



جامعة القاهرة

fMRI Report

A Study on Cognitive Control

Presented To: Prof. Meena Makary

Written By: Nariman Ahmed Sayed

Table of Contents:

Task 1: Explaining T1, T2, and T2*:	3
Task 2: Flanker Dataset Quality Control	6
Task 3: Preprocessing:	7
Task 4: Neuroanatomy Regions and Functions:	33
Task 5: Report on The Fusiform Gyrus.	48
Task 6: First-Level Analysis:	49
Task 7: Design Script Optimization:	84
Task 8: 2nd Level Analysis:	95
Task 9: 2nd Level Analysis Stats Tab Options:	100
Task 10: 3rd Level Analysis:	102
Task 11: ROI Analysis	110
The Flanker Task: An fMRI Study	129

Task 1: Explaining T1, T2, and T2*:

The terms T1, T2, and T2* refer to different ways protons in your body's tissues respond when placed in a magnetic field and disturbed by a radiofrequency (RF) pulse. The way these protons relax and return to their normal state is what gets detected by the scanner and produces the MRI images we see.

T1 Relaxation (AKA Spin-Lattice Relaxation):

It measures how long it takes for the protons to realign with the magnetic field (B0) after being disturbed by the RF pulse. It's the longest type of relaxation as it usually takes hundreds to thousands of milliseconds and it is affected by factors such as tissue type. For instance, fat molecules have a loose chain of carbon and hydrogen, rotate slowly, and have high energy transfer efficiency, this causes them to have a more rapid T1 relaxation period. On the other hand, water molecules have a lower density of hydrogen nuclei, they tumble freely and rapidly and have low energy transfer efficiency, causing them to have a slower T1 relaxation period.

T1 relaxation is also called Spin-Lattice Relaxation to highlight the key factors involved in the process. "Spin" refers to the protons' magnetic properties, as they behave like tiny spinning magnets, while "lattice" refers to the surrounding molecular environment like water and fat. As the proton molecules relax from the RF pulse, they transfer their extra energy to the surrounding molecules in order to slow down and return to the longitudinal position.

A T1-weighted image is an MRI scan where the contrast is based on T1 relaxation times of different tissues. Tissues with shorter T1 relaxation times –fatty tissues like white matter-- appear bright, while tissues with longer T1 relaxation times –water-based tissues-- appear dark. In fact, bone (without marrow) appears very dark, almost black. T1-weighted images give high anatomical detail but are not used for functional imaging, meaning it doesn't track real-time brain function. This is because they have high spatial resolution, clear differentiation between grey and white matter, and shorter scan times.

T2 Relaxation (AKA Spin-Spin Relaxation):

In a magnetic field, protons initially spin in sync. However, as time passes, they start to get out of step and that is due to the tiny differences in their local magnetic environments. T2 measures how long protons stay in sync with each other before dephasing and losing coordination. T2 is shorter than T1, typically lasting tens to hundreds of milliseconds. Similarly to T1, it's affected by different tissue and molecular interactions. For example, as mentioned before. Water molecules are less hydrogen dense and tumble around rapidly. Their low dissipation of energy makes the dephasing slow and thus cause them to have slow T2.

T2 relaxation is also termed as Spin-Spin Relaxation to highlight the event's factors. Unlike T1 which interacts with the environment, T2 happens due to the interactions between protons themselves. In a magnetic field, protons initially spin in sync. However, as time passes, they start to get out of step and that is due to the tiny differences in their local magnetic environments. This dephasing is caused by the protons interacting with one another, hence the term “spin-spin”.

A T2-weighted image is an MRI scan where the contrast is based on T2 relaxation times of different tissues. Tissues with longer T2 relaxation times appear bright, while tissues with shorter T2 relaxation times appear dark. Contrary to T1, fat may appear dark (low-contrast) while water-based tissue appears bright (high contrast). T2-weighted MRI highlights fluid and tissue differences. They are commonly used in medical diagnosis for detecting brain diseases, injuries, and abnormalities. However, they are not used for functional imaging because they are not sensitive to real-time changes in brain activity.

T2* Relaxation (Effective T2):

T2* is similar to T2, but it also includes magnetic field inhomogeneities, which are the small variations in the scanner's magnetic field and tissue properties. It's even shorter than T2, lasting around tens of milliseconds. T2* is very important as the BOLD signal (Blood Oxygen Level Dependent) contrast used in fMRI is based on it. This is due to the fact that oxygenated blood (oxyhemoglobin) reduces magnetic inhomogeneity, which means less signal loss and higher fMRI signal, while deoxygenated blood (deoxyhemoglobin) increases inhomogeneity, which means faster T2 decay* and lower fMRI signal.

T2* is actually the most important of relaxation times and is the cornerstone of fMRI experiments as fMRI relies on the BOLD contrast, which depends on T2* decay. The different areas with weak and strong signals are what allow researchers to map brain activity by detecting changes in blood oxygenation.

Understanding the Difference Between T2 and T2* Relaxation

While T2 and T2* both describe the decay of transverse magnetization, the mechanisms behind their decay — and their roles in MRI — differ significantly. The primary distinction lies in what contributes to the dephasing of spins. T2 reflects pure spin-spin interactions within tissues, assuming a perfectly uniform magnetic field. In contrast, T2* captures not only these intrinsic interactions but also additional dephasing caused by magnetic field inhomogeneities.

Spin-echo sequences are used to measure T2, as they apply a 180° refocusing pulse to correct for field imperfections. Gradient-echo sequences, however, do not include this refocusing pulse and thus are used to measure T2*, making them more sensitive to susceptibility differences.

Clinically, this difference influences how the two are used. T2-weighted imaging is preferred for general tissue characterization due to its consistency and specificity to pathology like edema or

tumors. T2*-weighted imaging, while less commonly used, is invaluable in detecting microbleeds, iron deposits, or changes in blood oxygenation — especially in applications like functional MRI or susceptibility-weighted imaging. Thus, while related, T2 and T2* offer distinct windows into tissue properties and magnetic environment, each serving unique diagnostic purposes.

Task 2: Flanker Dataset Quality Control

For this task, visual inspection of each run (for each subject) is conducted. Running the volumes quickly allows us to watch them as a video of the brain in motion. Any detection of sharp motion or sudden blurs in the volume tells us there are motion artifacts present in the data. Here are my inferences after scanning the data:

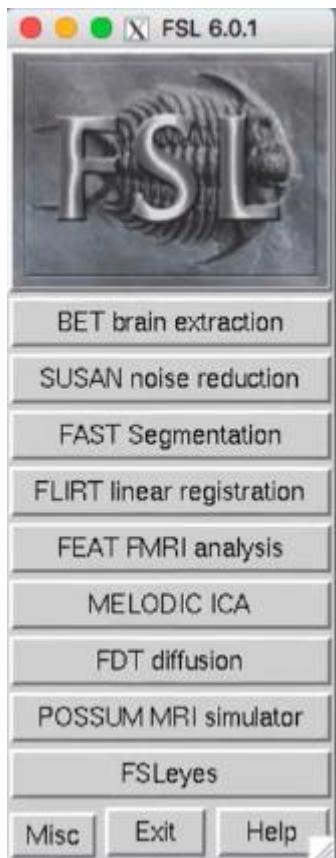
Functional dataset		Motion artifact detected?	Functional dataset		Motion artifact detected?
Sub 1	Run 1	No	Sub 14	Run 1	No
	Run 2	No		Run 2	No
Sub 2	Run 1	No	Sub 15	Run 1	No
	Run 2	No		Run 2	No
Sub 3	Run 1	No	Sub 16	Run 1	Yes
	Run 2	No		Run 2	No
Sub 4	Run 1	No	Sub 17	Run 1	No
	Run 2	No		Run 2	No
Sub 5	Run 1	No	Sub 18	Run 1	No
	Run 2	No		Run 2	No
Sub 6	Run 1	No	Sub 19	Run 1	No
	Run 2	No		Run 2	No
Sub 7	Run 1	No	Sub 20	Run 1	No
	Run 2	No		Run 2	No
Sub 8	Run 1	No	Sub 21	Run 1	No
	Run 2	No		Run 2	No
Sub 9	Run 1	No	Sub 22	Run 1	Yes
	Run 2	No		Run 2	No
Sub 10	Run 1	No	Sub 23	Run 1	No
	Run 2	No		Run 2	No
Sub 11	Run 1	No	Sub 24	Run 1	No
	Run 2	No		Run 2	No
Sub 12	Run 1	No	Sub 25	Run 1	Yes
	Run 2	Yes		Run 2	Yes
Sub 13	Run 1	No	Sub 26	Run 1	No
	Run 2	No		Run 2	No

Task 3: Preprocessing:

In a perfect world, our data would be crystal clear, with every feature neatly laid out and easy to read. But in reality, data extraction isn't as simple as pushing a button. Take an fMRI volume, for instance—it captures the oxygenated blood signal we care about, but it also brings along some unwanted guests like motion artifacts, slow drifts, and physiological noise from breathing and heartbeats. That's where preprocessing steps in: a crucial phase to isolate the meaningful signals and filter out the noise.

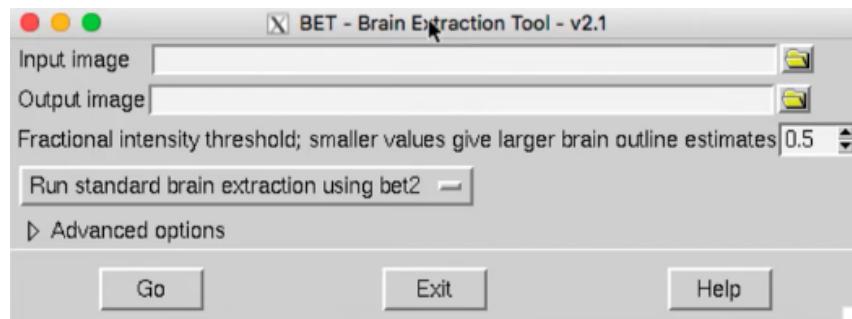
Step 1- Brain Extraction (or “Skull Stripping”)

Our first main step is brain extraction. Since our signal lies within the brain itself, there's no need to keep the skull around. Skull stripping makes it easier to focus on the brain alone as the main volume. Moreover, removing the skull improves registration and normalization steps later. In FSL, this could be done easily by a tool called bet, or the Brain Extraction Tool. We'll navigate to our sub-08 directory in our data set, and enter command `fsl` to start.

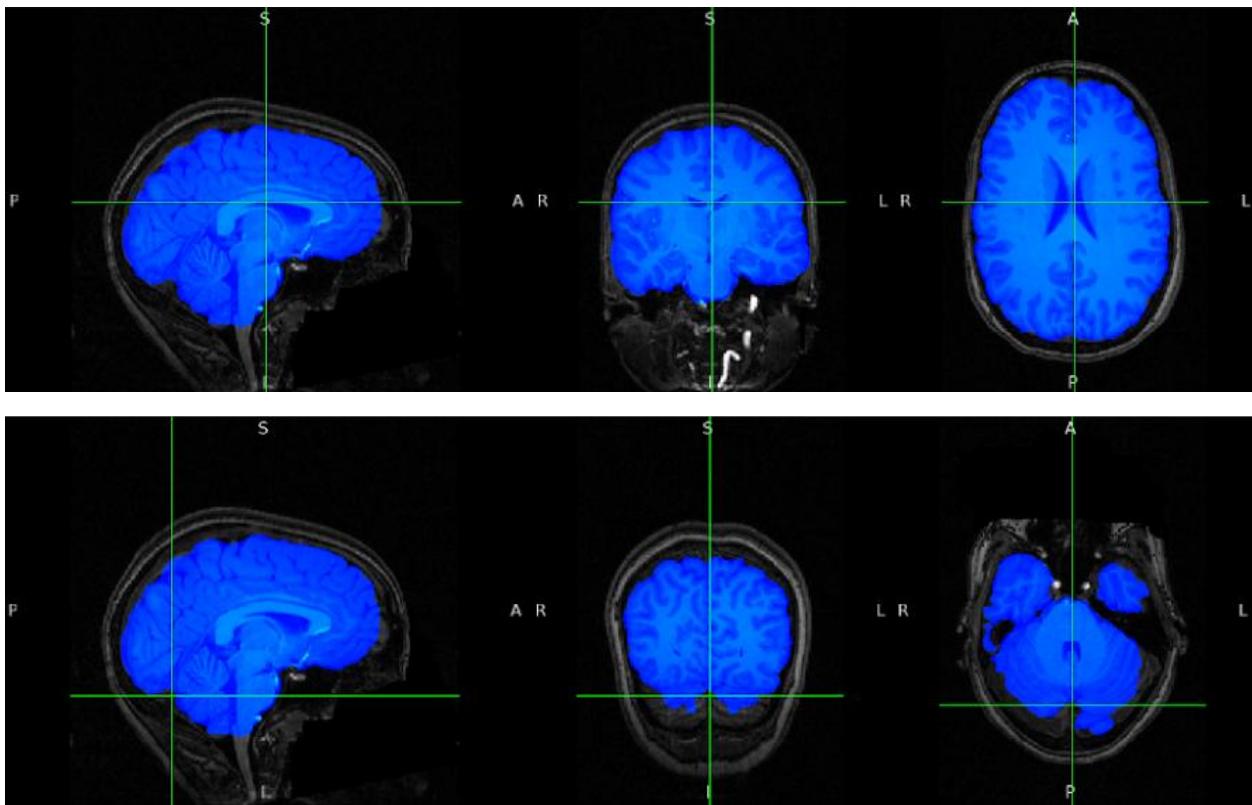


This opens up the `fsl` GUI window. We'll then click on bet. One of this tool's parameters is called fractional intensity threshold, where smaller threshold values give larger brain outline while larger values give smaller brain outline estimates. This threshold can be altered and tested until a reasonable volume is obtained. Of course, if removing more skull risks losing brain volume data, it's better to just leave some of it in place.

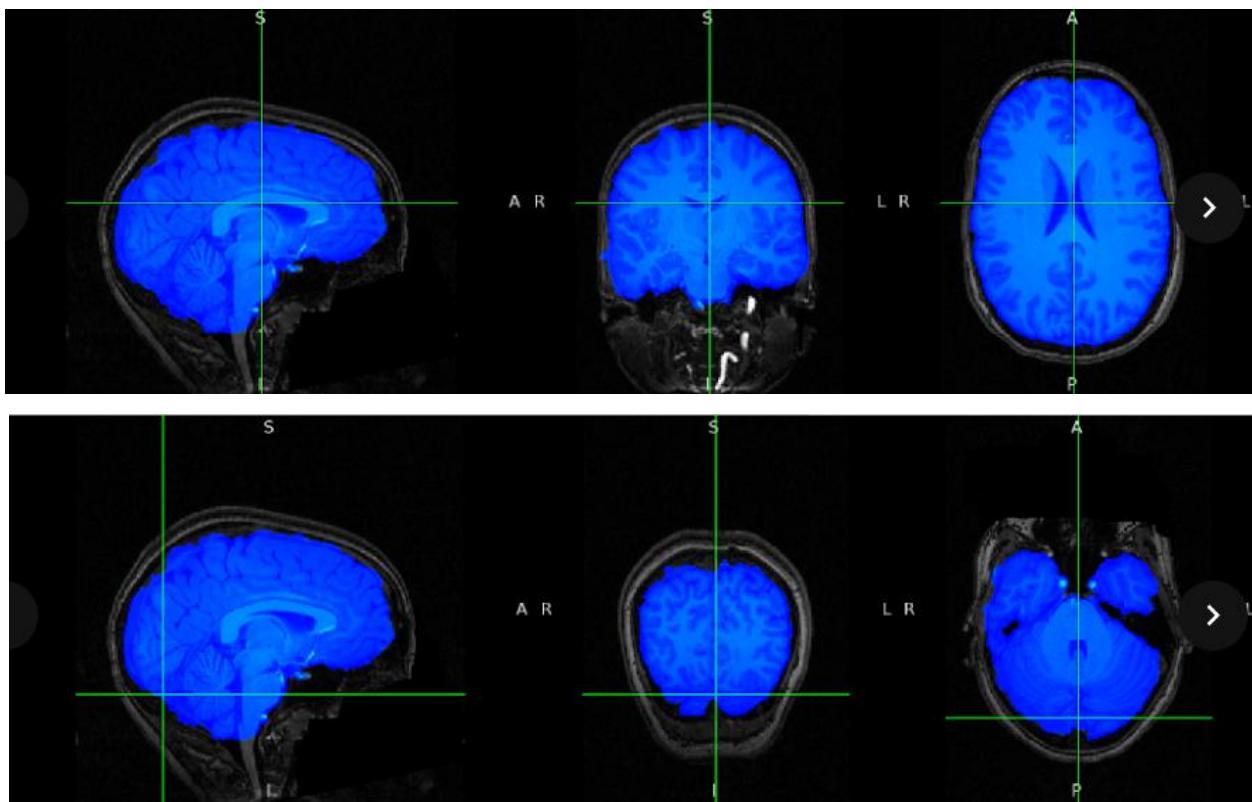
Let's start by trying out a threshold of 0.5



- **0.5 threshold:** It's a good start, yet we can notice that at far ends, like at the cerebellum, some portions of the brain are missing. We can try to make it better by decreasing the threshold. Brain is in blue.

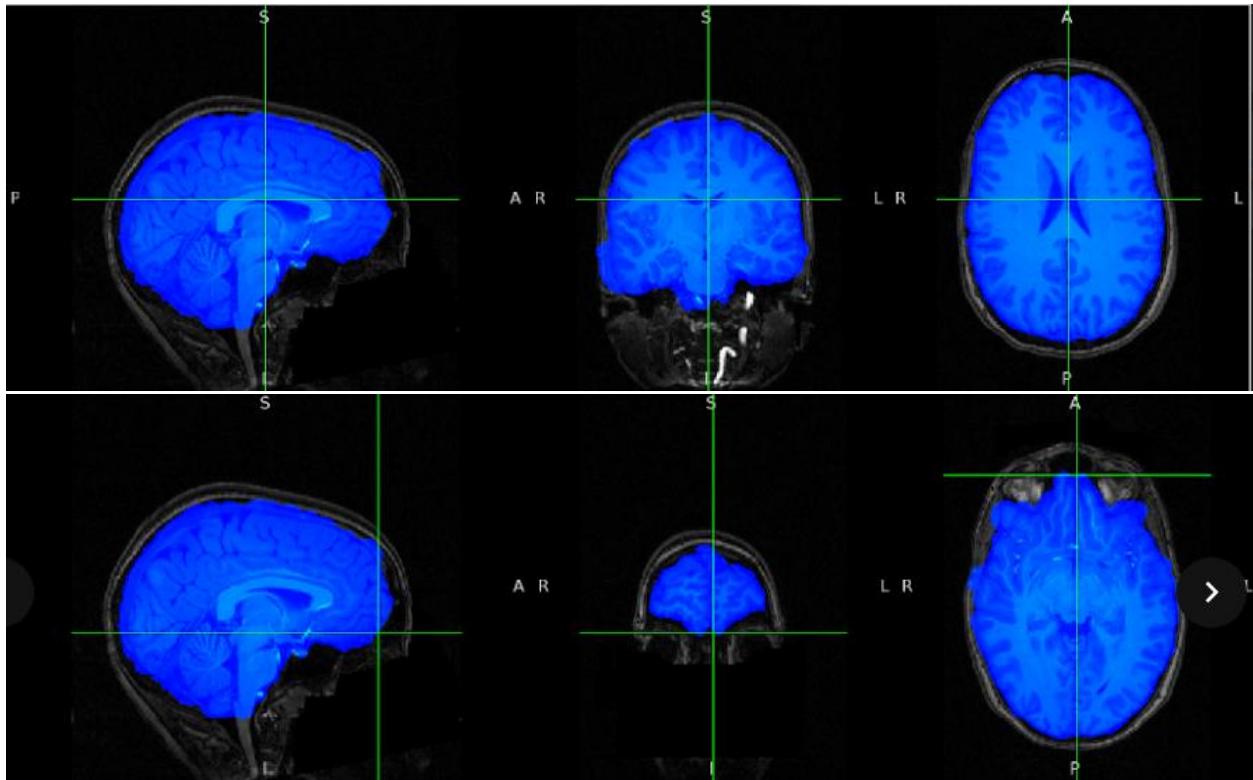


- **0.2 threshold:** Indeed, my brain region (in blue) is now a bit larger, which means we have more brain volume included. There are also very small skull areas included, nothing worth mentioning.

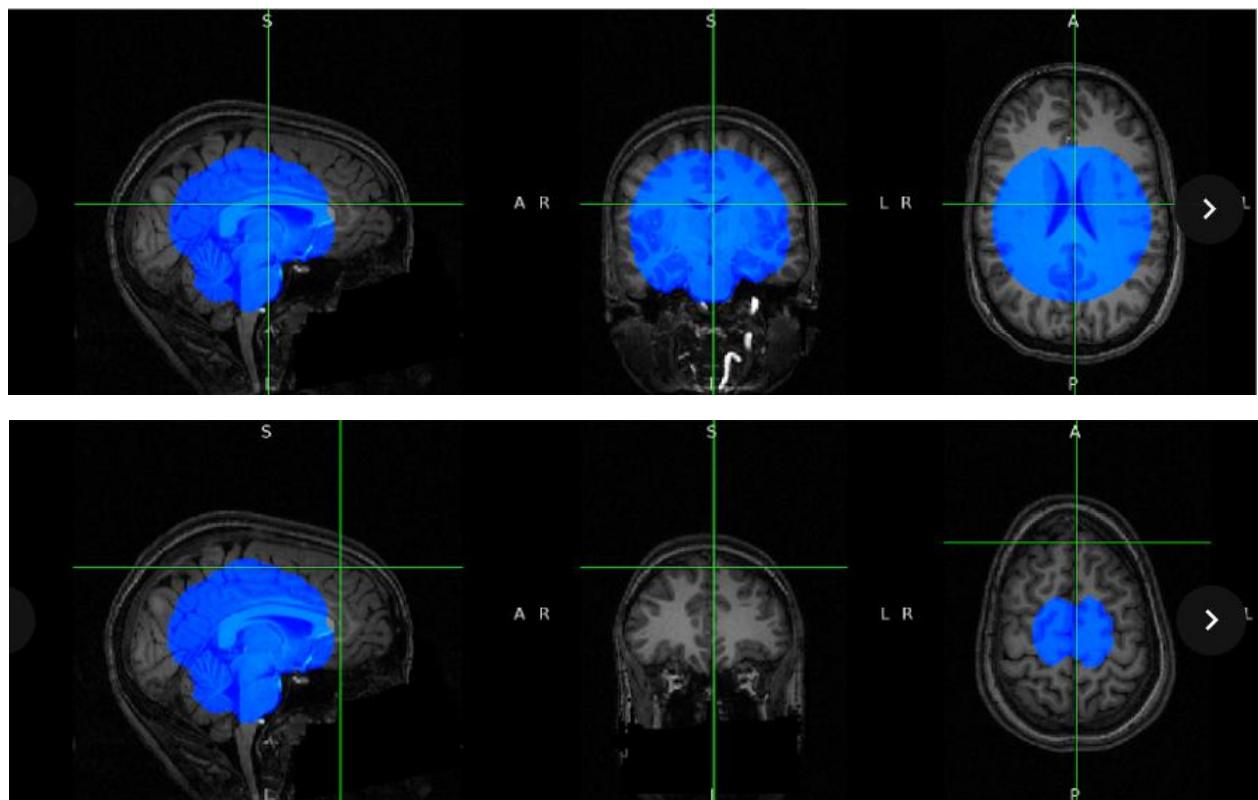


Exercise: try thresholds 0.1 and 0.9,

- **Threshold 0.1:** we can see that it's almost missing no brain regions, yet more skull is visibly included now.



- **Threshold 0.9:** Here, so much of the brain is missing as the algorithm put it at the cost of completely removing even any part of the brain close to the skull.



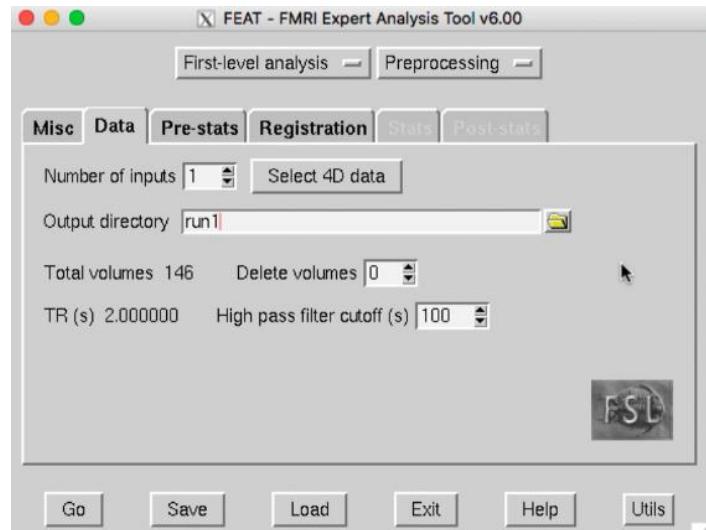
Steps 2,3,4,5 (motion correction, smoothing, time slicing correction, registration and normalization) are all done using the FEAT tool.

FEAT (standing for fMRI Expert Analysis Tool.) is an fMRI analysis tool within FSL that automates many analysis decisions, making it easy to run simple experiments while still offering the flexibility for complex ones. The data modelling which FEAT uses is based on general linear modelling (GLM), otherwise known as multiple regression. It allows you to describe the experimental design; then a model is created that should fit the data, telling you where the brain has activated in response to the stimuli.

From the fsl GUI again, we'll chose FEAT fMRI analysis.

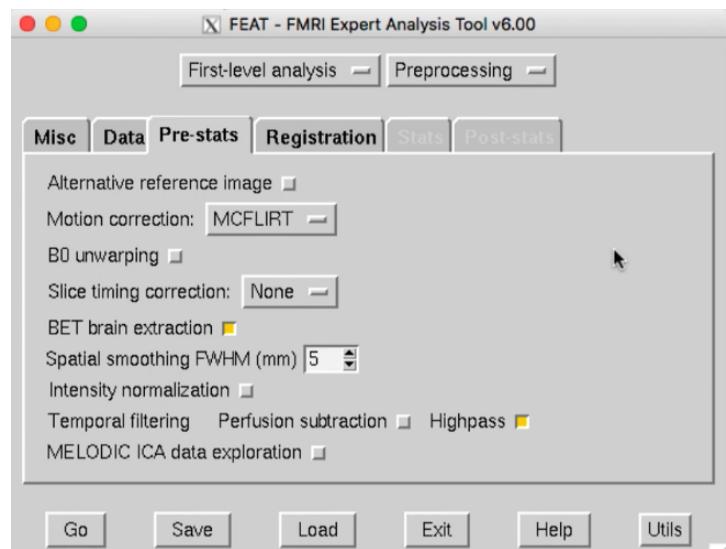
▪ Data

The first tab in the FEAT window is for loading our data. We'll add the first run of our sub-08 functional dataset, and make sure to choose out analysis on top of the window to be First-level analysis, preprocessing. Total volume and TR will be extracted from the header information of the image.



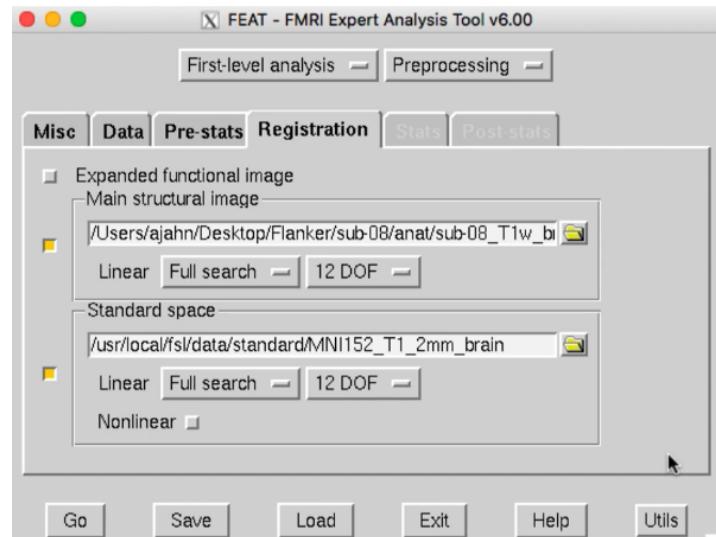
▪ Pre-stats

In this tab we can control motion correction, slice timing correction and spatial smoothing. The BET brain extraction button here does for the functional images what we did for the anatomical. Motion correction will undo any movements done by the subjects during the scan. For slice timing correction, fsl's default is to not do it, and instead account for the slice timing differences using temporal derivatives which will be discussed later. Smoothing Averages over nearby voxels can significantly help cancel out the noise and enhance the signal, even though it could decrease the functional data's spatial resolution. Not only that, but it's also a step to the goal of normalizing the brain. Two different smoothing kernels (4mm and 10mm) can be applied to an fMRI scan, where the image gets blurrier with larger kernels. For our test run, we'll leave the kernel size just as it's set by fsl, to 5mm.



▪ Registration

Our final tab will be the registration tab. Each subject has a different brain size and shape. If we want to do a group analysis, we need to ensure that each voxel for each subject corresponds to the same part of the brain. Volume registration and normalization is what ensures that every subject's visual cortex is in alignment with each other. This is done by normalizing or warping to a template (most common one being MNI152 brain) and using affine transformations, where the dimensions and coordinates of the template brain are referred to as standardized space.



As we can see the template is loaded by default in the standard space field. This is generally the best option for many scans, however it could be changes if working on special subjects such as children. The main structural image box is for our registration, while the standard space box is for our normalization. We will set them both as full search using 12 DOF.

This makes us finally done! Pressing go will take us to an html page where we can track the process of the FEAT analysis.

Interpreting the output:

- Registration and normalization results:

After conducting FSL FEAT, we are left with multiple output files including `example_func`, `func2highres`, and `func2standard` (and their derivatives like `func_highres`, `func_standardhighres`). These files refer to images at different registration stages in the preprocessing pipeline.

This is what each stage shows:

1) `example_func.nii.gz`

- This is a single representative functional volume (typically the middle timepoint of the functional scan).
- It serves as a reference volume for alignment (motion correction, registration).
- It is in the *functional (standard) space*.
- Resolution and orientation match the original fMRI scan.

2) `func2highres.nii.gz → registration file`

- This is the transformation matrix or warp that maps from functional space to *anatomical (T1) space*.
- Used to transform functional data to the subject's structural brain image.
- Output images like `func_highres.nii.gz` are often `example_func` aligned to `highres` (T1) space.

3) func_highres.nii.gz

- This is the example_func image transformed into the subject's *structural (T1-weighted) space*.
- Same image as example_func, but spatially aligned to the anatomical brain.
- Useful for visual inspection of functional-anatomical alignment

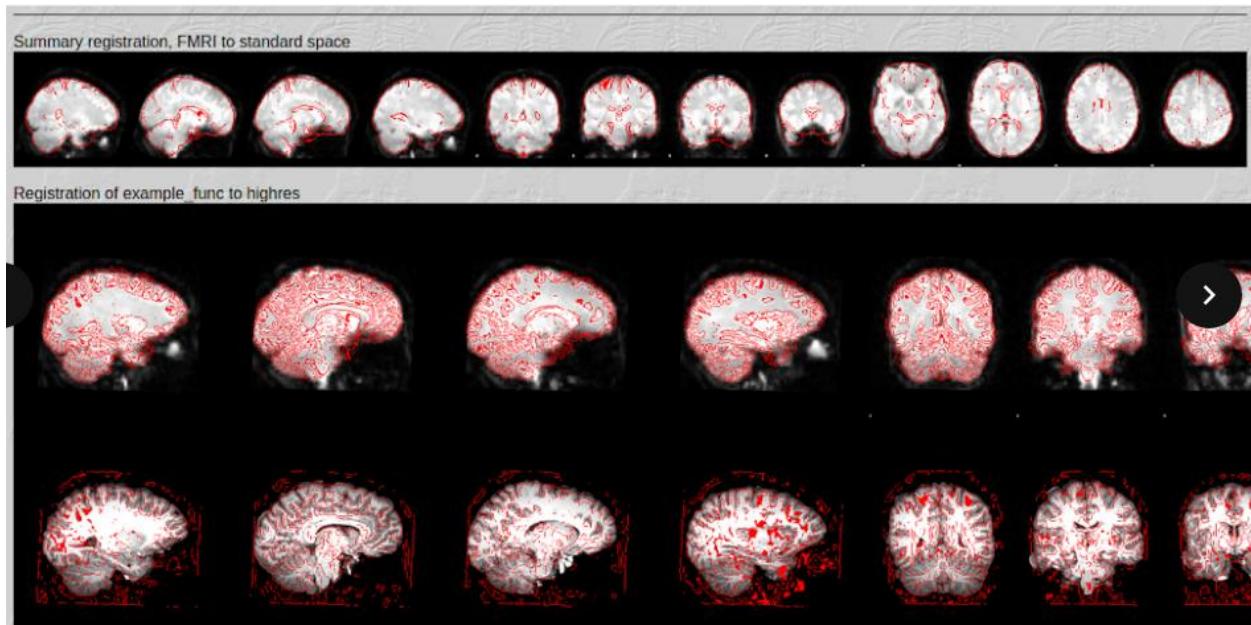
4) func2standard.nii.gz → registration file

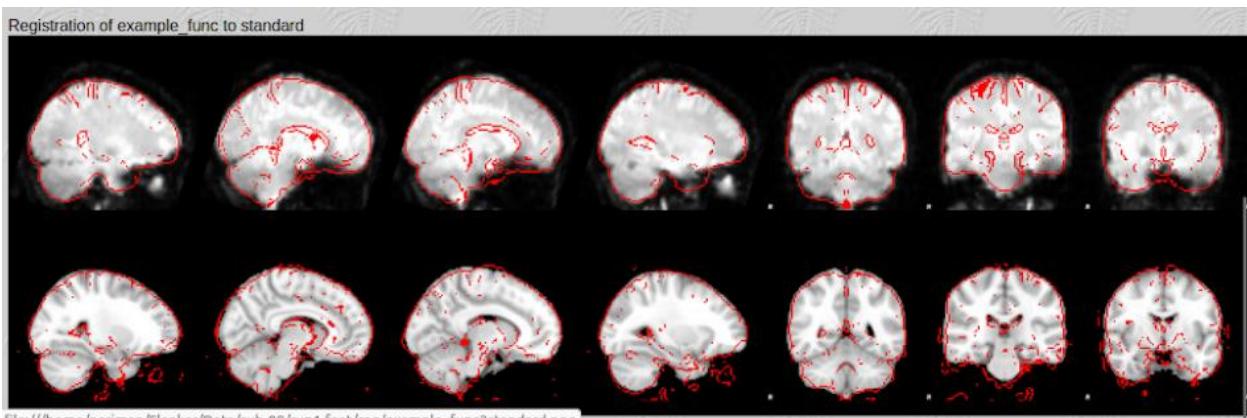
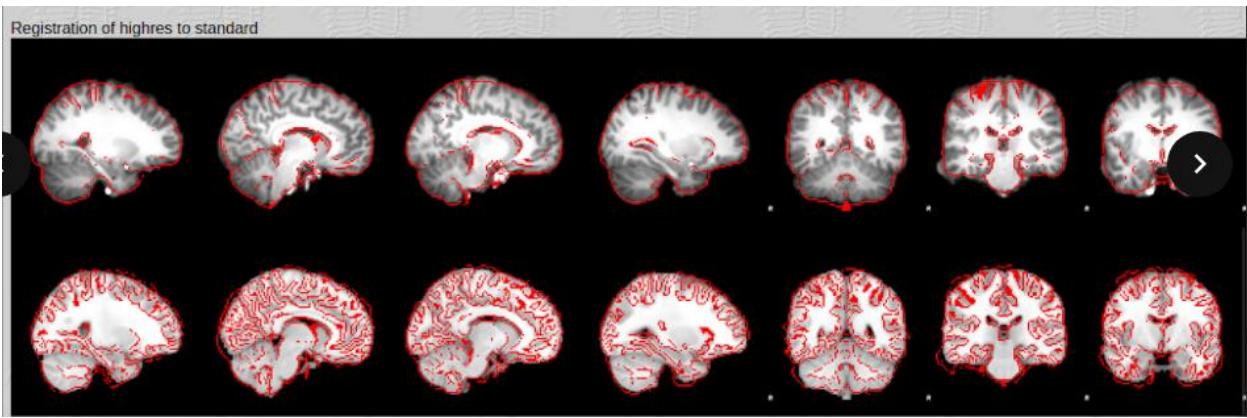
- This maps the functional data all the way to *MNI152 standard space*.
- Usually a combination of func2highres + highres2standard.

5) func_standard.nii.gz or func_standardhighres.nii.gz

- This is the example_func image registered to *MNI152 standard space*.
- Allows for group-level comparisons across subjects.
- Resolution matches the standard brain (e.g., 2mm or 1mm MNI152).

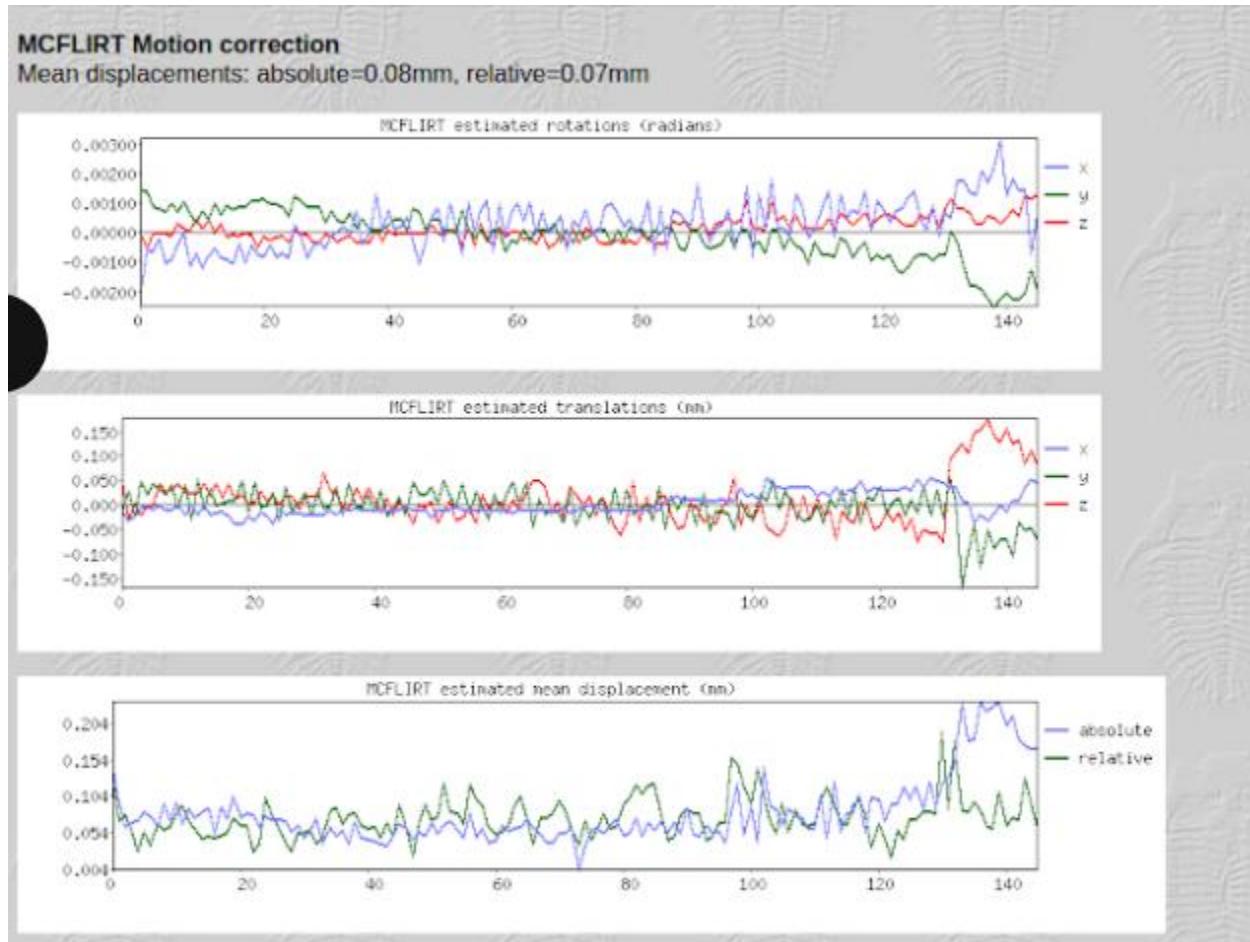
In the 4 different registration results shown in the attached image, the red outline is the template image while the greyscale image is the brain image (in the functional space in the summary, anatomical space in the func_to_highres, and MNI152 standard space in both the high_res_to_standard and the func_to_standard). We want to ensure that the outlines of both images are aligned as well as the internal structures. Obvious errors in the summary registration images will imply normalization and registration process failed. For our processed sub-08 dataset here, we can see that they are almost perfectly aligned.





- Motion correction data (Pre-stats)

These figures show us how much the subject has moved from volume to volume. First graph shows rotations, second graph shows rotations, and the third shows the combinations of both (general displacement). If we observe motion more than half a voxel size from volume to volume or 1 voxel of absolute motion, this will mean too much motion in the dataset, and could be only corrected by more advanced methods such as scrubbing.

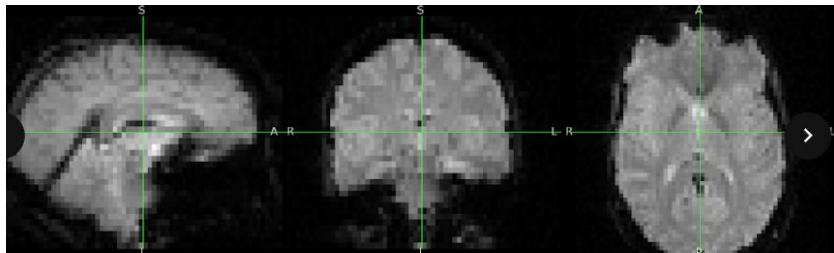


For our dataset, the pre-stats look good, and the motion data shows very little uniform movement of the subject throughout the test, increasing a bit in total displacement (translation and rotation) towards the end of the test. This makes it enough to be corrected using MCFLIRT.

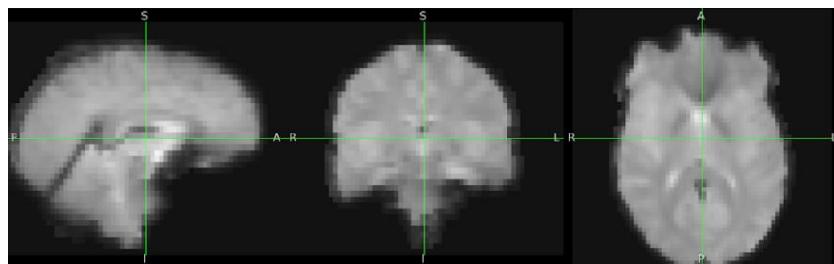
Now let's test the images alignment more closely on fsleyes!

We'll conduct a quality check to see how the processed functional data looks like. On fsl, we will upload the files *example_func.nii.gz* and *filtered_func_data.nii.gz* from our run1 folder. *example_func.nii.gz* is a representative volume of the time series. *filtered_func_data.nii.gz* is the functional data after the preprocessing has been applied to it. The most obvious difference is that it's blurrier due to the smoothing step. Overlaying the two files, we'll check to see if they are aligned correctly or not.

- The functional images:



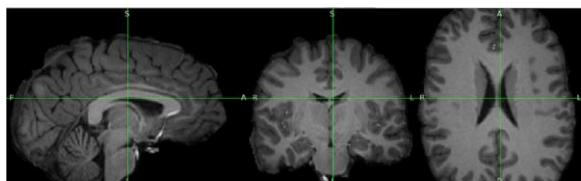
First image is the functional image.



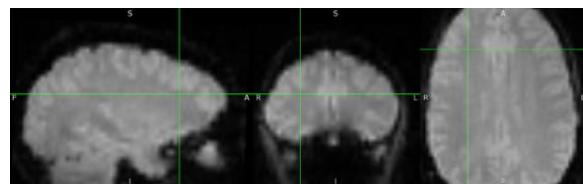
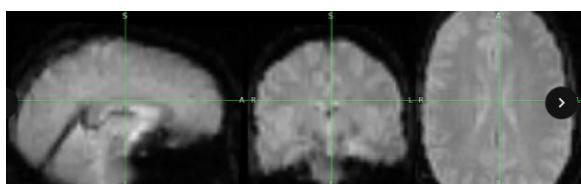
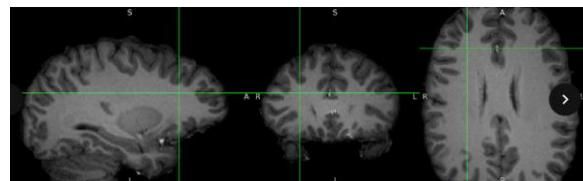
The second has the functional image with filtered functional image overlaid on top.

The images seem to be aligned very well.

- Quality checking the high-resolution images:



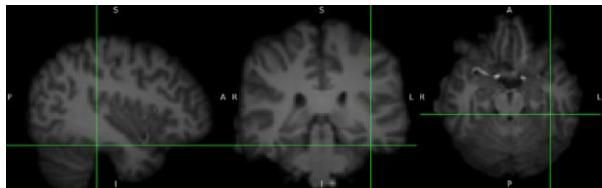
highres.nii.gz



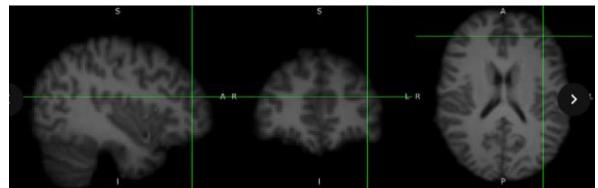
Overlay *example_func_highres.nii.gz*

After checking multiple angles, we can also tell the alignment is good.

- Quality checking the standard high-resolution images:



highres.nii.gz



Overlay *example_fun_standard.nii.gz*

Once again, almost perfectly aligned!

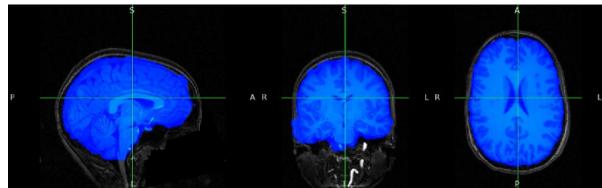
In conclusion, analyzing the output processed image from the generated html file and visually on the fsleyes GUI allows us to check if the preprocessing stage was successful or not. For our data, sub-08, it is successful.

Exercises

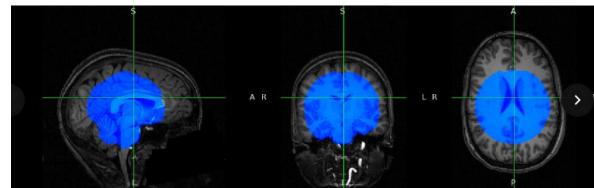
Now we want to try to change some of the preprocessing parameters and see if we observe any differences in the output images.

1. Run BET on the anatomical image sub-08_T1w.nii.gz with two separate fractional intensity thresholds: 0.1 and 0.9. Take a snapshot of each output image with FSLeyes using the camera button (it is located in the upper middle part of the viewer). Note the differences between the two. Is the output what you expected? If you had to use one image or the other, which one would you choose?

- Documented in the brain extraction step. Before seeing the results, I expected a threshold of 0.1 to output an image that has more skull included, while a threshold of 0.9 would cut off more of the brain to ensure complete removal of skull. This in fact did get shown in the image results.



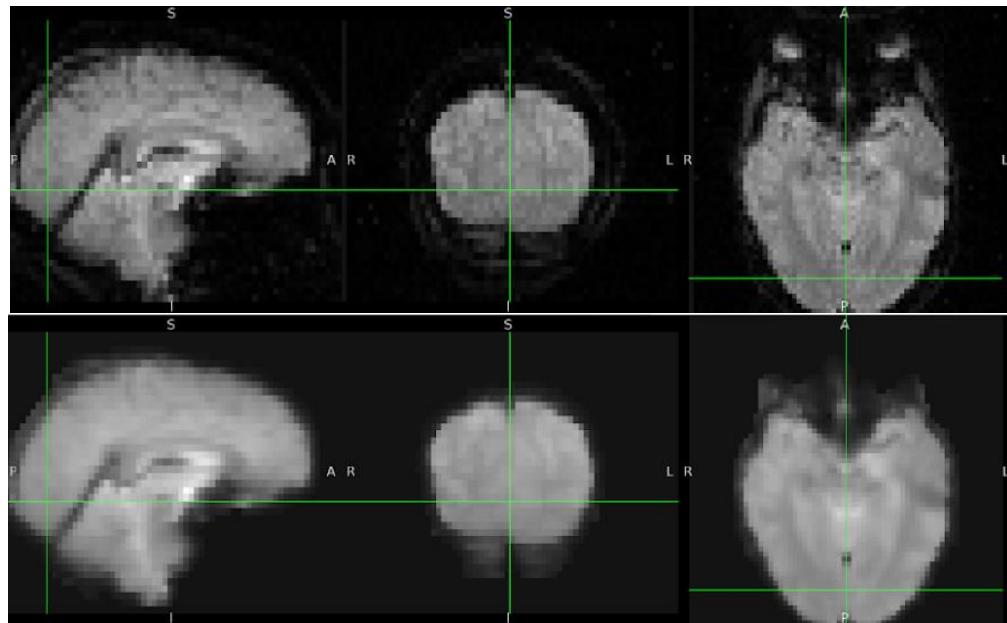
0.1 threshold



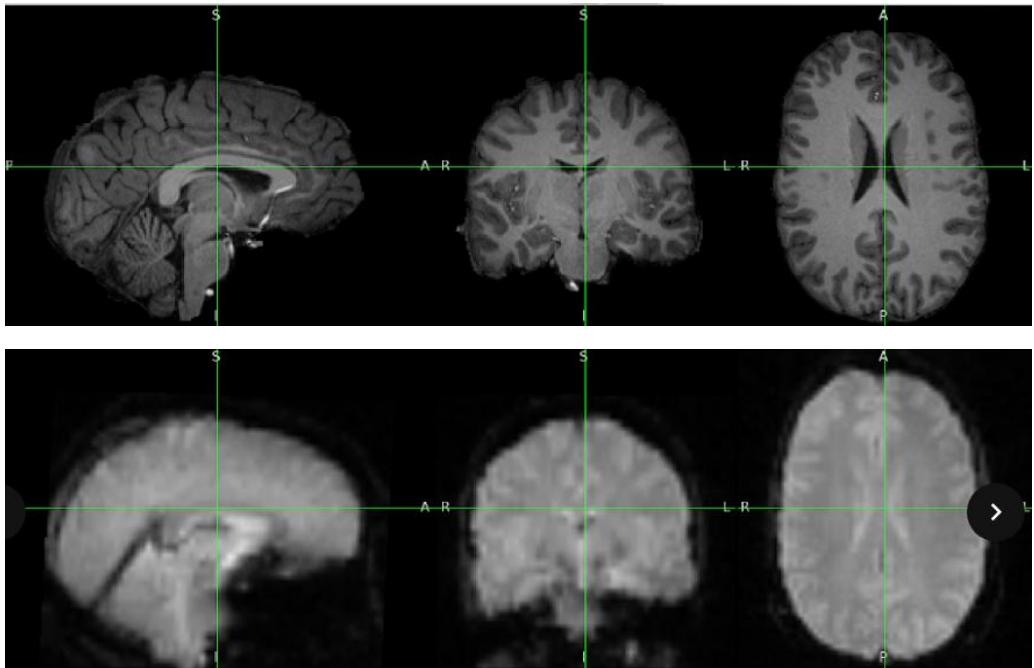
0.9 threshold

2. Preprocess run 2 of the functional data using the FEAT GUI. To do this, select sub-08_task-flanker_run2.nii.gz from the func directory, change the output directory to run2, and make sure Preprocessing is selected from the dropdown menu. Keep the other settings the same as when you analyzed run 1. Do the same quality checks that you did for run 1.

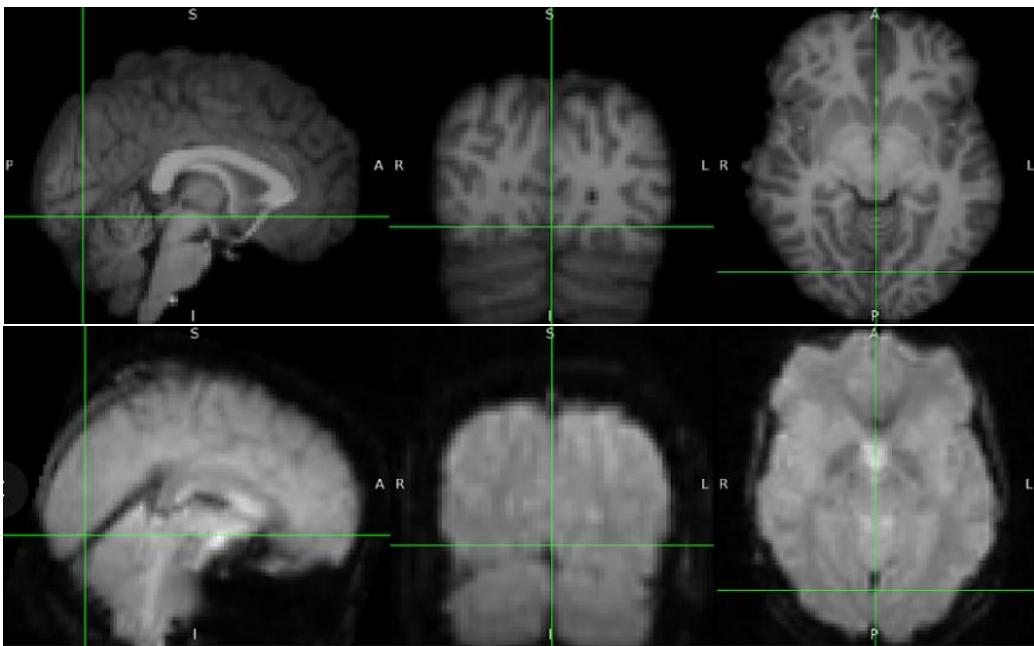
- Quality check on functional data:



- Quality check on high resolution data:

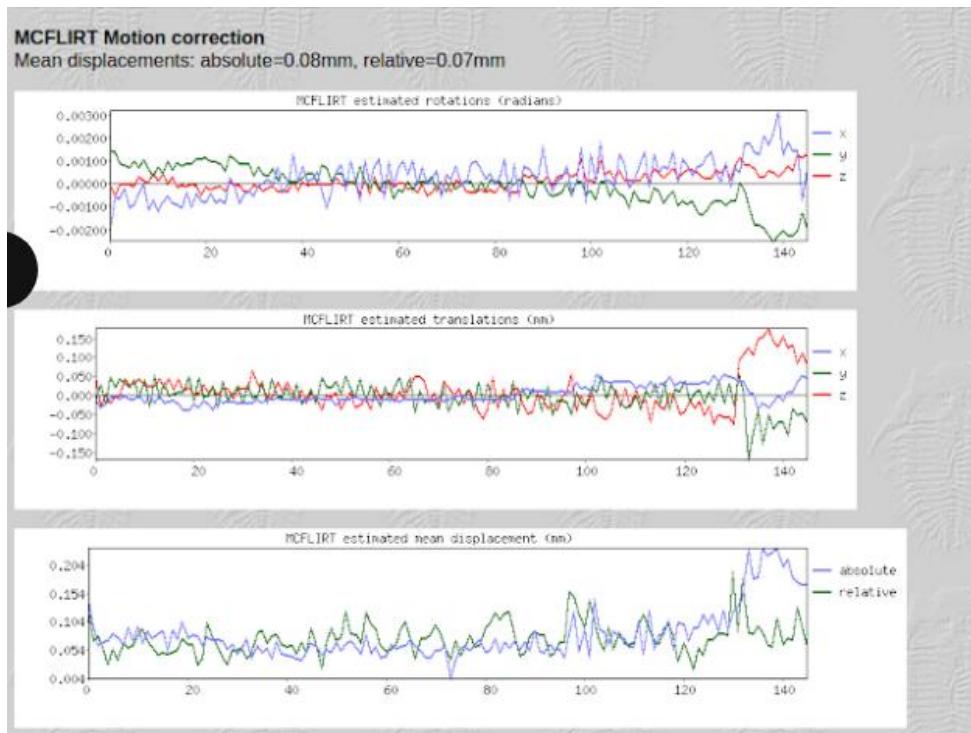


- Quality check on high resolution standard data:

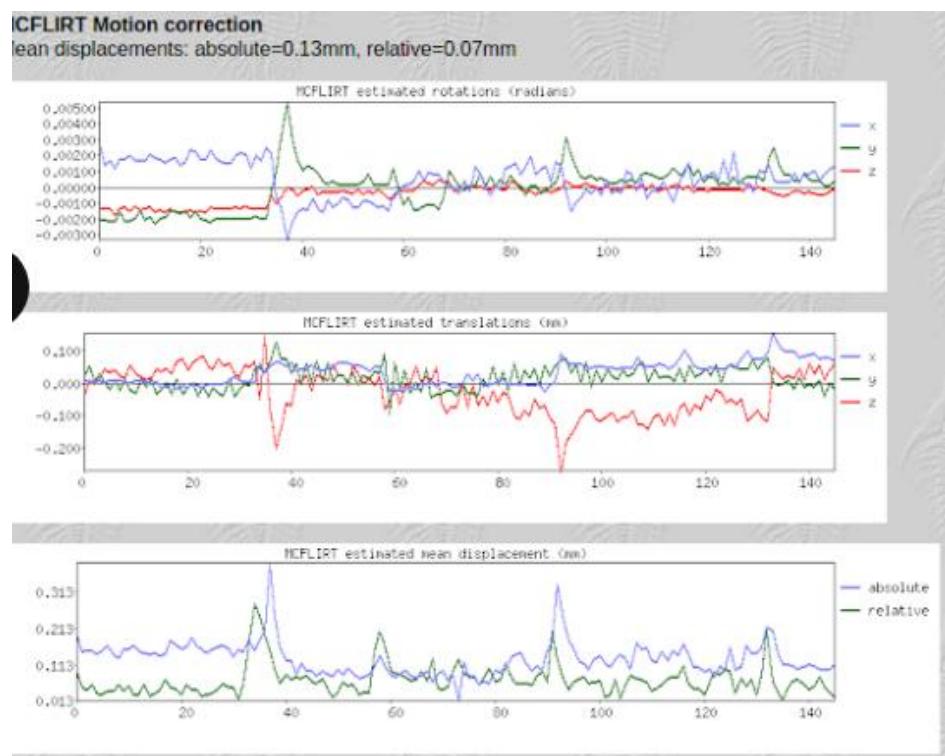


Run1 and run2 data seem to be similar if we analyze them visually. However, if we take a look at the pre stats from the generated html file, we'll find that there are different motion artifacts in both runs.

Pre-stats from the first run:



Pre-stats from the second run:

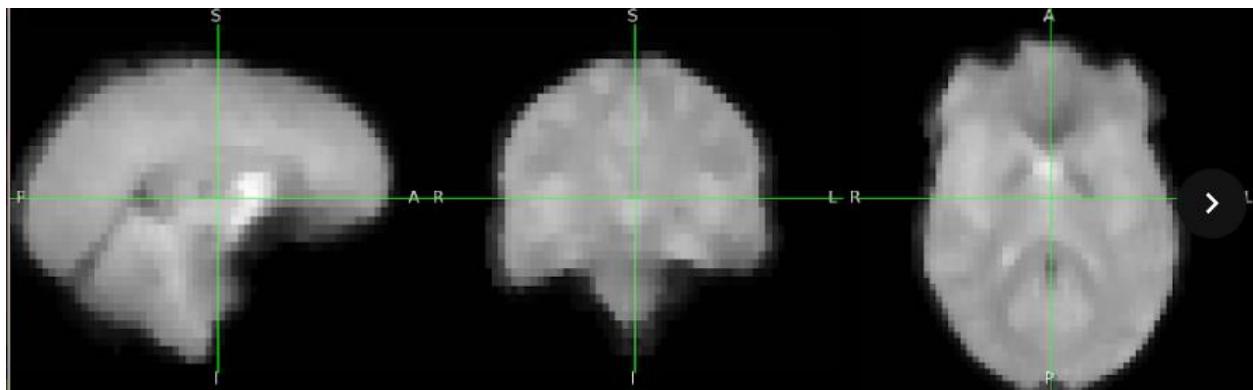


We can interpret from these graphs that in the first run, the subject moved more in the last few seconds of the run, while in the second run the subject moved through out almost all the volumes, and had sudden movements in the first and second third of the run.

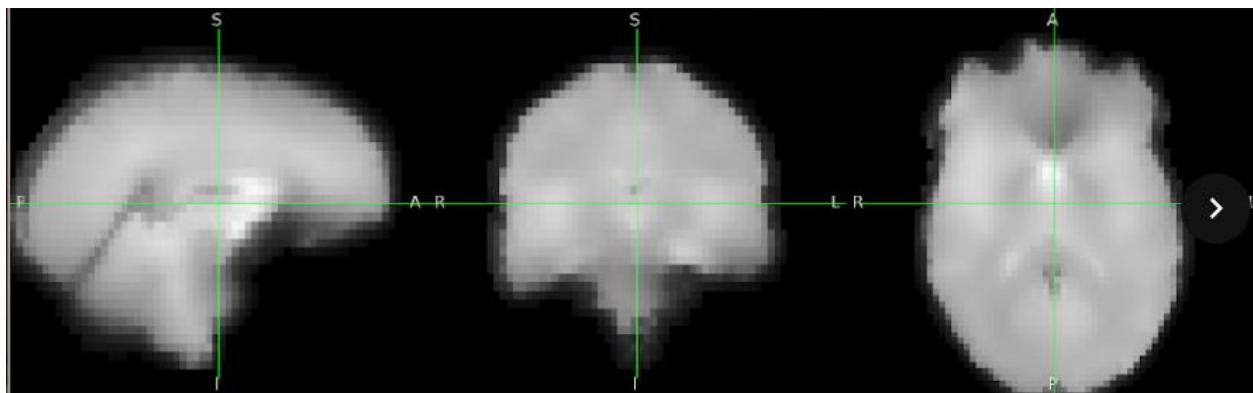
3. Preprocess run 1 using a 3mm smoothing kernel, keeping the other preprocessing options the same. (Make sure, however, to change the output directory to a new name in order to keep the output organized.) Before you look at the output, run another analysis with a 12mm smoothing kernel. Think about what 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?

Before processing the data, I predict that an image filtered by a 12mm kernel will appear blurrier than one that's filtered by a 3mm kernel.

Filtered image using 3mm

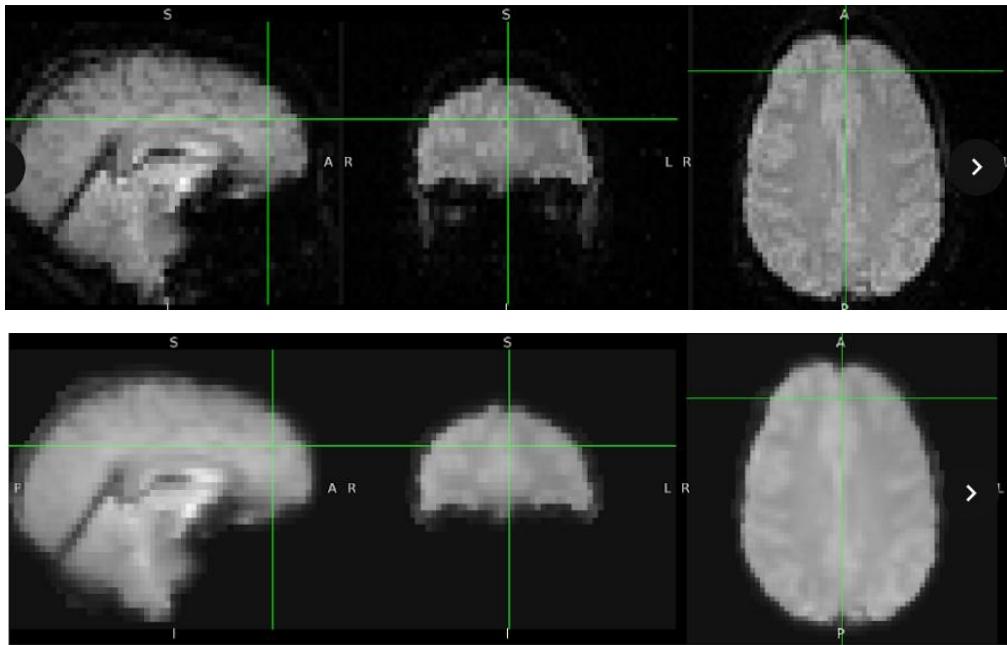


Filtered image using 12mm



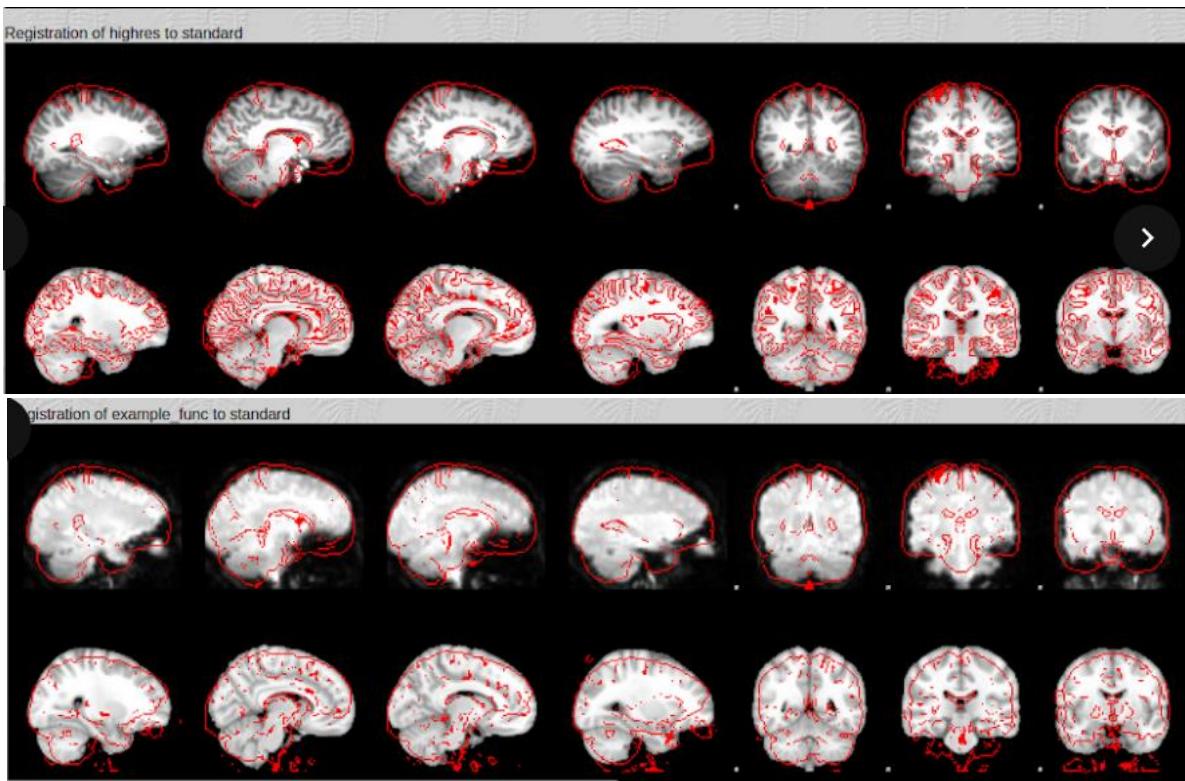
As we can observe, the image filtered using the 12mm kernel appears very smooth and even the color is almost unified. It seems to have lost a lot of the detail which happens when an image is highly smoothed. On the other hand, the filtered image using 3mm seems to retain more details, and it doesn't look as smooth and shiny. This is because the smoothing effect was less.

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?

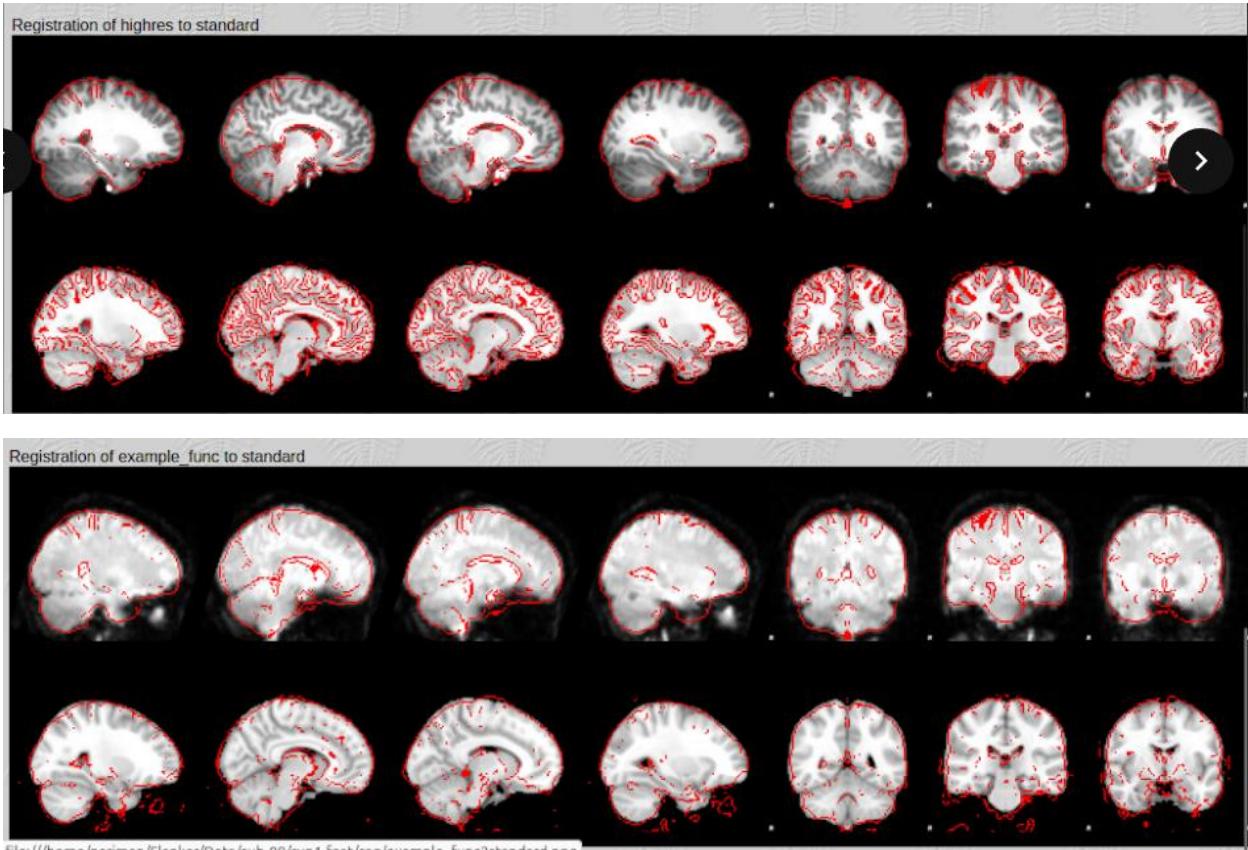


We can see the difference better from the data in the html fil:

Using 3 DOF

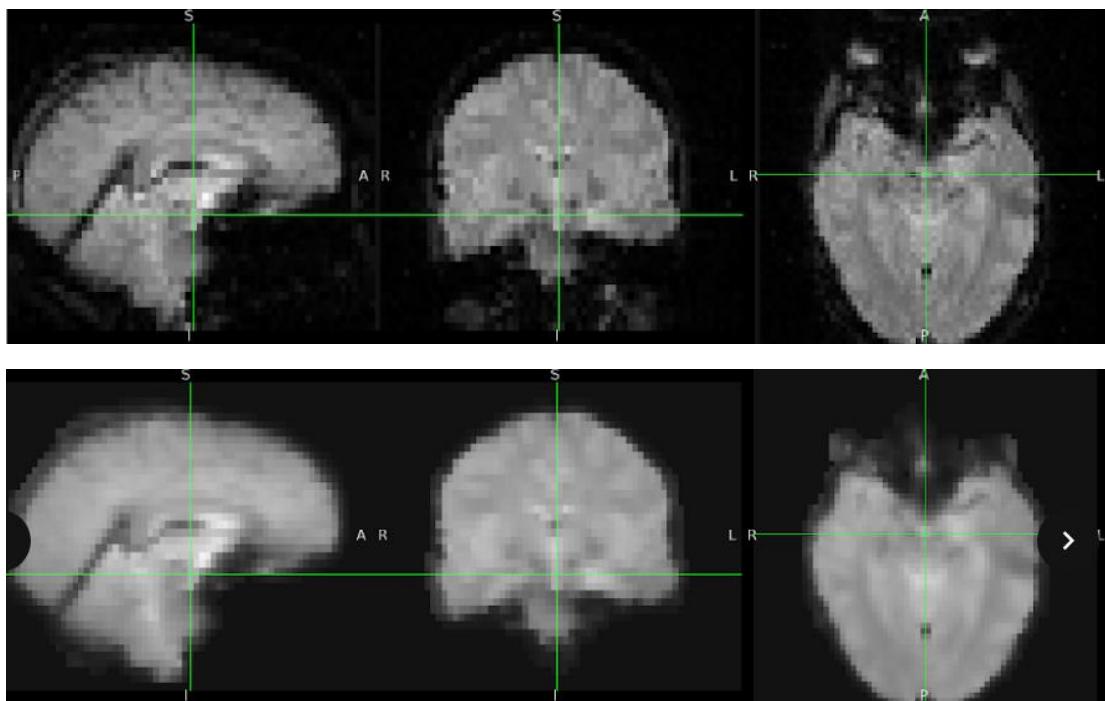


Using 12 DOF

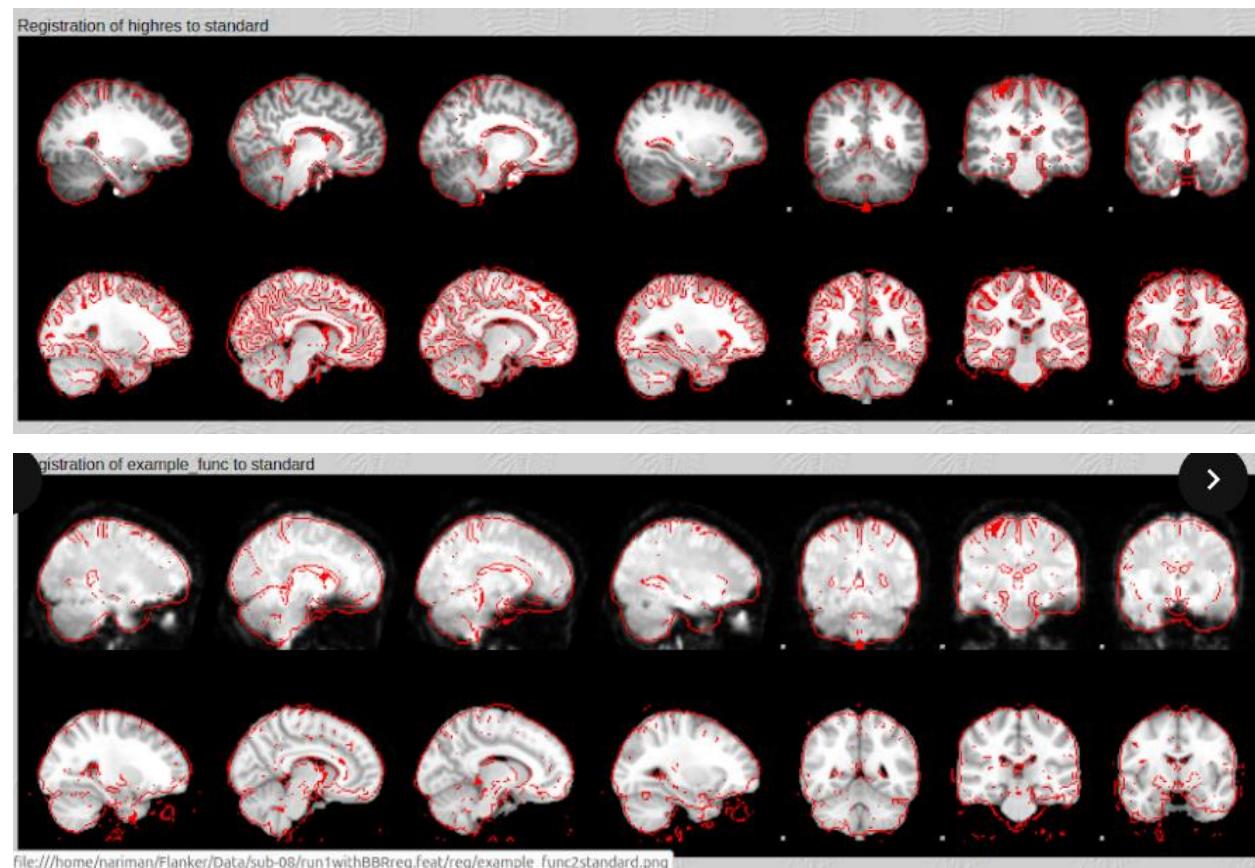


Overall, we can observe that a higher value for DOF makes the red outline align more to the gray brain image than a DOF with smaller value. It's as if increasing the DOF increases the force attracting the outline to the brain, thus leading to better results for normalization and registration!

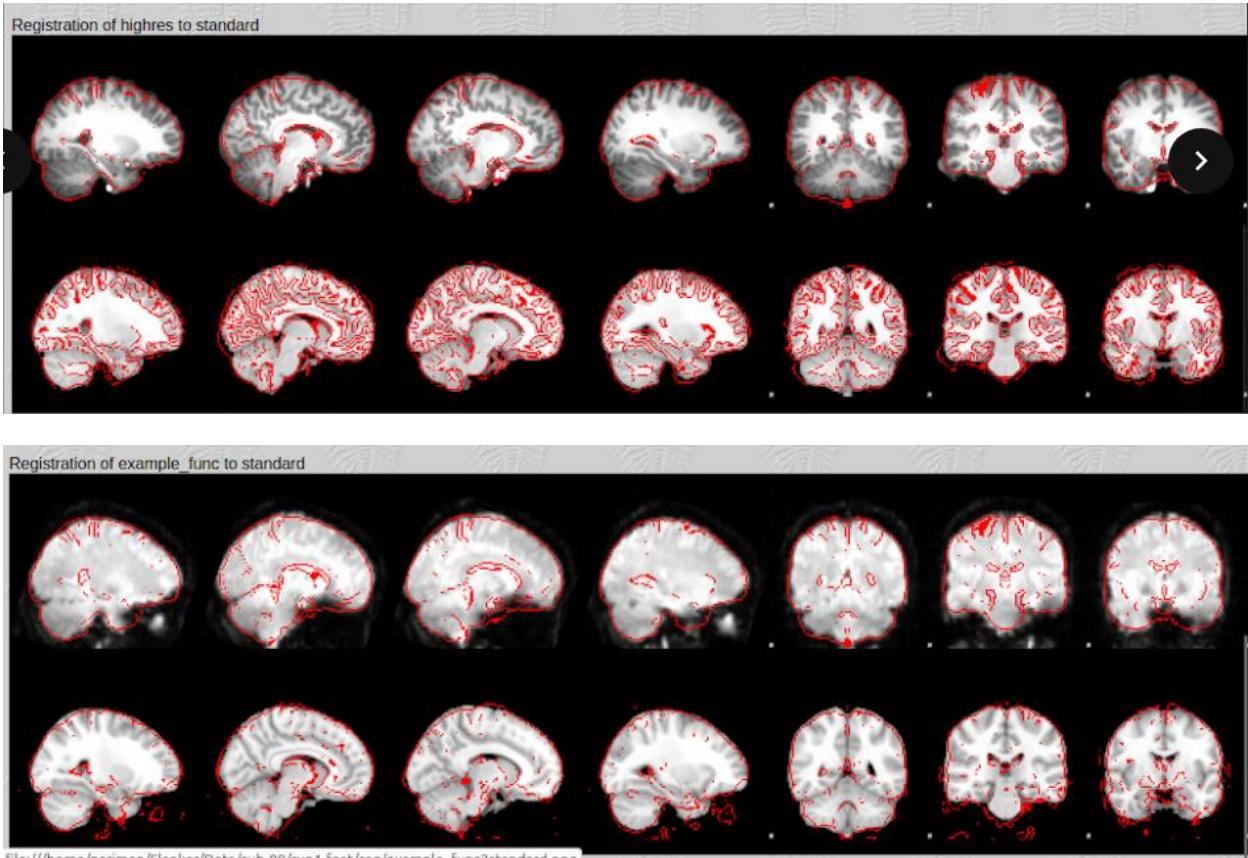
5. Rerun registration for run 1 using BBR instead of 12DOF. What difference does it make? How would you make a case to someone that you should use one instead of the other?



BBR



12 DOF



Observation

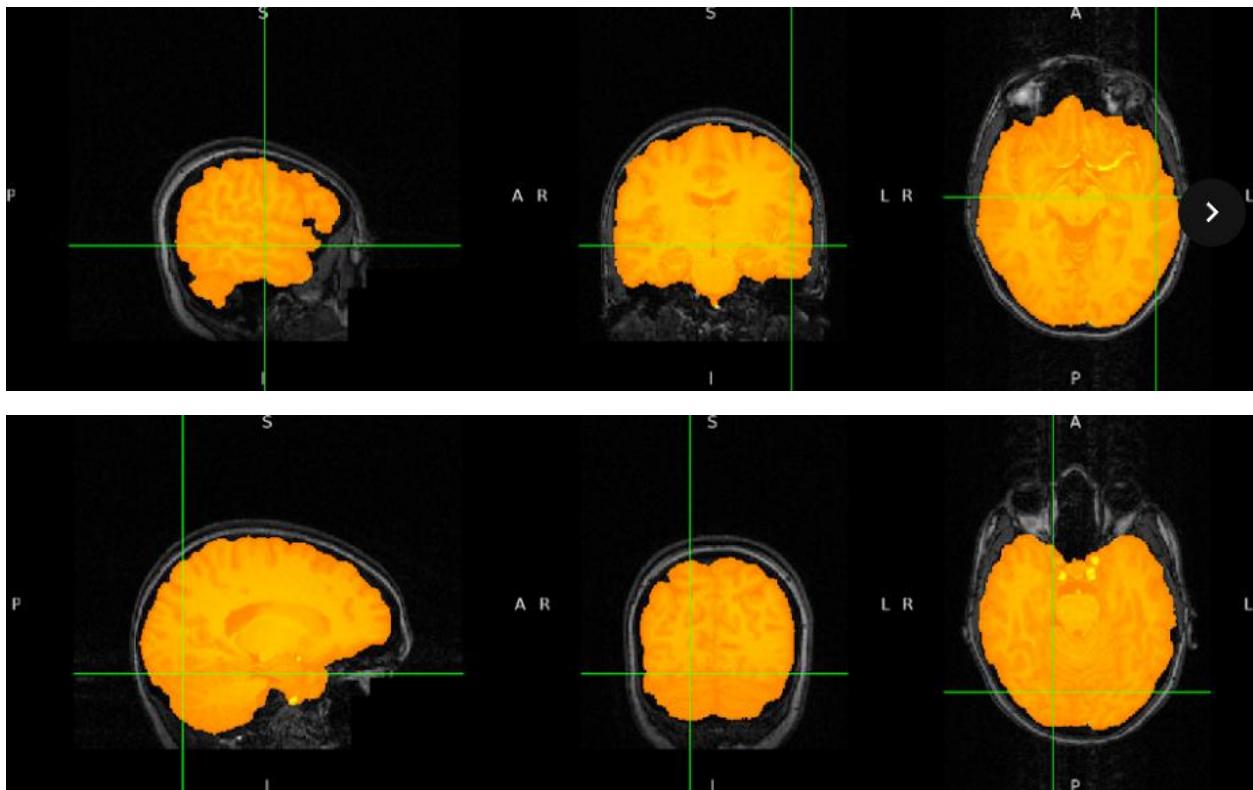
BBR is boundary-based registration. This means it uses the gray-white matter boundaries in the high-resolution structural image to more precisely align the lower-resolution functional image. Even though there's no significant difference that can be detected visually, BBR is more accurate and it yields tighter alignment around the cortex, especially in distorted areas of the functional images. This makes it slower than using 12 DOF, which aligns images using affine transformations.

Additional: Preprocessing steps on Sub-22:

1) Brain Extraction:

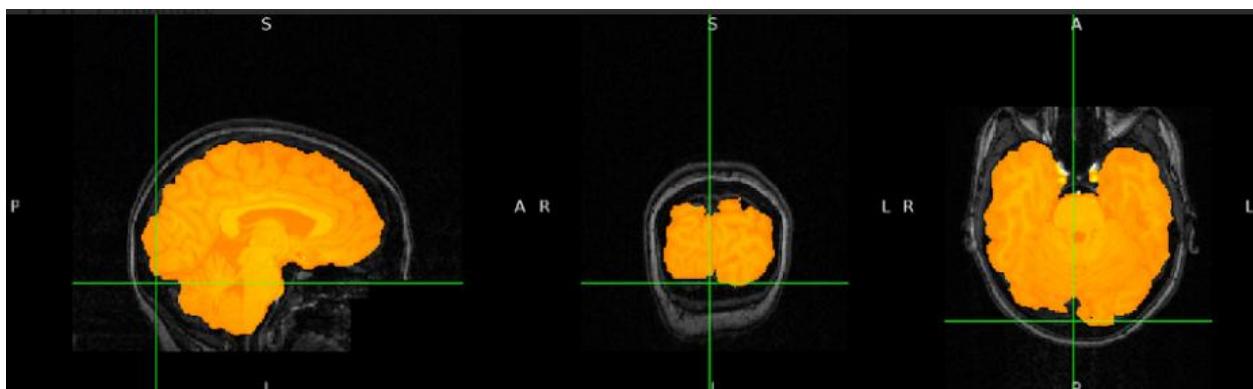
- Using a threshold of 0.2:

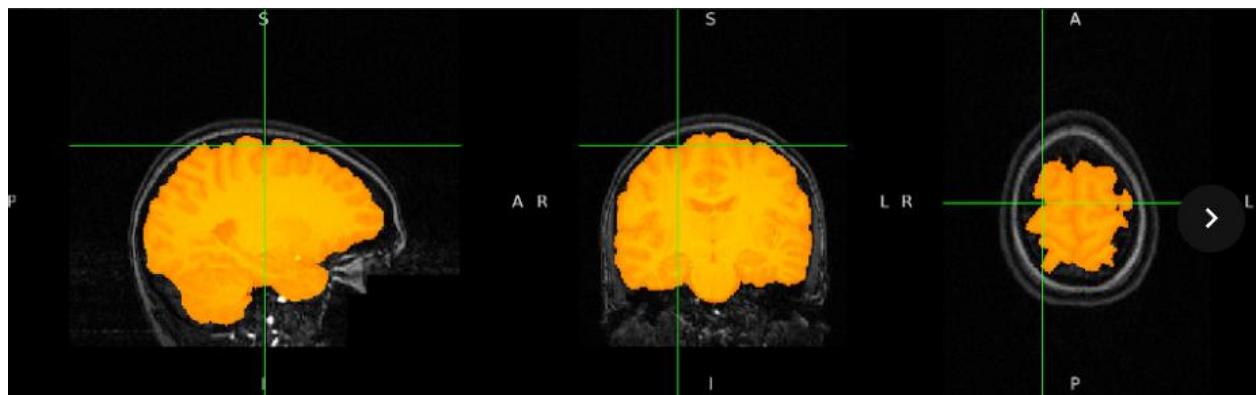
Again, this threshold shows to be very good as it includes almost all the brain perfectly while removing most of the skull. The brain is the yellow part.



- Using a threshold of 0.5:

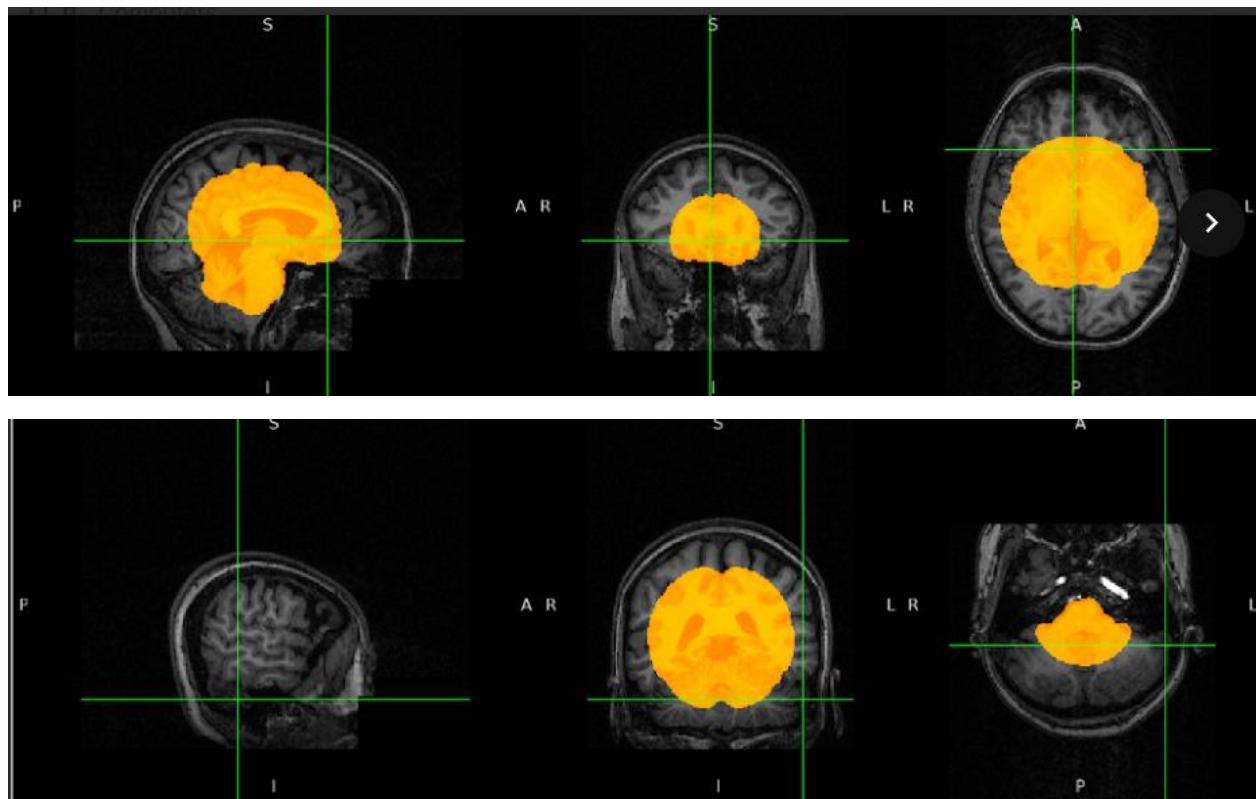
Not as bad as higher thresholds. The skull is fully excluded, However, some regions the brain got excluded.





- Using a threshold of 0.9:

Almost half of the brain got removed when we used a threshold of 0.9. It has conducted a very aggressive brain extraction.

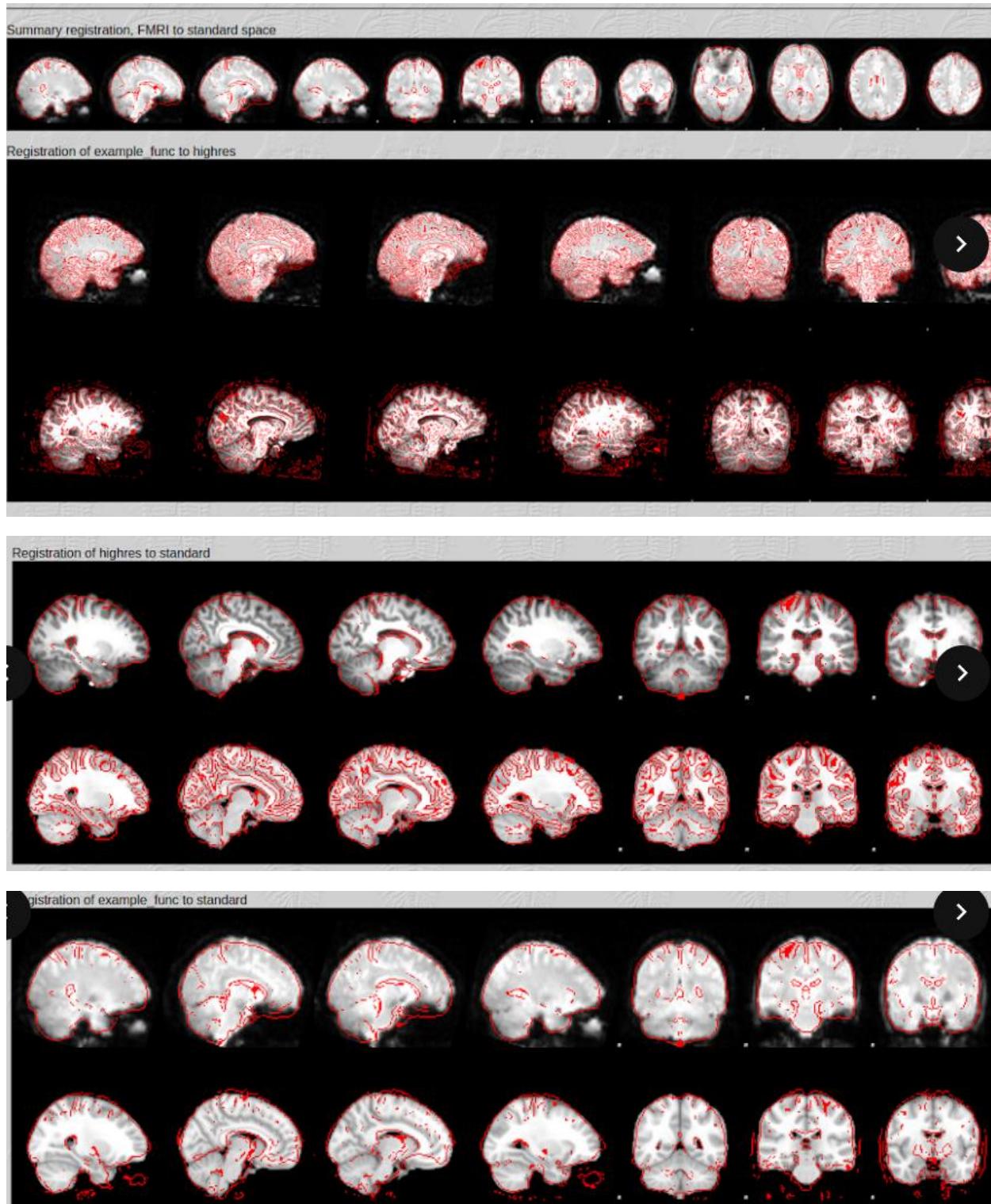


2) Feat Analysis – Run1:

Using a 5mm kernel for smoothing, MCFLIRT for motion correction and analysis, MNI template with normal search-12DOF for normalization, and normal search-BBR for registration

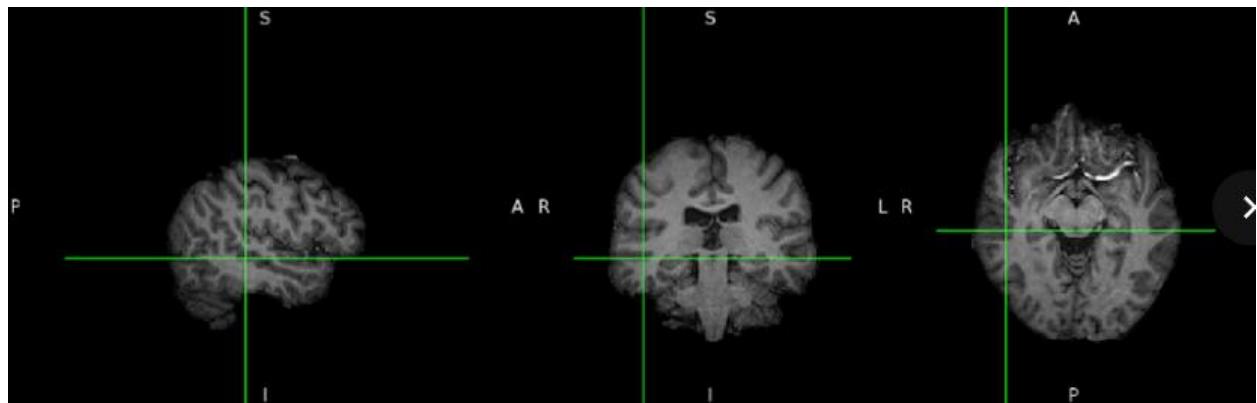
- Registration statistical results:

Looking at the attached images from all the results, we can see from the registration outcome that the structural image has been aligned very closely to the MNI template. Moreover, by overlaying the functional image over the processed image, we can visually test the alignment. The images are well aligned due to the normalization and registration steps. We can also notice the smoothing done by the 5mm kernel. Overall, images seem to be successfully preprocessed.

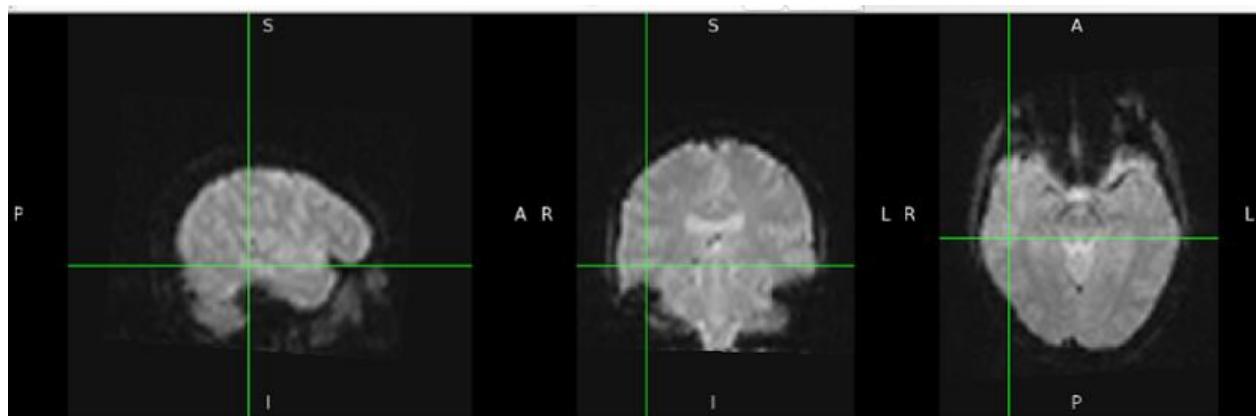


- **Visual test for alignment:**

highres.nii.gz



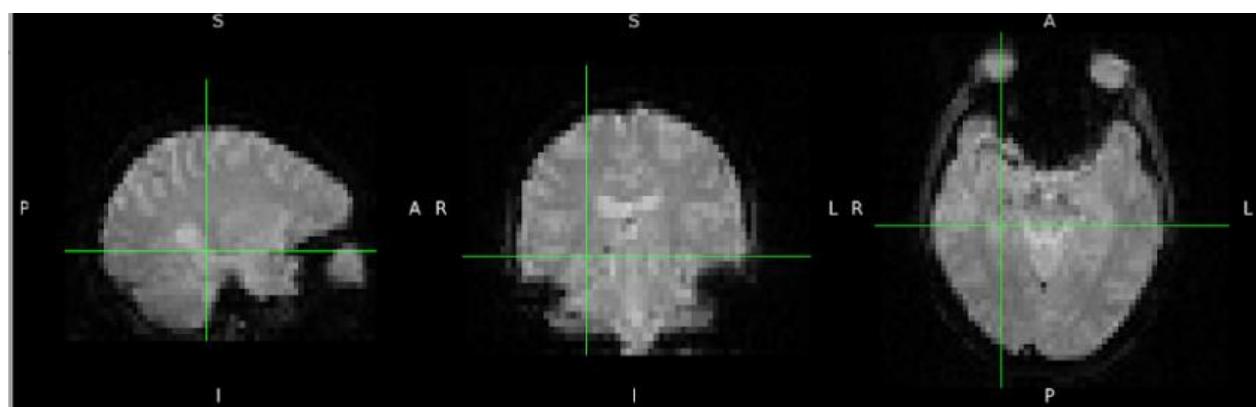
highres.nii.gz and *example_fun_highres.nii.gz* overlain on top of each other.



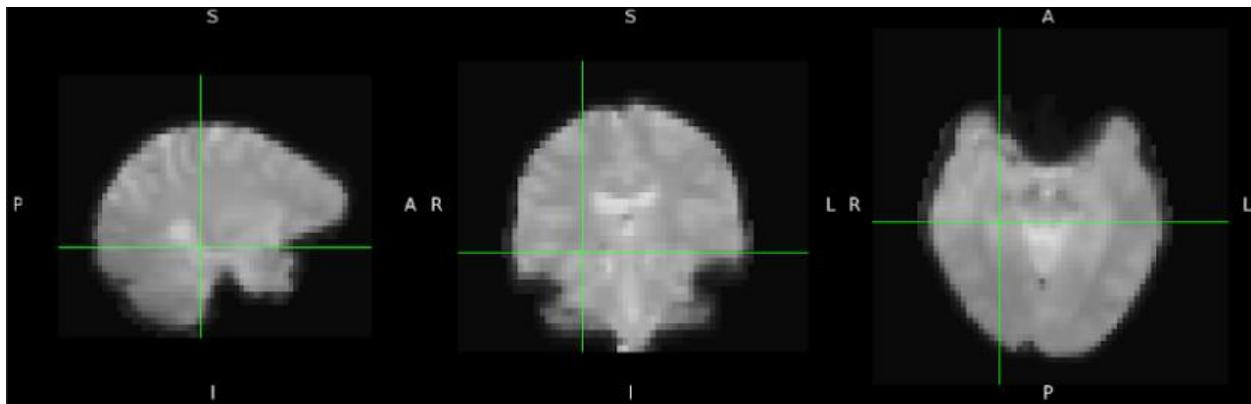
Observations: Images seem to be well aligned over each other.

- **Smoothing (and also alignment) result:**

example_func.nii.gz



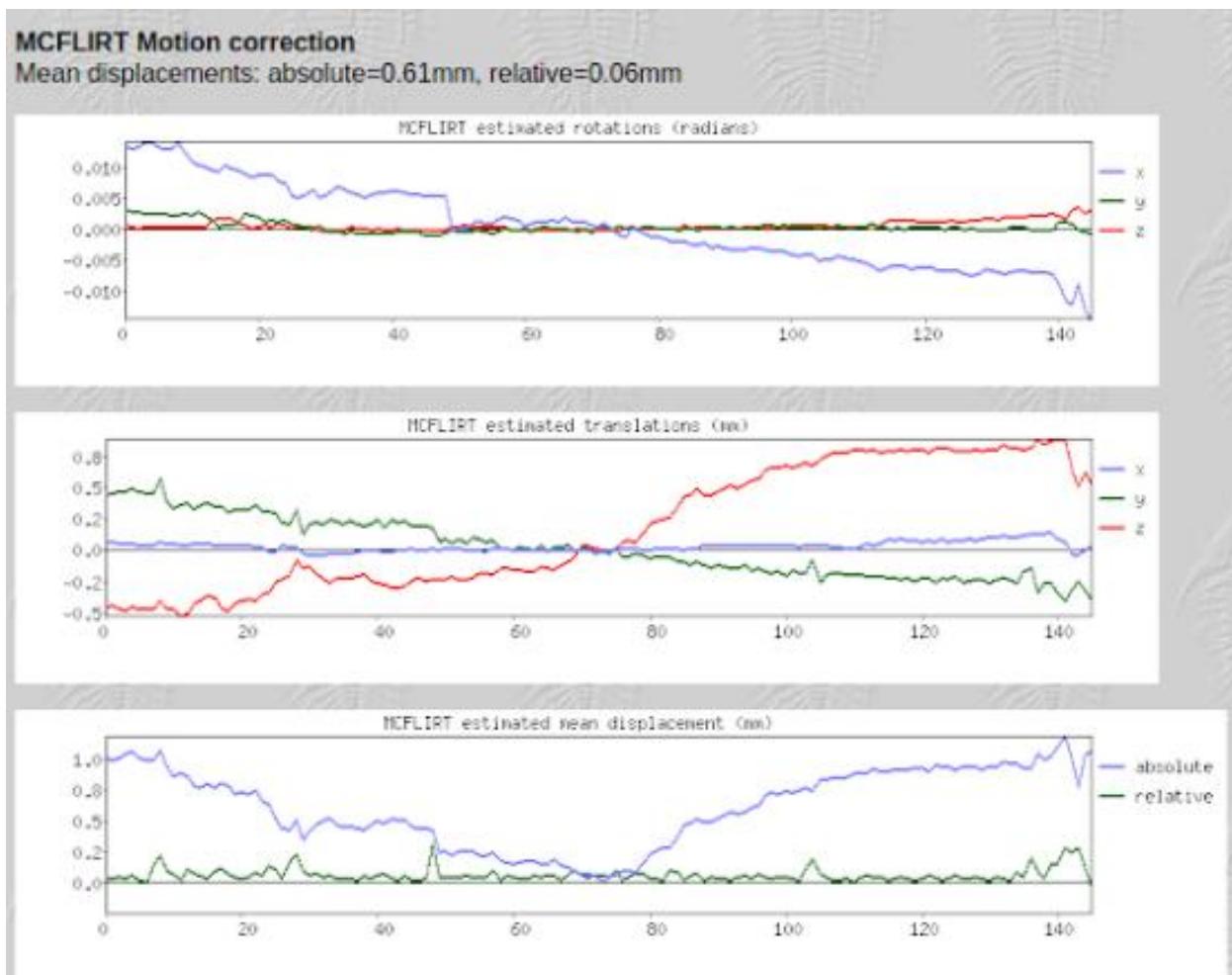
example_func.nii.gz and *filtered_func_data.nii.gz* overlain on top of each other.



Observation: Relevantly noticeable smoothing effect.

- **Motion analysis:**

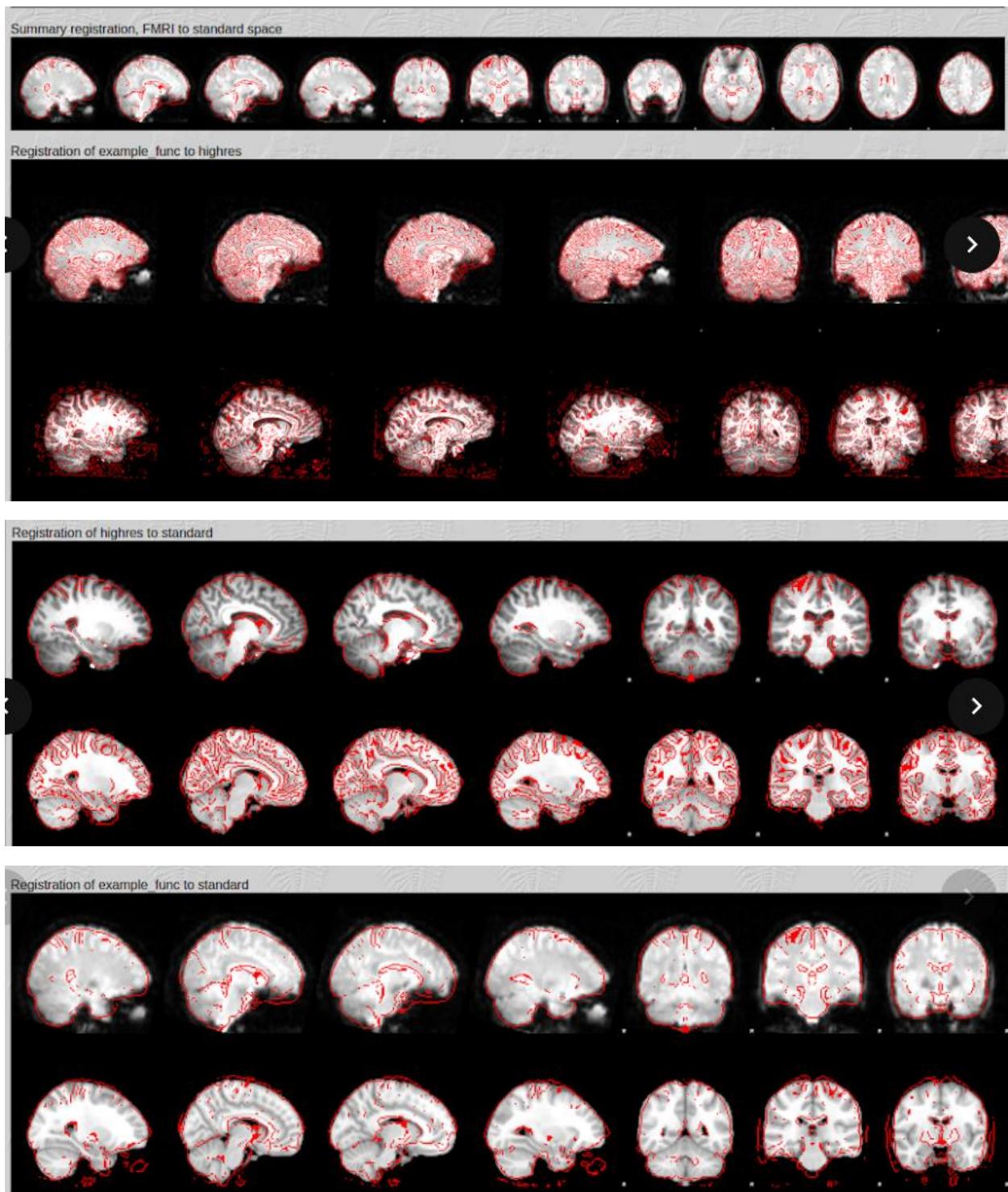
The motion analysis shows that the subject has generally moved at the first part of the run (rotations) and also in the second part (translations), leaving only a small part in the middle of the run with little to no motion.



3) Feat Analysis – Run2:

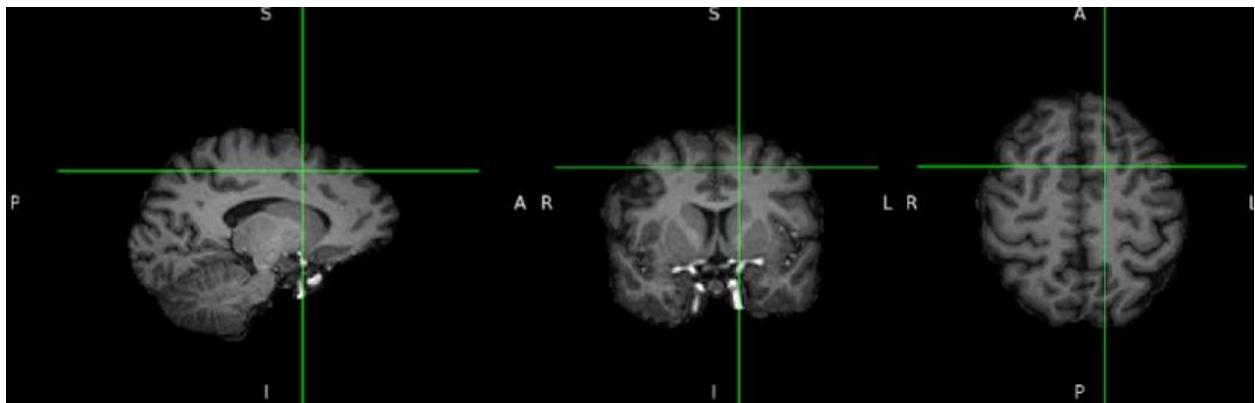
Same settings and as for run1. Outcome images are attached below.

- Registration statistical results: same observations as for run1.

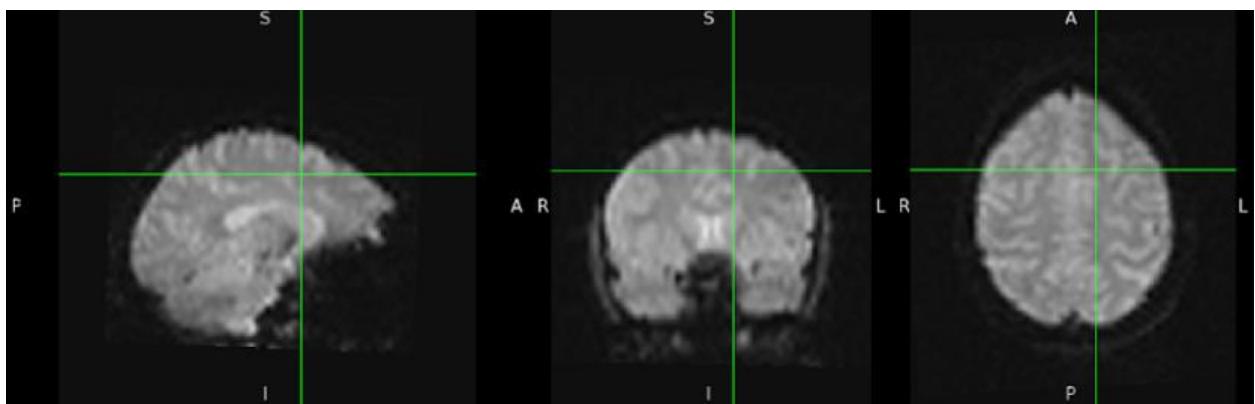


- Visual test for alignment:

highres.nii.gz

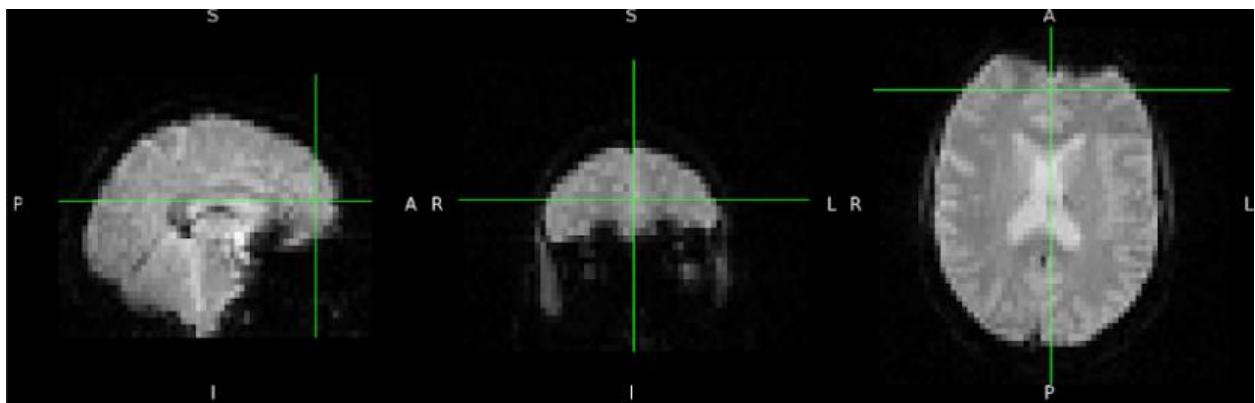


highres.nii.gz and *example_fun_highres.nii.gz* overlain on top of each other.

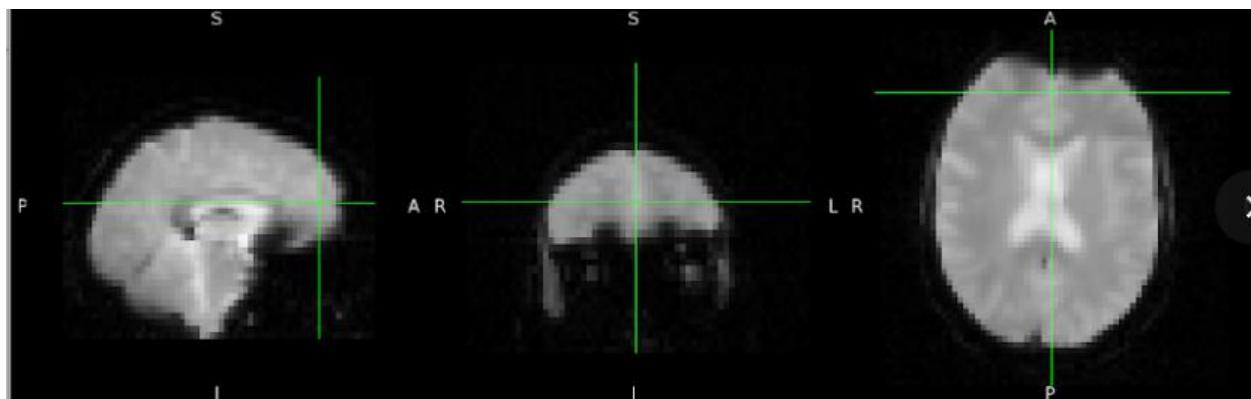


- Smoothing (and also alignment) result:

example_func.nii.gz

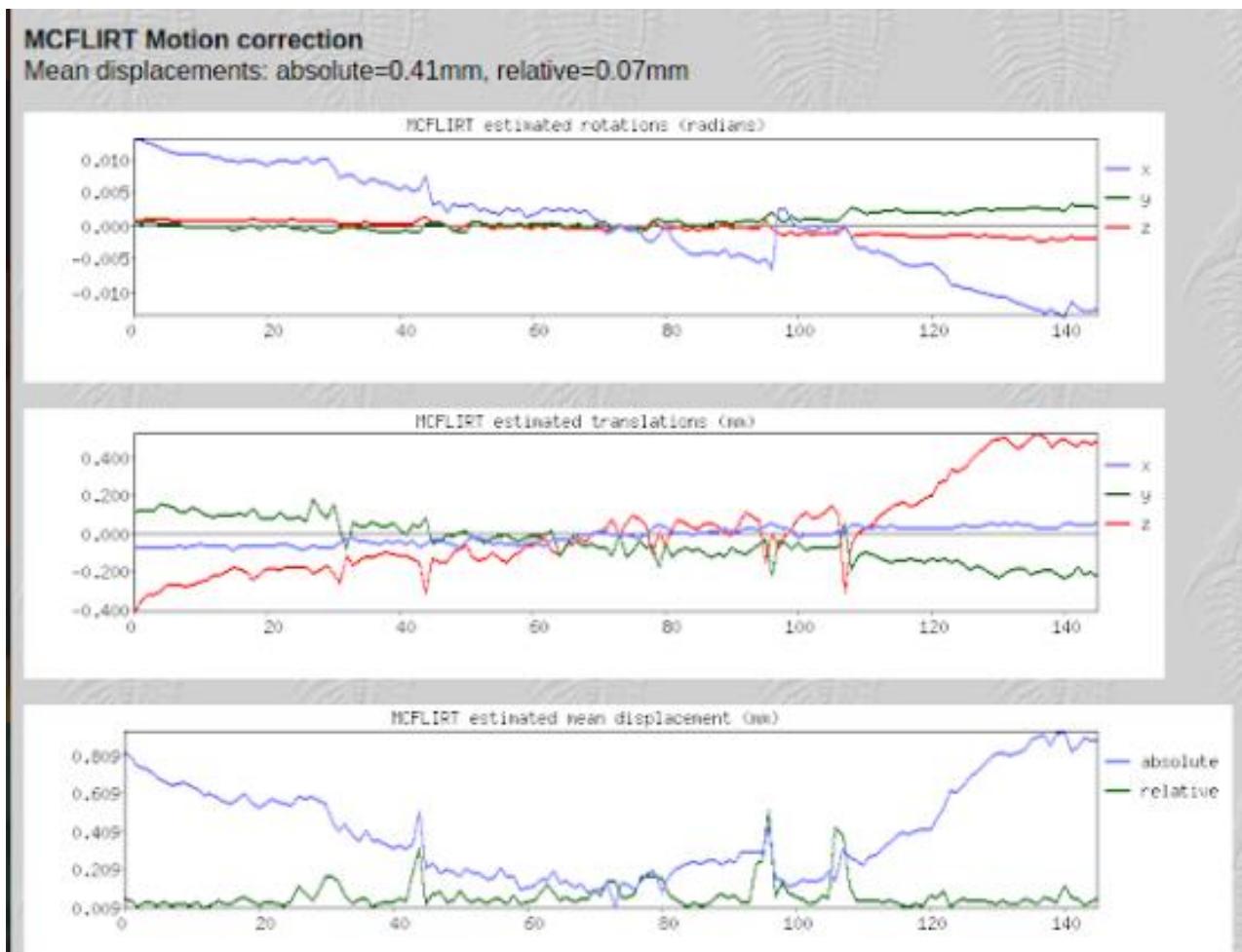


example_func.nii.gz and *filtered_func_data.nii.gz* overlain on top of each other.



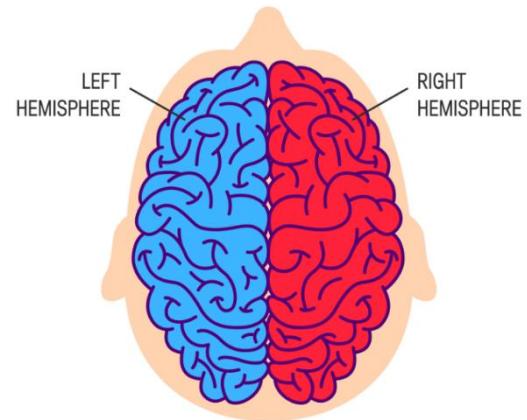
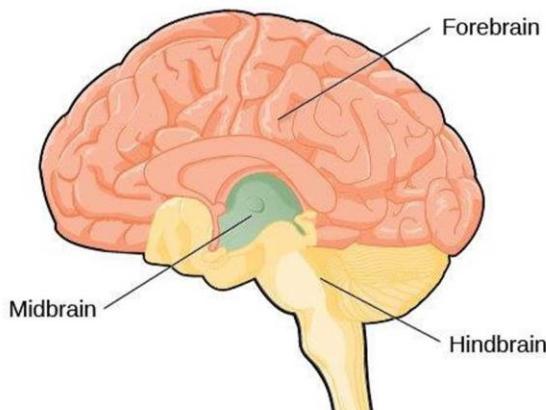
- Motion Analysis:

Same observations as for run1. Subject also moved throughout the first and second part, leaving the middle of the time series with relatively less motion.



Task 4: Neuroanatomy Regions and Functions:

▪ Major brain regions:

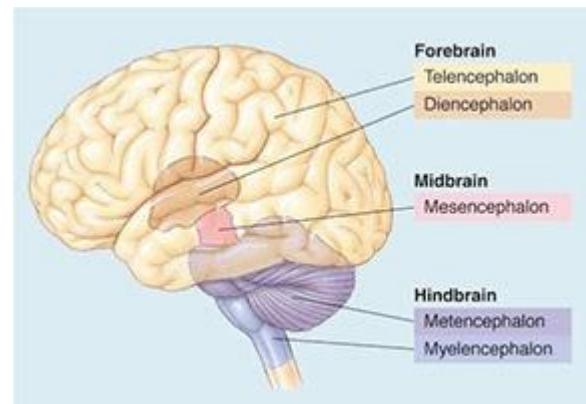


Forebrain/ Cerebrum

1. Telencephalon

Contains cerebral hemispheres that are responsible for:

1. Higher-level thinking (abstract reasoning)
2. Sensory interpretation (vision, touch, sound)
3. Voluntary muscle movements
4. Language production and comprehension
5. Personality traits and emotional responses

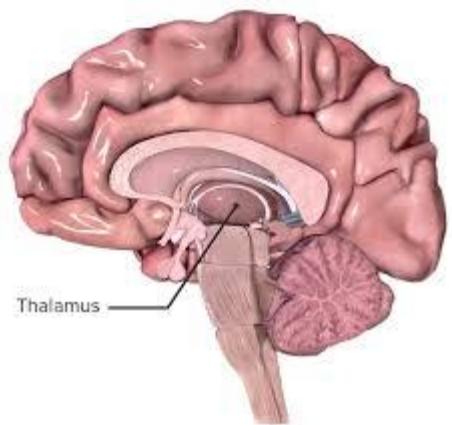
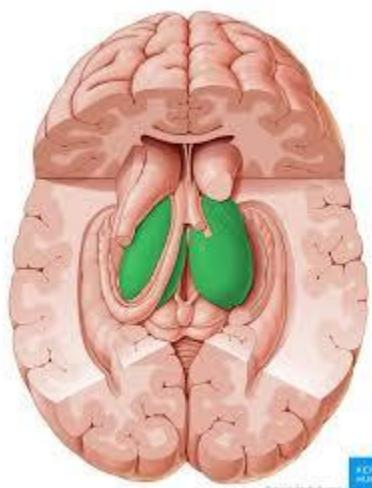


2. Diencephalon (sub cortical)

2.1 Thalamus

Functions:

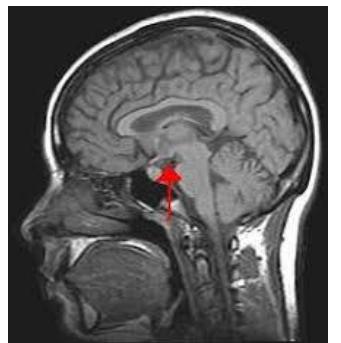
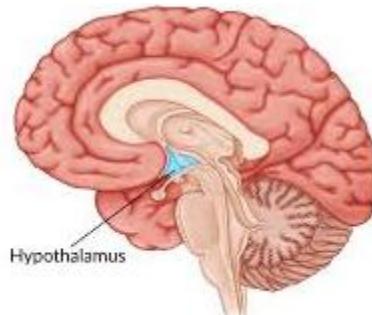
1. Sensory relay station (except smell)
2. Sleep regulation
3. Alertness and wakefulness control
4. Motor signal coordination
5. Emotional regulation (part of limbic relay)



2.2 Hypothalamus

Functions:

1. Temperature regulation ("thermostat" of the body)
2. Hunger and satiety control
3. Hormone release via pituitary gland control
4. Emotional response like rage and pleasure
5. Circadian rhythm management (biological clock)

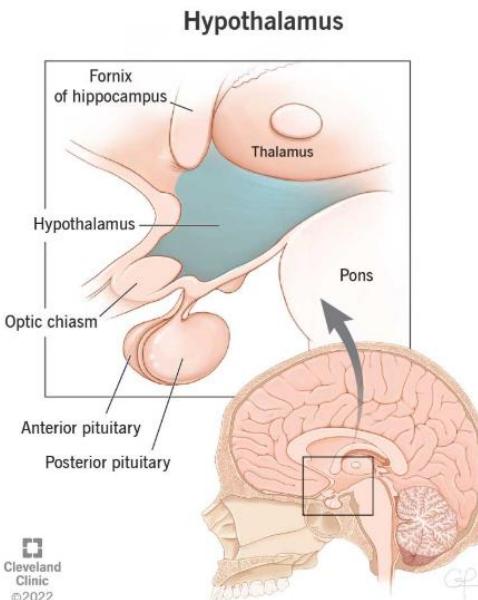


- *Inside the hypothalamus:*

2.2.1 Optic chiasm

Functions:

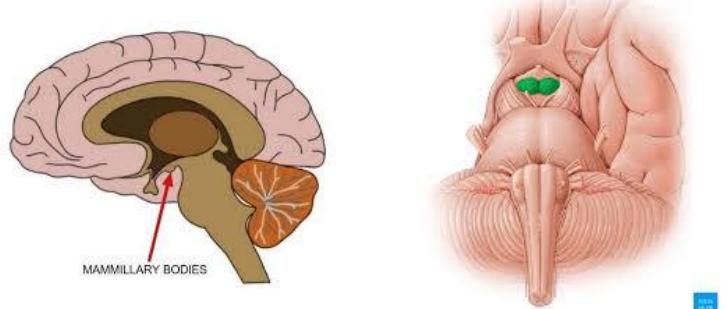
1. Crossing of visual signals
2. Depth perception enhancement
3. Binocular vision support
4. Peripheral vision coordination
5. Visual field integration



2.2.2 In Infundibulum (pituitary gland)

Functions:

1. Growth hormone release
2. Control of sexual maturation and reproduction
3. Water balance (ADH regulation)
4. Stress hormone release (ACTH → cortisol)
5. Thyroid regulation (via TSH)

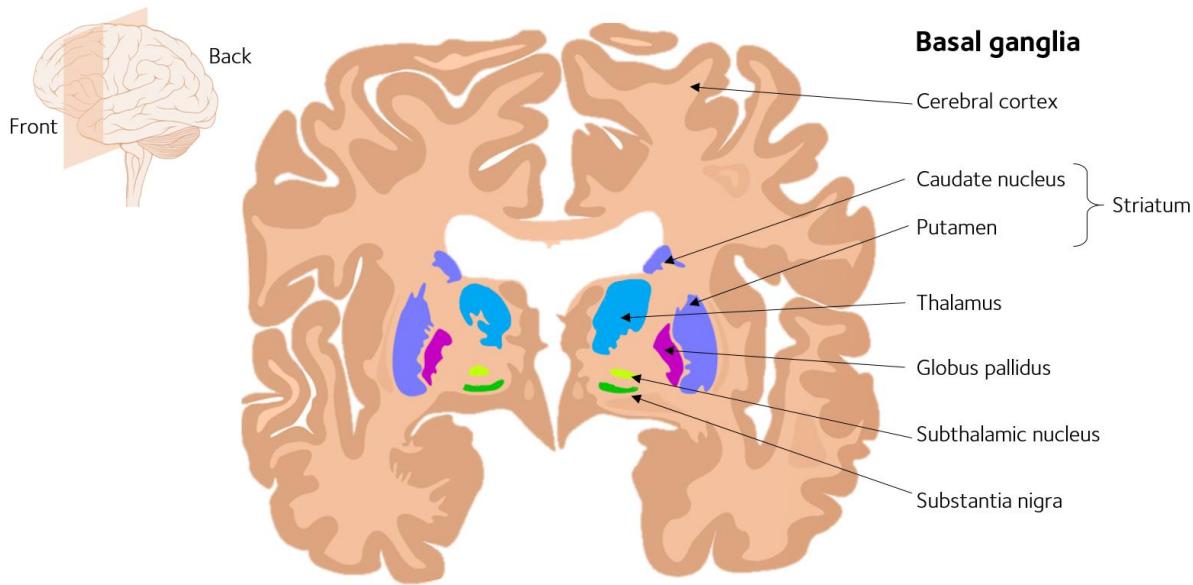


2.2.3 Mammillary bodies

Functions:

1. Memory recollection (especially spatial memory)
2. Relay information from hippocampus to thalamus
3. Emotional memory integration
4. Smell and memory association
5. Role in Wernicke-Korsakoff syndrome (clinical importance)

2.3 Basal Ganglia



2.3.1 Striatum

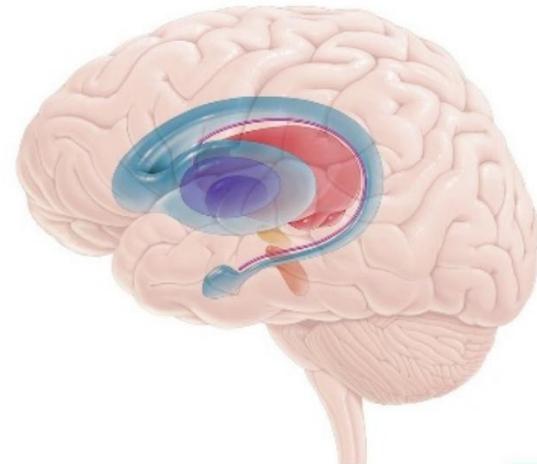
Responsible for:

1. Movement initiation
2. Habit formation
3. Reward and motivation processing
4. Cognitive flexibility
5. Action selection (choosing what to do)

➤ Caudate nucleus (head)

Functions:

1. Motor control precision
2. Goal-directed behaviour
3. Memory integration
4. Learning associations
5. Emotional regulation (OCD links)



➤ Lentiform nucleus (putamen)

Functions:

1. Voluntary movement regulation
2. Motor skill learning
3. Coordination of automatic movements
4. Reward-based habit formation
5. Processing of motivational stimuli

2.3.2 Lentiform nucleus (Globus pallidus)

Functions:

1. Inhibitory control of movement
2. Regulation of voluntary actions
3. Posture maintenance
4. Smooth muscle movement execution
5. Balancing excitatory and inhibitory motor signals

2.3.3 Thalamus lentiform

Functions:

1. Relay of motor and sensory signals
2. Regulation of consciousness
3. Sleep-wake transitions
4. Filtering sensory information
5. Processing movement coordination

2.3.4 Caudate lentiform

Functions:

1. Motor planning
2. Habitual behavior reinforcement
3. Cognitive integration with movement
4. Emotional learning circuits
5. Spatial working memory

2.4 Limbic System

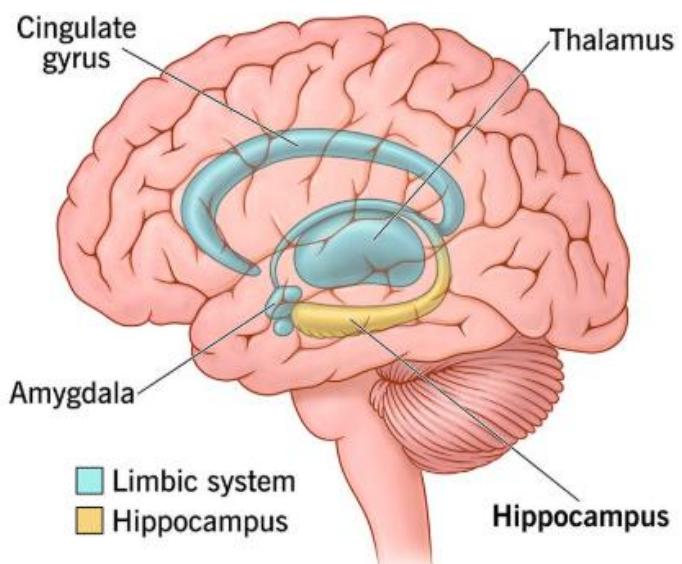
Responsible for:

1. Emotion processing
2. Memory formation
3. Behavioral motivation
4. Olfactory integration
5. Fear and reward evaluation

2.4.1 Hippocampus

Functions:

1. Formation of new memories
2. Spatial navigation



3. Memory consolidation (transferring short-term memories into long-term storage).
4. Emotional regulation (especially memories tied to emotions, working with the amygdala).
5. Learning

2.4.2 Amygdala

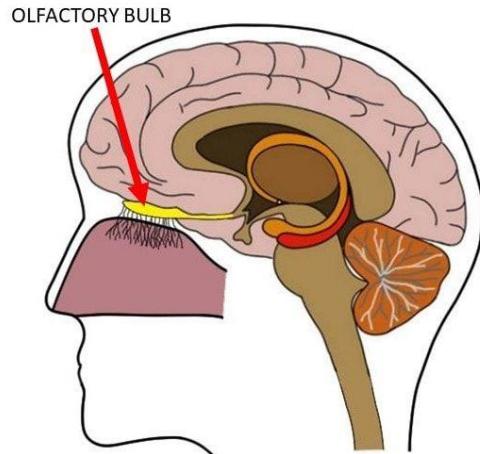
Functions:

1. Processing emotions
2. Emotional memory storage (attaching emotions to memories).
3. Fear response and threat detection
4. Modulation of aggression and social behaviours
5. Emotional learning

2.5 Olfactory Bulb

Functions:

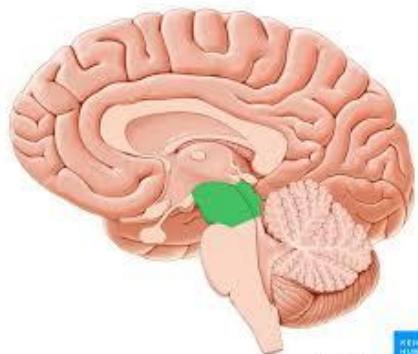
1. Smell detection
2. Emotional response to smells
3. Memory triggers through smell
4. Flavor perception (smell + taste integration)
5. Pheromone detection (social behaviors)



Midbrain

Functions:

1. Reflexive eye movements
2. Auditory processing (turn head to a sound)
3. Visual tracking
4. Pain suppression pathways (PAG — periaqueductal gray)
5. Basic motor control (start/stop movements)



Hindbrain

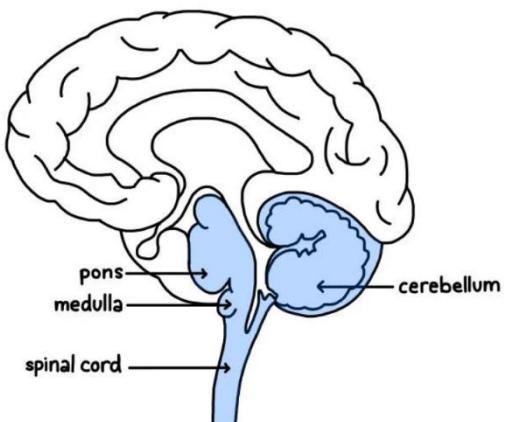
1. Brainstem

1.1 Pons

Functions:

1. Dream regulation (REM sleep)
2. Breathing rhythm fine-tuning
3. Relay between cerebrum and cerebellum
4. Hearing and balance assistance
5. Facial expressions motor control

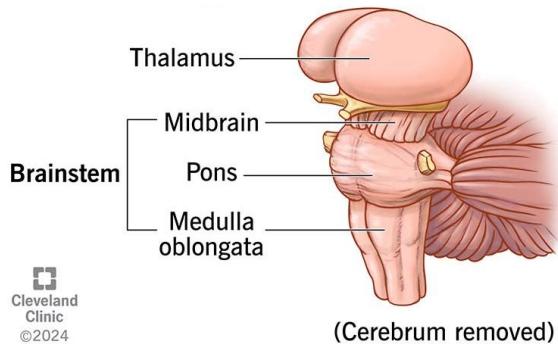
HINDBRAIN STRUCTURES



1.2 Medulla Oblongata

Functions:

1. Heart rate regulation
2. Breathing control
3. Blood pressure regulation
4. Vomiting reflex coordination
5. Coughing/sneezing/swallowing reflexes



(Cerebrum removed)

1.2.1 Pyramids

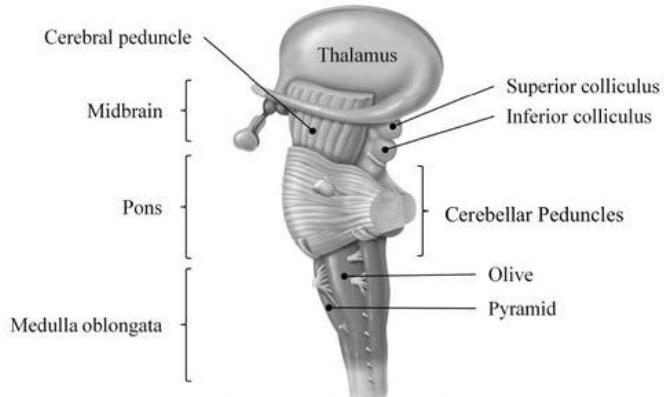
Functions:

1. Conduction of motor signals
2. Decussation (crossing over) of motor fibers
3. Voluntary movement control
4. Integration of motor output before reaching lower motor neurons.
5. Relay of corticospinal tract fibers

1.2.2 Inferior Olives

Functions:

1. Motor learning
2. Coordination of movement
3. Timing and precision of motor commands.
4. Error detection during movement (comparing intended vs. actual movement).
5. Sensory processing related to proprioception (body position sense).



1.2.3 Cerebral peduncles (in Midbrain but part of medulla oblongata)

Functions:

1. Carry motor output from cerebral cortex to brainstem and spinal cord.
2. Transmit corticospinal tract fibers
3. Transmit corticopontine fibers
4. Transmit corticobulbar tract fibers
5. Anchor the cerebrum to the brainstem structurally.

2. Cerebellum

- Cerebellar hemisphere

Functions:

1. Coordination of voluntary movements
2. Motor planning
3. Balance and posture regulation
4. Motor learning (adjusting motor programs through practice, like learning to ride a bike).
5. Cognitive functions



The 2 hemispheres are interconnected by

2.1 Corpus callosum

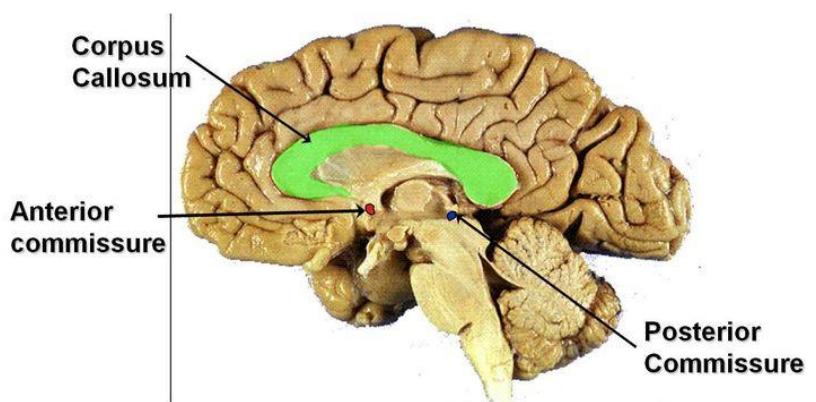
Functions:

1. Communication between the two cerebral hemispheres.
2. Integration of sensory, motor, and cognitive performances across sides.
3. Transfer of learned information and experiences (like skills learned by one side being shared with the other).
4. Coordination of bimanual motor activities (like using both hands together).
5. Support of lateralized brain functions

2.2 Anterior Commissure

Functions:

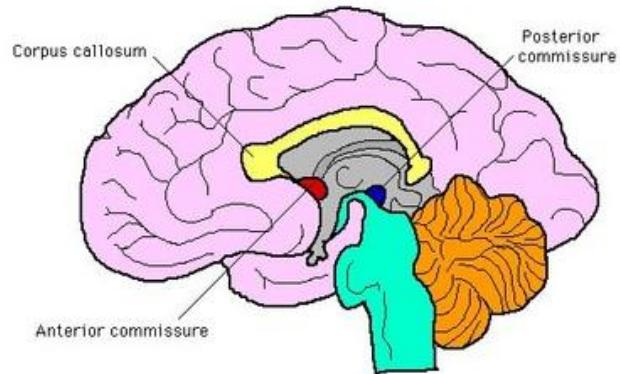
1. Connects temporal lobes
2. Involved in pain sensation
3. Smell information transfer
4. Memory linking across hemispheres
5. Emotional information sharing



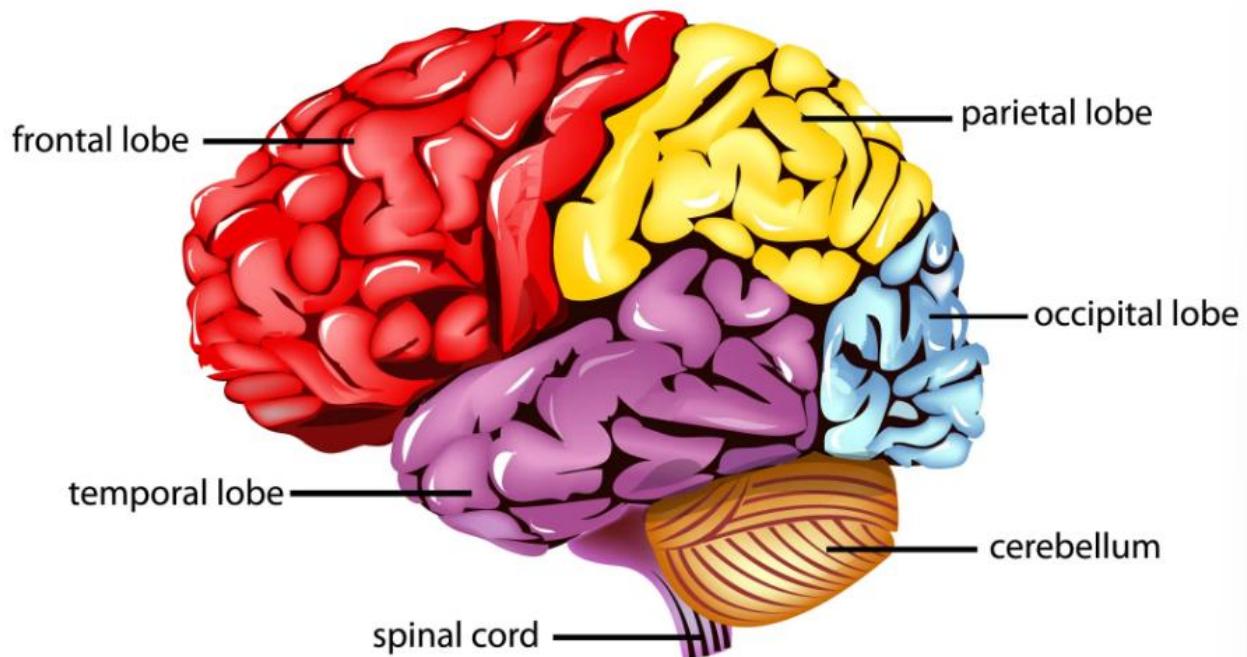
2.3 Posterior Commissure

Functions:

1. Visual reflexes control
2. Eye movement coordination
3. Light reflex (pupil constriction)
4. Motor coordination during light changes
5. Connects midbrain structures



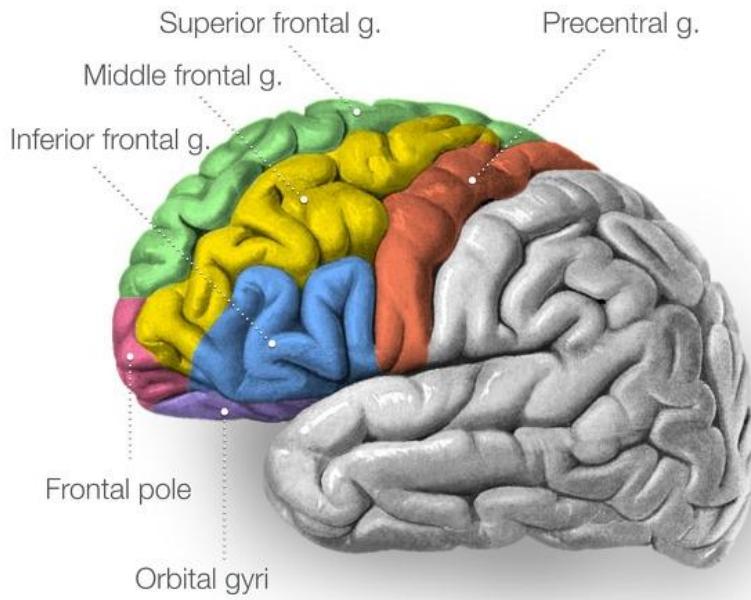
Cerebral Cortex



1. Frontal Lobe

Overall responsible for:

- 1 Voluntary motor control
- 2 Executive functions
- 3 Speech production
- 4 Personality and emotional regulation
- 5 Working memory



1.1 Superior frontal gyrus

Functions:

1. Self-awareness
2. Working memory
3. Voluntary eye movements
4. Decision making (deliberate choice)
5. Planning future actions

1.2 Middle frontal gyrus (MFG)

Functions:

1. Attention control
2. Goal setting and tracking
3. Abstract reasoning
4. Cognitive flexibility (switch tasks)
5. Inhibition of impulsive behaviors

1.3 Inferior frontal gyrus (IFG)

Functions:

1. Language production (Broca's area)
2. Response inhibition
3. Risk assessment
4. Empathy and theory of mind
5. Rhythm perception (music and language)

1.4 Precentral gyrus (Primary motor cortex, M1)

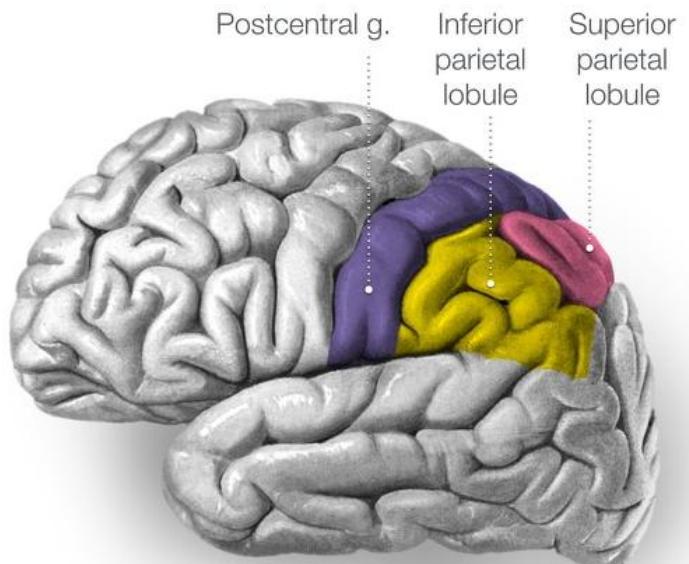
Functions:

1. Voluntary muscle movement initiation
2. Fine motor control (fingers, tongue)
3. Motor map organization (homunculus)
4. Coordination of bilateral movements
5. Strength force generation

2. Parietal Lobe

Overall responsible for:

1. Processing somatosensory information
2. Spatial awareness and navigation
3. Integration of sensory input
4. Mathematical reasoning
5. Attention and perception



2.1 Postcentral gyrus (Primary somatosensory cortex, S1)

Functions:

1. Processing touch
2. Pain perception
3. Temperature sensation
4. Proprioception (body position sense)
5. Sensory discrimination (texture, shape)

2.2 Superior parietal lobule

Functions:

1. Spatial orientation
2. Hand-eye coordination
3. Object manipulation (fine touch integration)
4. Visual attention
5. Mental rotation of objects

2.3 Inferior parietal lobule

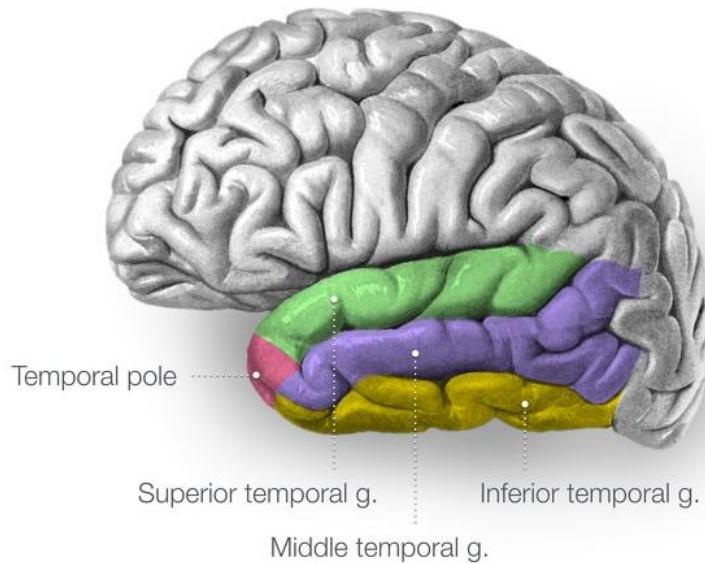
Functions:

1. Mathematical reasoning
2. Language comprehension
3. Body image perception
4. Tool use planning
5. Understanding intentions of others (social cognition)

3. Temporal Lobe

Overall responsible for:

1. Auditory processing
2. Language comprehension
3. Memory formation
4. Emotional response
5. Object and face recognition



3.1 Superior temporal gyrus

Functions:

1. Auditory processing (primary hearing center)
2. Language comprehension (Wernicke's area)
3. Sound localization
4. Voice recognition
5. Emotional tone decoding (prosody)

3.2 Middle temporal gyrus

Functions:

1. Word meaning storage
2. Concept retrieval (memory)
3. Perception of distance
4. Emotional memory linkage
5. Recognizing moving objects

3.3 Inferior temporal gyrus

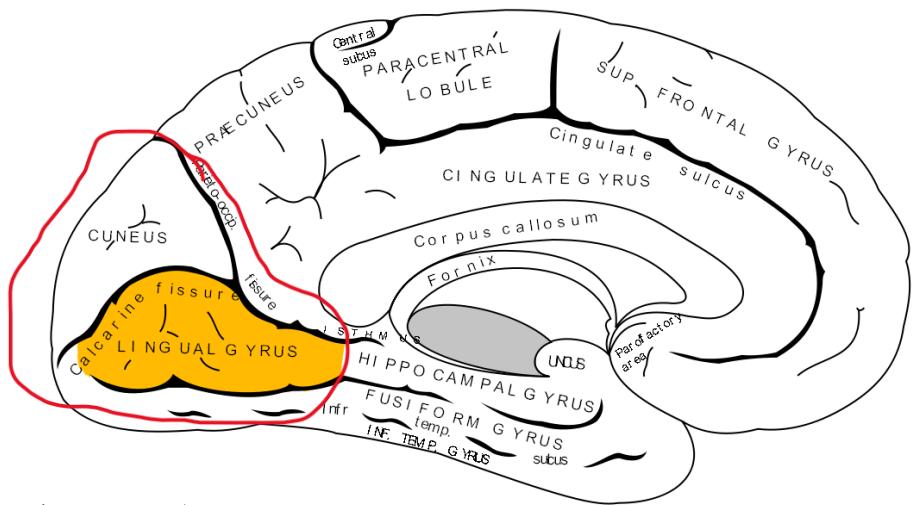
Functions:

1. Visual object recognition (faces, objects)
2. Colour perception
3. Complex scene analysis
4. Memory visual reinforcement
5. Learning through visual experiences

4. Occipital Lobe

Overall responsible for:

1. Visual perception
2. Color recognition
3. Depth perception
4. Movement detection
5. Visual memory storage



4.1 Cuneus

Functions:

1. Basic visual processing (visual cortex area)
2. Attention to visual stimuli
3. Visual imagery generation
4. Reading and interpretation of text
5. Mental reconstruction of visual scenes

4.2 Lingual gyrus

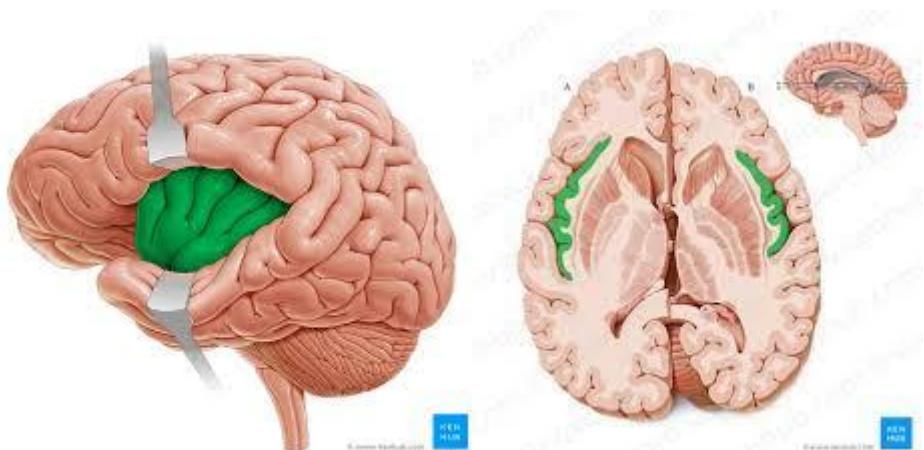
Functions:

1. Visual memory retrieval
2. Letter and word recognition (reading)
3. Colour processing
4. Visual imagination (creating mental images)
5. Dream visualization

5. Insula (5th lobe)

Functions:

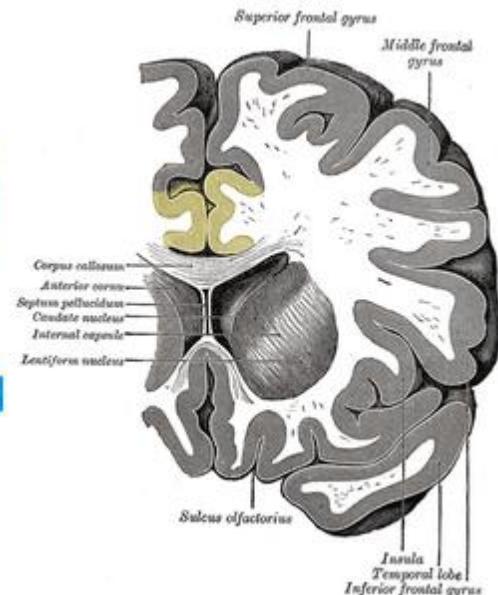
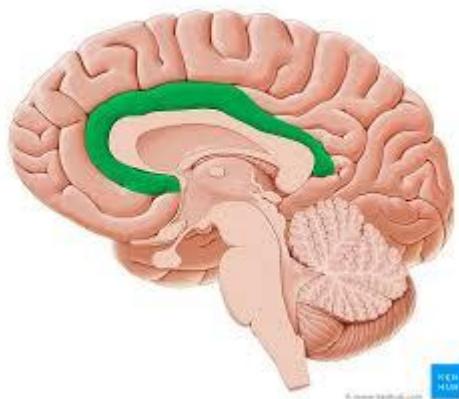
1. Emotional self-awareness
2. Empathy (feeling others' pain)
3. Gustatory processing (taste)
4. Visceral sensations (gut feelings)
5. Risk prediction and uncertainty evaluation



6. Cingulate Gyrus

Functions:

1. Emotional regulation
2. Decision-making under conflict
3. Pain emotional response
4. Autonomic regulation (heart rate, BP)
5. Reward-based learning

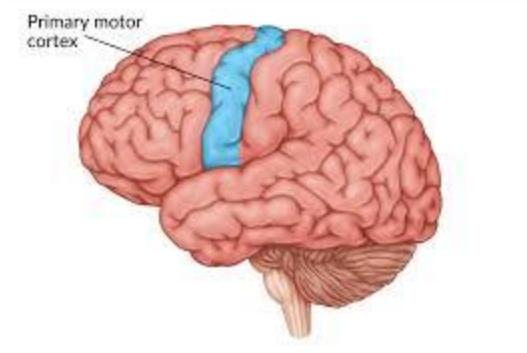


Functional Regions in The Cortex

1. Primary motor cortex (M1)

Functions:

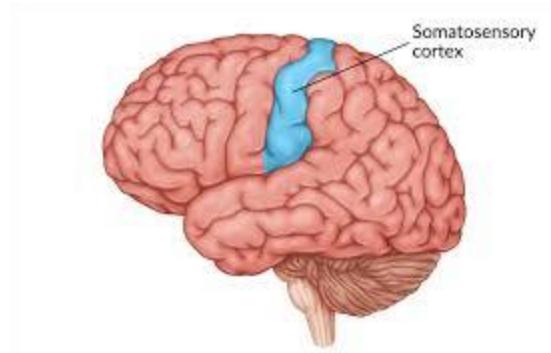
1. Initiates voluntary muscle movements
2. Controls fine motor skills
3. Executes movements based on planned actions
4. Maintains body posture through motor signals.
5. Adapts motor output with learning and practice

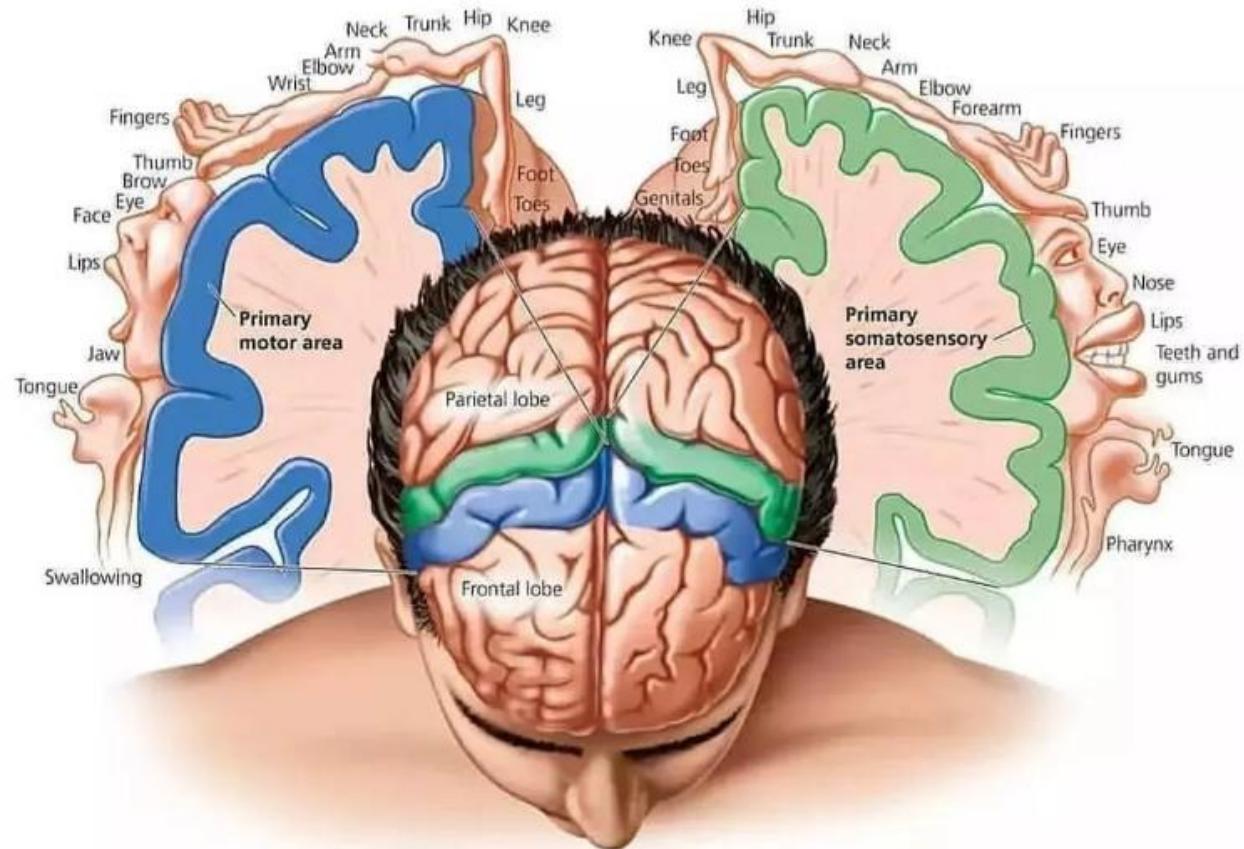


2. Primary somatosensory cortex (S1)

Functions:

1. Processes tactile (touch) information
2. Processes proprioception
3. Processes pain (nociception) and temperature sensations.
4. Distinguishes intensity and location of stimuli
5. Integrates sensory input for perception and movement guidance.



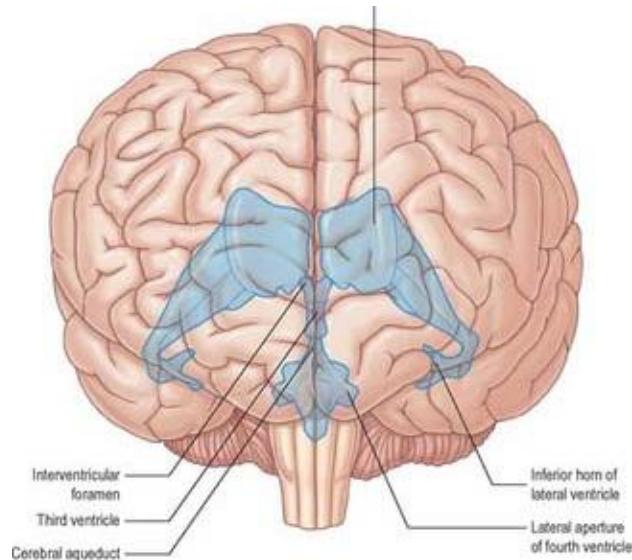


Ventricular System

1. Lateral ventricles

Functions:

1. Cerebrospinal fluid (CSF) production
2. Brain cushioning
3. Nutrient delivery
4. Waste removal
5. Pressure regulation



2. Third ventricle (3V)

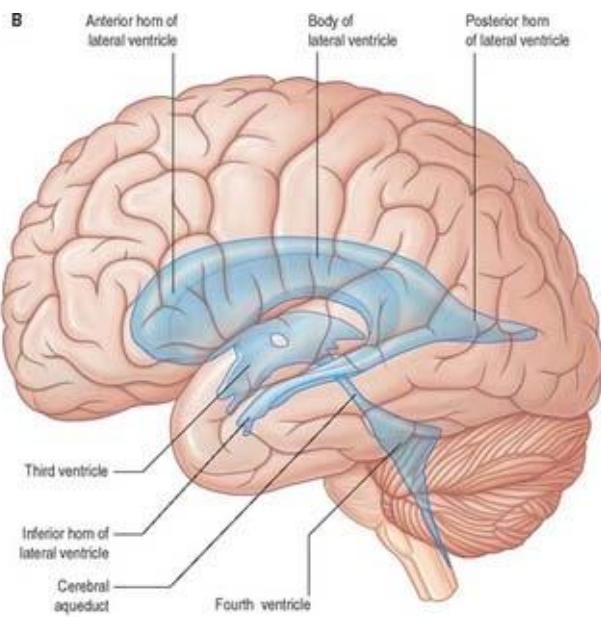
Functions:

1. CSF passageway
2. Communication between lateral and fourth ventricles
3. Hormone regulation (near hypothalamus)
4. Brain cushioning
5. Midline structure support

3. Fourth ventricle

Functions:

1. CSF flow continuation
2. Brainstem protection
3. Cushioning cerebellum and brainstem
4. CSF storage before exiting brain
5. Pressure balance maintenance



4. Cerebral aqueduct

Functions:

1. CSF connection between 3V and 4V
2. Regulation of CSF pressure
3. Narrowest CSF path (clinical importance: hydrocephalus)
4. Midbrain support
5. Relays auditory and visual stimuli information

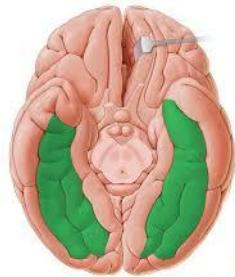
Task 5: Report on The Fusiform Gyrus

The fusiform gyrus (FG), also known as the lateral occipitotemporal gyrus, is a major structure located on the basal surface of the temporal and occipital lobes of the cerebral hemispheres. It plays a central role in specialized high-level visual functions, including face perception, object recognition, and reading.

Anatomy

The FG is the largest macro-anatomical structure within the ventral temporal cortex, an area primarily responsible for high-level visual processing. Its name reflects its shape — broader at the center and tapering toward both ends.

Structurally, the FG is composed of two occipitotemporal gyri: the medial occipitotemporal gyrus and the lateral occipitotemporal gyrus. These two gyri are divided by a distinct groove known as the mid-fusiform sulcus. The mid-fusiform sulcus also serves as a key landmark for locating an important functional area within the FG — the fusiform face area — which is involved in face recognition and processing. Anatomically, the FG is bordered by the collateral sulcus, which separates it from the parahippocampal gyrus, and by the occipitotemporal sulcus, which separates it from the inferior temporal gyrus. As the largest gyrus in the ventral temporal lobe, the FG contributes to Brodmann area 37, along with the inferior and medial temporal gyri.



Functions of the Fusiform Gyrus

The FG plays a key role in higher-order visual processing, particularly in the identification and differentiation of objects. Beyond visual functions, it also contributes to memory, multisensory integration, and perception. Specific regions within the FG have been associated with specialized tasks such as face recognition, body recognition, word recognition, and the identification of features within distinct categories. Additionally, the FG maintains a close functional relationship with the angular gyrus, a structure involved in the advanced processing of colour. Through its communication with both the visual pathway and the angular gyrus, the FG supports the association between colours and shapes.

Fusiform face area

The fusiform face area (FFA) is located on the lateral surface of the mid-portion of the FG. Its name reflects its crucial role in the identification of faces, including the recognition of one's own face. However, the FFA is not solely responsible for face identification; rather, it functions as part of a broader network of face-processing regions within the cortex, which also includes adjacent areas of the occipital lobe. In addition to facial recognition, the FG contributes to the perception of emotions conveyed through facial expressions.

Clinical Associations

The FG has been associated with several neurological conditions, including prosopagnosia, synesthesia, and dyslexia. Prosopagnosia, also known as face blindness, is characterized by an inability to recognize familiar faces, including one's own. It can occur as a result of isolated lesions to the fusiform face area (FFA) within the FG. Synesthesia is a condition in which stimulation of one sensory pathway involuntarily triggers experiences in a second sensory pathway — for example, perceiving colors when hearing certain sounds. Dyslexia, meanwhile, is characterized by difficulties in reading and has been linked to functional alterations in regions of the FG.

Conclusion

The FG plays a vital role in high-level visual processing, including face recognition, object identification, and reading. Its functional importance is further highlighted by its association with neurological conditions such as prosopagnosia, synesthesia, and dyslexia. Understanding this structure helps explain how the brain processes complex sensory information.

Task 6: First-Level Analysis:

Once we've finished preprocessing the functional runs for subjects 08 and 22, it's time to actually model the data. Each fMRI run is basically a sequence of brain scans — a stack of volumes stitched together over time. At every voxel (that's each tiny 3D pixel in the brain), we track changes in signal across the entire run, which gives us what's called a time series.

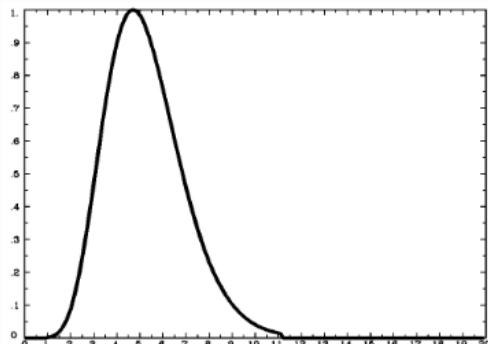
fMRI captures brain activity indirectly by measuring the BOLD signal — short for blood-oxygen-level-dependent signal. This signal doesn't measure neurons firing directly, but it gives us a window into brain activity by showing where blood flow increases. To figure out which brain areas react significantly to a certain stimulus, we need a good model of how neural activity translates into changes in blood flow — in other words, we need to understand the shape of the BOLD response.

Back in the 1990s, researchers noticed a consistent pattern: when a stimulus was shown, active brain regions showed a spike in BOLD signal, peaking about six seconds in and gradually returning to baseline. This predictable shape can be modeled using a Gamma Distribution. When we tune the Gamma parameters to reflect what's typically seen in studies, we get what's called the *canonical Hemodynamic Response Function* (HRF).

In the context of fMRI modeling, this Gamma Distribution is used as a *basis function* — essentially the core building block of our model. It helps us fit our data because, if we know what the response looks like for a quick blip of a stimulus, we can estimate how it should look for longer or more complex patterns of stimulation over time.

The HRF for a Single Impulse Stimulus:

When a stimulus is extremely short — like a finger snap — it's referred to as an *impulse stimulus*, meaning it essentially has no duration. In response to such a quick event, the BOLD signal typically follows the shape of a Gamma distribution: a sharp peak early in the time axis followed by a gradual decline.



The HRF generated by a single impulse stimulus. In this figure, the stimulus occurs at timepoint 0 on the x-axis.

The HRF for a Single Boxcar Stimulus:

If the stimulus lasts longer — say the subject views a flashing checkerboard for 15 seconds — the BOLD response behaves differently. Instead of a sharp spike, we see a broader, more sustained response that stays elevated during the stimulus and returns to baseline afterward. This type of stimulus is known as a *boxcar stimulus* because of its rectangular shape, resembling a train boxcar.

To model this, we convolve the Gamma distribution (which captures the basic shape of the Hemodynamic Response Function, or HRF) with the boxcar function. Convolution, in this context, means blending or averaging

the two over time. The result is a stretched version of the original Gamma curve — wider and more sustained — that aligns with the longer stimulus duration and gradually drops back once the stimulus ends.

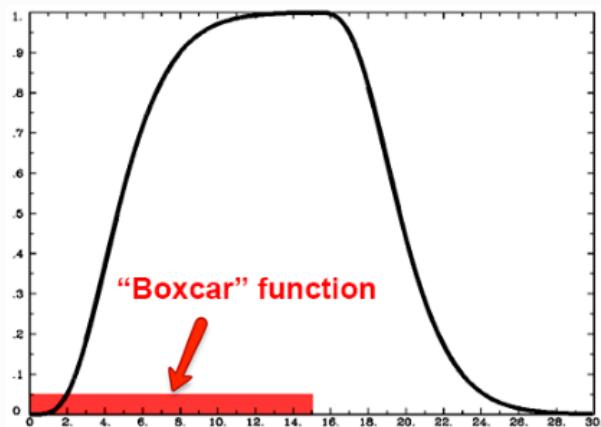
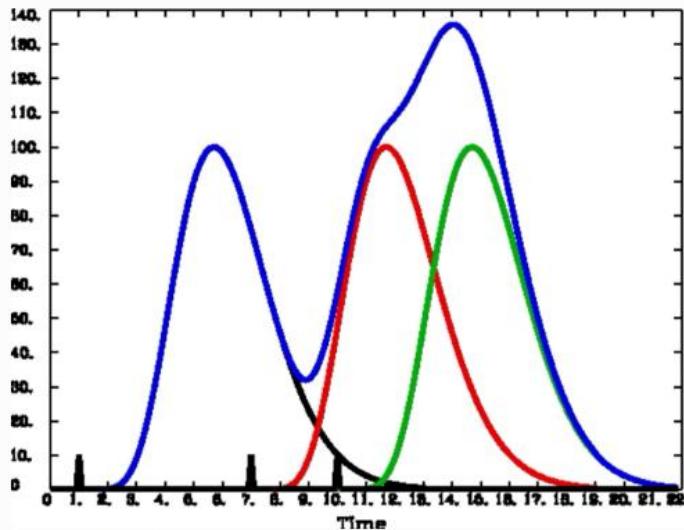


Illustration of the HRF generated by a boxcar stimulus lasting for fifteen seconds. Note that the BOLD signal begins descending back to baseline around the fifteen-second mark.

Things get more complex when stimuli are presented in rapid succession, before the BOLD signal from the previous one has fully returned to baseline. In this case, individual HRFs start to overlap, and we model the overall signal by summing them together. This results in a more intricate BOLD response that reflects a kind of moving average of the overlapping signals.



Convolution of the HRFs for individual stimuli. The overall BOLD response (blue) is a moving average of the individual HRFs outlined in black, red, and green. The vertical black lines on the x-axis represent impulse stimuli. Figure created by Bob Cox of AFNI.

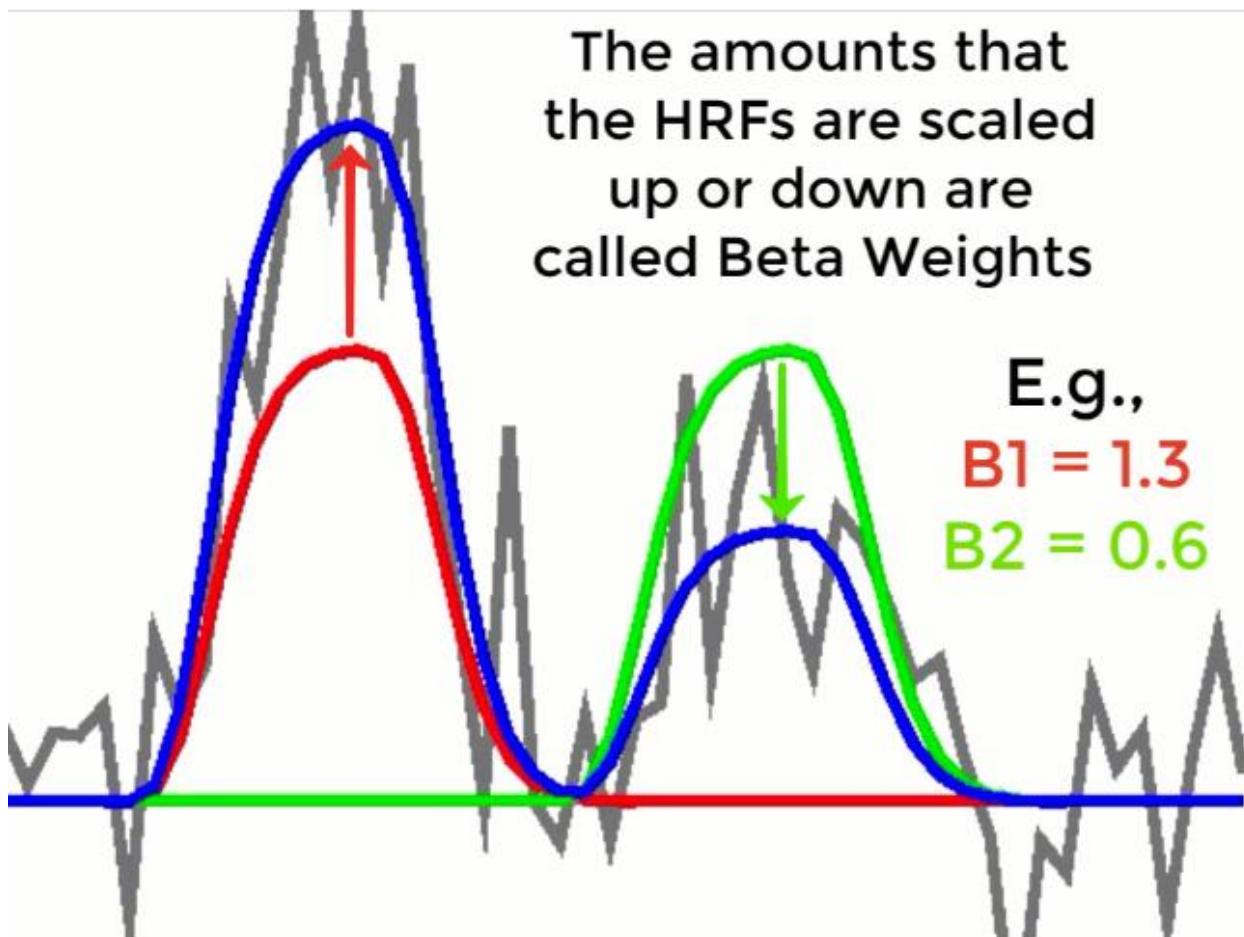
GENERAL LINEAR MODEL (GLM):

In a General Linear Model (GLM), we use one or more *regressors* — our independent variables — to model an outcome, which in this case is brain activity measured through the BOLD signal. The model assigns *beta weights* to each regressor, which basically reflect how much each one contributes to explaining the data. Any mismatch between the model's predictions and the actual data is called a *residual*.

For fMRI analysis, we typically include several regressors to represent different experimental conditions. Our main goal is to estimate the average amplitude of the BOLD response for each condition — that's our dependent variable.

In the image below, each colored line shows the BOLD response for a different condition (like Congruent vs. Incongruent), while the gray line shows the actual preprocessed data over time. The model tries to match the colored responses to the data as closely as possible. Some conditions might have a stronger (higher) response, others weaker — and sometimes the response might be close to zero or even negative.

The red and green curves represent the *ideal time-series* — what we expect the BOLD response to look like based on when each stimulus was presented. Once we estimate the beta weights to align this ideal response with the real data, we get the *fitted time-series* — the model's version of what the brain activity probably looked like.



Creating the Ideal Time-Series:

Before we can generate the fitted time-series and run a group-level analysis using the estimated beta weights, we need to construct what's known as the *ideal time-series* — our model of what we expect the BOLD signal to look like.

To do that, we start with the Flanker dataset. Inside each subject's func folder, there's an events.tsv file that holds all the critical timing info we need. Specifically, each file lists:

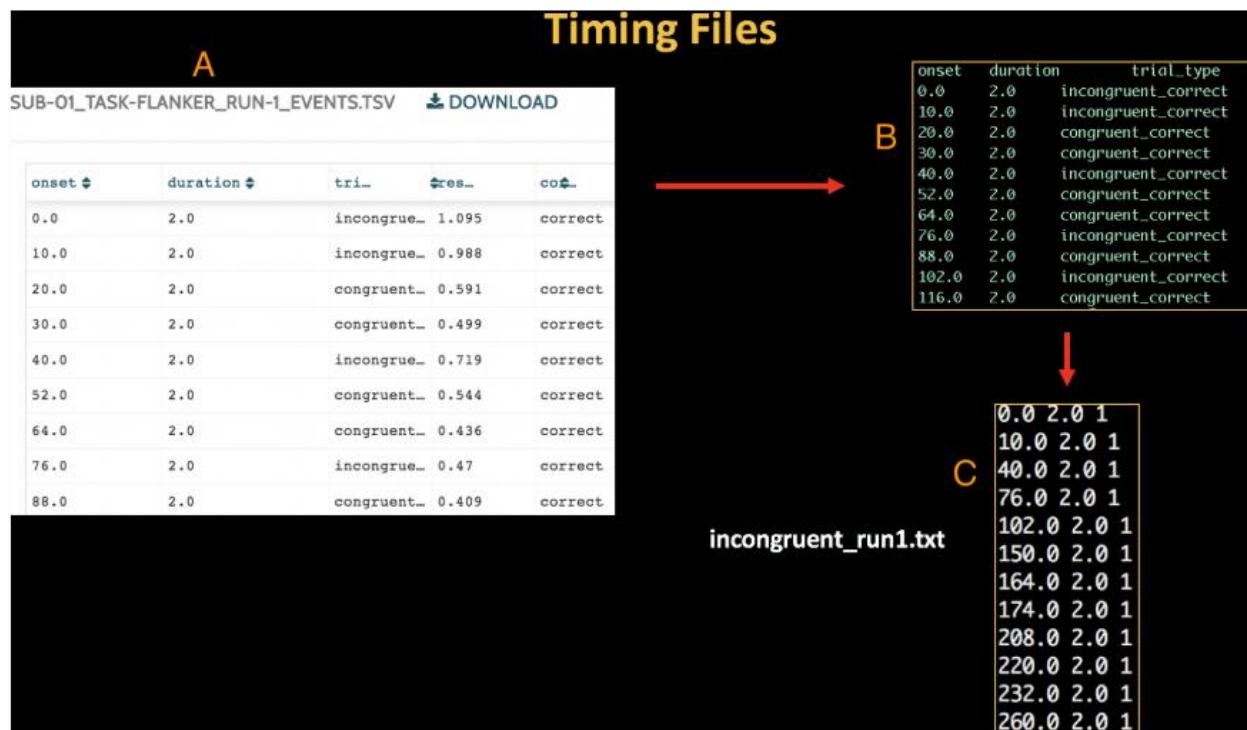
1. The condition name (e.g., Congruent or Incongruent),
2. The onset time of each trial, measured in seconds from the beginning of the scan,
3. The duration of each trial.

Our job is to extract that information and convert it into a format FSL can use — these are called *timing files* or *onset files*. For each condition, we'll split the trials by run and create four separate timing files:

- incongruent_run1.txt: Incongruent trials from Run 1
- incongruent_run2.txt: Incongruent trials from Run 2
- congruent_run1.txt: Congruent trials from Run 1
- congruent_run2.txt: Congruent trials from Run 2

Every timing file follows a standard structure with three columns:

1. Onset time (in seconds from scan start),
2. Trial duration (in seconds),
3. Parametric modulation value.

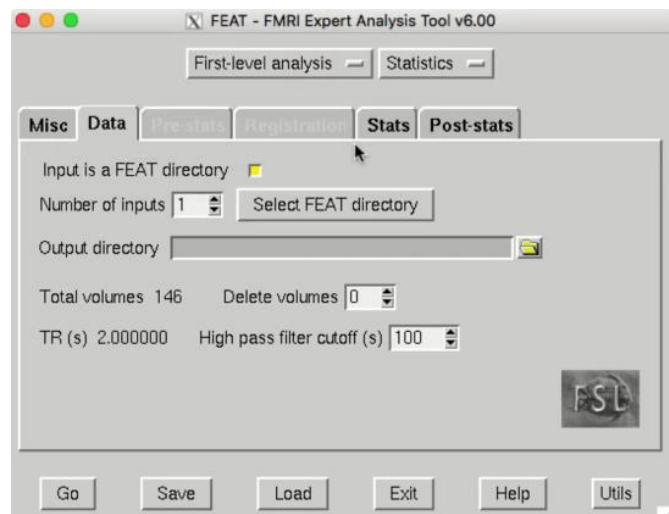


Once these files are set up, we can plug them into our model to fit the fMRI time-series. This gives us the foundation we need for further analysis and interpretation of brain responses under different experimental conditions.

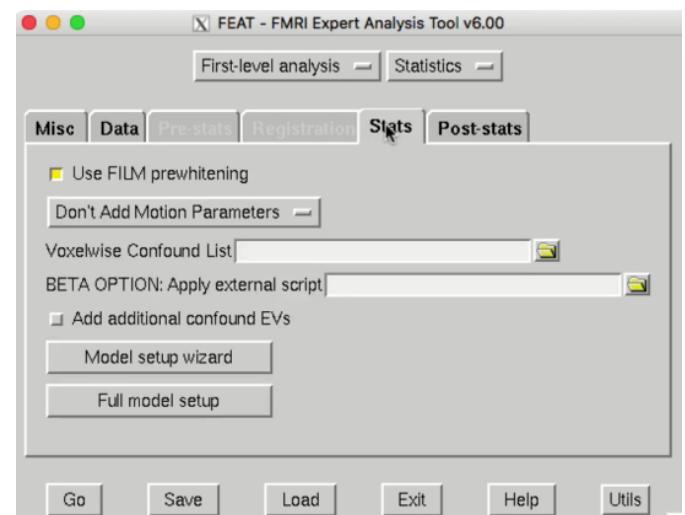
Running the First-Level Analysis:

- 1) To begin modeling the data, one first navigates to the appropriate subject directory (e.g., sub-08) and launches FSL by typing `fsl` from the command line.

This opens the FEAT GUI. Under the “Data” tab, the default analysis type is set to “Full Analysis”; changing this to “Statistics” allows for statistical modeling on preprocessed data.

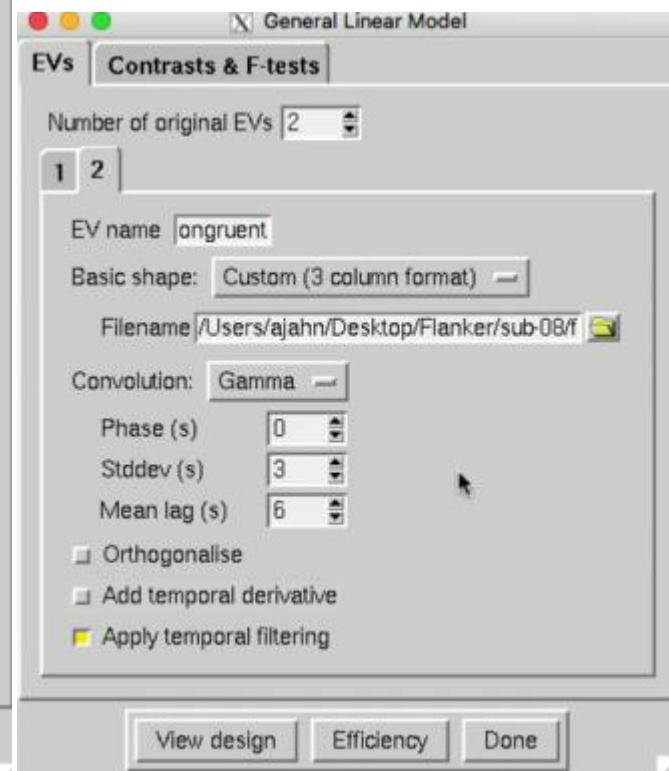
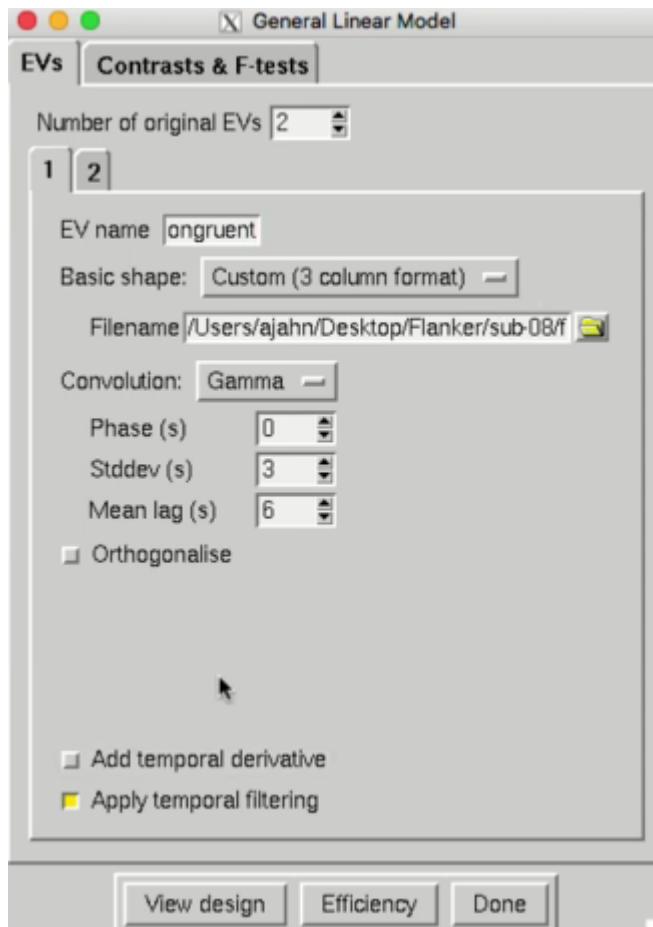


- 2) Next, a previously created FEAT directory — for example, `run1.feat` — is selected for further analysis. In the “Stats” tab, clicking on “Full model setup” brings up the interface for defining the model.



The number of original Explanatory Variables (EVs), which correspond to regressors in the General Linear Model, is set to 2 — one for each experimental condition.

For the first EV, the name is set to **incongruent**, and the “Basic shape” is changed to “Custom (3 column format)”. This option reveals a field for selecting a timing file, where the file `incongruent_run1.txt` is loaded. The option to add a temporal derivative is unchecked to keep the model focused on the main regressor. The second EV is then configured in the same way, labeled **congruent**, and linked to the timing file `congruent_run1.txt`.



3) With the model structure in place, attention turns to defining contrasts — comparisons of interest between the estimated beta weights of each condition. In this scenario, three contrasts are commonly specified:

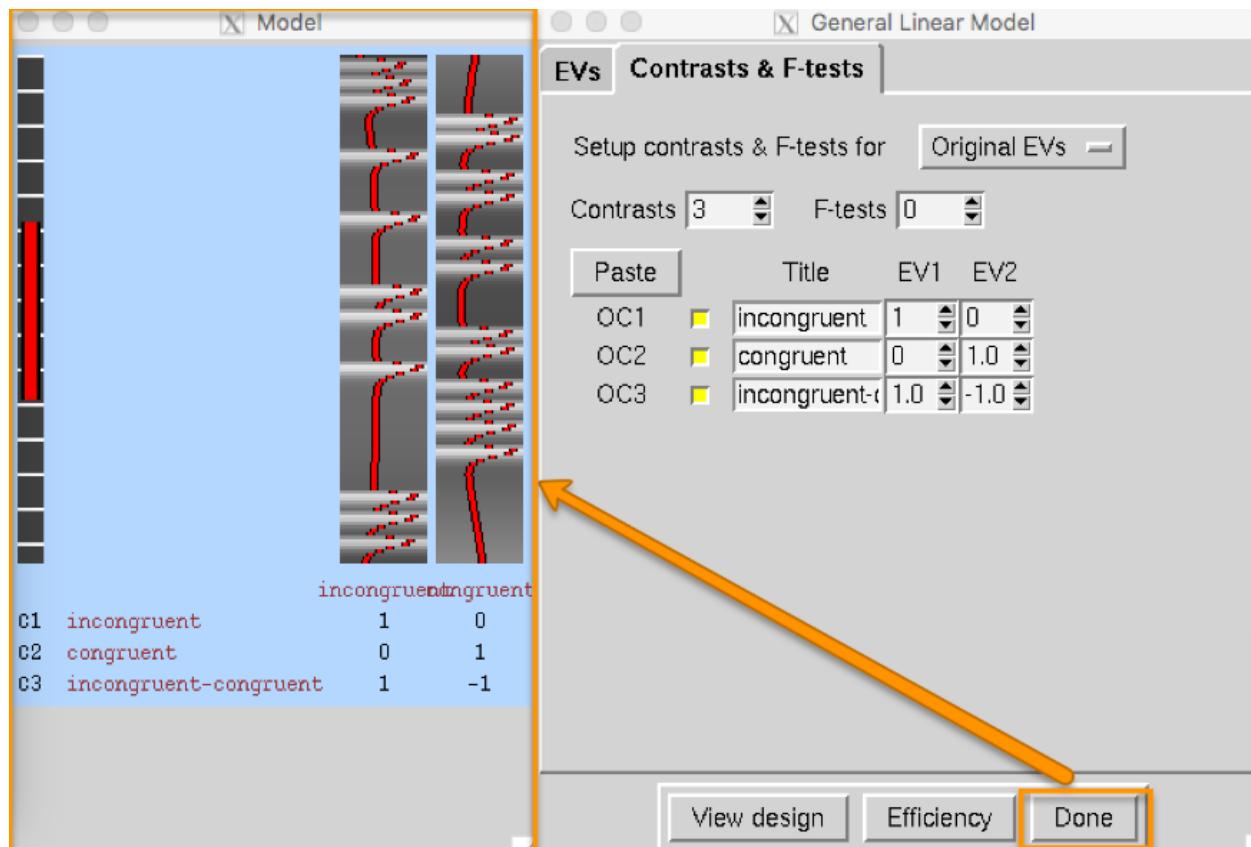
1. **Incongruent > Baseline** — capturing activation specific to the incongruent condition.
2. **Congruent > Baseline** — capturing activation for the congruent condition.
3. **Incongruent vs Congruent** — highlighting differences between the two conditions.

Each contrast is assigned a name and a weight vector:

- incongruent → [1 0]
- congruent → [0 1]
- incongruent-congruent → [1 -1]

	Title	EV1	EV2
OC1	incongruent	1	0
OC2	congruent	0	1.0
OC3	nt-congruent	1.0	0

After setting these contrasts, clicking “Done” opens the Design Matrix window. This visualization provides an overview of the modelled regressors and the applied filtering. The leftmost column represents the high-pass filter, which eliminates low-frequency drifts in the data. The next two columns correspond to the ideal time-series for the incongruent and congruent conditions, respectively. These columns represent the predicted BOLD responses based on the timing files.



This is a graphical representation of the design matrix and parameter contrasts.

The bar on the left is a representation of time, which starts at the top and points downwards. The white marks show the position of every 10th volume in time. The red bar shows the period of the longest temporal cycle which was passed by the highpass filtering.

The main top part shows the design matrix; time is represented on the vertical axis and each column is a different (real) explanatory variable (e.g., stimulus type). Both the red lines and the black-white images represent the same thing - the variation of the waveform in time.

Below this is shown the requested contrasts; each row is a different contrast vector and each column refers to the weighting of the relevant explanatory variable. Thus each row will result in a Z statistic image.

If F-tests have been specified, these appear to the right of the contrasts; each column is a different F-test, with the inclusion of particular contrasts depicted by filled squares instead of empty ones.

4) Lastly, the “Post-stats” tab includes options for thresholding the statistical maps. The “Z threshold” and “Cluster P threshold” fields control how stringent the statistical tests are — determining which voxels are considered significantly activated based on the defined contrasts.

The Ideal Time-Series and the GLM:

Finally, and after all steps have been completed, clicking Go allows FSL to begin the process of model fitting. This step generates a series of HTML pages that display progress and intermediate outputs. The process typically takes around 5–10 minutes. While the model runs, it's helpful to revisit how this setup maps onto the General Linear Model (GLM) framework used in fMRI analysis.

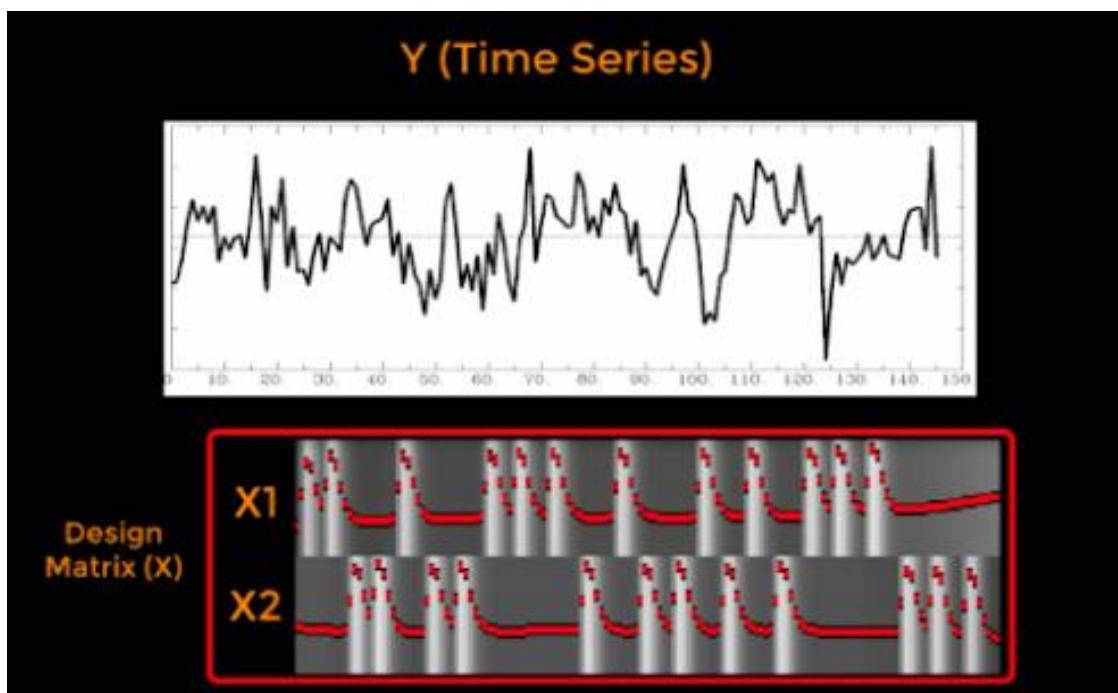
Each voxel in the brain has a corresponding BOLD signal time-series — this is the dependent variable in the GLM, denoted as Y . The predictors in the model, also known as regressors, are the convolved time-series generated from the condition-specific stimulus timings. These are referred to as x_1 and x_2 , and together they make up the design matrix, symbolized as X .

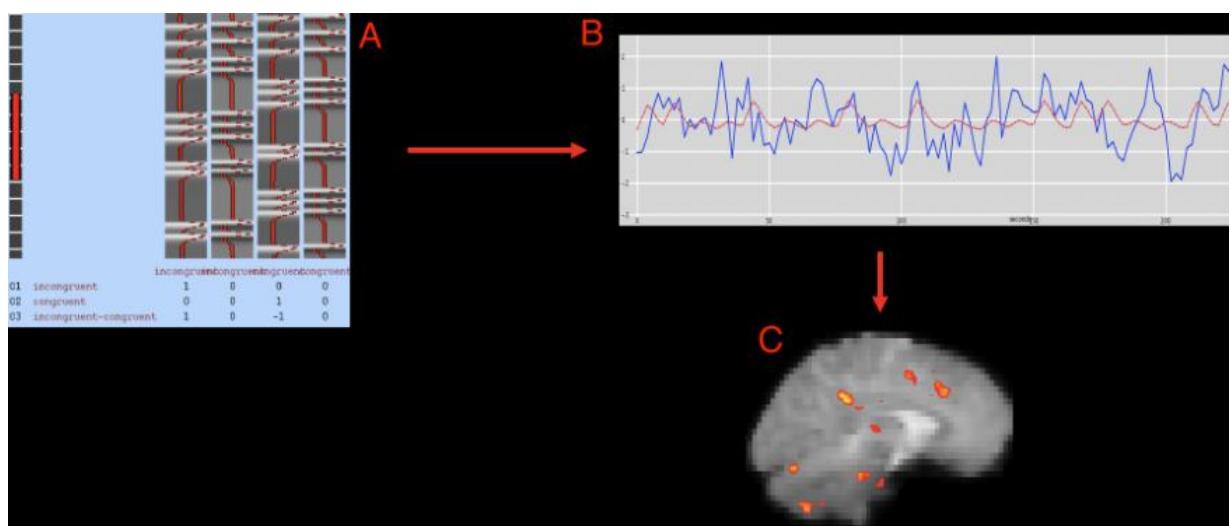
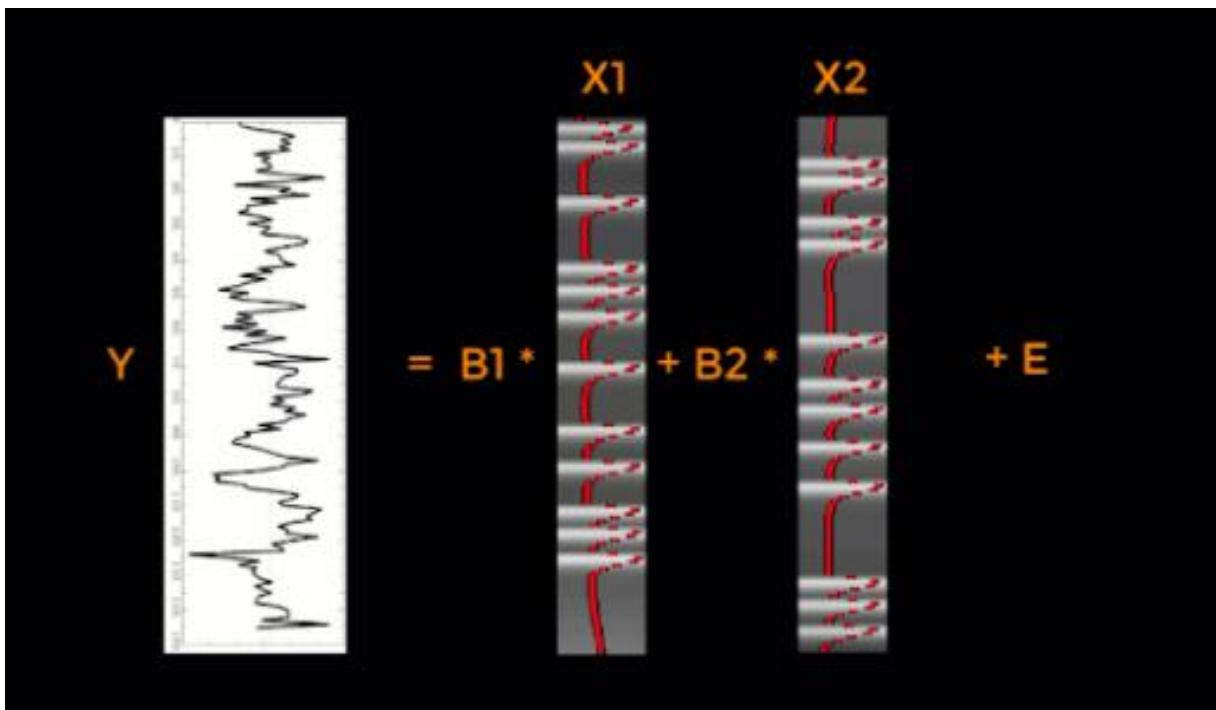
Importantly, fMRI design matrices are constructed using matrix algebra, and the data is organized so that time progresses from top to bottom rather than left to right. That is, the beginning of the scan is represented at the top of the matrix, and time flows downward.

The goal of the GLM is to estimate how much each regressor contributes to the observed signal in Y . This is done by calculating beta weights, represented as β_1 and β_2 . These values indicate how strongly the BOLD response for each condition matches the observed data. If, for example, a voxel responds more to incongruent stimuli, its corresponding β_1 will be higher.

The final component of the GLM is the residual term, E , which represents the error — the part of the data not explained by the model. A good model will minimize these residuals, indicating a closer match between the predicted and actual BOLD responses. When the residuals are small and the beta weights are large enough relative to the noise, the result is more likely to be statistically significant.

This structure — $Y = X\beta + E$ — serves as the mathematical backbone of the statistical analysis in fMRI, and the visualization tools in FSL help in interpreting how well this model fits the data.





- **Note on FSL Terminology:**

In FSL, **beta weights** are referred to as **parameter estimates**, or **PEs**, while **contrasts between beta weights** are termed **contrast of parameter estimates**, or **COPEs**. When examining the output from a model run, there will typically be two PE files — one for each condition — and three COPE files, corresponding to the three contrasts defined earlier.

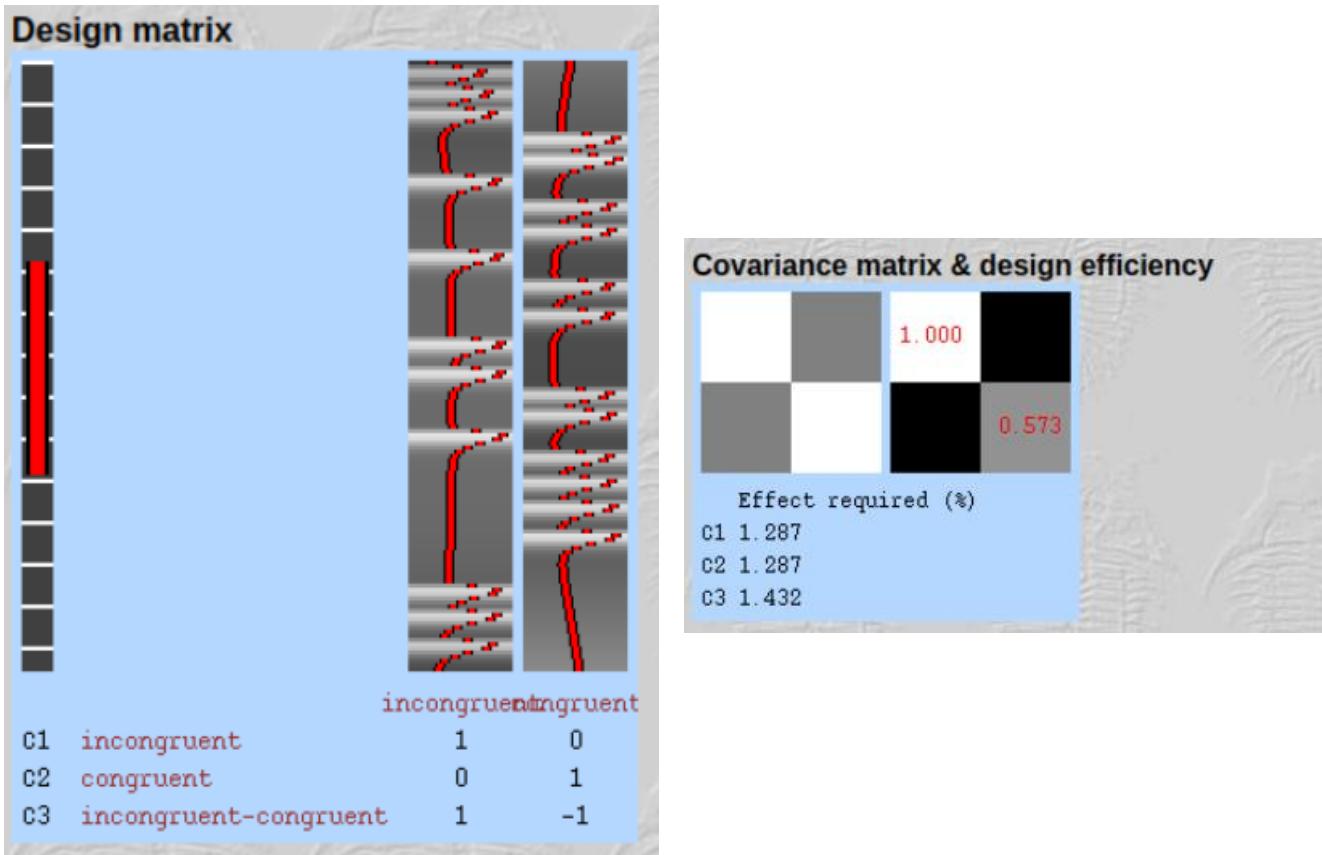
Although the first two contrasts (Incongruent and Congruent vs. baseline) will be numerically identical to the parameter estimates themselves, they are still included as separate COPEs. This may initially appear redundant, but it becomes necessary in later stages of analysis. Specifically, FSL requires these COPE-labeled files for **higher-level (group) analyses**, where comparisons are made across subjects or sessions. Maintaining this distinction ensures compatibility with FSL's pipeline and allows for consistent contrast mapping across different levels of analysis.

Location		Voxel location				
History Location	Coordinates: Scanner anatomical					
	-0.0001487732	-	+	32	-	+
	-1.445934	-	+	31	-	+
	-20.24117	-	+	19	-	+
Volume				0	-	+

pe1
[32 31 19]: 63.72538757324219
pe2
[32 31 19]: -26.283952713012695
cope1
[32 31 19]: 63.72538757324219
cope2
[32 31 19]: -26.283952713012695
cope3
[32 31 19]: 90.00934600830078

Now let's analyze the outputs of the 1st level analysis for both runs of each of sub-08 and sub-22!

Sub-08: Run 1:

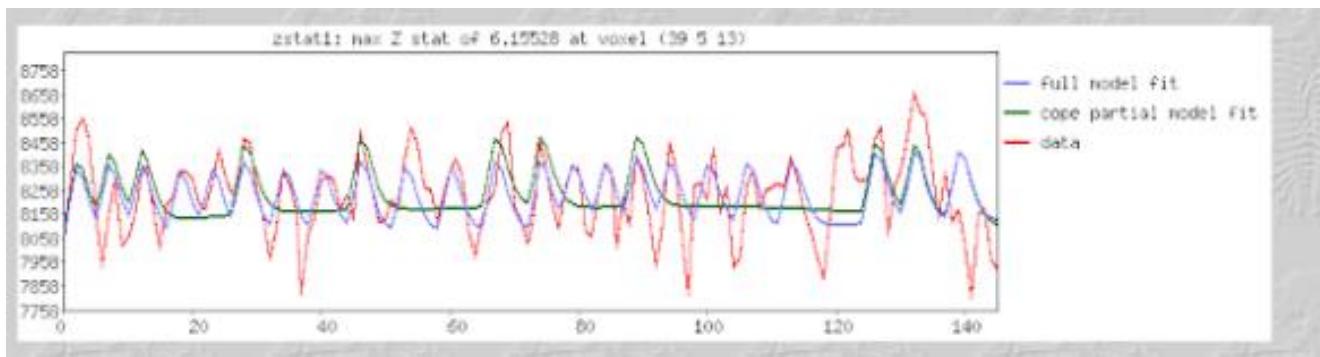


As we mentioned before, the design matrix describes our regressors (incongruent and congruent). Next we have the covariance matrix and design efficiency, which assess the quality of the GLM design. The covariance matrix indicates how much the regressors (conditions) overlap — high off-diagonal values (like 0.573) suggest some correlation, which can reduce the ability to distinguish effects. Below that, the "Effect required (%)" values reflect how strong a neural signal change must be for each condition or contrast to be reliably detected; lower values indicate a more efficient and sensitive design. In this case, C1 and C2 are equally efficient (1.287%), while the contrast C3 (incongruent – congruent) requires a slightly stronger signal (1.432%) to be detected.

- **Z-Stat 1: Incongruent Condition:**

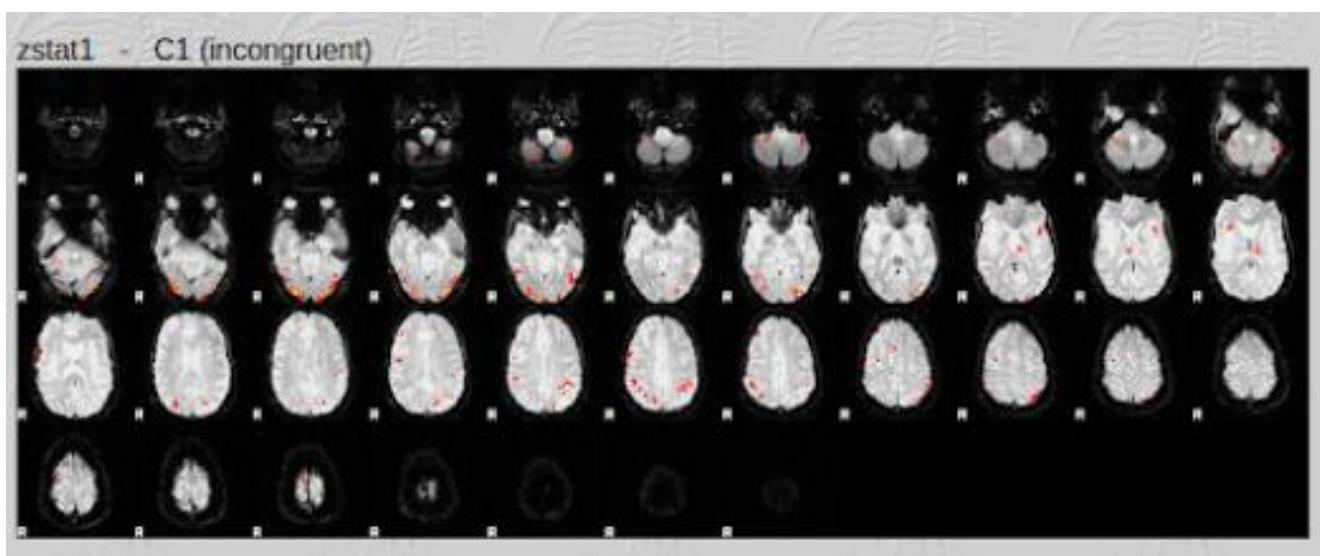
Time series plot:

The time series plot shows the BOLD signal of one voxel overlain on the GLM fitted model for the incongruent condition. This voxel is the peak voxel, or the one most activated during the incongruent task and has the closest similarity with the corresponding model.



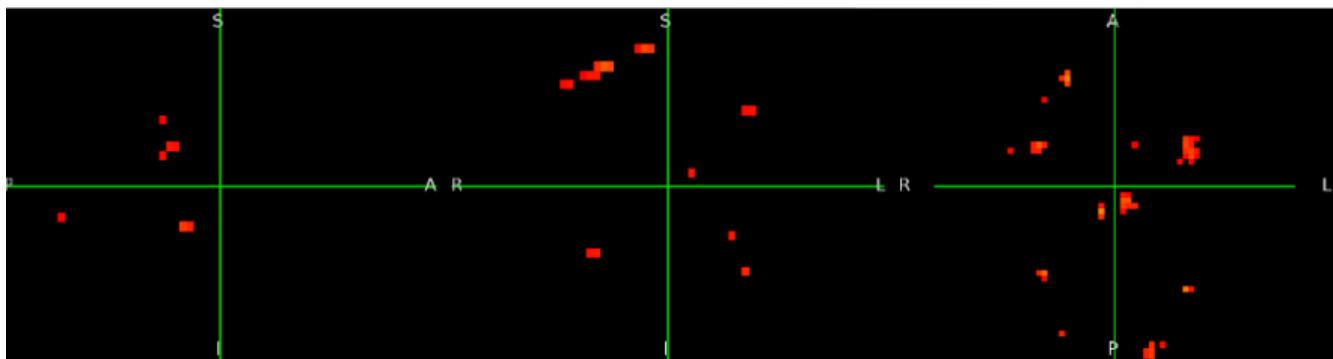
Threshold activations images:

Thresholded activation images represent statistical maps across the entire brain volume, highlighting where task-related activation is significant – in our case here the incongruent task. They display statistical results overlaid on anatomical slices of the subject's brain. Specifically, only voxels with $p < 0.05$ are shown.

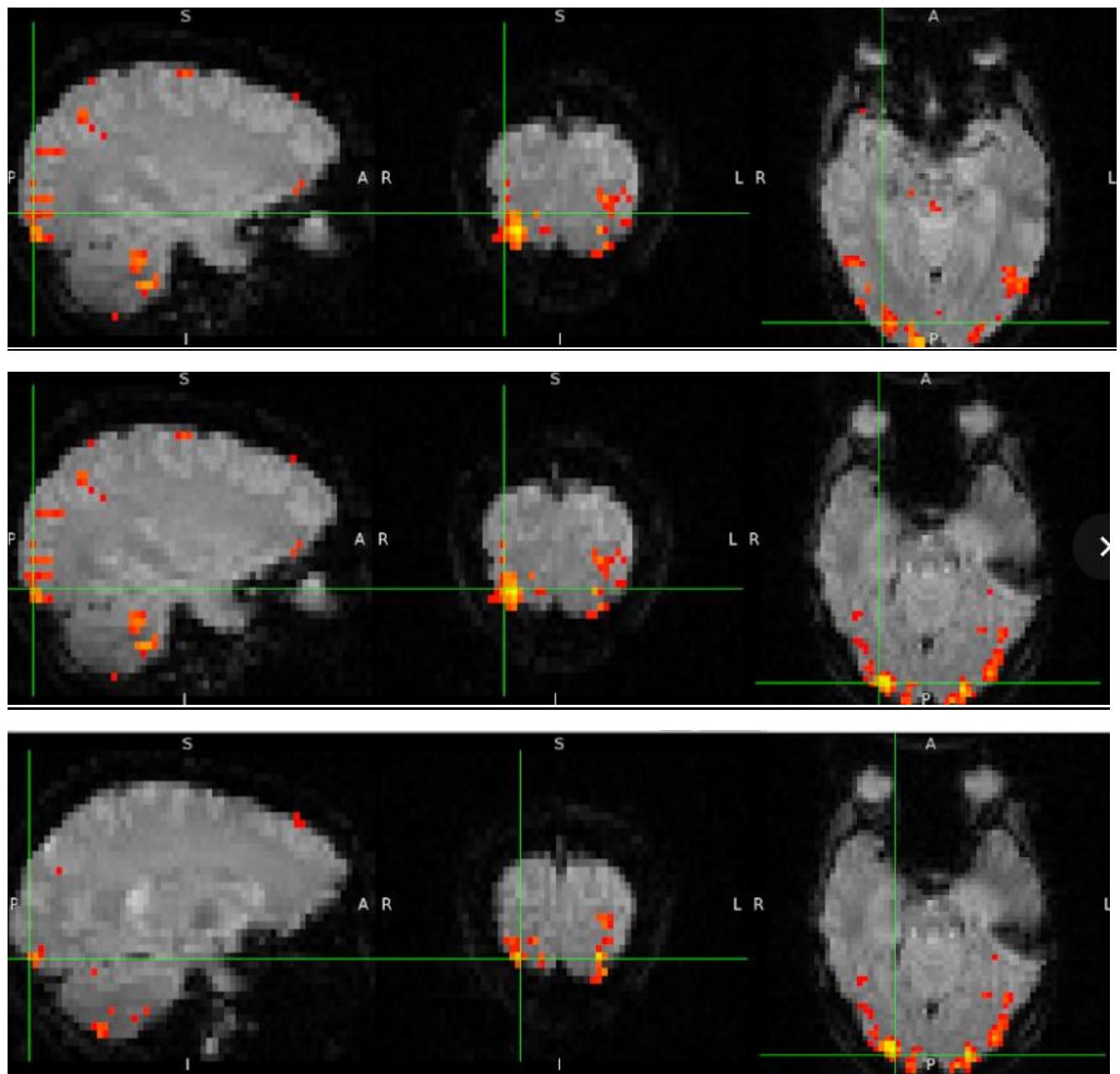


Visualization on FSL:

Opening Zstat1 file shows us highlighted locations of the active voxels

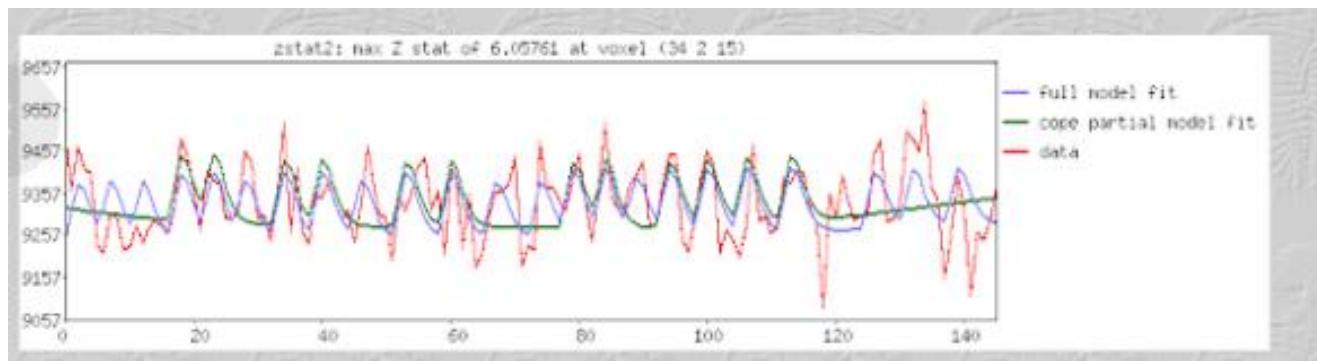


It's more logical to overlay them on the functional brain images to help clearly distinguish which areas the voxels are in.

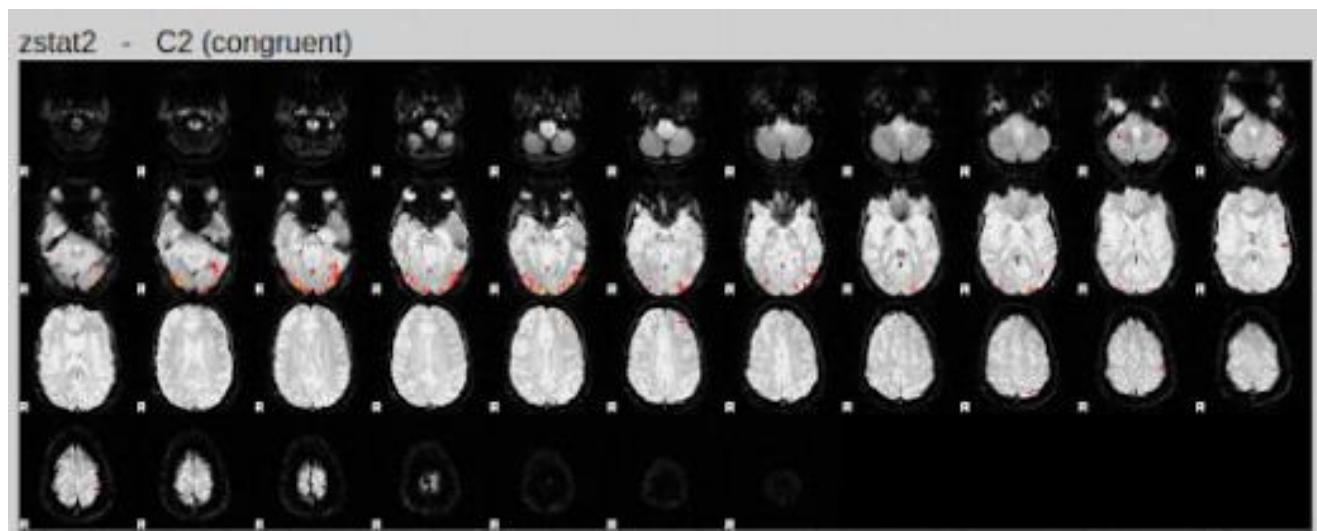


- **Z-Stat 2: Congruent Condition:**

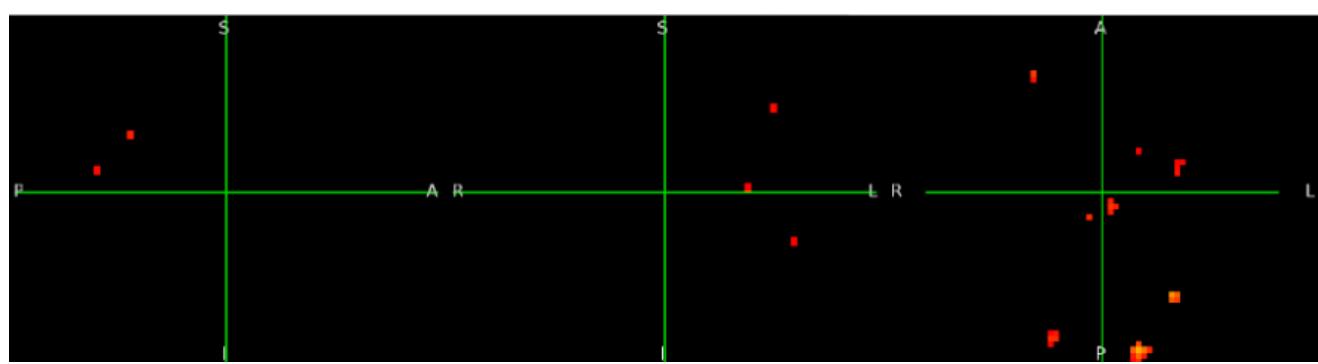
Time series plot:

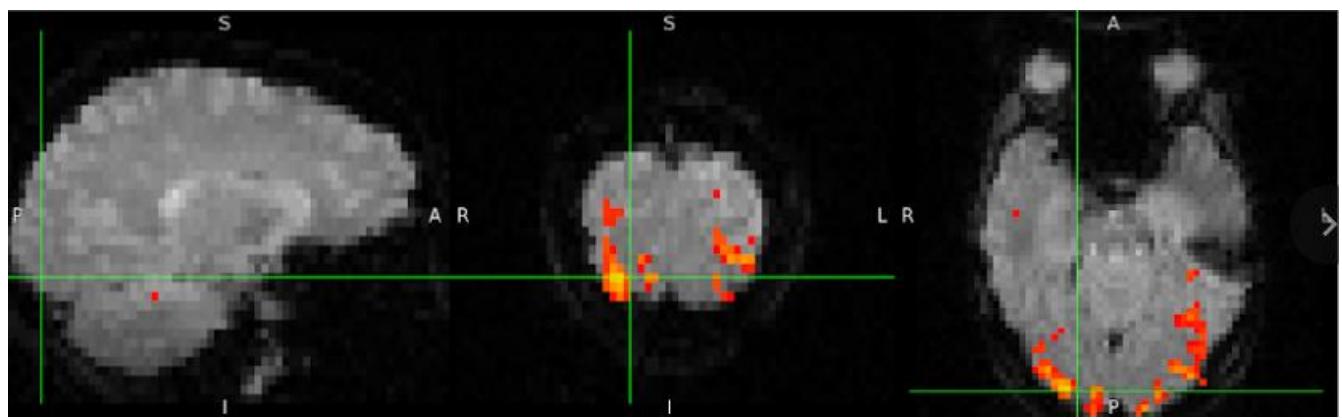
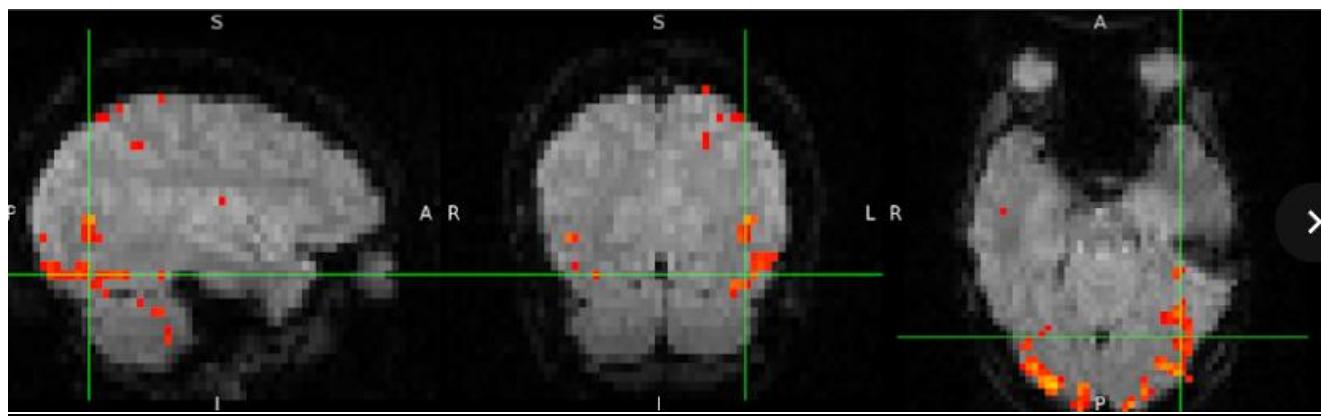
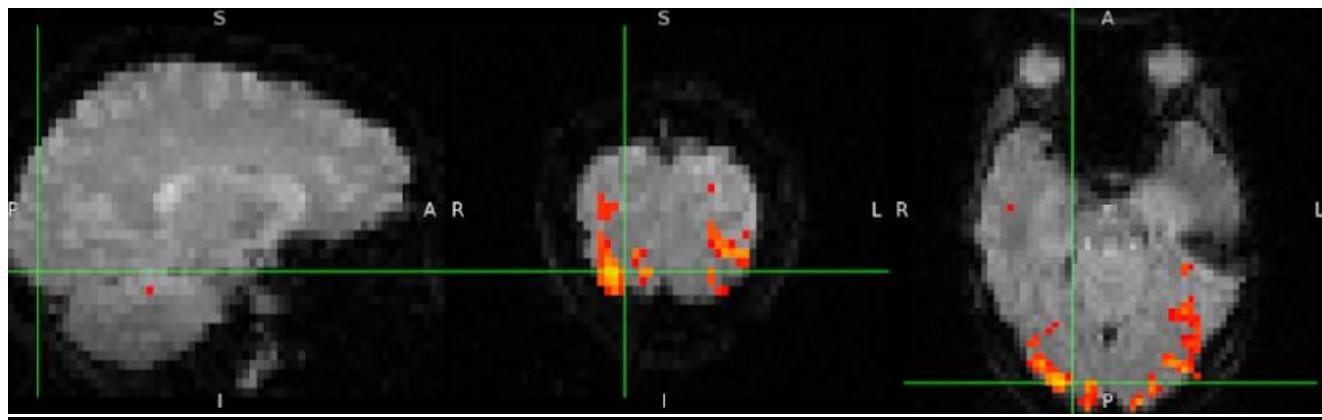


Threshold activations images:



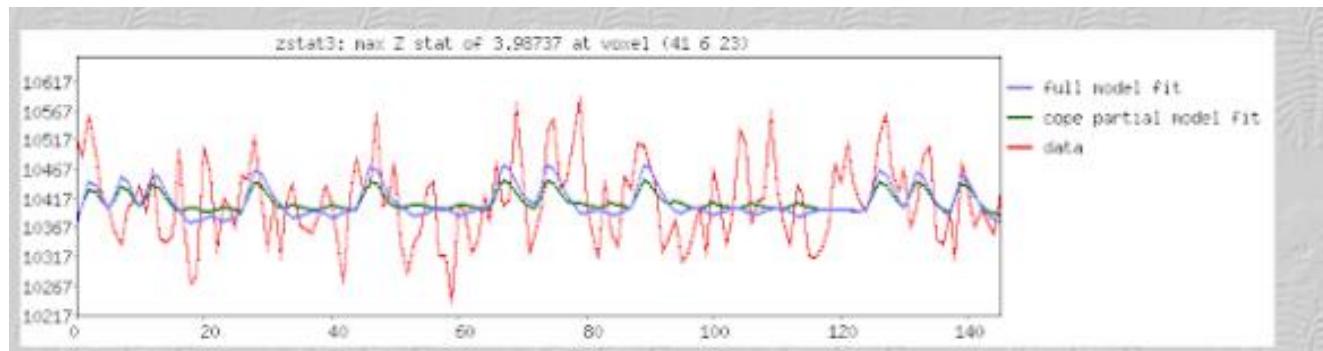
Visualization on FSL:



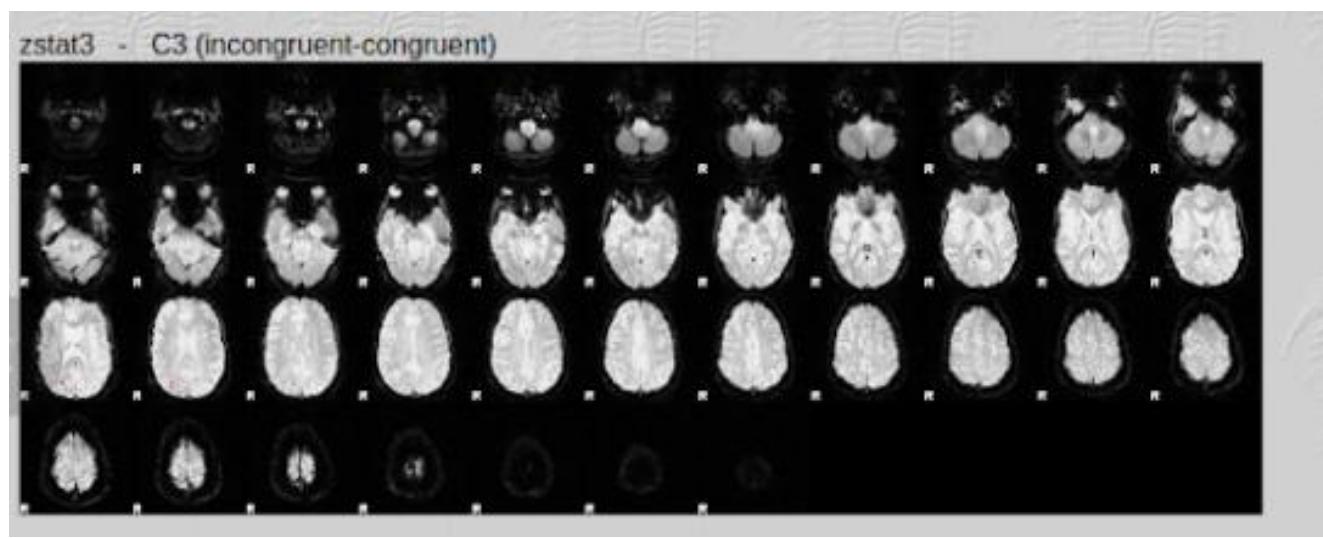


- **Z-Stat 3: Incongruent-Congruent Condition:**

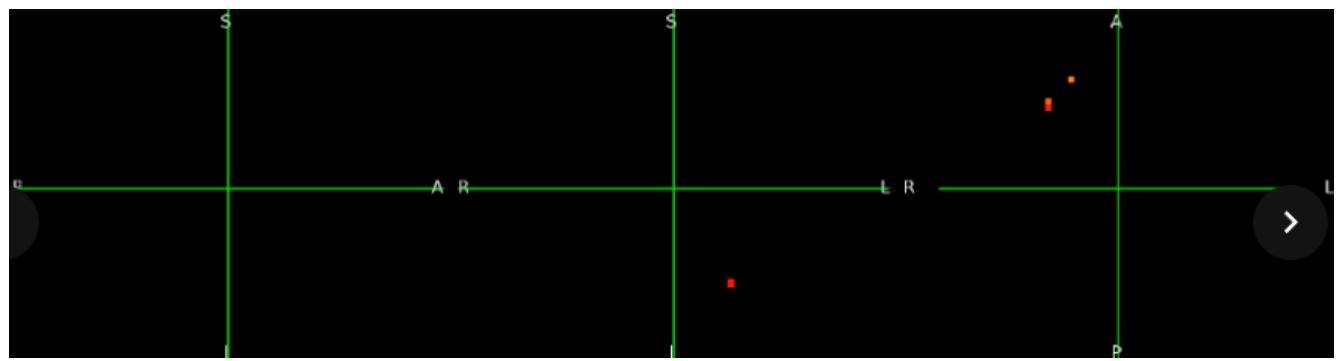
Time series plot:

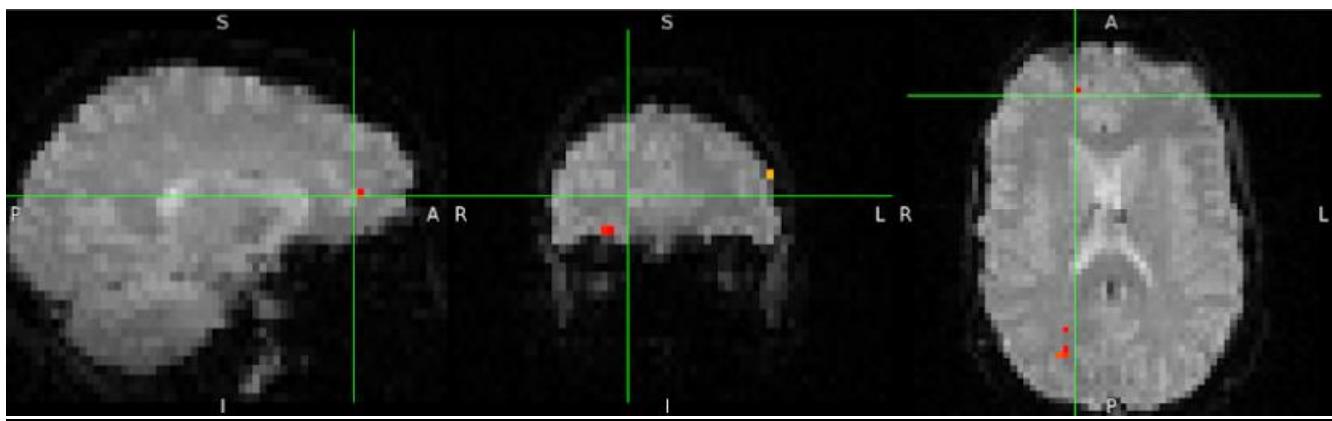
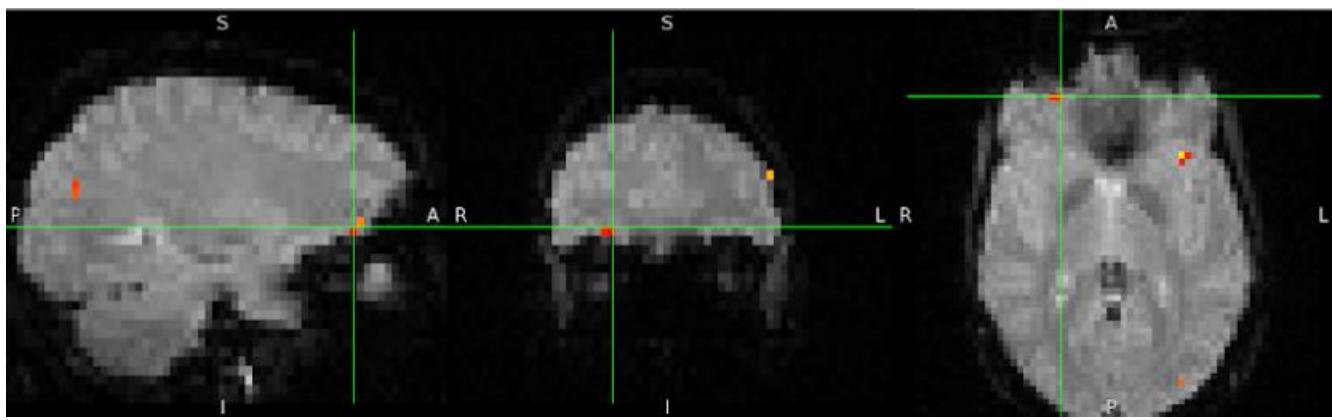


Threshold activations images:

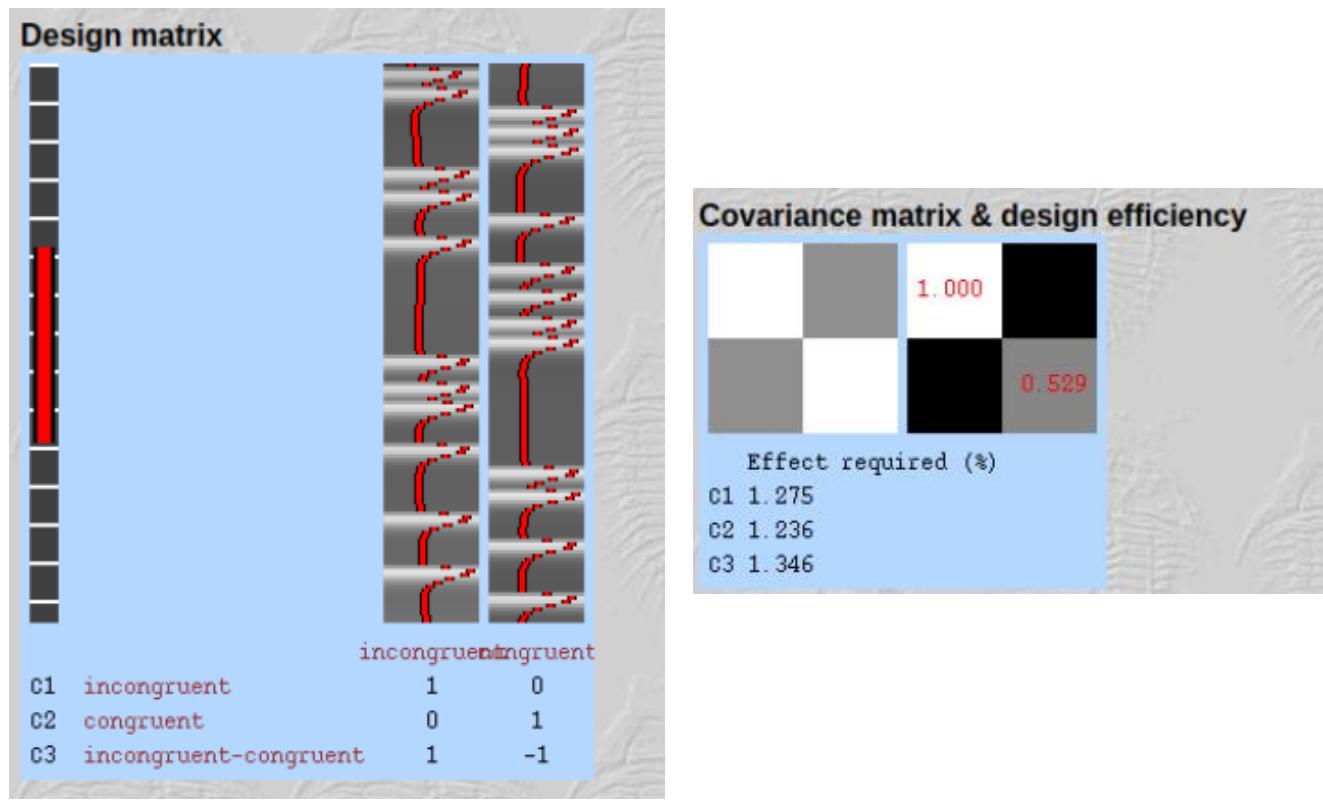


Visualization on FSL:



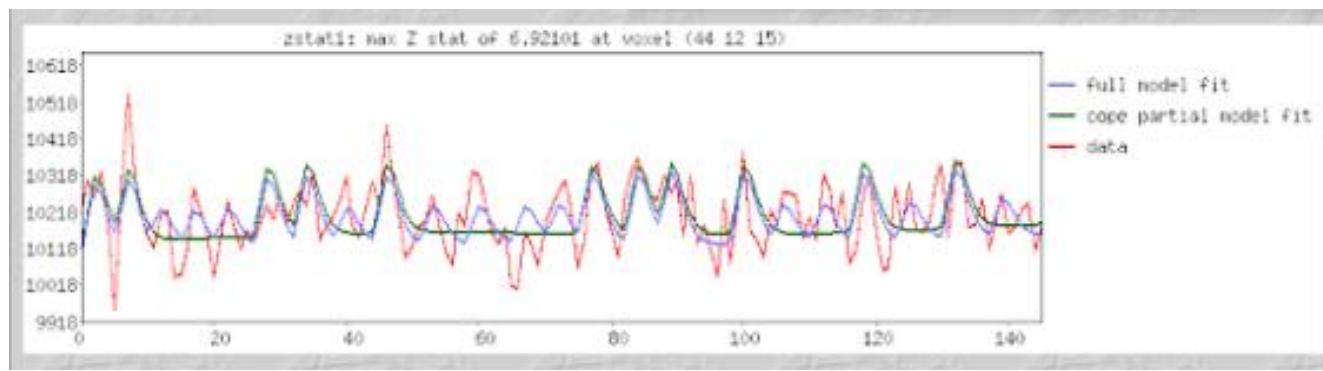


Sub-08: Run 2:

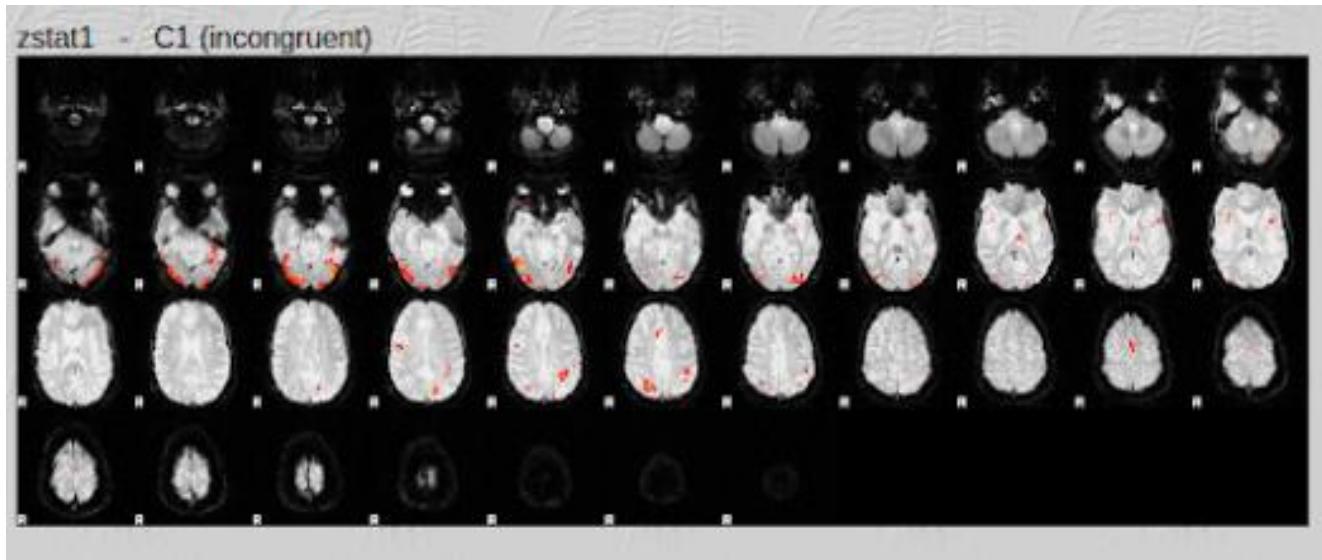


- Z-Stat 1: Incongruent Condition:

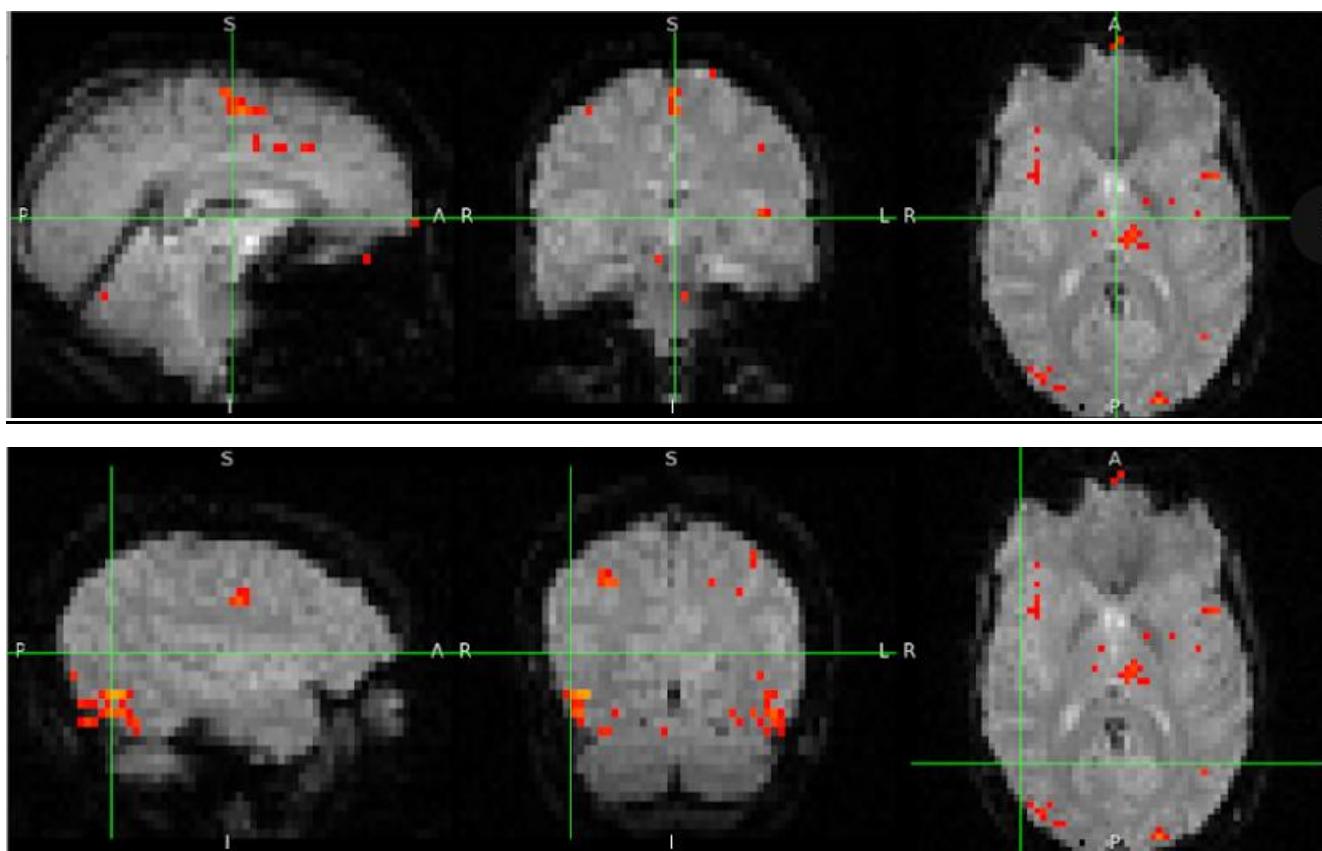
Time series plot:

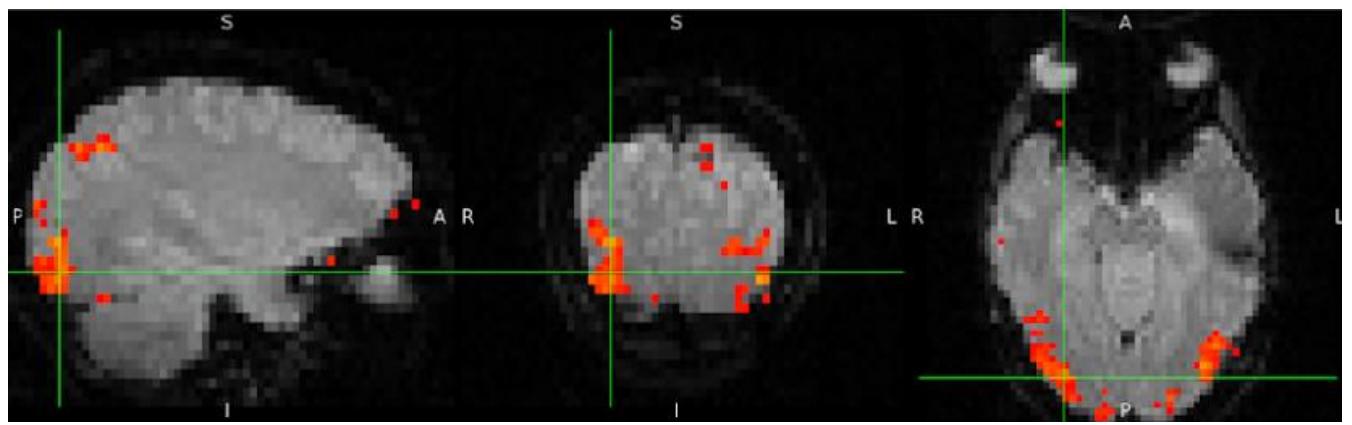
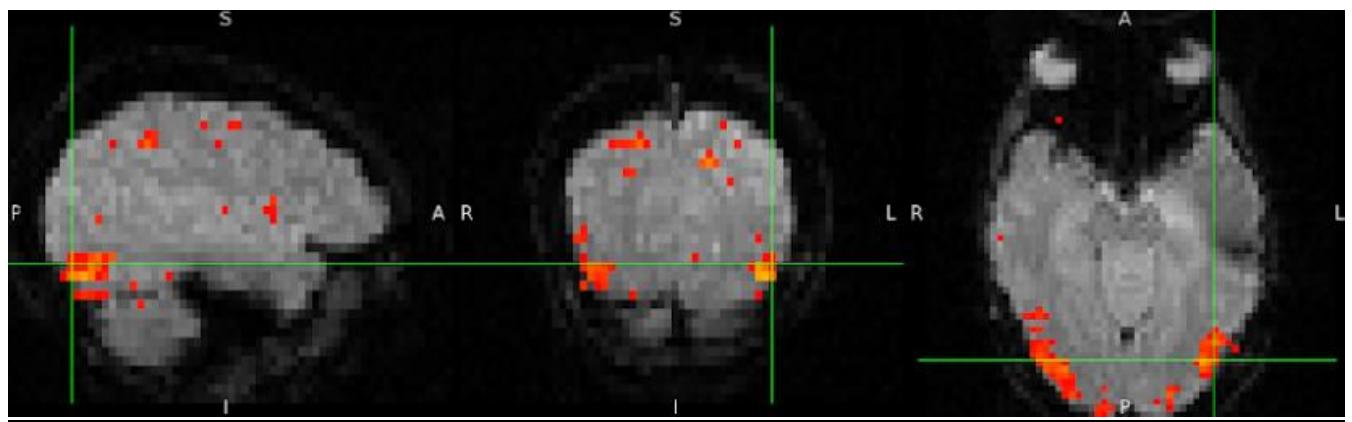


Threshold activations images:



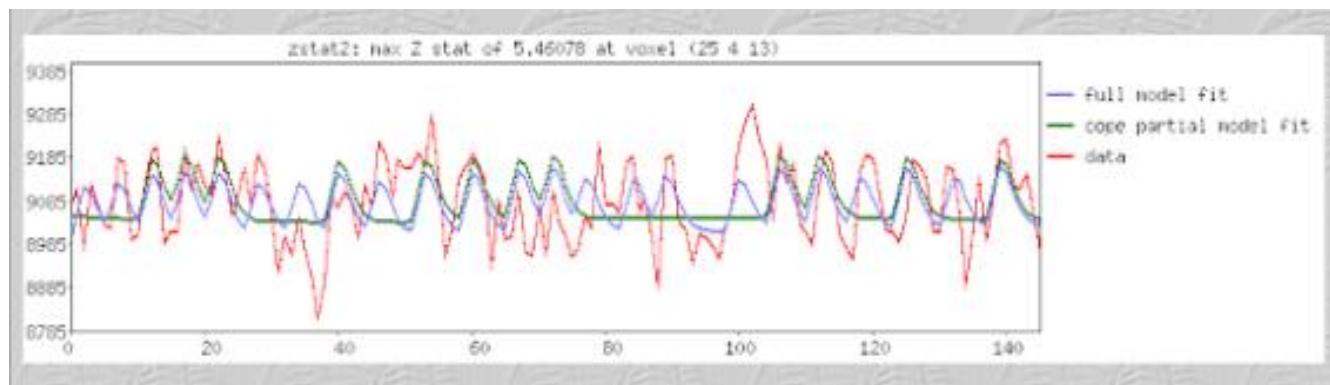
Visualization on FSL:



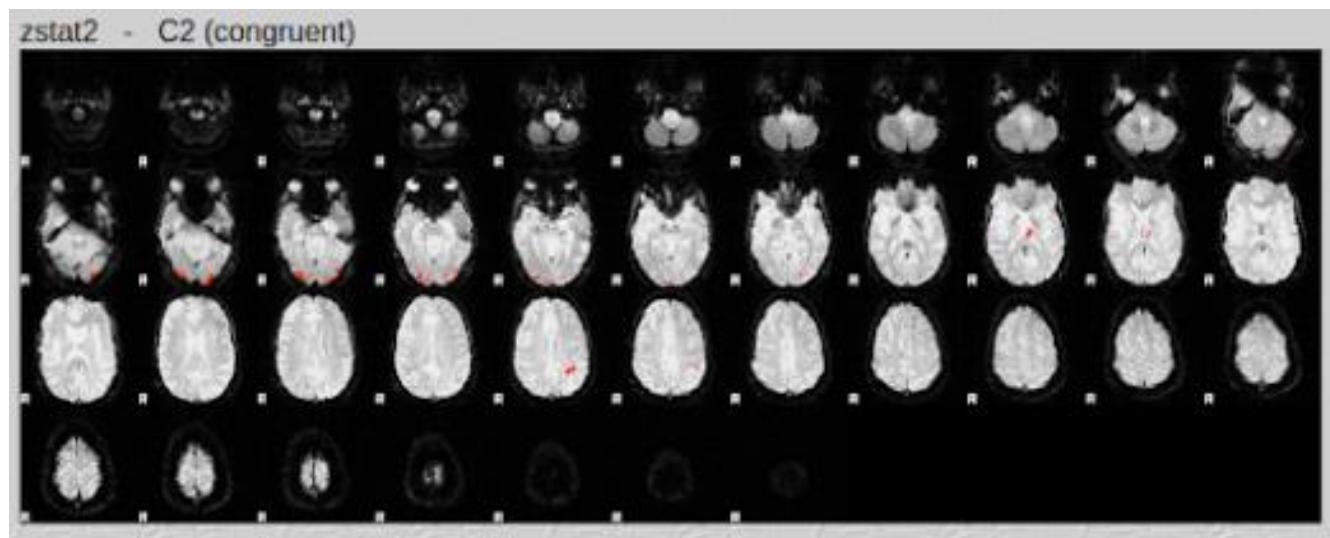


- **Z-Stat 2: Congruent Condition:**

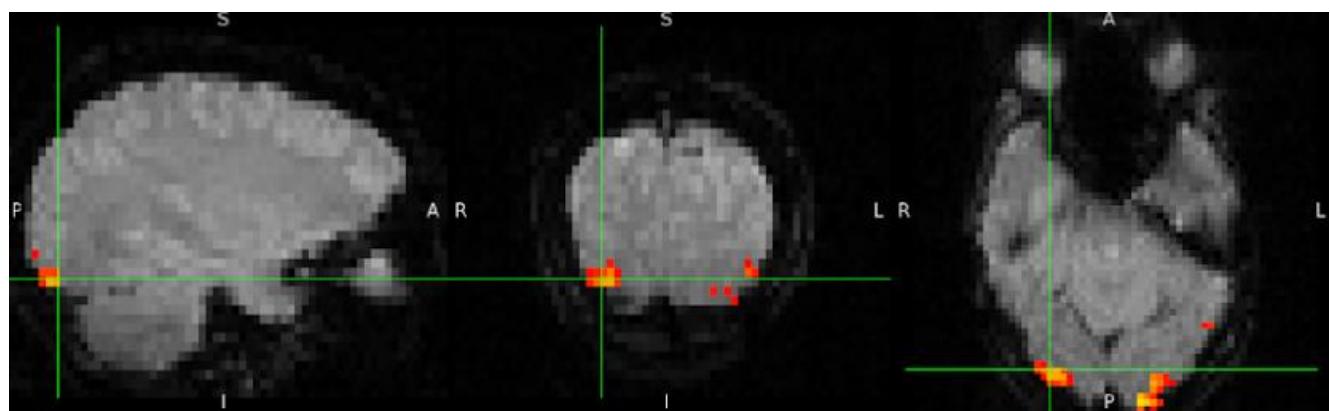
Time series plot:

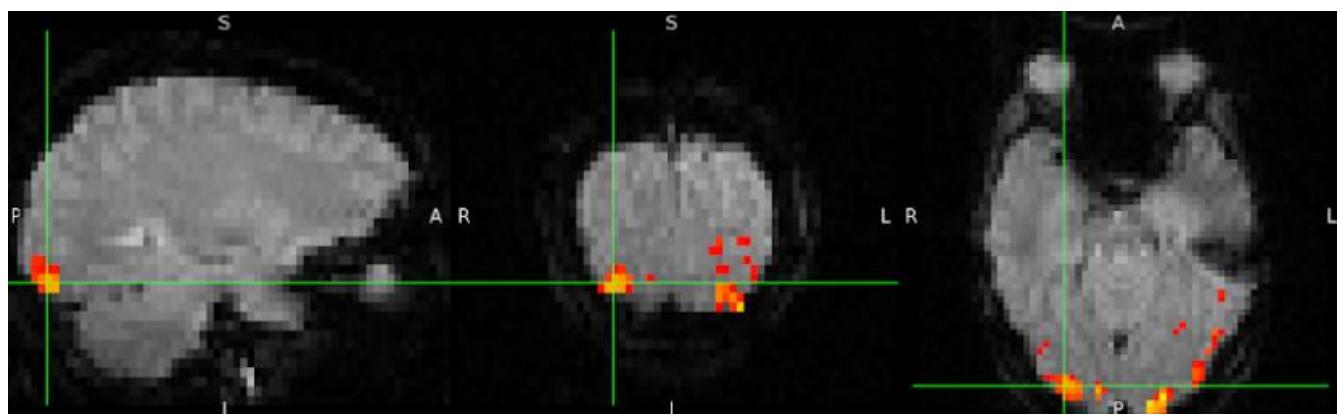
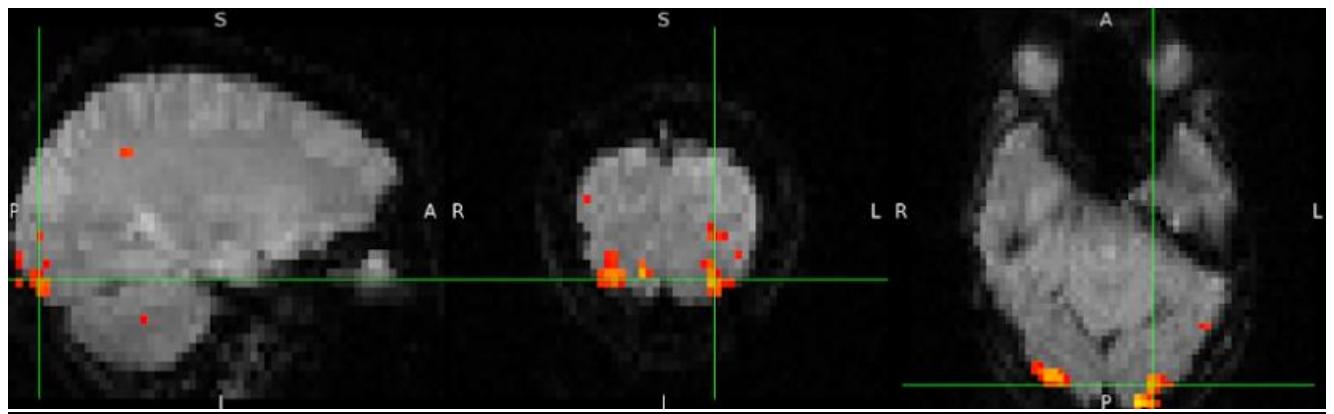


Threshold activations images:



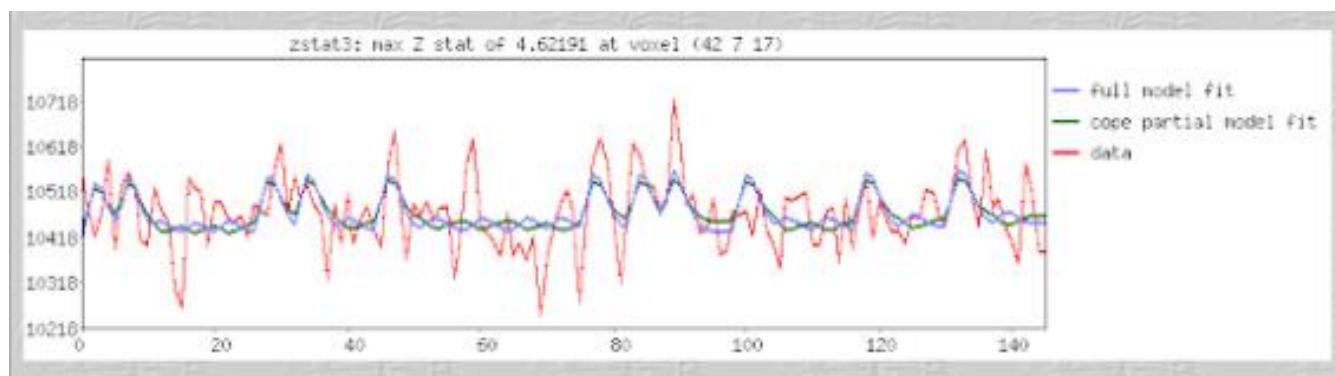
Visualization on FSL:



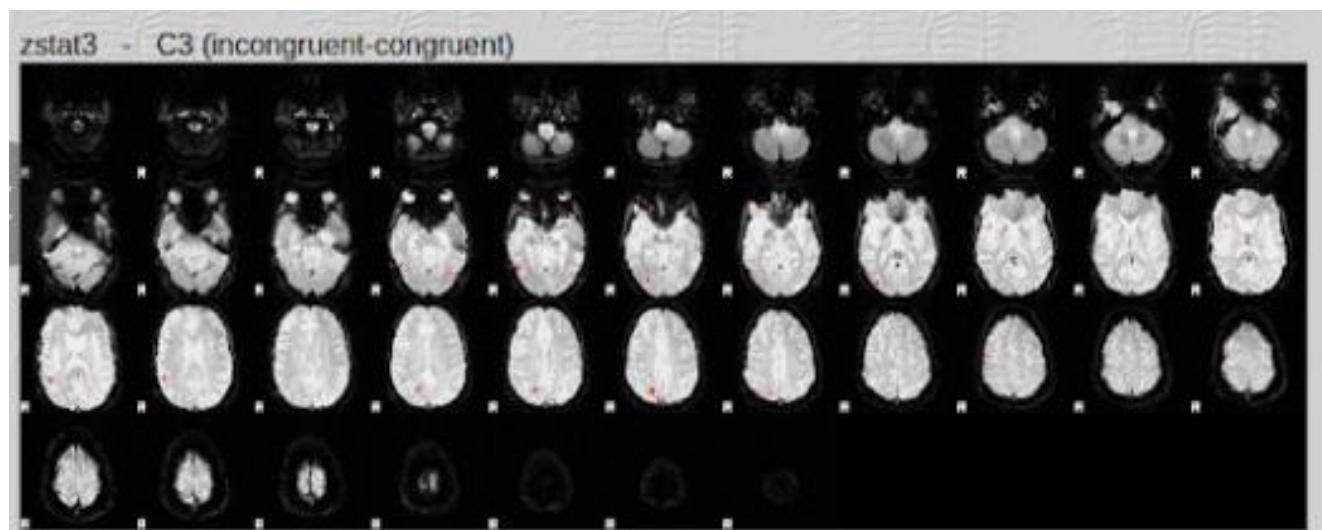


- **Z-Stat 3: Incongruent-Congruent Condition:**

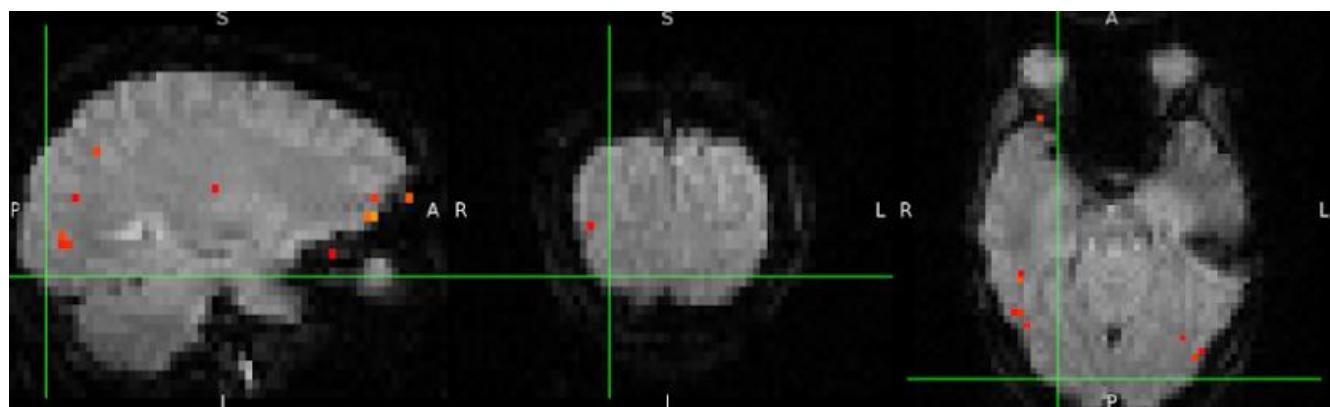
Time series plot:

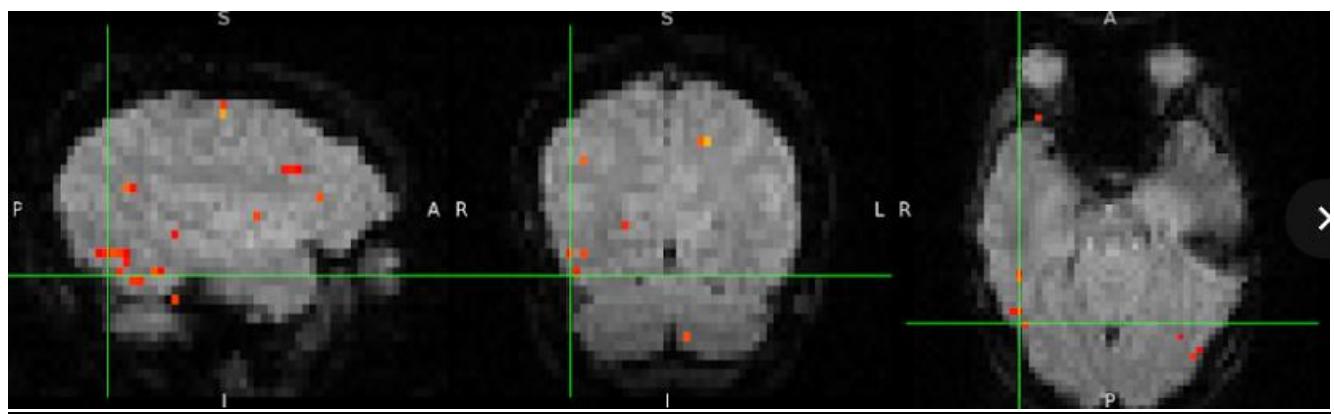
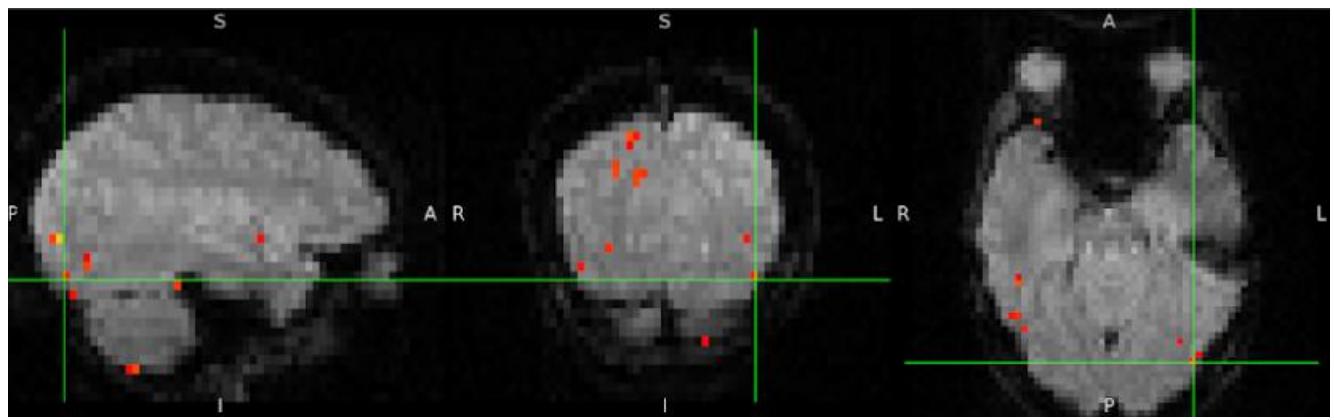


Threshold activations images:

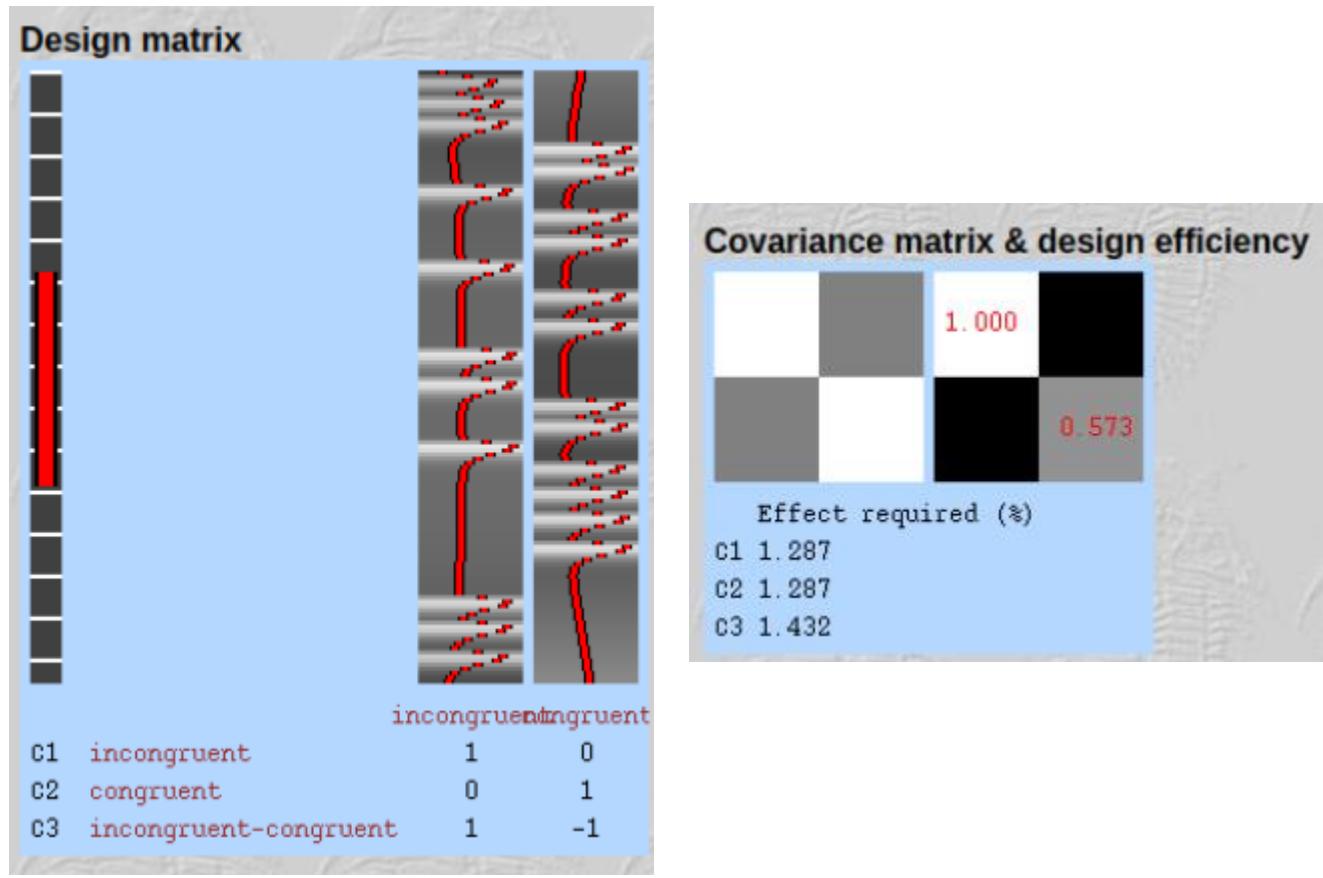


Visualization on FSL:



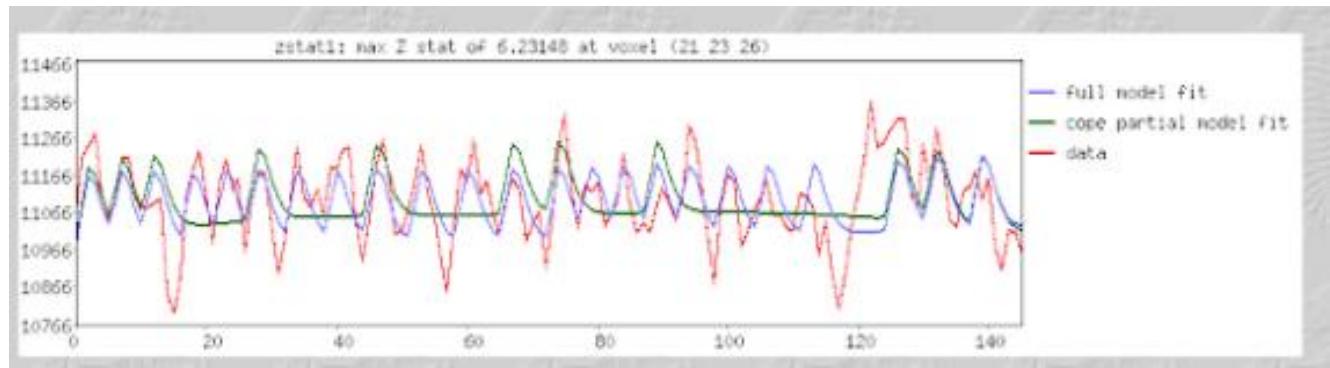


Sub-22: Run 1:

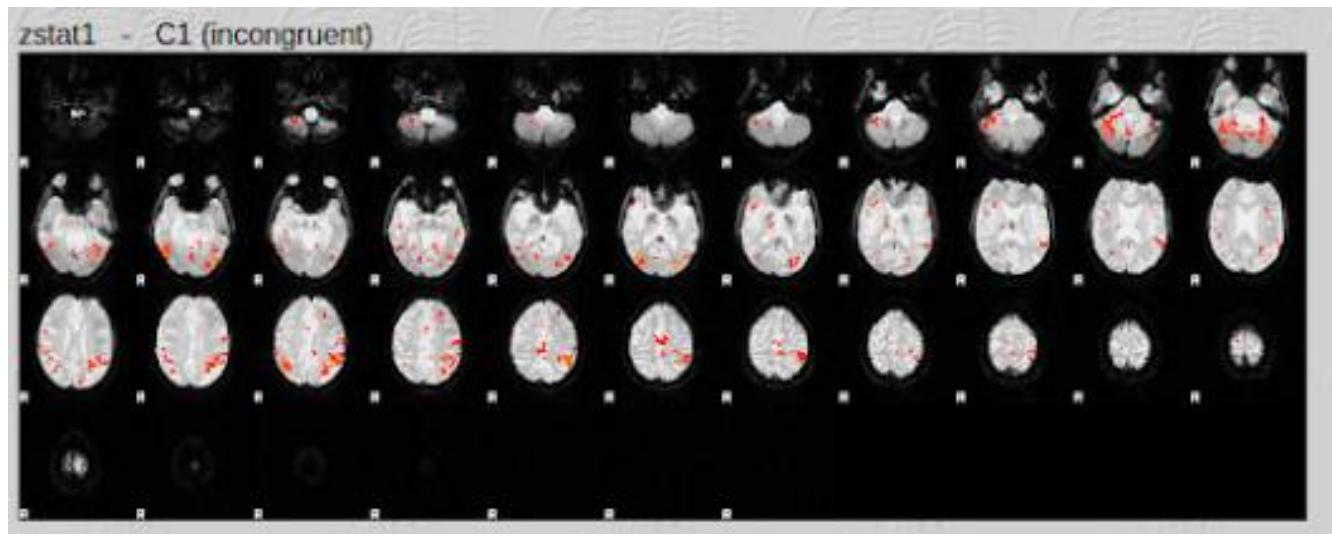


- **Z-Stat 1: Incongruent Condition:**

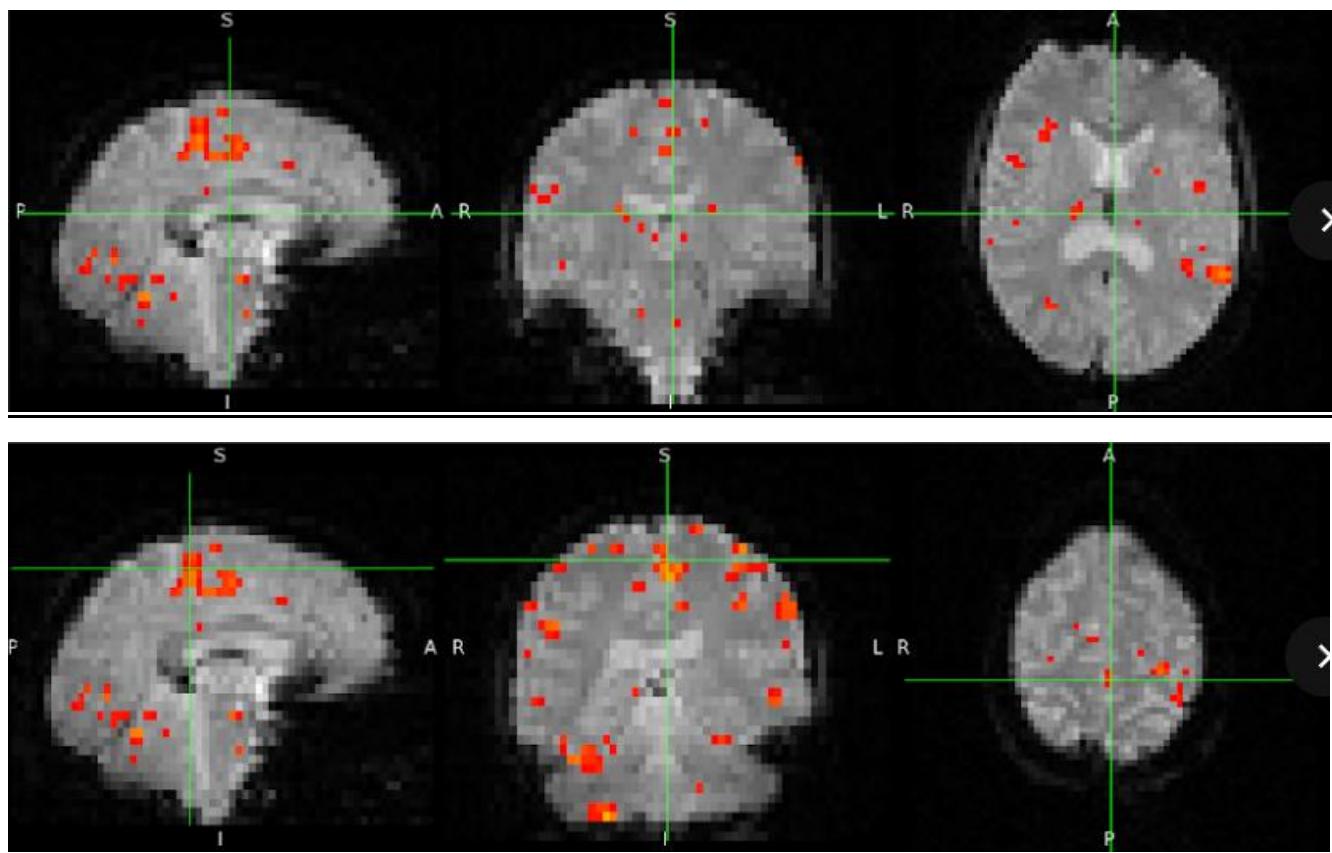
Time series plot:

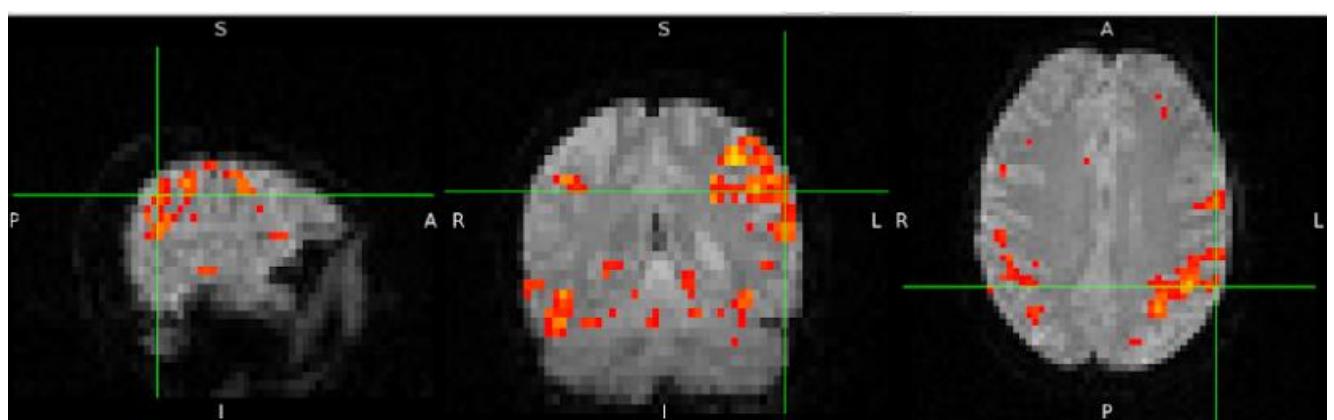
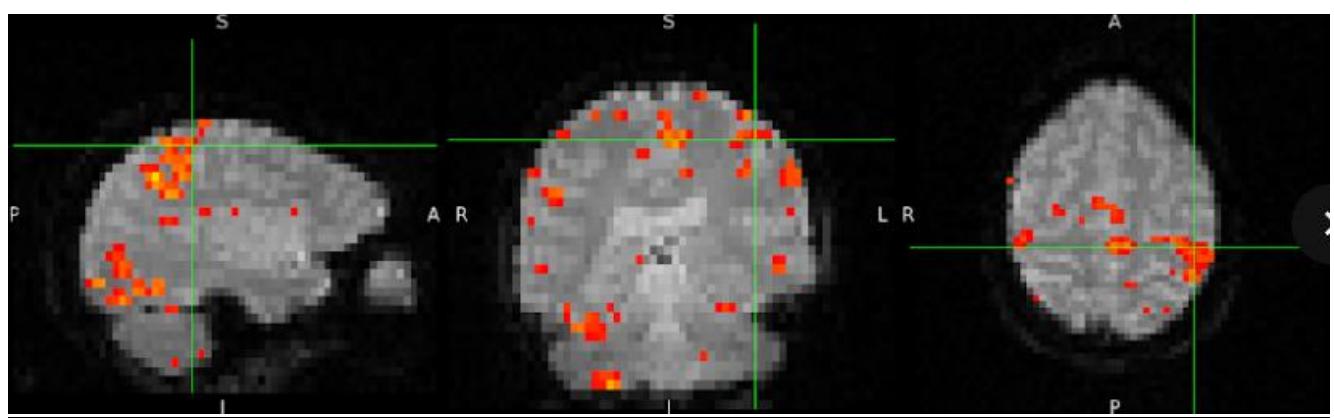
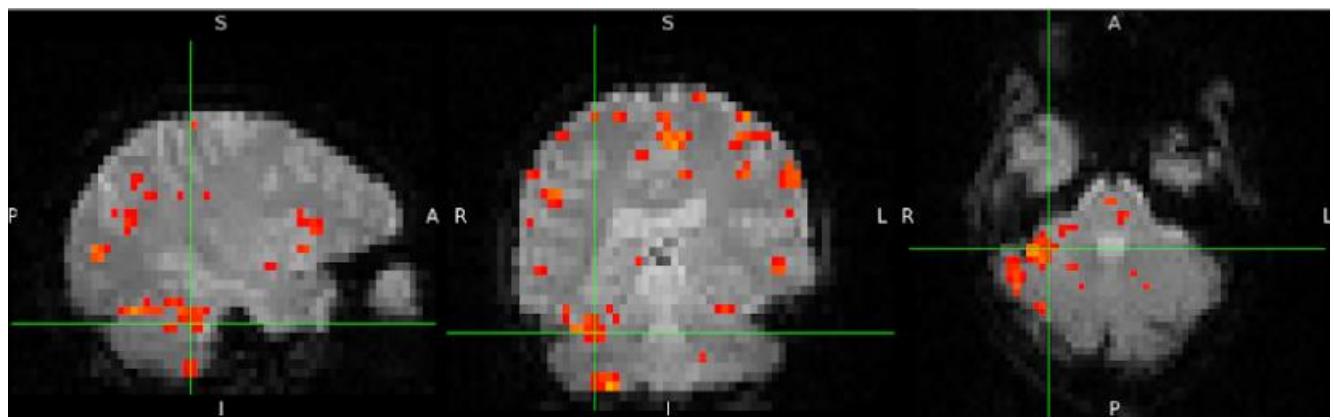


Threshold activations images:



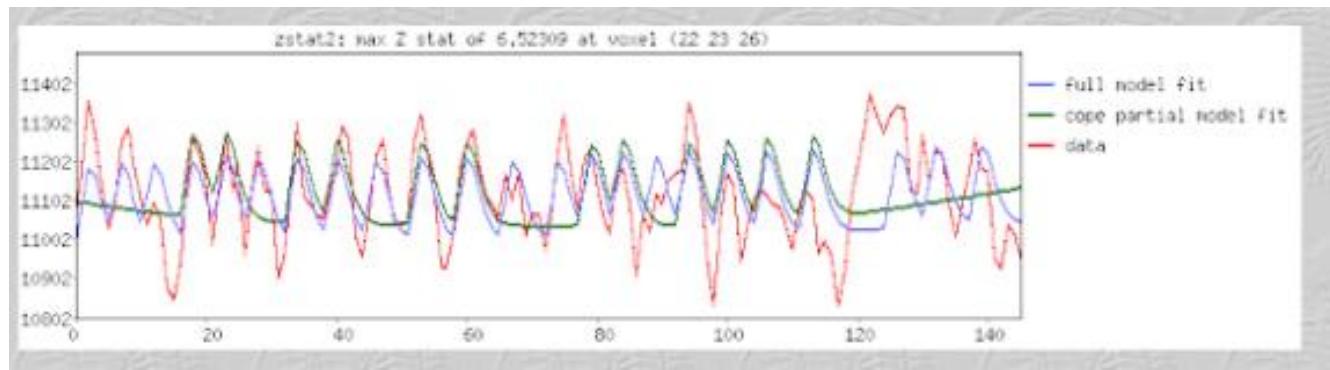
Visualization on FSL:



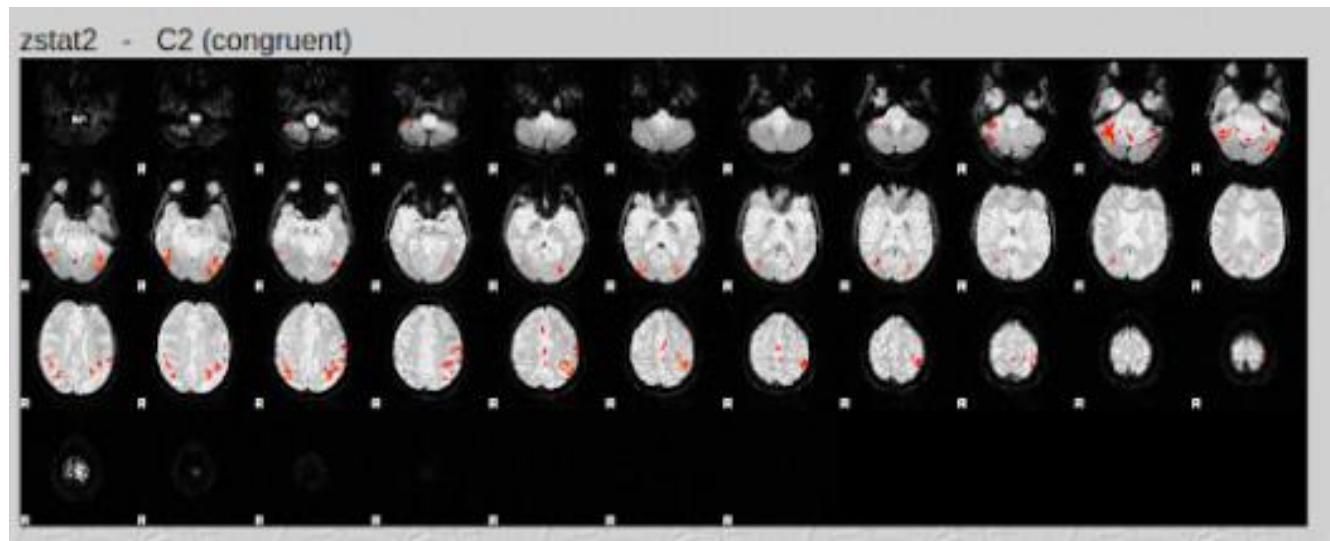


- **Z-Stat 2: Congruent Condition:**

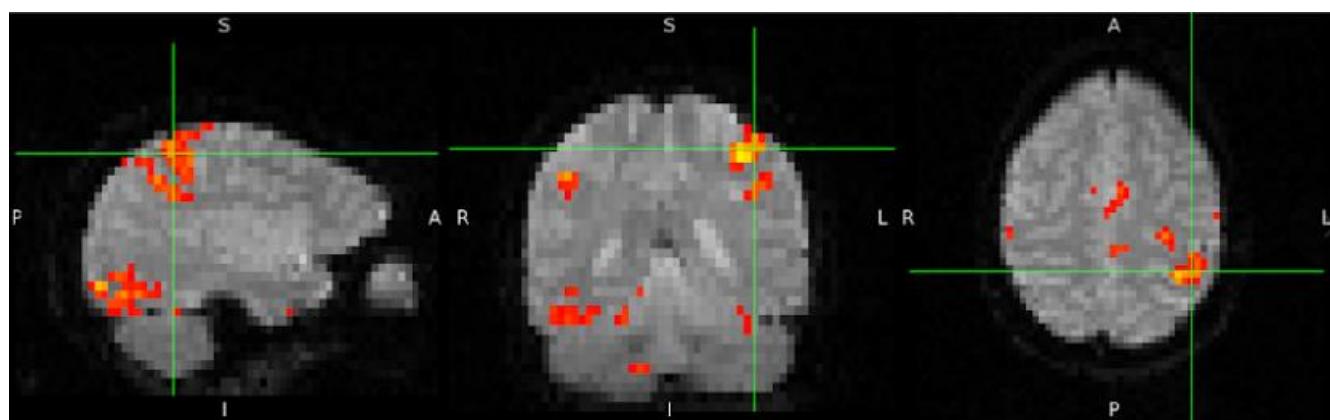
Time series plot:

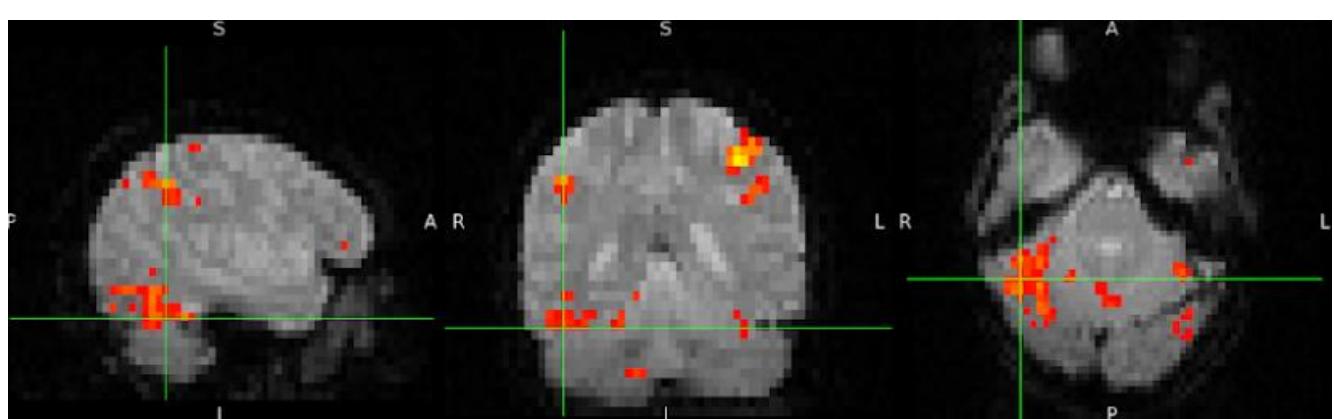
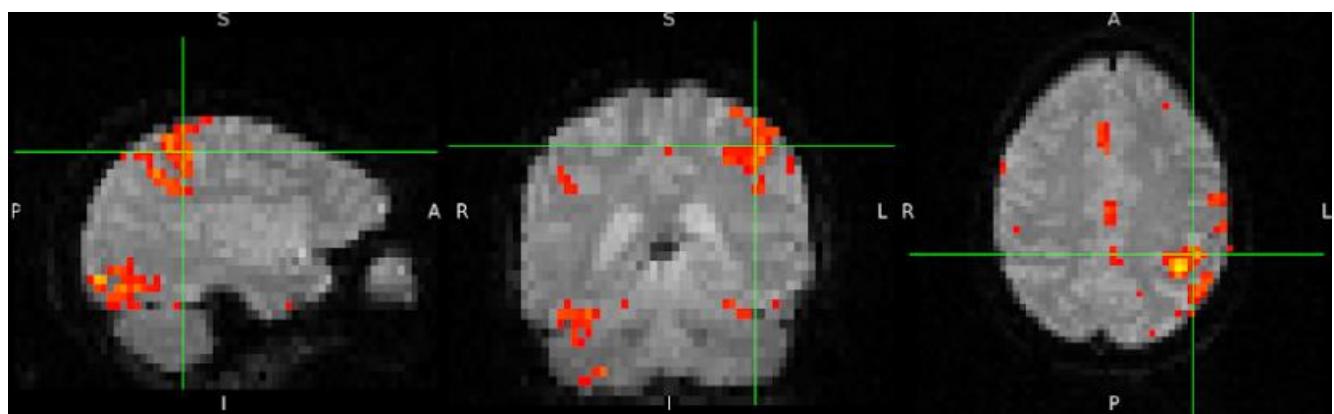
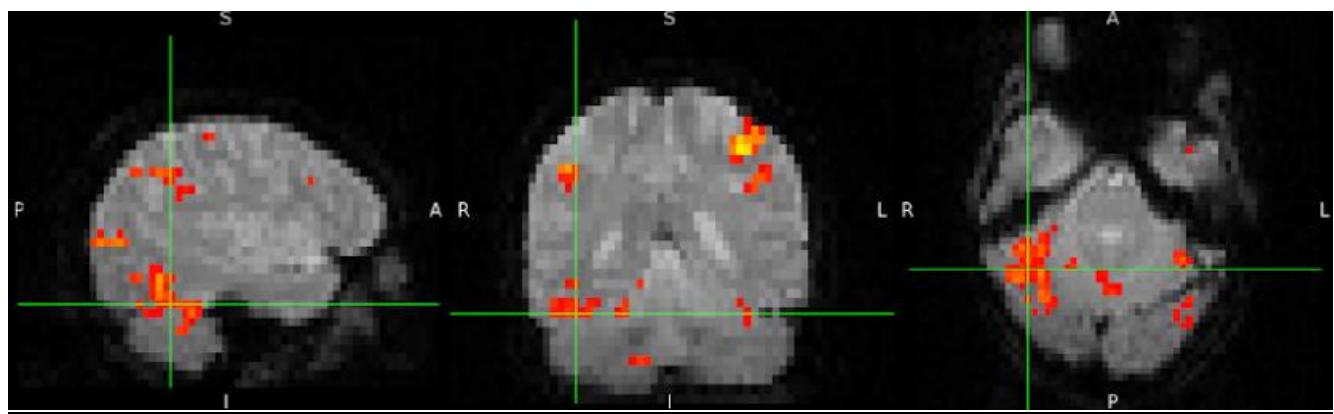
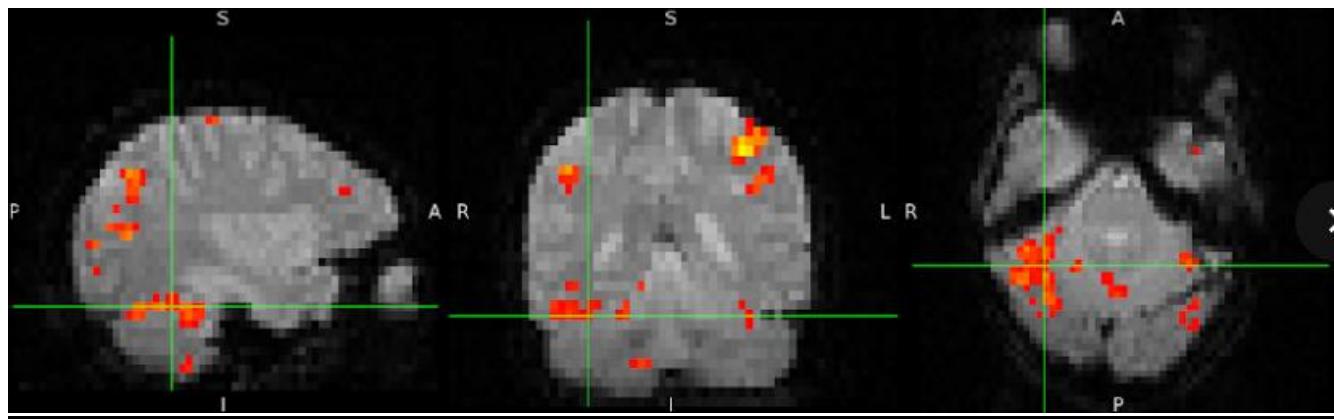


Threshold activations images:



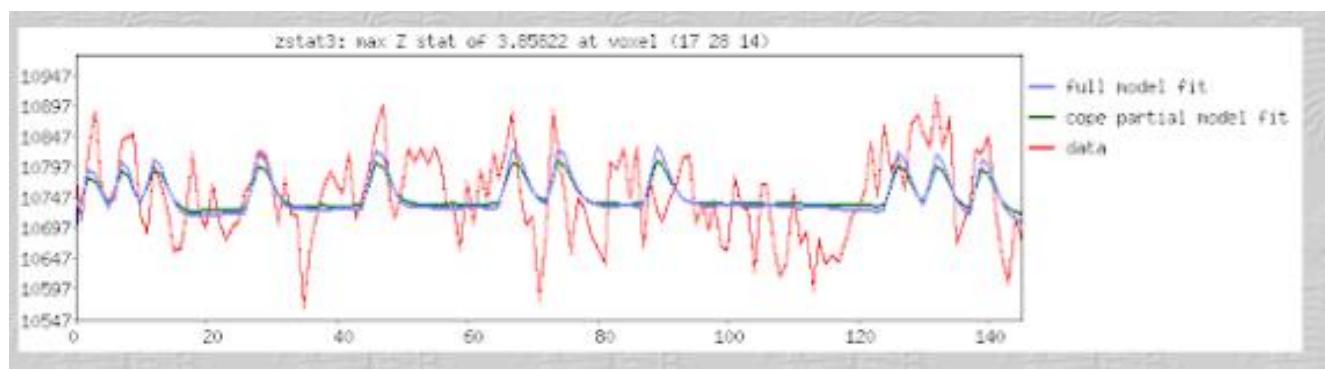
Visualization on FSL:



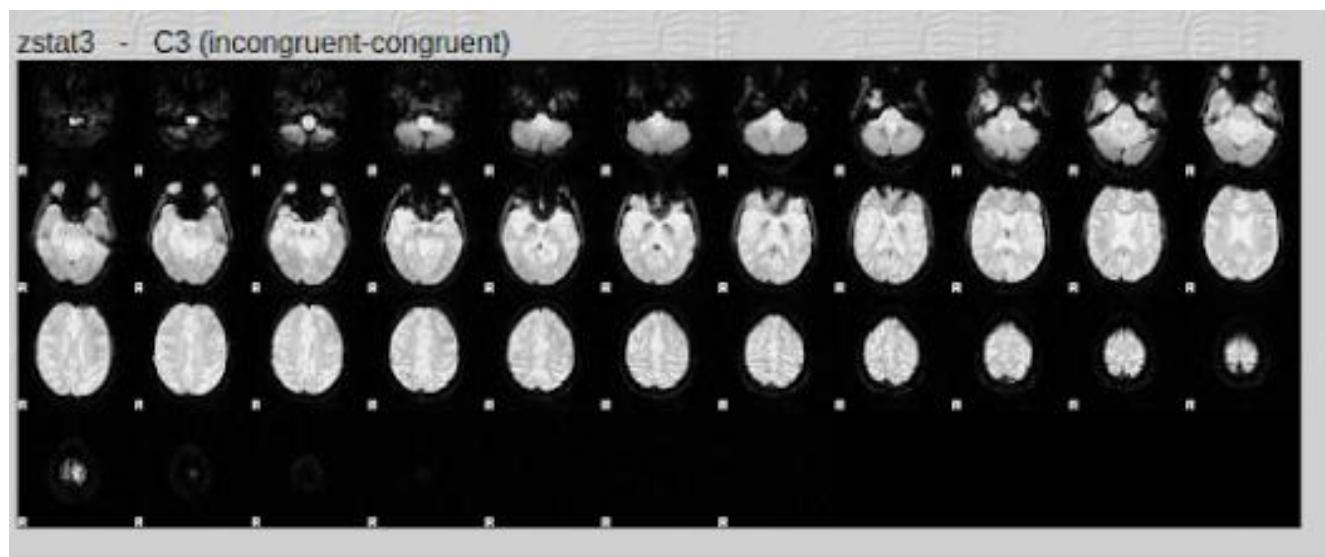


- **Z-Stat 3: Incongruent-Congruent Condition:**

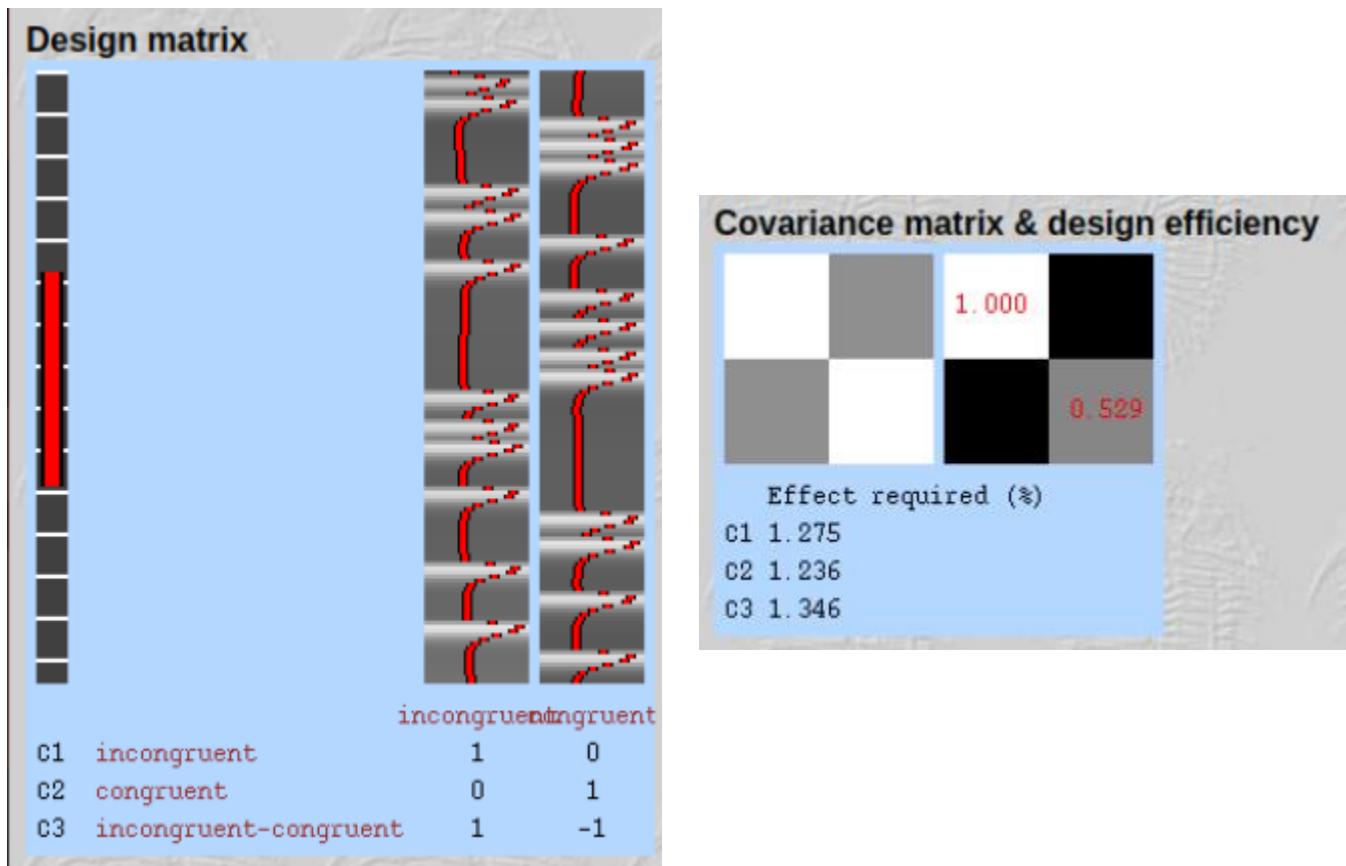
Time series plot:



Threshold activations images:

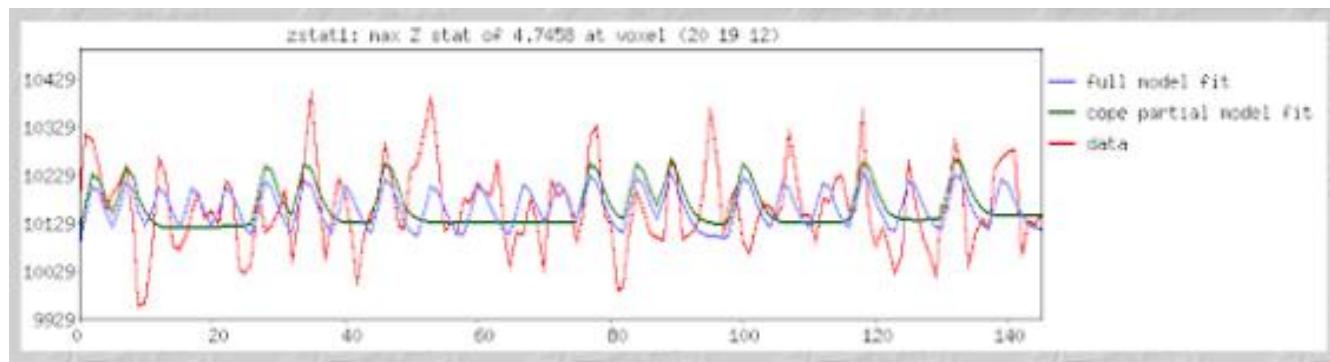


Sub-22: Run 2:

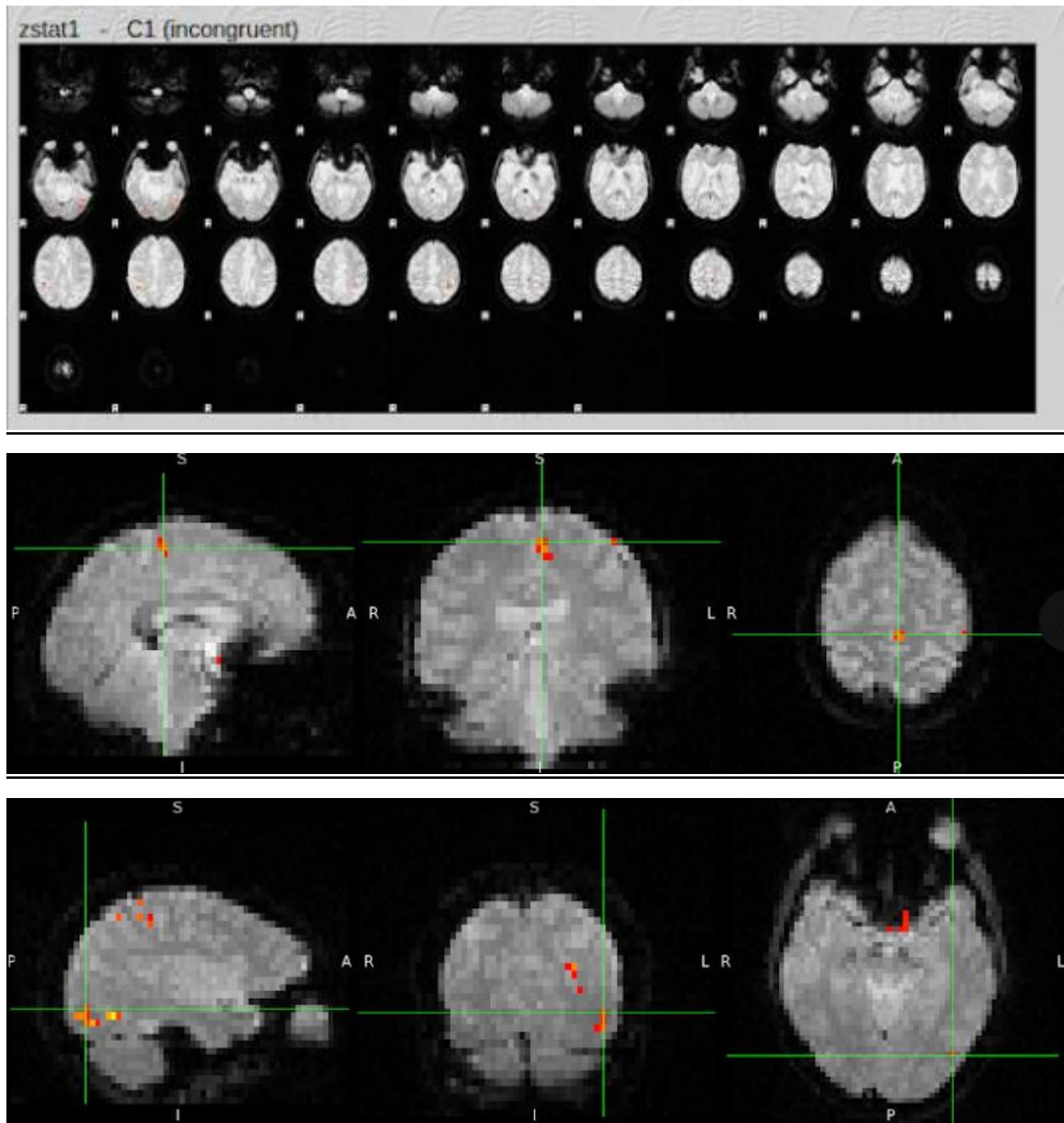


- **Z-Stat 1: Incongruent Condition:**

Time series plot:

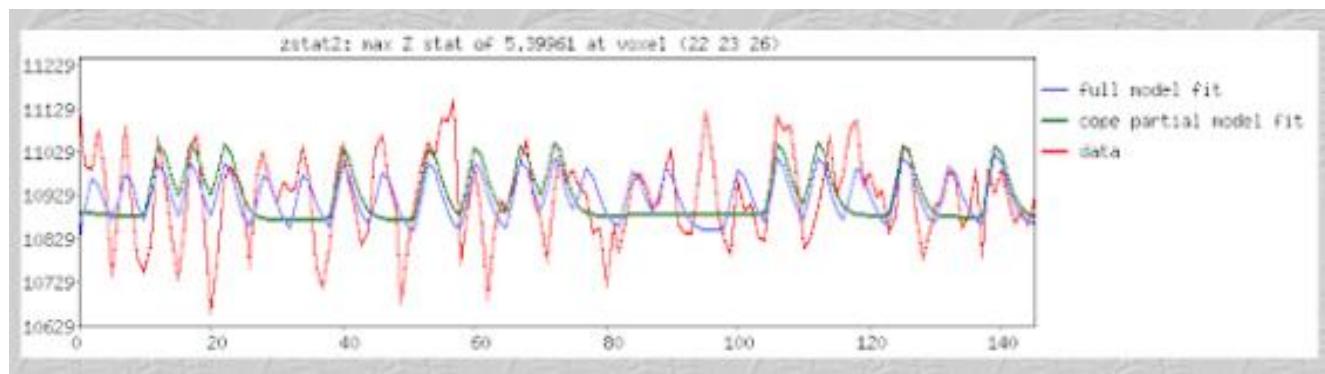


Threshold activations images:

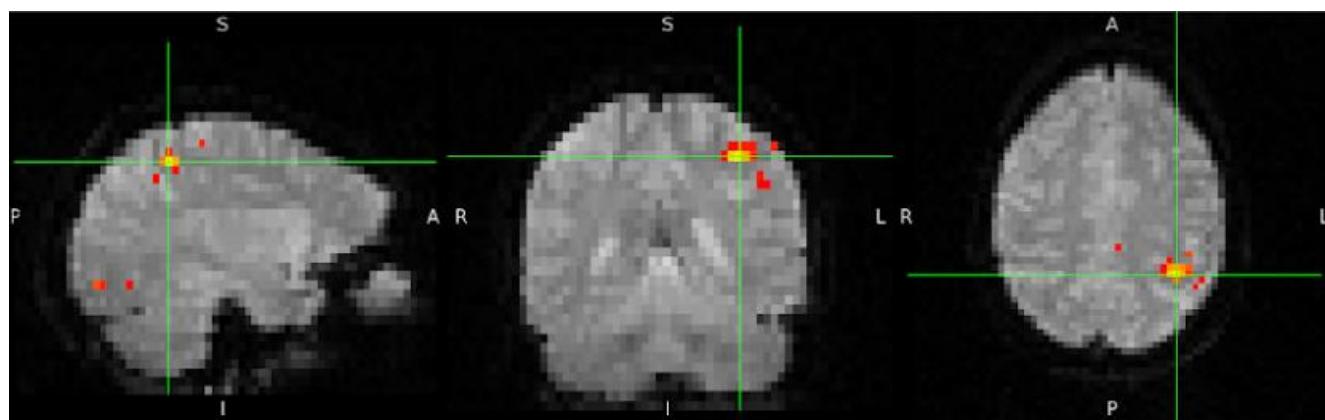
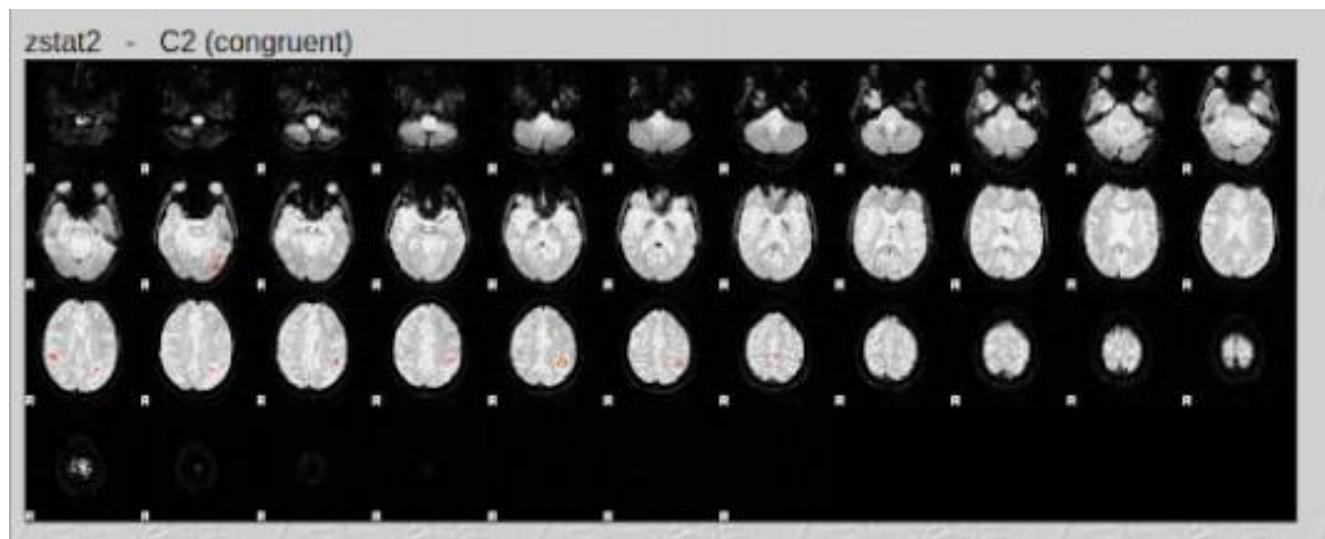


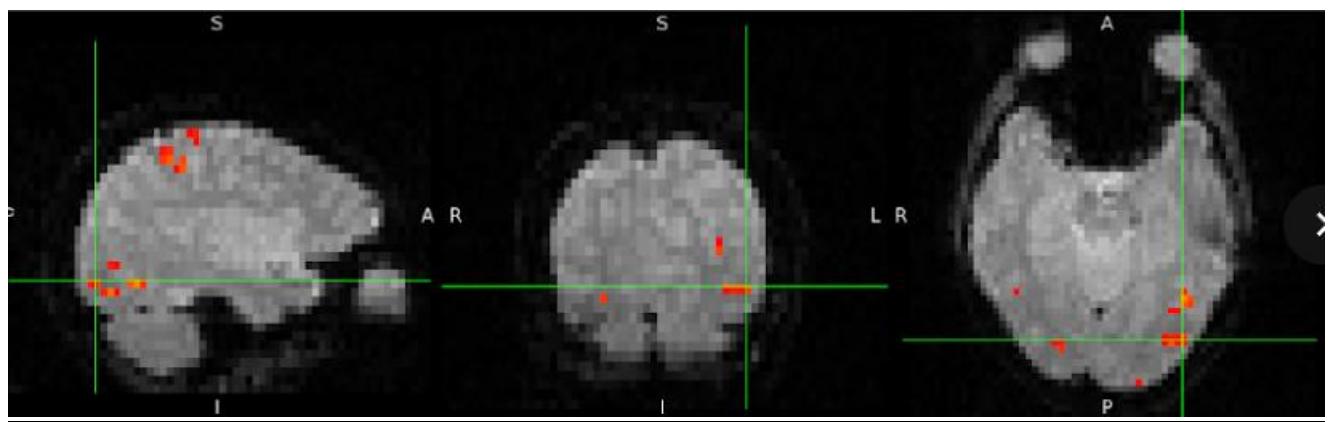
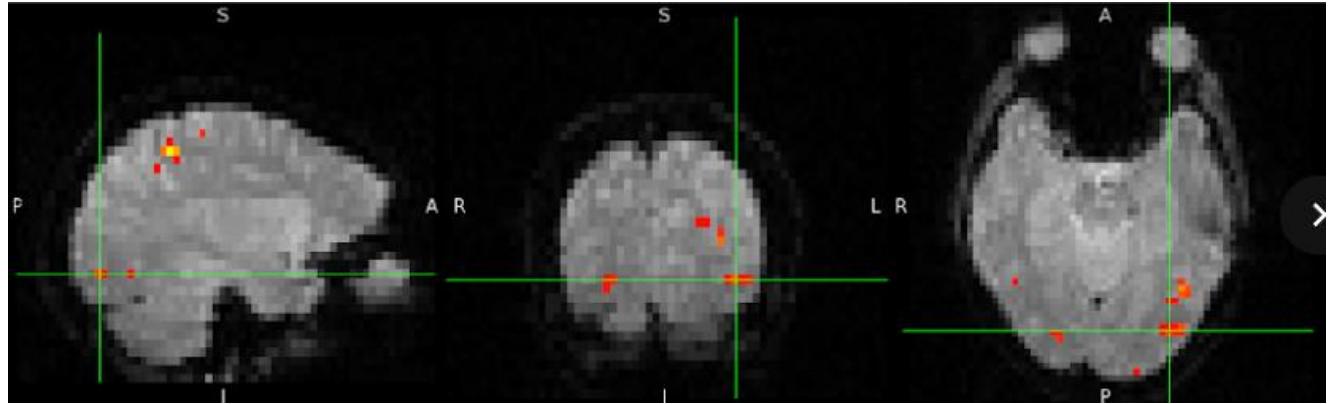
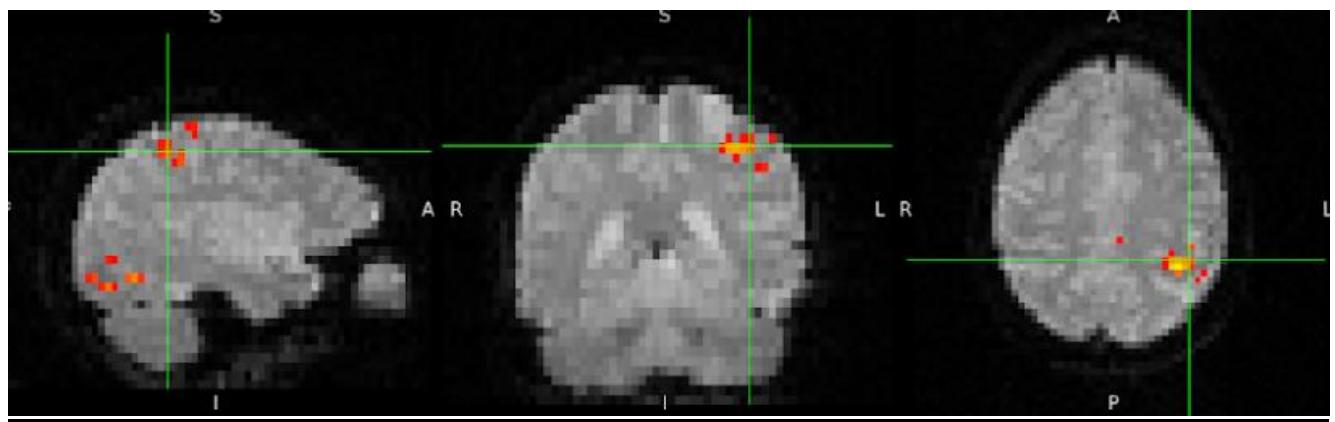
- **Z-Stat 2: Congruent Condition:**

Time series plot:



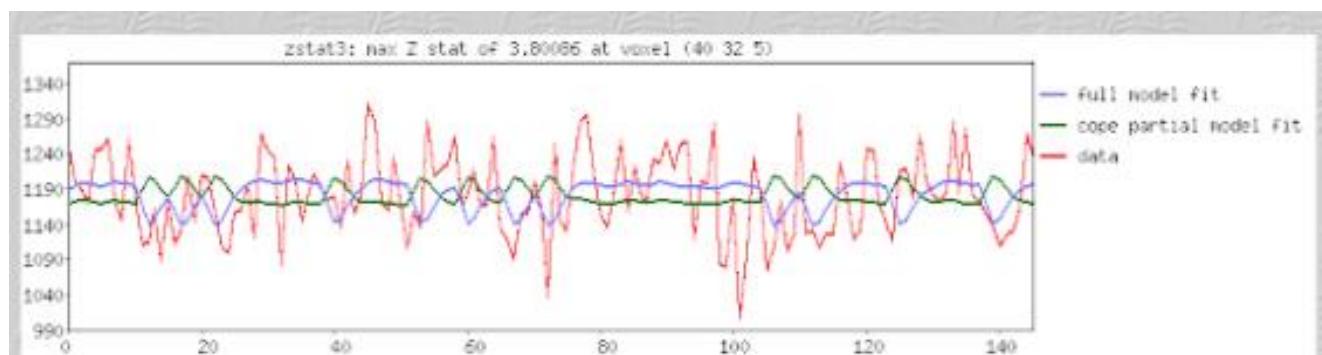
Threshold activations images:



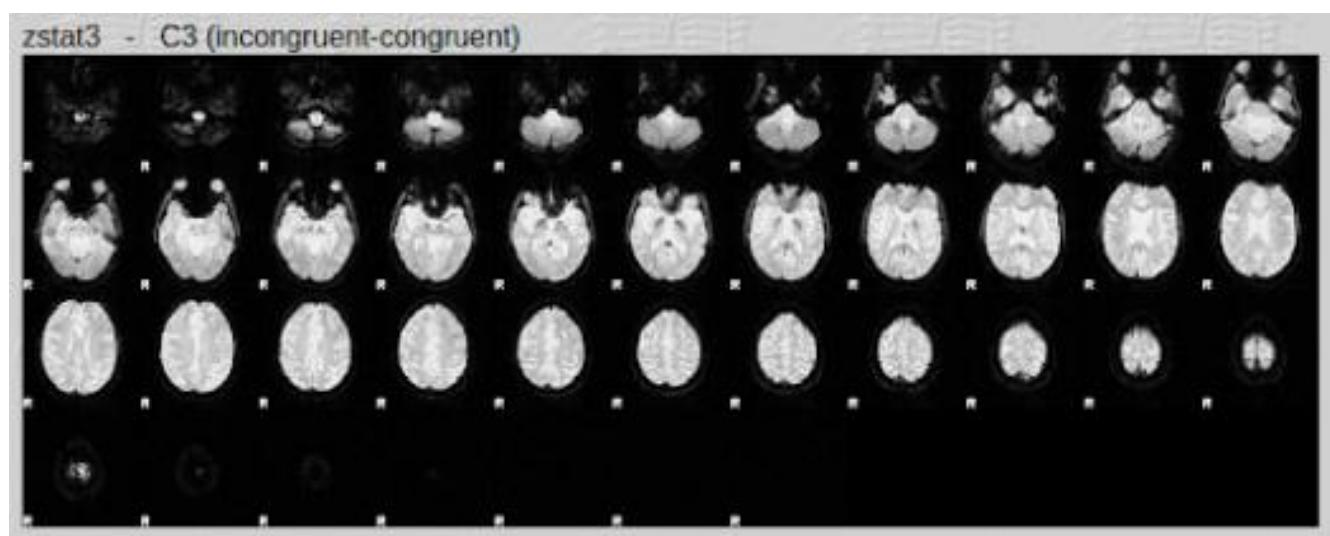


- **Z-Stat 3: Incongruent-Congruent Condition:**

Time series plot:



Threshold activations images:



Task 7: Design Script Optimization:

In order to move on to 2nd and higher level analysis, we need to apply 1st level analysis on all subjects for both runs of each. This could be a tedious task, especially if the dataset is large, so we can use scripts to help us automate that. We have a script –run_1stLevel.sh-- and 2 design files, one for each run file. In order to make this more efficient I joined the 2 design files together and modified the run_1stLevel.sh script to accommodate to that as well.

run_1stLevel.sh:

The modification done here is that instead of running design1.fsf or design2.fsf, it runs one file but passes a feat variable that tells the design file which run is specified.

```
#!/bin/bash

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/${subj}_T1w_brain_f02.nii.gz ]; then
        echo "Skull-stripped brain not found, using bet with a fractional
intensity threshold of 0.2"
        bet2 anat/${subj}_T1w.nii.gz \
            anat/${subj}_T1w_brain_f02.nii.gz -f 0.2
    fi

    # Copy the single design file into the subject directory and modify it
    cp ../design.fsf .

    # Modify the design file for run 1
    sed -i '' "s|sub-08|${subj}|g" design.fsf
    sed -i '' "s|RUN_NUMBER|1|g" design.fsf
    echo "====> Starting feat for run 1"
    feat design.fsf

    # Modify the design file for run 2
    sed -i '' "s|RUN_NUMBER|2|g" design.fsf
    echo "====> Starting feat for run 2"
    feat design.fsf

    echo
```

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

echo
```

Here we have the design.fsf

Given the variable `feat`, it uses it in if conditions to decide which paths to choose. Key parts this affects are choosing which `feat` directory to choose from and which `txt` file run to choose from. For example, choosing “`congruent_run1.txt`” or “`congruent_run2.txt`”.

```
# FEAT version number
set fmri(version) 6.00

# Are we in MELODIC?
set fmri(inmelodic) 0

# Analysis level
set fmri(level) 1

# Which stages to run
set fmri(analysis) 7

# Use relative filenames
set fmri(relative_yn) 0

# Balloon help
set fmri(help_yn) 1

# Run Featwatcher
set fmri(featwatcher_yn) 1

# Cleanup first-level standard-space images
set fmri(sscleanup_yn) 0

# Output directory - conditional based on run number
if $RUN_NUMBER == 1 then
    set fmri(outputdir) "run1"
else
    set fmri(outputdir) "run2"
end

# TR(s)
set fmri(tr) 2.000000
```

```
# Total volumes
set fmri(npts) 146

# Delete volumes
set fmri(ndelete) 0

# Perfusion tag/control order
set fmri(tagfirst) 1

# Number of first-level analyses
set fmri(multiple) 1

# Higher-level input type
set fmri(inputtype) 2

# Carry out pre-stats processing?
set fmri(filtering_yn) 1

# Brain/background threshold, %
set fmri(brain_thresh) 10

# Critical z for design efficiency calculation
set fmri(critical_z) 5.3

# Noise level
set fmri(noise) 0.66

# Noise AR(1)
set fmri(noisear) 0.34

# Motion correction
set fmri(mc) 1

# Spin-history (currently obsolete)
set fmri(sh_yn) 0

# B0 fieldmap unwarping?
set fmri(regunwarp_yn) 0

# GDC Test
set fmri(gdc) ""

# EPI dwell time (ms)
set fmri(dwells) 0.0

# EPI TE (ms)
```

```
set fmri(te) 0.0

# % Signal loss threshold
set fmri(signallossthresh) 10

# Unwarp direction
set fmri(unwarp_dir) y-

# Slice timing correction
set fmri(st) 0

# Slice timings file
set fmri(st_file) ""

# BET brain extraction
set fmri(bet_yn) 1

# Spatial smoothing FWHM (mm)
set fmri(smooth) 5

# Intensity normalization
set fmri(norm_yn) 0

# Perfusion subtraction
set fmri(perfsub_yn) 0

# Highpass temporal filtering
set fmri(temphp_yn) 1

# Lowpass temporal filtering
set fmri(templp_yn) 0

# MELODIC ICA data exploration
set fmri(melodic_yn) 0

# Carry out main stats?
set fmri(stats_yn) 1

# Carry out prewhitening?
set fmri(prewhiten_yn) 1

# Add motion parameters to model
set fmri(motionevs) 0
set fmri(motionevsbeta) ""
set fmri(scriptevsbeta) ""

# Robust outlier detection in FLAME?
```

```
set fmri(robust_yn) 0

# Higher-level modelling
set fmri(mixed_yn) 2

# Higher-level permutations
set fmri(randomisePermutations) 5000

# Number of EVs
set fmri(evs_orig) 2
set fmri(evs_real) 2
set fmri(evs_vox) 0

# Number of contrasts
set fmri(ncon_orig) 3
set fmri(ncon_real) 3

# Number of F-tests
set fmri(nftests_orig) 0
set fmri(nftests_real) 0

# Add constant column to design matrix? (obsolete)
set fmri(constcol) 0

# Carry out post-stats steps?
set fmri(poststats_yn) 1

# Pre-threshold masking?
set fmri(threshmask) ""

# Thresholding
set fmri(thresh) 3

# P threshold
set fmri(prob_thresh) 0.05

# Z threshold
set fmri(z_thresh) 3.1

# Z min/max for colour rendering
set fmri(zdisplay) 0

# Z min in colour rendering
set fmri(zmin) 2

# Z max in colour rendering
set fmri(zmax) 8
```

```
# Colour rendering type
set fmri(rendertype) 1

# Background image for higher-level stats overlays
set fmri(bgimage) 1

# Create time series plots
set fmri(tsplot_yn) 1

# Registration to initial structural
set fmri(reginitial_highres_yn) 0

# Search space for registration to initial structural
set fmri(reginitial_highres_search) 90

# Degrees of Freedom for registration to initial structural
set fmri(reginitial_highres_dof) 3

# Registration to main structural
set fmri(reghighres_yn) 1

# Search space for registration to main structural
set fmri(reghighres_search) 180

# Degrees of Freedom for registration to main structural
set fmri(reghighres_dof) 12

# Registration to standard image?
set fmri(regstandard_yn) 1

# Use alternate reference images?
set fmri(alternateReference_yn) 0

# Standard image
set fmri(regstandard) "/home/nariman/fsl/data/standard/MNI152_T1_2mm_brain"

# Search space for registration to standard space
set fmri(regstandard_search) 180

# Degrees of Freedom for registration to standard space
set fmri(regstandard_dof) 12

# Do nonlinear registration from structural to standard space?
set fmri(regstandard_nonlinear_yn) 0

# Control nonlinear warp field resolution
```

```
set fmri(regstandard_nonlinear_warpres) 10

# High pass filter cutoff
set fmri(paradigm_hp) 100

# Total voxels
set fmri(totalVoxels) 23920640

set fmri(fnirt_config) "T1_2_MNI152_2mm"

# Number of lower-level copes feeding into higher-level analysis
set fmri(ncopeinputs) 0

# 4D AVW data or FEAT directory (1) - conditional based on run number
if $RUN_NUMBER == 1 then
    set feat_files(1) "/home/nariman/Flanker/Data/sub-08/func/sub-08_task-
flanker_run-1_bold"
else
    set feat_files(1) "/home/nariman/Flanker/Data/sub-08/func/sub-08_task-
flanker_run-2_bold"
end

# Add confound EVs text file
set fmri(confoundevs) 0

# Subject's structural image for analysis 1
set highres_files(1) "/home/nariman/Flanker/Data/sub-08/anat/sub-
08_T1w_brain_f02"

# EV 1 title
set fmri(evttitle1) "incongruent"

# Basic waveform shape (EV 1)
set fmri(shape1) 3

# Convolution (EV 1)
set fmri(convolve1) 2

# Convolve phase (EV 1)
set fmri(convolve_phase1) 0

# Apply temporal filtering (EV 1)
set fmri(tempfilt_yn1) 0

# Add temporal derivative (EV 1)
set fmri(deriv_yn1) 0
```

```
# Custom EV file (EV 1) - conditional based on run number
if $RUN_NUMBER == 1 then
    set fmri(custom1) "/home/nariman/Flanker/Data/sub-
08/func/incongruent_run1.txt"
else
    set fmri(custom1) "/home/nariman/Flanker/Data/sub-
08/func/incongruent_run2.txt"
end
```

```
# Gamma sigma (EV 1)
set fmri(gamma_sigma1) 3

# Gamma delay (EV 1)
set fmri(gamma_delay1) 6

# Orthogonalise EV 1 wrt EV 0
set fmri(ortho1.0) 0

# Orthogonalise EV 1 wrt EV 1
set fmri(ortho1.1) 0

# Orthogonalise EV 1 wrt EV 2
set fmri(ortho1.2) 0

# EV 2 title
set fmri(evtitle2) "congruent"

# Basic waveform shape (EV 2)
set fmri(shape2) 3

# Convolution (EV 2)
set fmri(convolve2) 2

# Convolve phase (EV 2)
set fmri(convolve_phase2) 0

# Apply temporal filtering (EV 2)
set fmri(tempfilt_yn2) 0

# Add temporal derivative (EV 2)
set fmri(deriv_yn2) 0

# Custom EV file (EV 2) - conditional based on run number
if $RUN_NUMBER == 1 then
    set fmri(custom2) "/home/nariman/Flanker/Data/sub-08/func/congruent_run1.txt"
else
    set fmri(custom2) "/home/nariman/Flanker/Data/sub-08/func/congruent_run2.txt"
```

```
end

# Gamma sigma (EV 2)
set fmri(gammasigma2) 3

# Gamma delay (EV 2)
set fmri(gammadelay2) 6

# Orthogonalise EV 2 wrt EV 0
set fmri(ortho2.0) 0

# Orthogonalise EV 2 wrt EV 1
set fmri(ortho2.1) 0

# Orthogonalise EV 2 wrt EV 2
set fmri(ortho2.2) 0

# Contrast & F-tests mode
set fmri(con_mode_old) orig
set fmri(con_mode) orig

# Display images for contrast_real 1
set fmri(conpic_real.1) 1

# Title for contrast_real 1
set fmri(conname_real.1) "incongruent"

# Real contrast_real vector 1 element 1
set fmri(con_real1.1) 1

# Real contrast_real vector 1 element 2
set fmri(con_real1.2) 0

# Display images for contrast_real 2
set fmri(conpic_real.2) 1

# Title for contrast_real 2
set fmri(conname_real.2) "congruent"

# Real contrast_real vector 2 element 1
set fmri(con_real2.1) 0

# Real contrast_real vector 2 element 2
set fmri(con_real2.2) 1.0

# Display images for contrast_real 3
set fmri(conpic_real.3) 1
```

```
# Title for contrast_real 3
set fmri(conname_real.3) "incongruent-congruent"

# Real contrast_real vector 3 element 1
set fmri(con_real3.1) 1.0

# Real contrast_real vector 3 element 2
set fmri(con_real3.2) -1.0

# Display images for contrast_orig 1
set fmri(conpic_orig.1) 1

# Title for contrast_orig 1
set fmri(conname_orig.1) "incongruent"

# Real contrast_orig vector 1 element 1
set fmri(con_orig1.1) 1

# Real contrast_orig vector 1 element 2
set fmri(con_orig1.2) 0

# Display images for contrast_orig 2
set fmri(conpic_orig.2) 1

# Title for contrast_orig 2
set fmri(conname_orig.2) "congruent"

# Real contrast_orig vector 2 element 1
set fmri(con_orig2.1) 0

# Real contrast_orig vector 2 element 2
set fmri(con_orig2.2) 1.0

# Display images for contrast_orig 3
set fmri(conpic_orig.3) 1

# Title for contrast_orig 3
set fmri(conname_orig.3) "incongruent-congruent"

# Real contrast_orig vector 3 element 1
set fmri(con_orig3.1) 1.0

# Real contrast_orig vector 3 element 2
set fmri(con_orig3.2) -1.0

# Contrast masking - use >0 instead of thresholding?
```

```
set fmri(conmask_zerthresh_yn) 0

# Mask real contrast/F-test 1 with real contrast/F-test 2?
set fmri(conmask1_2) 0

# Mask real contrast/F-test 1 with real contrast/F-test 3?
set fmri(conmask1_3) 0

# Mask real contrast/F-test 2 with real contrast/F-test 1?
set fmri(conmask2_1) 0

# Mask real contrast/F-test 2 with real contrast/F-test 3?
set fmri(conmask2_3) 0

# Mask real contrast/F-test 3 with real contrast/F-test 1?
set fmri(conmask3_1) 0

# Mask real contrast/F-test 3 with real contrast/F-test 2?
set fmri(conmask3_2) 0

# Do contrast masking at all?
set fmri(conmask1_1) 0

#####
# Now options that don't appear in the GUI

# Alternative (to BETting) mask image
set fmri(alternative_mask) ""

# Initial structural space registration initialisation transform
set fmri(init_initial_highres) ""

# Structural space registration initialisation transform
set fmri(init_highres) ""

# Standard space registration initialisation transform
set fmri(init_standard) ""

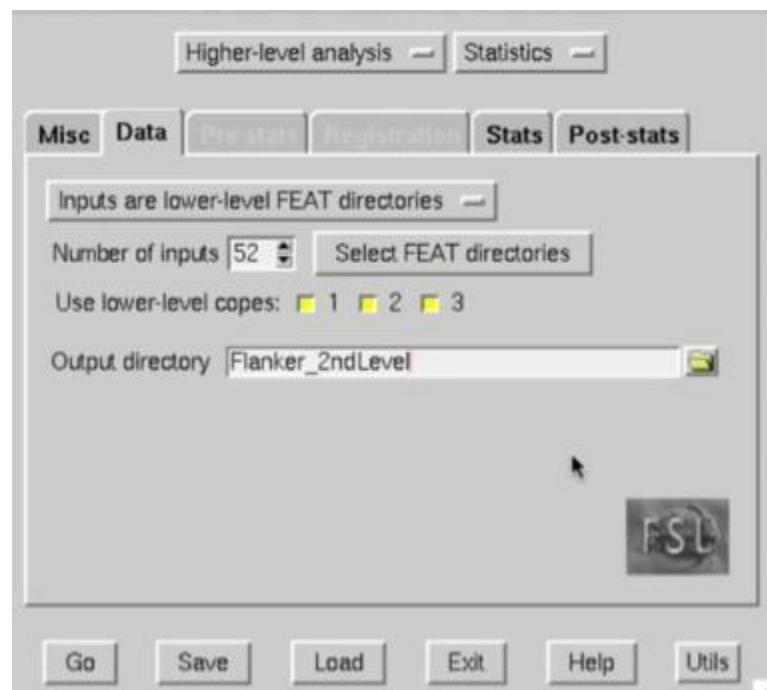
# For full FEAT analysis: overwrite existing .feat output dir?
set fmri(overwrite_yn) 0
```

Task 8: 2nd Level Analysis:

After completing the preprocessing and first-level analyses for all subjects in the Flanker dataset, the next step is to perform a second-level (group-level) analysis. This stage involves aggregating individual results by averaging the parameter estimates and contrast estimates from each subject's first-level analyses. The goal is to identify consistent patterns of activation across the entire cohort, providing meaningful insights into the neural mechanisms underlying the experimental task.

To begin, open the FEAT GUI from the Flanker directory by entering `Feat_gui` in the command line. Once the interface loads, navigate to the dropdown menu and select *Higher-Level Analysis*. This action adjusts the input settings, replacing the default options with *Select FEAT directories*—a necessary step for specifying the data to be included in the group analysis.

Given that the dataset consists of 26 subjects, each with two functional runs, there are a total of 52 FEAT directories to account for.



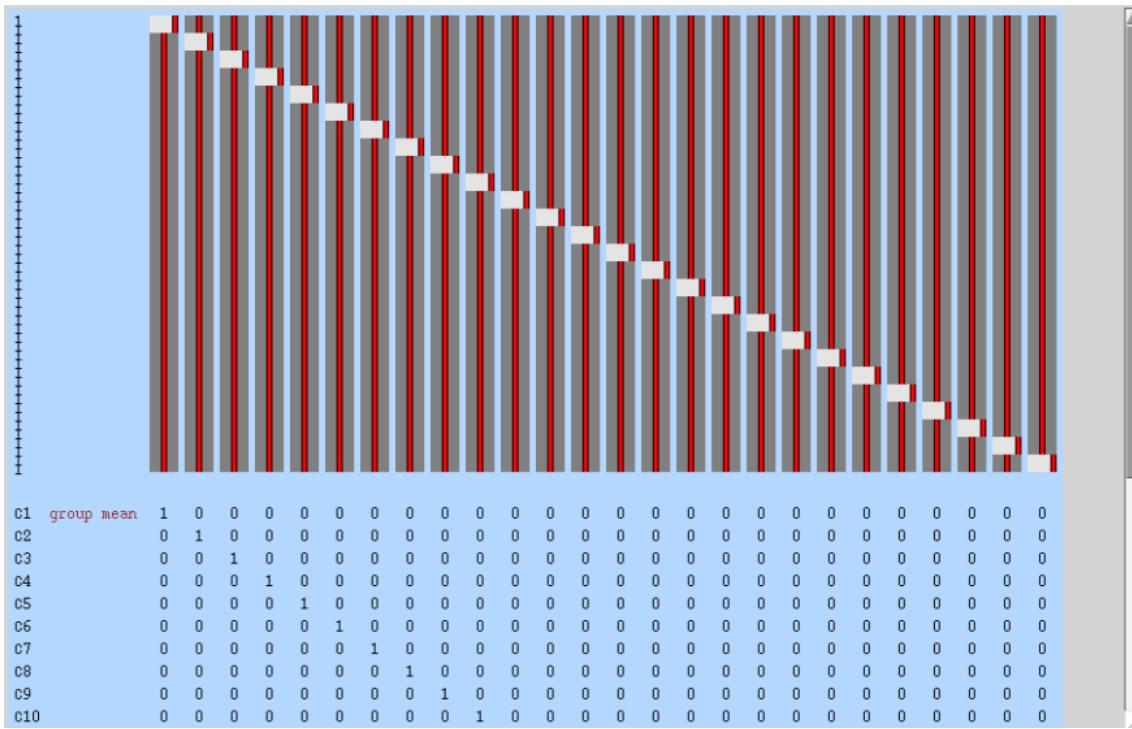
In the stats tab, we have multiple options to choose from. Since we simply want to take the average of the parameter estimates across the runs within each subject, we will use the Fixed Effects option. Once you have selected that, click on Full Model Setup.

EVs		Contrasts & F-tests															
		Number of main EVs	26	Number of additional, voxel-dependent EVs	0												
		Paste	Group	EV1	EV2	EV3	EV4	EV5	EV6	EV7	EV8	EV9	EV10	EV11	EV12		
Input 1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0		
Input 2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0		
Input 3	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
Input 4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
Input 5	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
Input 6	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0		
Input 7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
Input 8	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
Input 9	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0		
Input 10	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0		
Input 11	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0		
Input 12	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0		
Input 13	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0		
Input 14	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0		
Input 15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
Input 16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Input 17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Input 18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Input 19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Input 20	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

This will display a window with the number of rows representing the number of individual parameter estimates - in our case, 52. For the Number of main EVs, change this to 26, which is the number of subjects in our dataset. Then change the numbers in each column to 1 where you want to take the average for the parameter estimates for that subject. In our case, the first two rows for column 1 would be changed to 1, and next two rows for column 2 would be changed to 1, and so on.

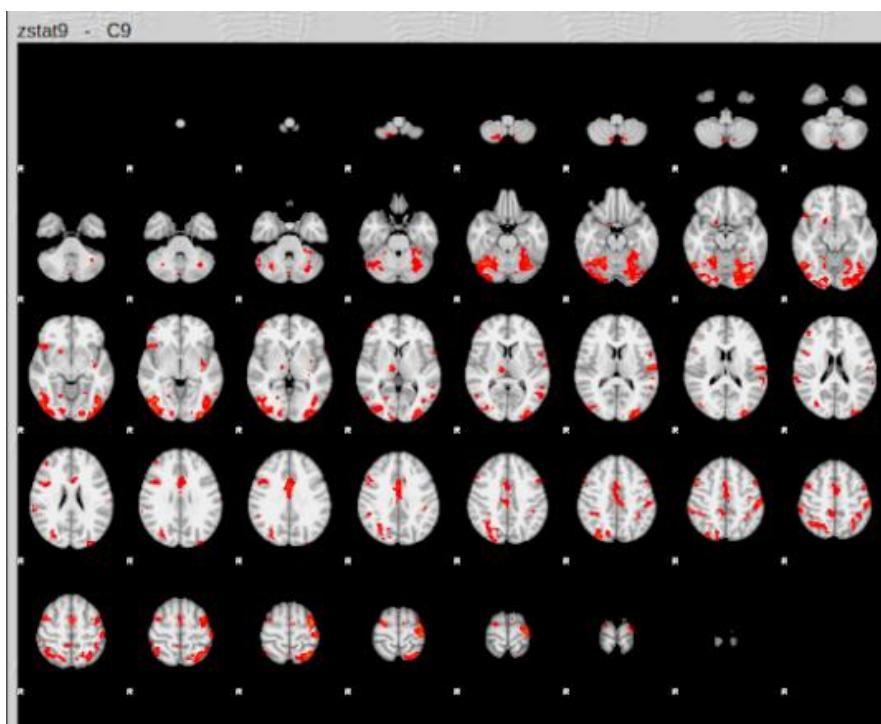
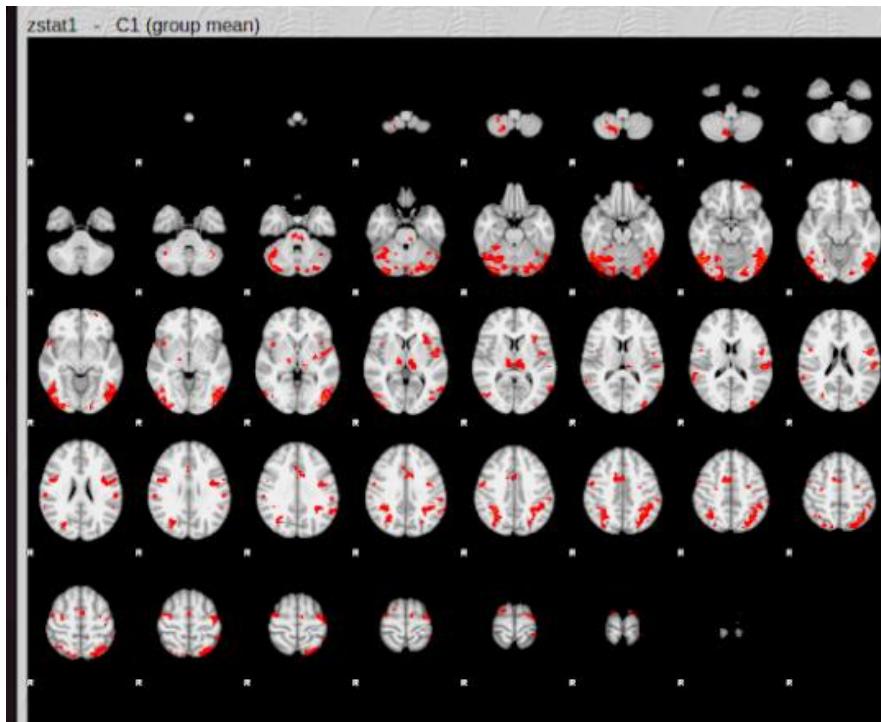
When you have finished, click on the Contrasts & F-tests tab, and change the number of Contrasts to 26. Change all of the numbers on the diagonal to 1; this will create a single contrast estimate for each subject that is the average of that subject's parameter estimates.

When you have finished setting up the GLM and contrasts and click Done, you should see something like this:

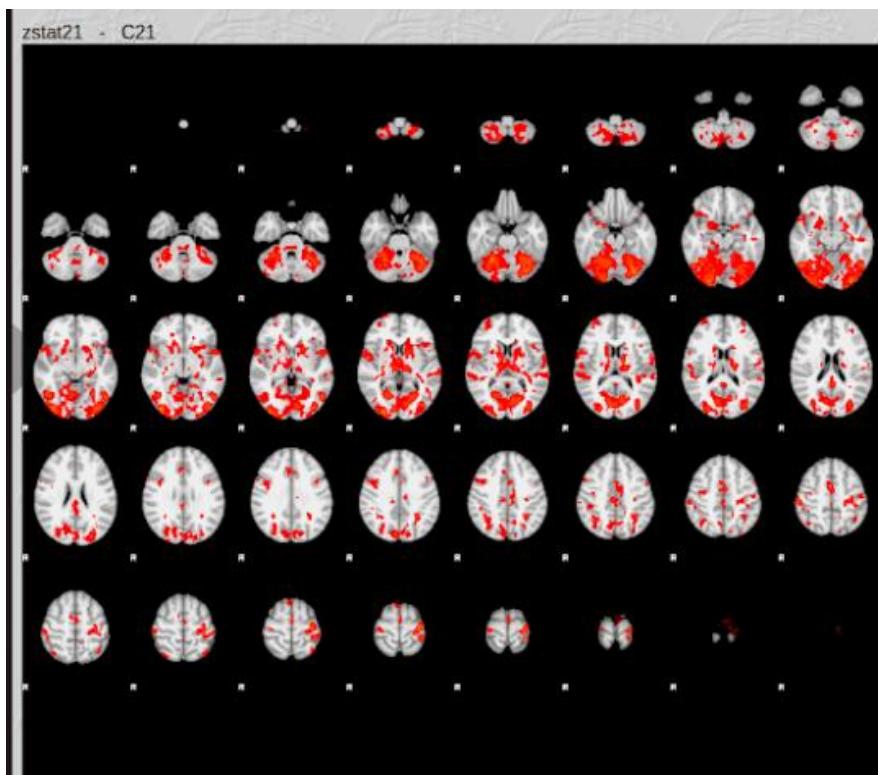
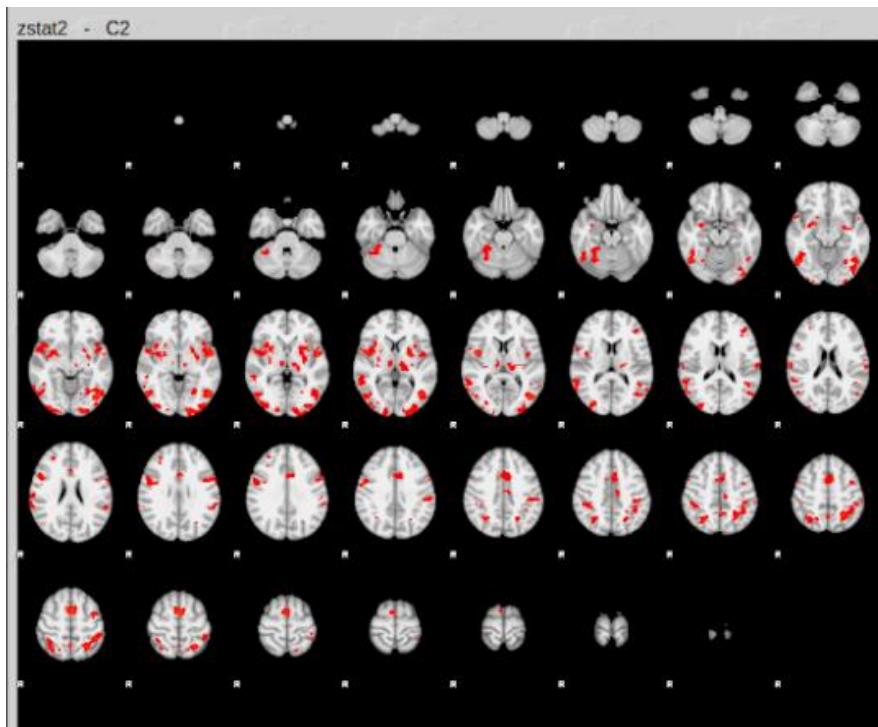


Analyzing Results: Showing random subjects:

- Cope 1 (Incongruent):

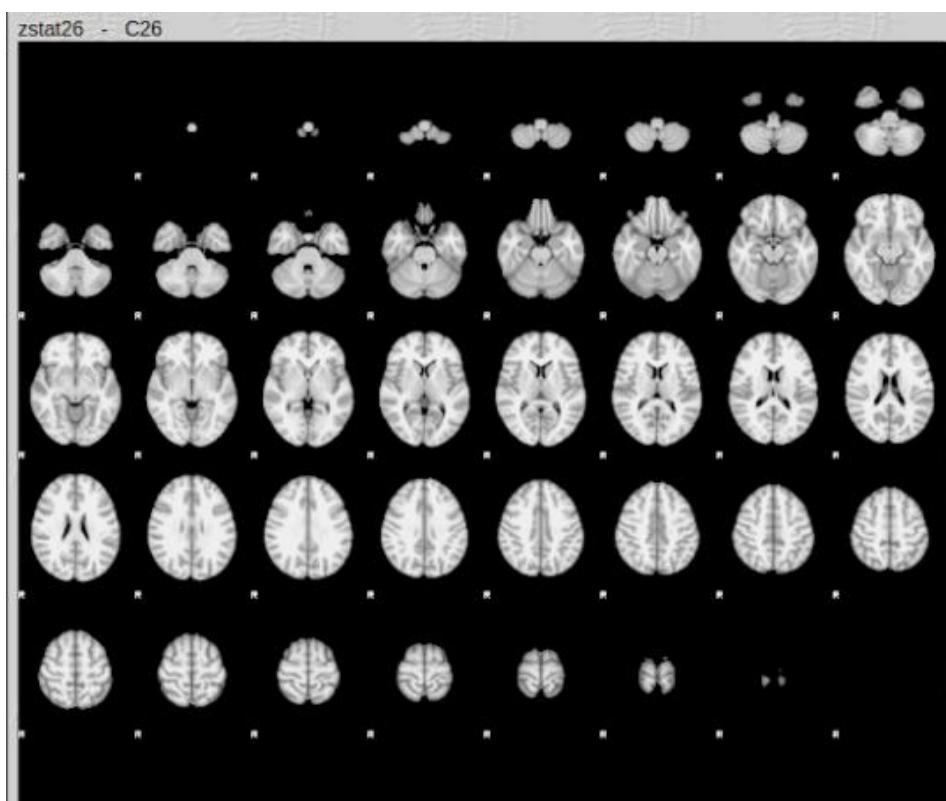
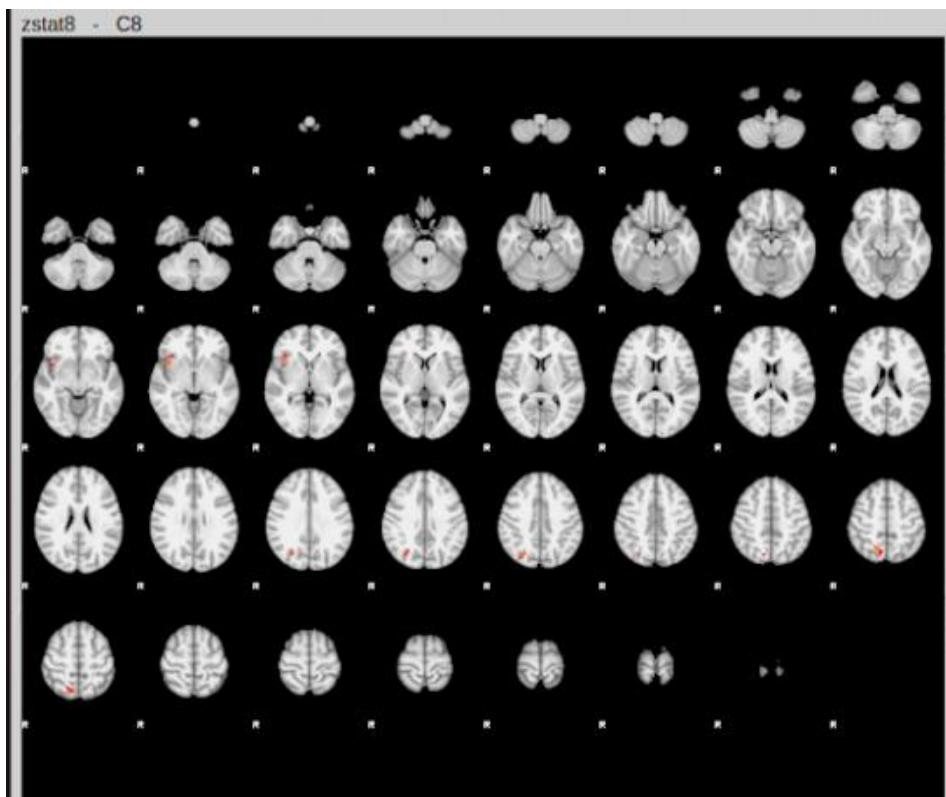


- Cope 2 (Congruent):



- Cope 3 (Incongruent-Congruent):

Activation showed in subjects 1,8 and 21 only while the rest look blank



Task 9: 2nd Level Analysis Stats Tab Options:

In FSL, second-level (and third-level) analyses refer to group-level or multi-session analyses where the aim is to generalize findings beyond a single subject or single run. These options presented in the Stats tab define how data from multiple subjects or sessions are combined and statistically tested.

Fixed Effects

This one is the most straightforward. Fixed Effects assumes that the effect we're looking at is the same across all inputs—no variability is considered between them. Basically, it just averages the data from each subject or session, usually giving more weight to more reliable data (lower variance). The key thing about this method is that it doesn't try to generalize the results beyond the specific data we're using. So it's great for comparing multiple runs within the same subject, but it's **not** meant for making population-level conclusions. I think of it like saying, “Here’s what happened in this dataset,” without trying to say it would happen again in new data.

Mixed Effects: Simple OLS

This one starts to move toward population-level inference, which is useful when you have data from multiple subjects. What FSL does here is take the average of each subject's contrast estimate and then runs a simple t-test on those. But the important catch is that it doesn't take into account how much variance there is for each subject. So if one subject has a noisy scan and another has a super clean one, they still count the same. This can be okay if your subjects are pretty similar in data quality, but if there's a lot of variability, it might give misleading results. It's fast, though, and sometimes that's useful if you're just doing an initial check.

Mixed Effects: FLAME 1

This is the method I use the most often. FLAME 1 stands for FMRIB's Local Analysis of Mixed Effects, and it's a proper mixed-effects model. That means it takes into account both the group-level effect we're interested in and the random differences between subjects. What I really like is that it actually weights each subject's contribution based on how reliable their data is. So a subject with a low-variance contrast estimate has more influence than one with a noisy result. It's more statistically sound than OLS, especially when you know your subjects differ in data quality or response. It takes a little longer to run, but I think it's worth it for the improved accuracy.

Mixed Effects: FLAME 1+2

FLAME 1+2 takes things a step further. It includes everything from FLAME 1, but also adds a more advanced statistical estimation using MCMC (Markov Chain Monte Carlo). This helps especially when you have a **small sample size** (like under 10 subjects), where the variability estimates can be really unstable. FLAME 1+2 is more rigorous and provides the most accurate results in that case. The downside is that it's much slower—it can take hours depending on your data. So unless I'm working with a small group and really need the precision, I usually stick with FLAME 1.

Randomise (Non-parametric)

This one is a completely different approach. Randomise is FSL's tool for non-parametric testing, which means it doesn't rely on any assumptions about the underlying data distribution. It works by running thousands of permutations—basically shuffling the data in various ways to see how often the effect happens by chance. It's super powerful for unusual designs or when you think the normal assumptions might not hold. I also use it when I want to apply TFCE (Threshold-Free Cluster Enhancement), which works really well with Randomise. It's slower than the parametric methods, but in tricky cases, it's the most reliable option.

Overall, FSL gives a lot of flexibility depending on what kind of analysis you're doing and how much you care about speed versus statistical accuracy. For within-subject analysis or exploratory checks, Fixed Effects or OLS might be enough. But for real group-level inferences, FLAME 1 is usually the best option. If you're working with a small sample, go with FLAME 1+2, and if you're really concerned about violating statistical assumptions or just want a robust check, Randomise is the way to go.

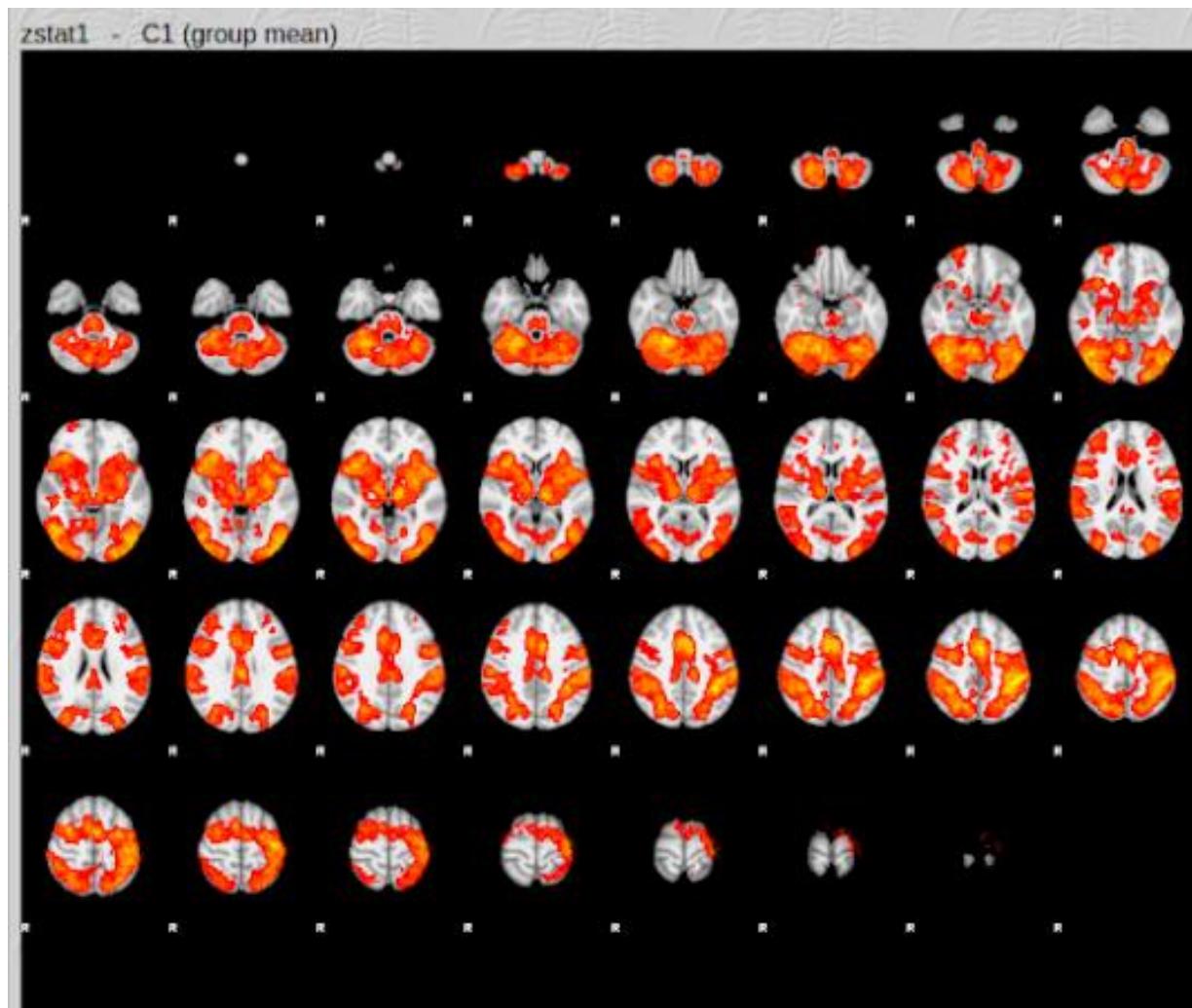
Task 10: 3rd Level Analysis:

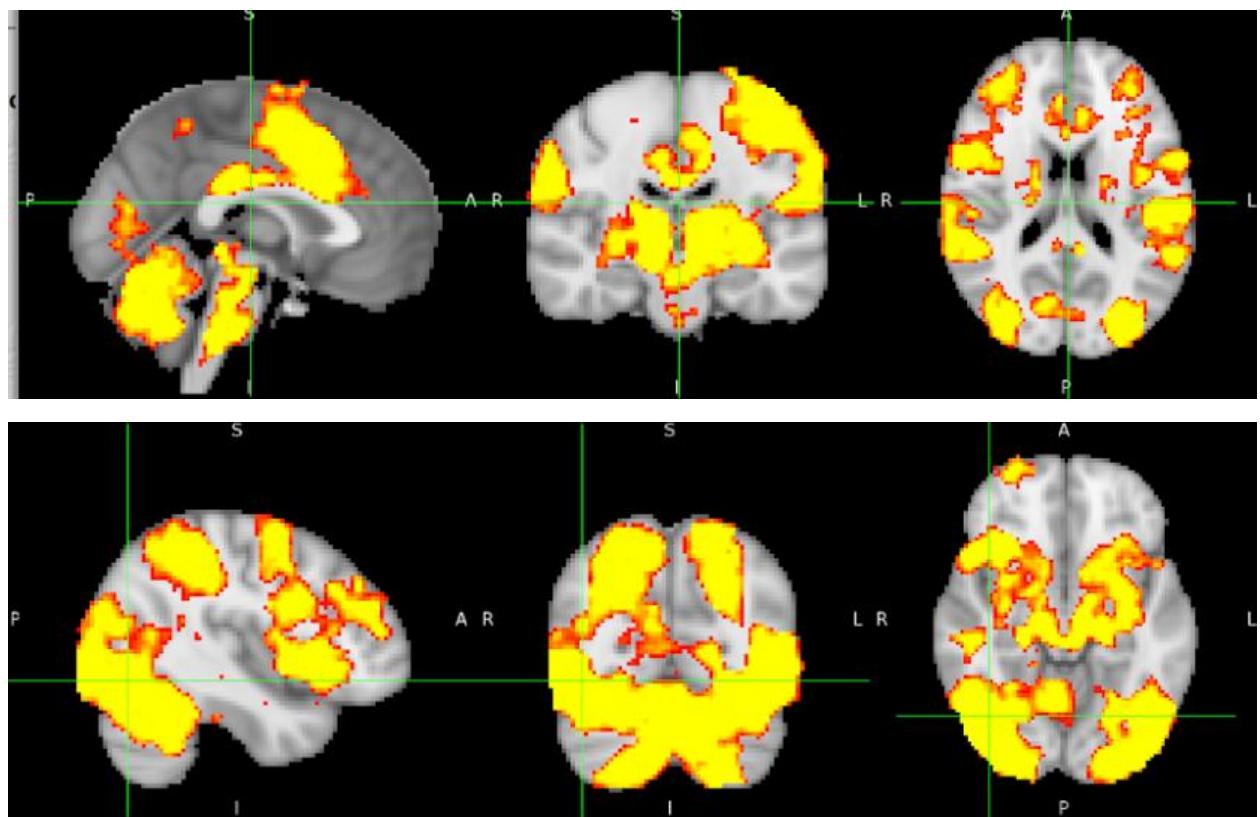
Our goal in analyzing this dataset is to generalize the results to the population that the sample was drawn from. In other words, if we see changes in brain activity in our sample, can we say that these changes would likely be seen in the population as well?

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.

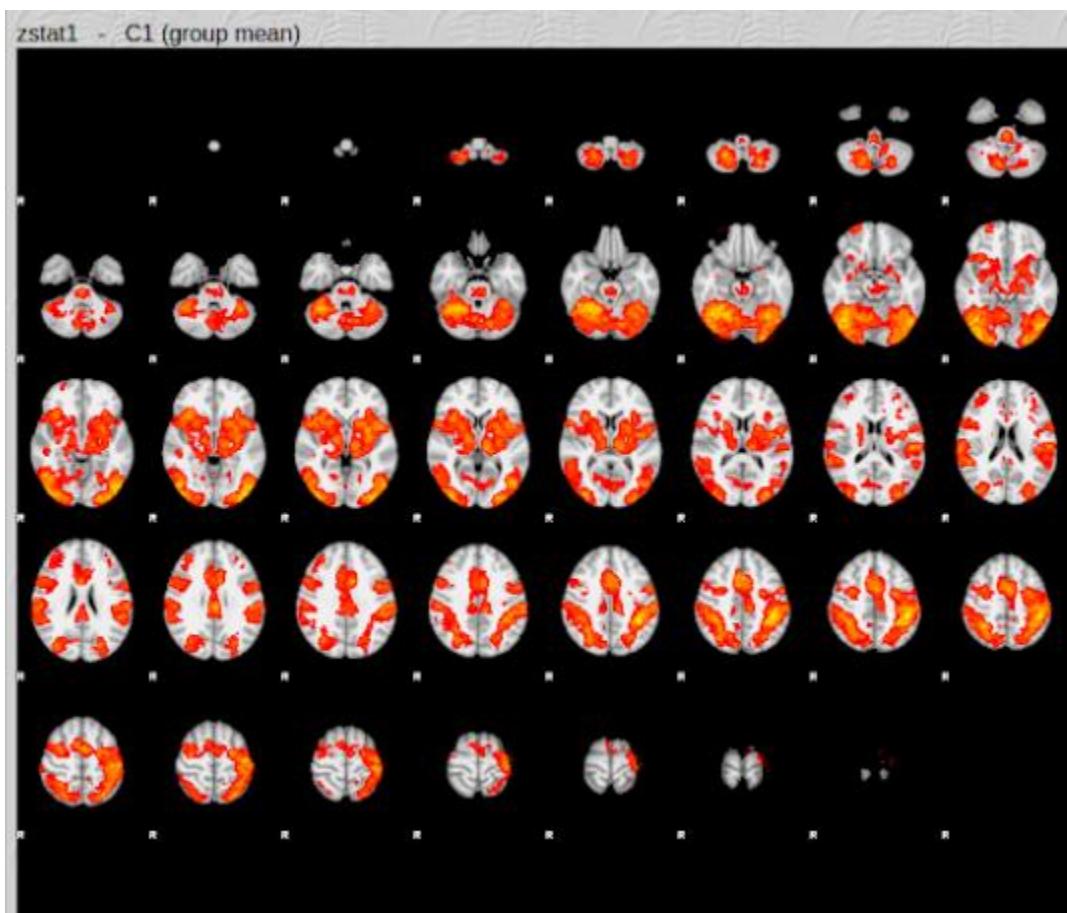
3rd Level Analysis Results:

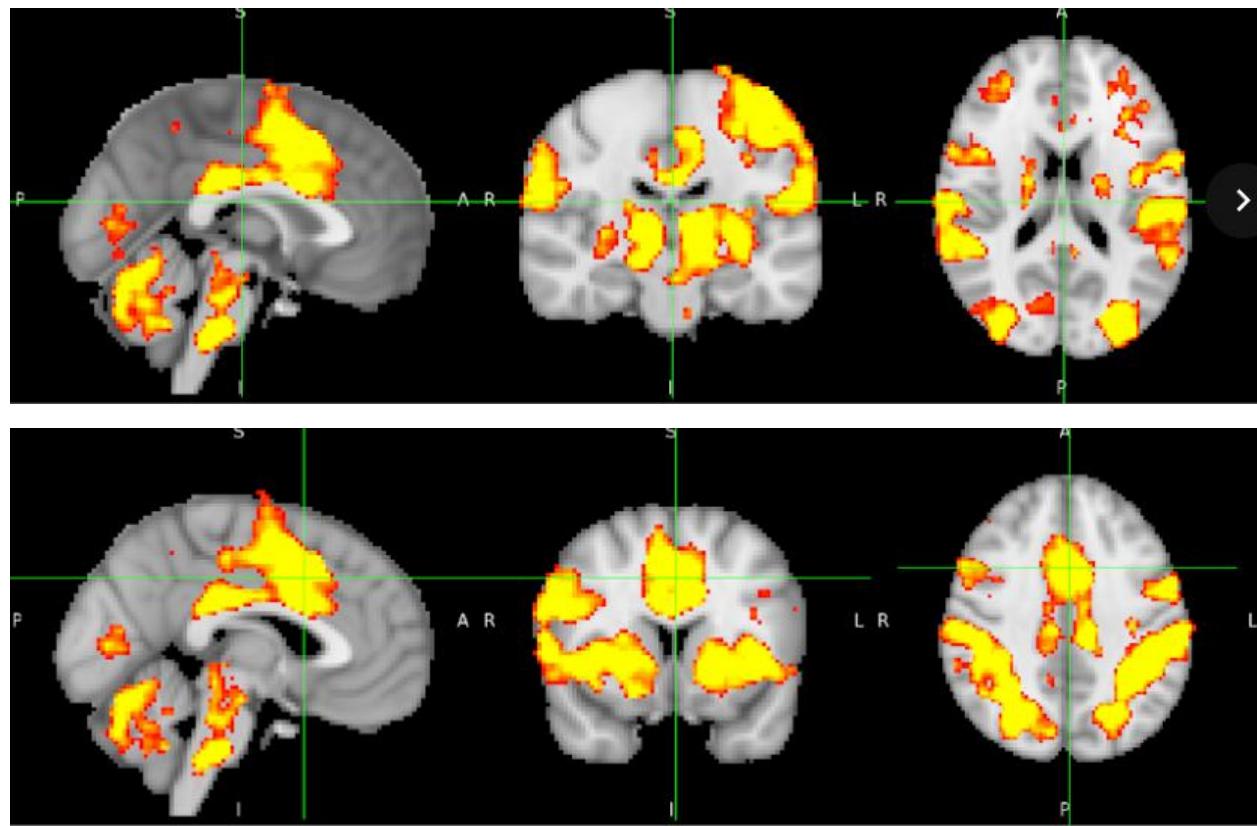
- **Cope 1 (Incongruent):**





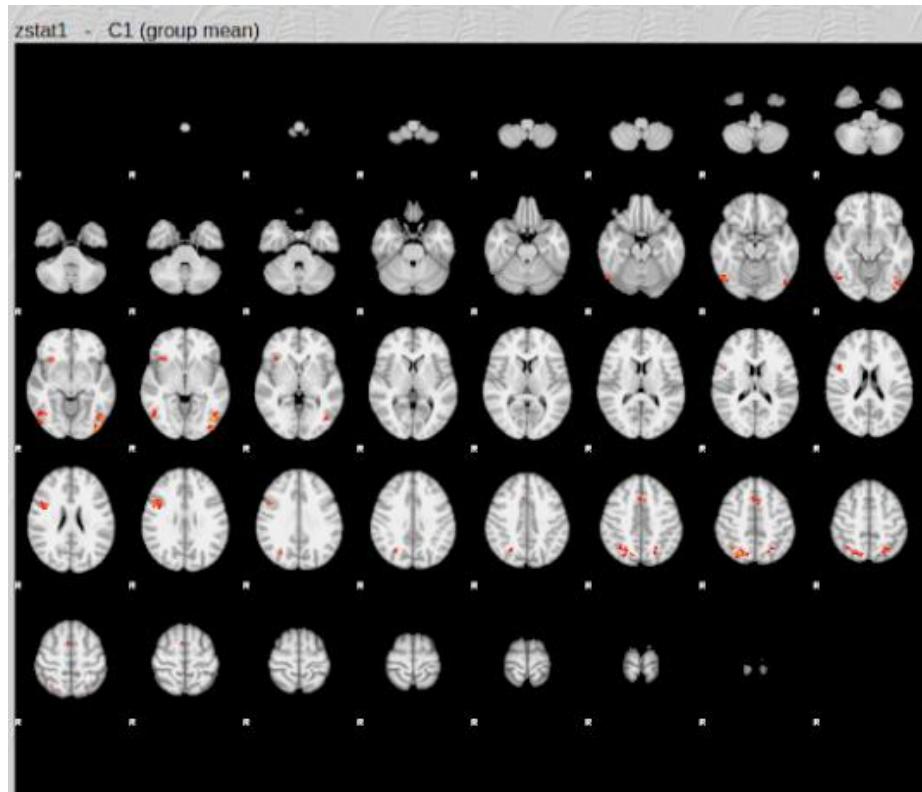
- Cope 2 (Congruent):

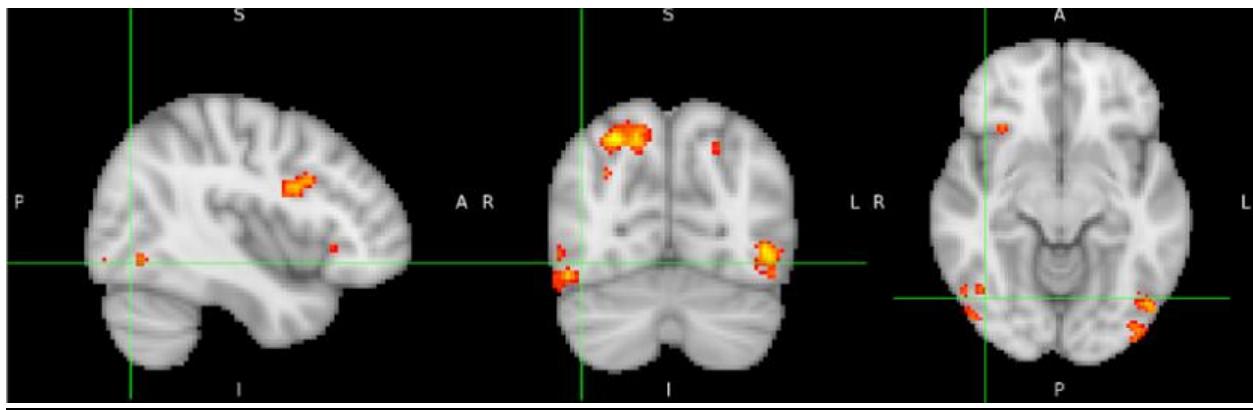




- **Cope 3 (Incongruent-Congruent):**

Activation is observed in subjects 1,8,21 while the rest seem to have no activated regions

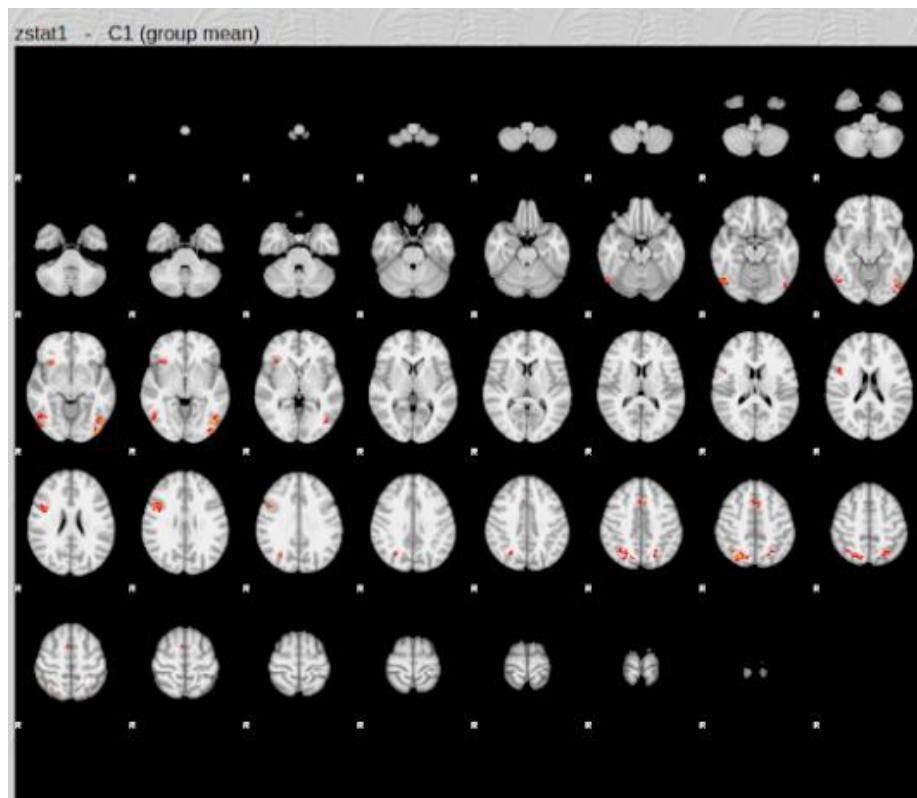




Exercises: → Done on Incongruent-Congruent

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?

- Cope 3 with threshold = Cluster

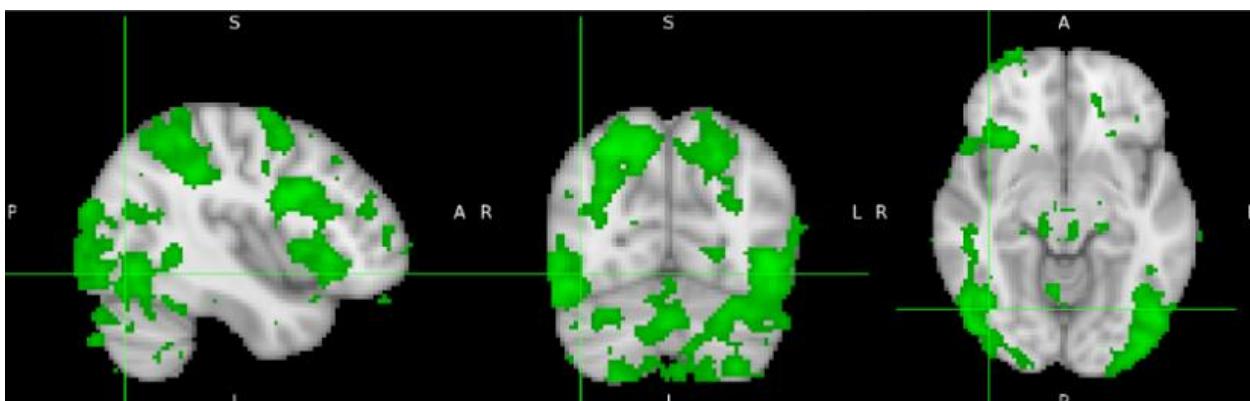
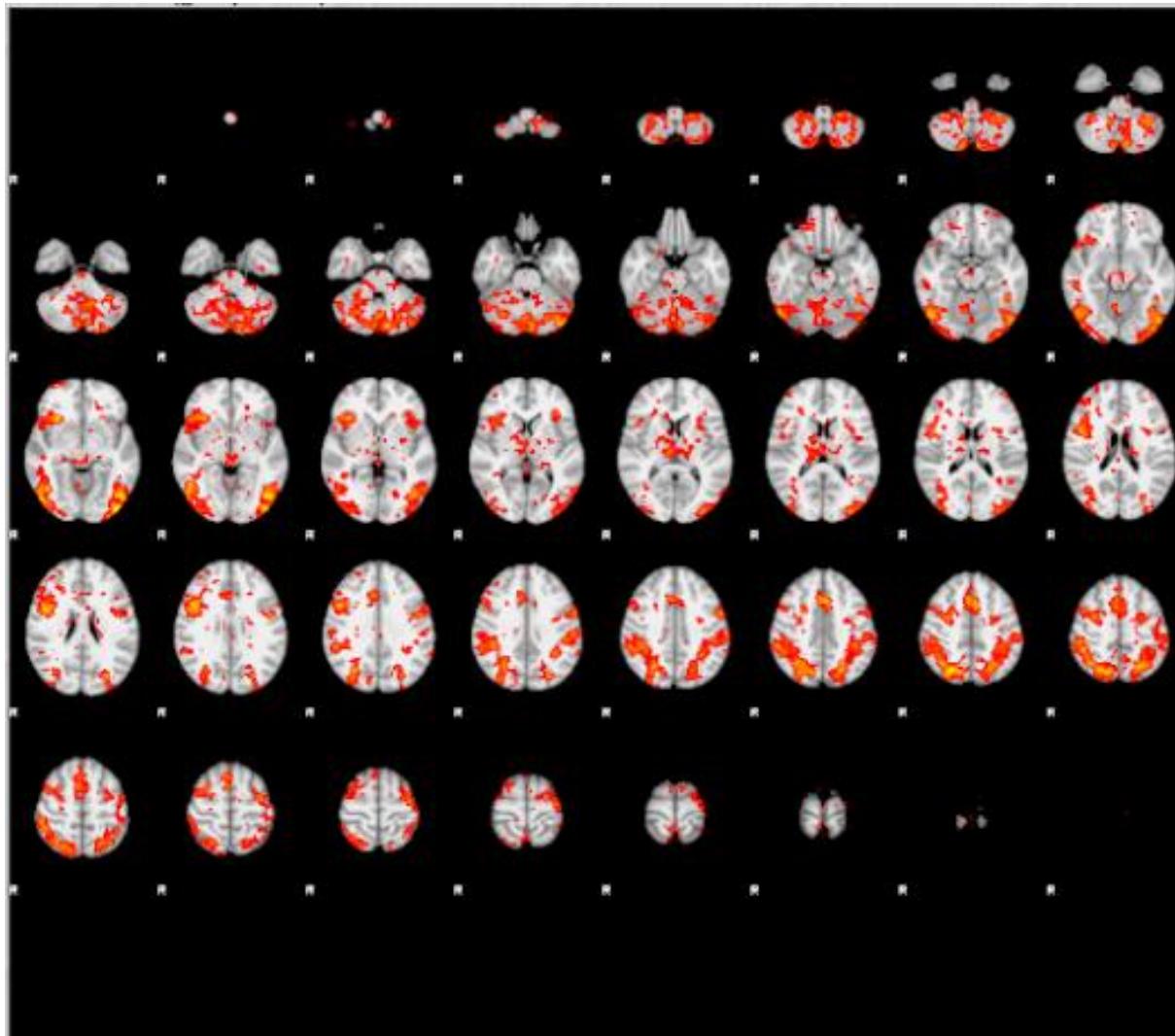


- Cope 3 with threshold = None

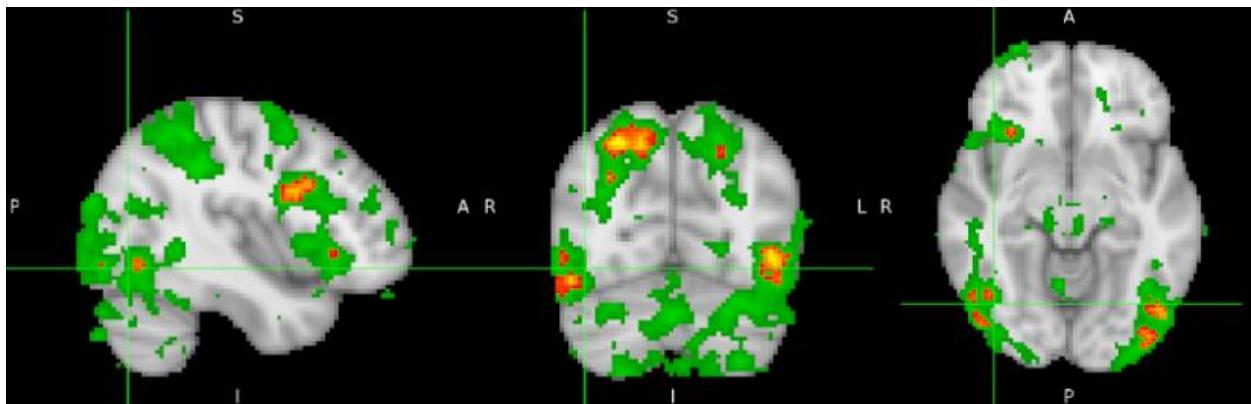
--→ Fails to yield results from the html file.

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.

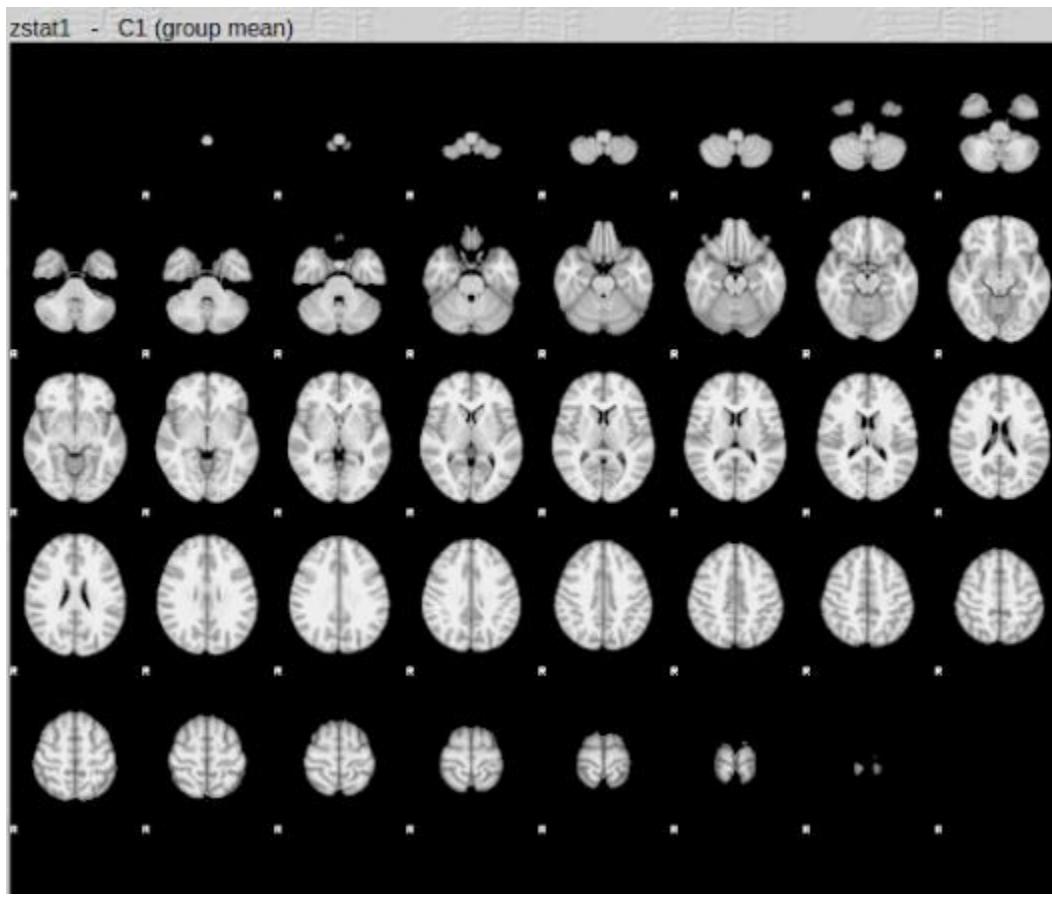
- Cope 3 with threshold = Uncorrected



Overlain with the cluster thresholded one:



- Cope 3 with threshold = Voxel





In FSL's 3rd level analysis (for group analysis), the different thresholding options (Cluster, None, Uncorrected, and Voxel) relate to how statistical significance is determined at the group level. Here's the breakdown of each:

1. Cluster Thresholding:

- Cluster thresholding is used to control for multiple comparisons by considering clusters of contiguous voxels that exceed a certain statistical threshold (e.g., p-value or Z-score).
- It incorporates both the **voxel-wise significance** (within the cluster) and the **extent** of the cluster (the size of the group of voxels).
- This method uses **cluster-level inference**, meaning it corrects for the spatial structure of the data and the potential for false positives due to large numbers of comparisons.
- Typically used for controlling family-wise error rate (FWER).

2. None:

- No thresholding is applied. This means all voxels with significant results (after the initial statistical analysis) are reported, regardless of the size of the cluster or the statistical significance.
- This option is often used for exploratory analyses or when you want to see all significant results, but you will need to be cautious of false positives.

3. Uncorrected:

- This thresholding option does not correct for multiple comparisons and only applies a basic voxel-wise threshold (e.g., $p < 0.001$ or $Z > 2.3$).
- It does not control for the family-wise error rate (FWER) or false discovery rate (FDR), so while it may identify significant voxels, it may include more false positives than corrected approaches.
- Typically used for identifying preliminary results, or when you are willing to accept a higher risk of false positives.

4. Voxel Thresholding:

- Voxel thresholding applies a statistical threshold (e.g., $p < 0.001$, $Z > 2.3$) to each voxel independently without considering the cluster structure.

- It is typically used to identify regions of the brain that show significant activity, but it doesn't account for the spatial extent of the activation (i.e., isolated voxels can be flagged as significant).
- Similar to uncorrected but can also involve more sophisticated voxel-wise thresholds depending on the specific settings you choose (e.g., for p-value or Z-score).

Observation from the processing results shown:

Uncorrected thresholding seems to have included more voxels. It didn't have strict methods and thus cause inaccurate results that contain errors. More voxels are highlighted now by the GLM model than there were in the cluster thresholding file.

For the voxel thresholding, it seems to have done the opposite. It used a very low threshold ($p < 0.001$) when actually the voxels actually activates might have p values higher. Thus, it ended up not highlighting any voxels and assuming non have been activated as non are below p-value of 0.001.

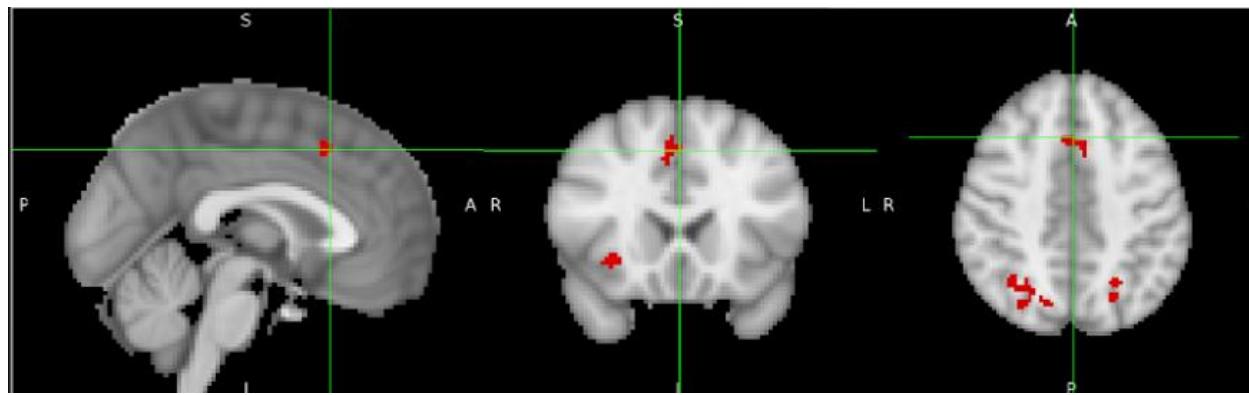
In my opinion I find that the cluster thresholding method is the most accurate.

Task 11: ROI Analysis

We've completed a whole-brain analysis showing where Incongruent and Congruent conditions differ significantly. This exploratory approach is useful when you lack specific hypotheses about effect locations.

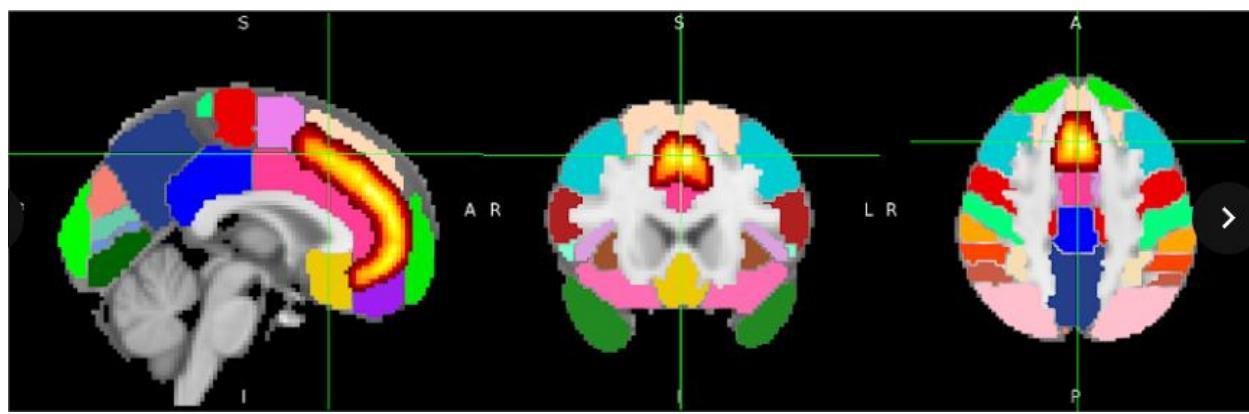
For well-studied domains like cognitive control, we can focus analyses on expected regions (e.g., dmPFC) - called ROI or confirmatory analysis. Whole-brain results alone can't reveal whether effects come from Incongruent increases, Congruent decreases, or both. ROI analysis is crucial for understanding what drives effects, especially in complex designs.

Let's look at the results of the 3rd Level Analysis for Cope3 (Incongruent-Congruent) for example:

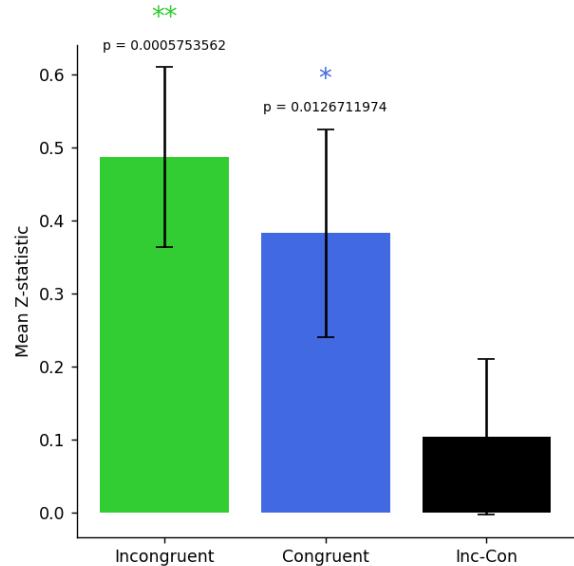


If we run an ROI analysis at the specified location, including all the functional regions showing in red, it will result in a p-value much larger than 0.05, meaning these aren't actually activated in Cope3. This is shown in the bar chart.

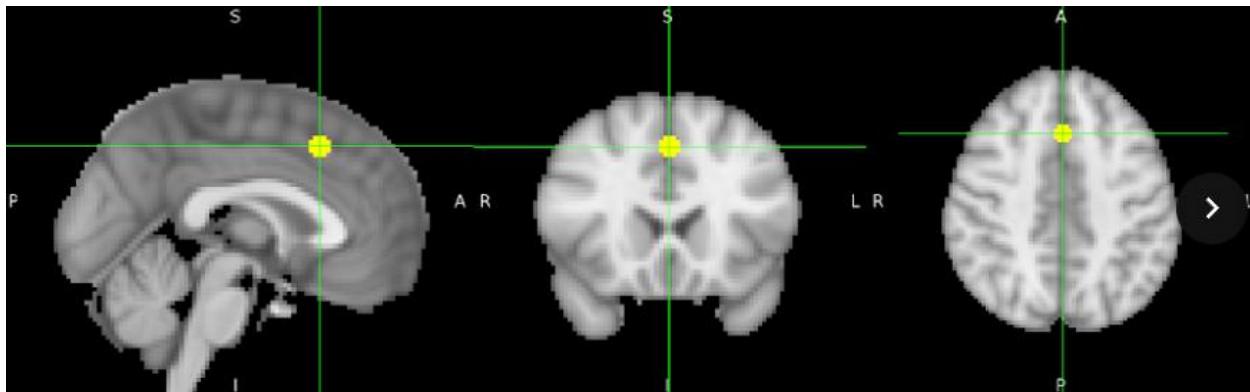
History Location	Coordinates: MNI152				Voxel location			
	0	-	+	45	-	+		
20	-	+		73	-	+		
44	-	+		58	-	+		



Inc-Con p-value exceeds 0.05



Now if we take a spherical mask around this voxel that has a radius of 5 voxels, this will set all other voxels to zero while only including the ones in the sphere. Now we are more focused on this area. Running an ROI analysis again for Cope 3, results are much better, showing p-values less than 0.05 for Inc-Con.

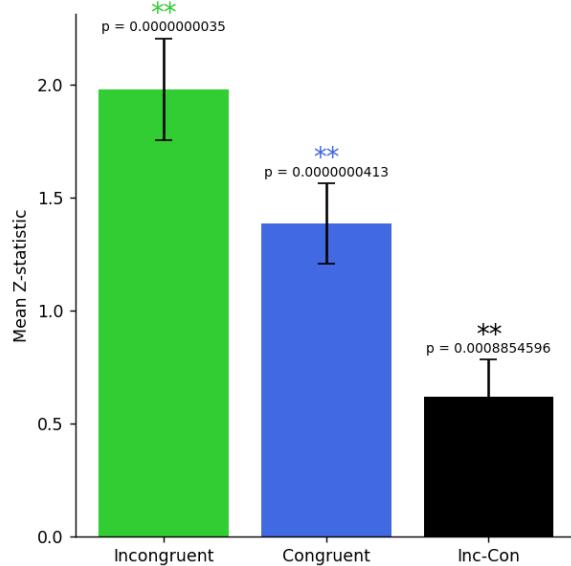


Region Activated: Paracingulate Gyrus

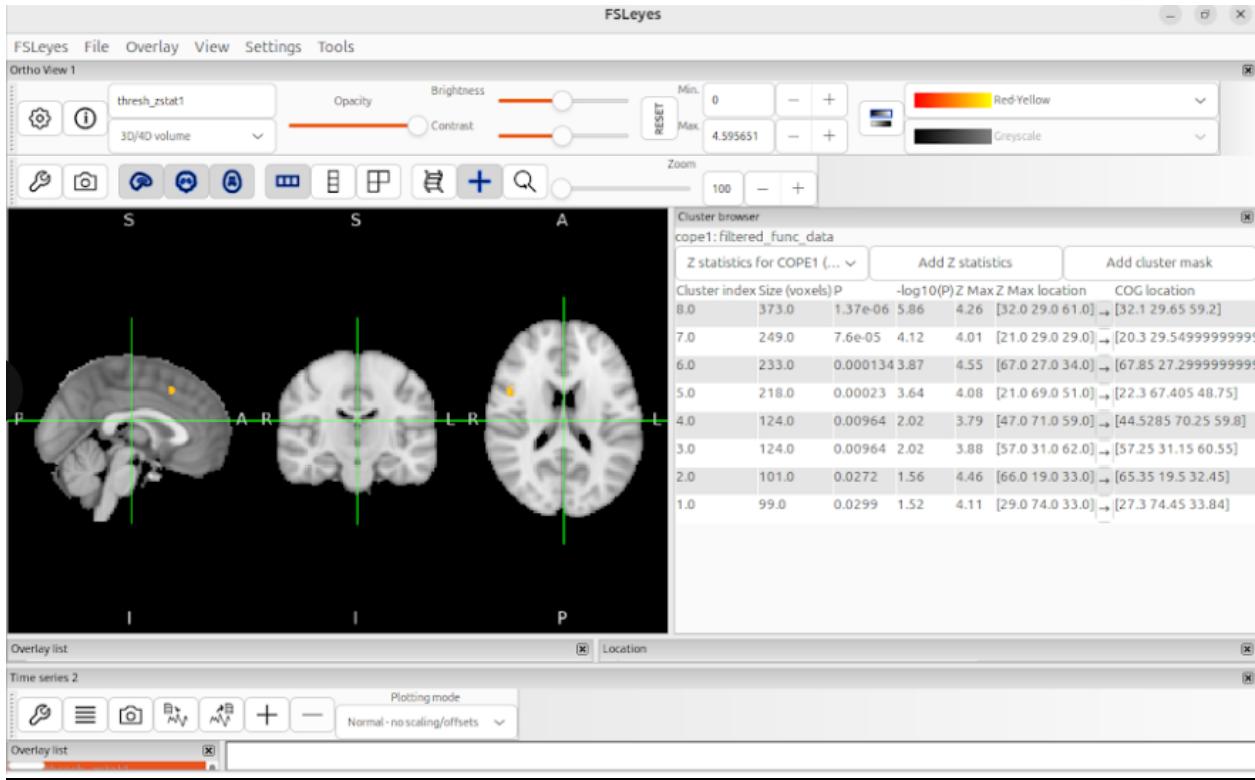
Linked with:

1. **Cognitive Role:** Key for decision-making, error monitoring, and mentalizing in the prefrontal cortex.
2. **Structural Variability:** Shows high anatomical variation between individuals and expanded evolution in humans.
3. **Clinical Link:** Implicated in schizophrenia, OCD, and depression due to functional disruptions.

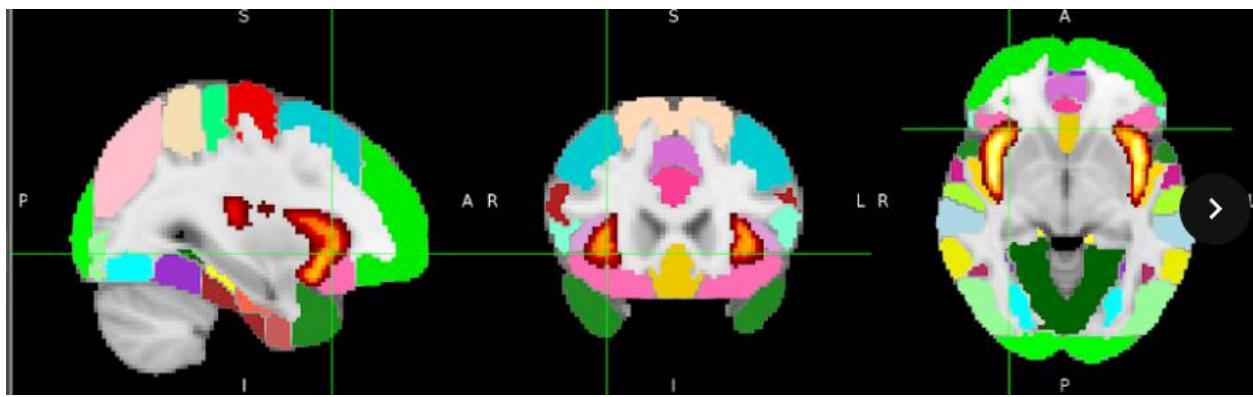
This region is most activated in Incongruent Tasks!



Now let's focus on the voxels showing maximum intensity (activation) in the 3rd Level analysis results for Cope 3. We have 8 max locations, i.e. 8 clusters we can focus on using a mask to identify in which task this region is most active in. The figure below shows the locations of the 8 maximum voxels.



Cluster 1:

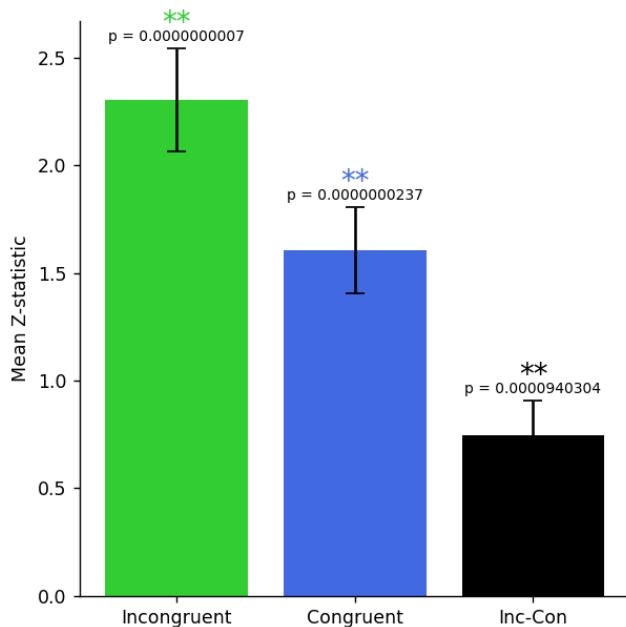
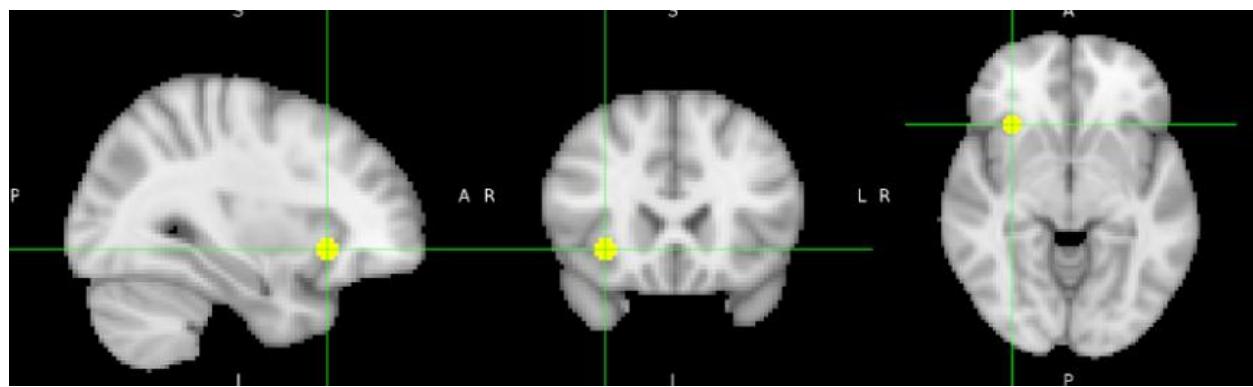


Region Activated: Insular Cortex

1. **Interoception Hub:** Processes bodily signals (pain, temperature, hunger) and self-awareness.
2. **Emotion & Cognition:** Links feelings to decision-making, empathy, and addiction pathways.
3. **Clinical Hotspot:** Implicated in anxiety, PTSD, and neurodegenerative disorders (e.g., Alzheimer's).

History Location	Location	Coordinates: MNI152		Voxel location	
		-	+	-	+
	32			29	- +
	22	-	+	74	- +
	-6	-	+	33	- +

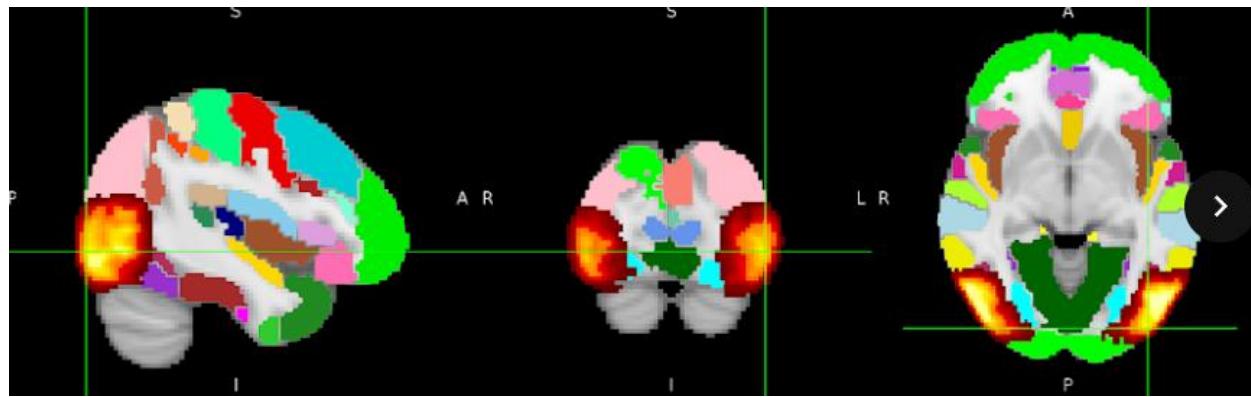
Mask taken:



Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

Cluster 2:

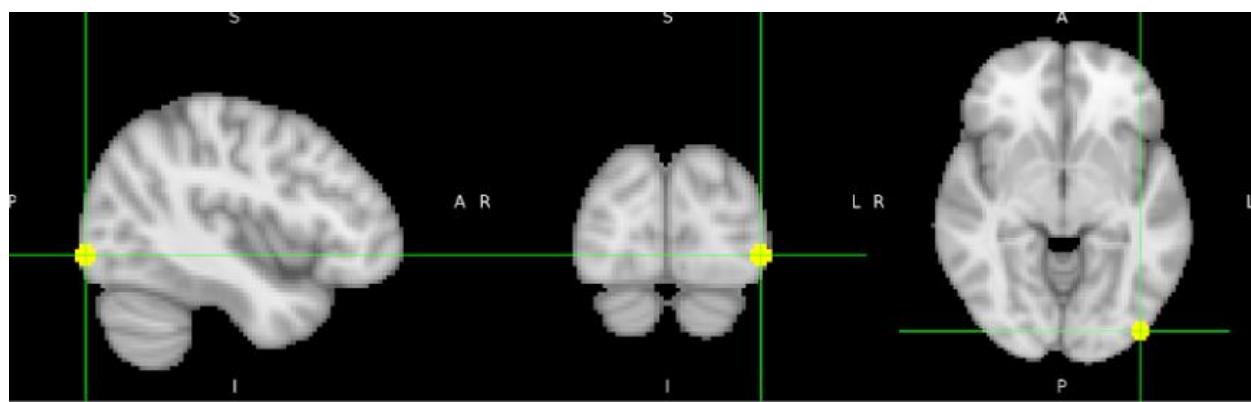


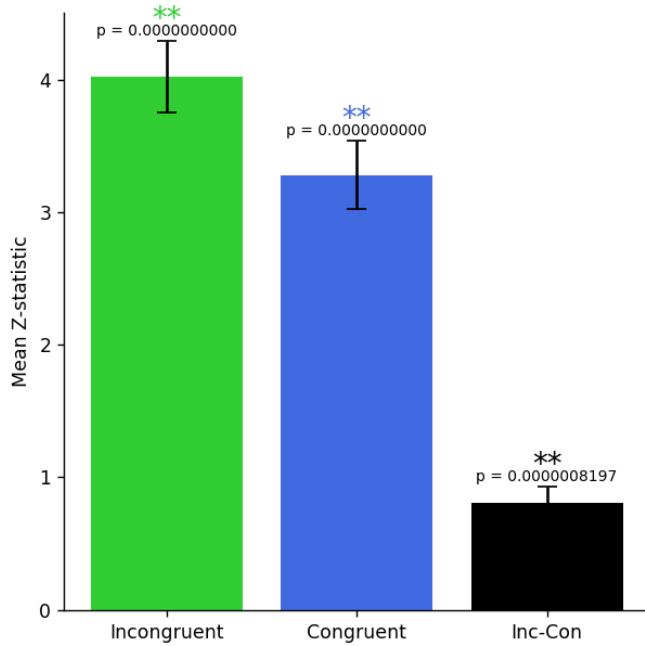
Region Activated: Lateral Occipital Cortex, Inferior Division

- Visual Processing:** Specialized in object recognition and shape perception within the ventral visual stream.
- Functional Connectivity:** Strong links to parietal and temporal regions for integrating visual-spatial information.
- Clinical Relevance:** Damage or dysfunction can cause visual agnosias (e.g., difficulty recognizing objects).

Location		Coordinates: MNI152		Voxel location							
History	Location	-42	-88	-6	66	19	33	-	+	-	+
		-	-	-	+	-	+	-	+	-	+
		-	-	-	+	-	+	-	+	-	+
		-	-	-	+	-	+	-	+	-	+

Mask taken:

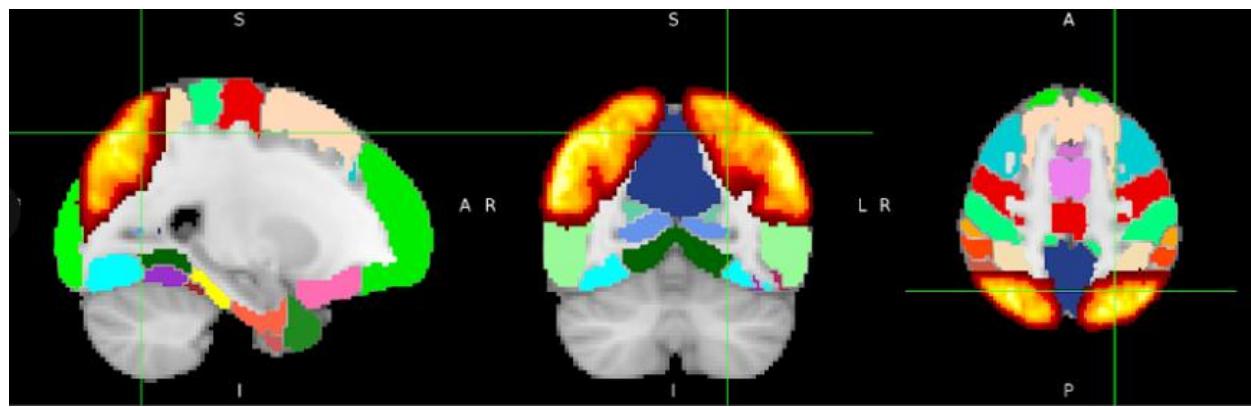




Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

Cluster 3:

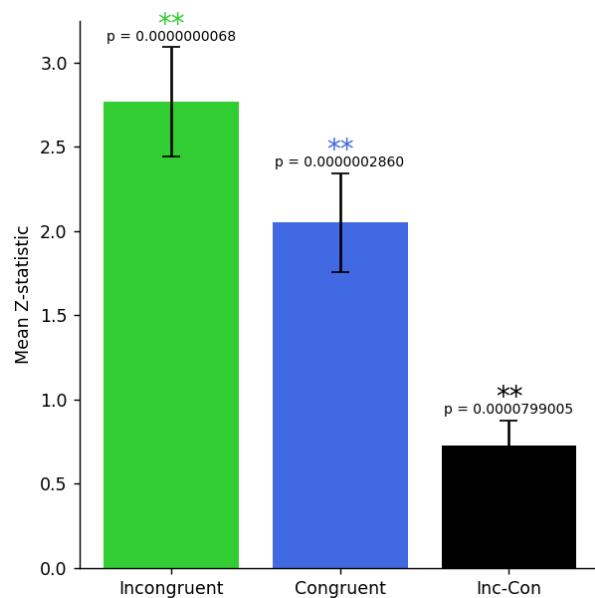
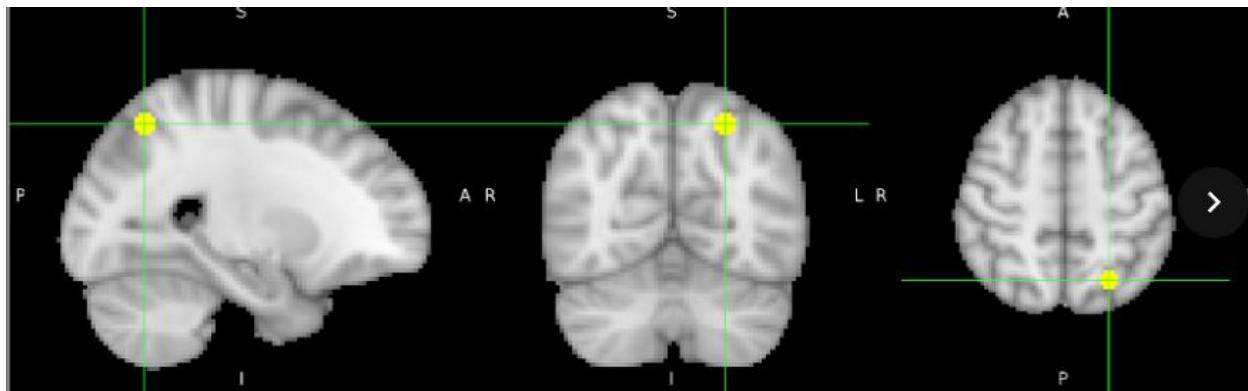


Region Activated: Lateral Occipital Cortex, Superior Division

- Visual Object Processing:** Involved in higher-level visual perception and 3D object recognition.
- Motion Integration:** Plays a role in combining shape and motion information for dynamic scene analysis.
- Neuroimaging Marker:** Altered activation patterns observed in visual processing disorders and autism spectrum conditions.

History Location	Location		Voxel location		
	Coordinates: MNI152				
-24			-	+	57
-64			-	+	31
52			-	+	62

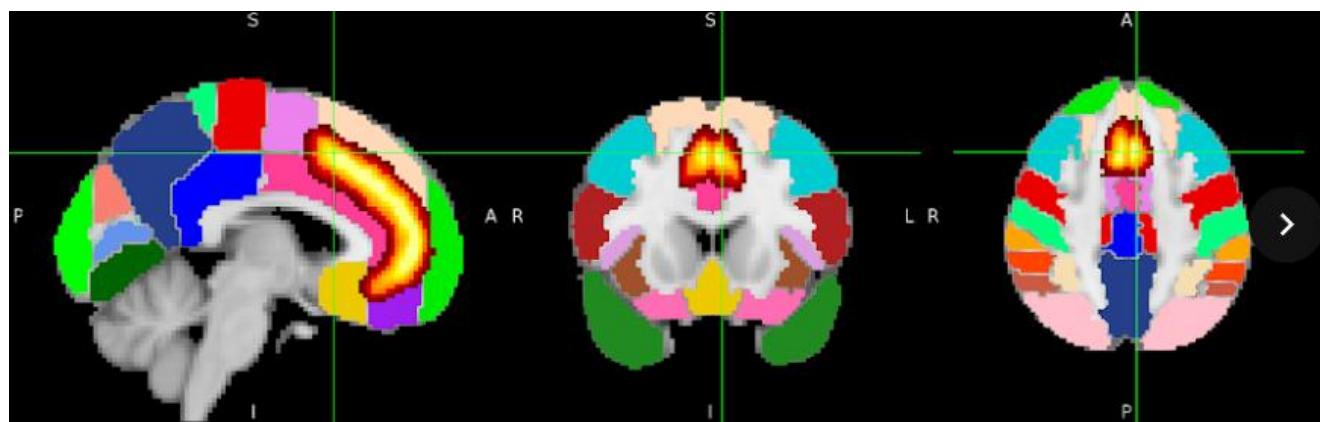
Mask taken:



Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

Cluster 4;

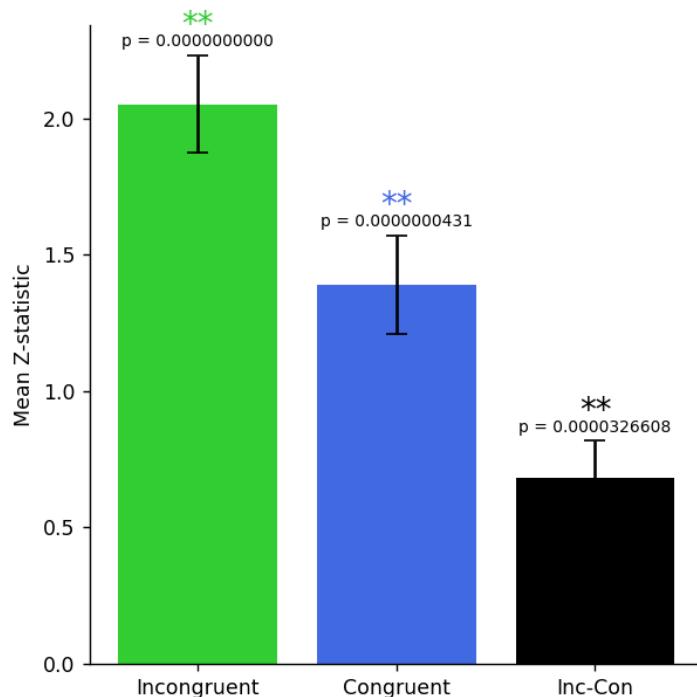
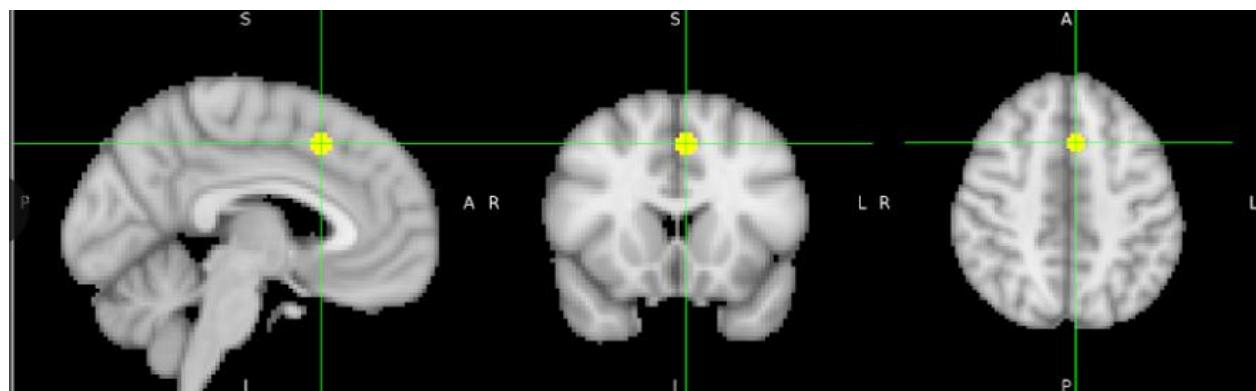


Region Activated: Paracingulate Gyrus

- Cognitive Control:** Critical for executive functions, conflict monitoring, and decision-making processes.
- Mentalizing Hub:** Key region for theory of mind (understanding others' intentions) and self-referential thought.
- Psychiatric Link:** Dysfunction associated with schizophrenia, OCD, and mood disorders due to its regulatory role.

Location	Coordinates: MNI152			Voxel location		
	-4	-	+	47	-	+
16	-	+		71	-	+
46	-	+		59	-	+
	...					

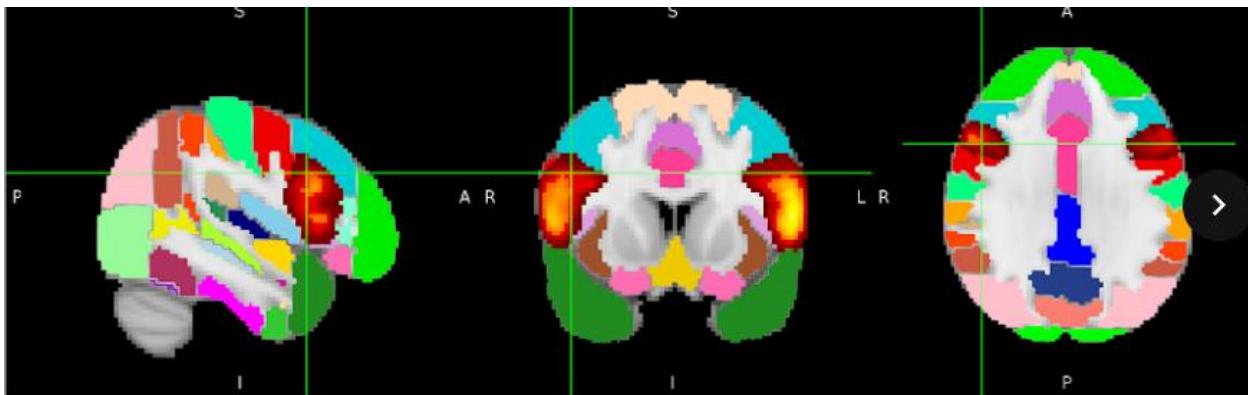
Mask:



Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

Cluster 5:

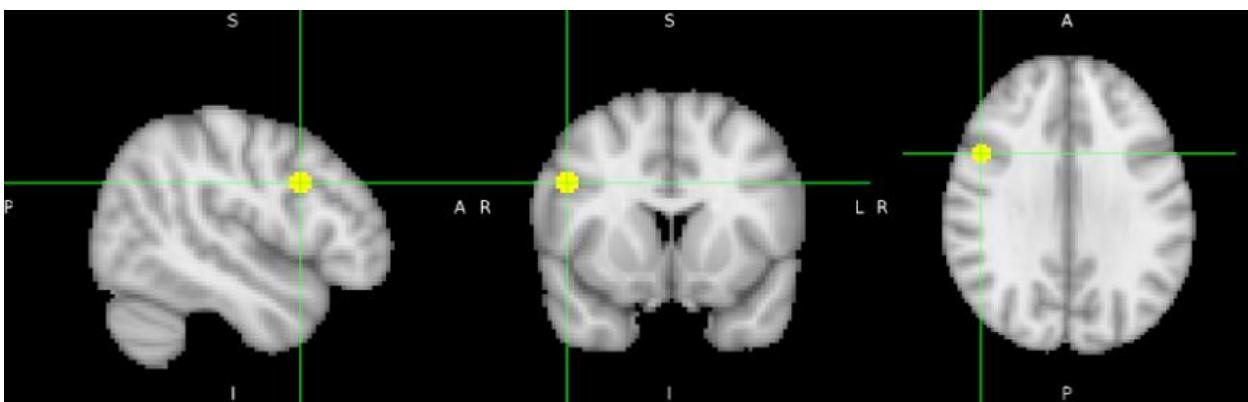


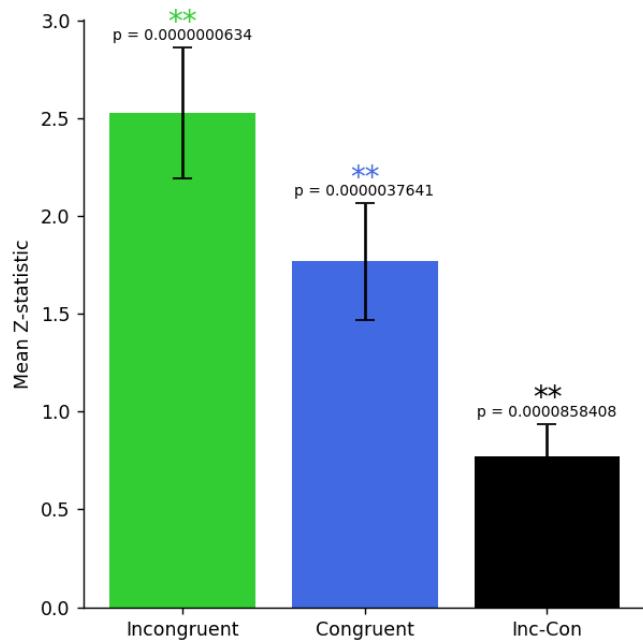
Region Activated: Inferior Frontal Gyrus

1. **Language & Speech:** Essential for language processing (Broca's area) and speech production.
2. **Cognitive Control:** Involved in inhibitory control, decision-making, and working memory.
3. **Clinical Impact:** Damage or dysfunction linked to aphasia, ADHD, and impulse control disorders.

Location	Coordinates: MNI152		Voxel location	
	48	12	21	69
History	-	+	-	+
Location	-	+	-	+
Memory	-	+	51	-

Mask:

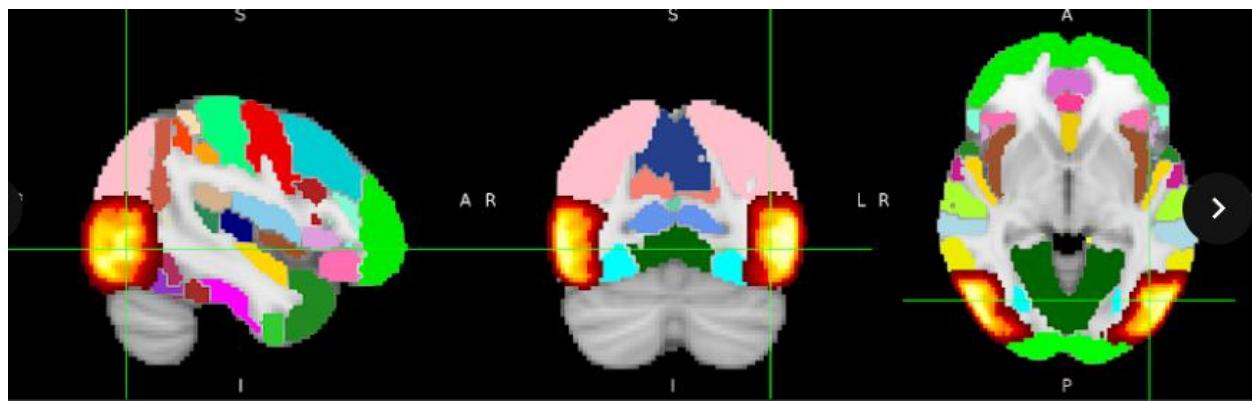




Results:

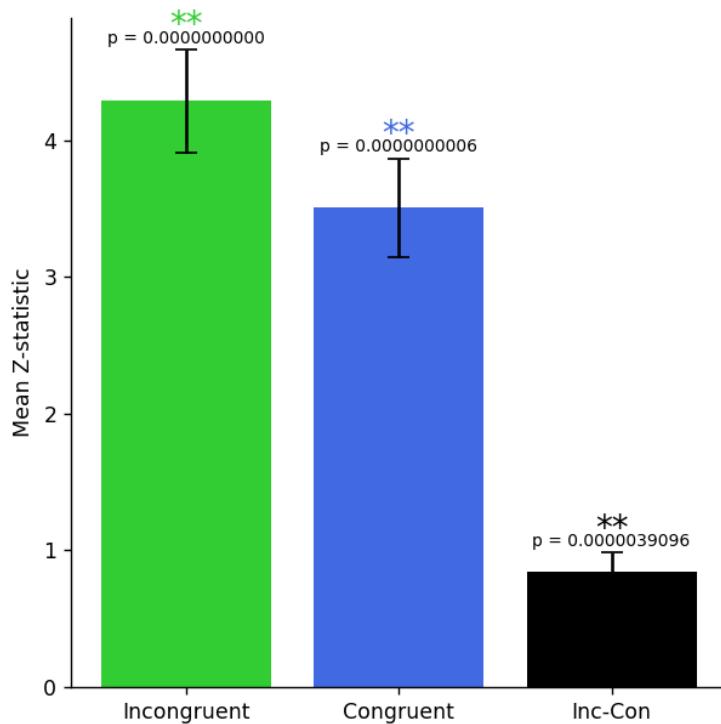
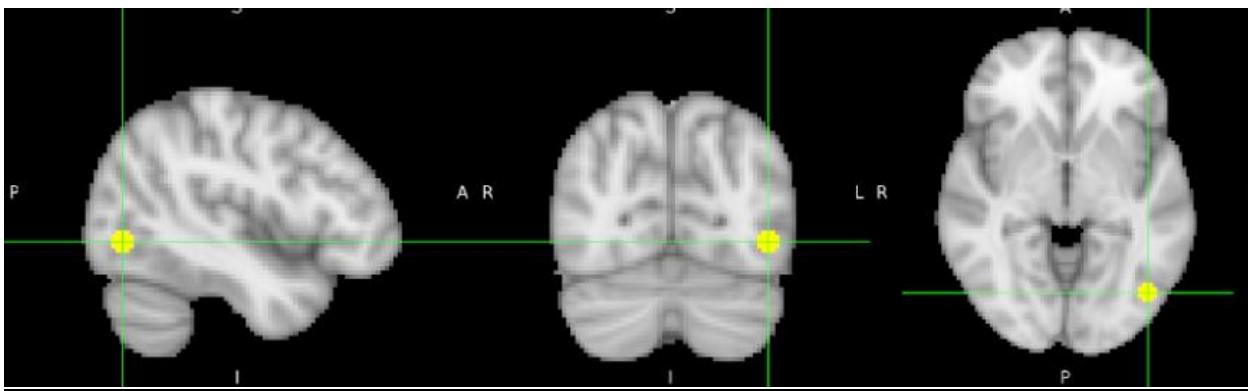
We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

Cluster 6:



Region Activated: Lateral Occipital Cortex,
Inferior Division

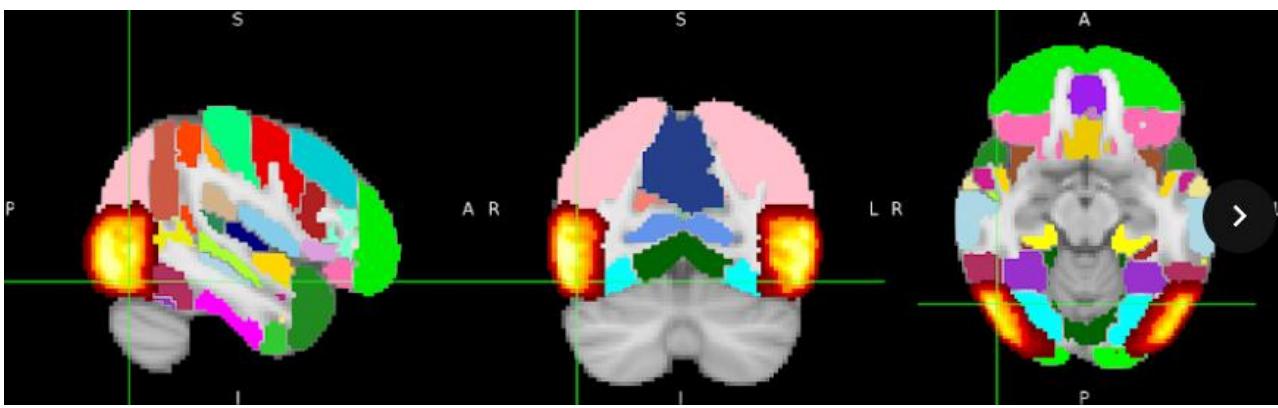
History Location	Location		Voxel location	
	Coordinates: MNI152			
-44	-	+	67	-
-72	-	+	27	-
-4	-	+	34	-
..



Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

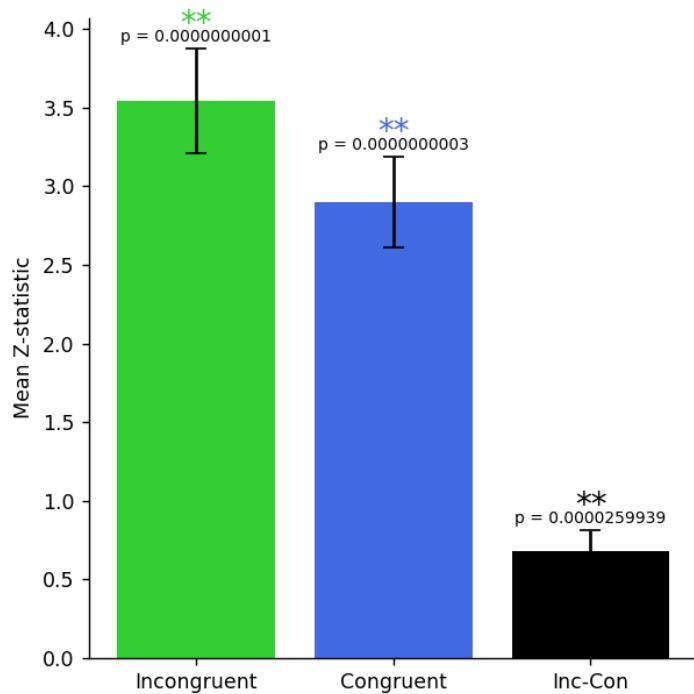
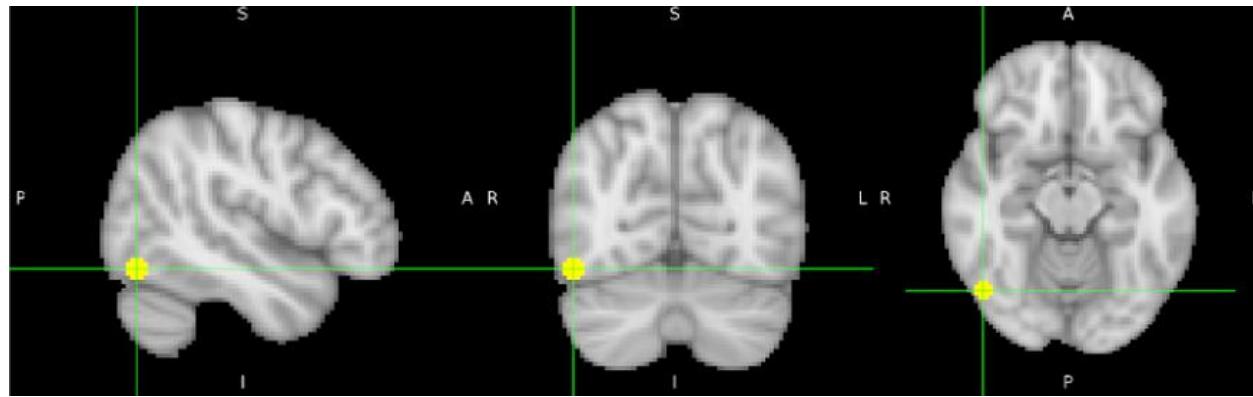
Cluster 7:



Region Activated: Lateral Occipital Cortex,
Inferior Division

Location	Coordinates: MNI152	Voxel location			
		-	+	-	+
48		-	+	21	-
-68		-	+	29	-
-14		-	+	29	+

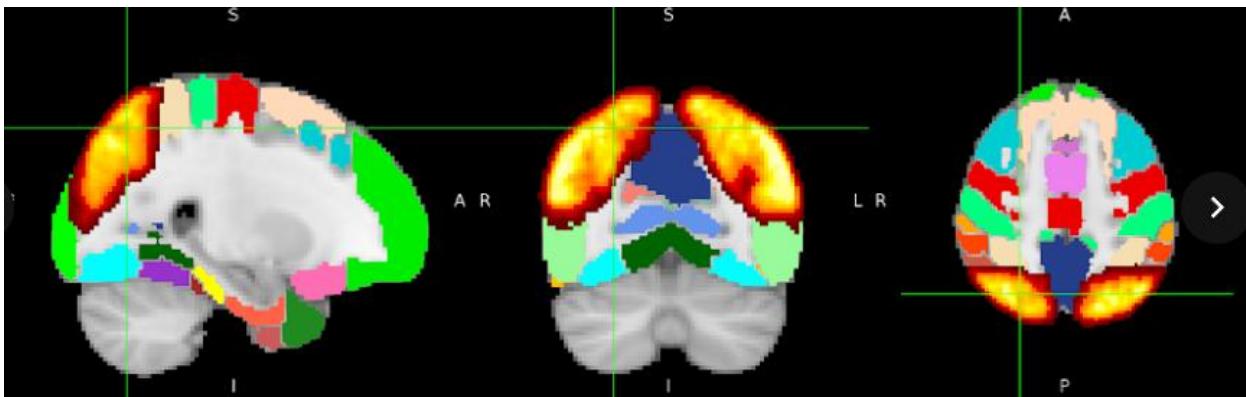
Mask:



Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

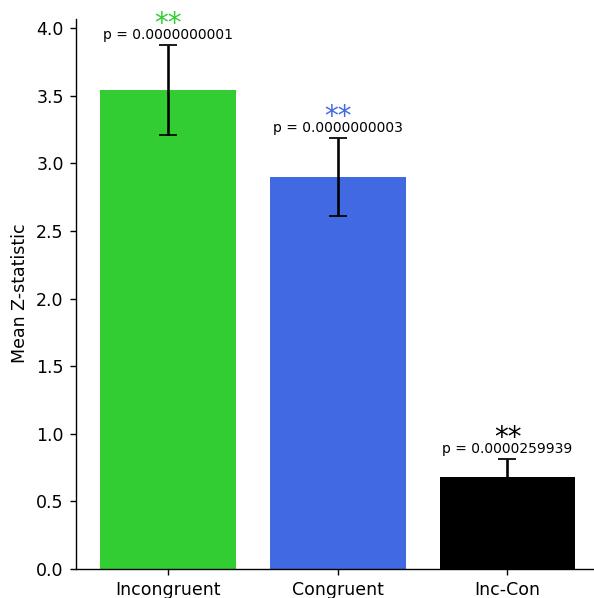
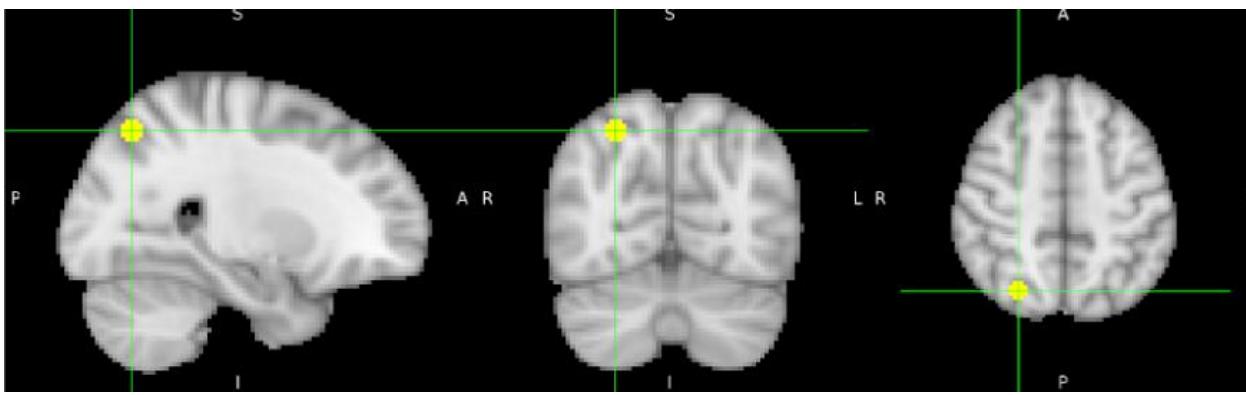
Cluster 8:



Region Activated: Lateral Occipital Cortex,
Superior Division

History Location	Coordinates: MNI152	Voxel location				
		-	+	-	+	
	26			32	-	+
	-68			29	-	+
	50			61	-	+

Mask:



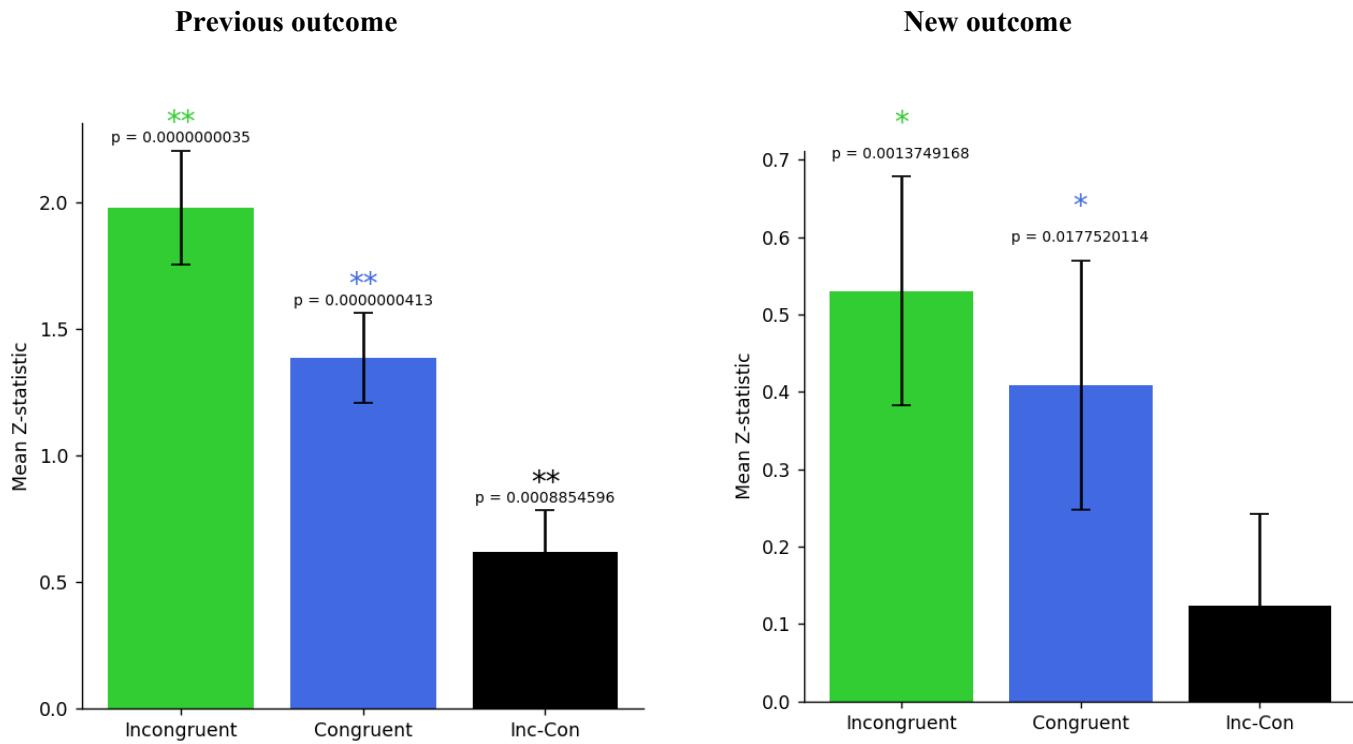
Results:

We can tell by the p-values that this cluster is most activated in Incongruent tasks as it has the lowest value.

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 what you would expect?

	Previous numbers	New numbers
1	Incongruent,Congruent,Inc-Con	1 Incongruent,Congruent,Inc-Con
2	1.863240,0.278328,1.639252	2 0.602768,-0.974578,1.532270
3	3.118286,2.289443,0.741809	3 1.159899,0.655569,0.427623
4	1.668114,0.481215,1.075685	4 0.364964,-0.801998,1.043110
5	1.828725,1.327740,0.412434	5 0.613565,0.781490,-0.177120
6	4.212729,3.566904,0.989408	6 1.673029,1.837627,0.036641
7	2.510646,0.855227,1.637930	7 -0.381951,-0.547032,0.198520
8	3.018982,1.323991,1.868548	8 0.321201,-0.481906,0.893546
9	2.917934,0.497867,2.311128	9 1.240840,0.254784,0.963207
10	1.246839,1.493219,-0.221740	10 0.536563,0.284575,0.259173
11	1.719434,1.976008,-0.243948	11 0.966487,0.969218,-0.006258
12	2.984716,2.874866,0.159991	12 1.279668,1.605688,-0.316923
13	3.300570,2.053051,1.410235	13 1.461084,1.142643,0.417749
14	1.027082,1.306486,-0.302093	14 0.574460,0.723879,-0.154946
15	1.401403,1.442943,-0.057510	15 -0.863215,-0.484436,-0.342898
16	3.857215,3.187680,0.767032	16 1.579159,1.699518,-0.111407
17	0.199767,-0.243863,0.406651	17 -0.489790,-0.097358,-0.428489
18	2.867935,1.832176,1.146441	18 0.789830,0.561985,0.260987
19	0.466228,1.209101,-0.650117	19 -1.497704,-0.400657,-1.160258
20	0.595518,1.302250,-0.620734	20 0.889497,1.468274,-0.541683
21	1.437114,1.121689,0.402211	21 0.236533,0.384713,-0.145458
22	2.610784,1.872697,0.547149	22 1.206225,1.195927,-0.006549
23	1.234587,0.143839,1.018010	23 0.333635,-0.169779,0.420421
24	2.733579,1.139664,1.459229	24 0.400239,-0.788322,1.050165
25	0.729099,0.598753,0.112053	25 0.268145,0.779172,-0.486959
26	-0.051981,0.892869,-0.849388	26 -0.033478,0.595585,-0.581711
27	1.962108,1.151631,0.889456	27 0.561865,0.425080,0.158486



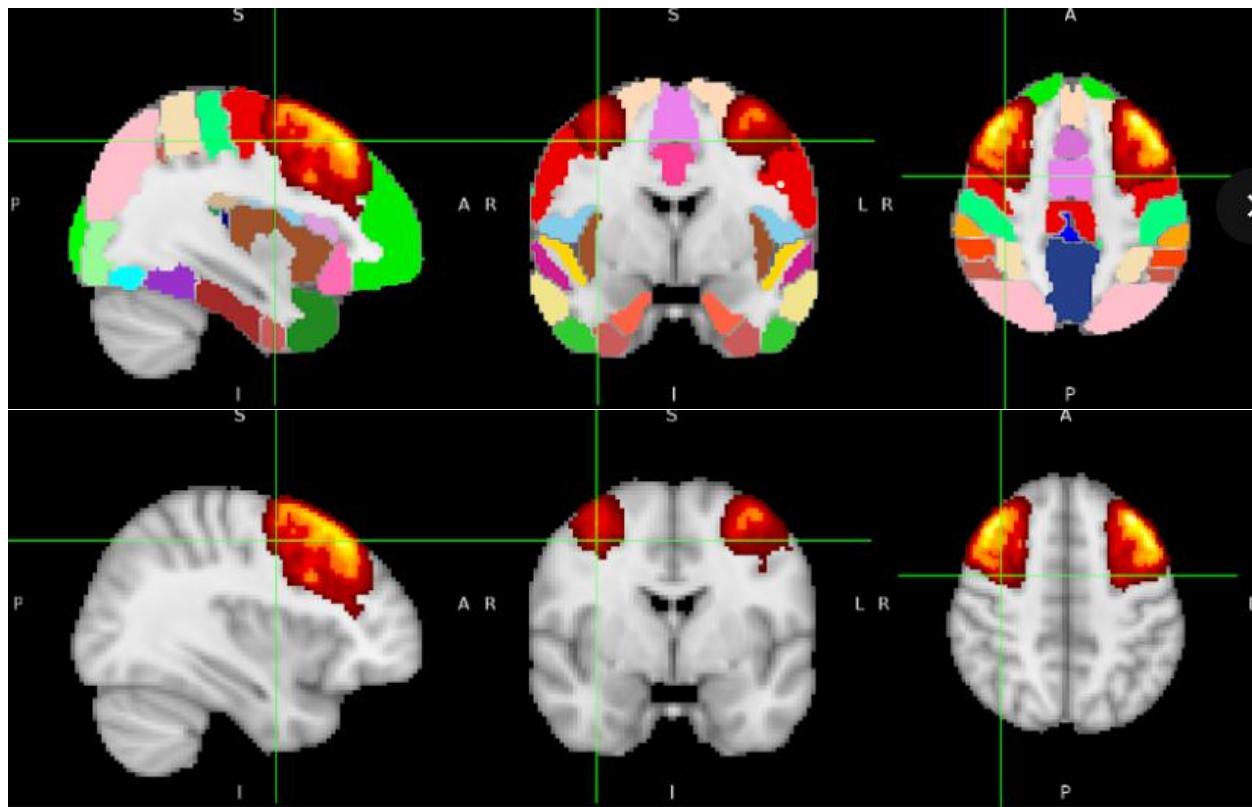
Observation:

P-values have noticeably increased, implying less strength of activation. The region was at first highly active in both Incongruent and Congruent task with p-values less than 0.001. Now they are less active, having p-values higher than 0.001 but still less than 0.05. On the other hand the Inc-Con p-value increased over the 0.05 limit of being counted as activated. Yes I did expect that result as its less probabilistic.

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.

Region accessed: Middle Frontal Gyrus

Location		Voxel location					
Coordinates: MNI152							
History	Location	36	-	+	27	-	+
		-2	-	+	62	-	+
		48	-	+	60	-	+



Commands used:

```
fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1
-roi 27 1 62 1 60 1 0 1 Jahn_ROI_dmPFC_EX2.nii.gz -odt float

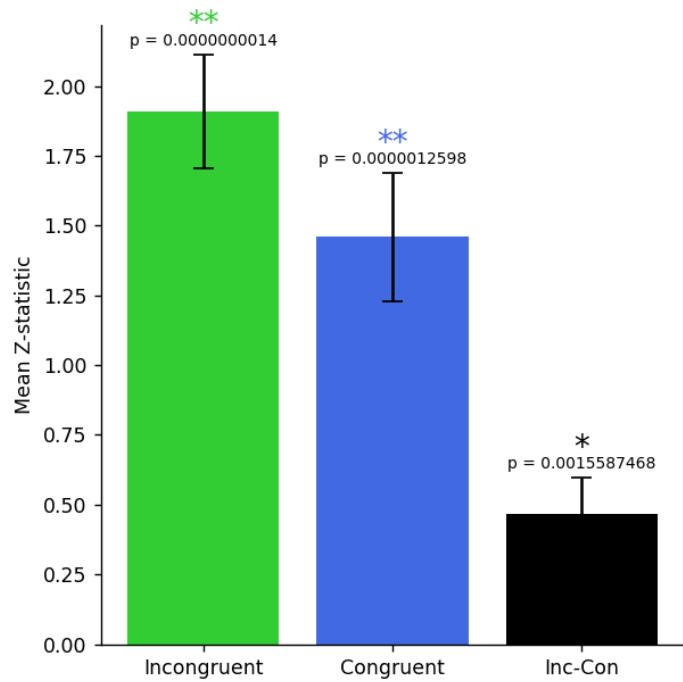
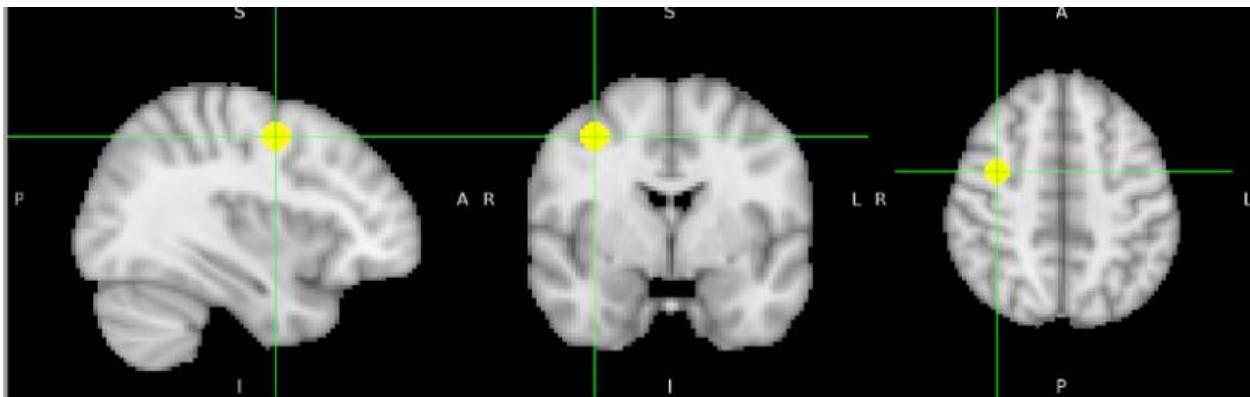
fslmaths Jahn_ROI_dmPFC_EX2.nii.gz -kernel sphere 7 -fmean
Jahn_Sphere_dmPFC_EX2.nii.gz -odt float

fslmaths Jahn_Sphere_dmPFC_EX2.nii.gz -bin
Jahn_Sphere_bin_dmPFC_EX2.nii.gz

//get new 26 numbers and get p value

fslmeants -i allZstats.nii.gz -m Jahn_Sphere_bin_dmPFC_EX2.nii.gz
```

Mask created:



Results show that this region is the most activated in an Incongruent task as its z-statistics yield the lowest result.

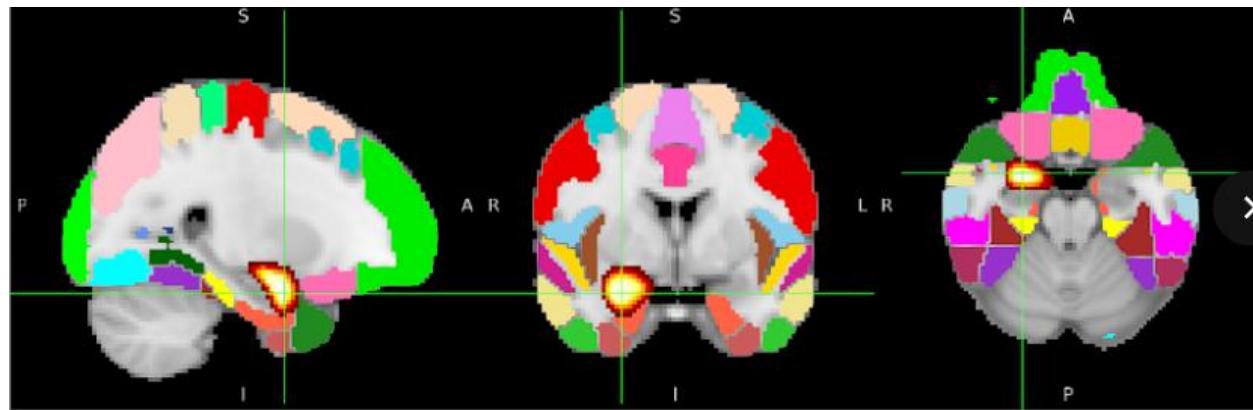
This region is highly active in both Incongruent and congruent as p-values is less than 0.001.

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).

Region: Amygdala

Found at coordinates:

Location	Coordinates: MNI152			Voxel location		
	26	-	+	32	-	+
History	Location					
	0	-	+	63	-	+
	-22	-	+	25	-	+



Commands used:

```
fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1
-roi 32 1 63 1 25 1 0 1 Jahn_ROI_dmPFC_EX3.nii.gz -odt float

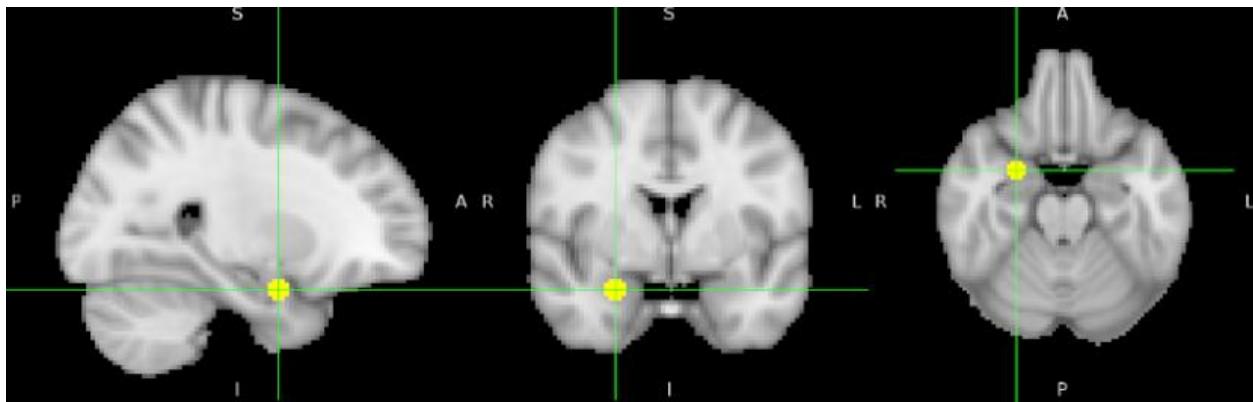
fslmaths Jahn_ROI_dmPFC_EX3.nii.gz -kernel sphere 5 -fmean
Jahn_Sphere_dmPFC_EX3.nii.gz -odt float

fslmaths Jahn_Sphere_dmPFC_EX3.nii.gz -bin
Jahn_Sphere_bin_dmPFC_EX3.nii.gz

//get new 26 numbers and get p value

fslmeants -i allZstats.nii.gz -m Jahn_Sphere_bin_dmPFC_EX3.nii.gz
```

Mask created:



Results: Incongruent

0.705747
0.476646
0.451510
1.241186
0.628021
-0.091458
-0.552552
0.741823
-0.744510
0.339546
-0.598860
0.573514
0.638146
-0.101584
3.329617
0.189988
0.591131
-1.807601
0.652986
0.881888
2.344716
-1.253398
0.565266
0.624003
-0.340301
0.436135

The Flanker Task: An fMRI Study

Abstract

The human mind has always been an exceptional and mesmerising area of study. For years scientists have been trying to study how the brain functions and reacts to different tasks, and how certain tasks lead to brain evolution or even deterioration. For our study, we were interested to find out if the same regions in the brain will be activated when doing similar tasks but with 2 different levels of difficulty.

Literature Review

The Flanker task, introduced by Eriksen and Eriksen (1974), is a classic paradigm used to assess attentional control and response inhibition. Participants respond to a central target while ignoring surrounding distractors (flankers), which may be congruent or incongruent. Incongruent trials typically produce slower reaction times and lower accuracy, highlighting the Flanker effect—a measure of interference from conflicting information. This task is widely used in studies of executive function, development, and clinical populations such as ADHD. Neuroimaging research links Flanker task performance to activity in the anterior cingulate cortex, a key region in conflict monitoring.

Methodologies

Our dataset consists of 26 subjects with 2 runs each. Throughout the course of the study, we have used the dataset obtained from the experiment in order to reach a final verdict of which exact regions in the brain are activated at which task. To reach that verdict we conducted multiple processing steps starting from preprocessing, 1st level analysis, higher level analysis and ROI analysis. All steps have been conducted on the FSL tool and observations have been thoroughly documented in a report.

Preprocessing

Our work starts with preprocessing the dataset we obtained from the flanker task. However, before conducting this step, we performed a quality control task. In this task we visually scanned each subject's brain volumes for both runs and looked for any sudden motions or blurs in the images that could infer the data contains motion artifacts. Because the data is very high quality. We didn't find any significant signs of motion artifacts.

Next, we used FSL for preprocessing. This step involves brain extraction to focus on only brain

matter, motion correction to correct for any minor motion artifacts, smoothing to decrease noise in functional images, and slice timing correction to account for delays in volume acquisition.

First Level Analysis

Next, we conducted a first level analysis. Our main goal is to estimate the average amplitude of the BOLD response for each condition, so we create a General Linear Model (GLM) to be able to map the responses for both conditions. Our outcome is from this step is brain image files that represent the regions activated in both the congruent and incongruent task. We also obtain one for the result of the difference of both.

Higher Level Analysis

This consists of 2 levels. The second level analysis is where we average the results of the first level analysis of both runs for each subject. Then the third level analysis where our aim is to generalize the result to the population. For this step we use all the 26 results from the previous step, and use an averaging method called fixed effects. This method accounts for changes within and between subject variability. At this point, we have one brain image file for each task – incongruent, congruent and their difference – that generalize the results of the 52 runs we have and that could be generalized to the population.

ROI Analysis

From the previous analysis, we had data of where the peak activated cluster locations in the brain are. In ROI (region of interest) analysis, we focus on these clusters by using spherical masks with a radius of 5. We then extract the z-statistics of these regions for both tasks to determine which is more significant based on the p-value. So, a brain cluster could have a smaller p-value for an incongruent task, meaning it was more activated in this task. This is our very final step in order to finalize our verdict.

Results

We have concluded that these regions where the most active in the brain during the flanker task: Paracingulate Gyrus, Insular Cortex, Lateral Occipital Cortex, both superior and inferior division, Inferior Frontal Gyrus. Activation was during both congruent and incongruent tasks but higher in the incongruent ones.