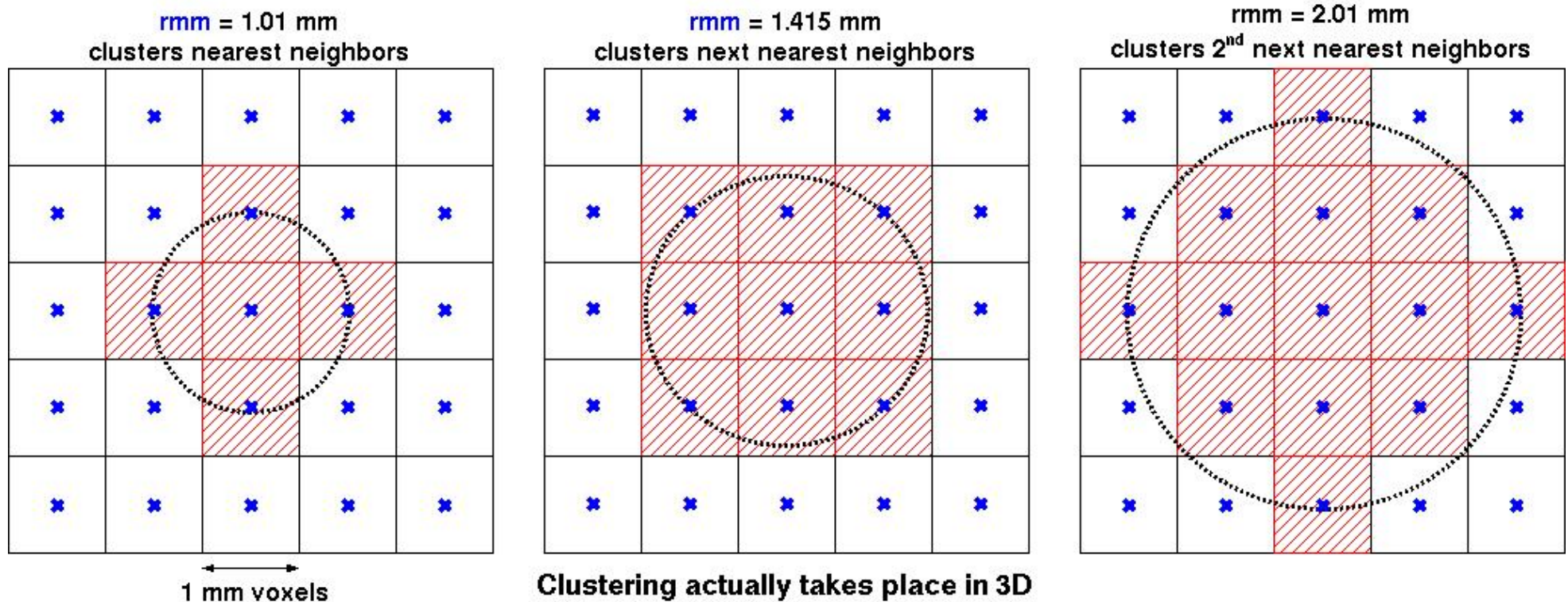
```
Dataset File:      astrip+orig
Identifier Code:   MCW_SJIVYPTEAOH  Creation Date: Wed Sep 29 07:50:58 1999
Dataset Type:      Spoiled GRASS (-spgr)
Byte Order:        MSB_FIRST [this CPU native = LSB_FIRST]
Data Axes Orientation:
  first (x) = Anterior-to-Posterior
  second (y) = Superior-to-Inferior
  third (z) = Left-to-Right    [-orient ASL]
R-to-L extent:     -74.000 [R] -to-    73.600 [L] -step-    1.200 mm [124 voxels]
A-to-P extent:     -119.531 [A] -to-   119.531 [P] -step-    0.938 mm [256 voxels]
I-to-S extent:     -119.531 [I] -to-   119.531 [S] -step-    0.938 mm [256 voxels]
R-to-L center:      -0.200 [R]
A-to-P center:       0.000 [P]
I-to-S center:      -0.000 [I]
Number of values stored at each pixel = 1
  -- At sub-brick #0 '#0' datum type is short:          0 to          733

----- HISTORY -----
[cox@varda.biophysics.mcw.edu: Wed Sep 29 07:50:58 1999] 3dIntracranial -min_val 30
-anat anat+orig -prefix astrip
```

- [3dnewid](#) — Change the ID code in a dataset's .HEAD file
 - ◇ ID codes used internally to identify datasets (e.g., in parent-child relationships)
 - ◇ [3dnewid -fun](#) shows one ID code (supposed to be globally unique)
- [3dcopy](#) — make a copy of a dataset (.HEAD and .BRIK files)
 - ◇ Is equivalent to using Unix [cp](#) on the .HEAD and .BRIK files, then using [3dnewid](#) on the copy
- [3drename](#) — rename a dataset (.HEAD and .BRIK files)
 - ◇ Is equivalent to using Unix [mv](#) on the .HEAD and .BRIK files
- [3dNotes](#) — attach notes to a dataset .HEAD file that will be printed by [3dinfo](#)
 - ◇ [Dataset NOTES](#) plugin provides an interactive way to do the same thing
- [3dbucket](#) — assemble various sub-bricks into a single “bucket” dataset
 - ◇ Lets you put diverse results into one place for easy viewing
- [3ddup](#) — make a warp-on-demand (.HEAD file only) copy of a dataset
 - ◇ Is a way to make a copy of a dataset at a new resolution
 - ◇ Then use [Define Datamode→Resam](#) and [→Write](#) to write to disk
- [3drefit](#) — modify parameters in a dataset's .HEAD file
 - ◇ Lets you “patch” mistakes made in [to3d](#) (not by you, but by someone else)

Spatial Utilities

- 3dclust — find clusters of “active” voxels and print out a report about them
 - ◇ “Active” means nonzero (survives thresholding operation)
 - ◇ Clusters are defined by a connectivity radius parameter **rmm**:



- ◇ Clustering starts by finding some nonzero voxel
- ◇ All nonzero voxels closer than **rmm** millimeters (center-to-center distance) to the given voxel are included in the cluster
- ◇ Cluster then grows outwards from all newly included voxels, using **rmm** again

◇ Clustering actually takes place in 3D:

↳ Assume cubical voxels with grid size L mm

↳ $L < rmm < \sqrt{2}L \Rightarrow$ connect voxels that share a common face

↳ $\sqrt{2}L < rmm < \sqrt{3}L \Rightarrow$ connect voxels that share a common edge

↳ $\sqrt{3}L < rmm < 2L \Rightarrow$ connect voxels that share a corner

↳ Larger values of **rmm** will jump over zero voxels

◇ You can override actual voxel size (which may not be cubical) by using the -dxyz=1 command line switch, which then pretends that voxel size L=1

◇ Sample report: **3dclust -1thresh 0.47 7 600 r1:time@1+orig**

Cluster report for file r1:time@1+orig

[Connectivity radius = 7.00 mm Volume threshold = 600.00]

[Single voxel volume = 98.4 (microliters)]

[Voxel datum type = short]

[Voxel dimensions = 3.750 mm X 3.750 mm X 7.000 mm]

Mean and SEM based on Absolute Value of voxel intensities:

Volume	CM RL	CM AP	CM IS	minRL	maxRL	minAP	maxAP	minIS	maxIS	Mean	SEM	Max Int	MI RL	MI AP	MI IS
3839	2.3	-15.3	4.4	-11.0	10.0	-28.1	-5.6	-9.4	20.6	0.0069	4.8e-04	0.0176	3.0	-13.1	5.6
2067	16.0	56.8	9.4	3.0	24.0	39.4	65.6	1.9	16.9	0.0059	4.3e-04	0.0107	17.0	61.9	13.1
1772	38.4	-5.3	0.2	24.0	52.0	-13.1	-1.9	-9.4	5.6	0.006	5.1e-04	0.0111	31.0	-1.9	-1.9
1575	-18.4	-36.7	4.5	-25.0	-18.0	-43.1	-28.1	-9.4	13.1	0.0072	0.001	-0.0181	-18.0	-43.1	5.6
1477	-1.4	-65.8	-31.7	-4.0	10.0	-69.4	-58.1	-39.4	-28.1	0.0109	0.001	-0.0201	-4.0	-65.6	-31.9
1280	24.5	-30.0	1.4	24.0	31.0	-35.6	-24.4	-9.4	9.4	0.0053	4.9e-04	-0.0089	24.0	-35.6	5.6
1181	0.7	-50.9	0.6	-4.0	10.0	-54.4	-46.9	-13.1	5.6	0.0071	0.0011	-0.0154	-4.0	-50.6	5.6
886	42.9	-28.8	-10.9	38.0	52.0	-31.9	-20.6	-13.1	-5.6	0.0059	7.7e-04	0.0096	38.0	-31.9	-9.4
689	33.2	-4.7	17.8	31.0	38.0	-9.4	-1.9	13.1	24.4	0.0074	0.0011	0.0133	31.0	-1.9	20.6
14766	9.8	-20.1	-1.8							0.0069	2.7e-04				

◇ **-1thresh 0.47**=threshold to apply to dataset ; **7**=**rmm** ; **600**=volume of smallest cluster to report (in mm³=microliters)

- [3dmerge](#) — spatially edit and/or combine datasets (the very first “3d” program!)
 - ◇ “Editing” options (applied to each input dataset) include:
 - ↪ Clipping (setting small values to zero)
 - ↪ Thresholding (setting voxels to zero based on some other sub-brick)
 - ↪ Spatial filtering (e.g., Gaussian blur)
 - ↪ Spatial clustering
 - ◇ “Combining” options include (all are voxel-wise across multiple input datasets, after the editing operations are carried out):
 - ↪ Mean; Mean of nonzero inputs
 - ↪ Maximum; Maximum absolute value
 - ↪ Count of nonzero input voxels
- [3dZeropad](#) — add planes of zeros around a dataset
 - ◇ Can also cut planes off edges of dataset
- [3dZcutup](#) and [3dZcat](#) — cut slices out of dataset; glue datasets together
 - ◇ In the slice (z) direction
 - ◇ Used when processing a big 3D+time dataset is too much

- [3dZregrid](#) — resample dataset in the slice (z) direction
 - ◇ Used when you want to compare datasets acquired with different slice thickness
 - ◇ But you should really be more careful when acquiring your datasets!
- [3dcalc](#) — voxel-by-voxel general purpose calculator
 - ◇ Useful for combining ROI masks in various ways
- [3dMean](#) — voxel-by-voxel mean of input datasets
 - ◇ Like [3dmerge -gmean](#), but simpler to use, and much faster than [3dcalc](#)
- [3dClipLevel](#) — estimate the voxel value at which to threshold an EPI dataset so as to remove most non-brain tissue
 - ◇ Can combine with [3dcalc](#) to produce a brain-only mask for data analysis
- [3dIntracranial](#) — strip the scalp and other non-brain tissue from a high-resolution T1-weighted anatomical dataset
 - ◇ Useful for volume rendering fun
 - ◇ Combined with [3dfractionize](#), is another way to make a brain-only mask
- [3daxialize](#) — rewrite a dataset in a new slice orientation
 - ◇ Useful for volume rendering — see FAQ #47

Masking Utilities

- [3dfractionize](#) — resample a high-resolution dataset to lower resolution
 - ◇ Used to take high-resolution mask (ROI) datasets to EPI resolution
 - ◇ Can compute fraction of each output voxel that is occupied by nonzero input voxels (default operation)
 - ◇ Can let input voxels “vote” on the value of output voxels (since multiple input voxels can overlay a larger output voxel, must have a scheme to decide which input value “wins”)
- [3dmaskdump](#) — print out all the voxel values indicated by a mask dataset
- [3dUndump](#) — take a text file and put its values into a dataset
 - ◇ The inverse of [3dmaskdump](#)
- [3dmaskave](#) — print out the average of voxels over an ROI
 - ◇ This can compute a 1D time series averaged from a 3D+time dataset over an ROI
 - ◇ Can also use [ROI Average](#) plugin
- [3dROIstats](#) — print out statistics of voxels from multiple ROIs
- [3dOverlap](#) — count number of voxels that are nonzero in all input sub-bricks

3D+time Utilities

- [3dTsmooth](#) — smooth a 3D+time dataset along the time axis
- [3dFourier](#) — filter time series in the Fourier domain
 - ◇ Allows more general filtering than [3dTsmooth](#)
- [3dTcat](#) — concatenate 3D+time datasets together
 - ◇ For use in [3dDeconvolve](#), for example
- [3dTstat](#) — basic statistics on 3D+time datasets
 - ◇ Voxel-wise mean, standard deviation, median, etc.
- [3dTqual](#) and [3dToutcount](#) — check 3D+time datasets for “outliers” (values that are unusual)
 - ◇ Now also included automatically in [to3d](#)
- [3dDetrend](#) — subtract least squares fits of “trend” functions from voxel time series
 - ◇ Usually better to do this at the same time as activation analysis

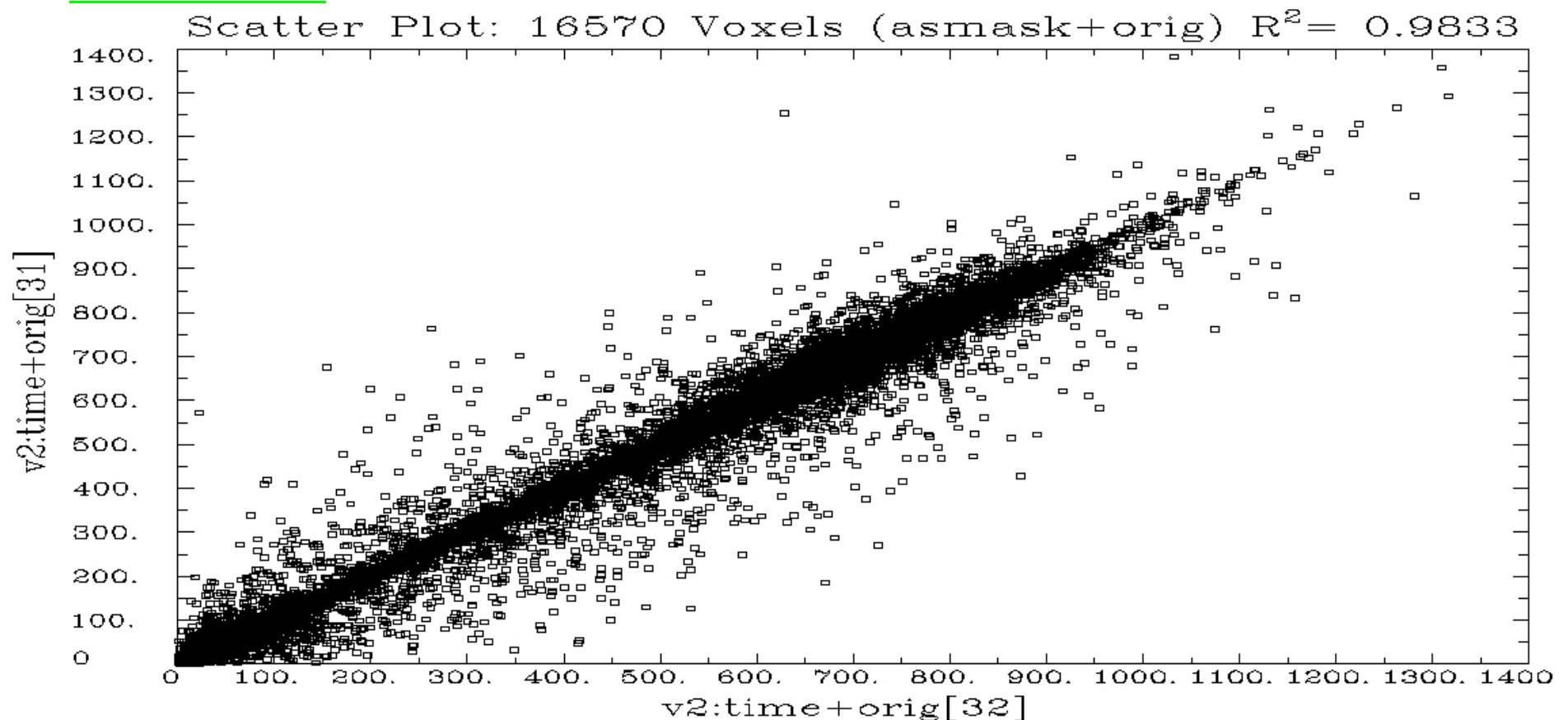
Miscellaneous Plugins

Define Datamode→Plugins

-- Cancel --	Dataset#2	Maxima
2D Registration	Deconvolution	NLfit & NLerr
3D Cluster	Draw Dataset	Nudge Dataset
3D Correlation	Dset Zeropad	Permutation Test
3D Dump98	Edit Tagset	Power Spectrum
3D Edit	Expr 0D	Render Dataset
3D Registration	Fourier Stuff	Reorder
3D+t Extract	Gyrus Finder	ROI Average
3D+t Statistic	Hemi-subtract	ScatterPlot
4D Dump	Hilbert Delay98	SingleTrial Avg
BRIK Compressor	Histogram	Threshold
Coord Order	Histogram: BFit	TS Generate
Dataset Copy	Histogram: CC	Wavelets
Dataset NOTES	LSqFit & Dtr	
Dataset Rename	maskcalc	

- 3D Edit — similar to the dataset editing -1* functions of 3dmerge
- Coord Order — set the order in which coordinates are displayed
 - ◇ AFNI default order is RAI: -x = Right +x = Left
 -y = Anterior +y = Posterior
 -z = Inferior +z = Superior
 - ◇ Can choose another order with this plugin: most common is LPI or “flipped”
- Dataset NOTES — see and edit Notes in the dataset .HEAD file
- Deconvolution — interactive version of 3dDeconvolve

- ◇ Can see model fits in graph window right away
- [Histogram](#) — graph a histogram of a sub-brick
 - ◇ Or of an ROI, or a spherical region about the AFNI focus point
- [NLfit & NLerr](#) — interactive version of [3dNLfim](#)
 - ◇ Nonlinear version of 3dDeconvolve: fit data to a general function written in C
- [ROI Average](#) — interactive version of [3dmaskave](#)
- [ScatterPlot](#) — plot values from 1 sub-brick vs. values from another



Environment Variables and .afnirc

- Operation of AFNI is affected by many Unix environment variables
 - ◇ Full documentation is in file README.environment (in AFNI distributions)
 - ◇ Environment variables can be set in your shell startup file (e.g., .cshrc) or in AFNI's startup file (.afnirc), in your home directory
 - ◇ Some environment variables can be set from the pseudo-plugin Define Datamode→Misc→Edit Environment
- Some useful environment variables (there are many more)
 - ◇ AFNI_PLUGINPATH gives the directory where AFNI will look for plugins when it starts up
 - ◇ AFNI_SESSTRAIL gives the number of directory levels to show in the Switch Session chooser
 - ◇ AFNI_HINTS can be used to turn off the popup hints (tooltips)
 - ◇ AFNI_COMPRESSOR can be used to tell AFNI programs to compress .BRIK files when they are written out
 - ◇ AFNI_AUTOGZIP can be used to tell AFNI programs to gzip compress .BRIK files if they appear like “good” candidates for compression (e.g., ROI datasets)

- ◇ AFNI_LEFT_IS_LEFT can be used to have axial and coronal images displayed with the subject's left on the display left (default is subject's left on the display right: radiological order)
- ◇ AFNI_ALWAYS_LOCK can be used to turn on inter-controller Lock at startup
- ◇ AFNI_NOSPLASH can be used to hide the AFNI splash window (but why?)
- ◇ AFNI_ENFORCE_ASPECT can be used to make defective window managers (KDE, Gnome) keep the image window aspect ratios when resizing (I then also recommend setting the window manager so that it doesn't redraw the windows during resizing operations)

- Sample .afnirc file:

```

***ENVIRONMENT
AFNI_LEFT_IS_LEFT      = YES      // images show subject's left on screen left
AFNI_graph_width       = 512      // in pixels
AFNI_graph_height      = 384
AFNI_graph_ggap        = 6        // gap between sub-graphs
AFNI_graph_data_thick  = 1        // use thick lines for data graphs
AFNI_SPLASHTIME        = 1.0      // shorten the splash screen display
AFNI_ALWAYS_LOCK       = YES      // locking windows together
AFNI_ENFORCE_ASPECT    = YES
AFNI_AUTOGZIP          = YES      // 02 Mar 2001

```

- See README.environment and README.setup for details on all environment variables and other setup issues

AFNI Command Line Switches

- Command line switches to the interactive AFNI program itself are not often needed, but are sometimes useful:
 - ◇ -R = recursively search directories for datasets (may take a long time)
 - ◇ -noqual = skip the “quality” check for marker transformations
 - ◇ -nopugins = don't load plugins
 - ◇ -skip_afnirc = don't read the .afnirc file
 - ◇ -ncolors nnn = use 'nnn' gray levels for image display (default=80)
 - ◇ -nomall = disables use of AFNI's internal malloc() (if it causes problems)

Final AFNI Fun

- Try clicking each mouse button in the blank area to the right of the Done button
- Try clicking in the splash screen
- Try using BHelp in a blank area of the AFNI controller
- To exit AFNI quickly, press the Shift key down, then click on Done