In this lab, you will explore fMRI data from a single subject that was collected during a simple language localizer of approximately 5 minutes.

During the scan, three kinds of trials were visually presented: well-formed sentences, lists of unrelated nouns, and consonant strings (the paradigm was modeled on an anterior temporal cortex localizer used by Rogalsky & Hickok, 2008 *Cerebral Cortex*). Each trial was 4 seconds long: an initial 400ms of fixation cross followed by 9 consecutive words presented using RSVP (rapid serial visual presentation) at 400ms stimulus onset-asynchrony (SOA), 200ms on and 200ms off). The three conditions were presented 20 times each in randomized order, intermixed with a total of 80 additional seconds of fixation periods (in which only a fixation cross was presented), in order to improve the deconvolution of the event-related responses.

**Example Stimuli:**
(A) - Sentence - The gray mouse quickly scurried underneath the dusty rug.
(B) – Noun list - pillow dragon tile clay milk ladder truck foil shell
(C) – Consonant string list – trkjcr tphqj rnd bspjsqc kdr bbqtgx ckldp bfqnkr rhvnj

The data were recorded from a 3T scanner. During the functional run, 160 functional volumes (36 axial slices, anterior commissure – posterior commissure AC-PC aligned, 3mm slice thickness, .3mm skip, in-plane resolution of 3.125mm) were acquired with a gradient-echo sequence (repetition time=2s, echo time=25ms, flip angle = 90deg, interleaved acquisition).

Note that the analysis procedure for this lab is greatly simplified compared to the state-of-the-art in fMRI analysis today. If you would like to develop an analysis pipeline for a ‘real’ fMRI project, come talk to me and I can direct you to some local experts. The analysis procedure in this lab has been streamlined in order to give you a sense of the effects of some of the most critical preprocessing steps without getting down in the weeds. These also represent a minimal set of procedures that you could apply to your class project data.

**I am offering an alternative assignment for those who may find it challenging to obtain computer resources for this assignment: write a 2-3 page (single-spaced) report discussing the benefits and drawbacks of the fMRI measure for cognitive neuroscience of language. Include some discussion of important consideration for experiment design and data analysis of language processing experiments.**

The data for this lab is available at ling.umd.edu/~ellenlau/courses/nacs642/MRI.zip.

You can do this lab in two ways:
(1) On your own Mac or Linux computer, after installing the AFNI software (just google afni, etc.)

(2) On the cephalopod analysis machine in the MEG lab in MMH 3416, on which you can get a user account by emailing Anna Namyst (amnamyst@umd.edu). If you have access to a Mac, you can then easily log into the machine remotely using Screen Sharing, so that you do not have to come to MMH to work on the assignment. The analysis files are in a shared directory that everyone has access to.

**Setting up a remote access for cephalopod (Mac users only)**

*Note: You need to have Screen Sharing enabled. Go to Applications-&gt;System Preferences-&gt;Sharing and turn it on by ticking the box for Screen Sharing on the left panel.*

After receiving a new user account from the lab manager, open Finder. Click on the ‘Go’ tab, then ‘Connect to server’. Enter the server address vnc://files.ling.umd.edu, and after entering your new credentials, select ‘Connect to a virtual display’. You should be directed to a new virtual desktop on the cephalopod machine.

**Setting up for analysis (on all machines):**

After logging in remotely or at the physical machine, open the Terminal application (Applications-&gt;Utilities-&gt;Terminal). To start running the C shell, type

```
tcsh
```

To make sure you can run AFNI commands, you need to add AFNI to your path by typing the following command:

```
echo 'set path = ( $path /Users/Shared/abin )' >> ~/.cshrc
```

Now close the terminal window, and re-open a new one, start running the C shell by typing `tcsh`, and now you can make sure that AFNI is working for you by running

```
afni &
```

When you do this, a number of AFNI windows should pop up (some of which you can close by clicking twice on the ‘Done’ button). Close them all for now. If an AFNI window didn’t pop up, secure help before going further.

**Analyzing the data**

*On cephalopod, the data for this lab is already downloaded in the following directory: /Users/Shared/Courses/NACS642/MRI_XX, where XX is your initials. Therefore, in the instructions below we will describe the file paths that would be appropriate if you were on this machine. If you’re working on another machine, just substitute the location of the directory where you downloaded the data.*
First, in the Terminal, go into the data directory by typing:

```bash
cd /Users/Shared/Courses/NACS642/MRI_XX
```

Inside the directory you should see a subdirectory called ‘dicoms’ (from the Terminal, you can see what’s in the directory by simply typing ‘ls’).

Most MRI data comes off the scanner in the form of ‘DICOM’ or .dcm files, which are in a standardized medical image format that usually returns one file per brain volume (so, for example, in a functional run that takes 200 images of the brain across the scan, you would get back 200 .dcm files). Take a look inside the directory (from the command line in Terminal, you can type ‘cd dicoms’ and then ‘ls’). You should see 6 subdirectories, that each contain a number of dicom files. These correspond to six of the scans that were conducted in this session. The first is the localizer, the second is the anatomical scan (MPRAGE), the third is a short test functional run to make sure the slice parameters look correct, and the fourth one (LANGLOC) contains the functional data that we are actually interested in. Since we collected 162 images in this run, there are 162 dicom files in the LANGLOC directory.

The multiple-dicom file format is kind of inconvenient, so the first thing we want to do is convert this to a format in which all of the dicoms from a single run are contained in a single file. We are going to do this for both the anatomical data and the functional data, since those are the two pieces that are most relevant for our analysis. We use AFNI’s `to3d` command to do the conversion.

**To convert the anatomical data:**

First, make sure you are in the top level of the MRI directory (`cd /Users/Shared/Courses/NACS642/MRI_XX`). Now, run the following command:

```bash
to3d -prefix p01_anat -session anat dicoms/2-t1_mpr_sag_p2_iso_0.9/i*
```

This command says to take the dicom files specified by the last argument (all the files in that folder that start with ‘i’) and combine them into a file that will be labeled with the prefix p01_anat and will be placed in the ‘anat’ subdirectory.

1. According to the screen output, how many 2D slices were detected? What are the voxel dimensions?

**To convert the functional data:**

Now do the same thing for the functional data by running the following command (this is all one line, no line break):

```bash
to3d -prefix p01_LANGLOC -session func -time:zt 36 162 2000 alt+z dicoms/7-ep2d_bold_nomoco_LANGLOC_162VOL/i*
```
For functional data, we needed to provide a little more information to the command about the parameters of our scan so that it could be parsed correctly, for example the number of slices that we acquired (36), the number of whole-brain images we acquired (162), and the time resolution (TR), the amount of time we’re using to acquire each of those images (2000ms or 2sec).

2. According to the screen output, what are the voxel dimensions? Is this bigger or smaller than the anatomical voxels?

Note that you’ll see a warning about outlier data points; although there are a number of methods to investigate and address outliers, we won’t be worrying about them in this lab.

**Viewing the data:**

Now view the data by typing `afni &` at the command line (including the & will allow you to use your Terminal while AFNI is open). This should pull up the AFNI viewer.

The anatomical image should be automatically loaded in the axial and sagittal views, you can add the coronal view by clicking ‘Image’ next to ‘Coronal’ in the viewer:
3. Take a screenshot of your favorite anatomical view. Does anything strike you as interesting about the data?

Now to view the functional data, you need to click on the ‘Switch’ button next to ‘DataDir’ and choose ‘func’ from the menu, then click ‘Set’. The functional dataset should automatically be loaded, as it’s the only thing in that directory.

4. Take a screenshot of the functional data. What do you notice, compared to the anatomical data?

5. You can flip through the 162 functional images we collected by clicking on the arrow buttons next to 'Index'. Do you detect very much motion from one image to the next?

Now we can try running the regression analysis to determine if we see effects of our manipulation (presenting language stimuli) on the activity in each voxel. To do this, we need to know when the stimuli were presented. Critically, this information is in the Matlab log that was output during stimulus presentation, in your ‘timecourses’ directory as ‘p1_LANGLOCSet1.log’.

6. Open up this file to take a look at it in the text editor (for example, with TextWrangler). What do you think each of the 3 columns represents?

Our goal is to regress the signal measured in each voxel across time against the timecourse of each stimulus condition. In order to this, we need to transform the timing data from the Matlab log into timecourses for each condition. There are several ways to do this--in the ‘timecourse_for_plot’ directory, we have done this such that we assign a ‘1’ to any timepoint where that stimulus condition is being presented, and a ‘0’ to any timepoint where it is not, saved into the .1D files. If you open up these files in a text editor (e.g. TextWrangler), you will see they are just a series of 1s and 0s. You can view these timecourses using AFNI’s 1dplot command. Make sure you are in the ‘timecourses’ directory (cd
1dplot p1_langloc_cons.1D p1_langloc_word.1D p1_langloc_sent.1D

7. Take a screenshot of the resulting plot. Is there much overlap between the timecourses? Are the stimuli from each condition presented at a regular rate? Why or why not?

We need to do one more step now, which is to convolve the stimulus timecourses with an estimate of the hemodynamic response. In other words, since we know that the BOLD response is not going to be at its maximum immediately after the stimulus is presented, but only ~6 seconds later, we want to adjust the stimulus timecourses so that they take this delay into account. We can do this with AFNI's waver command. In order to do this, we have to have the timecourses in a different format, where we list only the timepoints in which the stimuli appeared. These timecourses will be created in the timecourse_for_waver directory.

Click ‘Done’ on the timecourses plot. Change to the ‘timecourses’ directory (cd
/Users/Shared/Courses/NACS642/MRI_XX/timecourses) and then run the following commands. Note the funny quotes in this command; you should use the quote button with the ~ on the upper left of the keyboard.

waver -G -dt 2 -tstim `cat timecourse_for_waver/p01_LANGLOC_sentence.1D` > p01_LANGLOC_sentence_ideal.1D
waver -G -dt 2 -tstim `cat timecourse_for_waver/p01_LANGLOC_word.1D` > p01_LANGLOC_word_ideal.1D
waver -G -dt 2 -tstim `cat timecourse_for_waver/p01_LANGLOC_consonant.1D` > p01_LANGLOC_consonant_ideal.1D

Now you can plot the idealized BOLD response timecourses by using 1dplot:

1dplot p01_LANGLOC_consonant_ideal.1D p01_LANGLOC_word_ideal.1D
p01_LANGLOC_sentence_ideal.1D

8. Take a screenshot of the resulting plot. How is it different from the timecourses you plotted in (6)?

Now that we have the idealized timecourses, we can finally move ahead to regressing them against the MRI data. Click on ‘Done’ in the previous figure. Change into the functional data directory: cd /Users/Shared/Courses/NACS642/MRI_XX/func

First, take a look (e.g. with TextWrangler) at the 3 text files in this directory, sentVcons.txt, wordVcons.txt, and sentVword.txt. These files contain the codes for testing 3 contrasts between conditions: a contrast of sentences vs. consonant strings and a contrast of words
vs. consonant strings. We'll input these into the regression command to give us maps of these contrasts.

The command for running the regression is kind of long, so it's already been input for you in the text file in the main MRI directory called 'regression1.txt'. Copy the whole contents of this file, and paste them into the command window. The command will output a file called 'p01_LANGLOC_shrf+orig.BRIK' that contains the output of the regression. To view the result, open afni by typing 'afni &'. Click on the 'Underlay' button and choose the original functional dataset p01_LANGLOC. Click on the 'Overlay' button and choose the new regression dataset p01_LANGLOC_shrf. Now click on 'Define Overlay', and a new set of options should pop up. Approximately match the example image (below) by changing the values on the buttons underneath the colored bars. These settings make it so that all positive beta values are in warm colors and all negative ones are in blue, and so that shortly we will be seeing values for which the t-test is significant at p < .001 (you can later change this by moving the slider).

Now you can view the results of different contrasts by changing what is specified in the 'OLay' and 'Thr' fields on the right. The 'OLay' field should always have something labeled 'Coef' in it, since we want to display the beta coefficients. The 'Thr' should always have 'Tstat' in it, since we want to threshold the image based on the result of the t-test. You need to change both every time you switch to a new contrast.
9. First take a look at the consonant condition, as indicated in the image above. This contrast is contrasting the consonant string condition with rest (which was a fixation cross in this experiment). Scroll through the brain (you may need to enlarge the window and use the arrows at the bottom right of each plot). Do you see visual activity, as would be expected for any visual stimulus larger than a fixation cross? Illustrate with a screenshot.

10. Now take a look at the SentvsCons contrast. Since sentences would engage language processing more than consonant strings, we’d expect to see activity in left temporal cortex. Do you see any evidence of this here? (don’t forget that right=left as a default in most MRI viewers, including AFNI). Illustrate with a screenshot.

11. Take a look at the other contrasts between conditions. Do they look as you might expect or are there any surprises?

OK, now that we have looked at the results of the regression against ‘raw’ data, we are going to try a couple pre-processing steps that can allow us to reduce the amount of unexplained noise in the data and ‘smooth’ the image so that larger areas showing contrast will pop out and smaller areas that are more likely to be due to artifact will be masked out.

First, we’re going to do motion correction. This will align all of the images collected in the run to the first image. For this we will use the afni 3dvolreg command. Make sure you are in the func directory (cd /Users/Shared/Courses/NACS642/MRI_XX/func) and run

```
3dvolreg –base 2 –zpad 4 –prefix p01_LANGLOC_vr –dfile p01_LANGLOC_volreg p01_LANGLOC+orig
```

This will output a re-aligned dataset called p01_LANGLOC_vr.

Now, we smooth. This basically averages neighboring voxels together to result in a more ‘blurred’ set of activations. It reduces spatial precision, but can increase the reliability of your effects because subthreshold activity in neighboring voxels will act to strengthen each other. We use afni 3dmerge.

```
3dmerge -lblur_fwhm 6 –prefix p01_LANGLOC_vr_sm6 –doall p01_LANGLOC_vr+orig
```

This will apply a smoothing kernel of 6 to the re-aligned data, and will output it to p01_LANGLOC_vr_sm6.

Finally, let’s run the regression on this preprocessed data and see if the results look better. This version of the regression command is in ‘regression2.txt’, so you can just copy and paste the contents of that file to the command window.

12. Open up afni and load the new version of the data (p01_LANGLOC_shrf_vr_sm6) as the Overlay. Use the same viewing parameters you used earlier, and take a look at the
contrasts that you examined before. How have the results changed? Illustrate with a screenshot. Do you think that preprocessing was helpful or not?