# AFNI Start to Finish:
# How to Analyze Data with AFNI

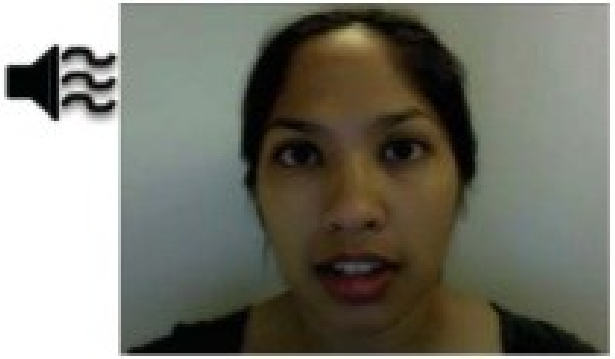
- picture this experiment as your own
  - decisions on processing were made by you (and your colleagues), hopefully before acquiring any data
  - there is no single "correct" way to analyze data,
- focus on understanding the processing steps
  - in light of your having chosen which steps to perform
- practice the good habit of reviewing results
  - do the initial images look good?
  - review each processing step along with data
  - are the EPI and anat well aligned by the end?
  - do the statistical results look reasonable?
- how does one create a processing script (based on design decisions)?
  - use **`afni_proc.py`**, or write script by hand
- how does one get data to a standard space for group analysis?
  - tell **`afni_proc.py`** do to it
  - or apply anatomical transformation to results via **`adwarp`**

# Review of stimulus conditions

- Speech Perception Task: Subjects were presented with audiovisual speech that was presented in a predominantly auditory or predominantly visual modality.
- A digital video system was used to capture auditory and visual speech from a female speaker.
- There were 2 types of stimulus conditions:

(1) **Auditory-Reliable**

Example: Subjects can clearly *hear* the word "cat," but the video of a woman mouthing the word is degraded.

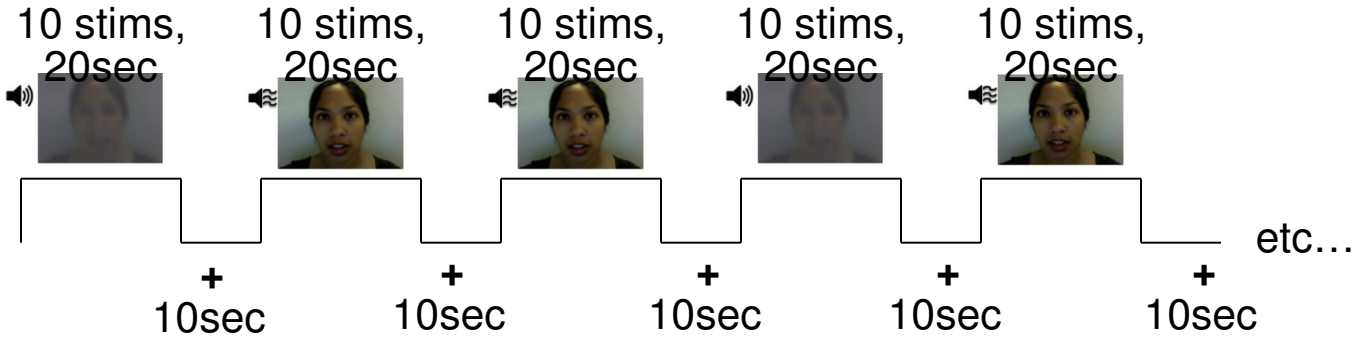


(2) **Visual-Reliable**

Example: Subjects can clearly *see* the video of a woman mouthing the word "cat," but the audio of the word is degraded.

## ❖ Experiment Design:

- There were 3 runs in a scanning session.
- Each run consisted of 10 blocked trials:
  - 5 blocks contained Auditory-Reliable (*Arel*) stimuli, and
  - 5 blocks contained Visual-Reliable (*Vrel*) stimuli.
- Each block contained 10 trials of *Arel* stimuli OR 10 trials of *Vrel* stimuli.
  - Each block lasted for 20 seconds (1 second for stimulus presentation, followed by a 1-second inter-stimulus interval).
- Each baseline block consisted of a 10-second fixation point.



## ❖ Data Collected:

- 2 Anatomical datasets for each subject, collected at 3 tesla.
  - 175 sagittal slices
  - voxel dimensions = 0.938 x 0.938 x 1.0 mm

- 3 Time Series (EPI) datasets for each subject.
  - 33 axial slices x 152 volumes = 5016 slices per run
  - TR = 2 sec; voxel dimensions = 2.75 x 2.75 x 3.0 mm

- Sample size, n = 10 (all right-handed subjects)

# afni_proc.py

- What is **afni_proc.py**?
  - a program used to generate processing scripts for single subject analysis
  - generated scripts are in **tcsh** syntax
  - scripts are written to be easily read and modified
  - why create a script?
    - it is a permanent record of the processing steps
    - it can be re-run or modified to run on more subjects

- What information is needed by **afni_proc.py**?
  - **minimum**: EPI data and stimulus timing files (in order to do regression)
  - basis functions for regression (GAM, BLOCK, etc.)
  - choose processing blocks: align EPI/anat? tlrc? despike? RETROICOR?
  - many options are available:
    - estimate smoothness, censor TRs with excessive motion, etc.
    - see "**afni_proc.py -help**" for details

- **Pros**
  - quick way to create a processing script
  - user does not need to be a master of shell scripting
  - more trust that syntax does not have typos
  - good for learning (FMRI processing, Unix/shell scripting, AFNI commands)
  - can compare against manually generated scripts
    - for sanity checks and bug detection
  - processing script generates many files to help review data/detect problems
    - outlier counts (**`outcount*.1D`**), motion estimates (**`dfile.*.1D, motion*.1D`**), ideal regressors/sum of ideals (**`ideal*.1D, sum_ideal.1D`**), estimates of data smoothness (**`blur_est*.1D`**), script to quickly review original EPI data (**`@epi_review.$subj`**)
- **Cons**
  - some users may not bother to review script
  - not every AFNI program has an **`afni_proc.py`** interface
    - have 'empty' processing block for such commands
  - not yet done:
    - -stim_times_IM/AM/AM2: requires (easy) script changes
    - varying basis functions: requires (easy) script changes
    - GUI (**on the way!**)

# afni_proc.py help sections

- there is a lot of help to be found in the "afni_proc.py -help" output
- list of main sections in the help:
  - program introduction : basic overview of the program
  - PROCESSING BLOCKS : list of possible processing blocks
  - DEFAULTS : basic defaults, per processing block
  - EXAMPLES : common examples of running this program
  - NOTE sections : details on various topics
    - TIMING FILE NOTE
    - MASKING NOTE
    - WARP TO TLRC NOTE
    - RETROICOR NOTE
    - RUNS OF DIFFERENT LENGTHS NOTE
    - SCRIPT EXECUTION NOTE
  - OPTIONS : descriptions of all program options
    - informational : options to get quck information and quit program
    - general execution : options not specific to a processing block
    - block options : specific to blocks, in default block order

- **Overview of Remaining Steps**

  - data is under **`AFNI_data6/FT_analysis`**
  - review directory contents and note suject data under directory **`FT`**
  - review the **`afni_proc.py`** command
  - execute the **`afni_proc.py`** command to create processing script
  - execute the "proc" script to process the data
  - review processed data
    - use "proc" script as a guide for what data to view
    - focus on run 1 here, to save time
    - use multiple **`afni`** controllers to view both input and output of each block
  - get results to standard space
  - run group analysis (**`3dANOVA2 or 3dMEMA`**)

- **Class Work:**

1. go to directory AFNI_data6/FT_analysis; see what is there

```
cd AFNI_data6/FT_analysis
ls -l FT
cat FT/AV1_vis.txt
```

2. review the afni_proc.py command

```
cat s01.ap.simple
```

3. execute s01.ap.simple (note the output script, **`proc.FT`**)

```
./s01.ap.simple
```
(or: **`tcsh s01.ap.simple`**)

```
ls -l
```

4. process the data (as suggested by **`afni_proc.py`**)
   1. takes ~5 minutes (on my laptop)

```
tcsh -xef proc.FT |& tee output.proc.FT
```

5. while processing data, review "proc" script

```
gedit proc.FT
```

6. review processed data (input and output of each step)

```
cd FT.results
afni &
```

1. Note what is under **AFNI_data6/FT_analysis**.

| | | |
|---|---|---|
| **FT** | - | subject data directory |
| **s01.ap.simple** | - | class afni_proc.py script |
| **s02.ap.align** | - | more advanced script |
| **s09.cleanup** | - | remove analysis results |
| **s11.proc.FT** | - | result of **s01.ap.simple** |
| **s12.proc.FT.align** | - | result of **s01.ap.align** |

under **FT**

| | | |
|---|---|---|
| **AV1_vis.txt** | - | visual reliable timing |
| **AV2_aud.txt** | - | autidory reliable timing |
| **FT_anat+orig.BRIK/HEAD** | - | anatomical dataset |
| **FT_epi_r1+orig.BRIK/HEAD** | - | EPI run 1 |
| **FT_epi_r2+orig.BRIK/HEAD** | - | EPI run 2 |
| **FT_epi_r3+orig.BRIK/HEAD** | - | EPI run 3 |

AV1_vis.txt:

```
60 90 120 180 240
120 150 180 210 270
0 60 120 150 240
```

2. Review the contents of the **`s01.ap.simple`** script.

```
afni_proc.py -subj_id FT                    \
        -dsets FT/FT_epi_r?+orig.HEAD       \
        -copy_anat FT/FT_anat+orig          \
        -tcat_remove_first_trs 2            \
        -regress_stim_times FT/AV*.txt      \
        -regress_stim_labels Vrel Arel      \
        -regress_basis 'BLOCK(20,1)'        \
        -regress_est_blur_errts             \
        -regress_opts_3dD                   \
            -gltsym 'SYM: Vrel -Arel'
```

Options:

- **`-subj_id`**: subject ID, which will be used in dataset names
- **`-dsets`**: the EPI datasets, one per run
- **`-copy_anat`**: the anatomical dataset - will be copied to the results dir
- **`-tcat_remove_first_trs`**: # TRs to remove from the beginning of each run (prior to magnetization steady state)
- **`-regress_stim_times`**: the list of stimulus timing files
- **`-regress_basis`**: basis function used by 3dDeconvolve in the regression
- **`-regress_est_blur_errts`**: estimate data smoothness from residuals
- **`-regress_opts_3dD`**: extra options given directly to 3dDeconvolve