AFNI Jazzercise Hints

NB: the code snippets below will likely **not** copy+paste well into your terminal. This is not just laziness on our part, but instead it is meant to be helpfully didactic by prompting you to type the commands yourself, thereby getting used to the AFNI program syntax. And while typing, *please do* make use of the auto-completion features from both the Linux terminal and AFNI itself (→ via the "apsearch -update_all_afni_help" command).

Below are some hints that should help you answer the AFNI Jazzercise Questions.

1. See examples of input sub-brick selection in **3dbucket –help**, and consider the –prefix option.

2. Use program **`3dMean`**. Check out the **3dMean –help** menu for further assistance.

3. The **–help** menus for **`3dAutomask`** and **`3dSkullStrip`** will help you type the correct commands. One way to view the 2 output files simultaneously is to open two separate AFNI viewers.

4. Creating and Playing with ROI Masks:
   a. Use **`3dinfo`** (or the AFNI GUI) to find out that sub-bricks 2 and 4 have the desired t-statistic values or use the names of the sub-bricks we need to answer this question. In **`3dcalc`**, use the '**ispositive'** or **'step'** function to create a mask for values where "(a-4.2) > 0", say. Multiply those mask values by the same expression for dataset 'b'.

   b. Note that 3 = 1 + 2. Add mask 'a' plus two times mask 'b'.

   c. In AFNI, set **`VA_mask_4+orig`** as the overlay. Display only 4 positive color ranges.

   d. Use **`VA_mask+orig`** as the mask, and apply the **–quiet** and the **–mask_f2short** option. Redirect the output to **VA_mean.1D**.

5. Understanding the regression matrix:
   a. The polynomial regressors at the bottom should visually divide the matrix into runs.
   b. The number and shapes of the polynomial regressors should imply the degree.
   c. The regressors should visually partition into baseline, those of interest and motion.
   d. With a fast, event-related design, the regressors of interest typically look like noise, whereas with a block design, the expected BOLD response patterns should be smooth and elongated.
   e. The baseline parameters can help determine the run lengths. But wasn't the length 150 TRs?
   f. Significant motion should be captured by jumps in the motion parameters. But wasn't there some large motion around TR #42?

6. Fun with 1D files:
   a. First, run the AFNI program **`count`** to create 3 rows of these numbers. Second, run the AFNI program **`1dtranspose`** to convert each of these 3 rows to a column. Alternatively, there is an option to do this only using **count**.

b. Now combine the 3 columns into one column with the AFNI program **`1dcat`**.

c. See **1dcat –help** for assistance in combining separate 1D files into one big 1D file.

d. Do arithmetic on the 1D files with AFNI program **1deval**. See **1deval –help** for further assistance. Note **3dTstat** can do this particular exercise too with a simpler command line. It is sometimes useful to have "3d" programs operate on "1d" files.

7. Fun with the AFNI GUI
   a. If you right-click on the gray-scale bar of any viewing plane (e.g.,, sagittal), you will find a hidden pop-up menu with several options. One of those options can be used to answer this question.

   b. All of the answers can be found in the Define OverLay control panel in the AFNI GUI. Hunt around for hidden popup menus by left-clicking in the color bar. Also place your cursor over the color bar panels to see what appears.

   c. The answer can be found in one of the buttons located at the bottom of the sagittal viewing plane (e.g., Disp, Sav1.ppm, Mont, etc…)

   d. Remember what we learned in the Talairach hands-on? Right-click in an image window.

   e. The answer can be found in one of the buttons located at the bottom of the sagittal viewing plane (e.g., Disp, Sav1.ppm, Mont, etc…)

   f. Right- and Left-click anywhere you can in the afni GUI in search of this hidden Mission Statement. There is one particularly large open space.

8. Doing Calculations in AFNI:
   a. Use **`3dinfo`** to find information about a dataset.

   b. **`ccalc`** is a simple calculator program in AFNI

   c. **`1deval`** is a simple 1D file calculator program; **`1dplot`** is a simple graphing program

9. Aligning data:
   The default is to align "anat2epi". The @AddEdge script is called by the –AddEdge option. Remember the @AddEdge script needs to be used to drive AFNI.

10. Image Filtering:
    a. The AFNI program **`3dmerge`** can be used for a variety of tasks, including smoothing. For this question, the Gaussian filter may be a good choice. The program 3dBlurInMask can also be used.

b. The AFNI program **`3dLocalstat`** looks in "neighborhoods" around each voxel. To get it to use voxels units for the neighborhood instead of mm, use a negative number.

c. The AFNI program **`3danisosmooth`** sharpens edges and smoothes images. It usually shows 10 iterations (default), but that may be too much for this example. Use the **–viewer** option to pick something lower and try it again with the new option.

11. Random Exercises with AFNI Datasets:
    a. First, use **`3dinfo`** to determine the xyz-orientation of the dataset. Then run **`3dresample`** or **`3daxialize`** to re-orient the dataset.

    b. Use **`3dbucket`** or **3dcalc** to create 2 separate datasets from **`func_slim+orig`**. Remember that in AFNI, sub-bricks begin at 0, not 1.

    c. Program **`3dbucket`** can also be used to combine datasets together.

    d. The AFNI program **adwarp** can be used to transfer the Talairach transformation of an anatomical dataset to a "follower" dataset like **`func_slim+orig`**. Pay special attention to the **–dxyz** option available in **adwarp** (see **adwarp –help**).

    e. Find the maximum voxels with **`3dExtrema`** and use **`whereami`** to find the atlas position of the maximum voxel.

    f. Use program **`3dZcutup`**. This program cuts up volumes in the z-direction. Check out the **3dZcutup –help** menu for further assistance.

12. Volume Rendering:
    a. Find the Volume Renderer in the Datamode, Plugins menu. (It's the "new" renderer).
    b. You will need to create the func_slim+tlrc dataset if you didn't do this in question 10d (easiest from the Datamode menu).
    c. If you didn't get the location from the 10e exercise, use the Interactive clusterize feature to find maxima, or just eyeball it and then right-click to show atlas regions. You will need to turn on show TT regions in both the overlay and the render overlay regions to see them in both viewers. Right-clicking on the image window accesses both Where Am I and Show Atlas Colors menus.

13. Simple statistics:
    d. Info is available in the GUI and from the command line with 3dinfo.
    e. Same as a, but ranges are also available in the Overlay panel
    f. The threshold slider is easiest, but the **cdf** and **fdrval** programs can be used on the command line and in scripts.
    g. Same as a.