

# Final Report - fMRI

An overview on fMRI concepts  
using FSL

Submitted to: Dr. Mina M. Makaary  
By: Mohand E. Attya

# **Table of Contents:**

[I. Part I - BET & FEAT FMRI Analysis](#)

[II. PART II - NeuroAnatomy](#)

[1. I. Precuneus Cortex](#)

[2. II. Cingulate Gyrus](#)

[3. III. Parietal Lobe](#)

[4. IV. Thalamus](#)

[III. Part III - Statistical Analysis](#)

[IV. PART IV - Scripting 1<sup>st</sup>. Level Analysis](#)

[V. PART V - 2<sup>nd</sup>. Level Analysis - Avg. run1 & run2](#)

[VI. PART VI - 3<sup>rd</sup>. Level Analysis - Group Analysis](#)

[VII. PART VII - ROI Analysis - Last Chapter](#)

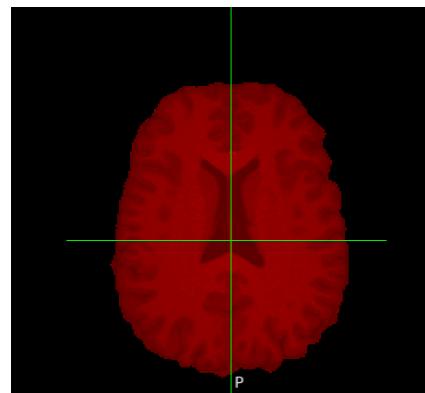
# PART I

Motion Correction and Brain Extraction

# 1. Lecture Task:

## Subject 1:

Applying skull striping (BEAT) using threshold 0.2



## Run 1:

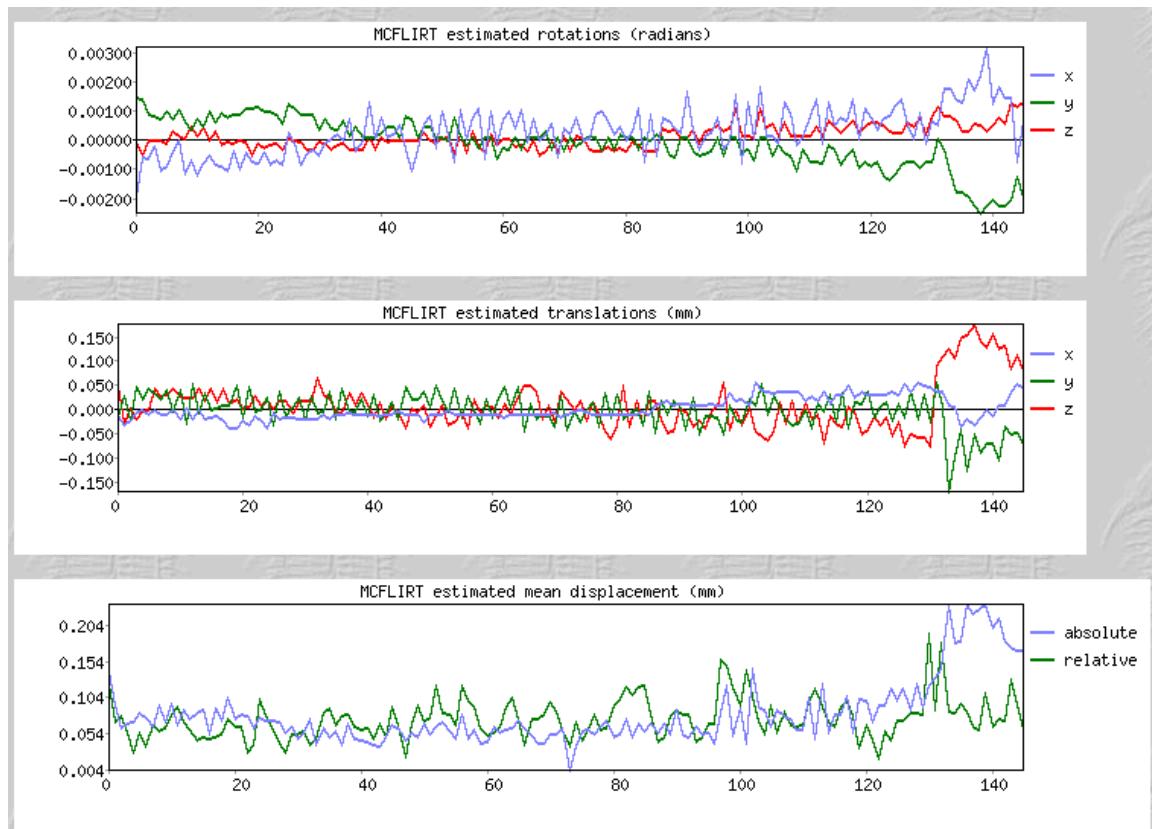
Smoothing Kernel Size: **5mm.**

Registration - DOF: **12 DOF**

Normalization - DOF: **12 DOF**

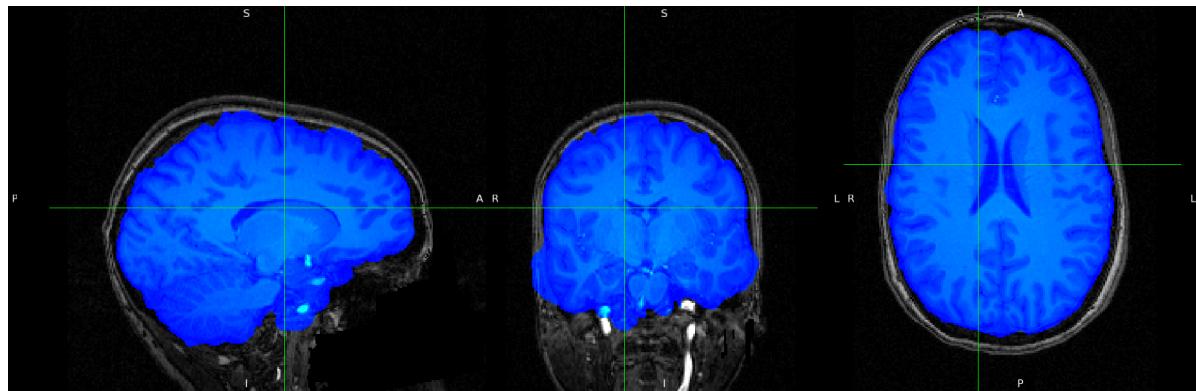
## Results of Motion Correction:

Mean Displacement: Relative 0.08 - Absolute 0.07

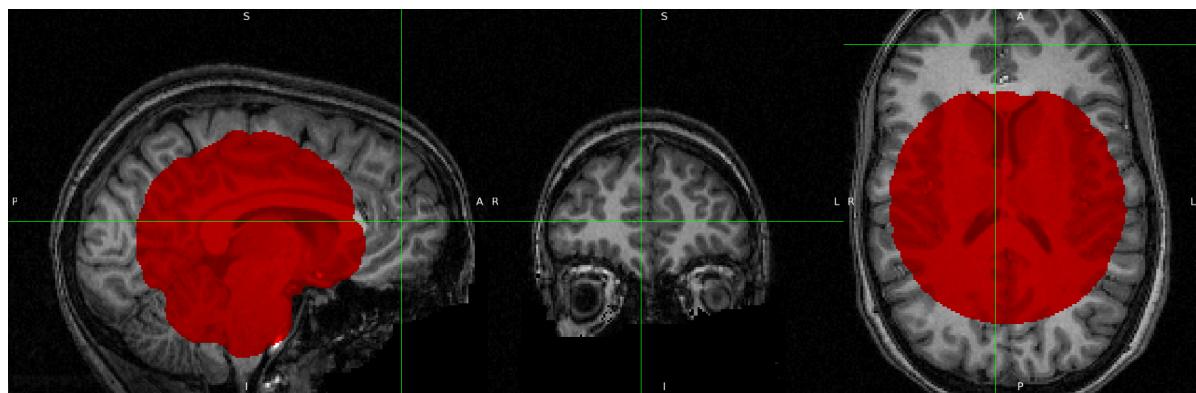


## 2. Exercise:

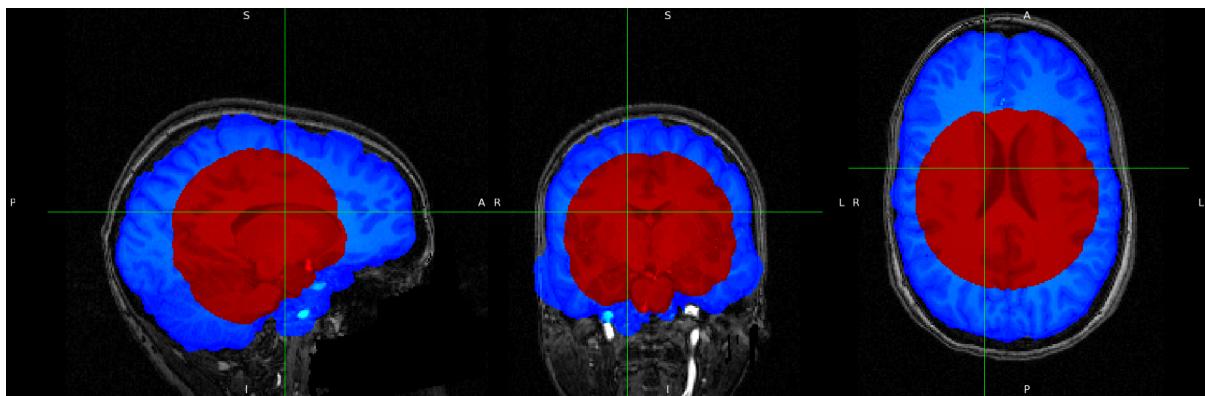
### Ex. 1:



Applying skull striping (BEAT) using threshold 0.1



Applying skull striping (BEAT) using threshold 0.9



Both Thresholds in one Image Red=0.9 and Blue=0.1

### Results:

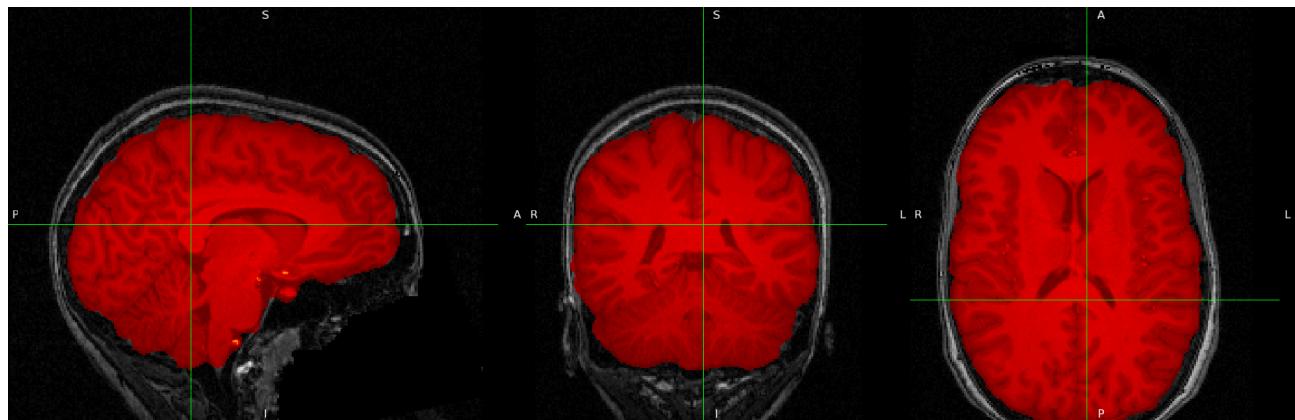
As expected the 0.9 threshold shows more certainty of containing Brain Voxels, but the loss in data is very large. However, the 0.1 threshold shows less certainty, while having a very low loss in data.

I will choose the 0.1 Threshold to avoid losing important voxels; sacrificing more computational time.

## Ex. 2:

### Subject 8:

Applying skull striping (BEAT) using threshold 0.2



### Run 2:

Smoothing Kernel Size: **5mm**.

Registration - DOF: **12 DOF**

Normalization - DOF: **12 DOF**

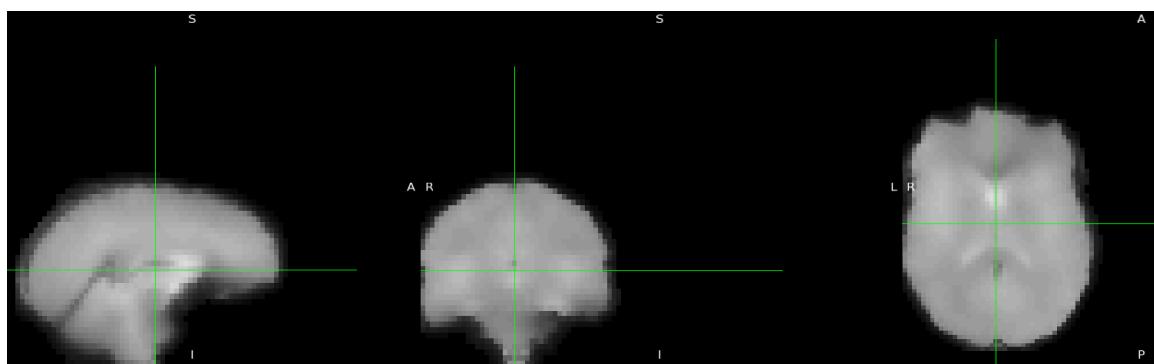
### Results of Motion Correction:

Mean Displacement: Relative 0.13 - Absolute 0.07

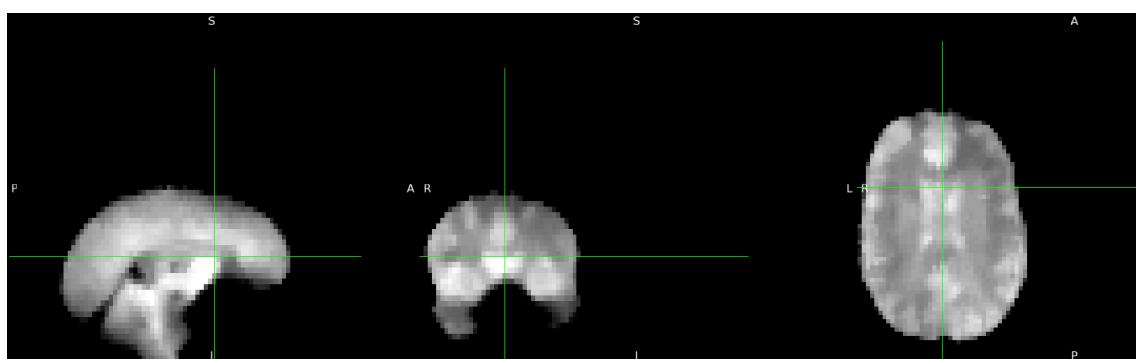


### Ex. 3:

The func image processed with the 3mm kernel had less spatial smoothing compared to the one processed with the 12mm kernel. Smoothing with a larger kernel (12mm) resulted in more blurring of the image, potentially reducing spatial detail.



12 mm. Kernel



3 mm. Kernel

## Ex. 4 & 5:

### Run 1:

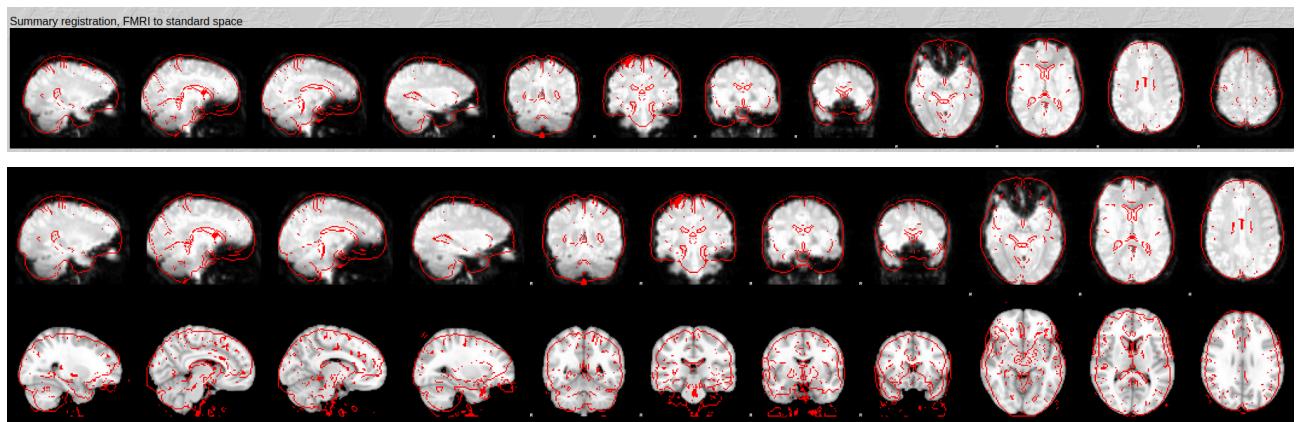
Smoothing Kernel Size: **5mm**.

Registration - DOF: **3 DOF**

Normalization - DOF: **3 DOF**

**Registration Accuracy:** BAD

**Time Efficiency:** Very Fast



### Run 1:

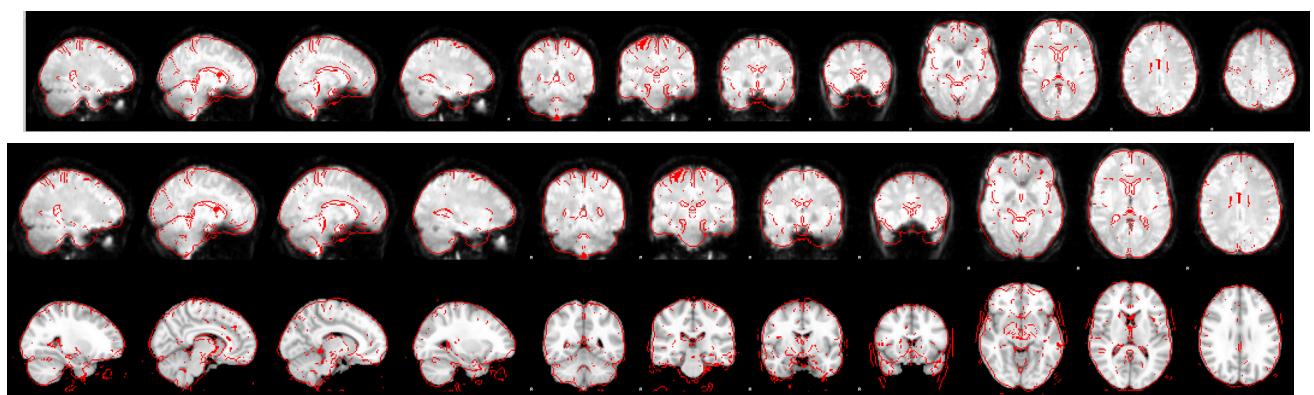
Smoothing Kernel Size: **5mm**.

Registration - DOF: **12 DOF**

Normalization - DOF: **12 DOF**

**Registration Accuracy:** GOOD

**Time Efficiency:** Slow

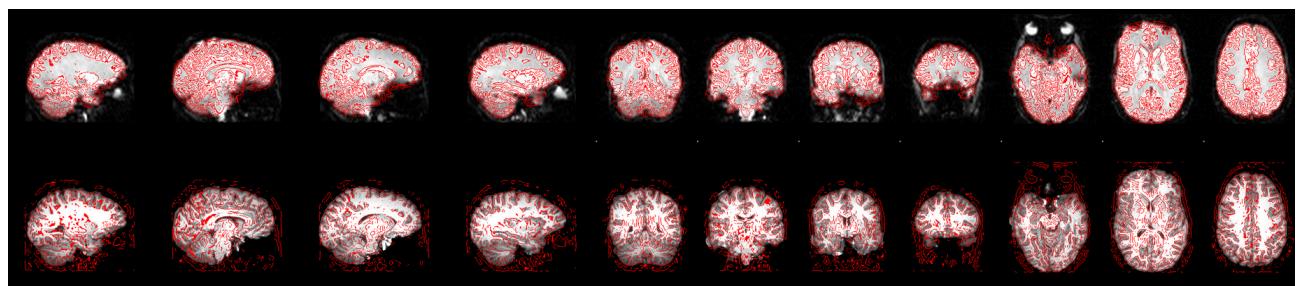


## **Run 1:**

Smoothing Kernel Size: **5mm**.  
Registration - DOF: **BBR DOF**  
Normalization - DOF: **12 DOF**

**Registration Accuracy:** GOOD

**Time Efficiency:** Very Slow



## **SUMMARY:**

Depending on the time utilization and the accuracy target I would recommend using BBR for best accuracy and 3-DOF for highest speed.

Personally, I would Choose 12-DOF as it gives a very acceptable accuracy in a very reasonable time; but again it mainly depends on the application.

# PART II

Neuro-Anatomy

# I. Precuneus Cortex

## 1. Memory:

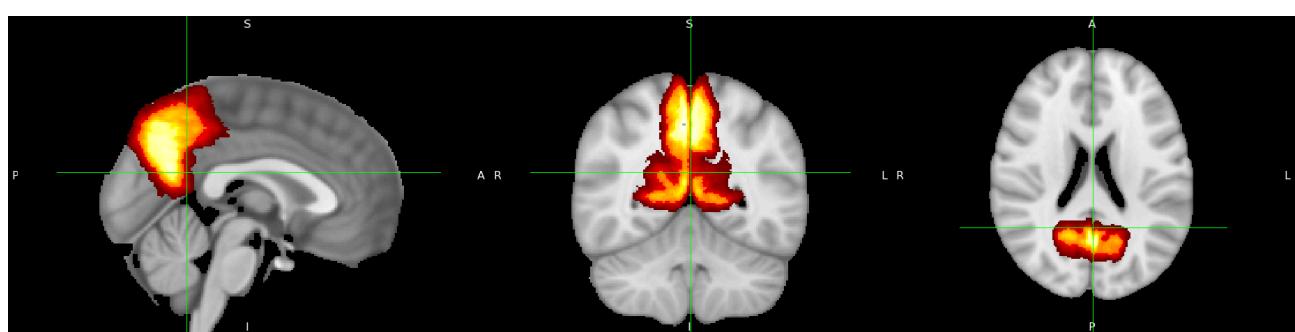
The precuneus is involved in memory tasks, such as when people look at images and try to respond based on what they have remembered in regard to verbal questions about their spatial details.

## 2. Self-Consciousness:

The precuneus is linked to the processes involved in self-consciousness, such as reflective self-awareness, that involve rating one's own personality traits compared to those judged of other people.

## 3. Visuospatial:

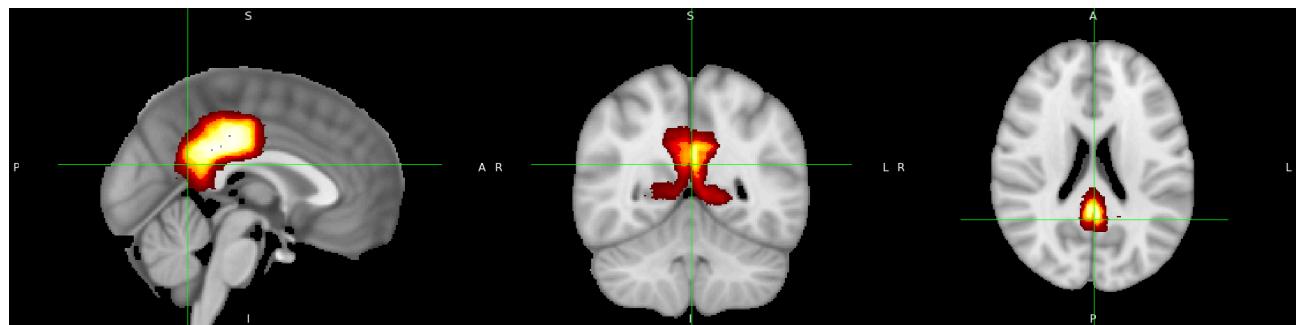
The precuneus has been suggested to be involved in directing attention in space both when an individual makes movements and when imaging or preparing them. It is involved in motor imagery and shifting attention between motor targets.



## II. Cingulate Gyrus

### Functions:

1. Coordinates Sensory Input With Emotions
2. Emotional Responses to pain
3. Regulates Aggressive Behavior
4. Communication
5. Maternal Bonding
6. Language Expression
7. Decision Making



# III. Parietal Lobe

## Functions:

### 1. Self-perception:

Your parietal lobe is a processing center for sensations you can feel with your sense of touch. These include temperature (hot and cold), pressure, vibration and pain.

### 2. Sensory integration

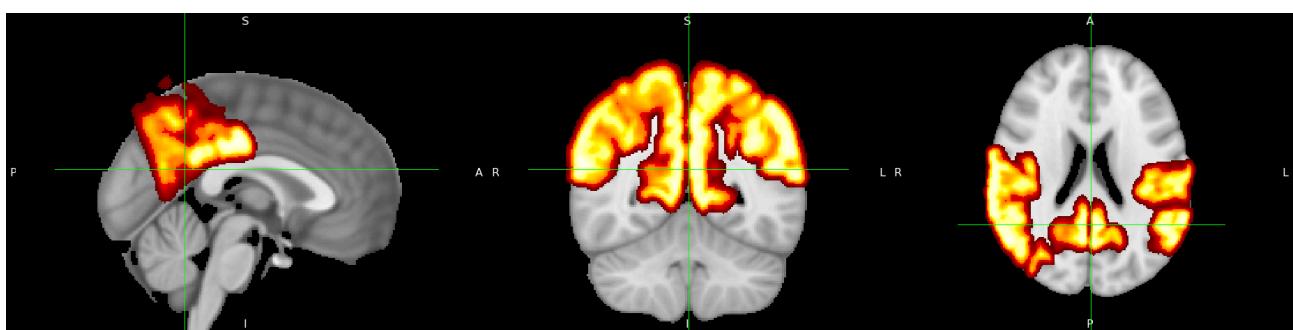
Other brain areas process sensory information they're responsible for, and then forward what they processed to your parietal lobe. Your parietal lobe takes that information, including the self-perception information mentioned above, and integrates it into a form you can understand. It then sends information to other areas of your brain so you can respond (or not respond) to what you sense.

### 3. Location awareness

Your parietal lobe plays a key role in how you understand where things are around you. Knowing if something is on your left or right side, for example, is a function that relies on your parietal lobe.

### 4. Learned movements

Your parietal lobe also helps you learn each time you plan and carry out complex, precise movements.

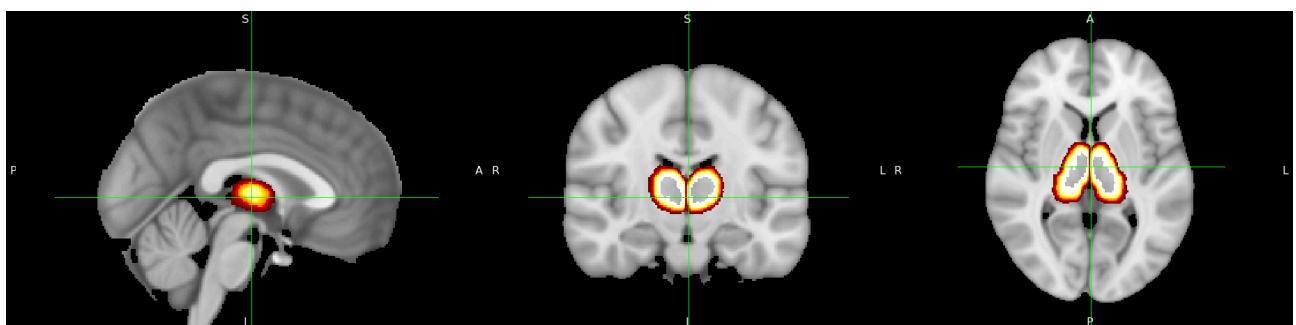


# IV. Thalamus

## Functions:

The thalamus has multiple functions, and is generally believed to act as a relay station, or hub, relaying information between different subcortical areas and the cerebral cortex. The thalamus also plays an important role in regulating states of sleep and wakefulness.

The mediodorsal thalamus (MD) may play a broader role in cognition. Specifically, the mediodorsal thalamus may "amplify the connectivity (signaling strength) of just the circuits in the cortex appropriate for the current context and thereby contribute to the flexibility (of the mammalian brain) to make complex decisions by wiring the many associations on which decisions depend into weakly connected cortical circuits.



Left and Right Thalamus

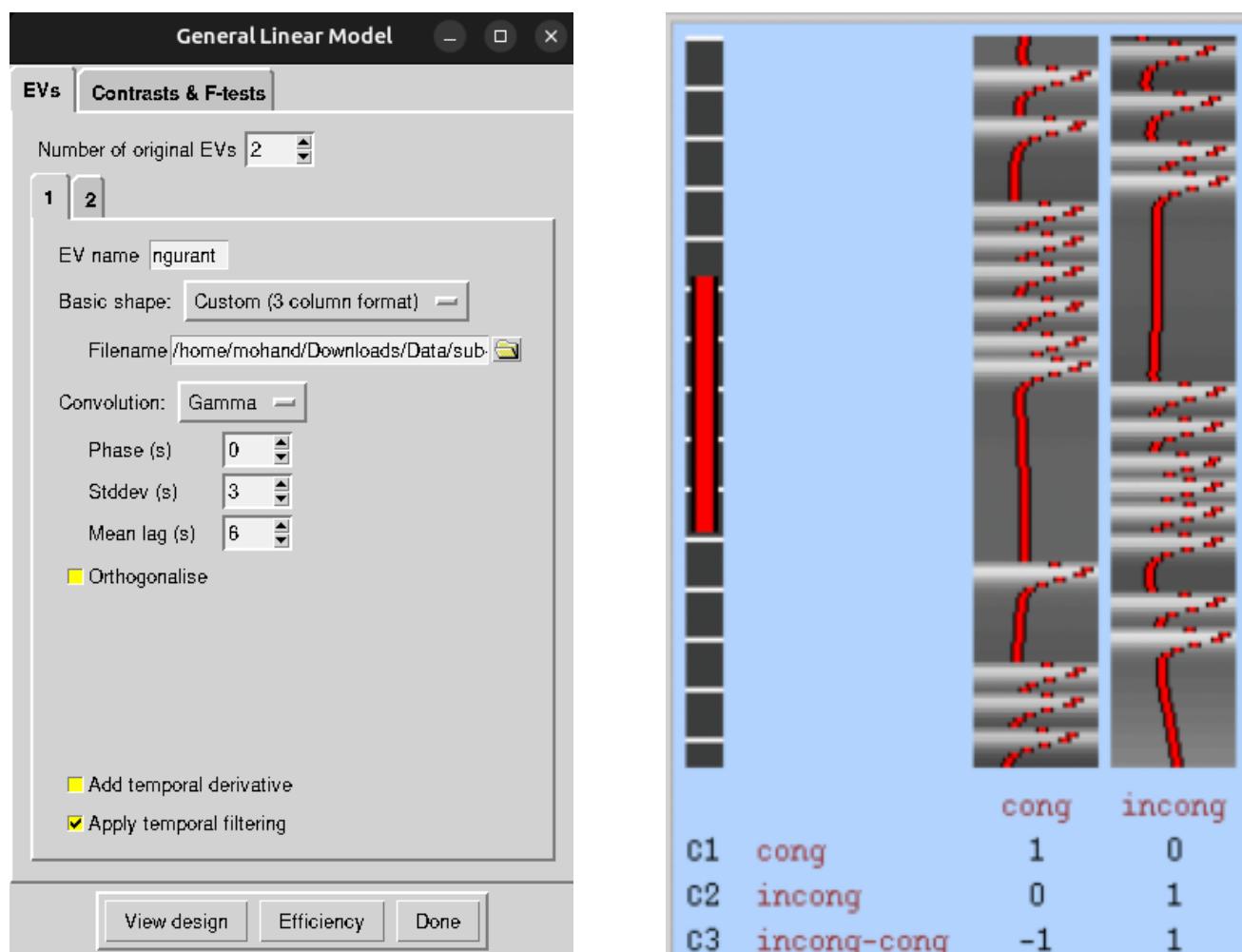
# PART III

Statistical Analysis

# General Linear Model (GLM) analysis:

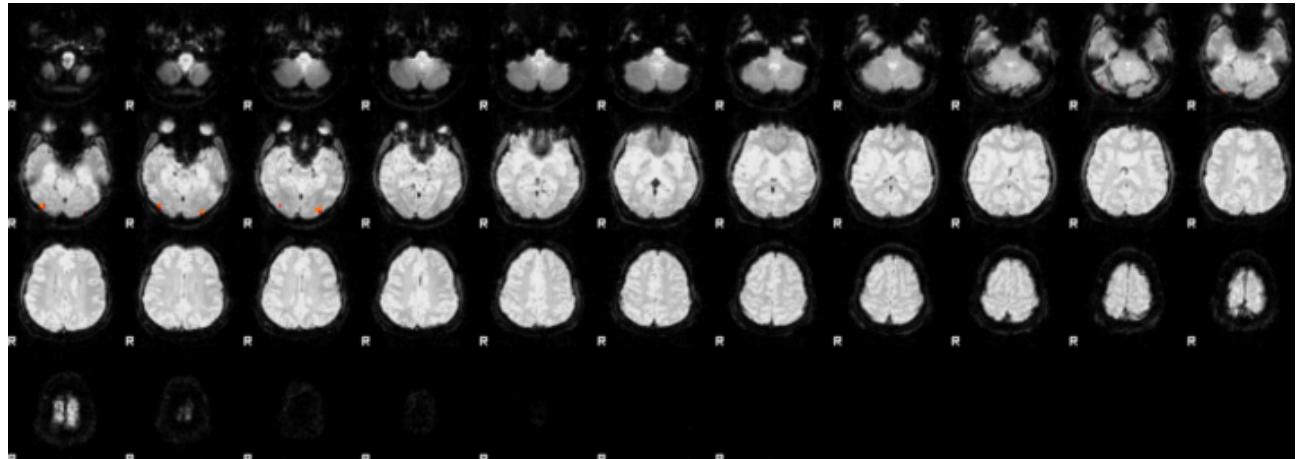
We now come to the General Linear Model, or GLM. With a GLM, we can use one or more regressors, or independent variables, to fit a model to some outcome measure, or dependent variable. To do this we compute numbers called beta weights, which are the relative weights assigned to each regressor to best fit the data. Any discrepancies between the model and the data are called residuals.

## Subject 08 - run 1:

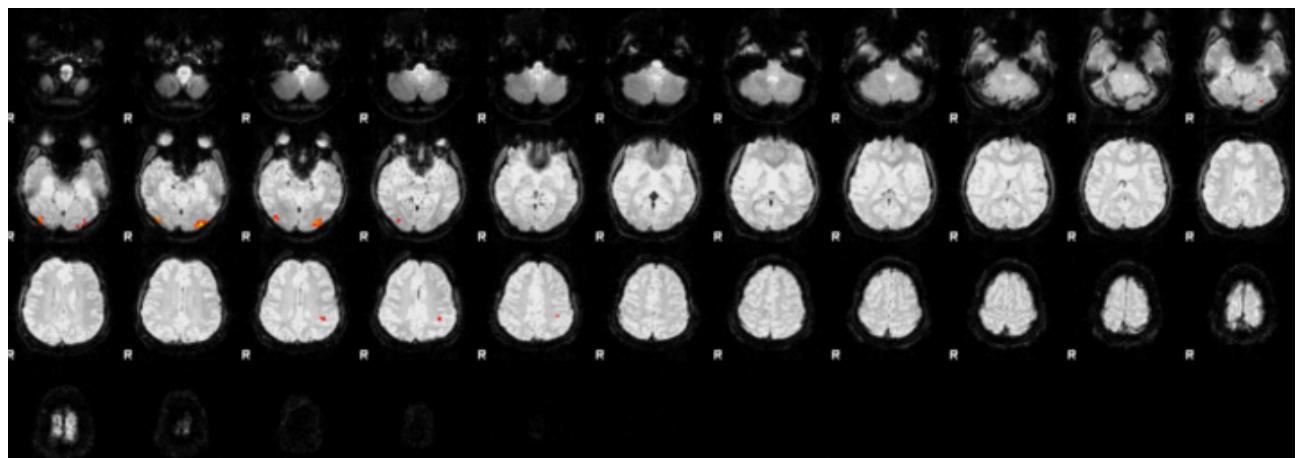


# **Results:**

**Congruent:**

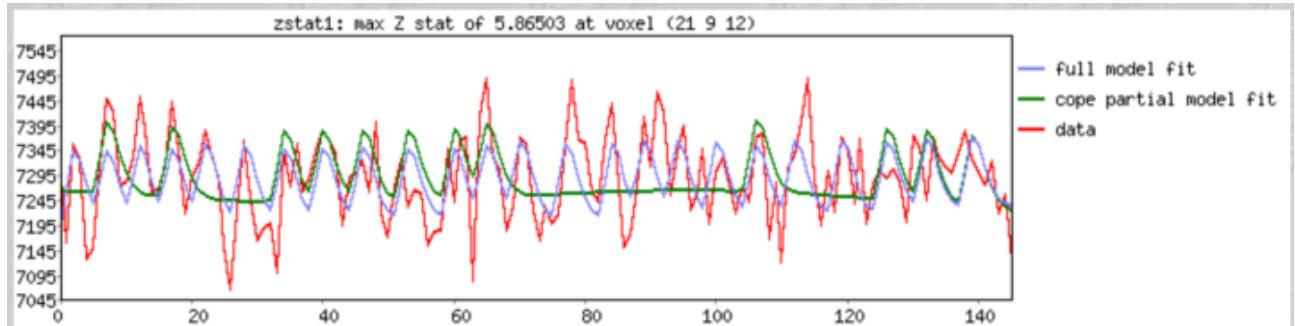


**Incongruent:**

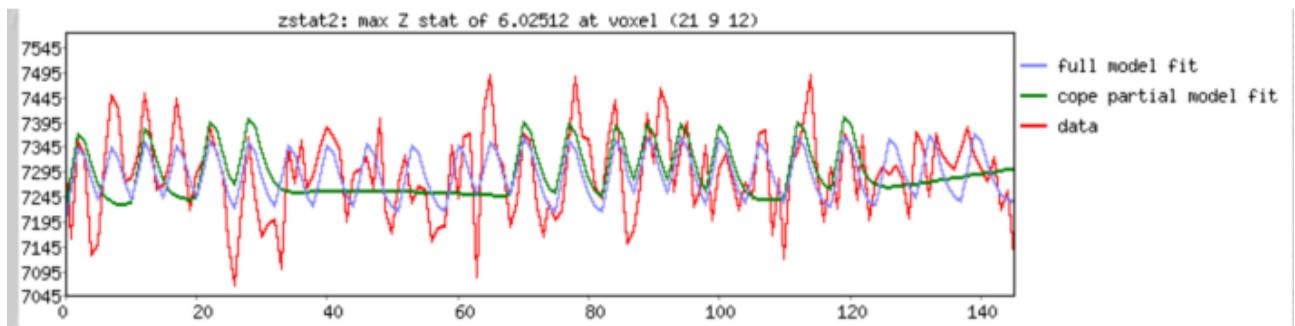


# Time series plots:

## Congruent:



## Incongruent:



# In Conclusion:

- **Purpose of GLM in fMRI Studies:**

The General Linear Model (GLM) is a crucial tool for understanding brain responses during specific tasks or stimuli in fMRI studies. It helps researchers identify active brain regions and draw meaningful conclusions about brain function.

- **Eliminating Interfering Factors:**

By using the GLM, researchers can confidently analyze fMRI data while ruling out potential factors that might interfere with their results.

- **Significance of GLM:**

The GLM allows us to pinpoint which brain regions are active during specific activities, contributing to our understanding of how our brains work.

# PART IV

Scripting 1<sup>st</sup>. Level Analysis

# In Order Run a Script for all the subjects performing 1<sup>st</sup>. Level Analysis:

- We Need a Design File for run1 & run2 using feat\_gui and saving option
- We Need to write script for:
  1. BET - Brain Extraction
  2. Timing Files
  3. Adjusting Design file for every subject
  4. Running Feat command for the new custom design

## 1. BET - Brain Extraction Script:

```
#!/bin/bash

# Generate the subject list to make modifying this script
# to run just a subset of subjects easier.

for id in `seq -w 1 26` ; do
    subj="sub-$id"
    echo "====> Starting processing of $subj"
    echo
    cd $subj

    # If the brain mask doesn't exist, create it
    if [ ! -f anat/${subj}_T1w_brain_f02.nii.gz ]; then
        echo "Skull-stripped brain not found, using bet with a fractional intensity threshold of 0.2"
        # Note: This fractional intensity appears to work well for most of the subjects in the
        # Flanker dataset. You may want to change it if you modify this script for your own study.
        bet2 anat/${subj}_T1w.nii.gz \
            anat/${subj}_T1w_brain_f02.nii.gz -f 0.2
    fi

    # Go back to the directory containing all of the subjects, and repeat the loop
    cd ..
done

echo
```

Bash script to check for skullstripping and perform it  
if it does not exist

## 2. Timing Files:

```
#!/bin/bash

#Check whether the file subjList.txt exists; if not, create it
if [ ! -f subjList.txt ]; then
    ls -d sub-?? > subjList.txt
fi

#Loop over all subjects and format timing files into FSL format
for subj in `cat subjList.txt` ; do
    cd $subj/func #Navigate to the subject's func directory, which contains the timing files

    #Extract the onset times for the incongruent and congruent trials for each run. NOTE: This script only extracts the trials in which the subject made a correct response.
    Accuracy is nearly 100% for all subjects, but as an exercise the student can modify this to extract the incorrect trials as well.
    cat ${subj}_task-flanker_run-1_events.tsv | awk '{if ($3=="incongruent_correct") {print $1, $2, "1"}}' > incongruent_run1.txt
    cat ${subj}_task-flanker_run-1_events.tsv | awk '{if ($3=="congruent_correct") {print $1, $2, "1"}}' > congruent_run1.txt

    cat ${subj}_task-flanker_run-2_events.tsv | awk '{if ($3=="incongruent_correct") {print $1, $2, "1"}}' > incongruent_run2.txt
    cat ${subj}_task-flanker_run-2_events.tsv | awk '{if ($3=="congruent_correct") {print $1, $2, "1"}}' > congruent_run2.txt

    cd ../..
done
```

## 3. Adjusting Design file for every subject:

By replacing sub-01 by \${subj} we can create a custom design file for the feat of each subj from subj-01 to subj-26

```
for id in `seq -w 1 26` ; do
    subj="sub-$id"
    echo "====> Starting processing of $subj"
    echo
    cd $subj

    # Copy the design files into the subject directory, and then
    # change "sub-08" to the current subject number
    cp ..../design_run1.fsf .
    cp ..../design_run2.fsf .

    # Note that we are using the | character to delimit the patterns
    # instead of the usual / character because there are / characters
    # in the pattern.
    sed -i -e "s|sub-01|${subj}|g" \
        design_run1.fsf
    sed -i -e "s|sub-01|${subj}|g" \
        design_run2.fsf

    # Go back to the directory containing all of the subjects, and repeat the loop
    cd ..
done

echo
```

4. Now, What is left for us is to run those files for each run and store the runs in the same dir as the subj by running the following bash script:

```
#!/bin/bash

# Generate the subject list to make modifying this script
# to run just a subset of subjects easier.

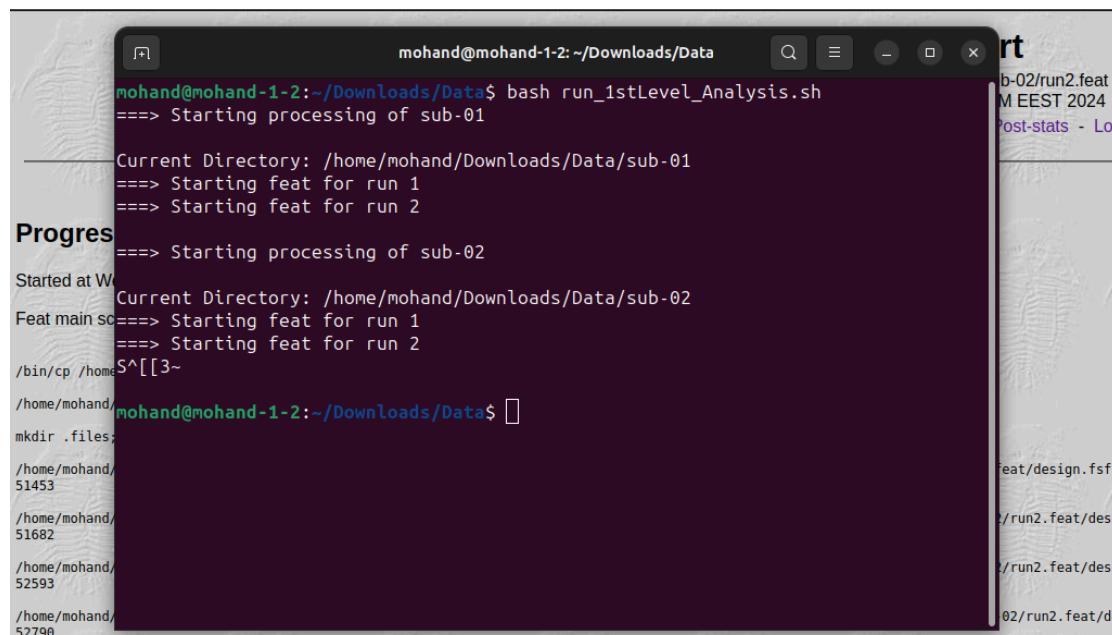
for id in `seq -w 1 2` ; do
    subj="sub-0$id"
    echo "====> Starting processing of $subj"
    echo
    cd $subj
    echo "Current Directory: $(pwd)"

    # Now everything is set up to run feat
    echo "====> Starting feat for run 1"
    feat design_run1.fsf
    echo "====> Starting feat for run 2"
    feat design_run2.fsf
    echo

    # Go back to the directory containing all of the subjects, and repeat the loop
    cd ..
done

echo
```

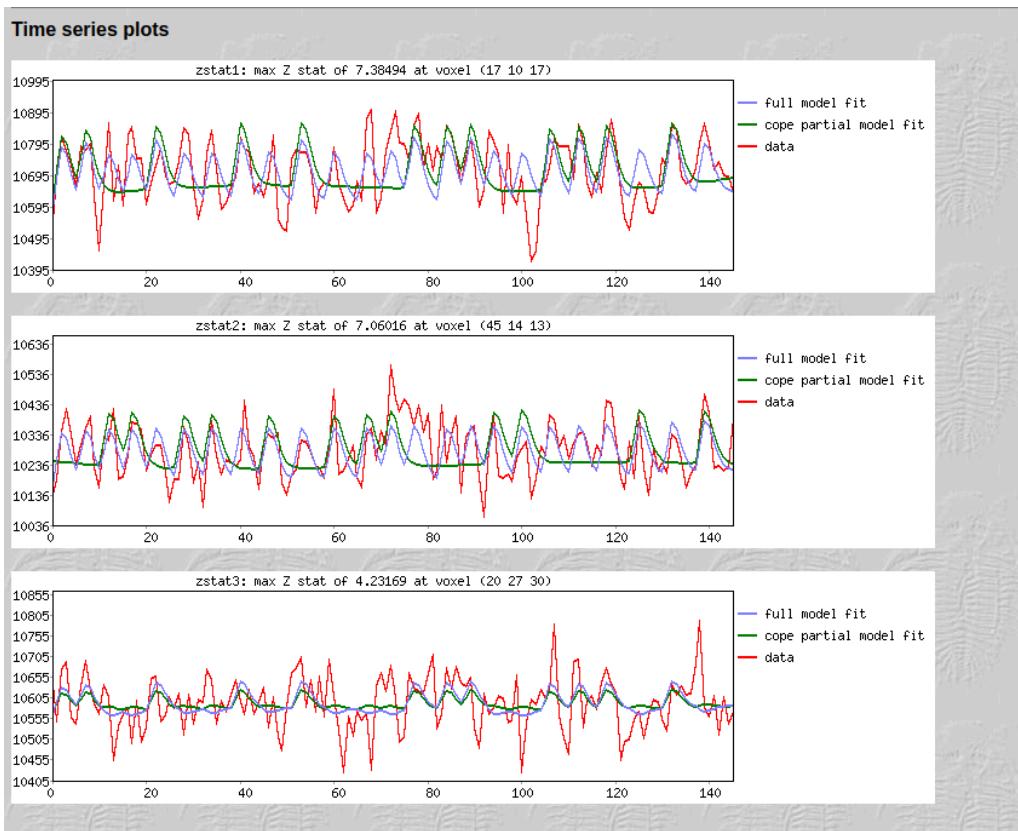
That will loop over each subj-?? & start the feat for run1 and run2 respectively until it loops over every subject into the Flanker's dataset.



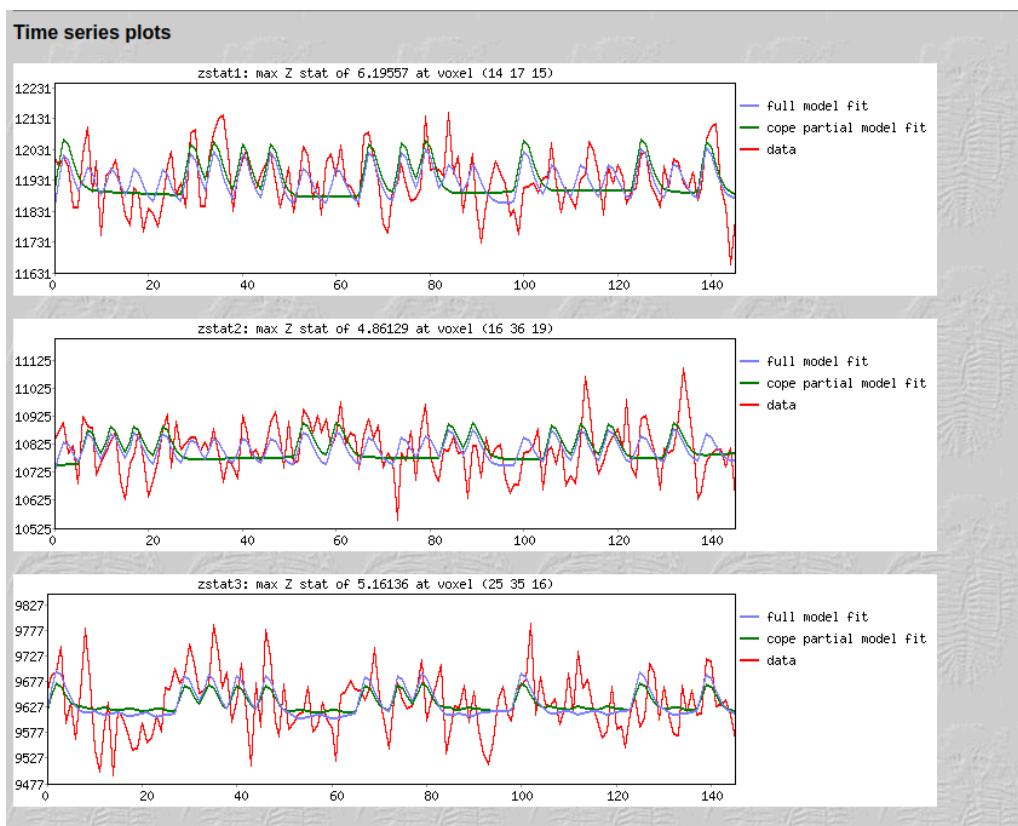
```
mohand@mohand-1-2:~/Downloads/Data$ bash run_1stLevel_Analysis.sh
====> Starting processing of sub-01
-----
Current Directory: /home/mohand/Downloads/Data/sub-01
====> Starting feat for run 1
====> Starting feat for run 2
Progress
====> Starting processing of sub-02
Started at W
Feat main sc
Current Directory: /home/mohand/Downloads/Data/sub-02
====> Starting feat for run 1
====> Starting feat for run 2
/bin/cp /home/mohand/Downloads/Data/sub-02/feat/design.fsf /home/mohand/Downloads/Data/sub-02/feat/run2.feat/design.fsf
mohand@mohand-1-2:~/Downloads/Data$ [3~]
mkdir .files;
/home/mohand/51453
/home/mohand/51682
/home/mohand/52593
/home/mohand/52790
```

# Post-Stat Time Series Examples:

## 1. Subj-01 - run1:



## 2. Subj-01 - run2:



### 3. Subj-02 - run1:



### 4. Subj-02 - run2:



# PART V

2<sup>nd</sup>. Level Analysis - Averaging the 2 runs of every  
subject

# Because of my Laptop I wasn't able to run all dataset so,

Here's how to setup model to take average of 2 runs for 9 subjects:

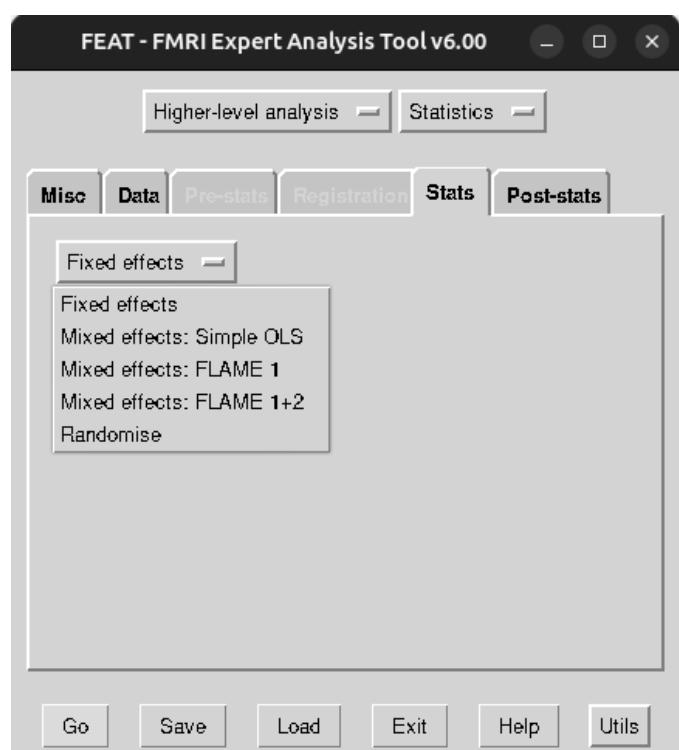
1. Using the current Bash Command we can have a list of all the .feat files in our dataset: `ls -d $PWD/sub-??/run*`



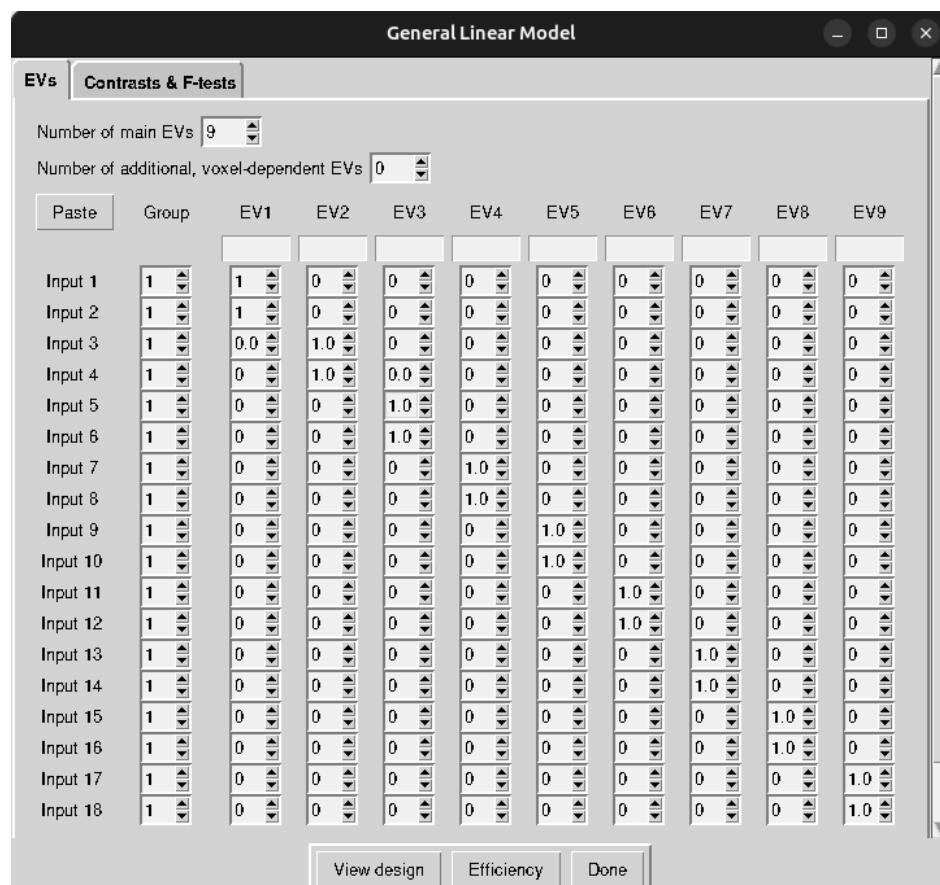
```
mohand@mohand-1-2:~/Downloads/Data$ ls -d $PWD/sub-??/run*
/home/mohand/Downloads/Data/sub-01/run1.feat
/home/mohand/Downloads/Data/sub-01/run2.feat
/home/mohand/Downloads/Data/sub-02/run1.feat
/home/mohand/Downloads/Data/sub-02/run2.feat
/home/mohand/Downloads/Data/sub-03/run1.feat
/home/mohand/Downloads/Data/sub-03/run2.feat
/home/mohand/Downloads/Data/sub-04/run1.feat
/home/mohand/Downloads/Data/sub-04/run2.feat
/home/mohand/Downloads/Data/sub-05/run1.feat
/home/mohand/Downloads/Data/sub-05/run2.feat
/home/mohand/Downloads/Data/sub-06/run1.feat
/home/mohand/Downloads/Data/sub-06/run2.feat
/home/mohand/Downloads/Data/sub-07/run1.feat
/home/mohand/Downloads/Data/sub-07/run2.feat
/home/mohand/Downloads/Data/sub-08/run1.feat
/home/mohand/Downloads/Data/sub-08/run2.feat
/home/mohand/Downloads/Data/sub-09/run1.feat
/home/mohand/Downloads/Data/sub-09/run2.feat
mohand@mohand-1-2:~/Downloads/Data$
```

2. After pasting the directories for our .feat we setup the model to take avg of the 2 runs of every .feat using:

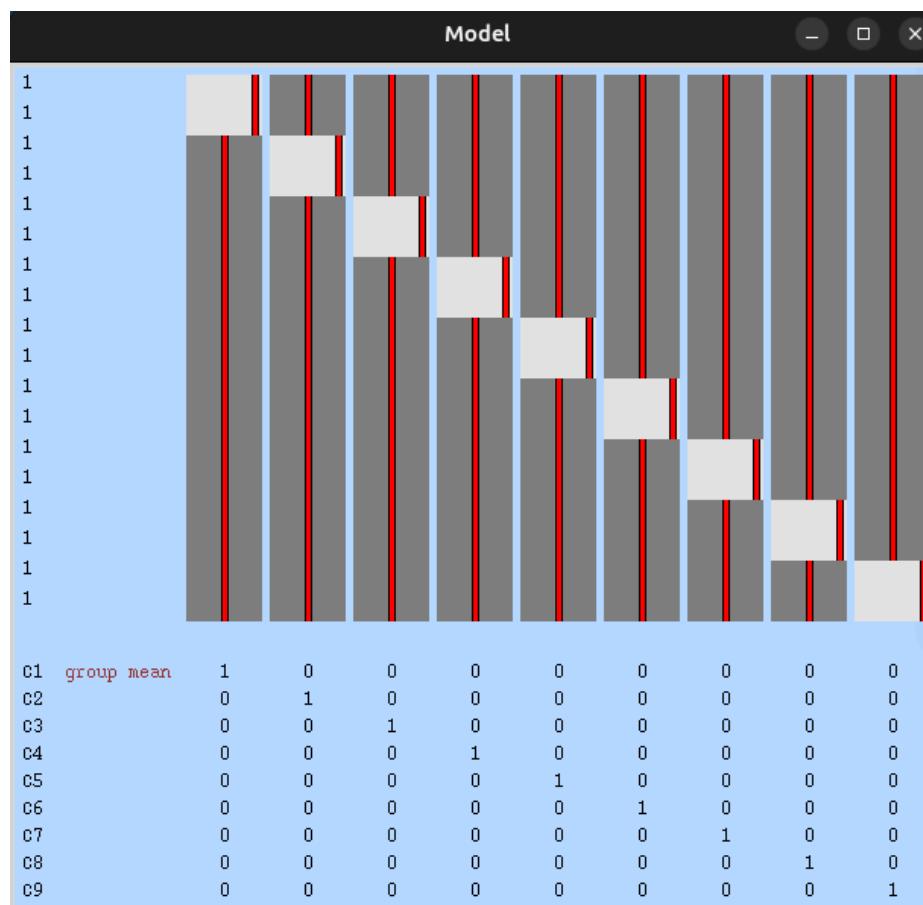
**Fixed Effects:** Do not generalize from the sample - just take the average



3. Here's how to setup GLM model to take average of 2 runs for 9 subjects:

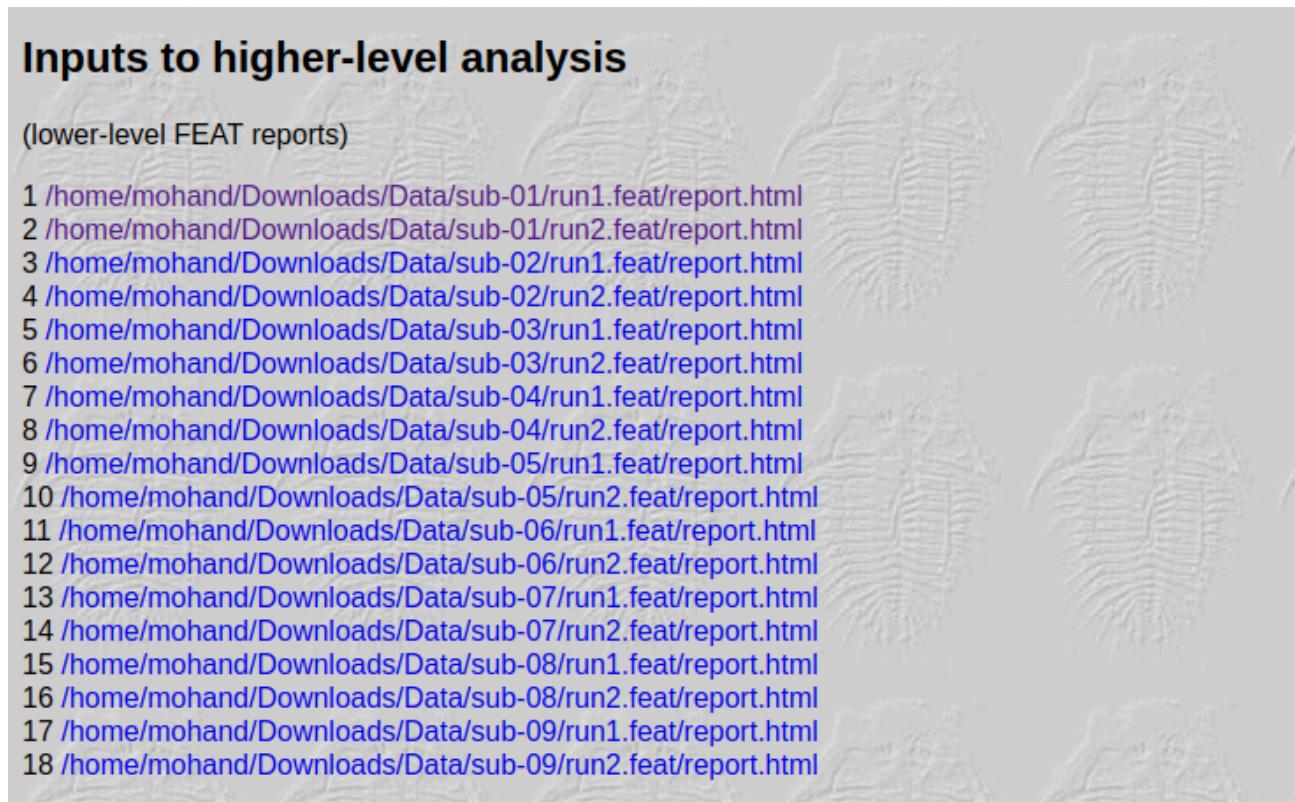


4. After setting up the contrasts we can see the following:

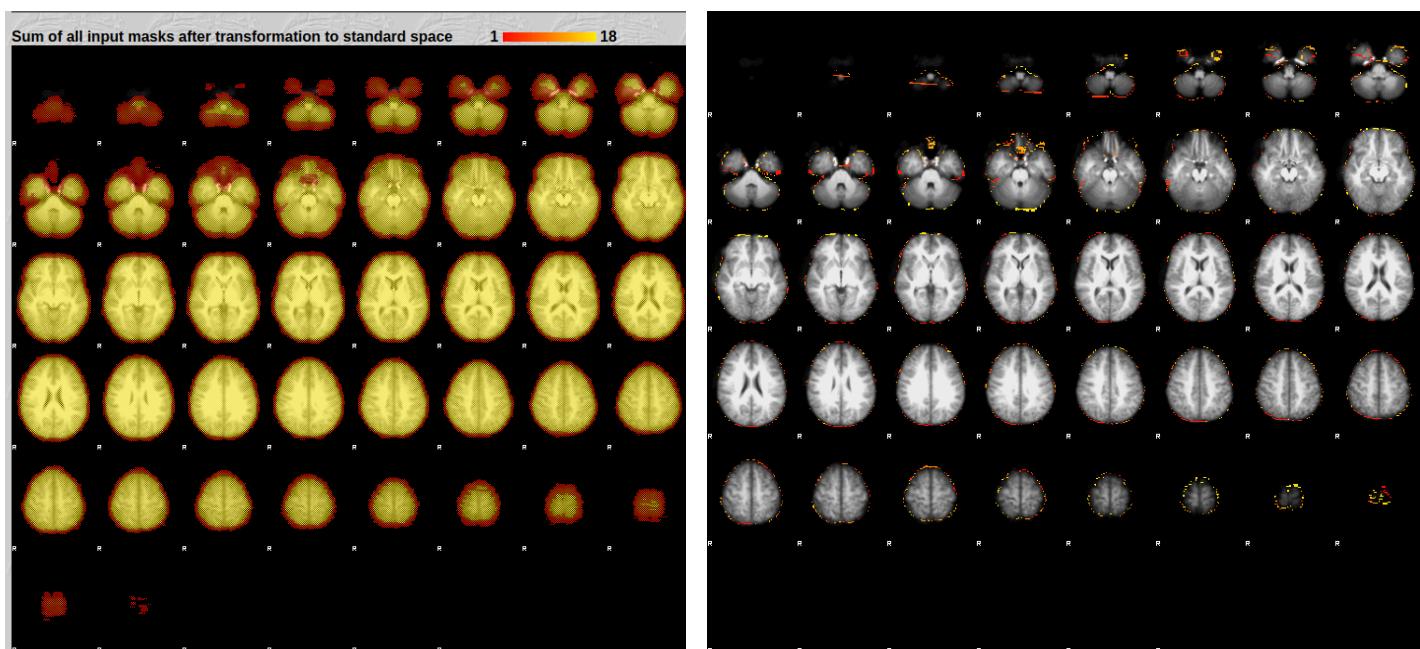


# The FSL REPORT SUMMARY:

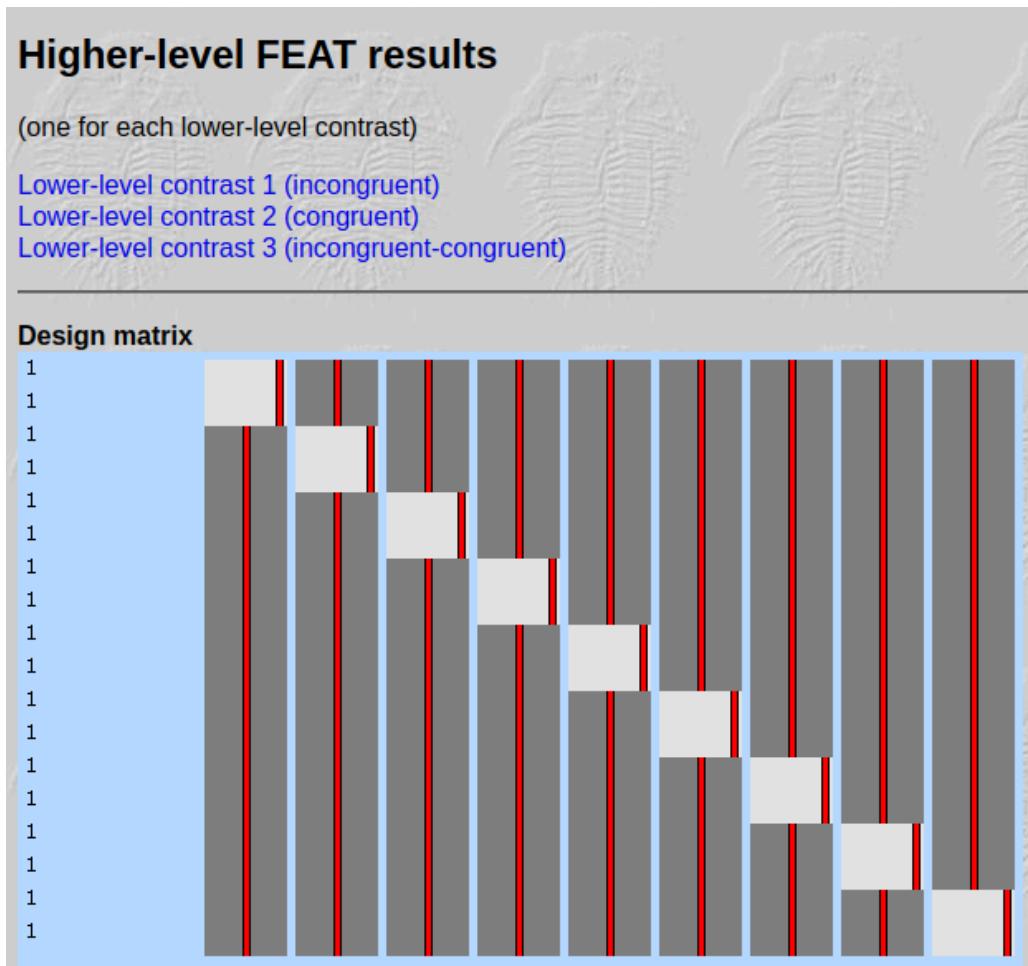
- Inputs:



- Registration Summary:



- Results:



## General Notes on the 2nd. Level Analysis:

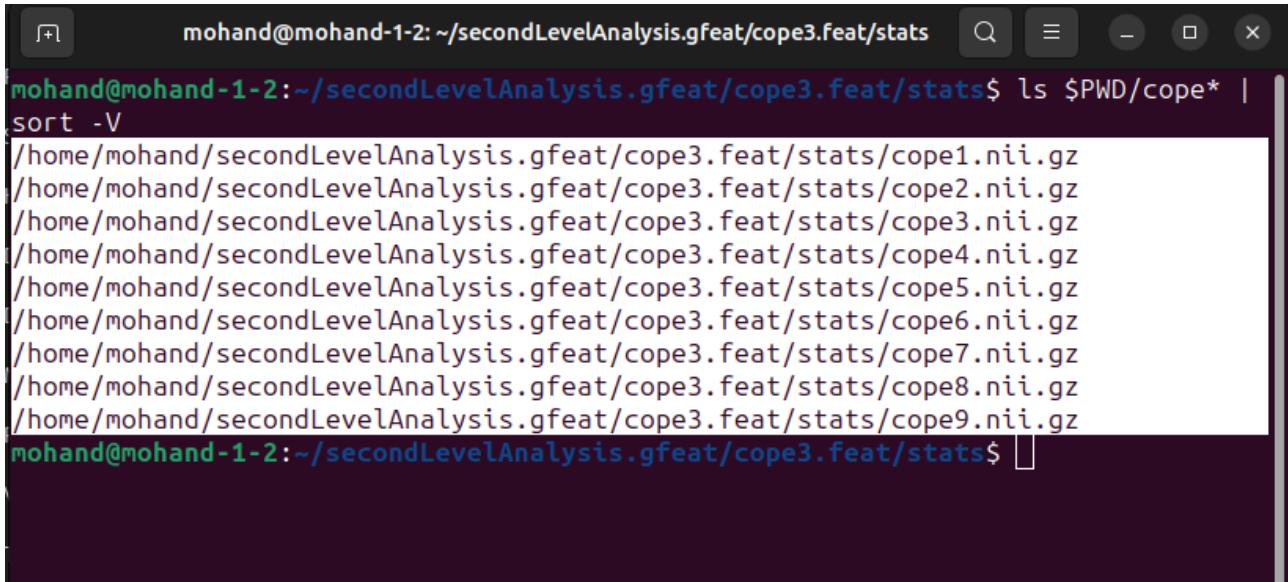
We are interested in averaging the run1 and run2 from each subject so we can have one file that represents the functional image of the subject so, in further steps we can use this average for group level Analysis where we can conclude outcomes based on the whole dataset.

# PART VI

3<sup>rd.</sup> Level Analysis - Group Analysis

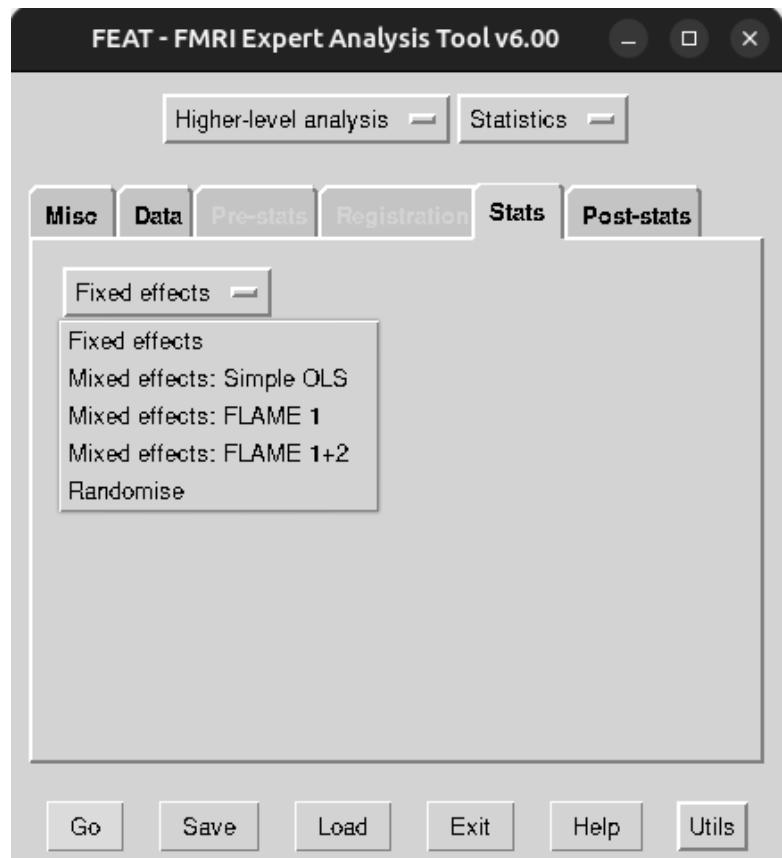
In FSL, a *3rd-level analysis* is a *group-level analysis* - we calculate the standard error and the mean for a contrast estimate, and then test whether the average estimate is statistically significant.

1. Using the current Bash Command we can have a list of all the .cope files (sorted) in our dataset: `ls $PWD/cope* | sort -V`



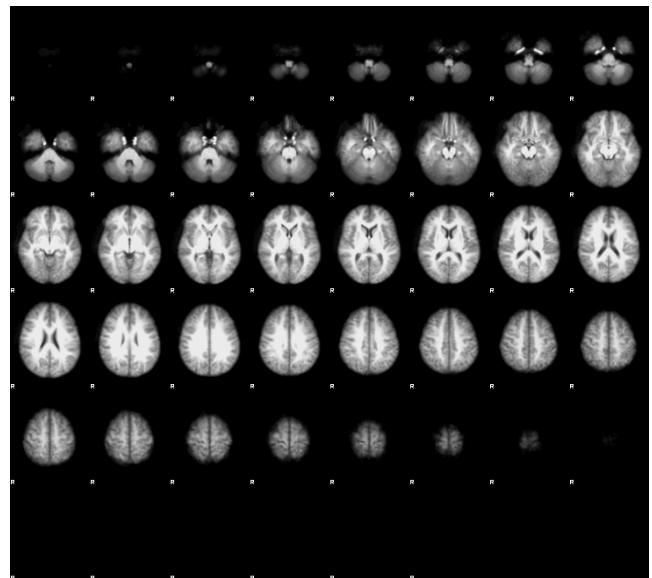
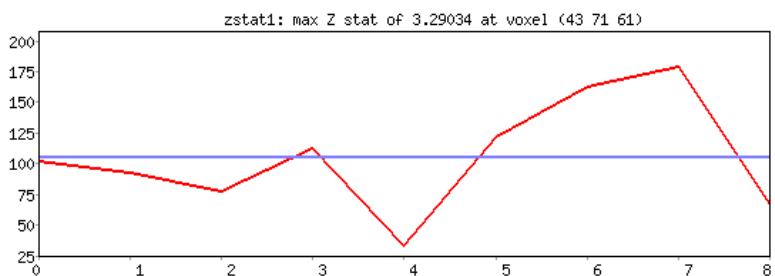
```
mohand@mohand-1-2:~/secondLevelAnalysis.gfeat/cope3.feat/stats$ ls $PWD/cope* | sort -V
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope1.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope2.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope3.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope4.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope5.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope6.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope7.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope8.nii.gz
/home/mohand/secondLevelAnalysis.gfeat/cope3.feat/stats/cope9.nii.gz
mohand@mohand-1-2:~/secondLevelAnalysis.gfeat/cope3.feat/stats$
```

2. For a 3rd-level analysis, we will use Mixed Effects. **FLAME 1** (FSL's Local Analysis of Mixed Effects) provides accurate parameter estimates by using information about both within-subject and between-subject variability.

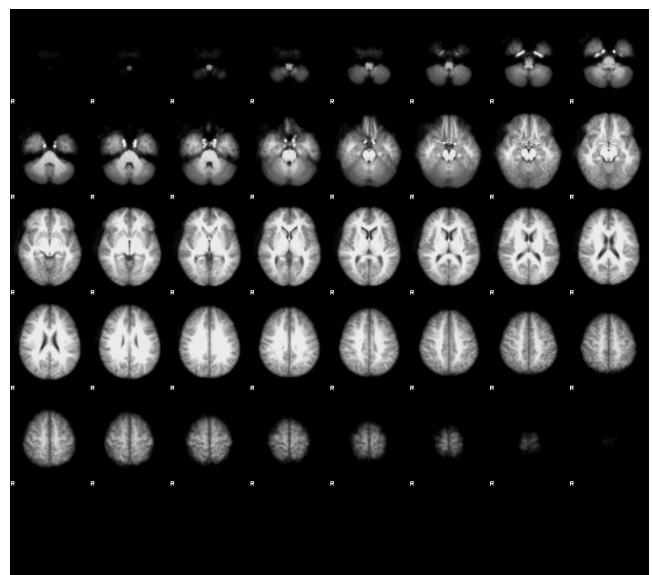
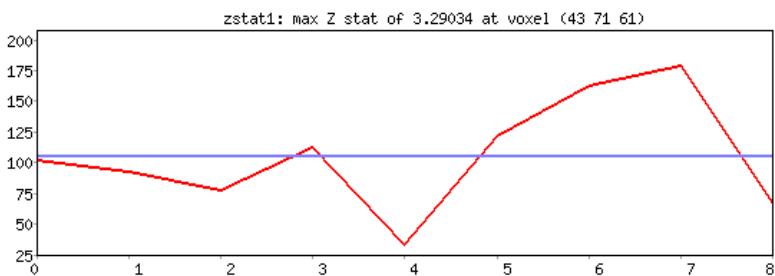


### 3. Using Different Post Statistical Thresholding and Results:

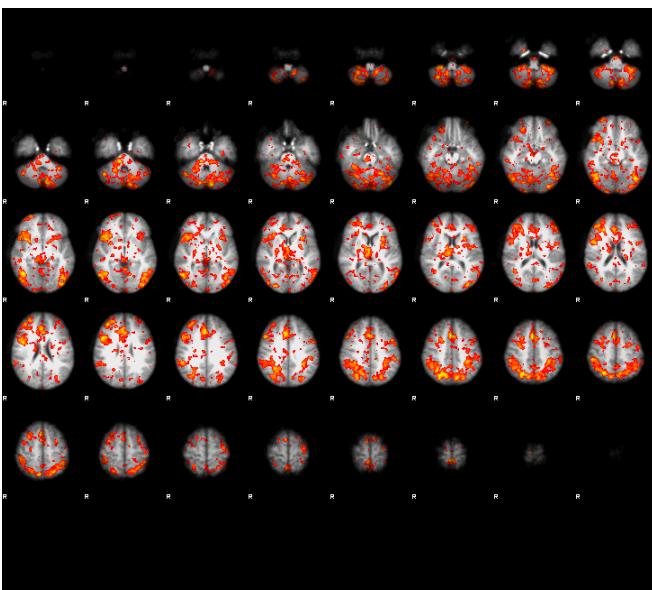
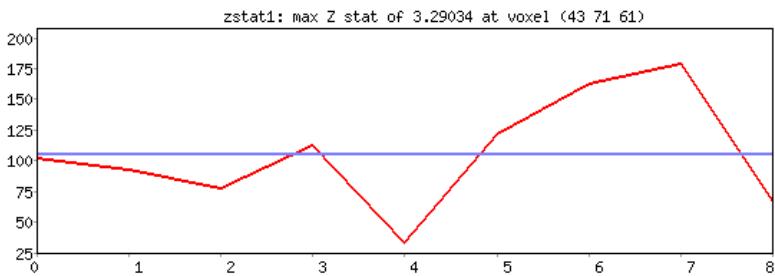
#### i. Cluster ( $Z = 3.1$ , $P = 0.05$ )



#### ii. Voxel ( $P = 0.05$ )



#### iii. Uncorrected ( $P = 0.05$ ):

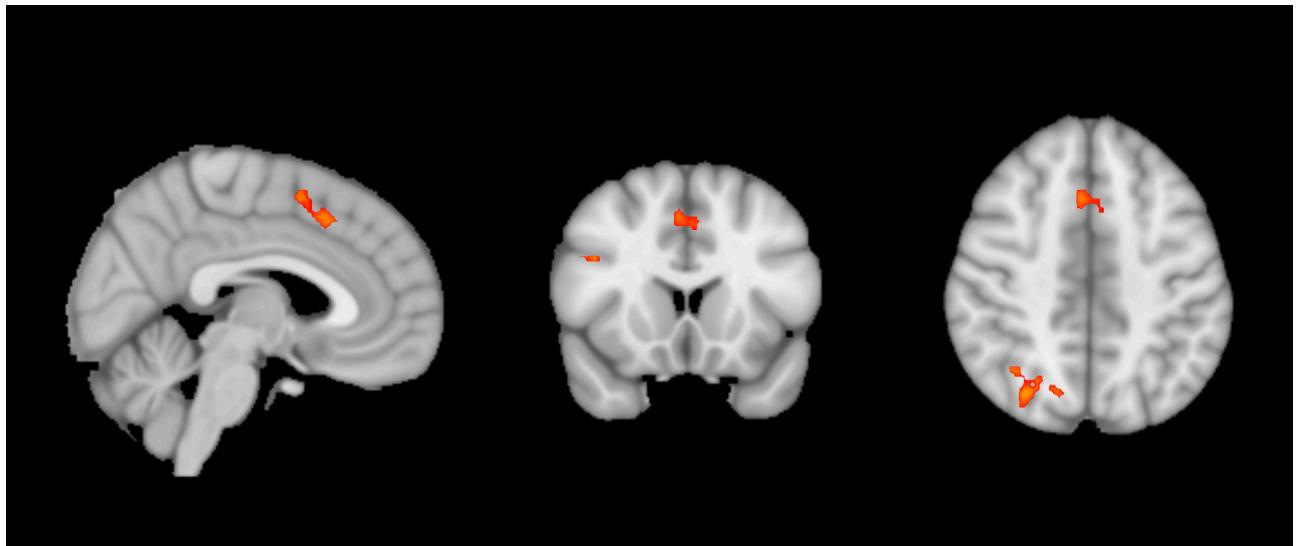


**iv. None Threshold:**

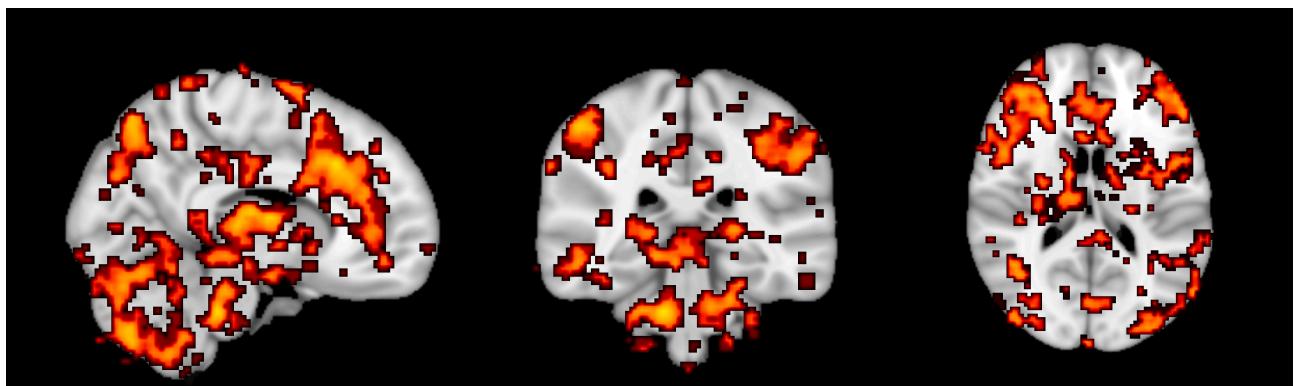
*Using this option for post stats in third level analysis result into an empty tab into the report.html; Hence, no post stats result to be shown*

4. Using FSLEyes to display some of the outcomes:

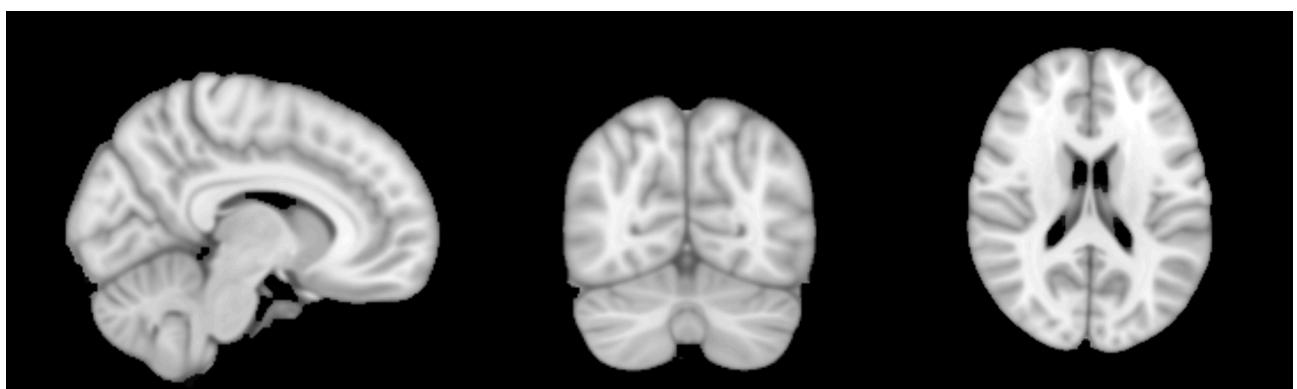
i. Cluster Threshold:



ii. Uncorrected Threshold



iii. Voxel Threshold



## **Summary:**

Based on the FSLeyes output we can notice that the uncorrected Threshold isn't as concise as the cluster or the voxel thresholds because it shows a larger area of interest than both of them.

*However, using only 9-subjects might has its say into the accuracy of these third level analysis but, in general Cluster and Voxel are more selective for the areas determined by the third level analysis, None and uncorrected show a no post-stats results and a very not concise area of interest respectively.*

---

## **General Notes on the 3rd. Level Analysis:**

In this analysis we are interested in analyzing the dataset to generalize the results to the population that the sample was drawn from. In other words, if we see changes in brain activity in our sample, can we say that these changes would likely be seen in the population as well?

To test this, we will run a **3rd-level analysis**. In FSL, a 3rd-level analysis is a group-level analysis - we calculate the standard error and the mean for a contrast estimate, and then test whether the average estimate is statistically significant.

# PART VII

ROI Analysis - Region Of Interest Analysis

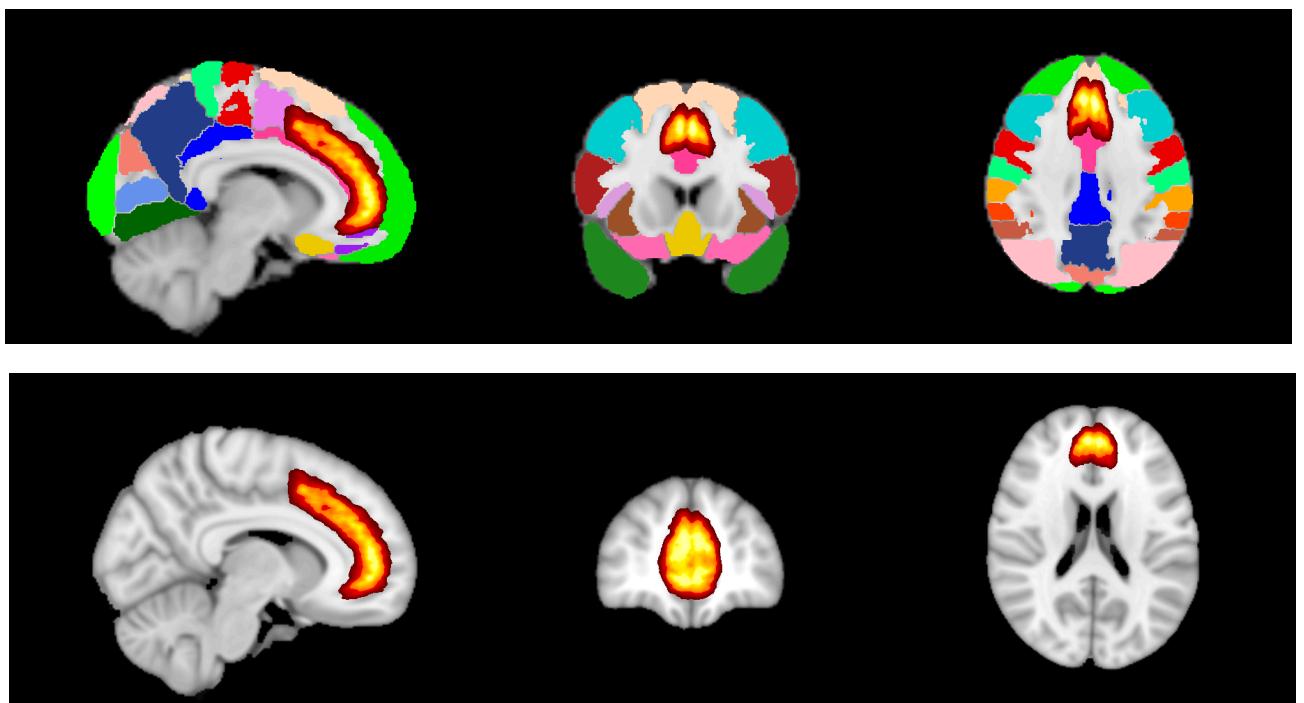
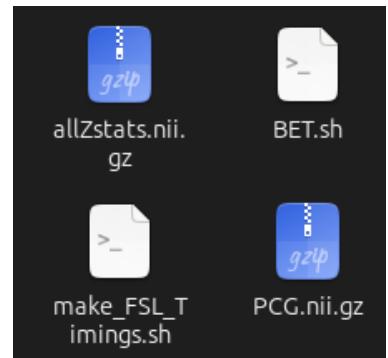
In ROI Analysis, we either have an atlas-based (anatomical) mask or a spherical mask

For an Atlas Mask:

1. To make our ROI analysis easier, we will merge all of the z-statistic maps into a single dataset using the following script:

```
fslmerge -t allZstats.nii.gz `ls  
zstat* | sort -V`
```

2. Create a mask using the atlas and standard MNI for the Paracingulate Gyrus PCG:



3. Then use the *fslmeans* command to extract the data from the PCG mask: `fslmeans -i allZstats.nii.gz -m PCG.nii.gz`

```
mohand@mohand-1-2:~/Downloads/Data$ fslmeans -i allZstats.nii.gz -m PCG.nii.gz  
1.363031  
0.222658  
1.062293  
-0.211900  
-0.101923  
0.207993  
0.720698  
0.701031  
0.265106
```

#### 4. Analyzing the numbers using R Studio

For a Spherical Mask:

1. We take MNI coordinates (x,y,z) and return a Voxel coordinates (x',y',z') using FSLEYES

#### For The Exercise Example: Ex.2

Use the code given in the section on spherical ROI analysis to create a sphere with a 7mm radius located at MNI coordinates 36, -2, 48

Its Voxel Coordinates will be 27, 62, 60

Coordinates: MNI152		Voxel location
36		27
-2		62
48		60
Volume		0

2. We use this script to create a sphere with some coordinates and a radius:

1. Create a point using voxels coordinates:

```
fslmaths  
$FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul  
0 -add 1 -roi $x' 1 $y' 1 $z' 1 0 1  
${File_name}_$x_$y_$z}.nii.gz -odt float
```

2. Create a sphere with radius (mm) out of this point

```
fslmaths ${File_name}_$x_$y_$z}.nii.gz -kernel  
sphere $radius -fmean  
${File_name}_sphere_$x_$y_$z}.nii.gz -odt float
```

3. Binarize the sphere

```
fslmaths ${File_name}_sphere_$x_$y_$z}.nii.gz -bin  
${File_name}_sphere_bin_$x_$y_$z}.nii.gz
```

## For The Exercise Example: Ex. 2

Use the code given in the section on spherical ROI analysis to create a sphere with a 7mm radius located at MNI coordinates 36, -2, 48  
Knowing that its voxel coordinates are 27, 62,60

1. Create a point using voxels coordinates:

```
fslmaths
```

```
$FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul  
0 -add 1 -roi 27 1 62 1 60 1 0 1  
ROI_36_-2_48.nii.gz -odt float
```

2. Create a sphere with radius (mm) out of this point

```
fslmaths ROI_36_-2_48.nii.gz -kernel sphere 7  
-fmean ROI_Sphere_36_-2_48.nii.gz -odt float
```

3. Binarize the sphere

```
fslmaths ROI_Sphere_36_-2_48.nii.gz -bin  
ROI_Sphere_bin_36_-2_48.nii.gz
```



3. Lastly, we will extract data from this ROI by typing:

```
fslmeants -i allZstats.nii.gz -m  
${File_name}_${x}_${y}_${z}.nii.gz
```

### For our example:

```
fslmeants -i allZstats.nii.gz -m  
ROI_Sphere_bin_36_-2_48.nii.gz
```

```
mohand@mohand-1-2:~/Downloads/Data$ fslmeants -i allZstats.nii.gz -m ROI_Sphere_
bin_36_-2_48.nii.gz
0.997455
-0.131412
1.430174
0.317560
0.001948
1.059744
0.957232
1.633888
0.169669
```

Finally using R we get our p-value:

Data	
⌚ t_test_result_new	List of 10
Values	
new_data	num [1:9] 0.99745 -0.13141 1.43017 0.31756 0.0...
null_value	0
p_value_t_test_n...	0.0101736784153075

## Ex. 1:

If we used `fslmeants -i allZstats.nii.gz -m PCG.nii.gz -w` instead:

```
mohand@mohand-1-2:~/Downloads/Data$ fslmeants -i allZstats.nii.gz -m PCG.nii.gz
-w
1.535634
0.410007
1.062549
-0.168919
0.051023
0.166882
0.865320
0.916935
0.242831

mohand@mohand-1-2:~/Downloads/Data$ fslmeants -i allZstats.nii.gz -m PCG.nii.gz
1.363031
0.222658
1.062293
-0.211900
-0.101923
0.207993
0.720698
0.701031
0.265106
```

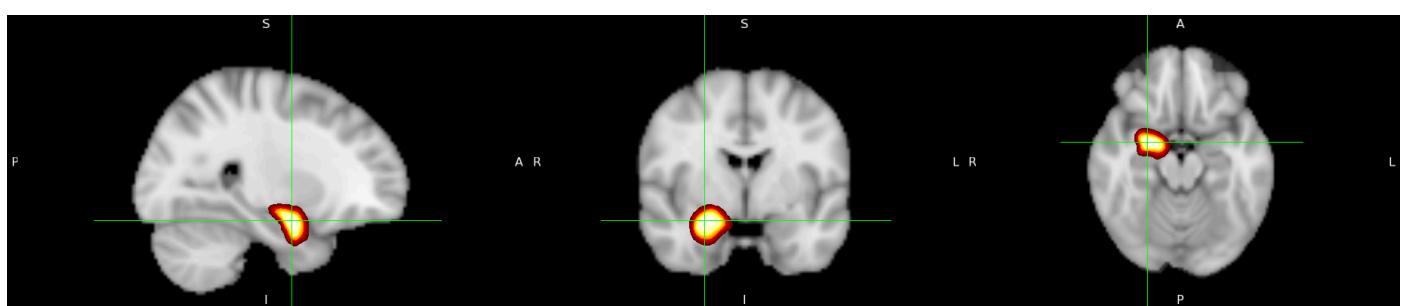
As we can observe the values are higher now for almost every subject of the 9 subjects

Data	
⌚ t_test_result	List of 10
⌚ t_test_result_new	List of 10
Values	
data	num [1:9] 1.536 0.41 1.063 -0.169 0.051 ...
new_data	num [1:9] 1.363 0.223 1.062 -0.212 -0.102 ...
null_value	0
p_value_t_test	0.0161904531904385
p_value_t_test_new	0.02832192555042

Also the p-value is Lower than before using R

### Ex. 3:

1. To make our ROI analysis easier, we will merge all of the z-statistic from .copel maps into a single dataset using the following script:  
`fslmerge -t allZstats.nii.gz `ls zstat* | sort -V``
2. Create a mask using the atlas and standard MNI for the Right Amygdala R\_AMGD:



3. Then use the *fslmeans* command to extract the data from the R\_AMGD mask: `fslmeans -i allZstats.nii.gz -m R_AMGD.nii.gz`

```
mohand@mohand-1-2:~/Downloads/Data$ fslmeants -i allZstats.nii.gz -m R_AMGD.nii.gz
0.536970
0.883927
-0.350269
1.258577
0.707172
-0.271934
0.326566
0.357452
0.437142
```

4. Using R we can calculate the p-test

Data	
● t_test_result	List of 10
Values	
data	num [1:9] 0.537 0.884 -0.35 1.259 0.707 ...
p_value_t_test	0.0353129524305823

## General Notes on the ROI Analysis:

Whole-brain maps can hide important details about the effects that we're studying. We may find a significant effect of incongruent-congruent, but the reason the effect is significant could be because incongruent is greater than congruent, or because congruent is much more negative than congruent, or some combination of the two.

The only way to determine what is driving the effect is with **ROI analysis**, and this is especially important when dealing with interactions and more sophisticated designs.