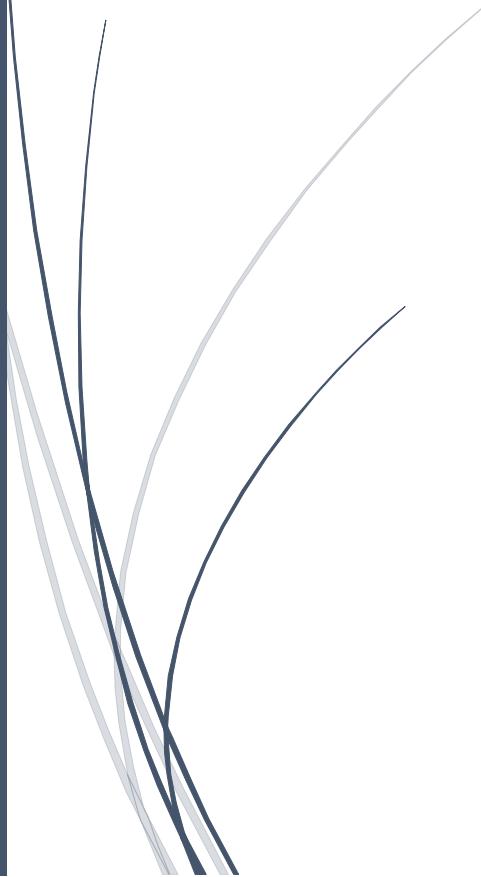




Advanced Topics in Biomedical Informatics

Flanker Dataset Report

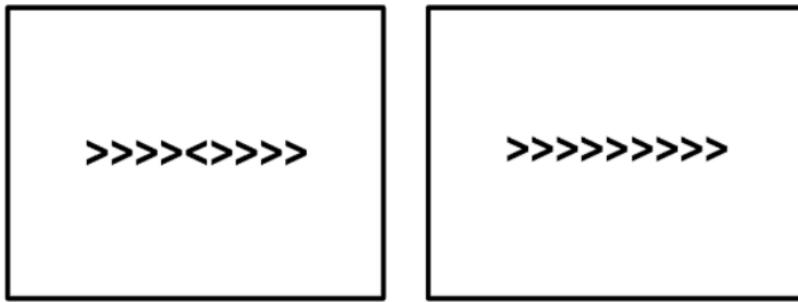
Meena M. Makary, Ph.D.



Mayar Fayez

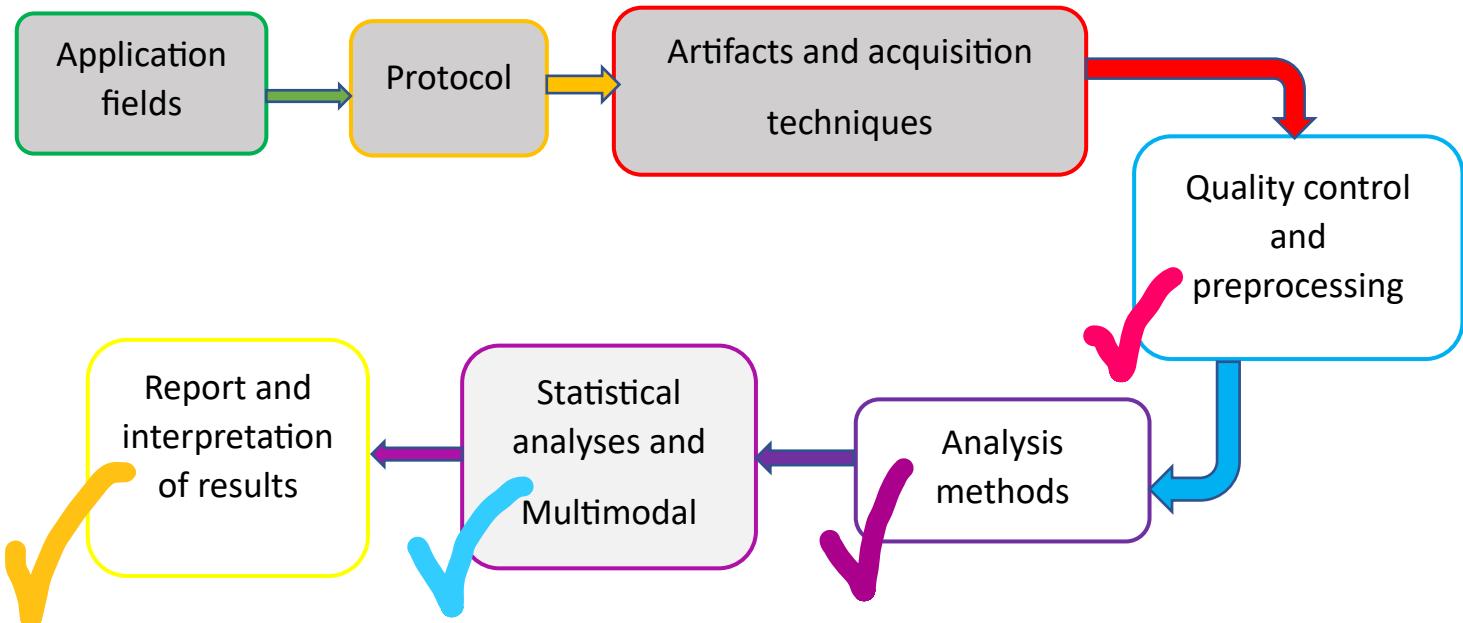
The Flanker Task

In the Flanker task, arrows point either to the left or the right, and the subject is instructed to press one of two buttons indicating the direction of the arrow in the middle. If it's pointing to the left, the subject presses the "left" button; if it's pointing to the right, the subject presses the "right" button. The middle arrow is flanked by other arrows which either point in the same direction as the middle arrow or point in the opposite direction from the middle arrow.



An example of the two conditions of the Flanker task. In the Incongruent condition, the central arrow (which the subject is focusing on) points in the opposite direction as the flanking arrows; in the Congruent condition, the central arrow points in the same direction as the flanking arrows. In this example, the correct response in the Incongruent condition would be to push the "left" button and the correct response in the Congruent condition would be to push the "right" button.

fMRI Experiment Life Cycle



➤ Quality control

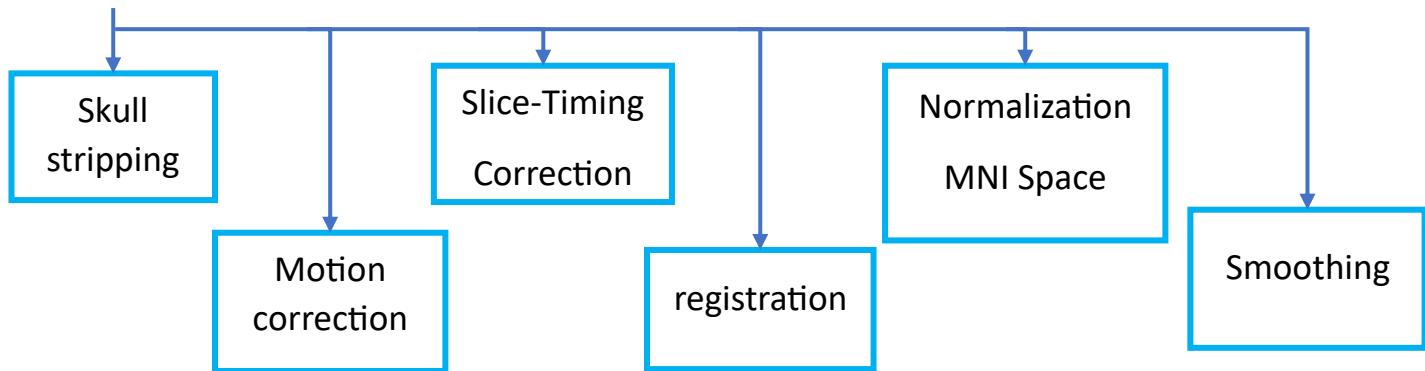
- Perform anatomical and function QC on all subjects.
- analysis artifacts in anatomical and functional and motion in functional (+ for low motion -> ++ high motion).

Subject name	T1	func		Motion
		Run1	Run2	
sub_01	Good	Good	Good	run1 +/run2++
sub_02	Good	Good	Good	run1 ++/run2++
sub_03	Good	Good	Good	run1 ++/run2+
sub_04	Good	Good	Good	run1 +/run2++
sub_05	RF inhomogeniy	Good	Good	run1 +/run2+
sub_06	aliasing	Good	Good	run1 ++/run2++
sub_07	Gibbs Ringing	Good	Good	run1+/run2+
sub_08	Good	Susceptibility	Susceptibility	run1 ++/run2+
sub_09	Good	Good	Good	run1+/run2+
sub_10	Good	Good	Good	run1++/run2++
sub_11	Aliasing	Good	Good	run1 +/run2++
sub_12	Chemical shift	Good	Good	run1 +/run2+
sub_13	Good	Good	Good	run1 +/run2++

Link for QC on all subjects:

https://docs.google.com/spreadsheets/d/17aU9ZjI7ggsIRwDO8irzQ4E8v-mQTt2h/edit?usp=share_link&ouid=108956132491236769138&rtpof=true&sd=true

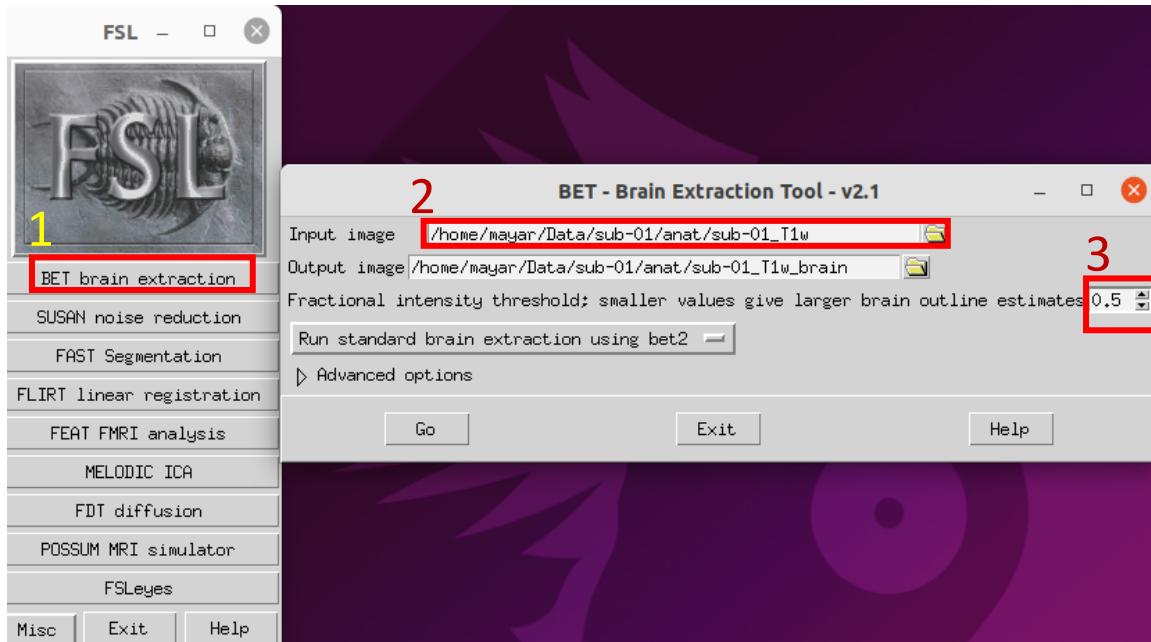
➤ Preprocessing



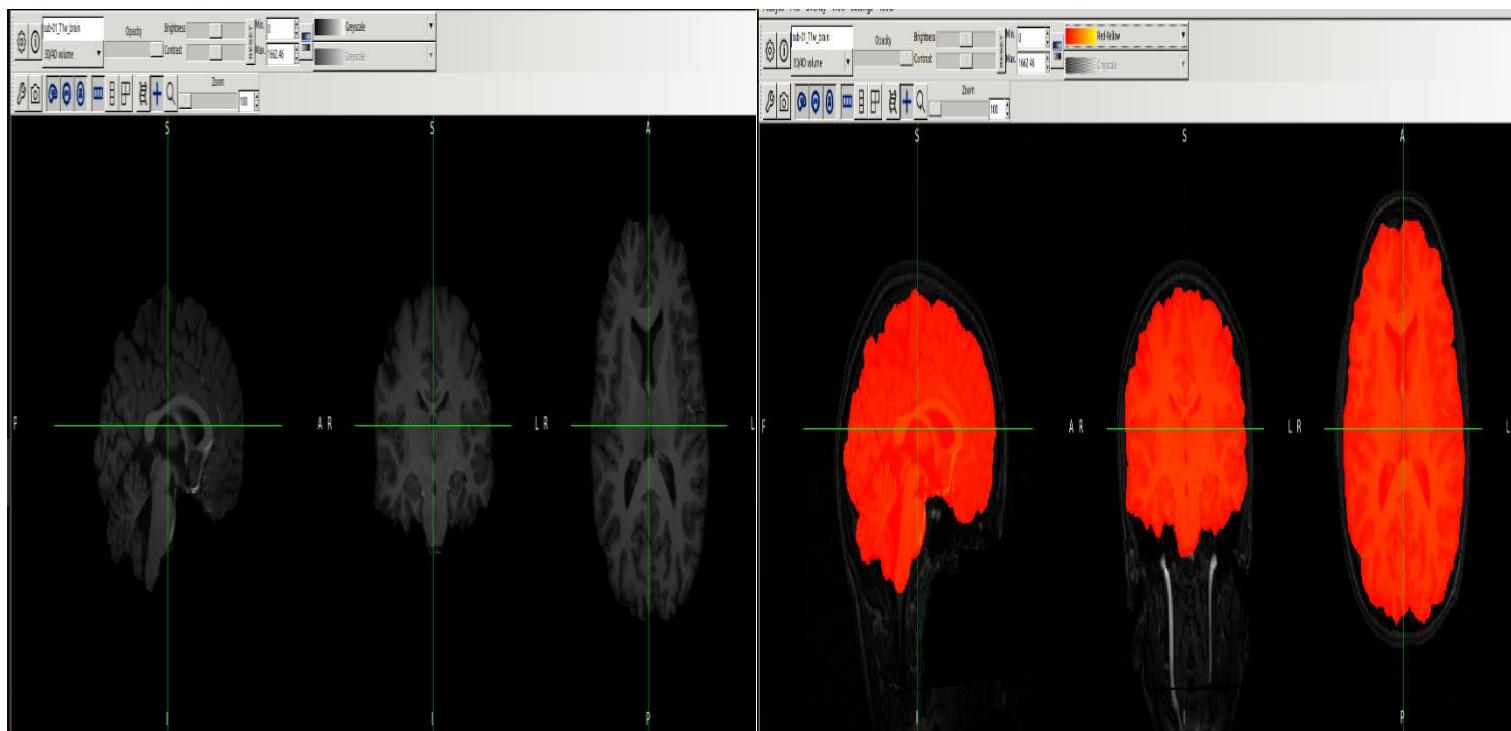
1) Brain Extraction (also known as “skull stripping”)

Why??

1. fMRI studies focus on brain tissue.
2. Improves registration and normalization.



The output:



1 2 3	Subject name	QC		Motion	Skull stripping (threshold)	
		T1	func			
			Run1	Run2		
4	sub_01	Good	Good	Good	run1 +/run2++	0.5
5	sub_02	Good	Good	Good	run1 ++/run2++	0.5
6	sub_03	Good	Good	Good	run1 ++/run2+	0.2
7	sub_04	Good	Good	Good	run1 +/run2++	0.5
8	sub_05	RF inhomogeniy	Good	Good	run1 +/run2+	0.5
9	sub_06	aliasing	Good	Good	run1 ++/run2++	0.4
10	sub_07	Gibbs Ringing	Good	Good	run1+/run2+	0.3
11	sub_08	Good	Susceptibility	Susceptibility	run1 ++/run2+	0.4
12	sub_09	Good	Good	Good	run1+/run2+	0.4
13	sub_10	Good	Good	Good	run1++/run2++	0.4
14	sub_11	Aliasing	Good	Good	run1 +/run2++	0.4

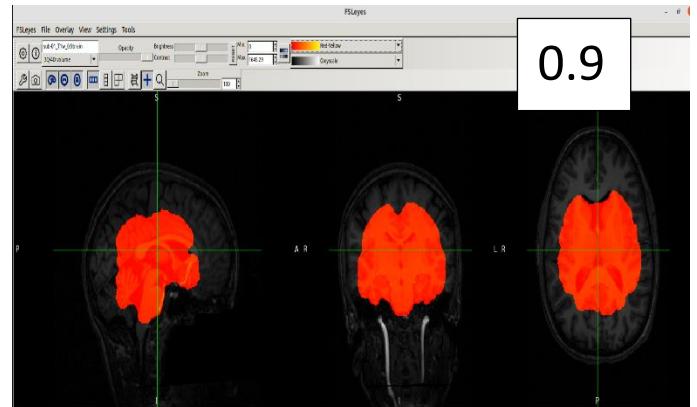
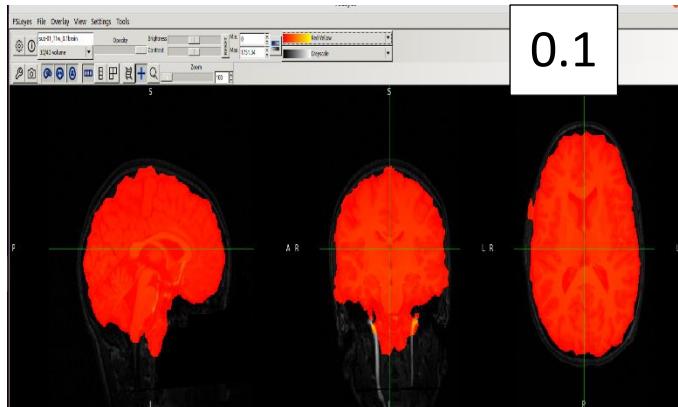
Link for skull stripping's fractional intensity threshold:

https://docs.google.com/spreadsheets/d/12iC7FPR9bPhP1qaLT-oT26pDRmOIHmf8/edit?usp=share_link&ouid=108956132491236769138&rtpof=true&sd=true

Exercises

- 1) Change the fractional intensity threshold to 0.1 and rerun BET, making sure to choose an appropriate output name to keep your files organized. View the result in FSLeys. Repeat these steps with a fractional intensity threshold of 0.9. What do you notice? What seems to be a good threshold?

The output with different fractional intensity threshold

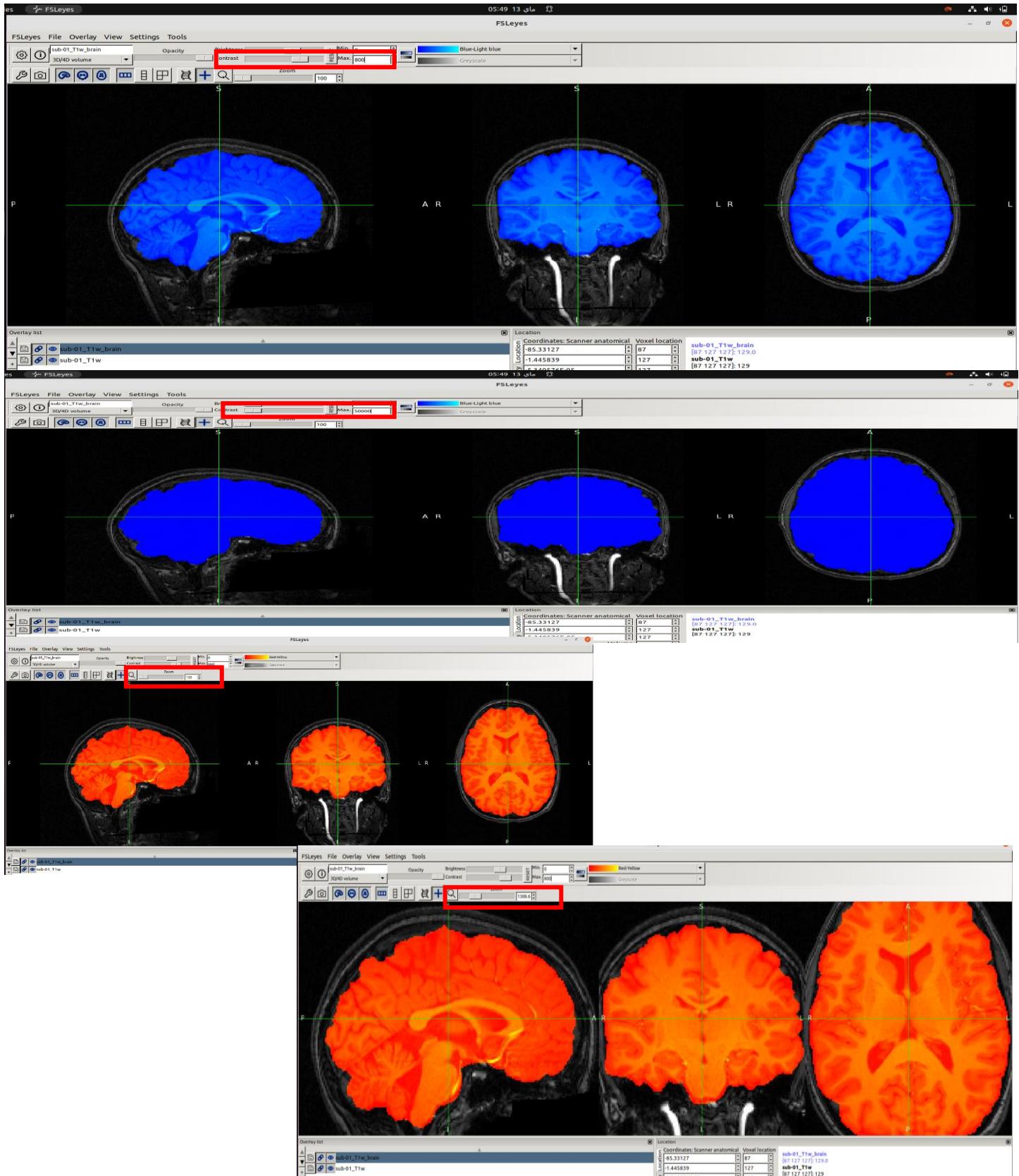


-when we used 0.1, the little skull was removed.

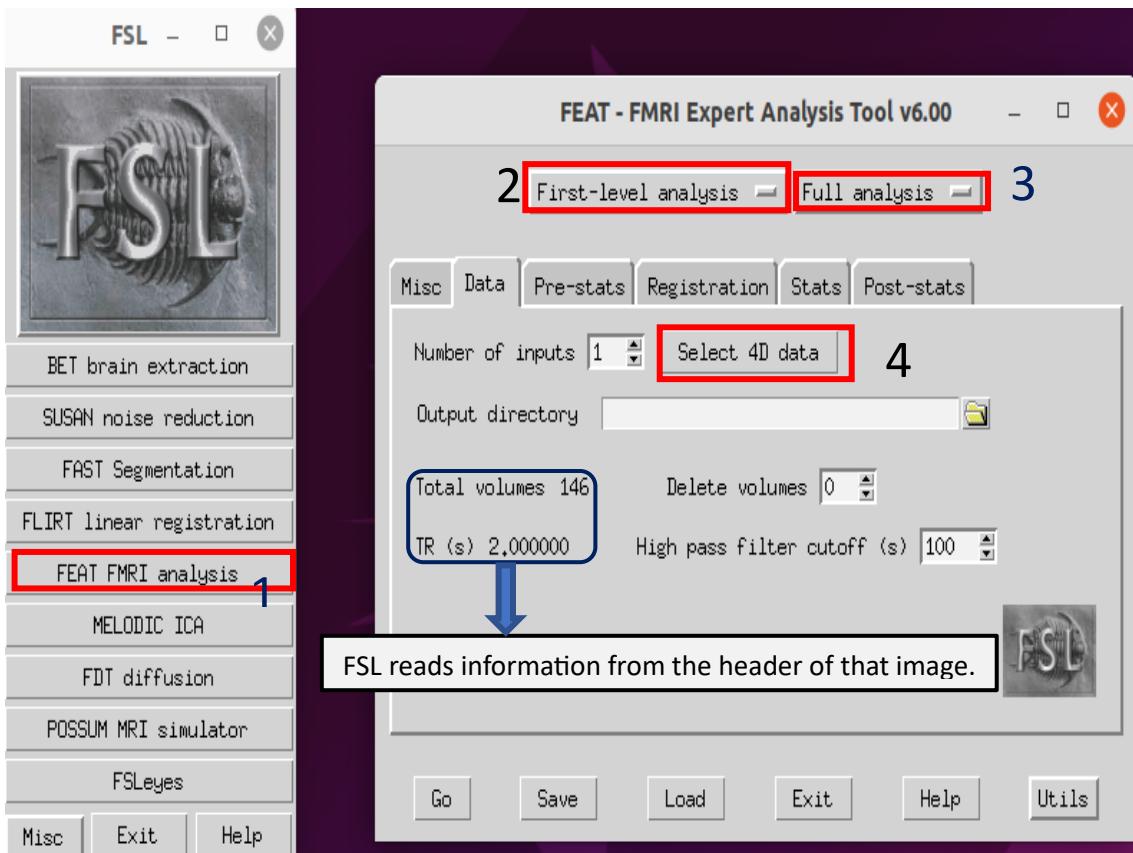
-when we used 0.9, the skull and too much brain tissue were removed.

good threshold → 0.1 (In general, keeping more brain is better than removing more skull.)

- 2) Experiment with different contrast colors for the overlay image in FSLeyes to see which one you like the best. Use the Zoom slider (next to the magnifying glass icon) to focus on a region you think hasn't been stripped well. Take a photo of the montage (i.e., all three viewing panes) by clicking on the Camera icon in the toolbar above the montage.

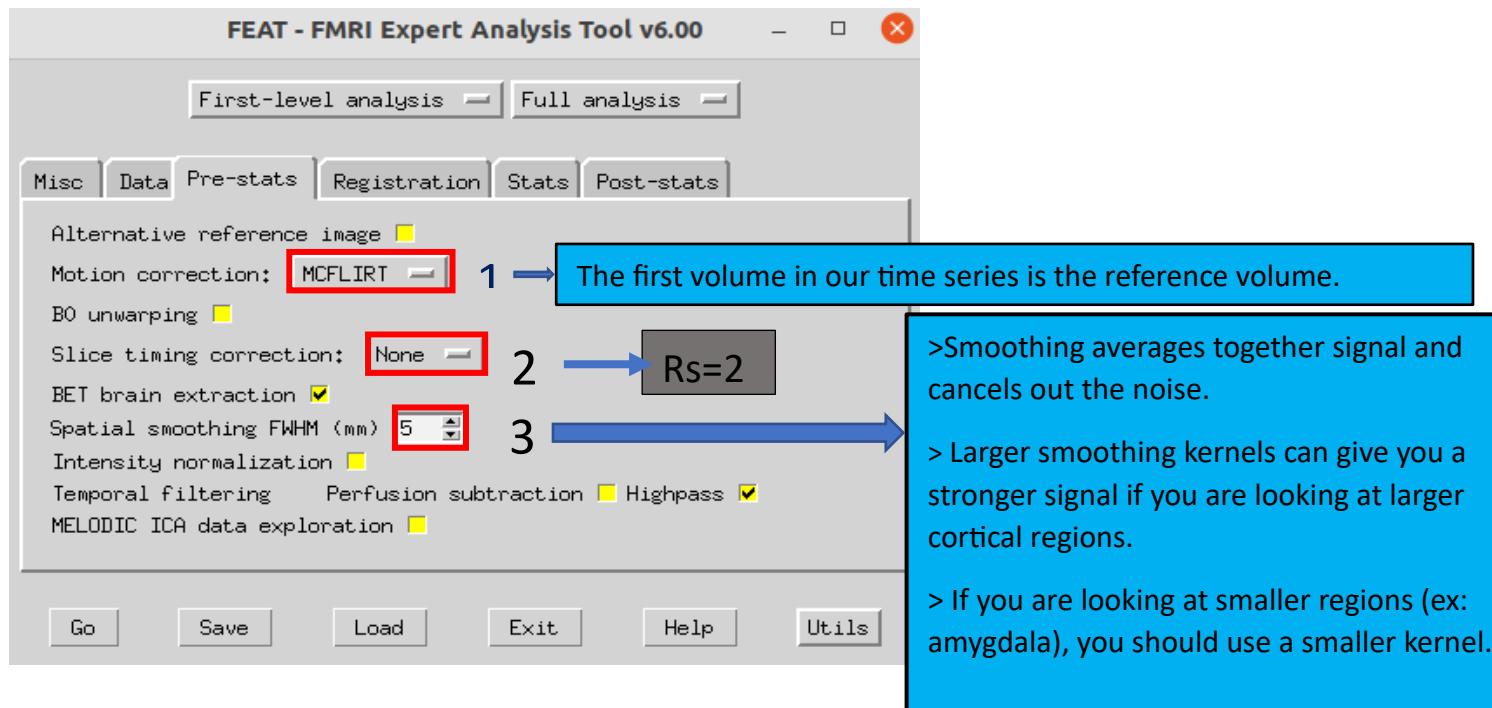


3) The FEAT GUI and loading the functional data.



- Motion Correction, Slice-Timing Correction and smoothing.

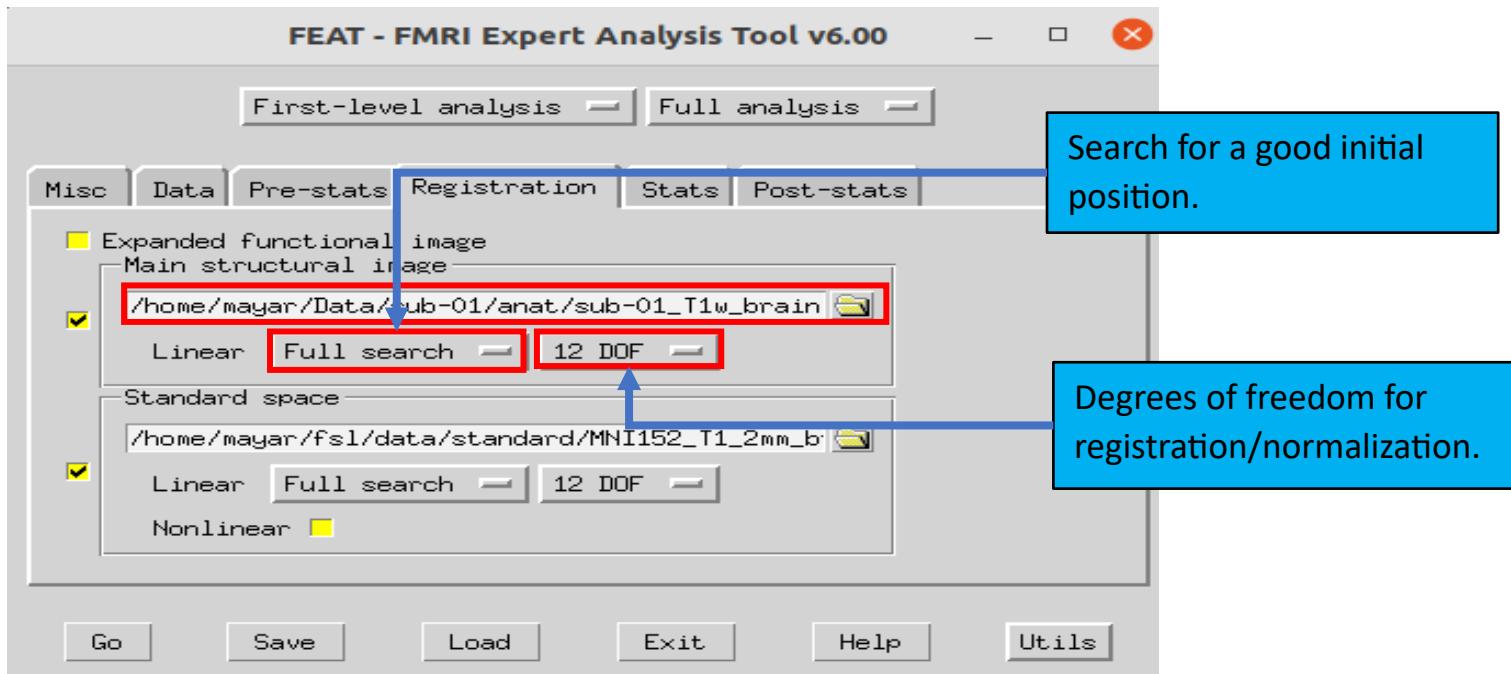
-Motion correction will undo any movements made by the subject during the scan slice.



- Registration and Normalization

Steps:

1. Align outlines of the images.
2. Use mutual information to align gyri, sulci, and ventricles.
3. When the best registration is found, normalize the images using the transformations that normalized the anatomical image.



• Statistics and Modeling (1st-Level Analysis)

-using this script to create timing files

-It creates 4 timing files in all subjects.

```

Activities Text Editor
Open make_Timings.sh
21:21 مای 1
#!/bin/bash

3
4 #check whether the file subjlist.txt exists; if not, create it
5 if [ ! -f subjlist.txt ]; then
6   ls | grep ^sub- > subjlist.txt
7 fi
8
9 #Loop over all subjects and format timing files into FSL format
10 for subj in `cat subjlist.txt`; do
11   cd $subj/func
12   cat ${subj}_task-flanker_run-1_events.tsv | awk '{if ($3=="incongruent_correct") {print $1, $2, 1}}' > incongruent_run1.txt
13   cat ${subj}_task-flanker_run-1_events.tsv | awk '{if ($3=="congruent_correct") {print $1, $2, 1}}' > congruent_run1.txt
14
15   cat ${subj}_task-flanker_run-2_events.tsv | awk '{if ($3=="incongruent_correct") {print $1, $2, 1}}' > incongruent_run2.txt
16   cat ${subj}_task-flanker_run-2_events.tsv | awk '{if ($3=="congruent_correct") {print $1, $2, 1}}' > congruent_run2.txt
17
18 #Now convert to AFNI format
19 #timing_tool.py -fsl_timing_files congruent*.txt -write_timing congruent.1D
20 #timing_tool.py -fsl_timing_files incongruent*.txt -write_timing incongruent.1D
21
22   cd ../..
23
24 done

```

Activities LibreOffice Calc

	A	B	C	D	E	F	G	H	I
1	onset	duration	trial_type	response_time	correctness	StimVar	Response	Stimulus	cond
2	0	2	incongruent_correct	1.095	correct	2	1	incongruent	cond003
3	10	2	incongruent_correct	0.988	correct	2	1	incongruent	cond003
4	20	2	congruent_correct	0.591	correct	1	1	congruent	cond001
5	30	2	congruent_correct	0.499	correct	1	1	congruent	cond001
6	40	2	incongruent_correct	0.719	correct	2	1	incongruent	cond003
7	52	2	congruent_correct	0.544	correct	1	1	congruent	cond001
8	64	2	congruent_correct	0.436	correct	1	1	congruent	cond001
9	76	2	incongruent_correct	0.47	correct	2	1	incongruent	cond003
10	88	2	congruent_correct	0.409	correct	1	1	congruent	cond001
11	102	2	incongruent_correct	0.563	correct	2	1	incongruent	cond003
12	112	2	congruent_correct	0.400	correct	1	1	incongruent	cond003

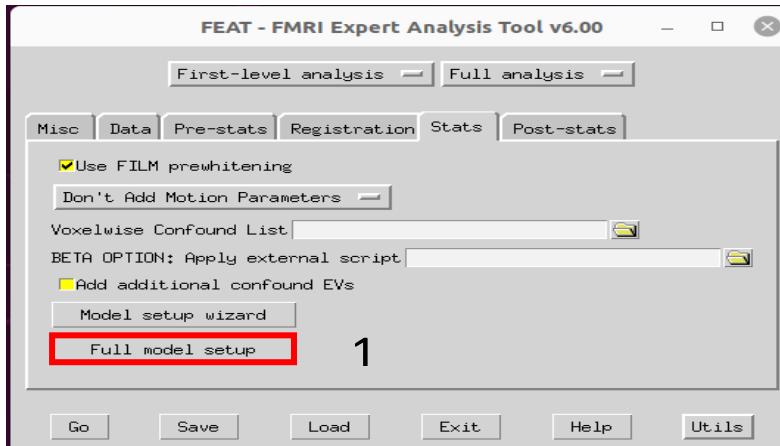
Activities Text Editor

1 20.0 2.0 1
2 30.0 2.0 1
3 52.0 2.0 1
4 64.0 2.0 1
5 88.0 2.0 1
6 116.0 2.0 1
7 130.0 2.0 1
8 140.0 2.0 1
9 184.0 2.0 1
10 196.0 2.0 1
11 246.0 2.0 1
12 274.0 2.0 1

Files

t	sub-01_task-flanker_r...	sub-01_task-flanker_r...	congruent_run1.txt	congruent_run2.txt	incongruent_run1.txt	incongruent_run2.txt	sub-01_task-flanker_r...	sub-01_task-flanker_r...	sub-01_task-flanker_r...	sub-01_task-flanker_r...
---	--------------------------	--------------------------	--------------------	--------------------	----------------------	----------------------	--------------------------	--------------------------	--------------------------	--------------------------

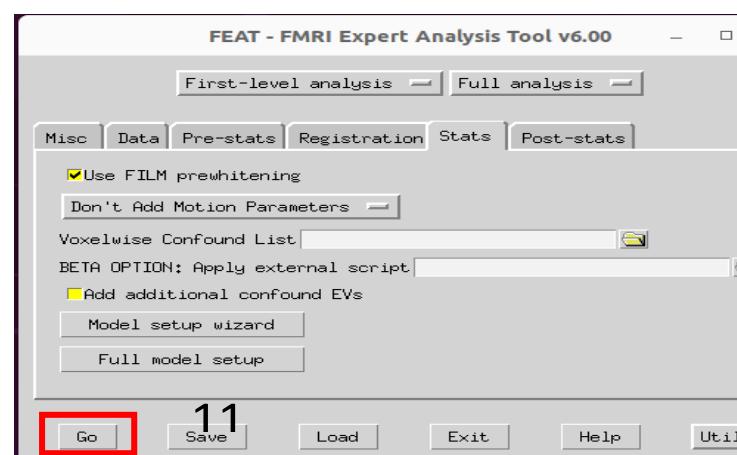
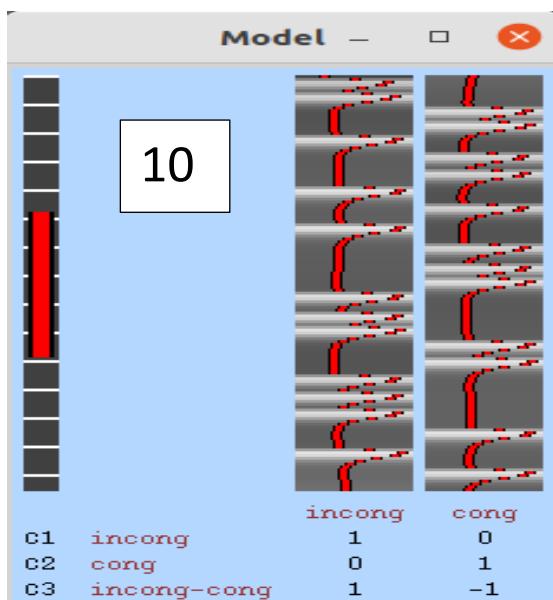
• Steps to make The General Linear Model (GLM)



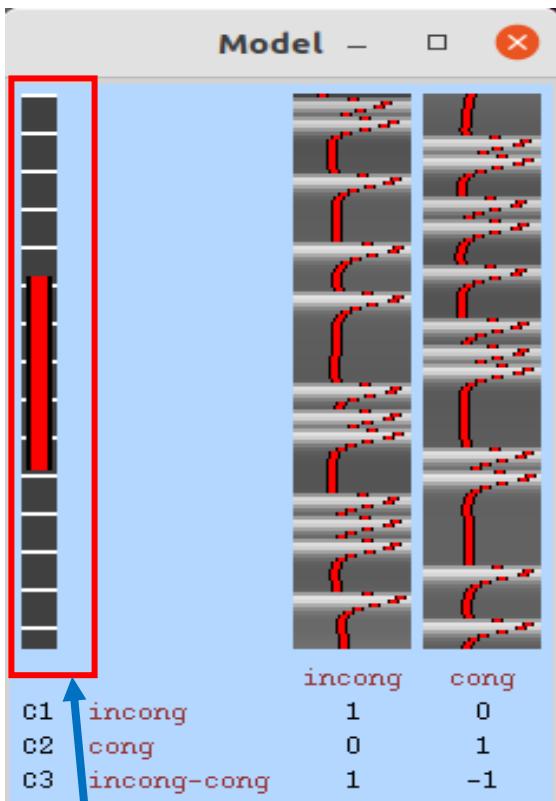
EV=Explanatory Variable (i.e., aRegressor)

1. Beta weight for Incongruent
2. Beta weight for Congruent
3. Contrast between Incongruent and Congruent beta weights

The image shows three stacked 'General Linear Model' dialog boxes. The first box (step 2) has 'Number of original EVs' set to 2, 'EV name' as 'incong', 'Basic shape' as 'Custom (3 column format)', and 'Convolution' as 'Gamma'. The second box (step 5) adds 'EV name' as 'cong', 'Basic shape' as 'Custom (3 column format)', and 'Convolution' as 'Gamma'. The third box (step 7) adds 'Contrasts' (set to 3) and 'F-tests' (set to 0). A red box highlights the 'Number of original EVs' field in the first two steps. Arrows point from step 2 to step 5, and from step 5 to step 7. A red box also highlights the 'Contrasts' and 'F-tests' fields in the third step.



11



1st Column: Ideal time-series for the Incongruent Condition.

2nd Column: Ideal time-series for the Congruent Condition.

White bars: Indicate when an onset time for that condition is convolved with the HRF.

High-Pass Filter

Removes Frequencies that are longer than the length of the red bar.

➤ The output of full analysis

FEAT Report
/home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat
Finished at 29 2023 , 12:29:40 EET
Registration - Pre-stats - Stats - Post-stats - Log

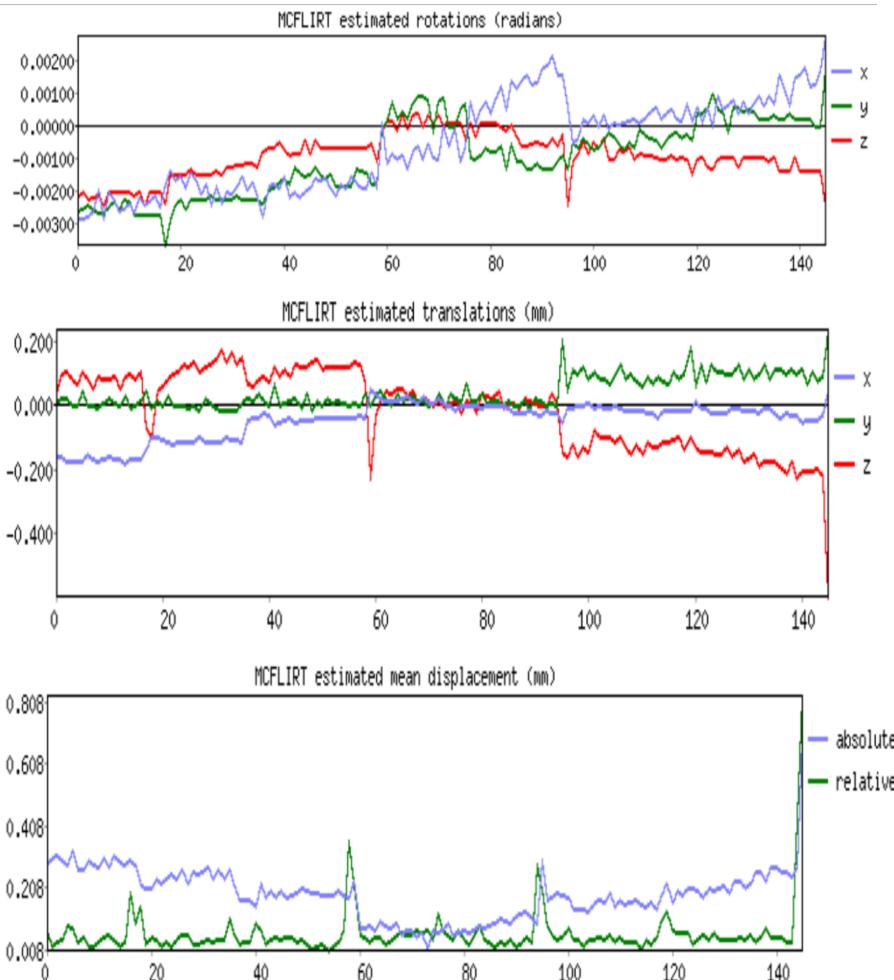
Progress Report / Log
Started at 29 2023 , 12:29:18 EET
Feat main script

```
/bin/cp /tmp/feat_9ek5904h.fsf design.fsf
mkdir -p /home/mayar/fsl/doc/fsl.css /files;cp -r /home/mayar/fsl/doc/images .files/images
/home/mayar/fsl/bin/fsl_sub -T 10 -l logs -N feat0_init /home/mayar/fsl/bin/feat /home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat/design.fsf -D /home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat -I 1 -init
3236
/home/mayar/fsl/bin/fsl_sub -T 44 -l logs -N feat2_pre -j 3236 /home/mayar/fsl/bin/feat /home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat/design.fsf -D /home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat -I 1 -prestats
4044
/home/mayar/fsl/bin/fsl_sub -T 1 -l logs -N feat5_stop -j 4044 /home/mayar/fsl/bin/feat /home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat/design.fsf -D /home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold.feat -stop
4551
/bin/cp /tmp/feat_kp69fc17.fsf design.fsf
/home/mayar/fsl/bin/feat_model design
```

● Pre-stats

MCFLIRT Motion correction

Mean displacements: absolute=0.18mm, relative=0.05mm



Too much motion:
more than 1/2
voxel size from
volume to
volume, or 1 voxel
absolute motion

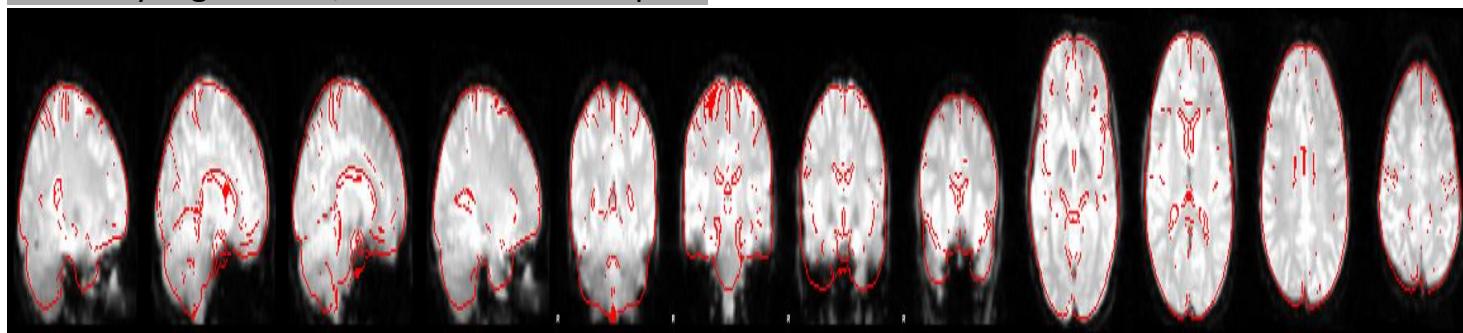
Consider
scrubbing volume
or removing run
from analysis.

- ## Registration

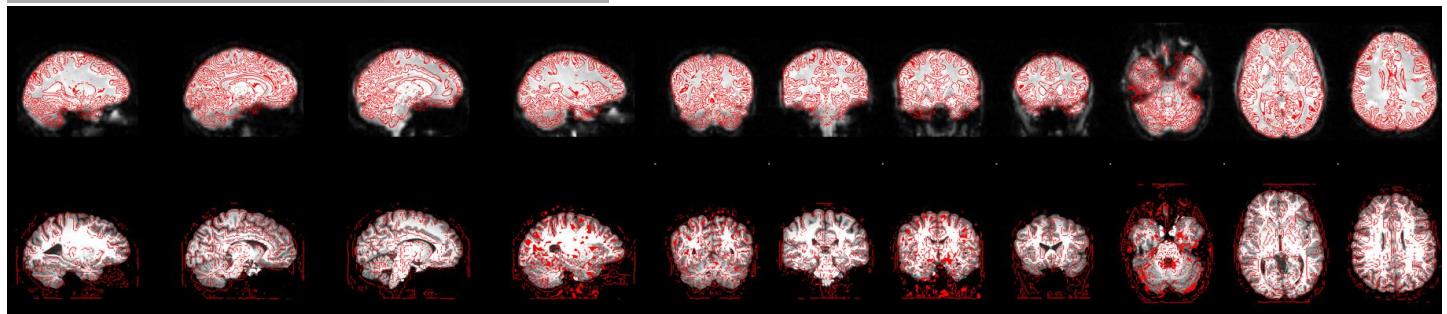
Each image overlays the red outline of a brain onto a greyscale image of another brain. The first montage, Summary Registration, shows a representative functional image - in this case, the median image in the fMRI time-series - as the underlay, and the template brain as the red lines. This image is shown first, because if there were any problems in any of the previous registration or normalization steps, there would be obvious errors in this image, such as the image being skewed or largely outside of the red outline.

Look for whether the red outlines approximately trace the outline of the greyscale image. Also check whether some of the internal structures of the images, such as the ventricles, are aligned. Do these same quality checks for the other alignments, such as the registration of the example functional image to the highres image (i.e., the anatomical image) and the normalization of the highres image to the standard space template.

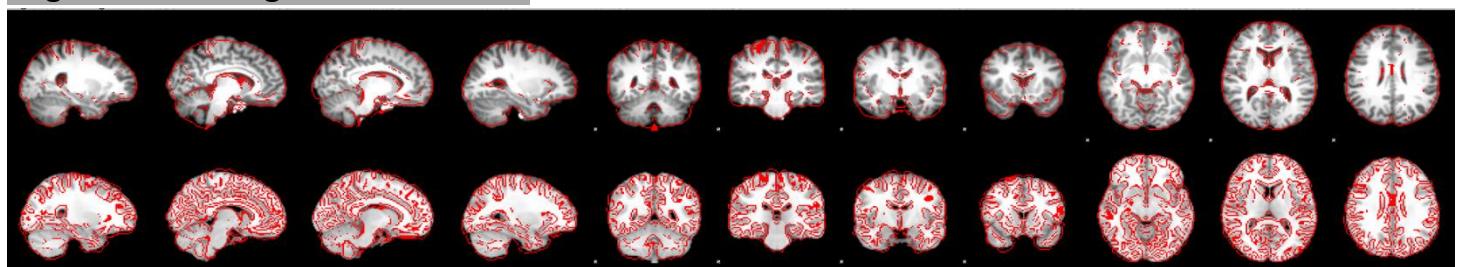
Summary registration, FMRI to standard space



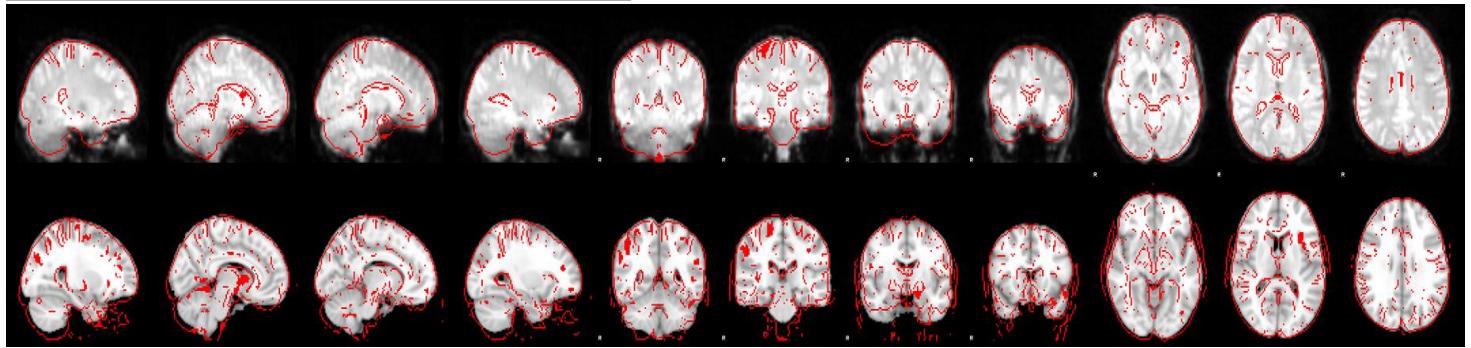
Registration of example_func to highres



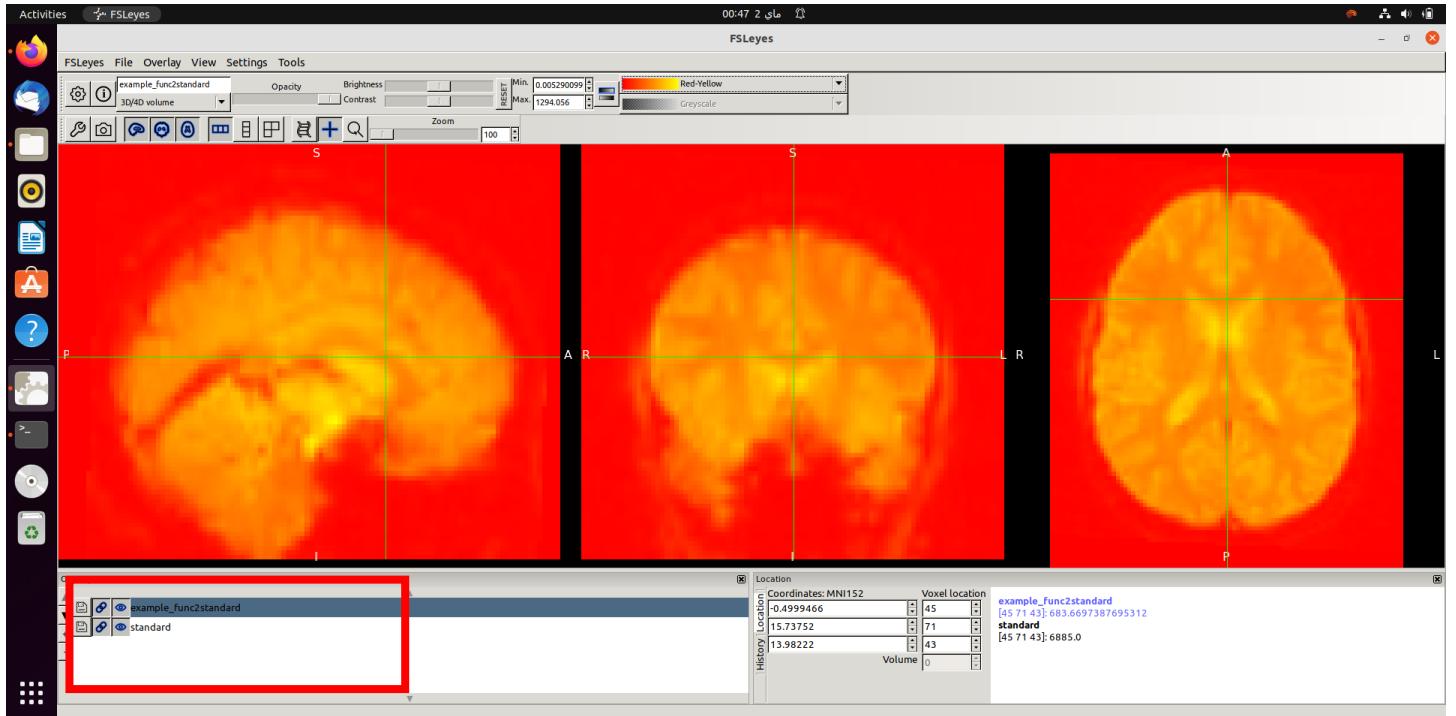
Registration of highres to standard



Registration of example_func to standard



-After Registration and Normalization, the function applies to the standard.

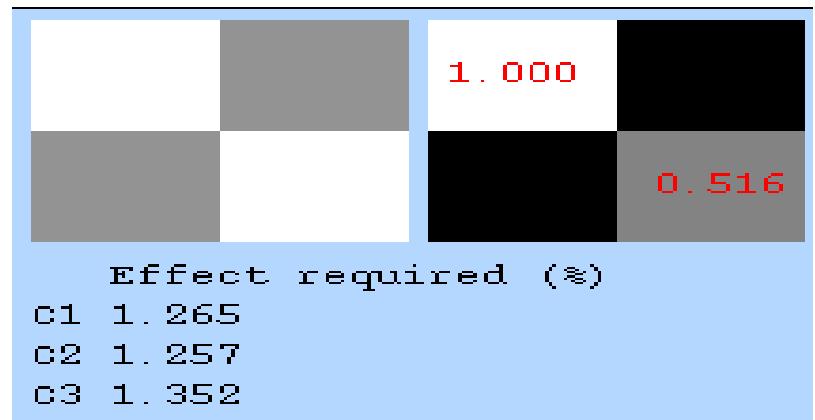


- Stats

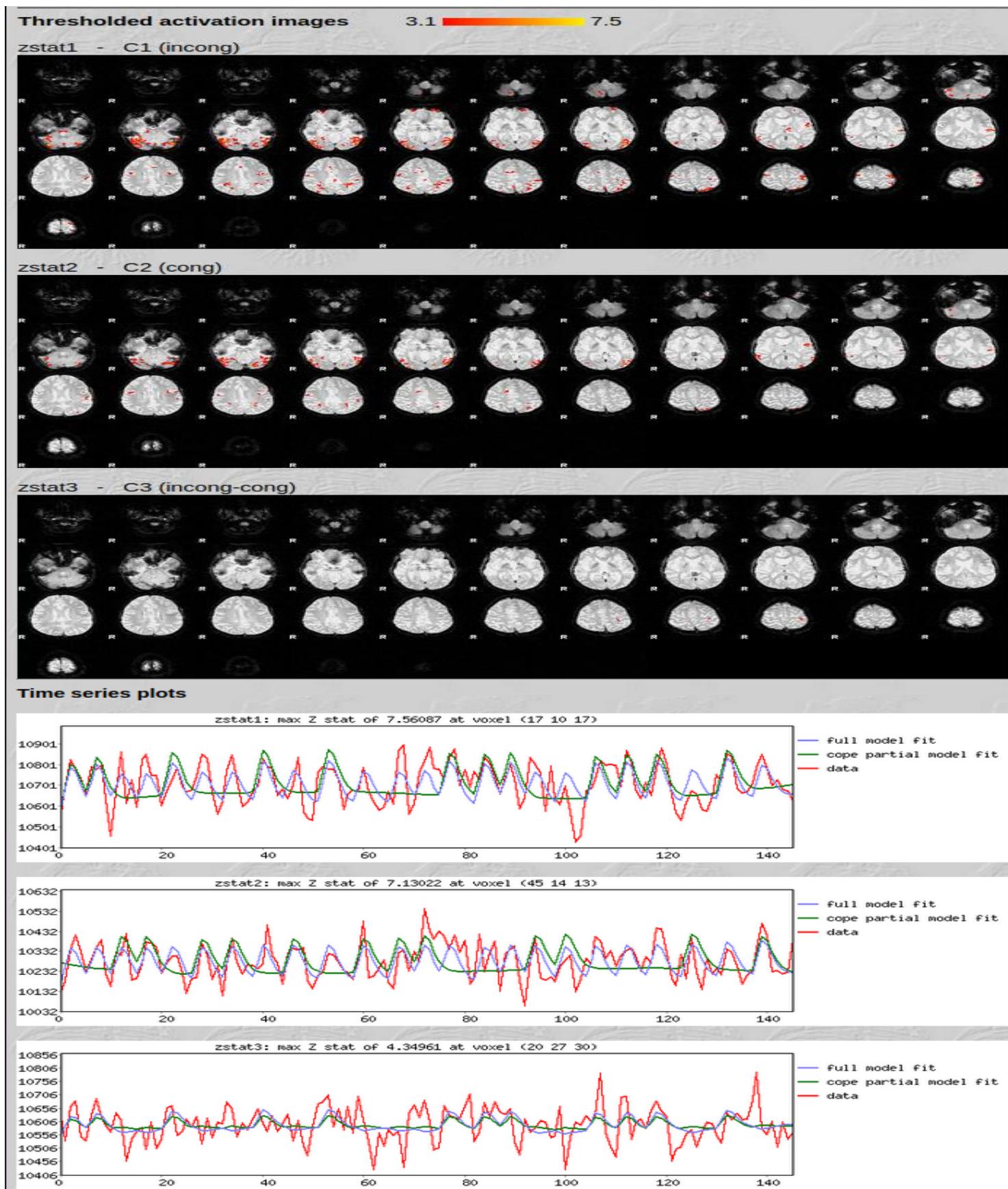
Design matrix

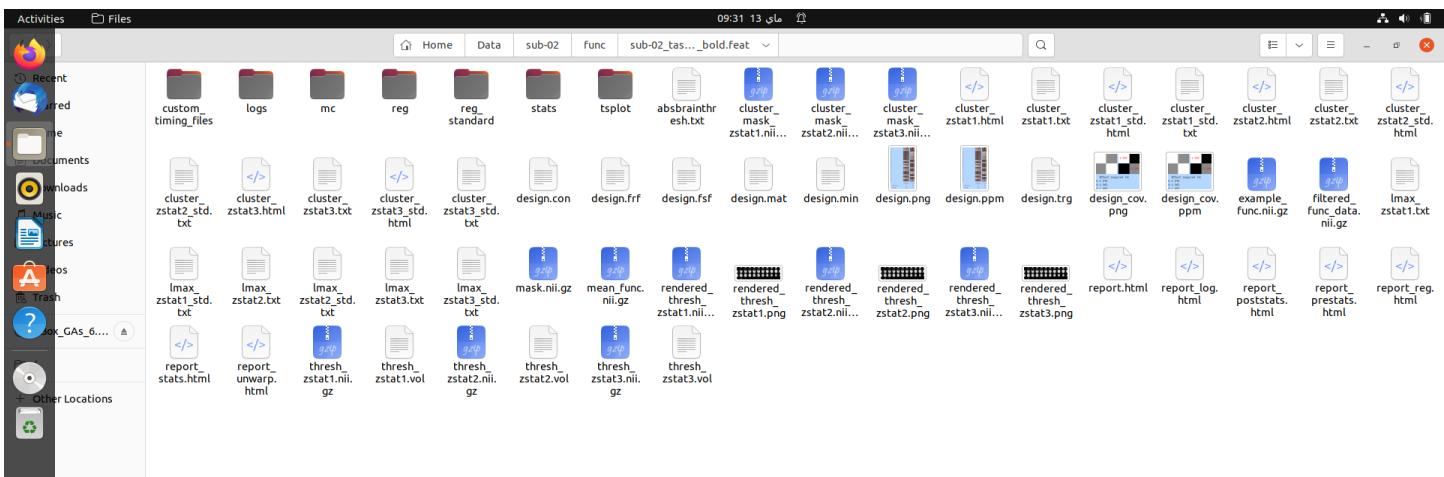
	incong	cong
c1	incong	1
c2	cong	0
c3	incong-cong	1

Covariance matrix & design efficiency



- Post-stats

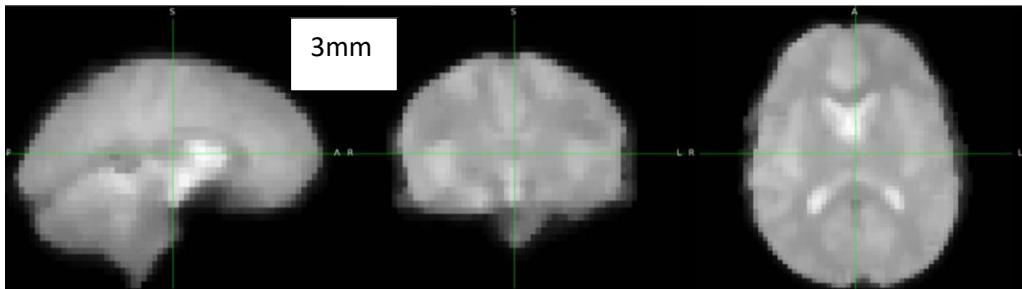




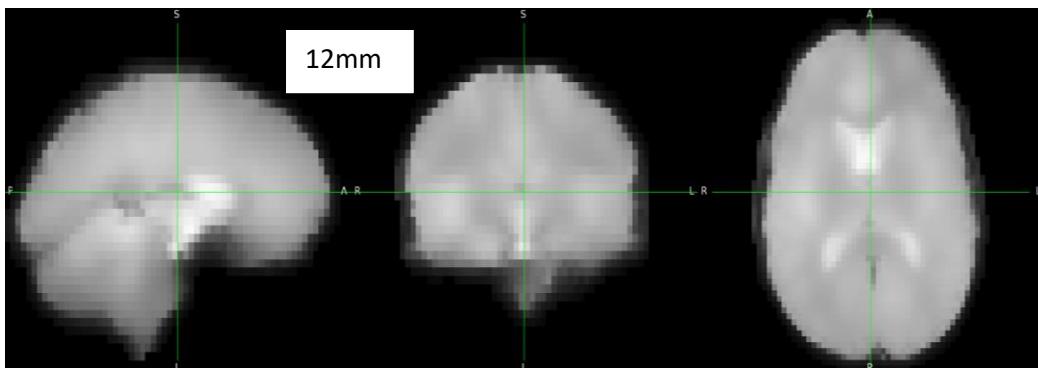
- The logs folder contains text files reporting what was done for each pre-processing step.
- The MC folder contains files generated during motion correction.
- The reg folder contains the output from the registration and normalization.
- The stats folder contains the output from the GLM (cope1, cope2, cope3).

Exercises

3. Preprocess run 1 using a 3mm smoothing kernel, keeping the other preprocessing options the same. (Make sure, however, to change the output directory to a new name in order to keep the output organized.) Before you look at the output, run another analysis with a 12mm smoothing kernel. Think about what you would expect the preprocessed functional data to look like, and then load the filtered_func_data.nii.gz images from each analysis into FSLeyes. How do they compare to your predictions?

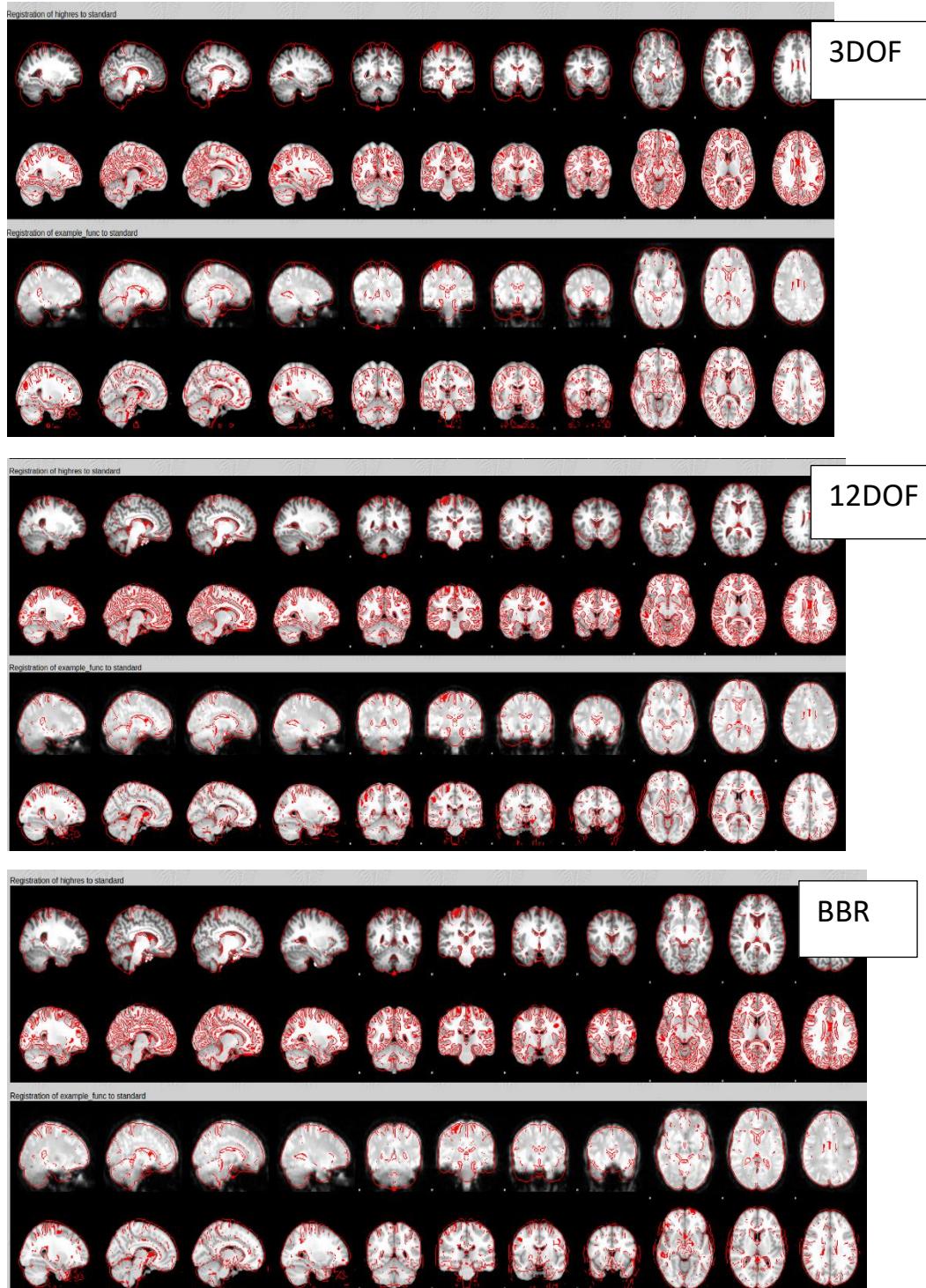


-We notice that the larger the kernel size, the blurrier the image becomes.



4. Preprocess run 1 using 3DOF for registration and normalization. How is the output different from what you saw when you ran the preprocessing with 12DOF? Why?

5. Rerun registration for run 1 using BBR instead of 12DOF. What difference does it make? How would you make a case to someone that you should use one instead of the other?



Links for results of full analysis for 5 subjects

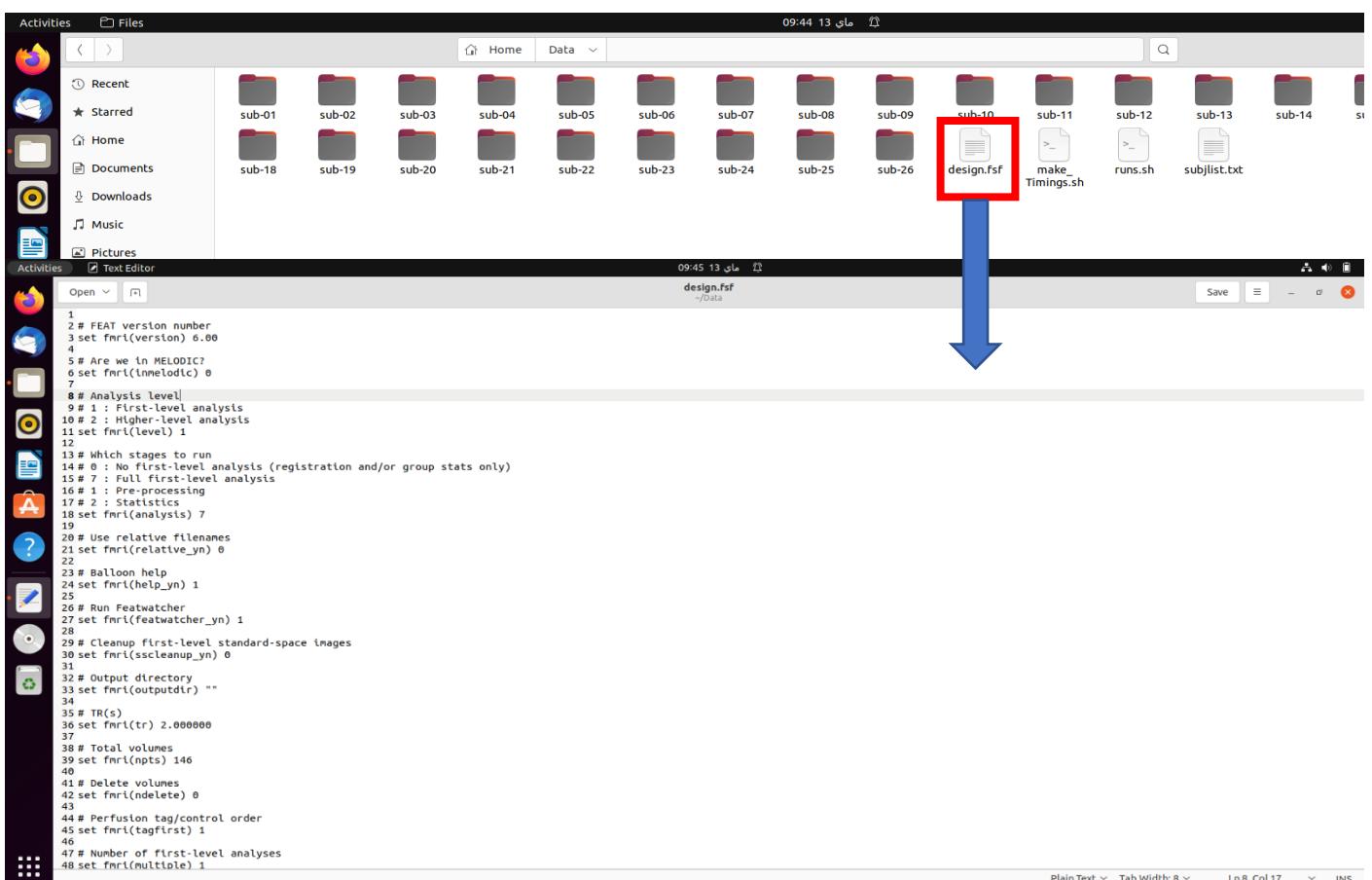
1. file:///home/mayar/Data/sub-01/func/sub-01_task-flanker_run-2_bold.feat/report_log.html
2. file:///home/mayar/Data/sub-02/func/sub-02_task-flanker_run-1_bold.feat/report_log.html
3. file:///home/mayar/Data/sub-02/func/sub-02_task-flanker_run-2_bold.feat/report_log.html
4. file:///home/mayar/Data/sub-03/func/sub-03_task-flanker_run-1_bold.feat/report_log.html
5. file:///home/mayar/Data/sub-03/func/sub-03_task-flanker_run-2_bold.feat/report_log.html
6. file:///home/mayar/Data/sub-04/func/sub-04_task-flanker_run-1_bold.feat/report_log.html
7. file:///home/mayar/Data/sub-04/func/sub-04_task-flanker_run-2_bold.feat/report_log.html
8. file:///home/mayar/Data/sub-05/func/sub-05_task-flanker_run-1_bold.feat/report_log.html
9. file:///home/mayar/Data/sub-05/func/sub-05_task-flanker_run-2_bold.feat/report_log.html

- Using this script to make a full analysis from sub_06 to sub_26, it uses design.fsf because it contains all steps for full analysis.



A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Text Editor". The terminal displays a shell script named "runs.sh" located at "/Data". The script performs several tasks: it loops through subject IDs (seq 6 to 26), processes each subject's T1w_brain.nii.gz file, copies "design1.fsf" to "design2.fsf", changes the subject name in "design2.fsf" using sed, and runs FLA (feat) for both runs 1 and 2. The terminal window has a dark theme, and the desktop background is visible behind it.

```
1 for id in `seq -w 6 26`; do
2     subj="sub-$id"
3     echo "Start processing $subj"
4     echo
5
6     cd $subj
7     if [ ! -f anat/${subj}_T1w_brain.nii.gz ]; then
8         bet2 anat/${subj}_T1w.nii.gz anat/${subj}_T1w_brain.nii.gz -f 0.4
9         fi
10
11    cp ../design.fsf design1.fsf
12    #change subject to the current subject using "sed" command
13    sed -i "s/sub-05/$subj/g" design1.fsf
14
15    cp design1.fsf design2.fsf
16    #change run1 with run2 in destgn2.fsf using "sed" command
17    sed -i "s/run1/run2/g" design2.fsf
18    sed -i "s/run-1/run-2/g" design2.fsf
19
20    echo "FLA RUN 1"
21    echo
22    feat design1.fsf
23    echo "FLA RUN 2"
24    echo
25    feat design2.fsf
26
27    cd ..
28
29
30 done
31 echo
```



➤ 2nd Level Analysis

-In FSL a 2nd-level analysis is the averaging together within each subject the parameter estimates and contrast estimates from the 1st-level analyses.

Cope 1 → incongr

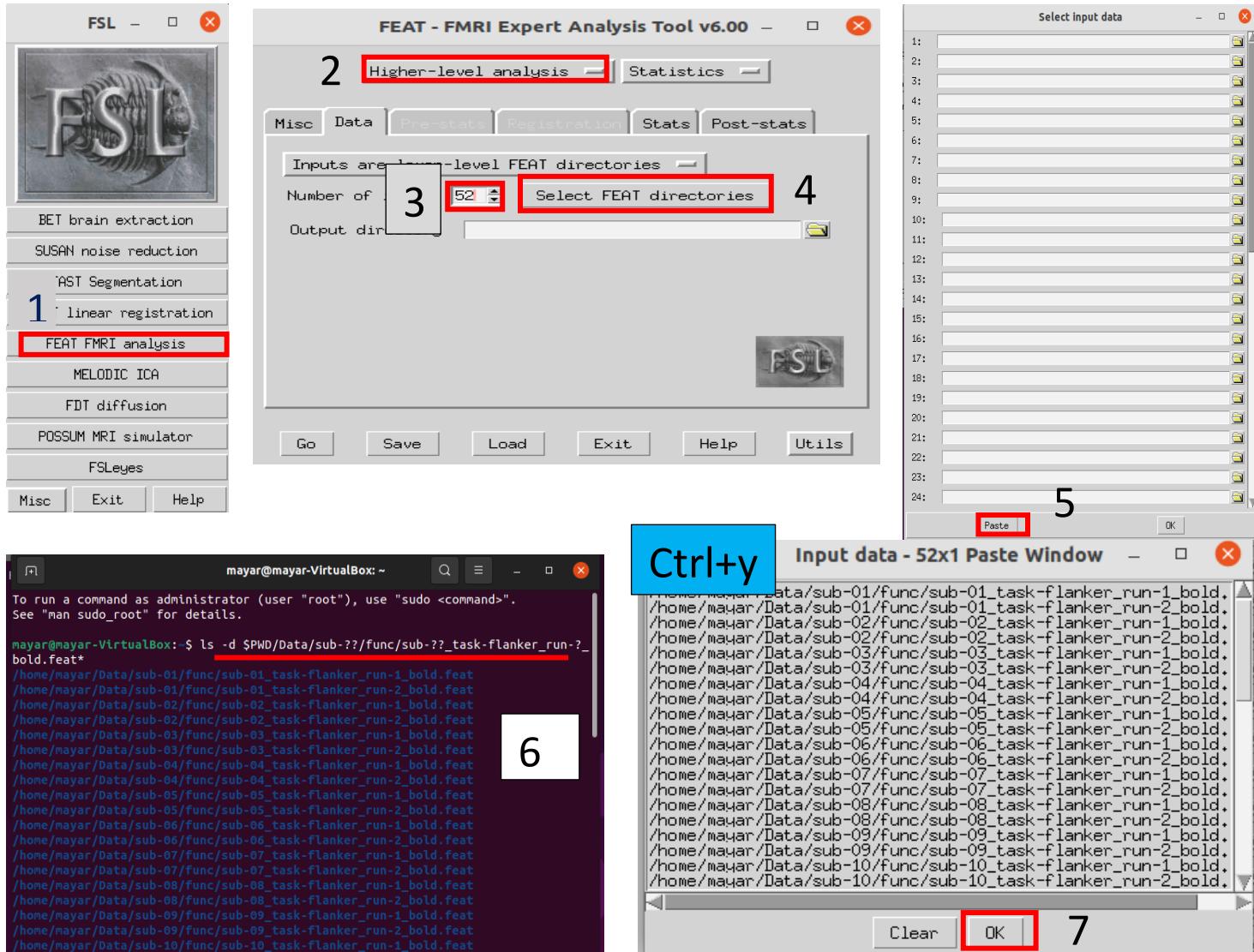
Cope 2 → cong

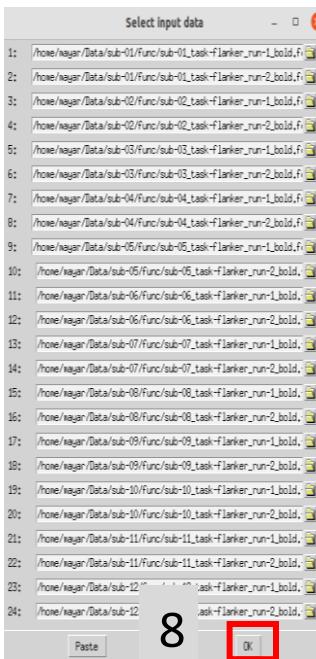
26 subjects →(Run1 & Run2)

Cope 3 → incong – cong

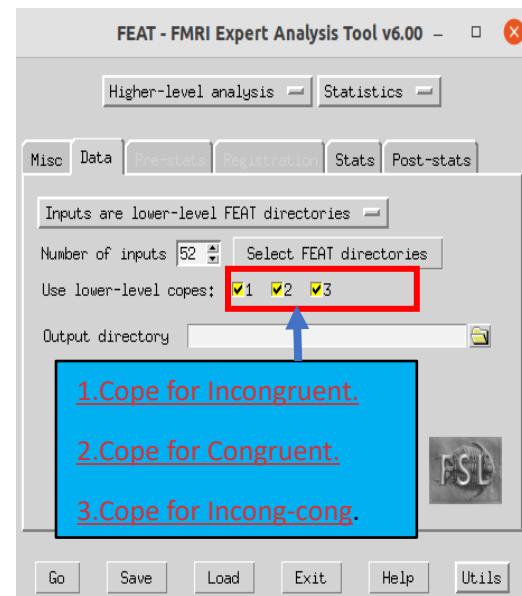
/Desktop/Flanker/sub-01/run1.feat/stats/cope1.nii.gz
+
/Desktop/Flanker/sub-01/run2.feat/stats/cope1.nii.gz
÷2
↓
2nd-level cope1.nii.gz

Steps:

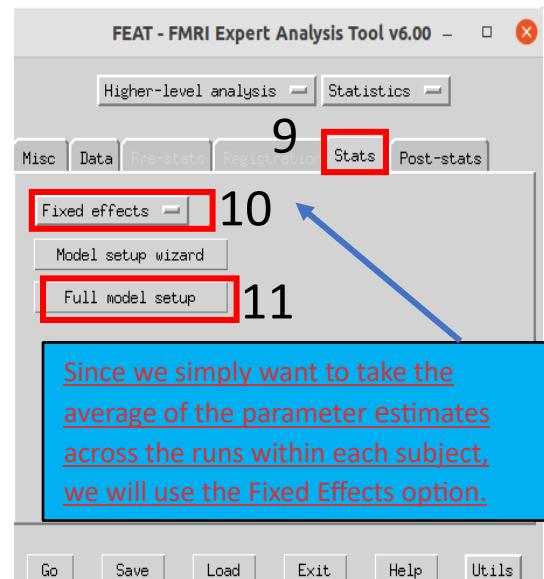




8

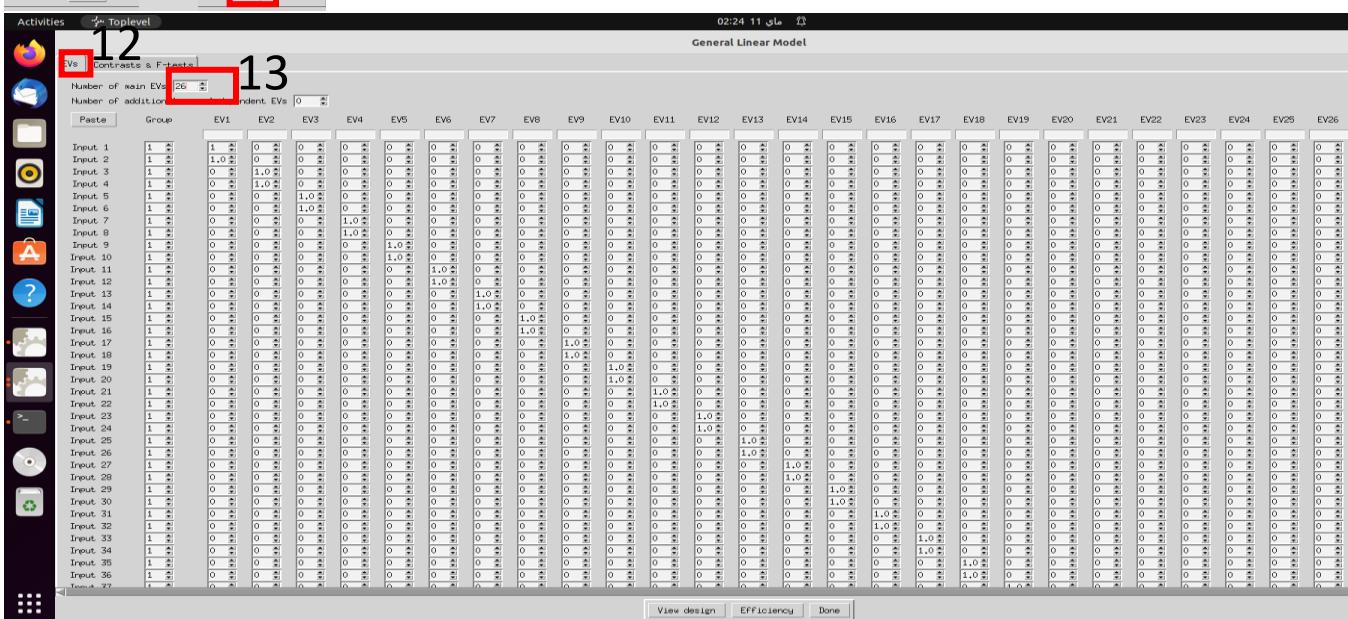


9



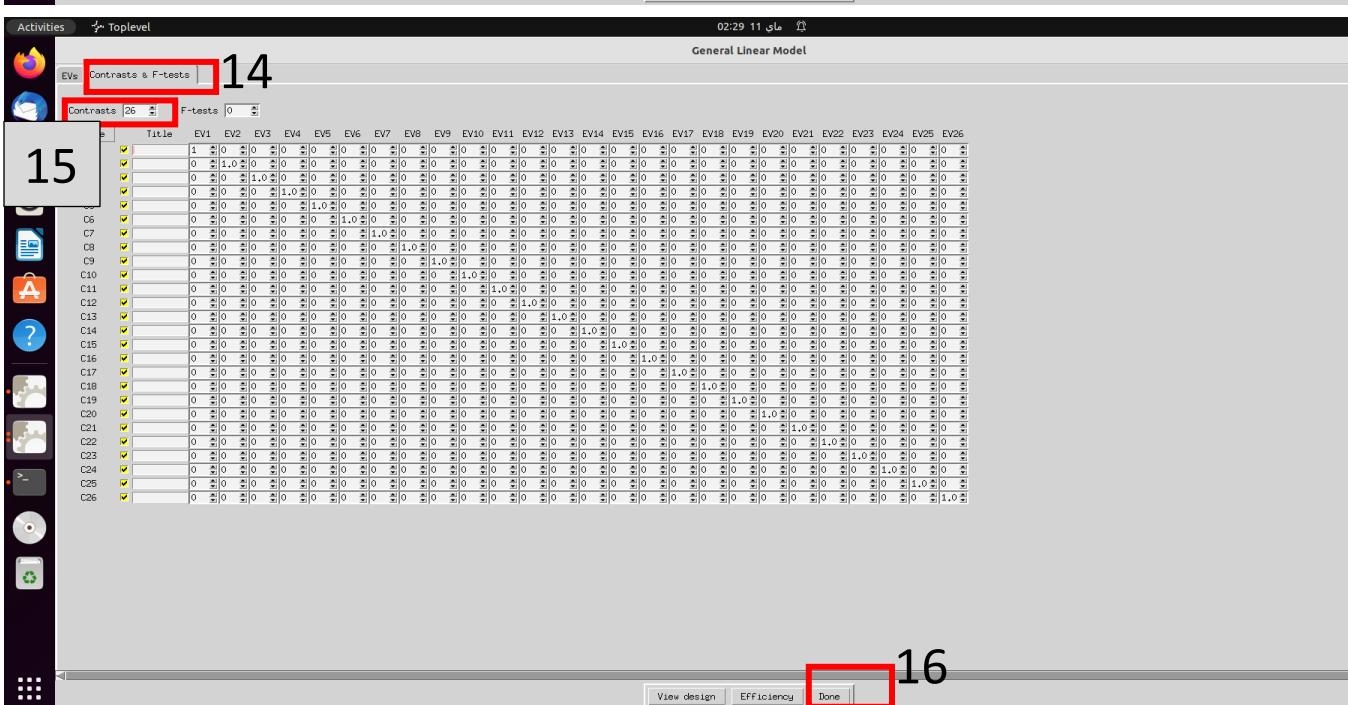
10

11



12

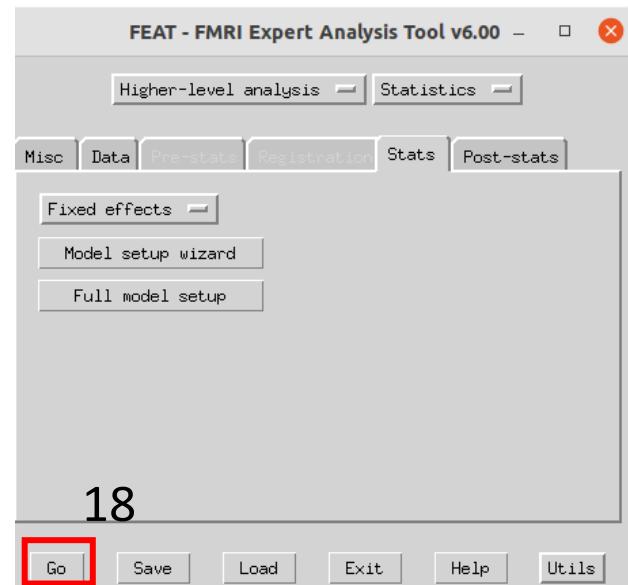
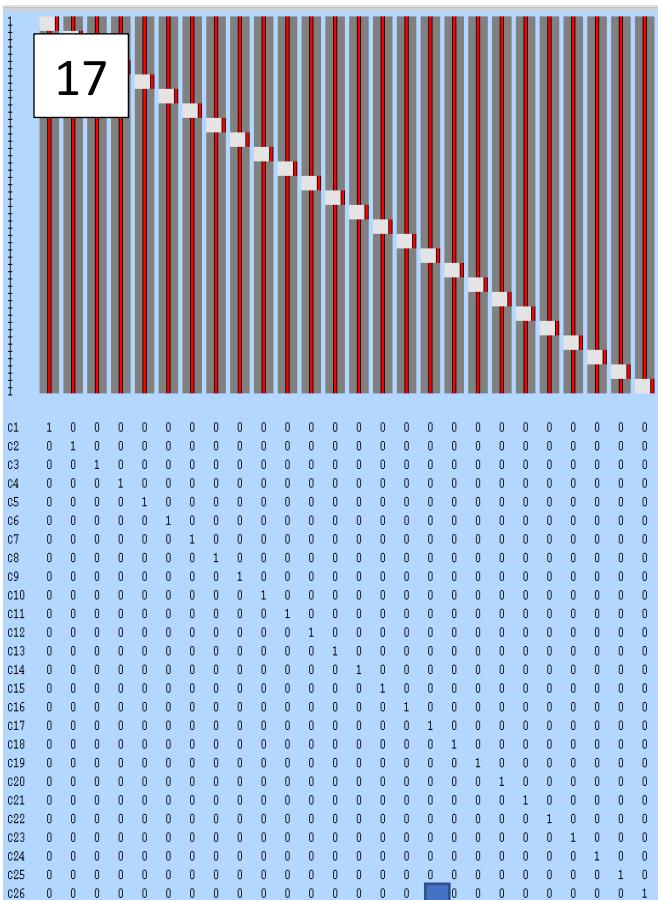
13



14

15

16



The design matrix checks to make sure that gray boxes are on the diagonal and that the contrast weights down below are a straight diagonal of ones.

The screenshot shows the FEAT - FMRI Expert Analysis Tool v6.00 window. At the top, there are tabs for 'Higher-level analysis' and 'Statistics'. Below these are tabs for 'Misc', 'Data', 'Pre-stats', 'Registration', 'Stats', and 'Post-stats', with 'Stats' being the active tab. A dropdown menu titled 'Mixed effects' is open, listing several options: 'Mixed effects: FLAME 1', 'Fixed effects', 'Mixed effects: Simple OLS', 'Mixed effects: FLAME 1', 'Mixed effects: FLAME 1+2', and 'Randomise'. A blue arrow points from the text in the main content area down to the 'Mixed effects' dropdown menu.

These are different statistical methods used in the analysis of neuroimaging data, specifically in the context of functional magnetic resonance imaging (fMRI) studies.

1. Fixed Effects: This method involves taking the average of the parameter estimates calculated for each subject in the sample. It does not allow for generalization beyond the sample and assumes that the effects observed in the sample are representative of the population. This method is useful when the focus is on the group rather than individual differences.
2. Mixed Effects: Simple OLS (Ordinary Least Squares): This method performs a t-test on the average parameter estimates calculated for each subject, without considering the variability between the runs for each subject. It assumes that the effects observed in the sample are the same for each subject and does not account for individual differences. It is a simple and quick method but may not be appropriate when there is substantial variability between subjects.
3. Mixed Effects: FLAME 1: This method weights each subject's parameter estimate by the variance of that contrast estimate. This means that a subject with a relatively low variance will be weighted more, and a subject with relatively high variance will be weighted less. This method accounts for individual differences and allows for generalization beyond the sample. It is a more complex method but is appropriate when there is variability between subjects.
4. Mixed Effects: FLAME 1+2: This is a more rigorous version of FLAME 1 and takes much longer to perform. It is only helpful for analyzing small samples (e.g., 10 subjects or fewer). It accounts for individual differences and allows for generalization beyond the sample.
5. Randomise: This is a non-parametric test used to assess the significance of brain activation or connectivity patterns. It is commonly used in studies with small sample sizes or with non-normal data distributions. The method involves randomly permuting the condition labels across all participants to create a null distribution of the test statistic. The observed test statistic is then compared to the null distribution to obtain a p-value, which represents the probability of obtaining a test statistic as extreme or more extreme than the observed value, under the null hypothesis. If the p-value is below a pre-specified threshold (e.g., 0.05), the null hypothesis is rejected, and the result is considered statistically significant.

➤ The output of 2nd Level Analysis

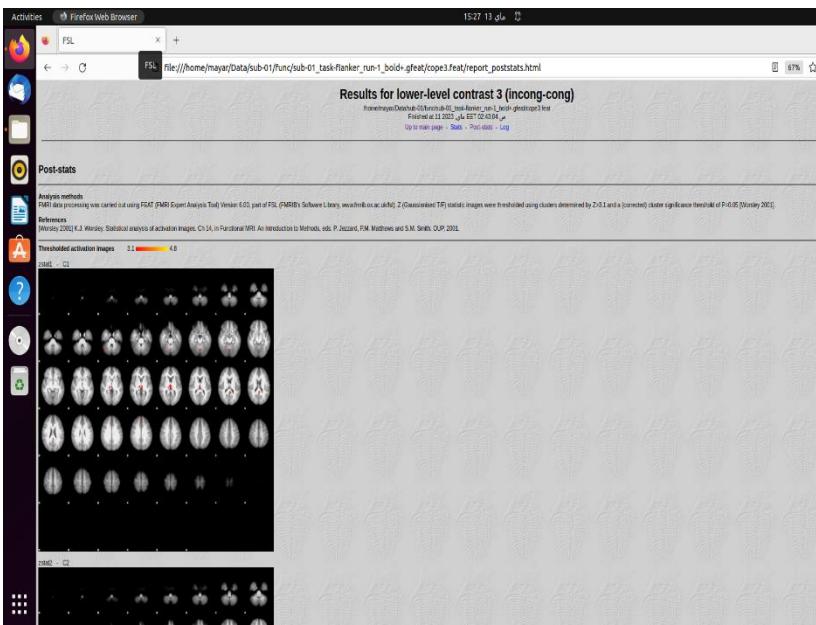
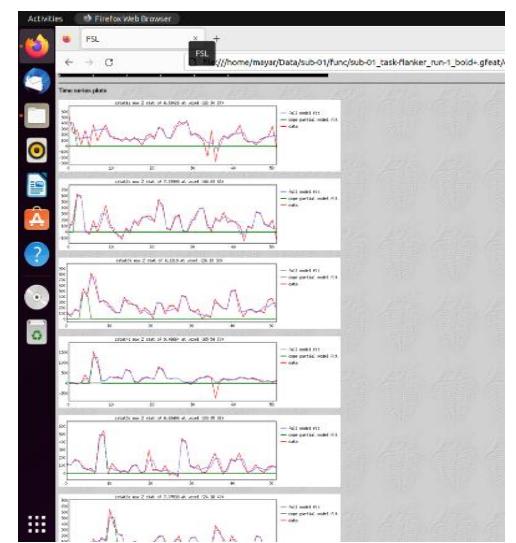
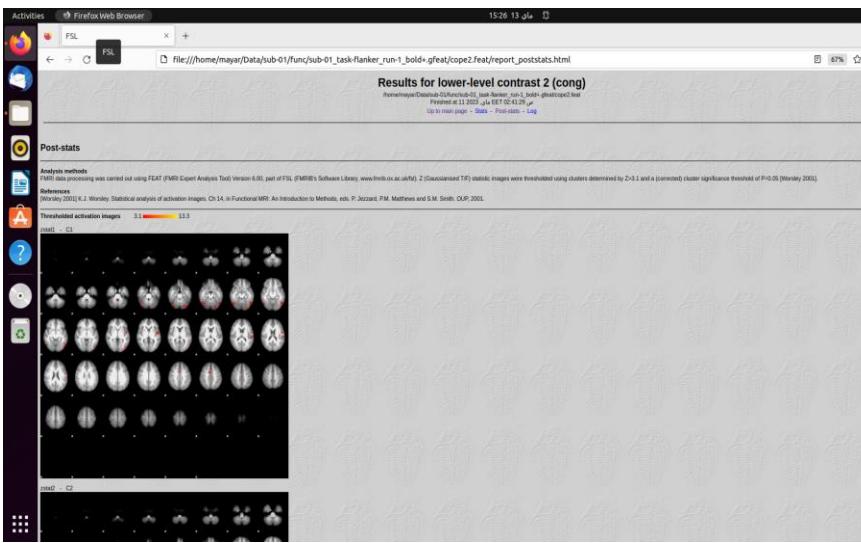
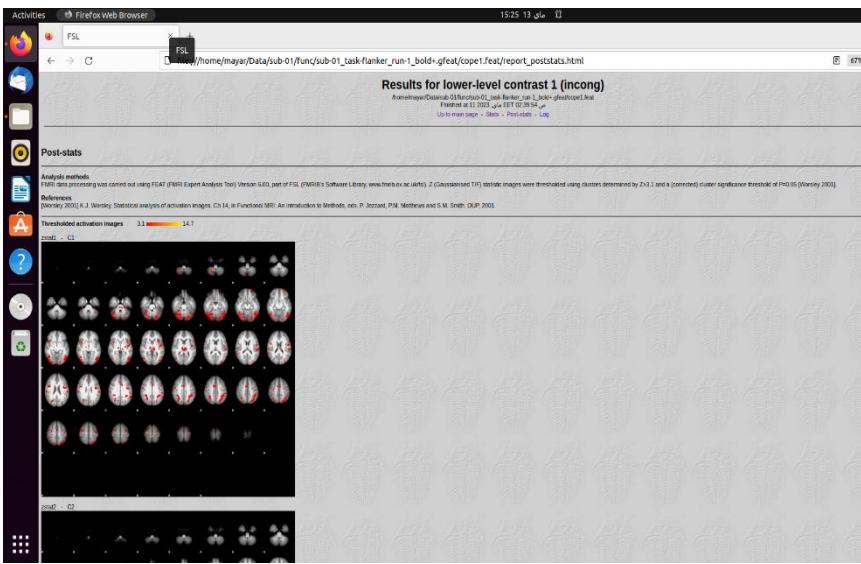
The screenshot shows a web-based report titled "Higher-Level FEAT Report". At the top, it displays the command used: "/home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold+gfeat/report_firstlevel.html". Below this, the text "Higher-Level FEAT Report" is centered, followed by the path and a timestamp: "Finished at 11 2023, 02:43:04 EET". A navigation menu at the bottom includes "Inputs", "Registration summary", "Results", and "Log". The main content area is titled "Inputs to higher-level analysis" and contains a list of 25 lower-level FEAT reports, each starting with "/home/mayar/Data/...".

-The inputs tab shows the pre-processing and first level results for each feed directory for each subject.

The screenshot shows a web-based report titled "Higher-Level FEAT Report". At the top, it displays the command used: "/home/mayar/Data/sub-01/func/sub-01_task-flanker_run-1_bold+gfeat/report_reg.html". Below this, the text "Higher-Level FEAT Report" is centered, followed by the path and a timestamp: "Finished at 11 2023, 02:43:04 EET". A navigation menu at the bottom includes "Inputs", "Registration summary", "Results", and "Log". The main content area is titled "Summary of first-level registrations and masks" and contains a section titled "Summaries of functional-to-standard registrations for all inputs". It lists three entries: 1 /home/mayar/Data/..., 2 /home/mayar/Data/..., and 3 /home/mayar/Data/.... Each entry is accompanied by a row of 12 brain slices showing the registration and mask information.

-The registration summary tab shows how the functional data has been normalized to a template.

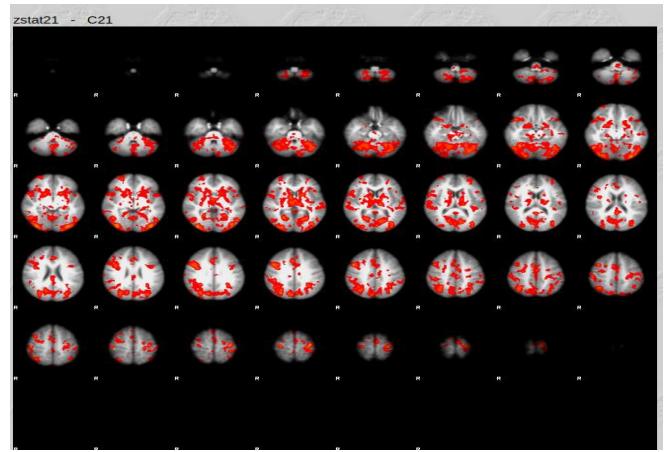
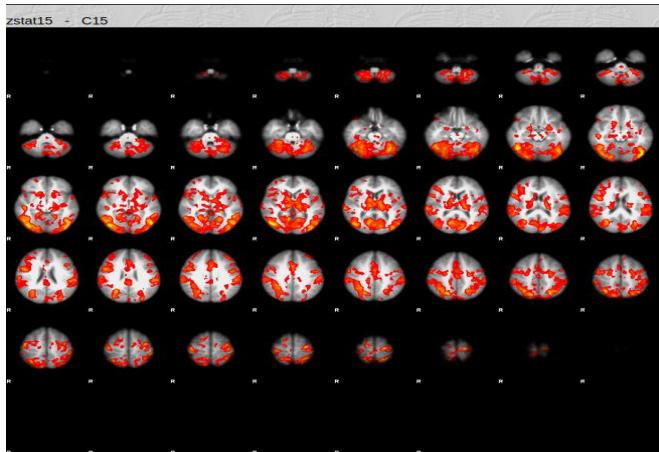
-The results tab will show you the GLM for the second level analysis along with results for each of the contrasts.



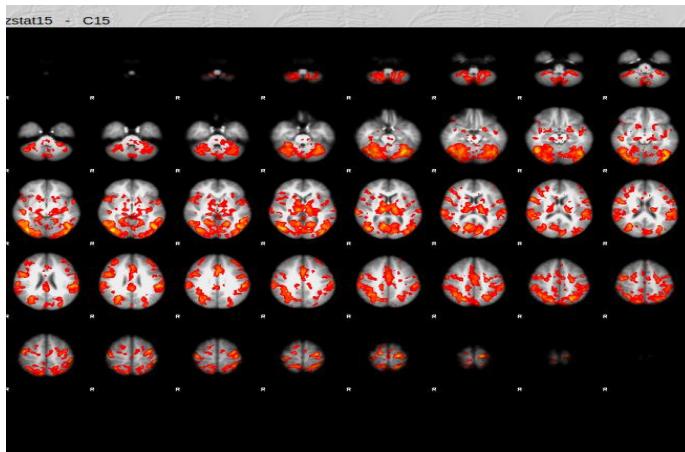
-Check them for any activation in the ventricles or strong activation at the edges of the brain which may suggest movement artifacts or other issues.

- Images with Artifact

Cobe1 (Incong)

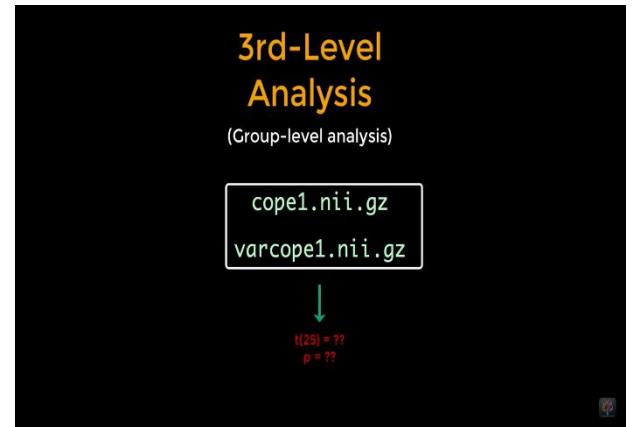
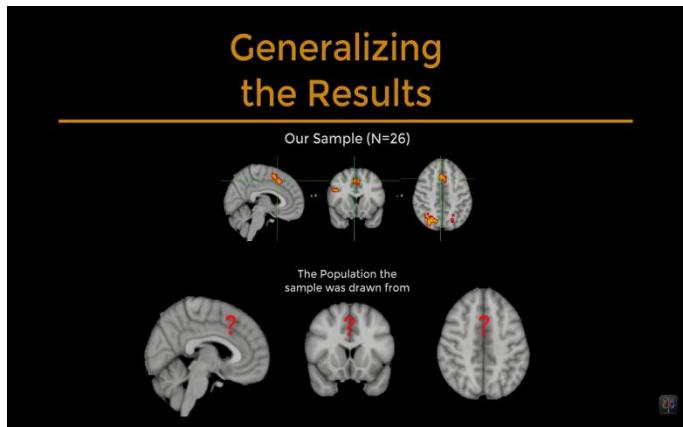


Cobe2 (Cong)



➤ 3rd Level Analysis

-Our goal in analyzing this data set is 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 in order to test this we will run a third level analysis which is a group level analysis we calculate the mean and the variance for a contrast estimate and then test whether that mean is statistically significant .



Steps:



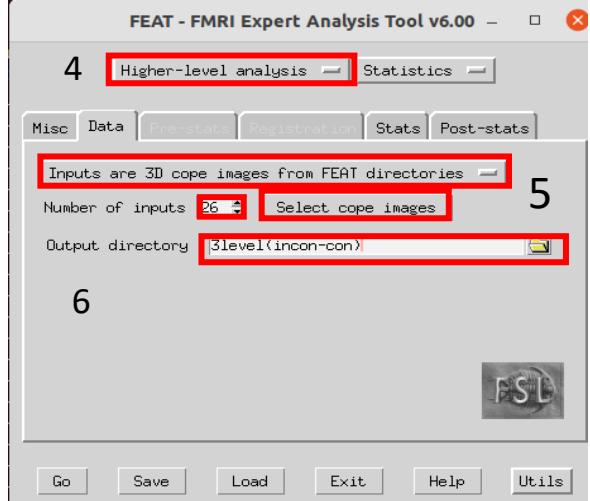
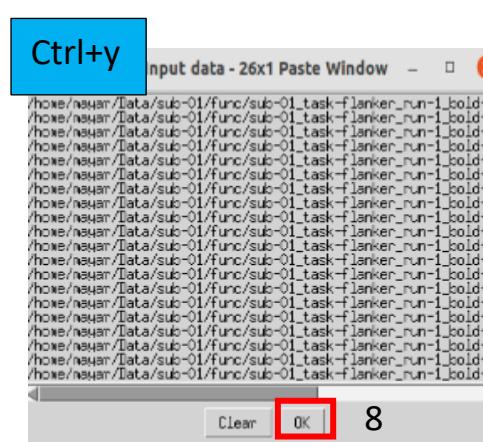
Select Input data

1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24:

7

Paste

OK



Select input data	
1:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
2:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
3:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
4:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
5:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
6:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
7:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
8:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
9:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
10:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
11:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
12:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
13:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
14:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
15:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
16:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
17:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
18:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
19:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
20:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
21:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
22:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-bold
23:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-b
24:	/home/kayar/Beta/sub-01/func/sub-01_task-flanker-run_1-b

The figure consists of three screenshots of the FEAT - FMRI Expert Analysis Tool v6.00 interface, arranged vertically. Each screenshot shows a different step in the analysis process.

Screenshot 10: The 'Model' tab is selected. A dropdown menu under 'Mixed effects' is open, showing options: 'Mixed effects: FLAME 1', 'Fixed effects', 'Mixed effects: Simple OLS', 'Mixed effects: FLAME 1', 'Mixed effects: FLAME 1+2', and 'Randomise'. The 'Mixed effects: FLAME 1' option is highlighted with a red box. A large blue arrow points downwards from this screen towards the explanatory text below.

Screenshot 11: The 'Statistics' tab is selected. A dropdown menu under 'single group average' is open, showing options: 'two groups, unpaired' and 'two groups, paired'. The 'two groups, unpaired' option is highlighted with a red box. A large blue arrow points downwards from this screen towards the explanatory text below.

Screenshot 12: The 'Process' button is highlighted with a red box. A large blue arrow points downwards from this screen towards the explanatory text below.

Screenshot 13: The 'Thresholding' section is shown. A dropdown menu under 'Cluster' is open, showing options: 'None', 'Uncorrected', 'Voxel', and 'Cluster'. The 'Cluster' option is highlighted with a red box. A large blue arrow points downwards from this screen towards the explanatory text below.

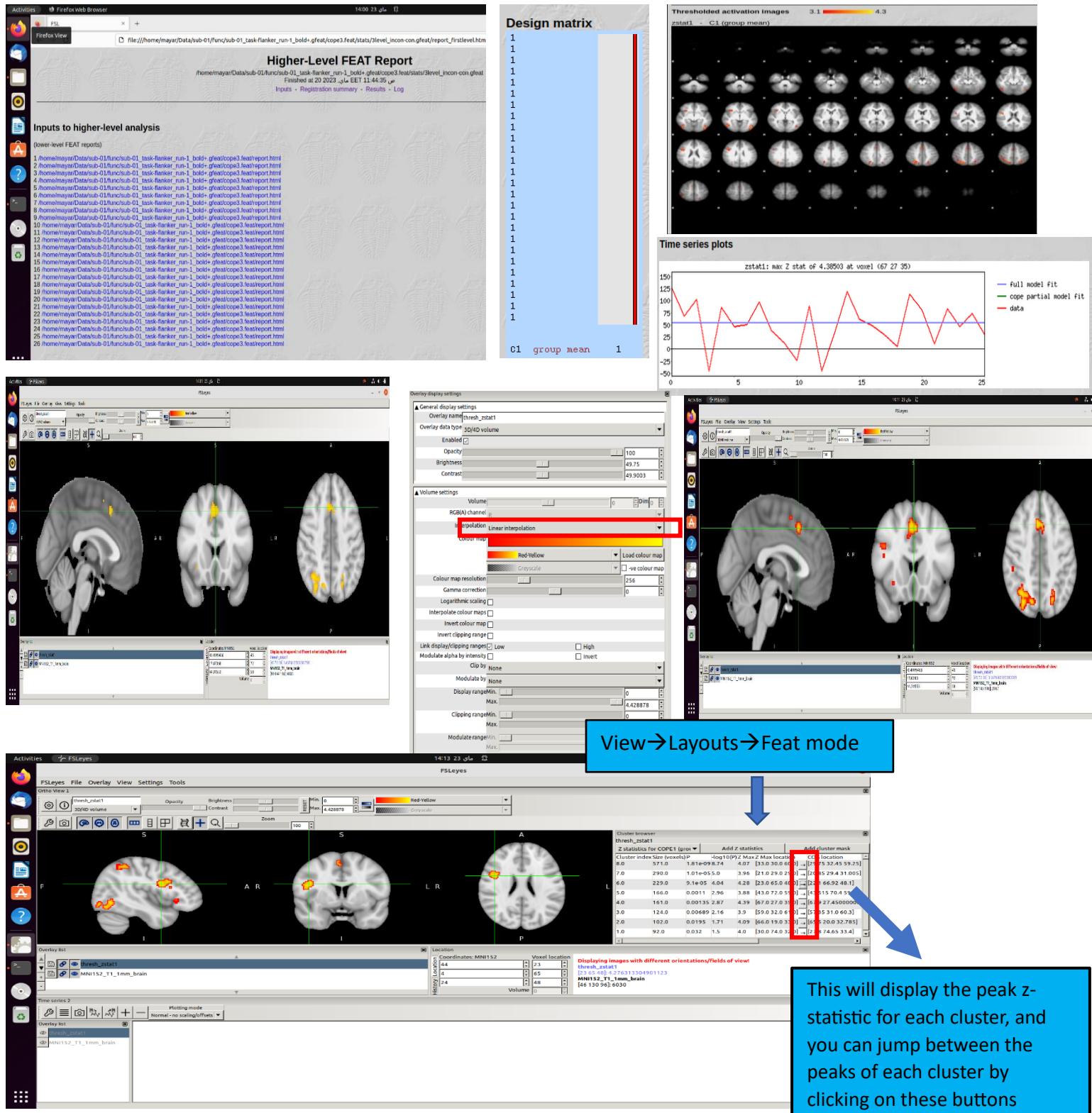
Screenshot 14: The 'Go' button is highlighted with a red box. A large blue arrow points downwards from this screen towards the explanatory text below.

We will use mixed effects this model of variance so that our result is generalizable to the population that our sample was drawn from flame 1f s l's local analysis of mixed effects estimates by using information about both within subject and between subject variability flame one plus 2 is considered more accurate but the additional benefit is usually minimal and it takes much longer, randomized is a non-parametric test.

-Uncorrect will allow for individual voxels to pass the threshold specified in Zthreshold which in this case only show voxels that have a value greater than a Z statistic of 3.1.
-voxel will perform a Bonferroni correction by dividing the alpha threshold by number of voxels.
-cluster who uses a cluster defining threshold to determine whether a cluster of voxels is significant the logic is that neighboring voxels are not independent of one another this reduced number of independent tests is considered.

➤ The output of 3rd Level Analysis

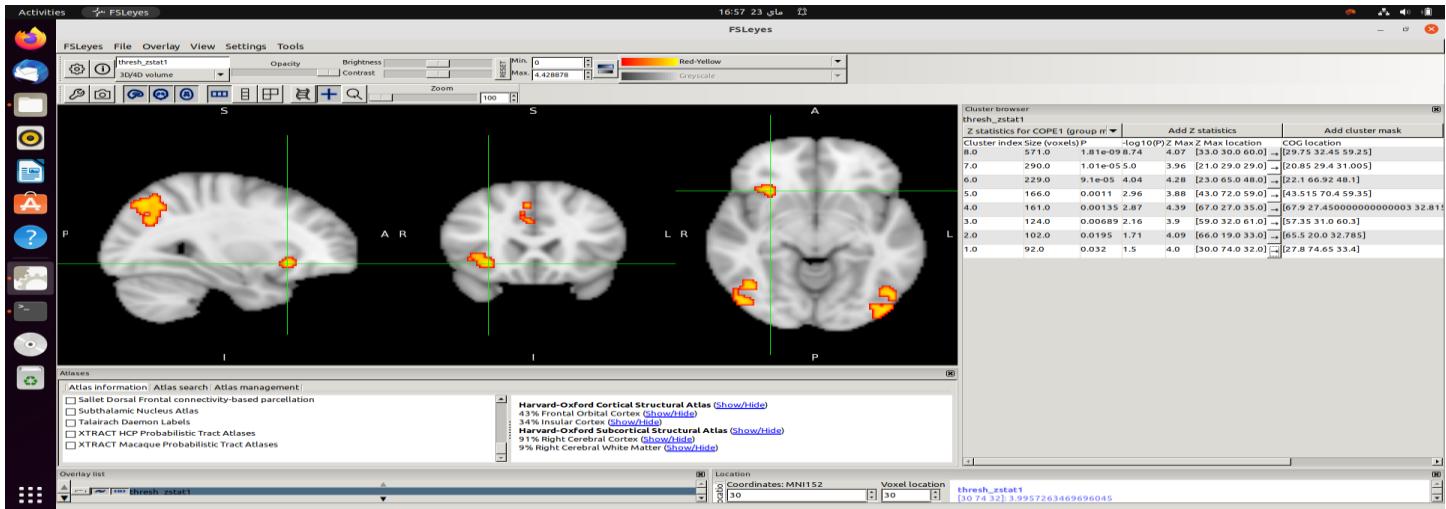
Cope3(incongruent -congruent)



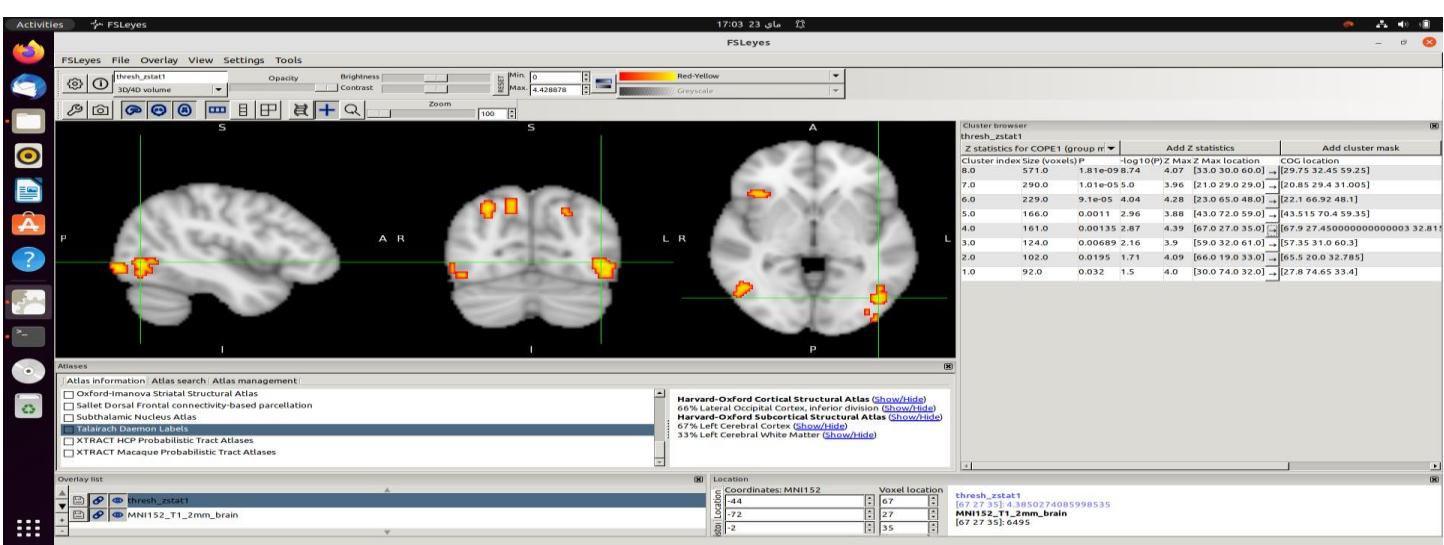
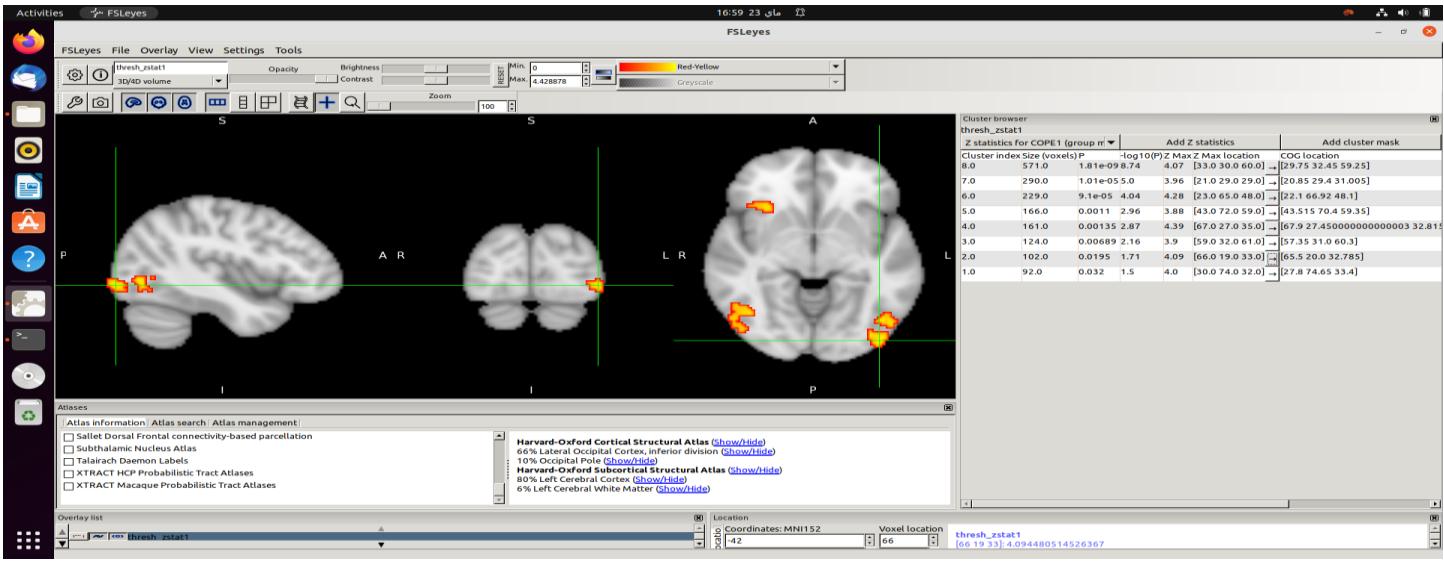
The brain regions activated during Incongruent minus congruent

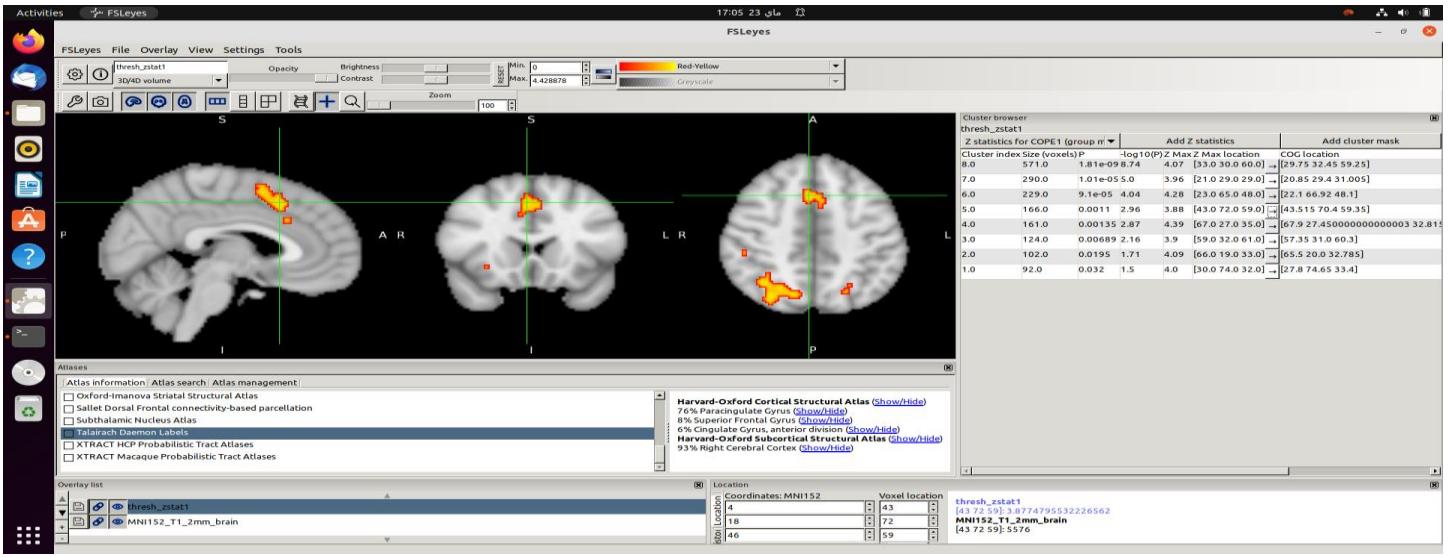
Cluster index	Coordinates: MNI152			Brain regions
	X	Y	Z	
1	30	22	-8	<ul style="list-style-type: none"> • 43% Frontal orbital cortex • 34% Insular cortex
2	-42	-88	-6	<ul style="list-style-type: none"> • 66% Lateral occipital cortex, inferior division • 10% Occipital pole
3	-28	-62	50	<ul style="list-style-type: none"> • 47% Lateral occipital cortex, superior division • 13% Superior parietal lobule
4	-44	-72	-2	<ul style="list-style-type: none"> • 66% Lateral occipital cortex, inferior division
5	4	18	46	<ul style="list-style-type: none"> • 76% Paracingulate gyrus • 8% Superior frontal gyrus • 6% Cingulate gyrus, anterior division
6	44	4	24	<ul style="list-style-type: none"> • 29% Precentral gyrus • 18% Inferior frontal gyrus, pars opercularis
7	48	-68	-14	<ul style="list-style-type: none"> • 68% Lateral occipital cortex, inferior division • 6% Occipital fusiform gyrus
8	24	-66	48	<ul style="list-style-type: none"> • 49% Lateral occipital cortex, superior division

Brain region	importance to the task(incongruent-congruent)
Frontal orbital cortex	<ul style="list-style-type: none"> it helps individuals to inhibit irrelevant information and focus on the target stimulus, particularly when presented with incongruent stimuli. It also plays a crucial role in a wide range of cognitive and behavioral processes and is associated with several psychiatric disorders.
Lateral occipital cortex, inferior division	<ul style="list-style-type: none"> it plays a critical role in processing visual information and determining the congruency of the flankers with the target stimulus. It also integrates visual information across different regions of the brain, allowing individuals to quickly process the stimuli and inhibit irrelevant information.
Lateral occipital cortex, superior division	<ul style="list-style-type: none"> is involved in other cognitive and behavioral processes, highlighting its importance in brain function.
Paracingulate gyrus	<ul style="list-style-type: none"> It is involved in the integration of information from other brain regions and the modulation of emotional responses to cognitive conflict.
Precentral gyrus	<ul style="list-style-type: none"> It plays a critical role in the response selection and execution processes required to indicate the correct response to the target stimulus while inhibiting the irrelevant information presented by the flankers. It is also involved in other cognitive and behavioral processes and has been linked to several neurological disorders.

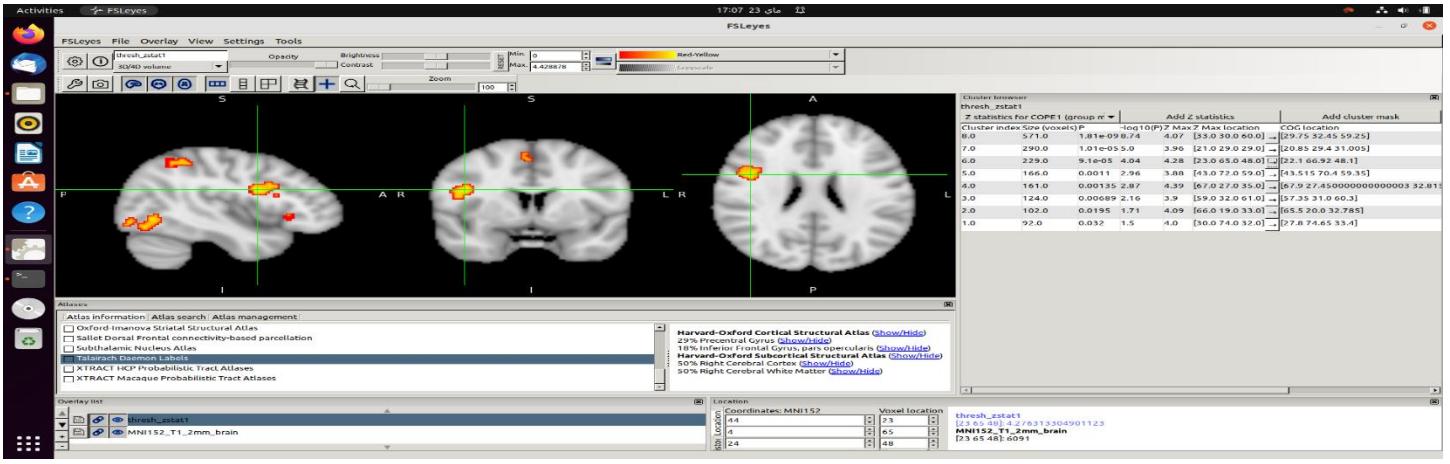


cluster1

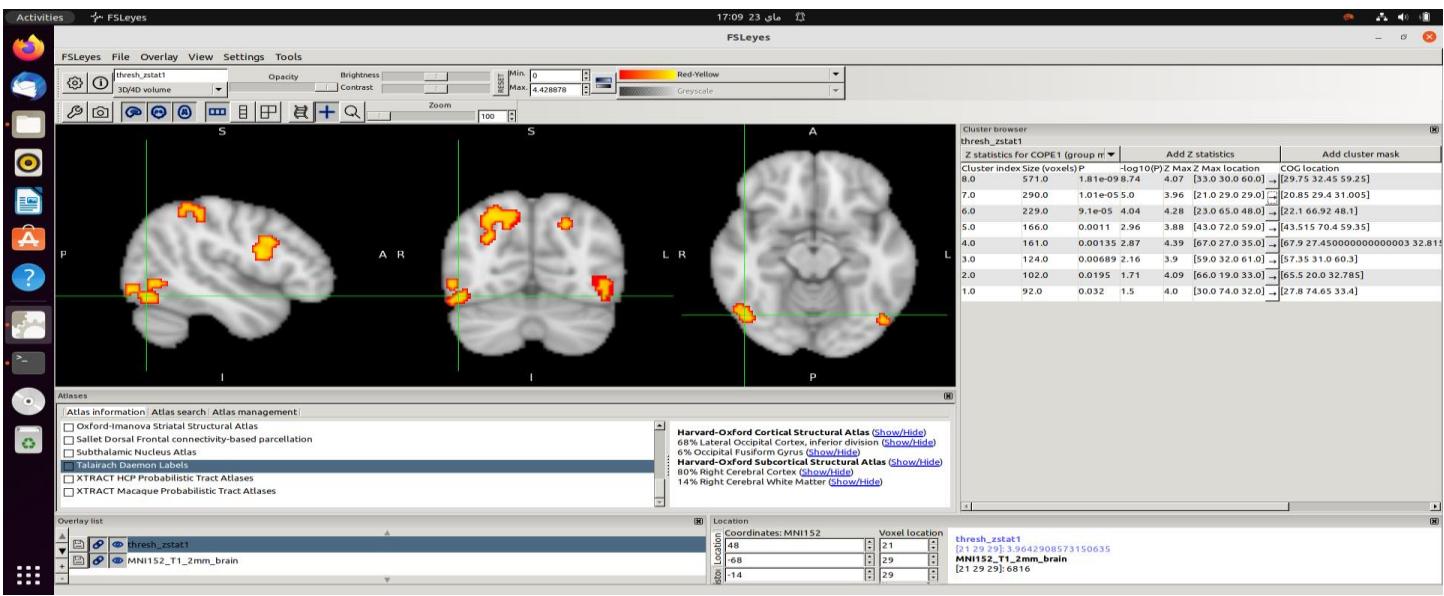




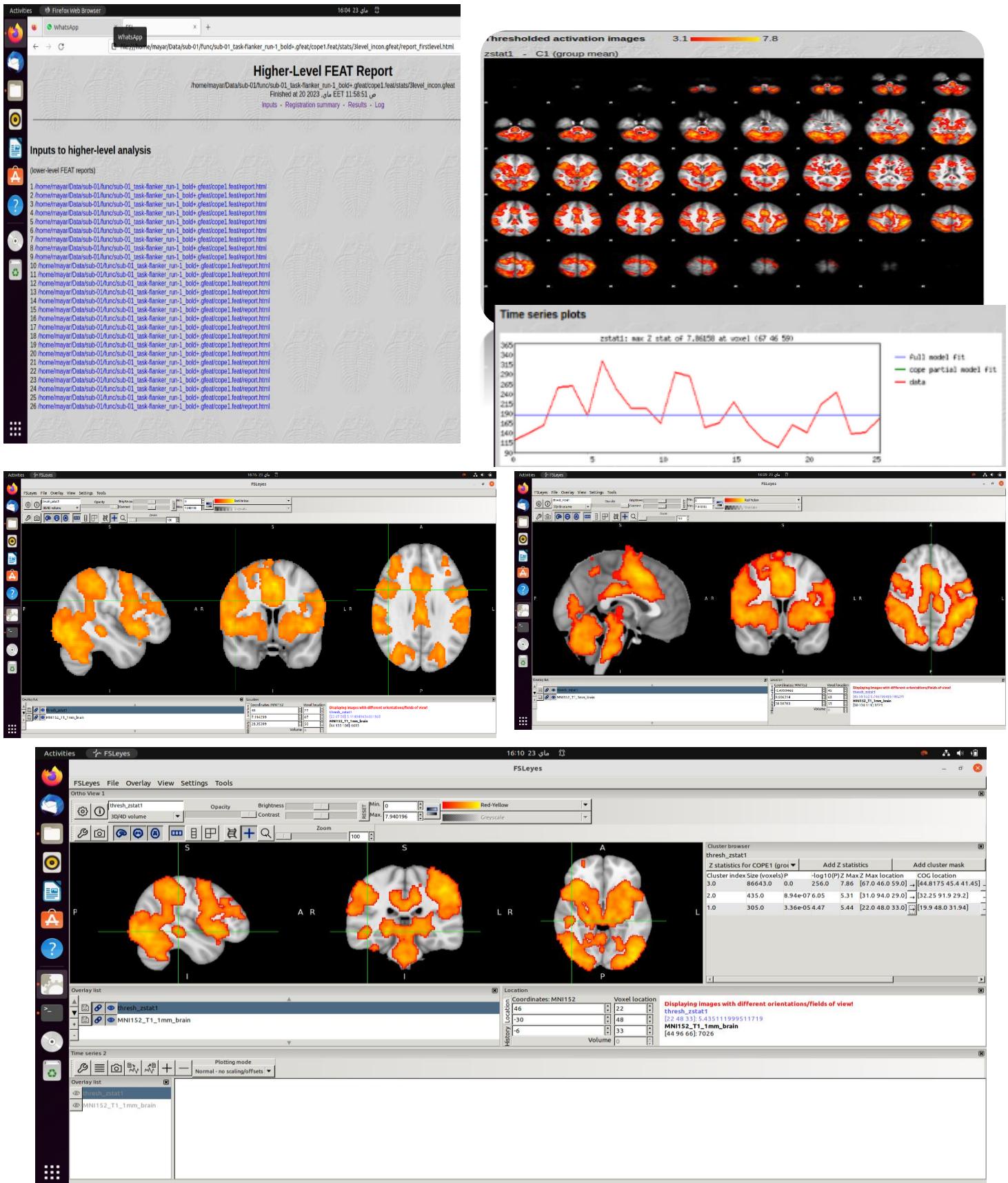
cluster5



cluster6

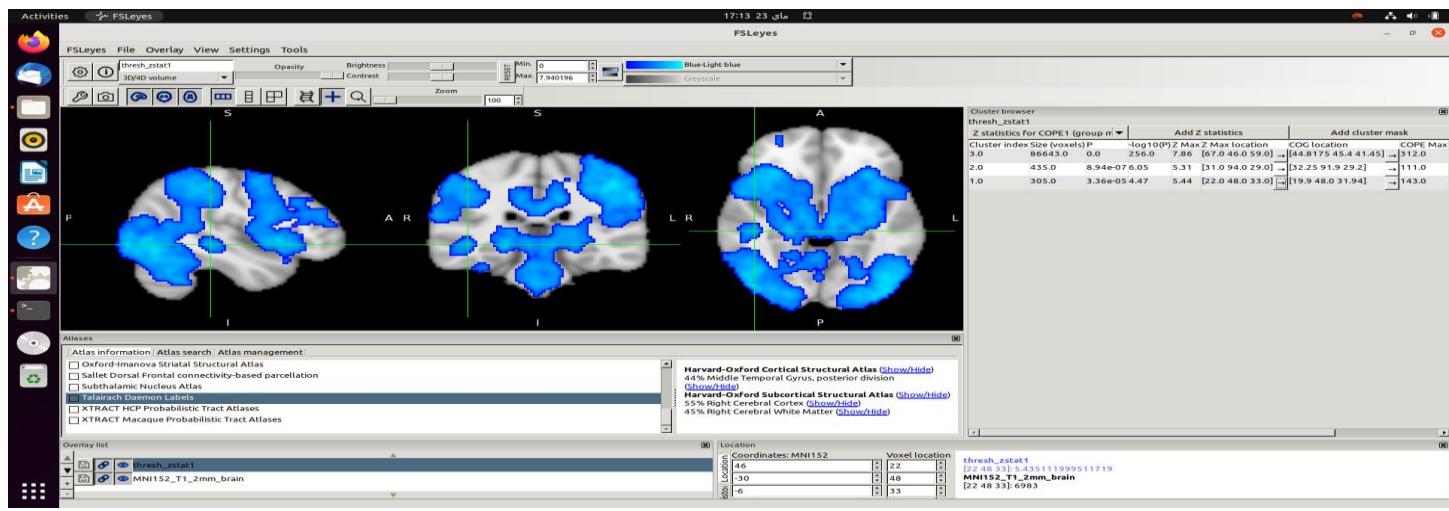


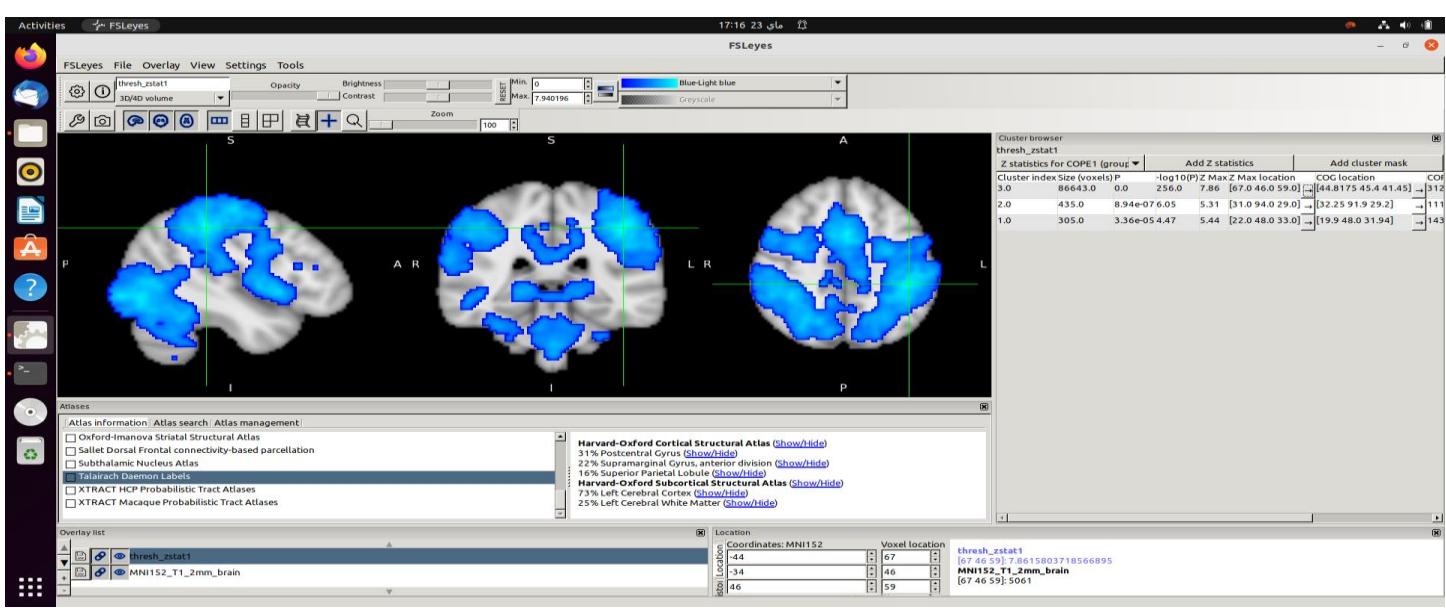
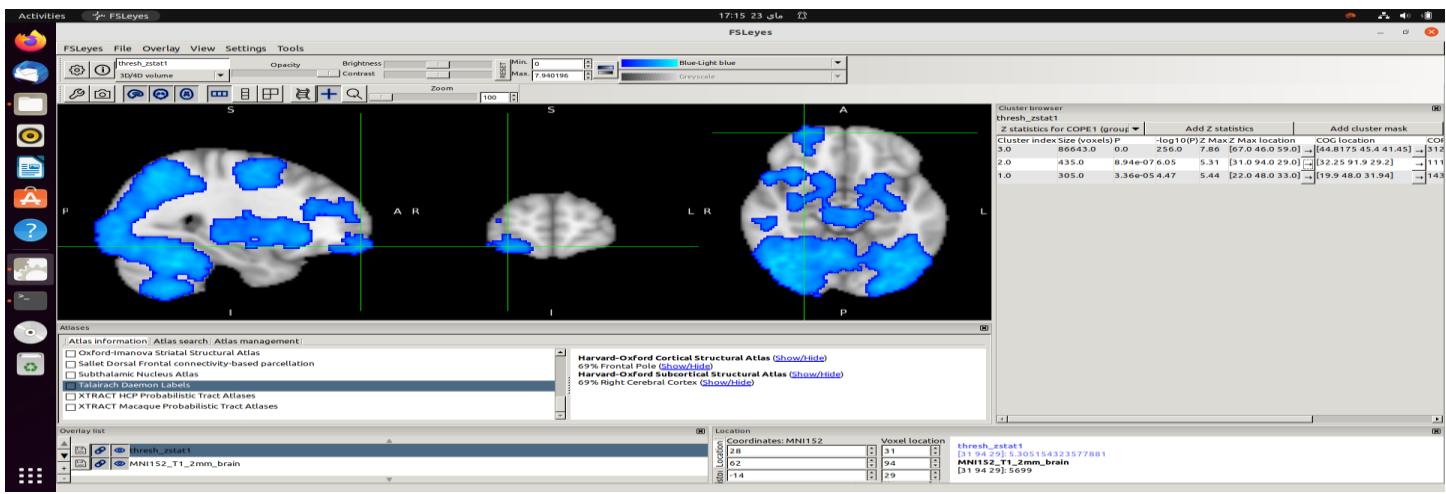
Cope1(incongruent)



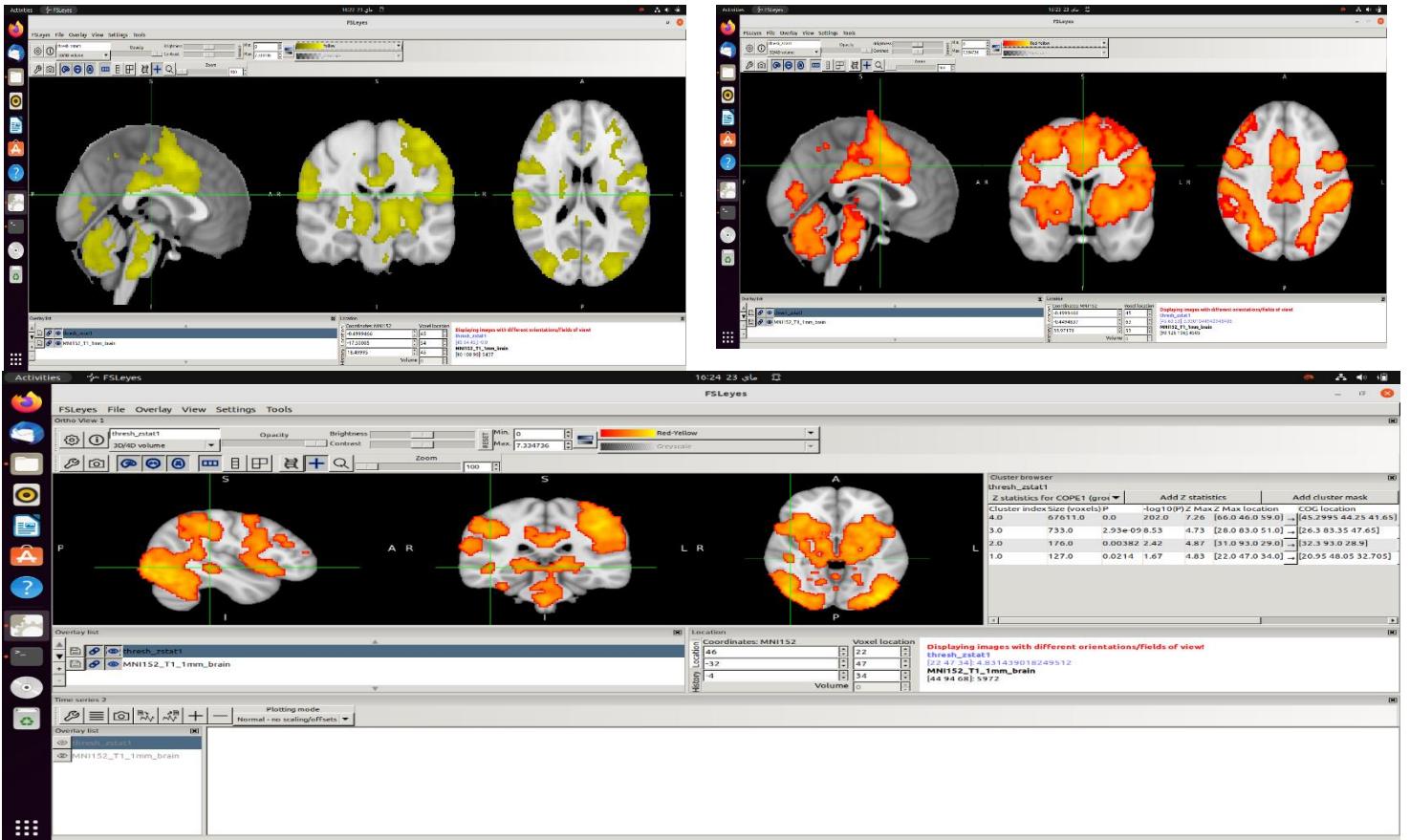
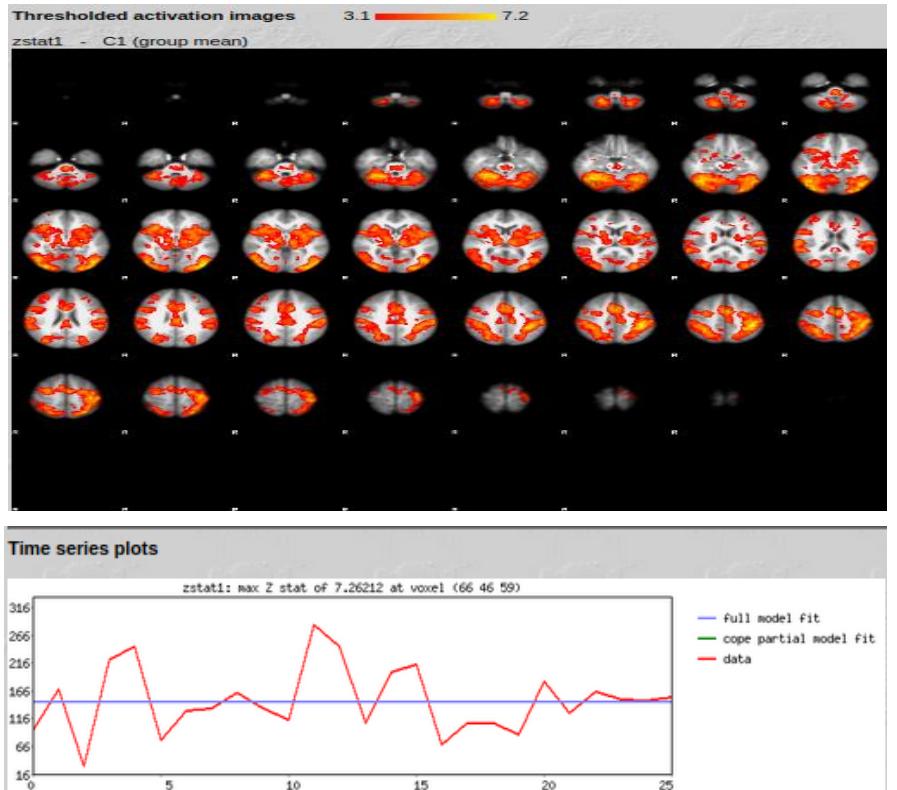
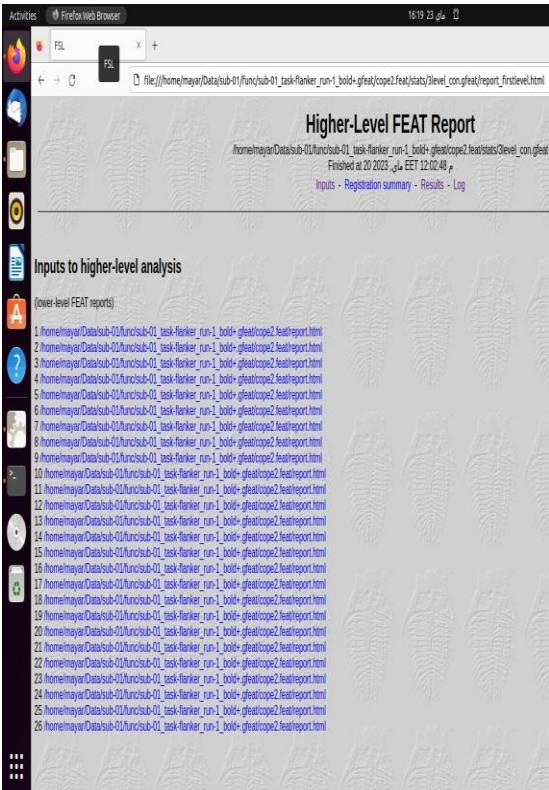
The brain regions activated during Incongruent

Cluster index	Coordinates: MNI152			Brain regions
	X	Y	Z	
1	46	-30	-6	<ul style="list-style-type: none"> • 44% Middle temporal gyrus, posterior division
2	28	62	-14	<ul style="list-style-type: none"> • 66% Frontal lobe
3	-44	-34	46	<ul style="list-style-type: none"> • 31% Postcentral gyrus • 22% Supramarginal gyrus, anterior division • 16% Superior posietal lobule



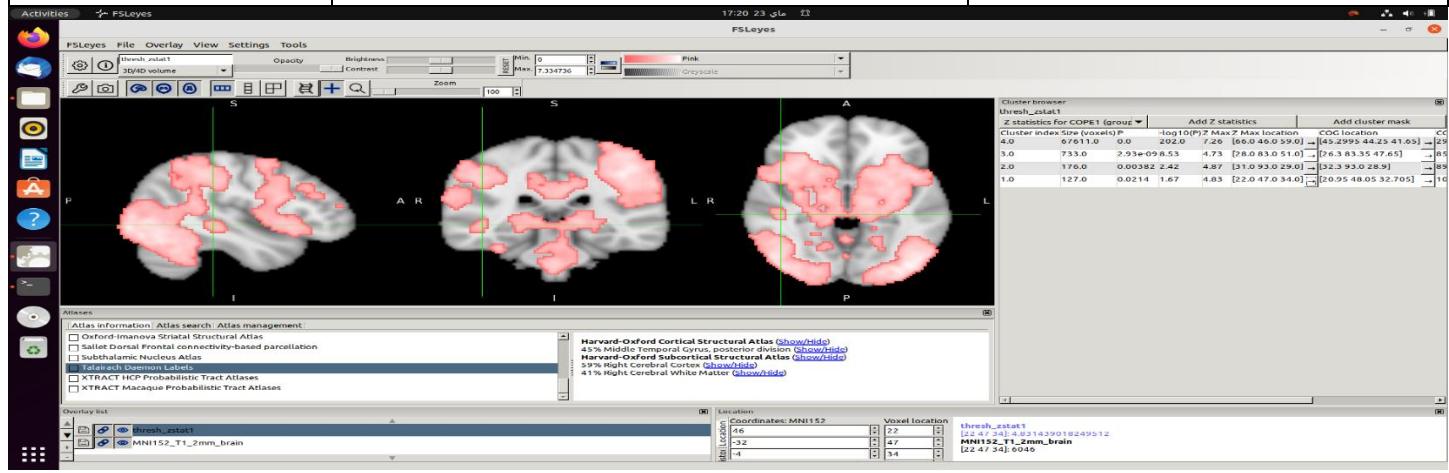


Cope2(congruent)

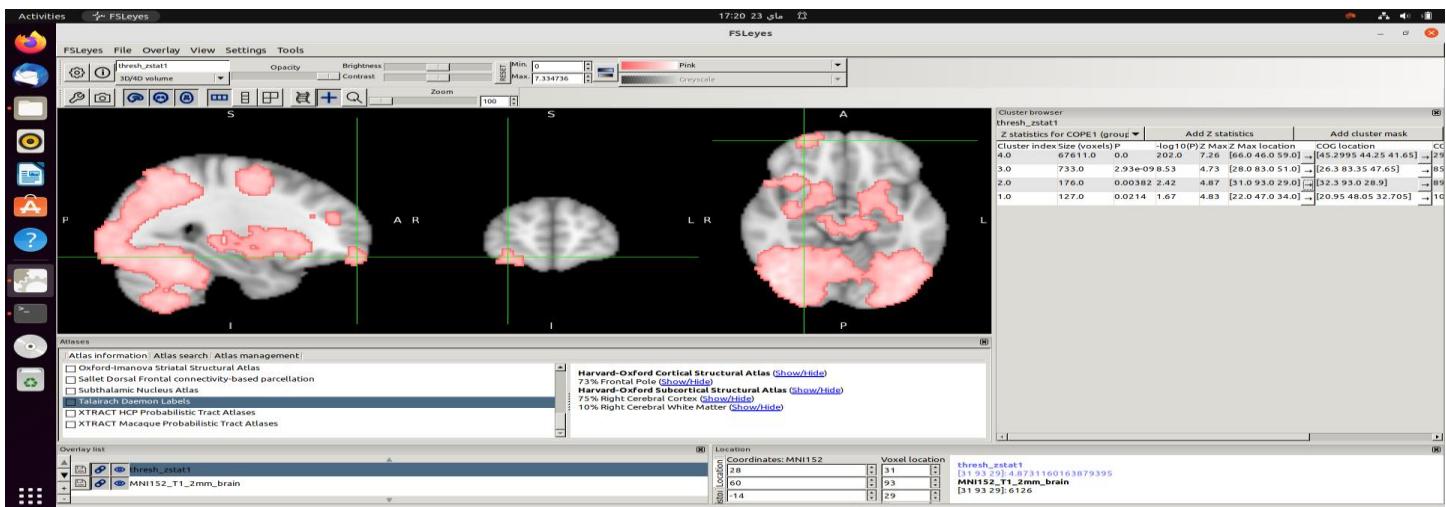


The brain regions activated during congruent

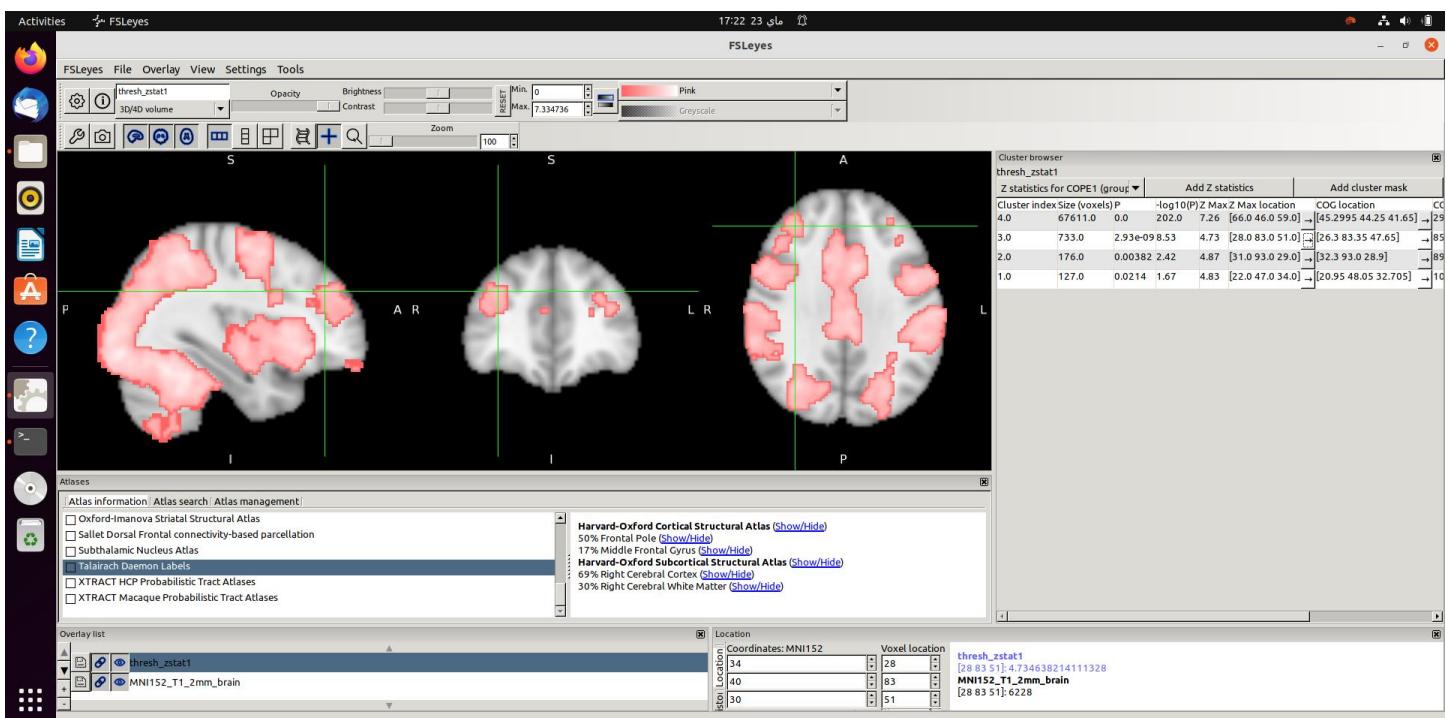
Cluster index	Coordinates: MNI152			Brain regions
	X	Y	Z	
1	-44	-34	46	<ul style="list-style-type: none"> • 45% Middle temporal gyrus, posterior division
2	28	60	-14	<ul style="list-style-type: none"> • 73% Frontal pole
3	34	40	30	<ul style="list-style-type: none"> • 50% Frontal pole • 17% Middle frontal gyrus
4	-42	-34	46	<ul style="list-style-type: none"> • 32% Postcentral gyrus • 17% Supramarginal gyrus, anterior division • 15% Superior posietal lobule



cluster1



cluster2

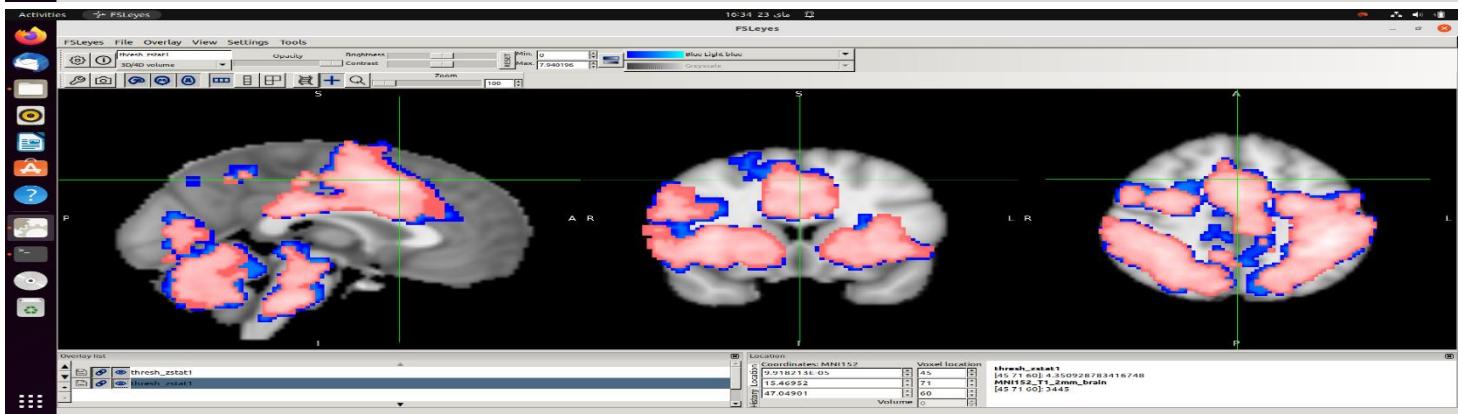
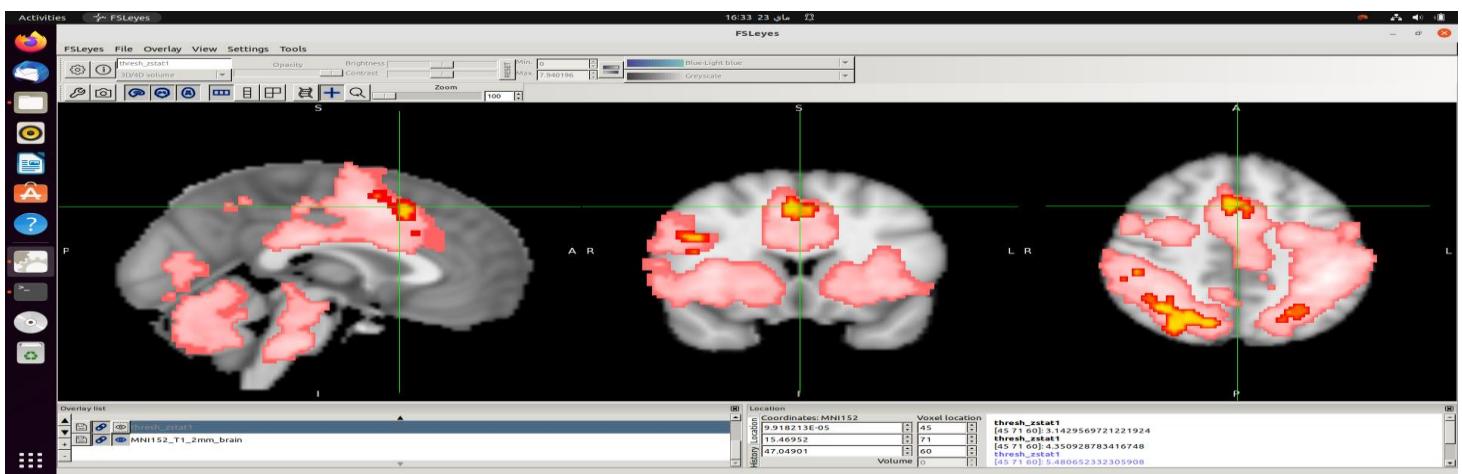
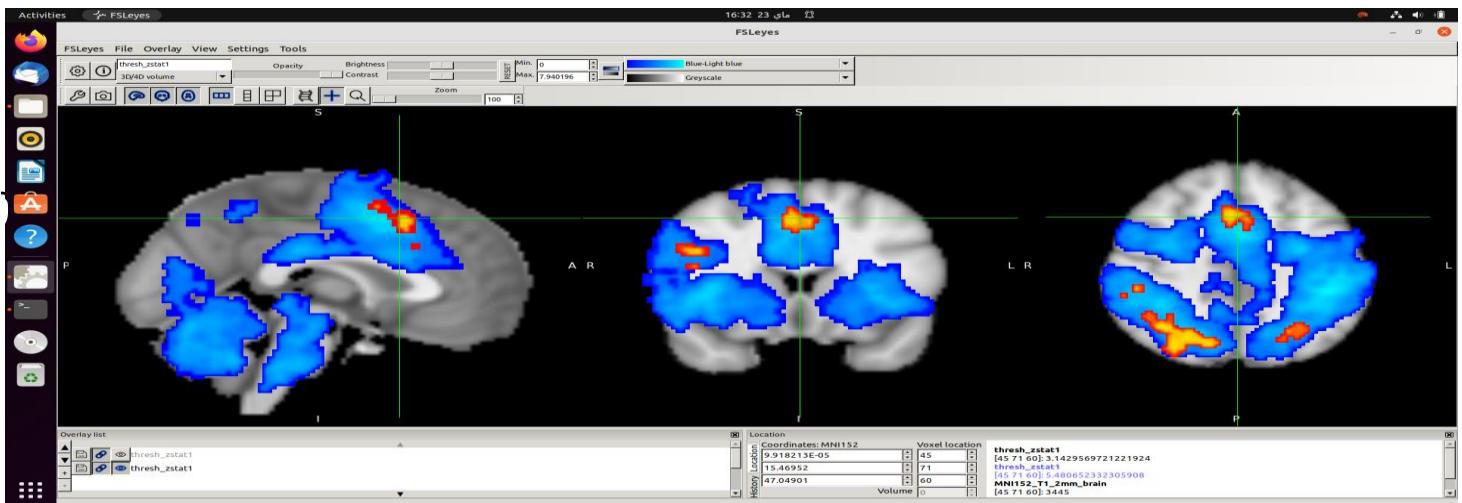


cluster3



cluster4

Group -level analysis



Incongruent



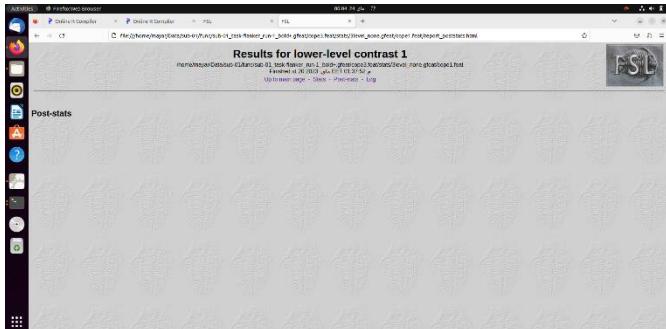
Incongruent - congruent



Congruent

Exercises

1. In the Post-stats tab, set the Thresholding to None, and re-run the analysis (changing the output directory to something that indicates that no threshold is being used). Examine the results in fsleyes. How do they compare to the cluster-corrected results?

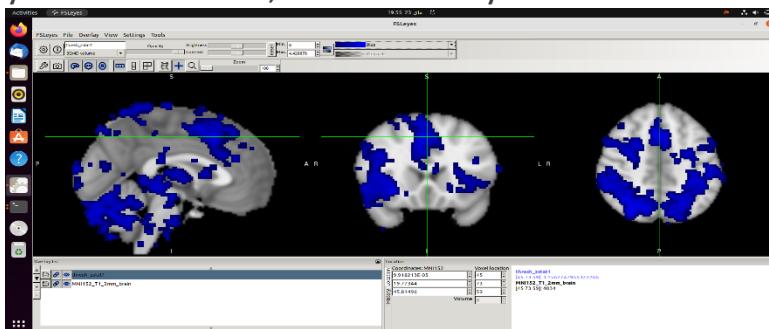


None

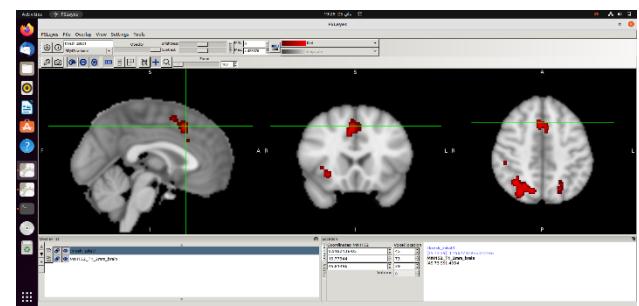
Cluster

We note that these None results do not contain active brain regions due to the lack of a threshold.

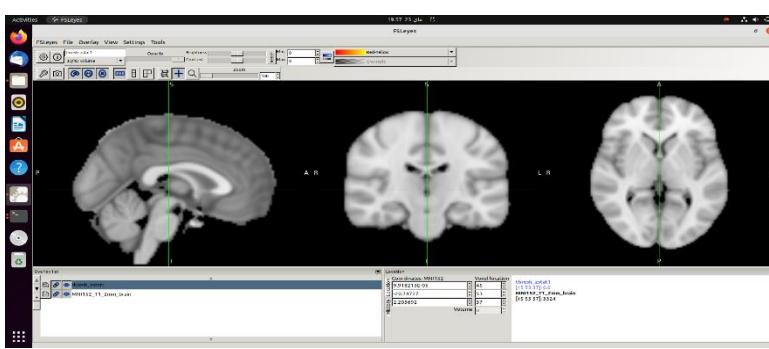
2. Do the same procedure in the previous exercise, this time using an Uncorrected threshold. Then, repeat the procedure with a Voxel threshold. Note any differences between these results and what you generated with the cluster corrected results. In your own words, describe why the results are different.



Uncorrected



cluster



Voxel

We note that uncorrected results contain more active brain regions than cluster results, but the Voxel results do not contain active brain regions.



Flame 1+2

We note that Flame 1+2 is more accurate than Flame 1.



Flame1

➤ ROI Analysis

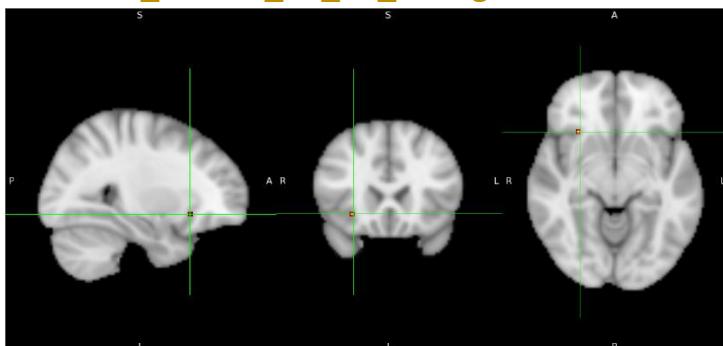
Extracting Data from a Sphere

The spherical ROI approach is another method. In this instance, a triplet of specified x-, y-, and z-coordinates serves as the center of a sphere with a particular diameter. Frequently, these coordinates are derived from the peak activation of another study that makes use of the same experimental design as you are using, or one that is comparable. Given that the ROI is determined using data from a different study, this analysis is regarded as independent.

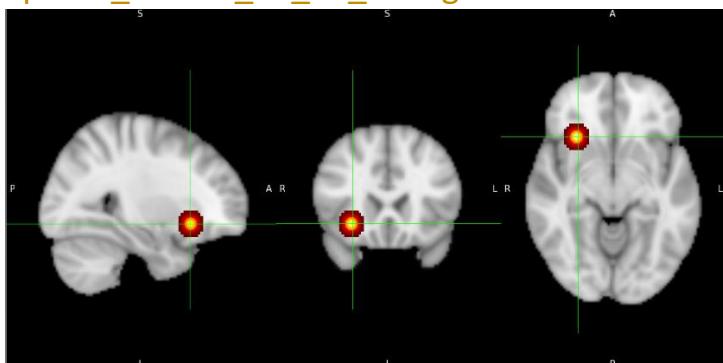
-We will need to locate the peak coordinates from a different study in order to generate this ROI; choose a study at random.

The next few steps are complicated, so pay close attention to each one:

1. Open fsleyes and load an MNI template. In the fields under the label “Coordinates: MNI152” in the Location window, type 30 22 -8 Just to the right of those fields, note the corresponding change in the numbers in the fields under the Voxel location. In this case, they are 30 74 32. Write down these numbers.
2. In the terminal, navigate to the Flanker directory and type the following:
`fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1 -roi 30 1 74 1 32
1 0 1 ROI_dmPFC_30_22_8.nii.gz -odt float`

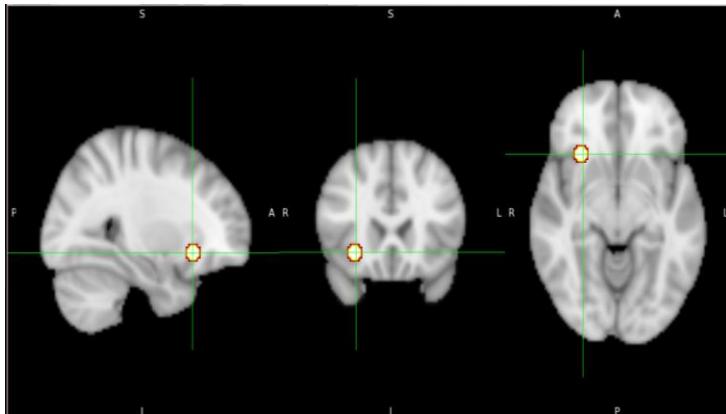


3. Next, type: `fslmaths ROI_dmPFC_30_22_8.nii.gz -kernel sphere 5 -fmean
Sphere_dmPFC_30_22_8.nii.gz -odt float`



4. Now, type: `fslmaths Sphere_dmPFC_30_22_8.nii.gz -bin`

`Sphere_bin_dmPFC_30_22_8.nii.gz`



5. Lastly, we will extract data from this ROI by typing: `fslmeants -i allZstats.nii.gz -m`

`Sphere_bin_dmPFC_30_22_8.nii.gz.`

```
0.087953
0.314823
1.269009
0.488194
0.115387
0.575384
1.294173
2.035314
-0.788351
-0.051404
1.737985
0.495685
1.642745
1.875126
1.550314
-0.355474
-0.010161
0.240529
0.313567
1.595331
0.546794
1.093009
-0.552439
0.844148
1.334179
0.159106
```



R

Example of using R compiler to get P-value and mean:

The R code in the main.r file is as follows:

```
1 x <- c(
2  0.087953 ,
3  0.314823 ,
4  1.269009 ,
5  0.488194 ,
6  0.115387 ,
7  0.575384 ,
8  1.294173 ,
9  2.035314 ,
10 -0.788351 ,
11 -0.051404 ,
12 1.737985 ,
13 0.495685 ,
14 1.642745 ,
15 1.875126 ,
16 1.550314 ,
17 -0.355474 ,
18 -0.010161 ,
19 0.240529 ,
20 0.313567 ,
21 1.595331 ,
22 0.546794 ,
23 1.093009 ,
24 -0.552439 ,
25 0.844148 ,
26 1.334179 ,
27 0.159106 )
28 t.test(x)
```

The output pane shows the results of the t-test:

```
Rscript /tmp/WG13hJVz7b.r
One Sample t-test
data: x
t = 4.444, df = 25  p-value = 0.0001575
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
0.3683876 1.0047605
sample estimates:
mean of x
0.6865741
```

COPE 1 (Incongruent)

Rscript /tmp/WGI3hJVz7b.r

One Sample t-test

1

```
data: x
t = 8.3339, df = 25, p-value = 1.106e-08
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
1.480809 2.452945
sample estimates:
mean of x
1.966877
```

One Sample t-test

2

```
data: x
t = 15.116, df = 25, p-value = 4.407e-14
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
3.354805 4.413189
sample estimates:
mean of x
3.883997
```

Rscript /tmp/WGI3hJVz7b.r

One Sample t-test

3

```
data: x
t = 10.545, df = 25, p-value = 1.087e-10
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
2.26892 3.37029
sample estimates:
mean of x
2.819605
```

One Sample t-test

4

```
data: x
t = 11.671, df = 25, p-value = 1.301e-11
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
3.412573 4.875094
sample estimates:
mean of x
4.143833
```

One Sample t-test

5

```
data: x
t = 9.3762, df = 25, p-value = 1.157e-09
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
1.775436 2.774961
sample estimates:
mean of x
2.275199
```

One Sample t-test

6

```
data: x
t = 7.2688, df = 25, p-value = 1.288e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
1.766119 3.162631
sample estimates:
mean of x
2.464375
```

One Sample t-test

7

```
data: x
t = 10.882, df = 25, p-value = 5.673e-11
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
2.862382 4.198827
sample estimates:
mean of x
3.530604
```

One Sample t-test

8

```
data: x
t = 9.7328, df = 25, p-value = 5.522e-10
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
2.241431 3.444646
sample estimates:
mean of x
2.843038
```

COPE2(Congruent)

One Sample t-test

1

```
data: x
t = 6.4076, df = 25, p-value = 1.042e-06
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.9011908 1.7549132
sample estimates:
mean of x
 1.328052
```

One Sample t-test

2

```
data: x
t = 13.067, df = 25, p-value = 1.132e-12
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 2.693306 3.701152
sample estimates:
mean of x
 3.197229
```

One Sample t-test

3

```
data: x
t = 8.3373, df = 25, p-value = 1.098e-08
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.587755 2.629541
sample estimates:
mean of x
 2.108648
```

One Sample t-test

4

```
data: x
t = 9.7375, df = 25, p-value = 5.47e-10
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 2.717014 4.174636
sample estimates:
mean of x
 3.445825
```

One Sample t-test

5

```
data: x
t = 7.2417, df = 25, p-value = 1.374e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.109309 1.991046
sample estimates:
mean of x
 1.550177
```

One Sample t-test

6

```
data: x
t = 5.5342, df = 25, p-value = 9.42e-06
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.043210 2.279904
sample estimates:
mean of x
 1.661557
```

One Sample t-test

7

```
data: x
t = 10.037, df = 25, p-value = 2.977e-10
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 2.271854 3.444929
sample estimates:
mean of x
 2.858392
```

One Sample t-test

8

```
data: x
t = 7.1225, df = 25, p-value = 1.826e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.443357 2.617638
sample estimates:
mean of x
 2.030498
```

COPE3(Incongruent-Congruent)

```
One Sample t-test  
data: x  
t = 4.444, df = 25, p-value = 0.0001575  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.3683876 1.0047605  
sample estimates:  
mean of x  
0.6865741
```

1

```
One Sample t-test  
data: x  
t = 6.3515, df = 25, p-value = 1.197e-06  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.5107747 1.0009743  
sample estimates:  
mean of x  
0.7558745
```

2

```
One Sample t-test  
data: x  
t = 4.5611, df = 25, p-value = 0.0001162  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.3987814 1.0554106  
sample estimates:  
mean of x  
0.727096
```

3

```
One Sample t-test  
data: x  
t = 5.2518, df = 25, p-value = 1.945e-05  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.4628854 1.0601635  
sample estimates:  
mean of x  
0.7615245
```

4

```
One Sample t-test  
data: x  
t = 4.9313, df = 25, p-value = 4.452e-05  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.4416393 1.0750931  
sample estimates:  
mean of x  
0.7583662
```

5

```
One Sample t-test  
data: x  
t = 6.2004, df = 25, p-value = 1.745e-06  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.575547 1.148063  
sample estimates:  
mean of x  
0.8618047
```

6

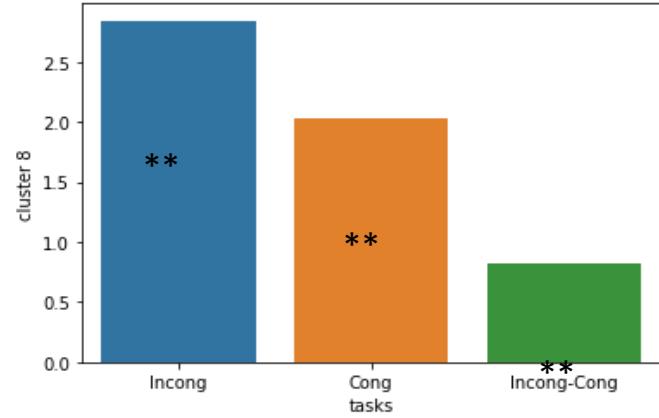
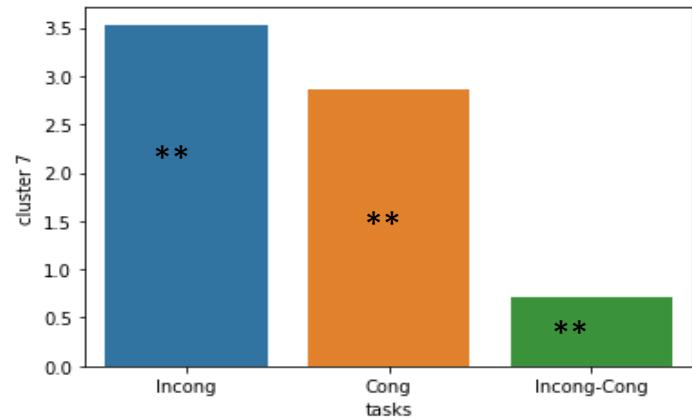
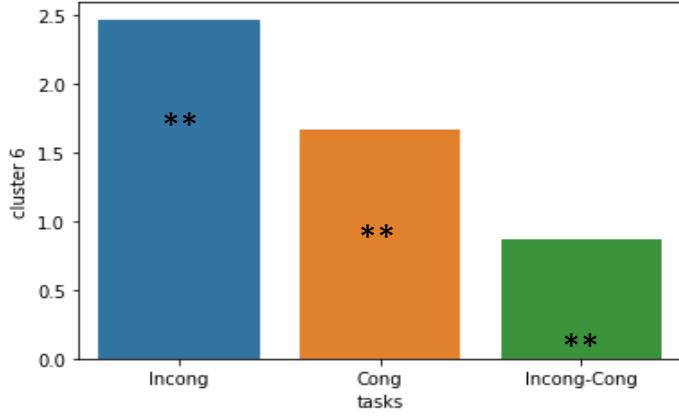
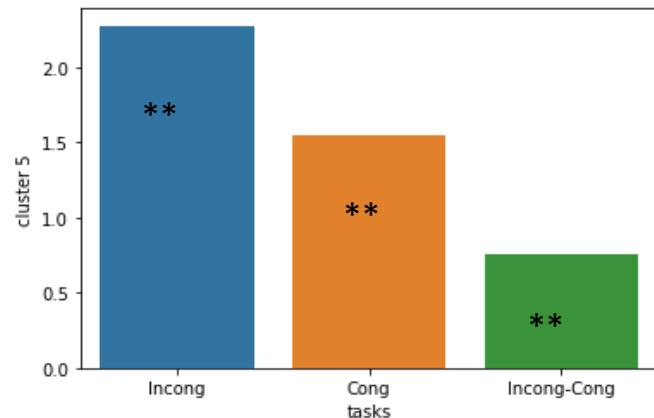
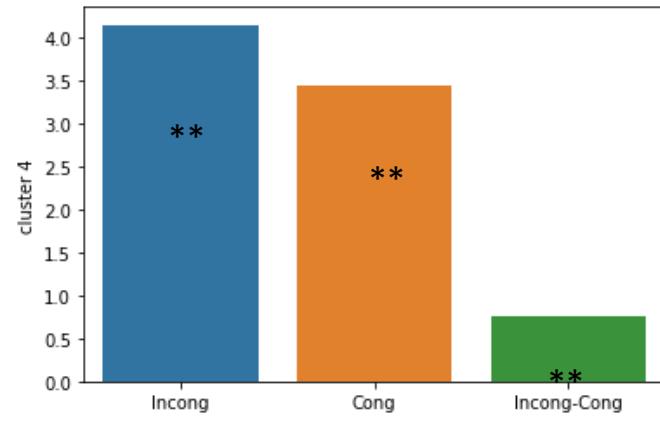
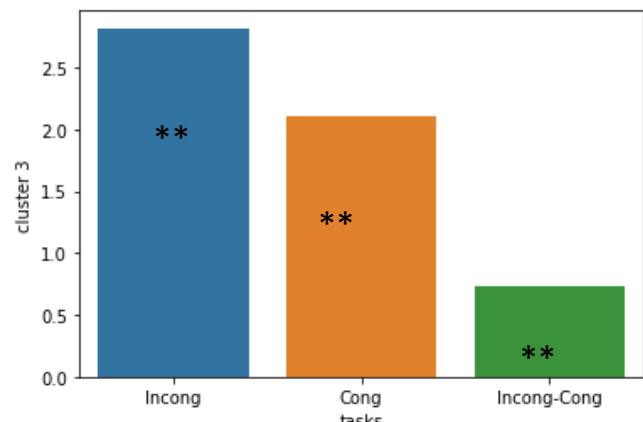
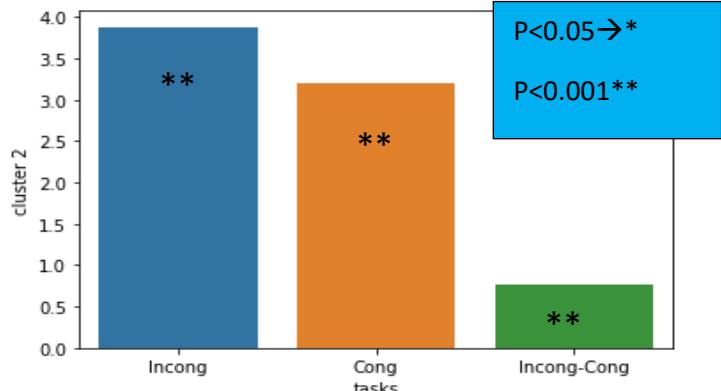
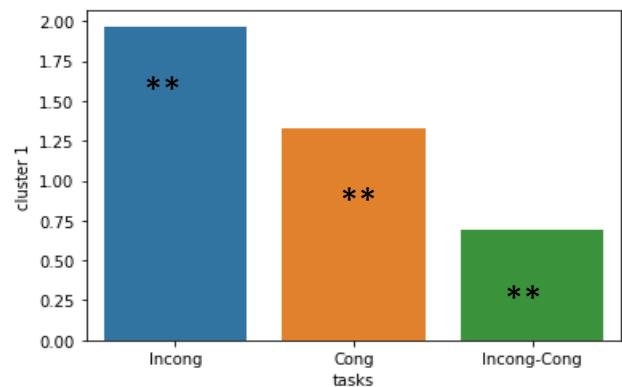
```
One Sample t-test  
data: x  
t = 5.9052, df = 25, p-value = 3.665e-06  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.4634751 0.9599001  
sample estimates:  
mean of x  
0.7116876
```

7

```
One Sample t-test  
data: x  
t = 6.1922, df = 25, p-value = 1.781e-06  
alternative hypothesis: true mean is not equal to 0  
95 percent confidence interval:  
 0.550537 1.099269  
sample estimates:  
mean of x  
0.824903
```

8

Barcharts of mean of (Incongruent ,Congruent ,Incongruent-Congruent)

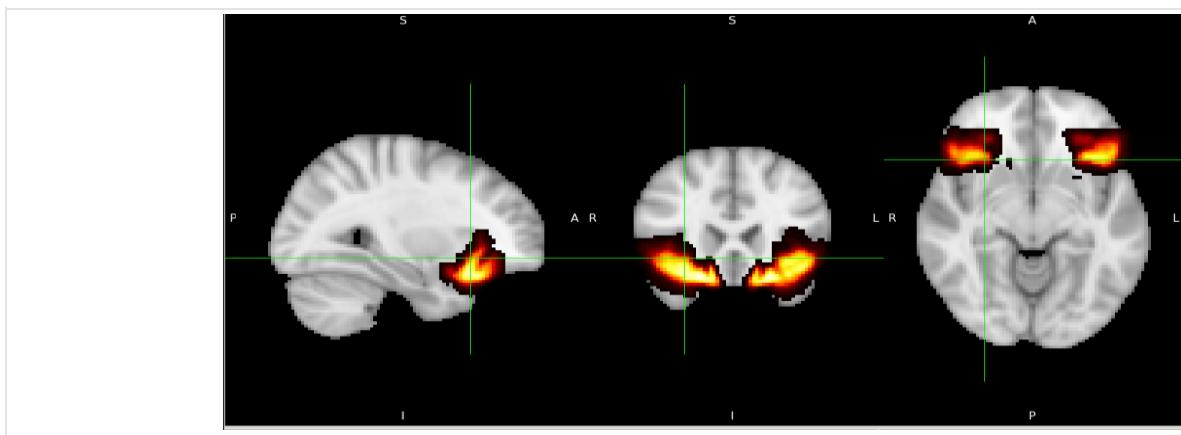


Using Atlases

One way to create a region for our ROI analysis is to use an **atlas**, or a map that partitions the brain into anatomically distinct regions.

You can access the several atlases that FSL already provides by using the FSL viewer. You may access the Atlas Panel window by selecting Settings -> Ortho View 1 -> Atlas Panel. The Harvard-Oxford Cortical and Subcortical Atlases are loaded by default. By selecting the Show/Hide button next to the atlas' name, you may view how the atlas divides the brain. A likelihood of being a brain structure will be assigned to the voxel in the viewing window's crosshairs.

To save one of these regions as a file to extract data from, also known as a **mask**, click on the **Show/Hide** link next to the region you want to use as a mask - in our example, let's say that we want to use the Frontal orbital cortex as a mask. Clicking on the link will show that region overlaid on the brain, as well as load it as an overlay in the Overlay List window. Click on the disk icon next to the image to save it as a mask. Save it to the Flanker directory and call it **FOC.nii.gz**

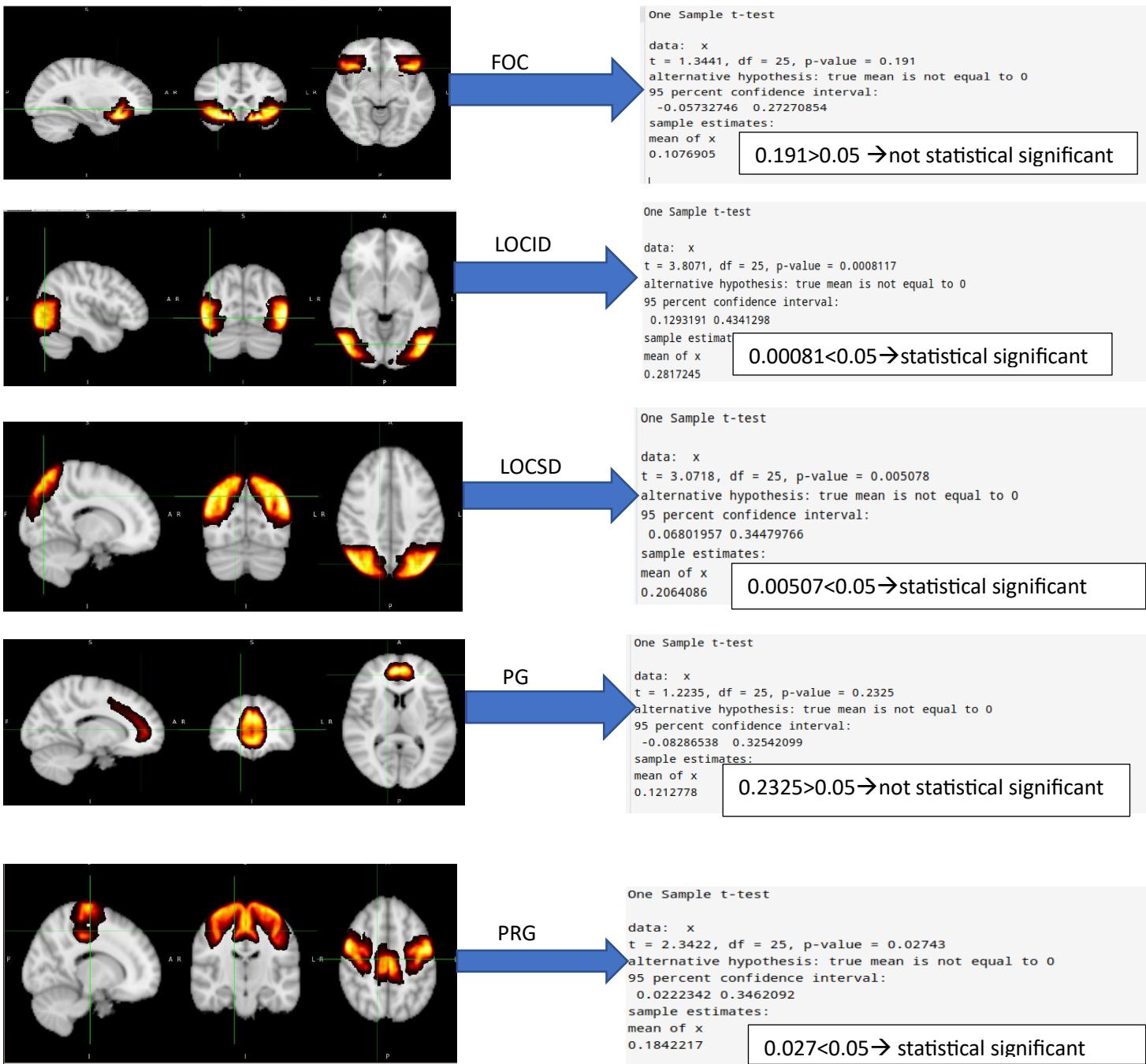


Extracting Data from an Anatomical Mask

1. The objective of ROI analysis is to extract the contrast estimates for each subject individually, not from the 3rd-level analysis.
2. For the Incongruent-Congruent contrast estimate, for example, you can find each subjects' data maps in the directory Flanker_2ndLevel.gfeat/cope3.feat/stats. The data maps have been calculated several different ways, including t-statistic maps, cope images, and variance images.

3. It is ideal to take data from the z-statistic maps because these data have been transformed into a normally distributed form, which is, in my opinion, simpler to plot and understand.
4. To make our ROI analysis easier, we will merge all of the z-statistic maps into a single dataset. To do this, we will use a combination of FSL commands and Unix commands. Navigate into the Flanker_2ndLevel.gfeat/cope3.feat/stats directory, and then type the following: `fslmerge -t allZstats.nii.gz `ls zstat* | sort -V``
5. Move the allZstats.nii.gz file up three levels so that it is in the main Flanker directory (i.e., type `mv allZstats.nii.gz ../../..`). Then use the `fslmeans` command to extract the data from the FOC mask: `fslmeans -i allZstats.nii.gz -m FOC.nii.gz`.
6. This will print 26 numbers, one per subject. Each number is the contrast estimate for that subject averaged across all the voxels in the mask.

The brain regions activated from the 3rd level for cope3:



Compare between the output of atlas ROI and MNI_ROI for cope 3:

0.356880	0.127534	0.116634	0.042374	0.370292	
-0.198616	0.070628	-0.206920	-0.292050	-0.217306	
-0.095058	0.126660	-0.021641	0.231808	0.173548	
-0.172071	-0.362022	-0.181234	-0.125119	-0.328466	
0.115090	0.436010	0.197034	-0.124917	0.183274	
0.130401	0.552109	0.315855	-0.049088	0.056999	
-0.009306	0.248215	0.336652	-0.350358	0.181511	
0.173898	0.543202	0.502030	0.264312	0.396913	
-0.902234	-0.427143	-0.110293	-0.838529	-0.151926	
-0.123678	0.109106	-0.117262	-0.358703	-0.307701	
1.102259	1.190217	1.049568	1.357093	0.751393	
-0.176598	-0.060467	-0.099793	-0.262661	-0.221342	
0.184918	0.207747	0.465282	0.204619	0.434767	
0.247161	0.096379	0.309448	0.428320	0.623650	
0.908645	0.997066	0.898754	0.893273	0.792102	
0.092345	0.031765	0.016804	-0.206089	0.152948	
-0.503508	0.121600	-0.128986	-0.519756	-0.345953	
0.003060	0.064358	-0.185968	0.131700	0.186650	
-0.212909	0.201518	-0.013811	0.221920	0.089198	
0.454993	0.649232	0.598371	1.062275	0.942590	
0.012887	0.186469	-0.015207	-0.207075	-0.392517	
0.078714	0.004888	0.048014	-0.236283	-0.427566	
0.248645	0.694544	0.350760	0.270046	0.602920	
0.483716	0.848795	0.718995	0.679171	0.694408	
0.628191	0.329210	0.229866	0.636238	0.342308	
-0.027871	0.337216	0.293672	0.300702	0.207070	
	FOC	LOCID	LOCSD	PG	PRG
					
	C1	C2	C3	C5	C6
					
0.087953	0.381751	1.377975	-0.611692	0.005338	
0.314823	0.699349	0.721264	0.967006	0.376511	
1.269009	1.048136	0.453193	1.610150	2.358631	
0.488194	1.063827	0.004855	-0.074008	0.214716	
0.115387	-0.546112	0.463784	-0.279419	0.364570	
0.575384	1.309439	0.990899	0.855094	1.572287	
1.294173	0.755169	2.077821	0.607472	0.914060	
2.035314	0.961141	1.063718	0.973687	0.663295	
-0.788351	-0.111686	1.199866	0.113776	0.107698	
-0.051404	0.589968	0.240504	-0.377307	0.049283	
1.737985	1.986013	2.324769	1.430575	2.287115	
0.495685	0.239941	-0.746512	-0.320046	-0.199661	
1.642745	1.755752	0.441074	1.186675	1.700399	
1.875126	0.098032	0.763923	0.955733	0.580292	
1.550314	1.320744	1.520045	1.531645	1.099275	
-0.355474	0.135951	1.309479	0.532294	0.504906	
-0.010161	0.326435	-0.769995	-0.287278	0.306141	
0.240529	0.545612	0.511820	0.607295	0.385535	
0.313567	0.193235	-0.442342	0.903719	0.731416	
1.595331	0.966368	1.212015	0.975167	1.619046	
0.546794	1.968219	1.018309	1.360915	1.141252	
1.093009	0.740817	-0.577330	0.836919	1.325342	
-0.552439	0.946883	0.818078	1.737747	0.781134	
0.844148	0.795825	1.853603	2.322679	1.730201	
1.334179	0.891072	0.818782	1.951299	1.441622	
0.159106	0.590857	0.254895	0.207424	0.346519	

Exercises

1. The mask used with fslmeans is binarized, meaning that any voxel containing a numerical value greater than zero will be converted to a “1”, and then data will be extracted only from those voxels labeled with a “1”. You will recall that the mask created with fsleyes is probabilistic. If you want to weight the extracted contrast estimates by the probability weight, you can do this by using the -w option with fslmeans. Try typing: fslmeans -i allZstats.nii.gz -m PCG.nii.gz -w and observe how the numbers are different from the previous method that used a binarized mask. Is the difference small? Large? Is it that you would expect?

```
mayar@mayar-VirtualBox:~/Data$ fslmeans -i allzstats.nii.gz -m PCG.nii.gz
0.042374
-0.292050
0.231808
-0.125119
-0.124917
-0.049088
-0.350358
0.264312
-0.838529
-0.358703
1.357093
-0.262661
0.204619
0.428320
0.893273
-0.206089
-0.519756
0.131700
0.221920
1.062275
-0.207075
-0.236283
0.270046
0.679171
0.636238
0.300702
```

```
mayar@mayar-VirtualBox:~/Data$ fslmeans -i allZstats.nii.gz -m PCG.nii.gz -w
0.034069
-0.284553
0.370655
-0.201239
-0.350743
-0.008618
-0.376839
0.352596
-1.004073
-0.234025
1.527951
-0.247431
0.286447
0.502183
0.921077
-0.306132
-0.562197
0.054951
0.416264
1.062533
-0.164290
-0.131319
0.211018
0.823812
0.845165
0.312397
```

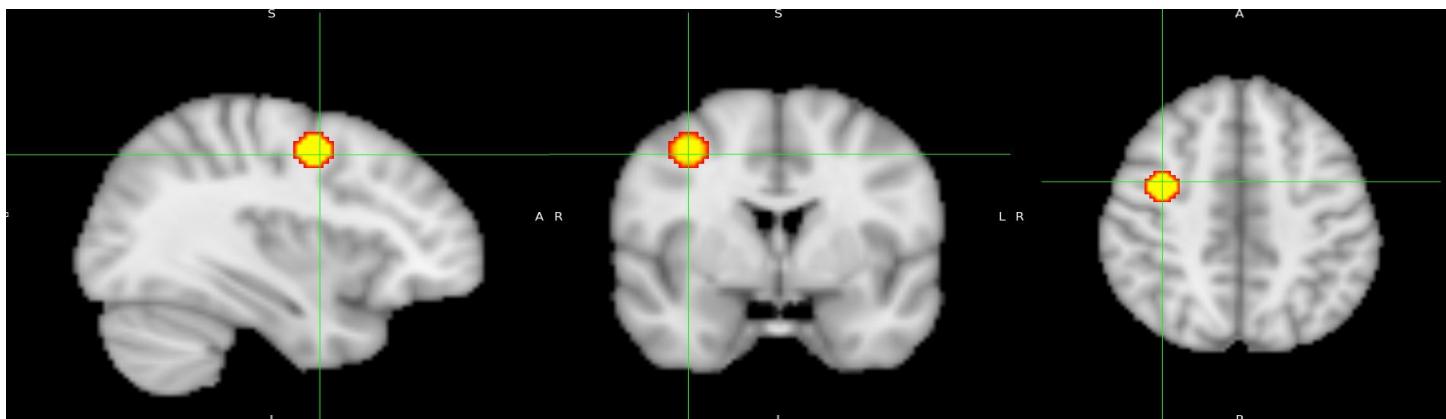
We note that weights numbers are larger than numbers using only binarized mask without

-W.

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.

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

```
mayar@mayar-VirtualBox:~/Data$ fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1 -roi 27 1 62 1 60 1 0 1 ROI_dMPFC_36_2_48.nii.gz -odt float  
mayar@mayar-VirtualBox:~/Data$ fslmaths ROI_dMPFC_36_2_48.nii.gz -kernel sphere 7 -fmean Sphere_dMPFC_36_2_48.nii.gz -odt float  
mayar@mayar-VirtualBox:~/Data$
```



3. Use the Harvard-Oxford subcortical atlas to create an anatomical mask of the right amygdala. Label it whatever you want. Then, extract the z-statistics from cope1 (i.e., the contrast estimates for Incongruent compared to baseline).

