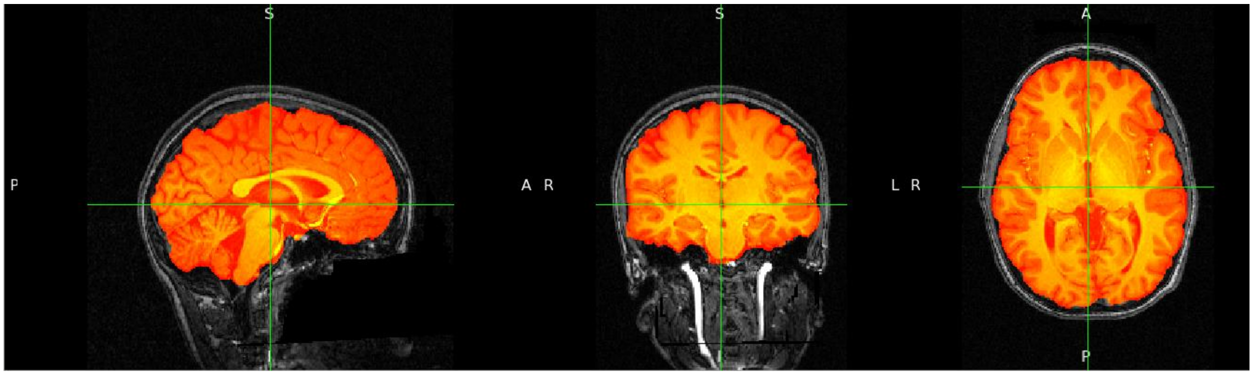
Flanker Dataset Report

Name: Rawda Yousry

Contents:

First Level Analysis:
First, we will do skull stripping the suitable threshold for the first subject is 0.5.

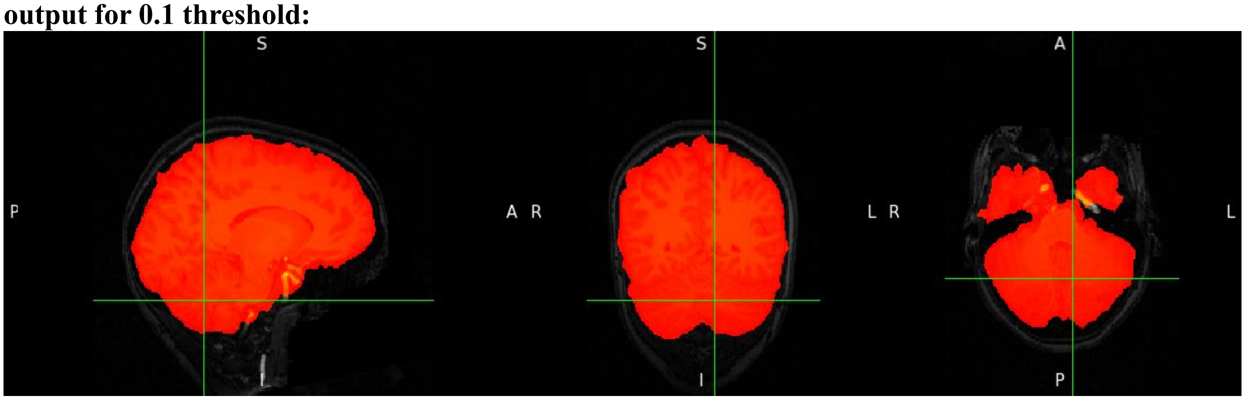
Then the output will be:



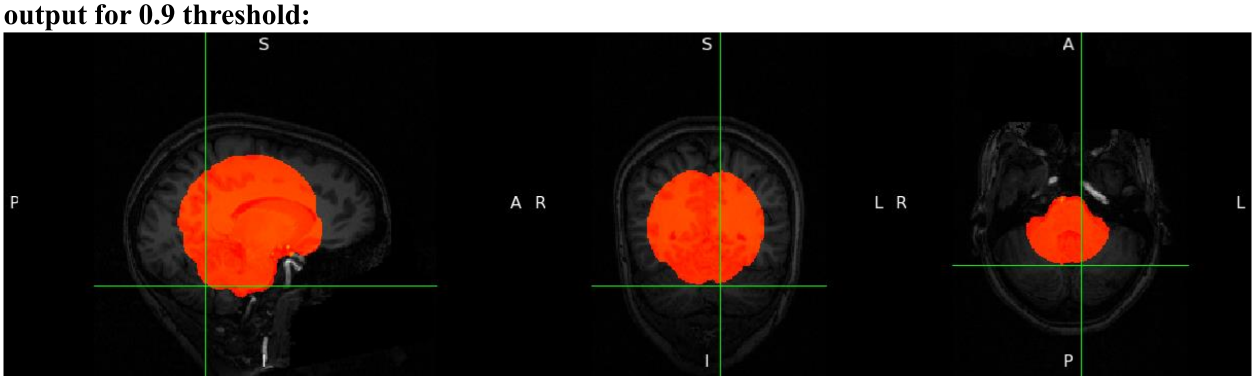
So, the brain is the orange region, and it is extracted successfully.

Exercise on brain extraction:
I will do skull stripping with 0.1 and 0.9 fractional intensity to see what will happen,
I expect as threshold intensity increases, may more of the brain will be stripped which is something that not needed

output for 0.1 threshold:



output for 0.9 threshold:



For sure with threshold 0.1 is much better as I will have more information about the brain.

Secondly, for the prestatistics.

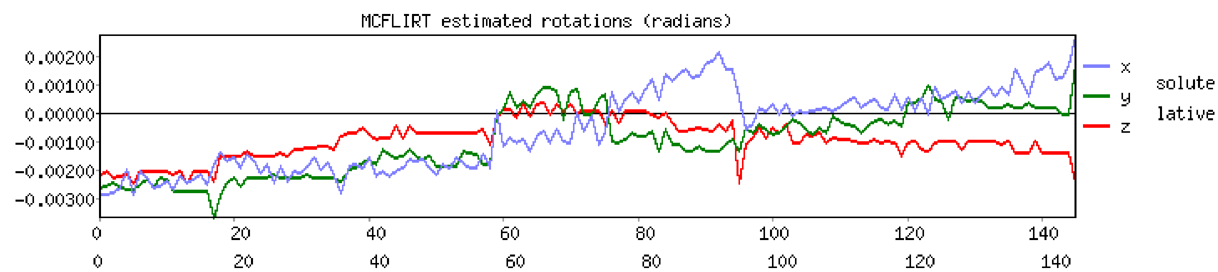
Exercise on quality control of all subjects

	Quality Control		
Subjects	**T1**	**Functional**	
		Run 1	Run 2
sub_01	good	good	good
sub_03	aliasing	+	+
sub_04	good	good	good
sub_05	good	good	good
sub_06	aliasing	good	++
sub_07	good	good	good
sub_08	good	good	good
sub_09	aliasing	good	good
sub_10	good	good	good
sub_11	aliasing	good	+
sub_12	good	good	good
sub_13	good	good	good
sub_14	artifact	good	+
sub_15	aliasing	+	good
sub_16	aliasing	good	good
sub_17	zipper	+	++
sub_18	good	+	+
sub_19	zipper	++	++
sub_20	aliasing	good	good
sub_21	good	++	+
sub_22	aliasing	+	++
sub_23	good	good	good
sub_24	good	good	good
sub_25	motion	++	+++
sub_26	good	good	good

Results of motion correction of subject 1 run 1:

We will do motion correction for subject. If there is any motion, it will be corrected by doing inverse for any rotation or translation.

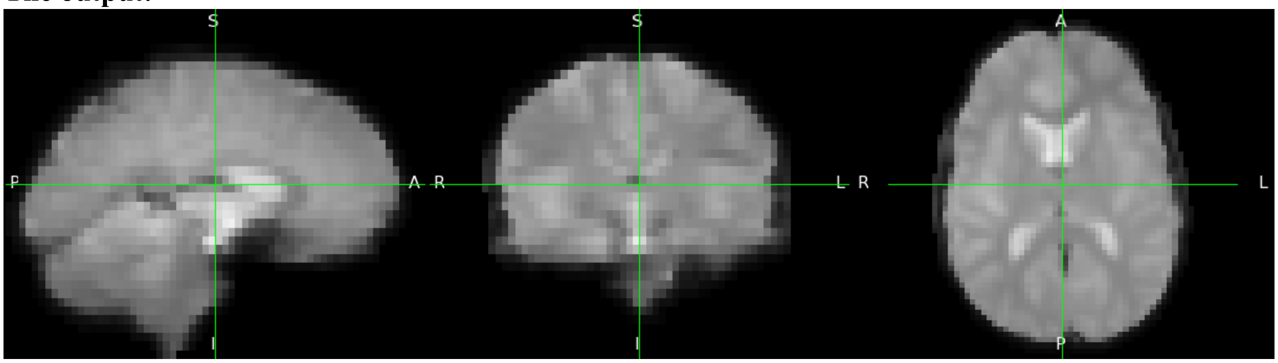
MCFLIRT Motion correction
Mean displacements: absolute=0.18mm, relative=0.05mm



Then, Smoothing:

We will do smoothing using 5mm kernel.
The smoothing kernel will average the signal intensity of each voxel with its neighboring voxels within a 5mm radius. This process helps to smooth out noise.

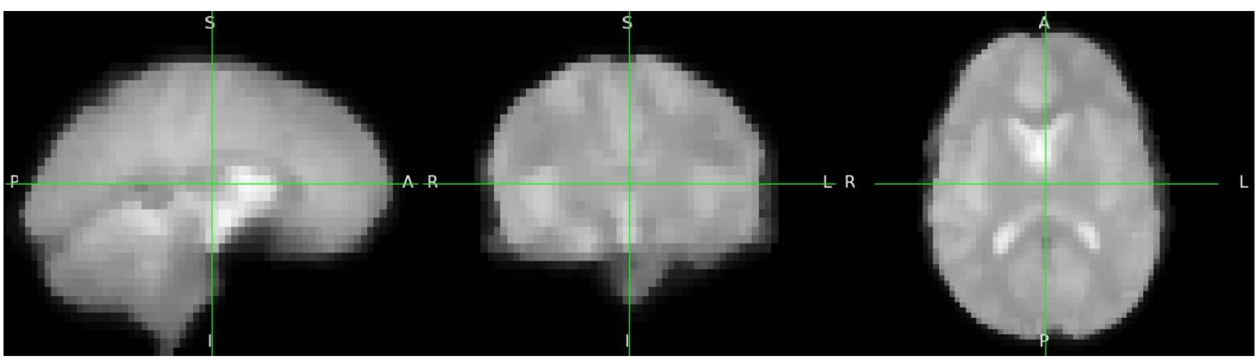
The output:



So, This step will not be important if the region is too small.

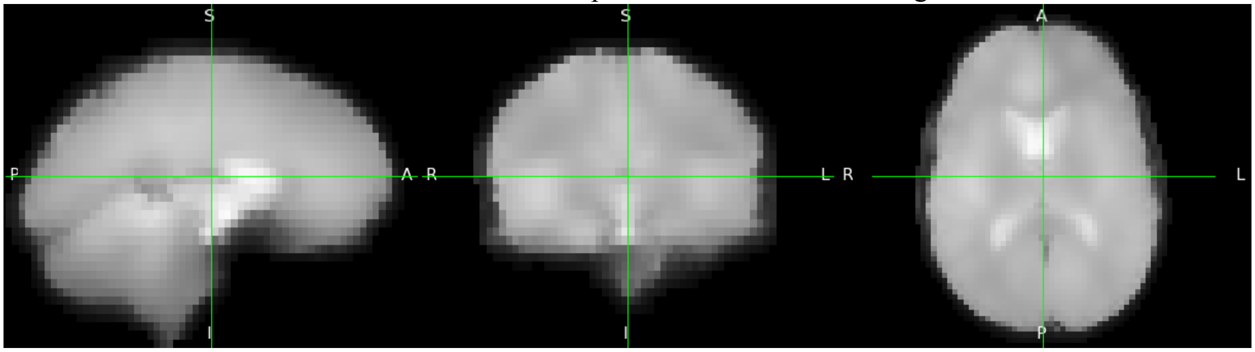
Some exercice on smoothing:

1. 3mm Smoothing Kernel:
   - The images will appear slightly smoother compared to the original data.
   - Fine details and noise will be reduced.



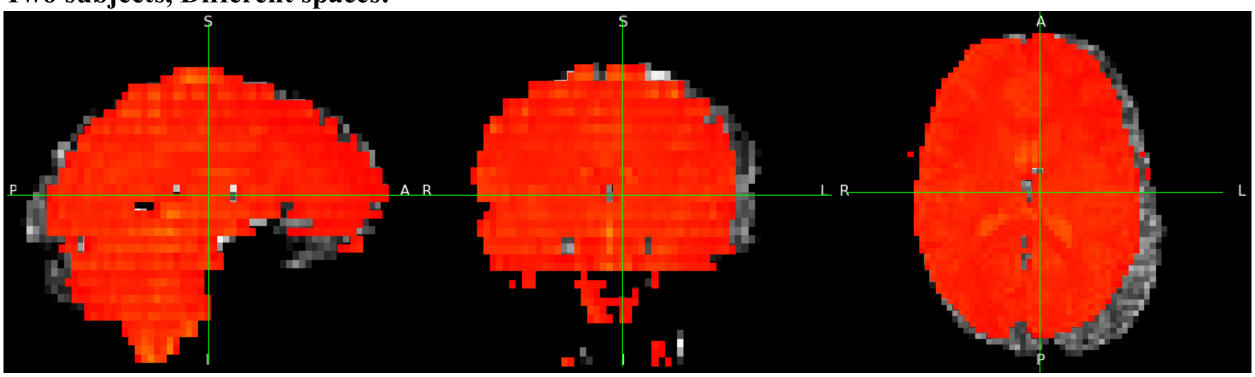
2. 12mm Smoothing Kernel:
   - The images will appear significantly smoother compared to both the original data and the 3mm smoothed images.

- Fine details and noise will be further reduced compared to the 3mm smoothing.



Thirdly, we will do registration and normalization.

Two subjects, Different spaces!

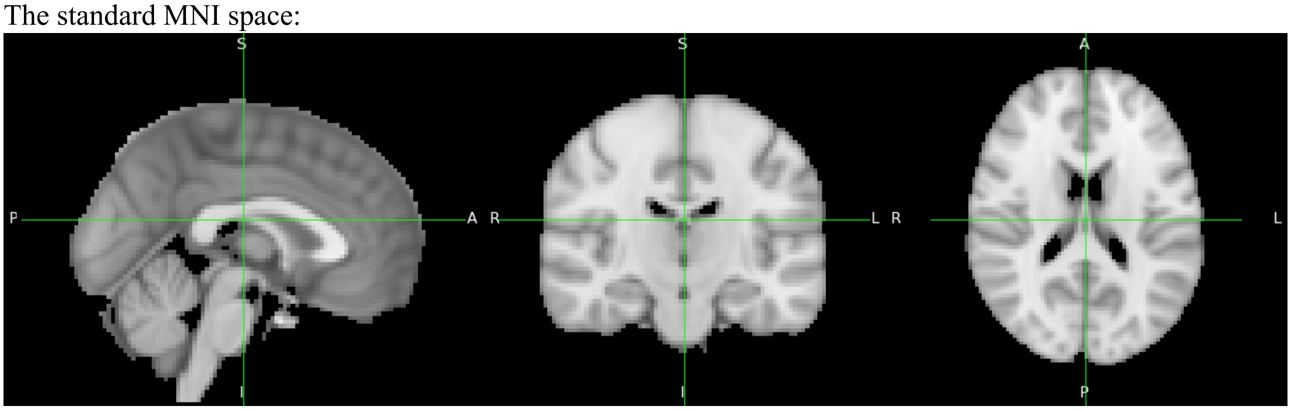


This is done due to the difference in brain shapes and sizes between the people. Therefore, if I said that a voxel in certain location in subj1 is activated it will not be the same voxel activated in subj19 because of different orientation.

Registration is from FMRI to anatomical image while the normalization is from FMRI to MNI or from anatomical to MNI.

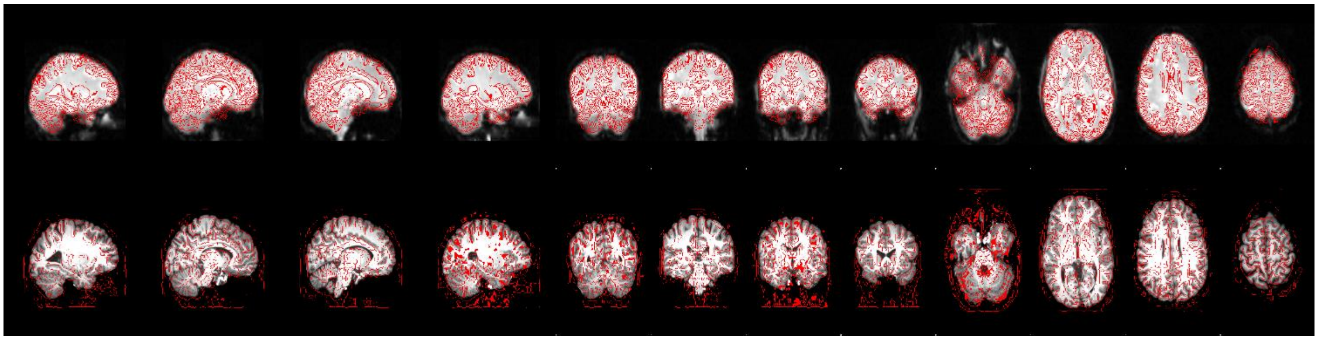
Therefore, we need to convert from FMRI space to the standard MNI space.

The standard MNI space:

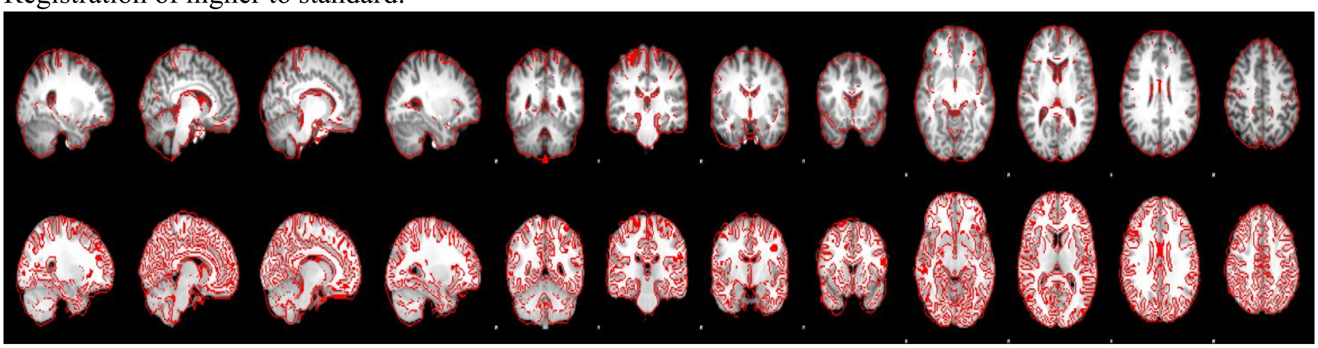


It is easier to transform the structural image to MNI space as it is of high resolution and so do the same transformations for the FMRI image after zooming.

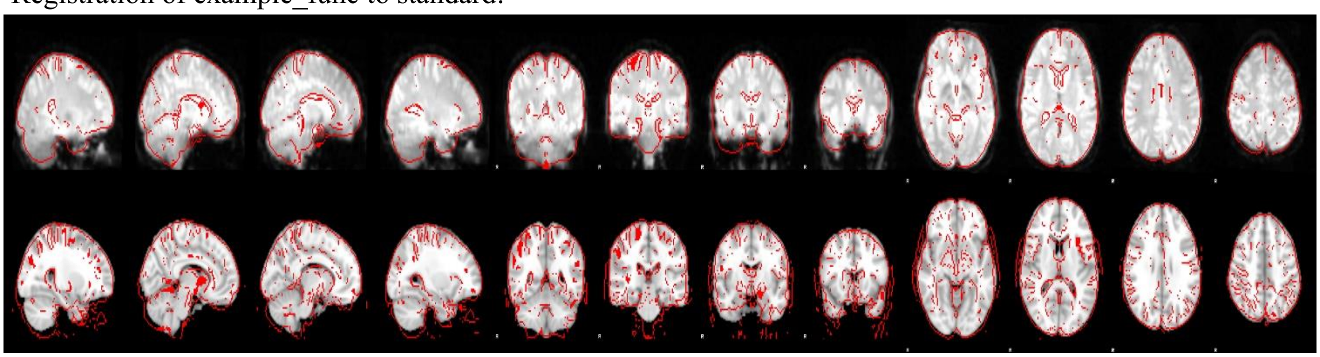
Registration of example_func to highers:



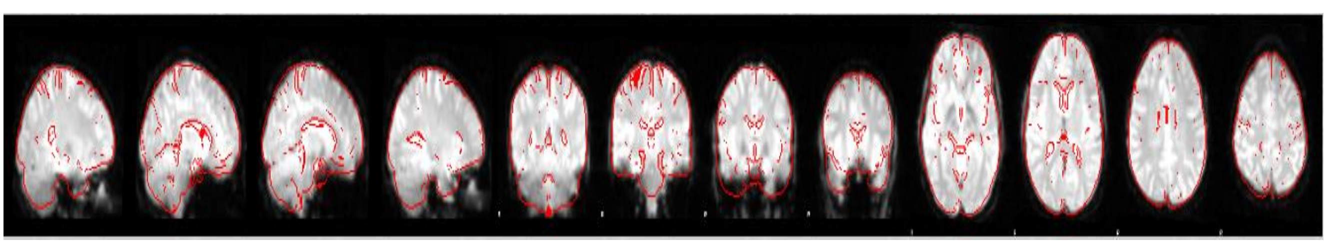
Registration of higher to standard:



Registration of example_func to standard:



Summary of registration from FMRI to standard



Exercise for registration and normalization:

When running the preprocessing with 3DOF for registration and normalization instead of 12DOF:

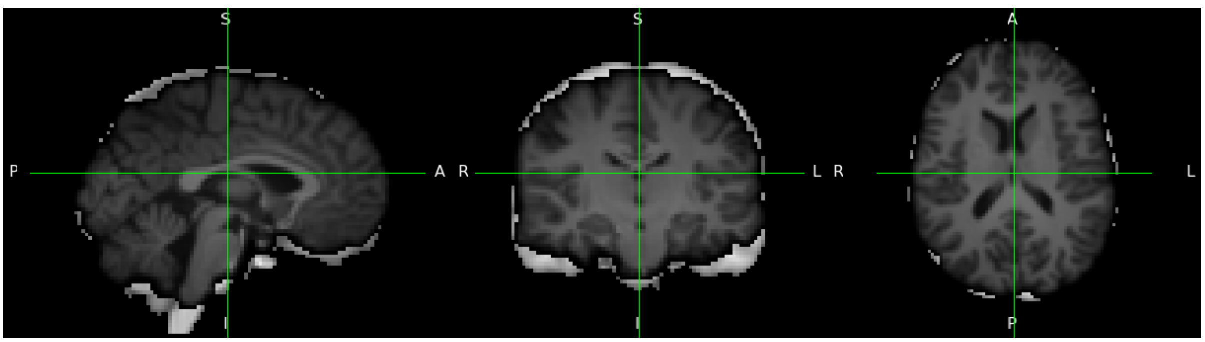
The output will surely differ from the preprocessing in terms of the alignment and spatial normalization of the functional data

1. **3DOF:**

   - The registration will primarily correct for translations in the functional data.
   - The normalization step will transform the functional data to match the overall size and position of the MNI template.
   - The resulting output will have limited capacity to account for local anatomical deformations.

** highers2standard represents the functional data after both motion correction and spatial normalization Using NDOFs:

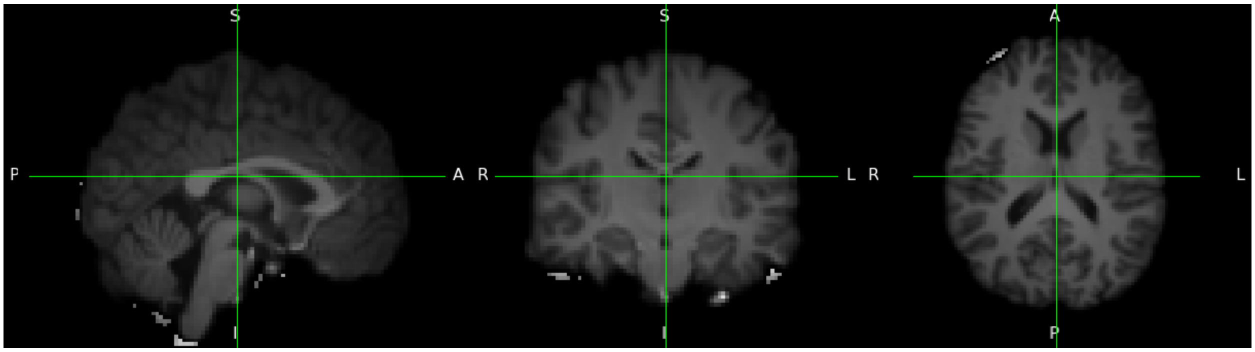
The visualization of highers2standard image(the black one) using 3DOFs on MNI template:



2. **12DOF:**

   - The registration will include additional parameters to account for shearing and non-linear deformations (scaling and rotations).
   - The normalization step will perform a more detailed mapping of the functional data to the MNI template, considering local anatomical variations and deformations.
   - The resulting output will have a higher level of spatial alignment and better account for subtle anatomical differences across individuals.

The visualization of highers2standard image(the black one) using 12DOFs on MNI template (alignment on MNI space more accurate):

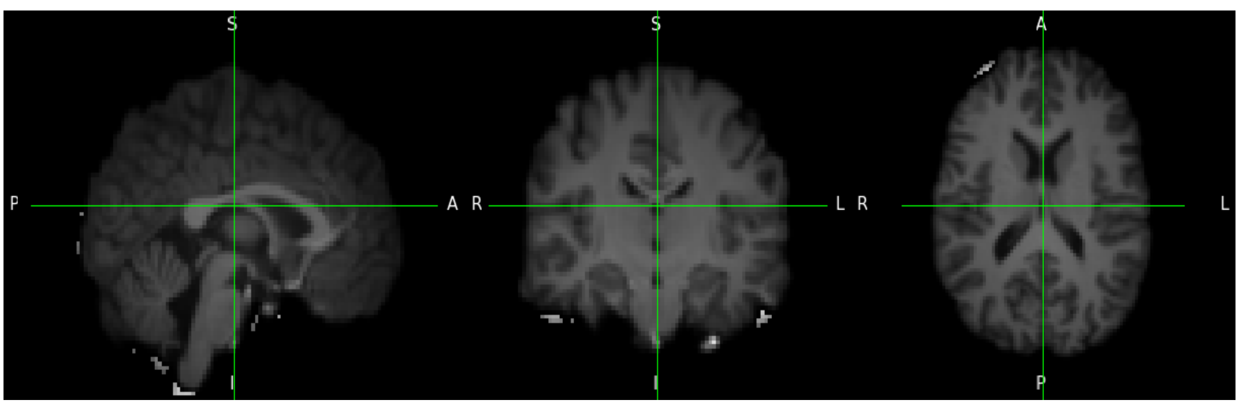


What if we done preprocessing with BBR?!
BBR (Boundary-Based Registration) is a registration algorithm optimizes functional-anatomical alignment by using boundary information, enhancing registration quality between high-resolution anatomical and functional data.

Therefore, the difference using BBR and 12DOF registration:
- Alignment more precise.
- More effective in capturing local anatomical variations and deformations.
- Takes much longer time.

I think the case to use such algorithm can be Research as we will need more accurate fine results.
and motion artifact robustness

The visualization of highers2standard image(the black one) using BBR on MNI template (alignment on MNI space more accurate):



Then for General linear model or the statistics we will do:

Firstly, we have timing files for run.

The timing file:

```
onset   duration        trial_type      response_time   correctness     StimVar Rsponse Stimulus
0.0     2.0     incongruent_correct     0.56    correct 2       1       incongruent     cond003
10.0    2.0     congruent_correct       0.356   correct 2       1       incongruent     cond001
20.0    2.0     congruent_correct       0.416   correct 1       1       congruent       cond001
30.0    2.0     congruent_correct       0.363   correct 1       1       congruent       cond001
42.0    2.0     congruent_correct       0.406   correct 2       1       incongruent     cond001
54.0    2.0     incongruent_correct     0.537   correct 1       1       congruent       cond003
64.0    2.0     incongruent_correct     0.484   correct 1       1       congruent       cond003
76.0    2.0     incongruent_correct     0.479   correct 2       1       incongruent     cond003
88.0    2.0     incongruent_correct     0.49    correct 1       1       congruent       cond003
102.0   2.0     congruent_correct       0.42    correct 2       1       incongruent     cond001
116.0   2.0     congruent_correct       0.374   correct 1       1       congruent       cond001
130.0   2.0     incongruent_correct     0.483   correct 1       1       congruent       cond003
144.0   2.0     incongruent_correct     0.521   correct 1       1       congruent       cond003
154.0   2.0     incongruent_correct     0.471   correct 2       1       incongruent     cond003
164.0   2.0     congruent_correct       0.448   correct 2       1       incongruent     cond001
174.0   2.0     congruent_correct       0.38    correct 2       1       incongruent     cond001
184.0   2.0     incongruent_incorrect   0.392   incorrect       1       congruent       cond004
196.0   2.0     incongruent_correct     0.604   correct 1       1       congruent       cond003
208.0   2.0     congruent_correct       0.53    correct 2       1       incongruent     cond001
220.0   2.0     congruent_correct       0.466   correct 2       1       incongruent     cond001
232.0   2.0     congruent_correct       0.427   correct 2       1       incongruent     cond001
246.0   2.0     incongruent_correct     1.022   correct 1       1       congruent       cond003
260.0   2.0     congruent_correct       0.455   correct 2       1       incongruent     cond001
274.0   2.0     incongruent_correct     0.616   correct 1       1       congruent       cond003
```

So we know the timing of each stimulus and its duration by running a script file to separate congruent task from incongruent.

we will have a timing file for incongruent task and another one for congruent task for each run.

This step will help FSL to do the GLM as by knowing starting of stimulus and its duration it can generate the BOLD signal or the hemodynamic response and can do regression between voxel and the signal to see which voxels are related to the task.

The script:

```
#!/bin/bash
#Check whether the file subjList.txt exists; if not, create it
if [ ! -f subjList.txt ]; then
        ls -d sub-?? > subjList.txt
fi
#Loop over all subjects and format timing files into FSL format
for subj in `cat subjList.txt` ; do
        cd $subj/func
        #Navigate to the subject's func directory, which contains the timing files
        #Extract the onset times for the incongruent and congruent trials for each run. NOTE: This script only extracts the trials in which
        #the subject made a correct response. Accuracy is nearly 100% for all subj ects, but as an exercise the student can modify this to extract
        #the incorrect trials as well,
        cat ${subj}_task-flanker_run-1_events.tsv | awk '{if ($3=="incongruent_correct") {print $1, $2, "1"}}' > incongruent_run1.txt
        cat ${subj}_task-flanker_run-1_events.tsv | awk '{if ($3=="congruent_correct") {print $1, $2, "1"}}' > congruent_run1.txt

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

        cd ../..
done
```

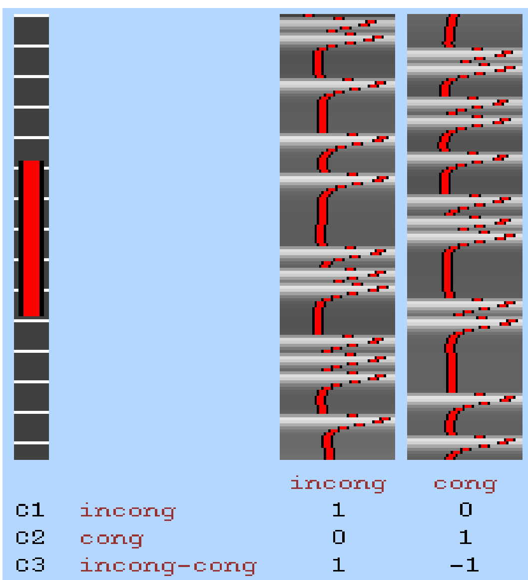
The file for congruent timing:

```
20.0 2.0 1
30.0 2.0 1
52.0 2.0 1
64.0 2.0 1
88.0 2.0 1
116.0 2.0 1
130.0 2.0 1
140.0 2.0 1
184.0 2.0 1
196.0 2.0 1
246.0 2.0 1
274.0 2.0 1
```

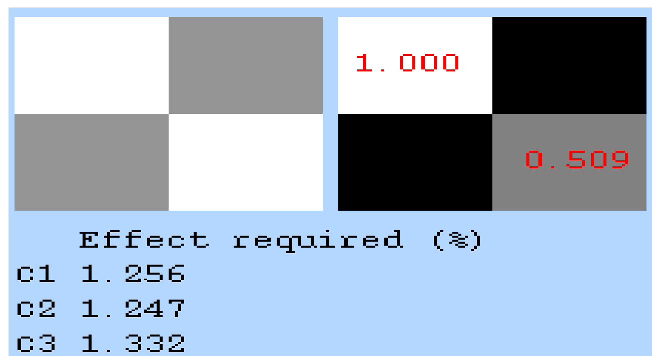
The file for incongruent timing:

```
0.0 2.0 1
10.0 2.0 1
40.0 2.0 1
76.0 2.0 1
102.0 2.0 1
150.0 2.0 1
164.0 2.0 1
174.0 2.0 1
208.0 2.0 1
220.0 2.0 1
232.0 2.0 1
260.0 2.0 1
```

The Design matrix:

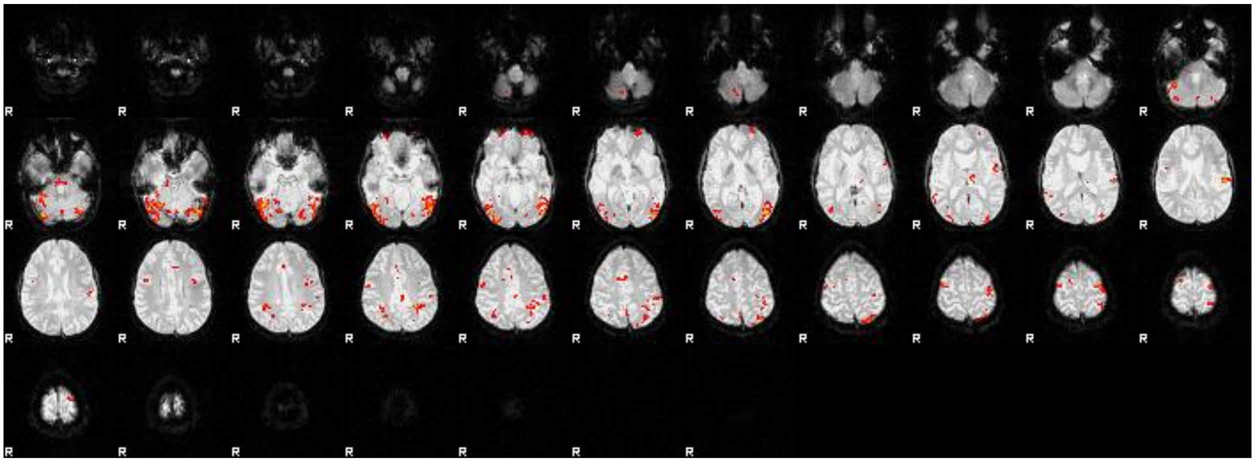


Covariance matrix & design efficiency:

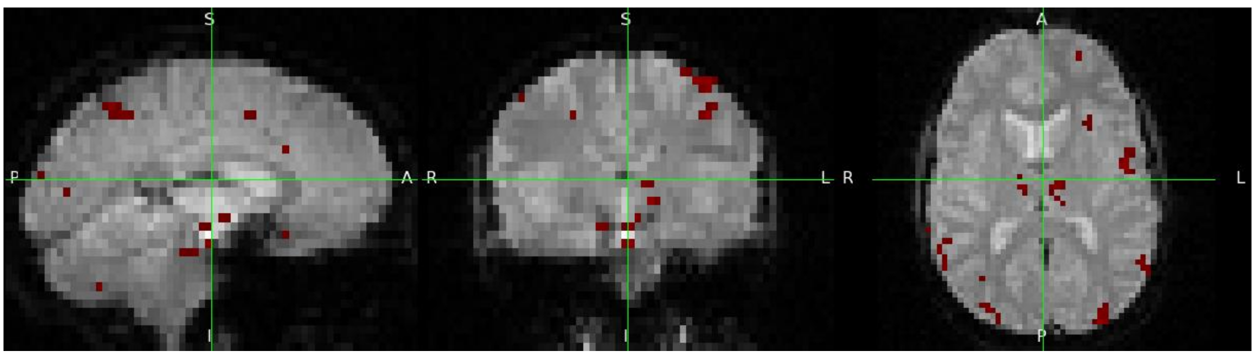


Statistic images were thresholded using clusters determined by Z>3.1 and a corrected cluster significant threshold of P=0.05.

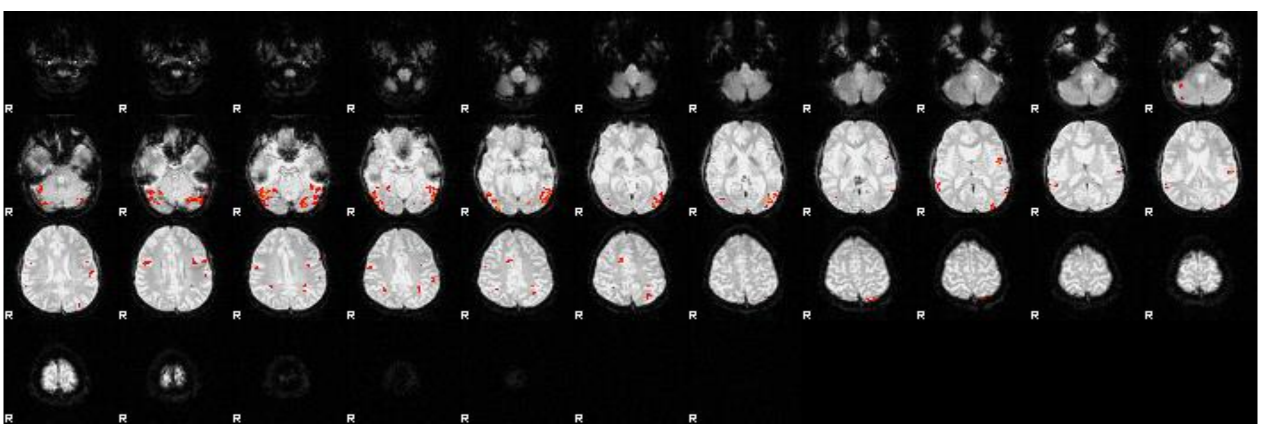
The colored regions are the voxels that activated due incongruent task (more voxels than in congruent as incongruent task needs more brain work):



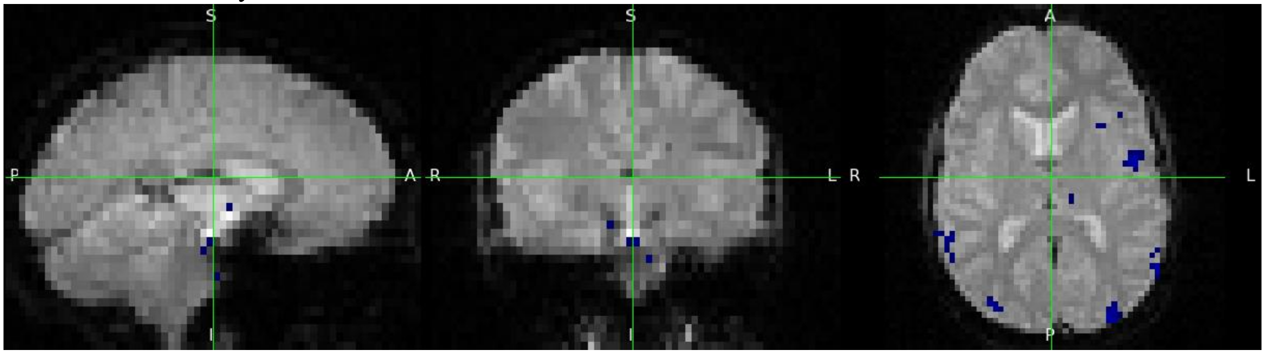
Visualization in fsleyes:



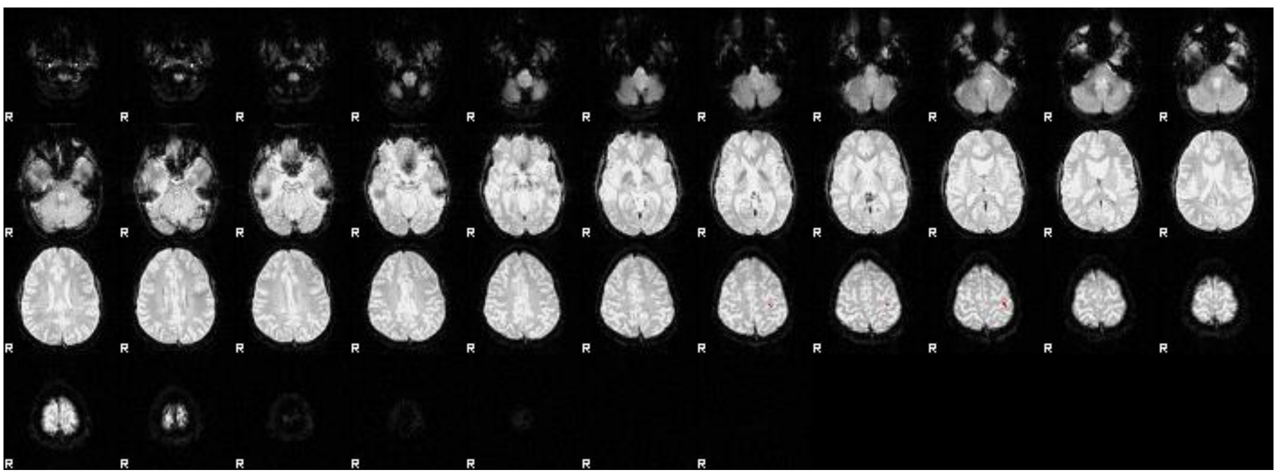
The colored regions are the voxels that activated due congruent task:



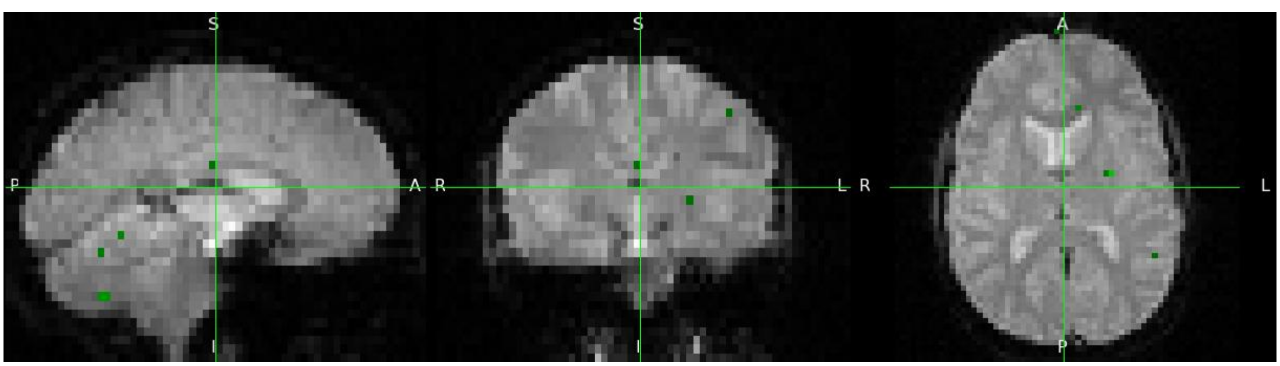
Visualization in fsleyes:



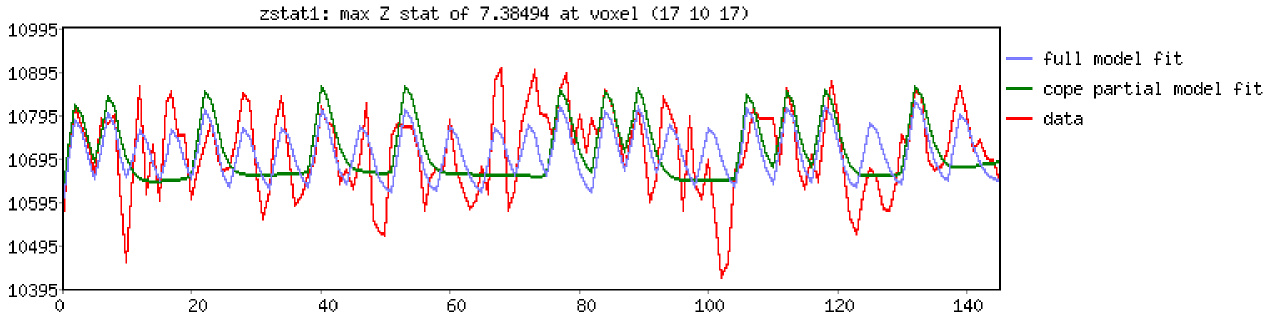
The colored regions are the voxels that activated due incongruent task but not congruent task they are very few and will appear more with another thresholds in group analysis:

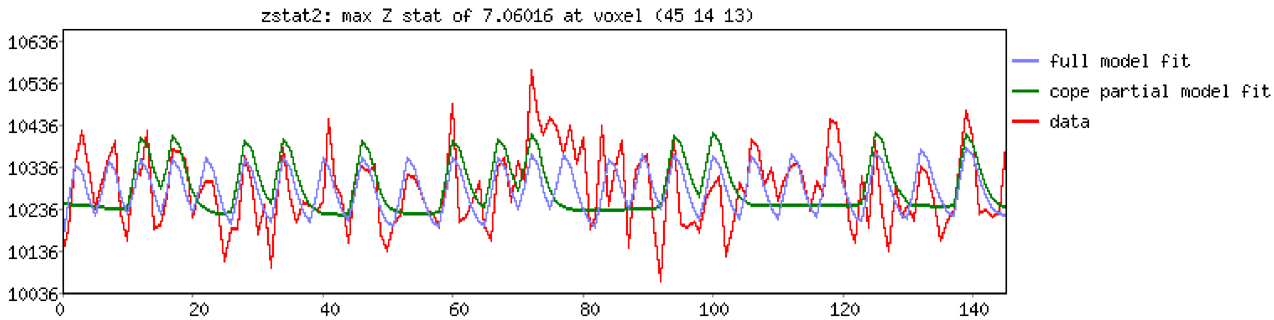


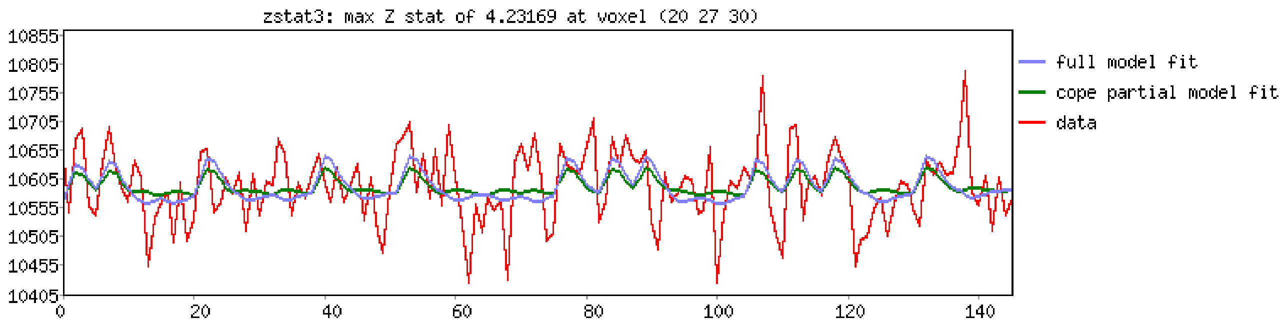
Visualization in fsleyes:



Finally, Time Series Plots:







We can do all of the above steps for two runs by just running a script and all we need is design.fsf file from one feat directory of any run and we can do text replacement in this file by the script to change subj name and switch between run1 and run2.

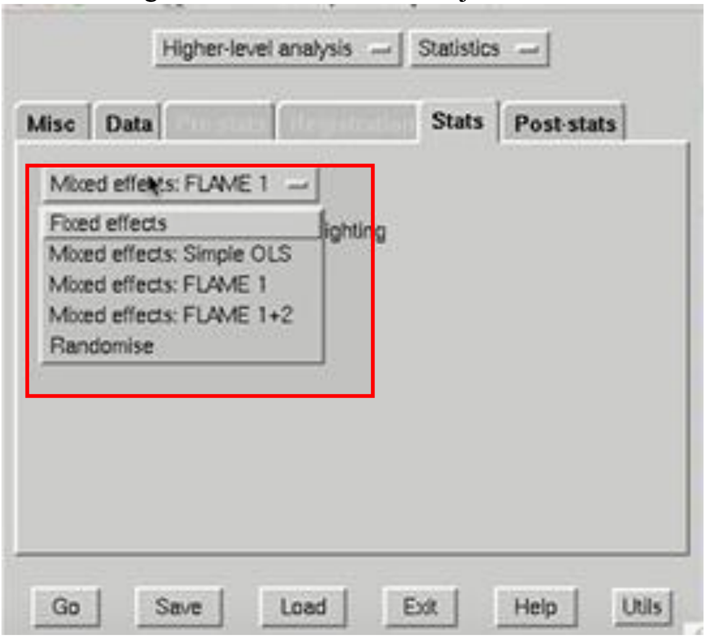
The script iterates over 26 subjects using a for loop, where each subject is identified by a unique id (sub-01 to sub-26).
For each subject, the script checks if a preprocessed anatomical image exists and if not, it will apply brain extraction called **bet2** to generate the brain extracted image. The script then copies a template design file (design.fsf) and creates two new design files (design1.fsf and design2.fsf) for each of two functional runs.
Then **sed** command to replace the subject id and run numbers in the design files with the current subject id and run number. Then, runs FSL's **feat** command on each design file to perform first-level analysis for each run. Finally, the script moves back to the main directory and repeats the process for the next subject.

My script:
```
for id in `seq -w 1 26`; do
           subj="sub-$id"


        echo "Start processing $subj"
        echo

        cd $subj
        if [ ! -f anat/${subj}_T1w_brain.nii.gz ]; then
        bet2 anat/${subj}_T1w.nii.gz anat/${subj}_brain.nii.gz -f 0.4
        fi


        cp ../design.fsf design1.fsf

        #change subject to current subject using "sed" command
        sed -i "s/sub-01/${subj}/g" design1.fsf

        cp design1.fsf design2.fsf

        #change run1,run-1 with run2,run-2 in design2.fsf using "sed" command
        sed -i 's/run1/run2/g; s/run-1/run-2/g' design2.fsf

        echo "FLA RUN 1"
        echo
        feat design1.fsf
        echo "FLA RUN 2"
        echo
        feat design2.fsf

        cd ..
done
echo
```

Higher Level Analysis:

Second level analysis,
It means calculating the average of the contrast estimates from the first level analysis to get a single second level cope image for each contrast for each subject.

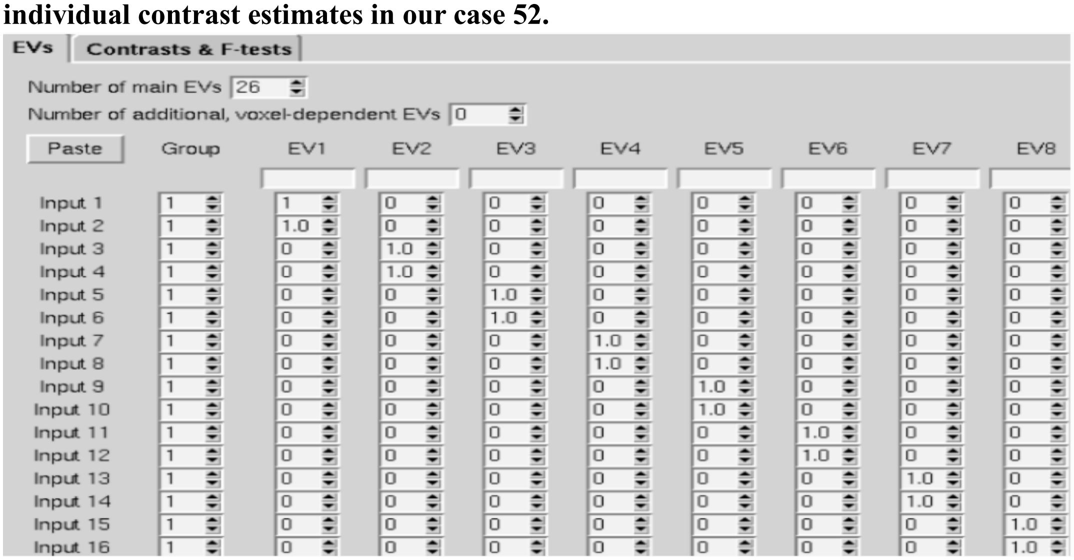
After loading the two runs for each subject



Different from the Stats tab in First level analysis here we will find this and which will make us able to choose between different types of inference or how you want the results to generalize to the population.
I will discuss them in **3rd analysis part.** For now I will just give a hint about Fixed effects:
In this model, The data from each subject or condition is treated as independent and combined to estimate the overall effect or response. This model is often used when the primary interest is to examine the average effect across all subjects or conditions, rather than investigating individual variability or differences between subjects/conditions.

And since we simply want to take the average of the parameter estimates across the runs within each subject so we will use it.

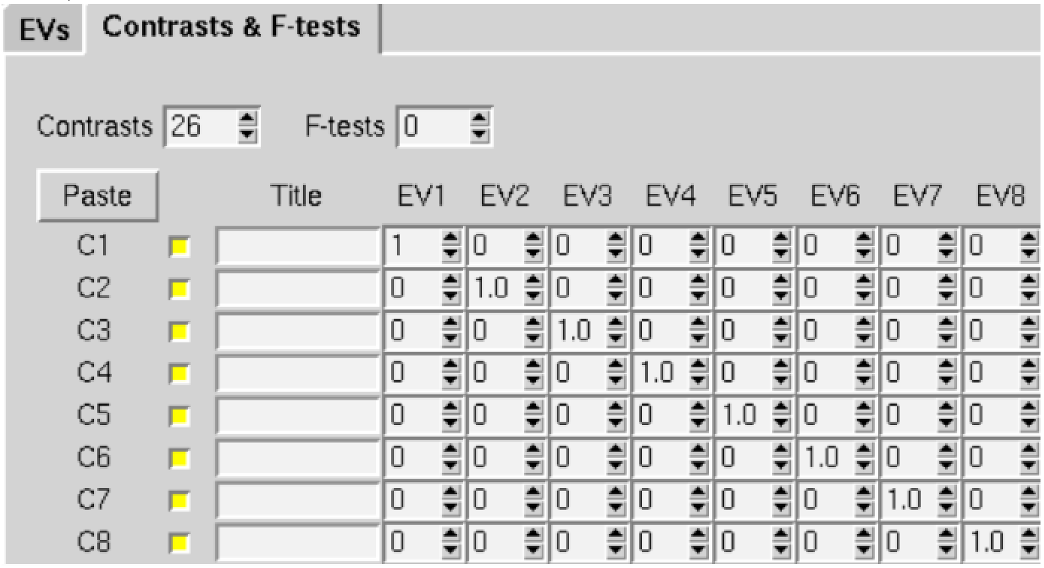
Then click on Full model setup, we will get a window with number of rows representing of individual contrast estimates in our case 52.



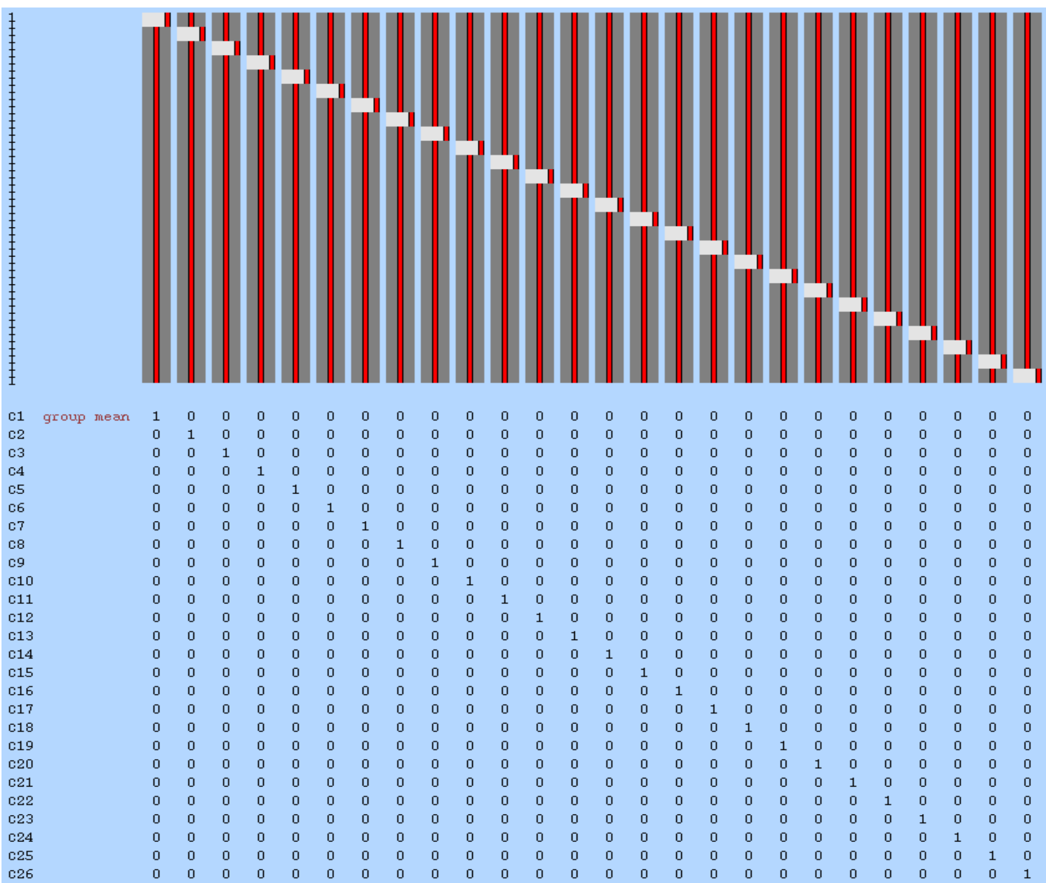
	Group	EV1	EV2	EV3	EV4	EV5	EV6	EV7	EV8
Input 1	1	1	0	0	0	0	0	0	0
Input 2	1	1.0	0	0	0	0	0	0	0
Input 3	1	0	1.0	0	0	0	0	0	0
Input 4	1	0	1.0	0	0	0	0	0	0
Input 5	1	0	0	1.0	0	0	0	0	0
Input 6	1	0	0	1.0	0	0	0	0	0
Input 7	1	0	0	0	1.0	0	0	0	0
Input 8	1	0	0	0	1.0	0	0	0	0
Input 9	1	0	0	0	0	1.0	0	0	0
Input 10	1	0	0	0	0	1.0	0	0	0
Input 11	1	0	0	0	0	0	1.0	0	0
Input 12	1	0	0	0	0	0	1.0	0	0
Input 13	1	0	0	0	0	0	0	1.0	0
Input 14	1	0	0	0	0	0	0	1.0	0
Input 15	1	0	0	0	0	0	0	0	1.0
Input 16	1	0	0	0	0	0	0	0	1.0

We will set 1 in the field where we want to take the average of the contrast estimates for that subject.

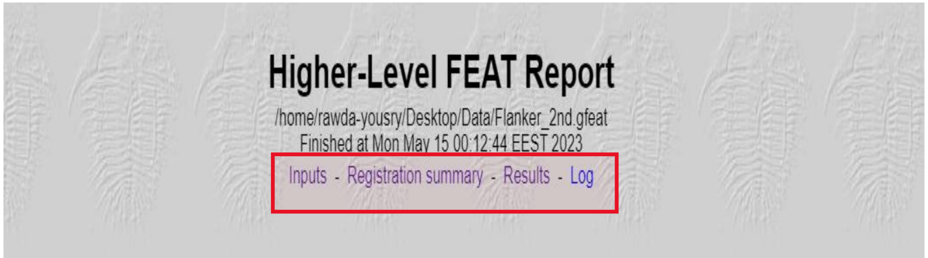
Then, Click on Contrasts & F-tests tab



Change the numbers in the diagonal to 1 this tells FSL to calculate an average estimate for each that was specified in the data tab.
Once finished hit Done then Go.
The resulted GLM:



Now The Fsl Report will appear, Registaration summary is the same a



In inputs tab it contains the results of 1st level analysis for each feat directory.
Registration Summary: shows how the functional data have been normalized to a template.
As we already checked the normalization during the preprocessing so no problem should appear here.
Results tab will show us the GLM for second level analysis along with the results for each of the contrasts.
As first level we are not interested in post stats but I found that just we can check them for any activation in the ventricles or strong activation at the edges of the brain which tell us that there is a movement.

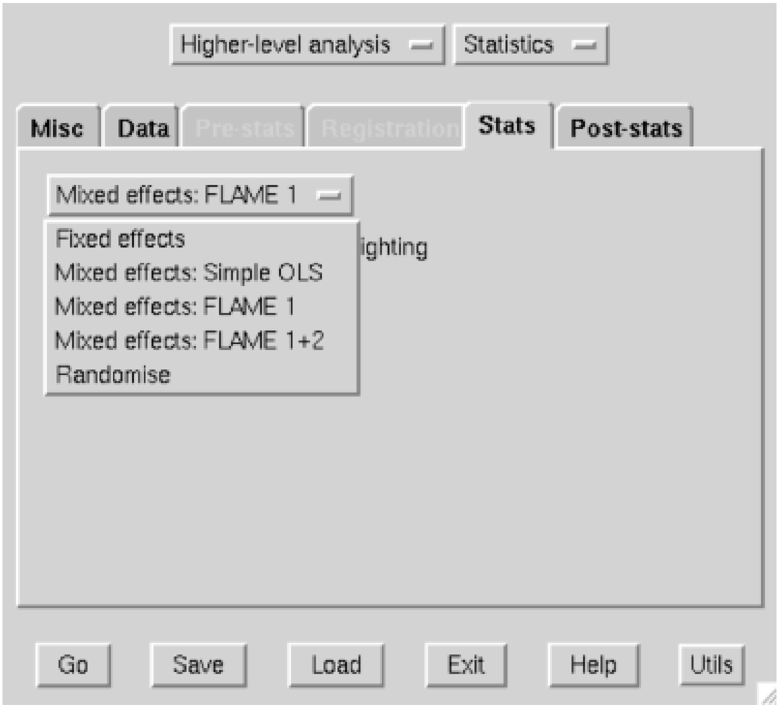
Third-Level analysis:

We need to generalize our results to the population!
Our point now is if we see changes in brain activity in our sample can we say that these changes would be seen in the population as well?!

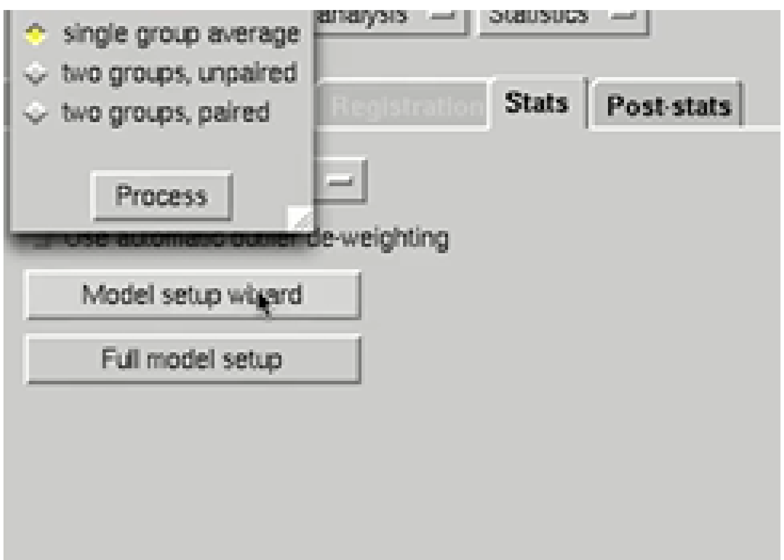
To test this we will need to run **Third-Level analysis** which is a group-level analysis:

We will calculate the mean and the variance for a contrast estimate and then do a t-test to see whether that mean is statistically significant or not.

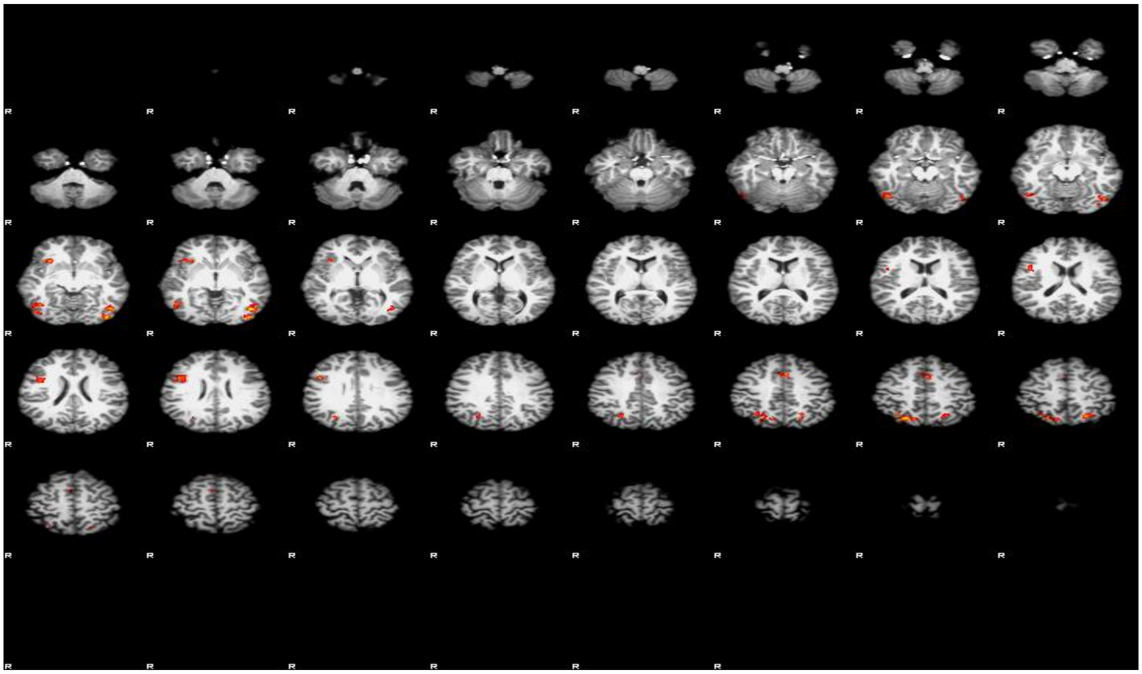
After loading our data, we will go to tab stats,
We will use mixed effects, this models of variance so that our results are generalizable to the population that our sample was drawn from.
FLAME 1 estimates parameters by using information about both within subject and between subject variability.
FLAME 1+2 considered more accurate but the additional benefit is usually minimal and it takes much longer.
Randomise is a non-parametric test.



As we have already taken the contrast for each subject so we can choose single group average.

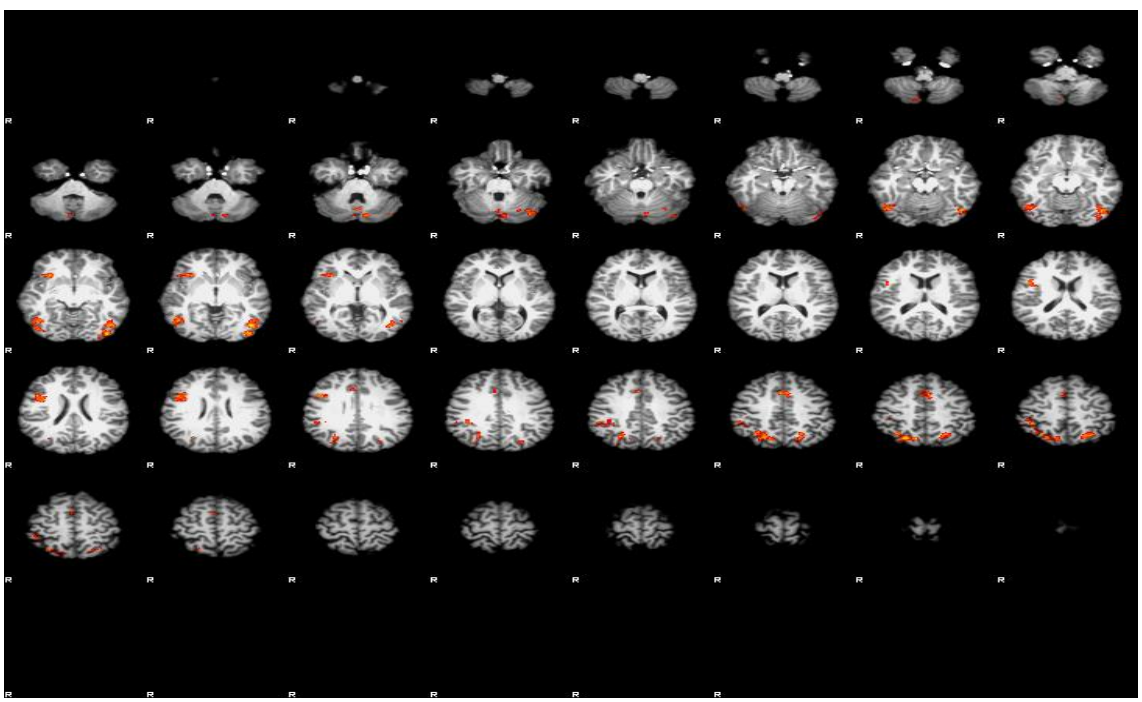


The resulted thresholded activation images when used FLAME 1:

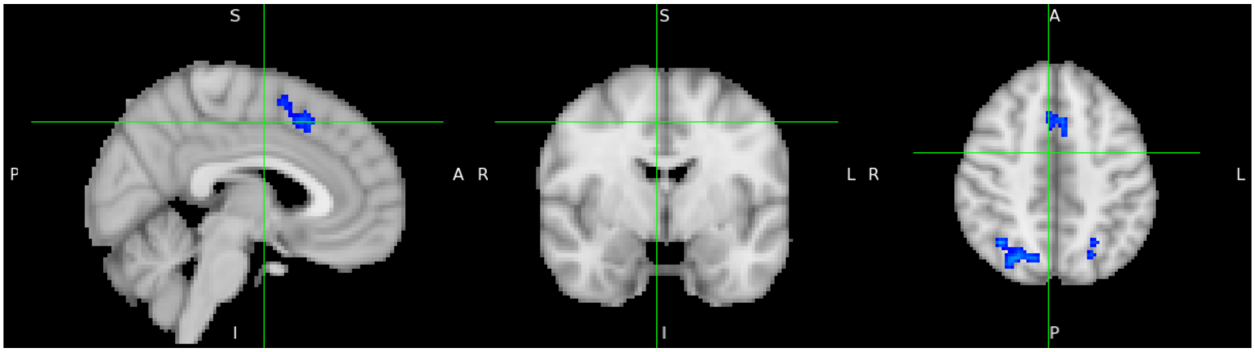


Some exercise:

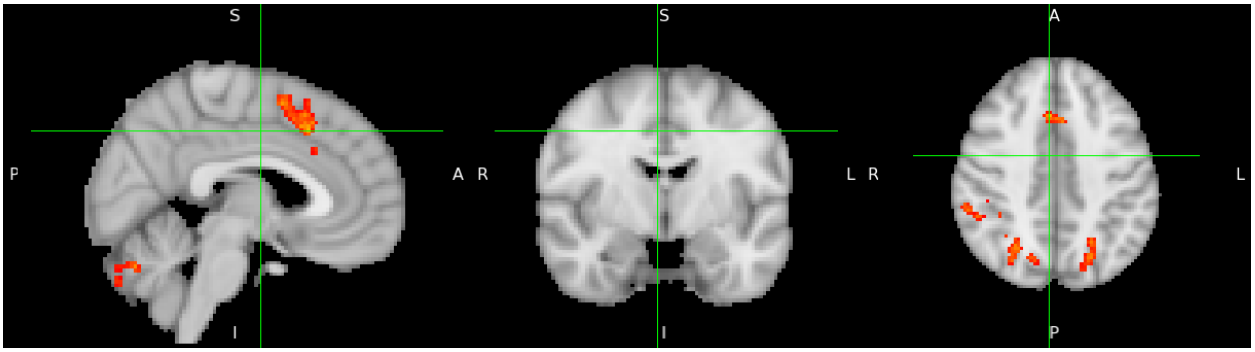
The resulted thresholded activation images when used FLAME 1+2:



For cleaner images I will visualize using fsleyes:
FlAME1

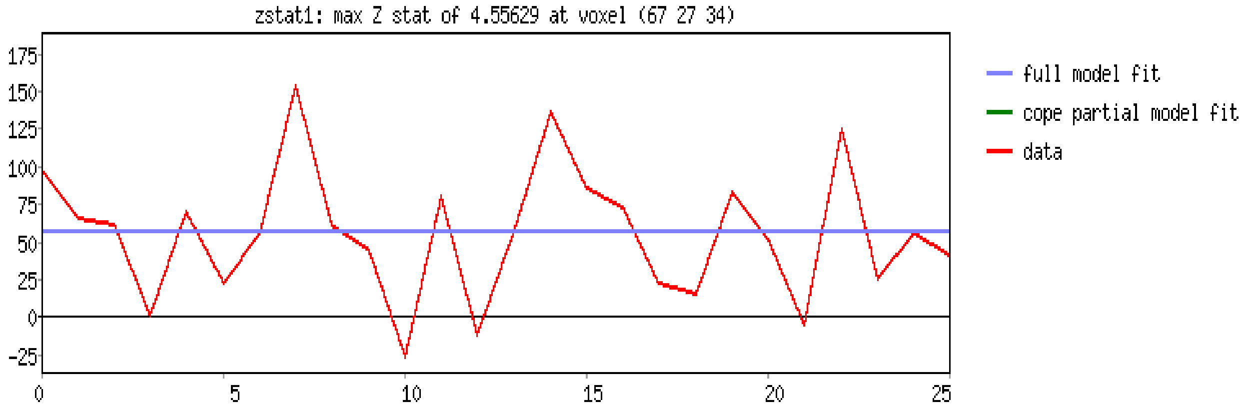


FLAME1+2

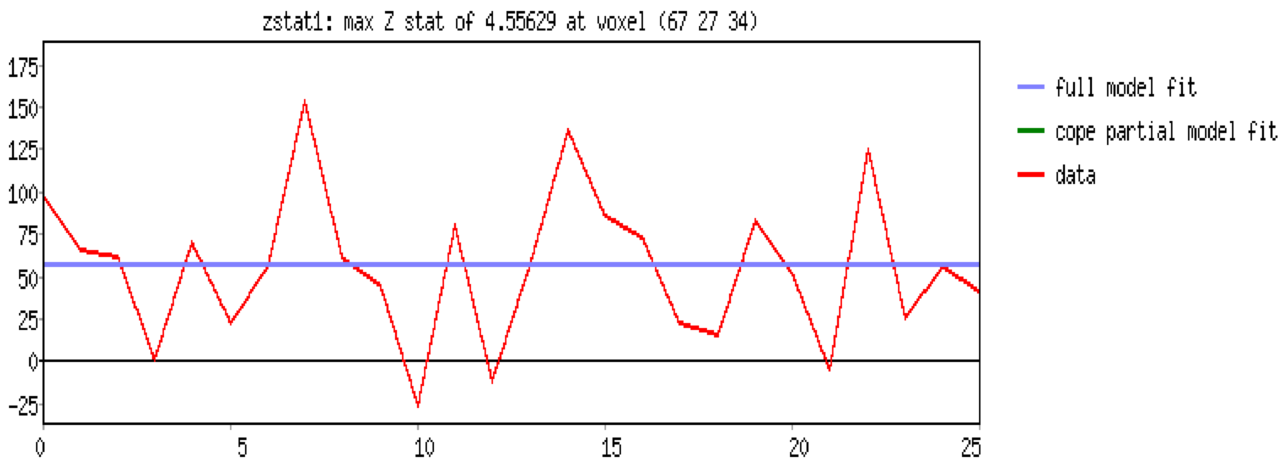


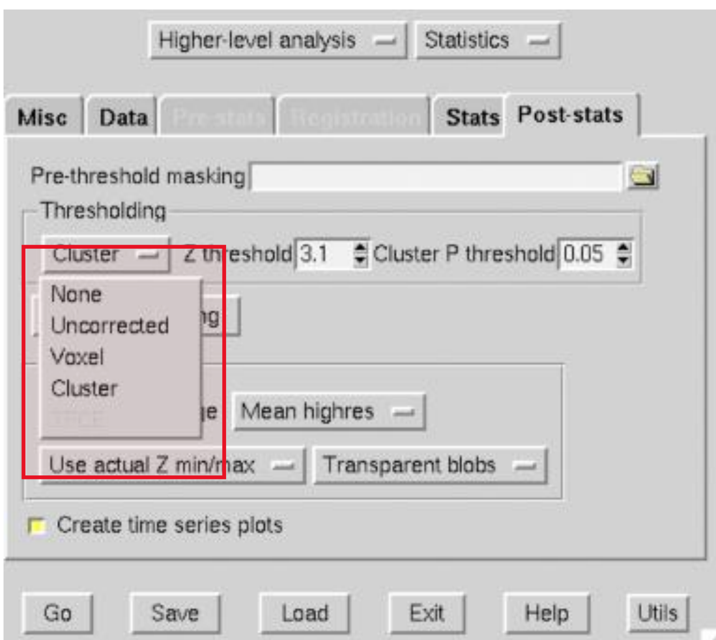
Time series plots:

FlAME 1:

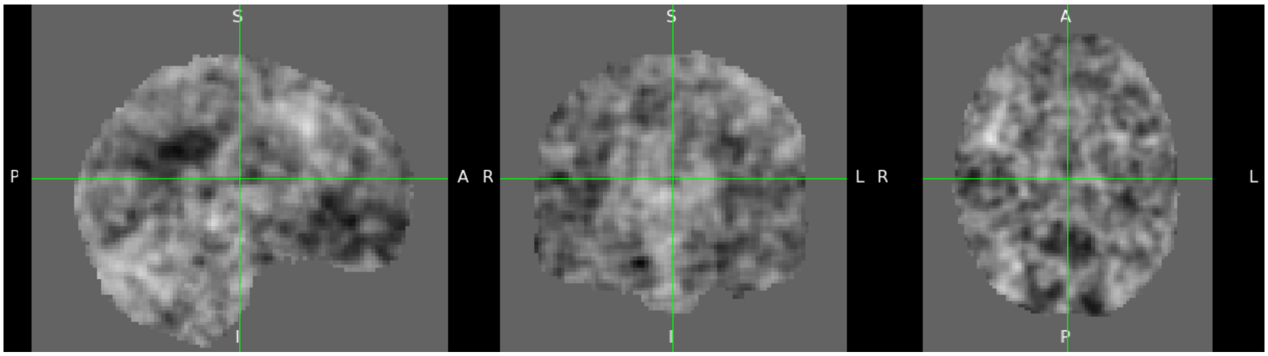


FLAME1+2:


zstat1: max Z stat of 4.55629 at voxel (67 27 34)
full model fit
cope partial model fit
data


Higher-level analysis
Statistics
Misc
Data
Pre-stats
Registration
Stats
Post-stats
Pre-threshold masking
Thresholding
Cluster
Z threshold 3.1
Cluster P threshold 0.05
None
Uncorrected
Voxel
Cluster
Mean highres
Use actual Z min/max
Transparent blobs
Create time series plots
Go
Save
Load
Exit
Help
Utils

Here if we clicked None instead of Cluster:
Nothing appeared in the folder no threshold is applied to the statistical results. This results in a map that includes all voxels, regardless of their statistical significance. (no thresh_zstat in the folder I found just zstat).

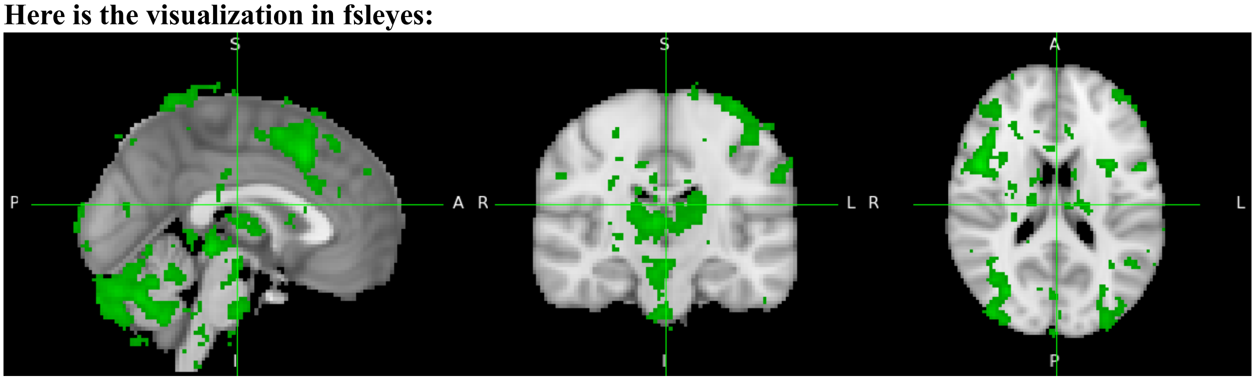


Comparison between unthresholded(None) result and the cluster-corrected result:

None	Cluster-corrected
Activation of signal across a larger extent of the brain compared to the cluster-corrected results. This is because the it includes both true activations and potential false positives.	offer a more strict approach by applying statistical thresholds and accounting for multiple comparisons, resulting in a more reliable identification of significantly activated clusters.

When I clicked on Uncorrected:
Statistical threshold is chosen without correction for multiple comparisons. This means that each voxel's statistical value is evaluated independently, without considering the neighboring voxels. The threshold is typically set based on a p-value, and voxels exceeding this threshold are considered significant.

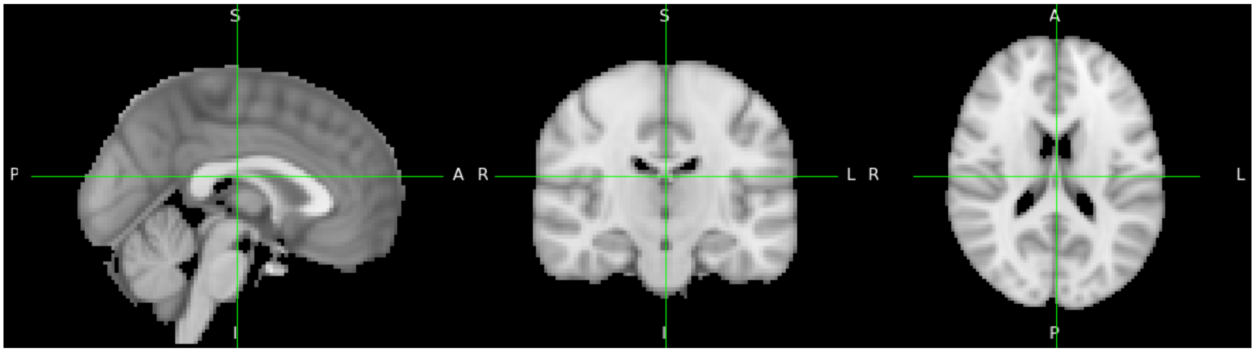
Here is the visualization in fsleyes:



If we compared with the cluster-corrected results, the Uncorrected threshold reveal more numerous, smaller clusters of activation. As, the Uncorrected threshold does not take into account the spatial extent of activation but focuses on individual voxel values.

Voxel Threshold:
A specific threshold is set based on the voxel-level statistical values. Only voxels that exceeds this threshold are considered significant, regardless of their spatial connectivity or cluster size.

But I found the result of applying this and visualize in fsleyes:



I think no colored voxels appeared there are no individual voxels that meet the specified criteria for significance. I think this happened as the threshold is very strict.

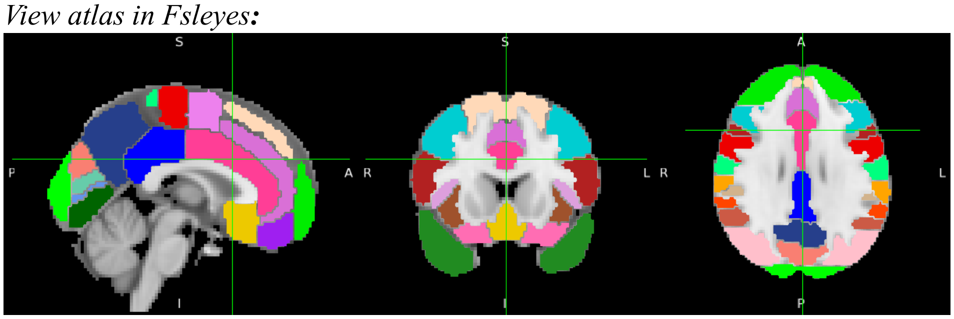
To summarize, the cluster-corrected results provide a more reliable identification of significant clusters, accounting for multiple comparisons and spatial extent of activation. In contrast, Uncorrected and Voxel thresholding methods may yield more numerous but fake activations, lacking the control and interpretation provided by the cluster-based approach.

ROI analysis:
It is a method of extracting data from a subset of voxels also called a mask.

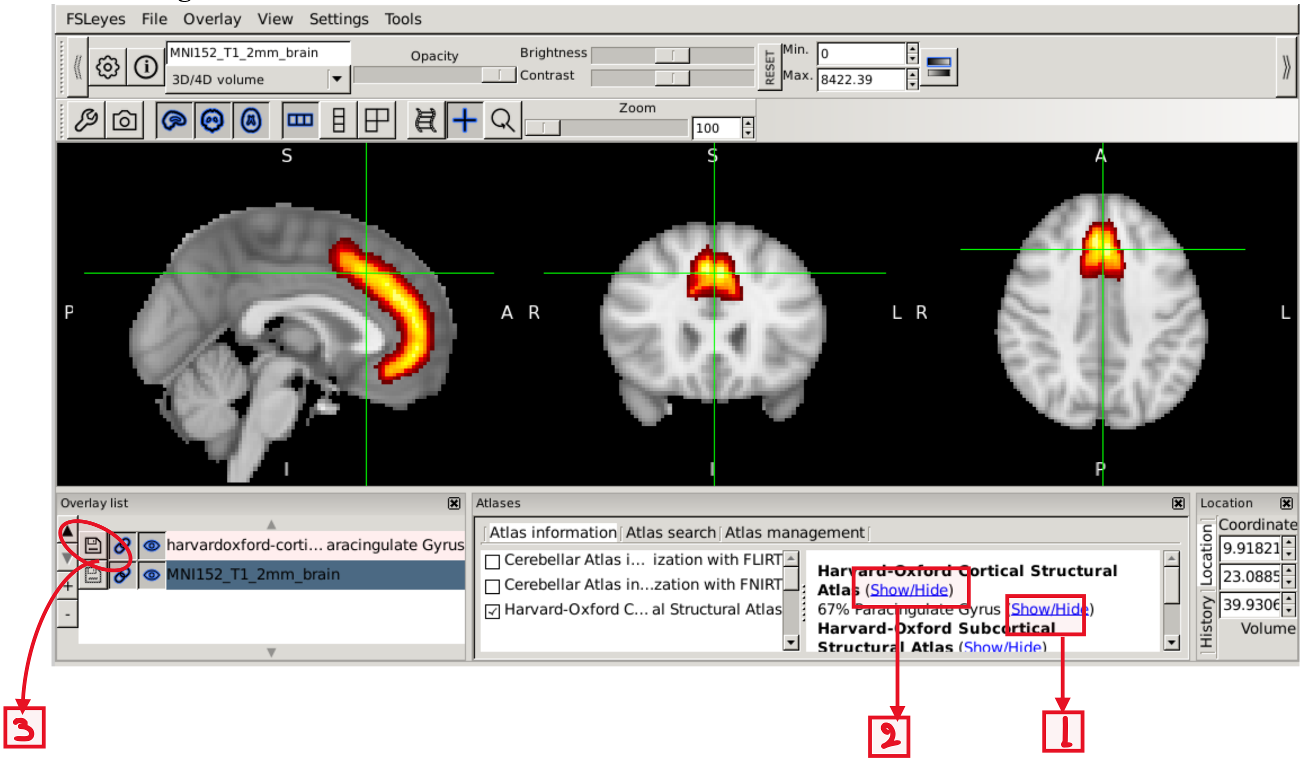
Ways to create a mask are:

- To use an atlas
- Spherical ROI approach in which a sphere of a given radius is centered on coordinates that based on peak activation of another study that uses a similar experimental design.

Firstly we will create a mask using atlas,
View atlas in Fsleyes:



To save region as a mask:



Then in step 3 mask will be saved.
Once mask created we can extract the contrast estimates from it.
For example, for incongruent-congurent contrast estimate:
We can find each subject data maps in the 2^{nd} level gfeat from cope3.feat then stats.
Well, we can extract data from the z-statistic maps, since they are standardized and easy to interpret.
We will need to merge all of the z-statistic maps into a single data set.
To do this we will use fslmerge
Once merged, use fslmeants command from the PCG mask that we created.
then 26 numbers will be printed.

Put these numbers on R for t-test to see whether contrast estimates are significantly different from zero :

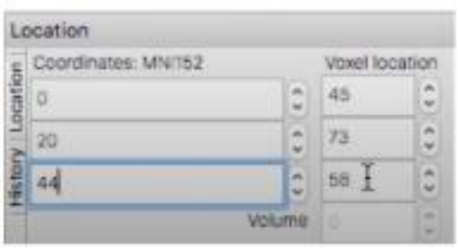
```
x <- scan()
1.381024
0.050253
-0.272654
0.265128
-0.084951
-0.133633
-0.114938
-0.389024
0.210950
-0.964052
-0.657895
0.245036
-0.192762
-0.039086
0.357768
0.997227
-0.360570
-0.546491
0.216187
1.051421
-0.205906
-0.087538
0.228115
0.745255
0.745922
0.267038
t.test(x)
```

```
Read 26 items
One Sample t-test

data:  x
t = 0.9739, df = 25, p-value = 0.3394
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.1162668  0.3248687
sample estimates:
mean of x
0.1043009
```

As We see the results aren't significantly different.
It may be because the mask is large.

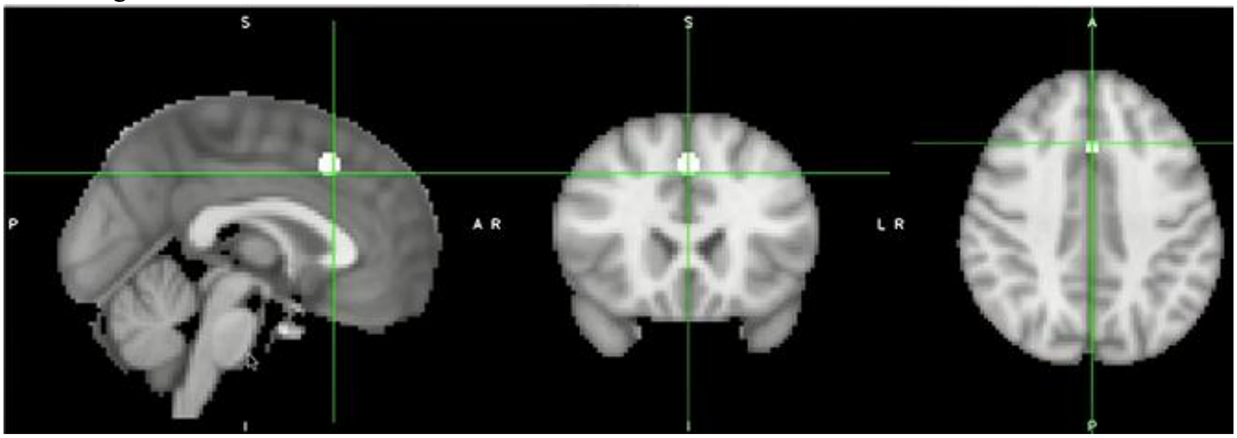
In this case we can instead use spherical ROI analysis:
Now to create a sphere from known MNI coordinate, we will do the following:

1. Convert MNI Coordinate to voxel space by entering the MNI coordinate we will get the Voxel Location.



2. Now will create a single voxel mask to be used as a base for next commands.
3. Expanding the mask into a sphere with a 5mm radius.

4. Binarizing the mask.



Lastly extracting data from the ROI by fslmeants to get the 26 numbers.

Again, put the 26 numbers on R for t-test,

```
x <- scan()
1.611508
-0.243948
0.159991
1.410235
-0.302093
-0.057510
0.767032
0.406651
1.146441
-0.650117
-0.620734
0.741809
0.402211
0.547149
1.018010
1.459229
0.112053
-0.849388
0.889456
1.075685
0.412434
0.989408
1.637930
1.868548
2.311128
-0.221740
t.test(x)
```

```
Read 26 items
One Sample t-test

data:  x
t = 3.7725, df = 25, p-value = 0.0008865
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.2798009 0.9526128
sample estimates:
mean of x
0.6162068
```

As We see the results are significantly different.
Because the ROI is more localized we have more power.

Some exercises:

when run probabilistic mask: I got these results on R by t-test.

```
x <- scan()
1.537802
-0.006258
-0.316923
0.417749
-0.154946
-0.342898
-0.111407
-0.428489
0.260987
-1.160258
-0.541683
0.427623
-0.145458
-0.006549
0.420421
1.050165
-0.486959
-0.581711
0.158486
1.043110
-0.177120
0.036641
0.198520
0.893546
0.963207
0.259173
t.test(x)
```

```
Read 26 items
One Sample t-test

data:  x
t = 1.0296, df = 25, p-value = 0.3131
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.1233780  0.3700527
sample estimates:
mean of x
0.1233373
```

The expectation is that when using a probabilistic mask and weighting the contrast estimates, the impact on the results would be relatively minor compared to using a binarized mask. This is because the probabilistic mask already provides more strict information about the probability of voxels belonging to the target region, and weighing the contrast estimates takes this into account. So difference between them is INDEED small.

Difference between using a probabilistic mask and a binarized mask may be more pronounced, especially if the mask contains voxels with varying degrees of probability within the target region.

Another exercise is done that run the following commands to create a sphere at another MNI location 36, -2, 48.

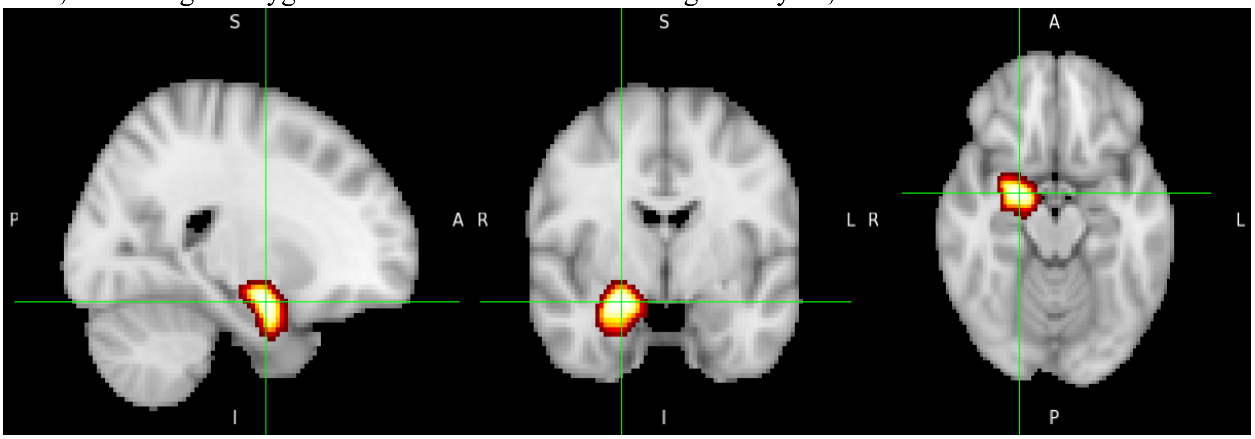
The equivalent voxel location is:

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

By running these commands, I can get a sphere of 7 mm:

```
fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1 -roi 27 1 62 1 60 1 0 1 ROI_dmPFC_27_62_60.nii.gz -odt float

fslmaths ROI_dmPFC_27_62_60.nii.gz -kernel sphere 7 -fmean ROI_Sphere_dmPFC_27_62_60.nii.gz -odt float

fslmaths ROI_Sphere_dmPFC_27_62_60.nii.gz -bin ROI_Sphere_bin_dmPFC_27_62_60.nii.gz.nii.gz
```

Also, I tried Right Amygdala as a mask instead of ParacingulateGyrus,

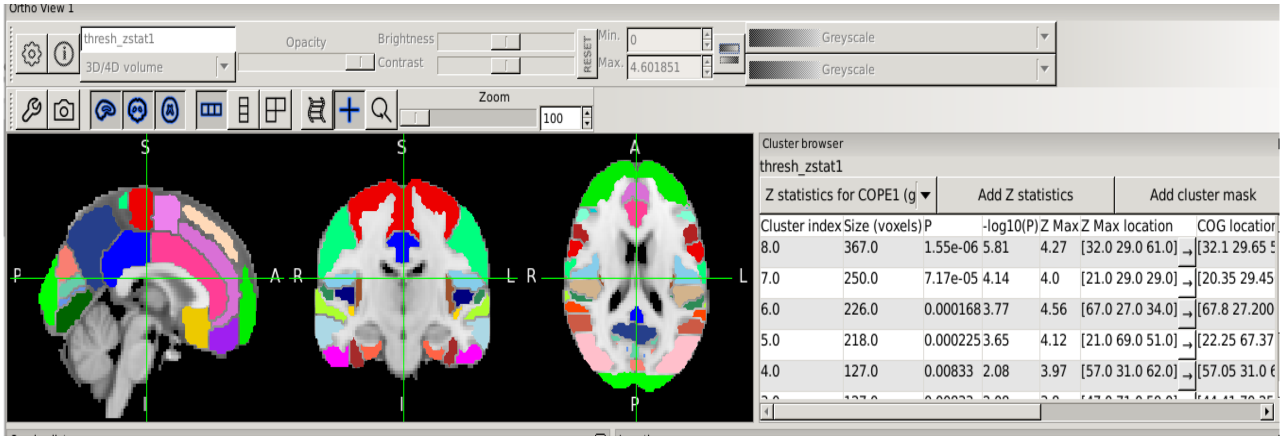


I got these numbers:

```
x <- scan()
1.176771
0.116265
-0.326218
0.165725
-0.251736
-0.521291
0.416361
-0.395260
0.117568
-1.134843
-0.258955
-0.284084
-0.296952
0.198067
-0.258478
0.810042
0.111055
-0.271156
0.462764
0.280954
0.291987
-0.130843
0.022621
0.620615
0.542818
0.241412
t.test(x)
```

```
Read 26 items
One Sample t-test

data:  x
t = 0.59692, df = 25, p-value = 0.5559
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.1361986  0.2473686
sample estimates:
 mean of x
0.05558496
```

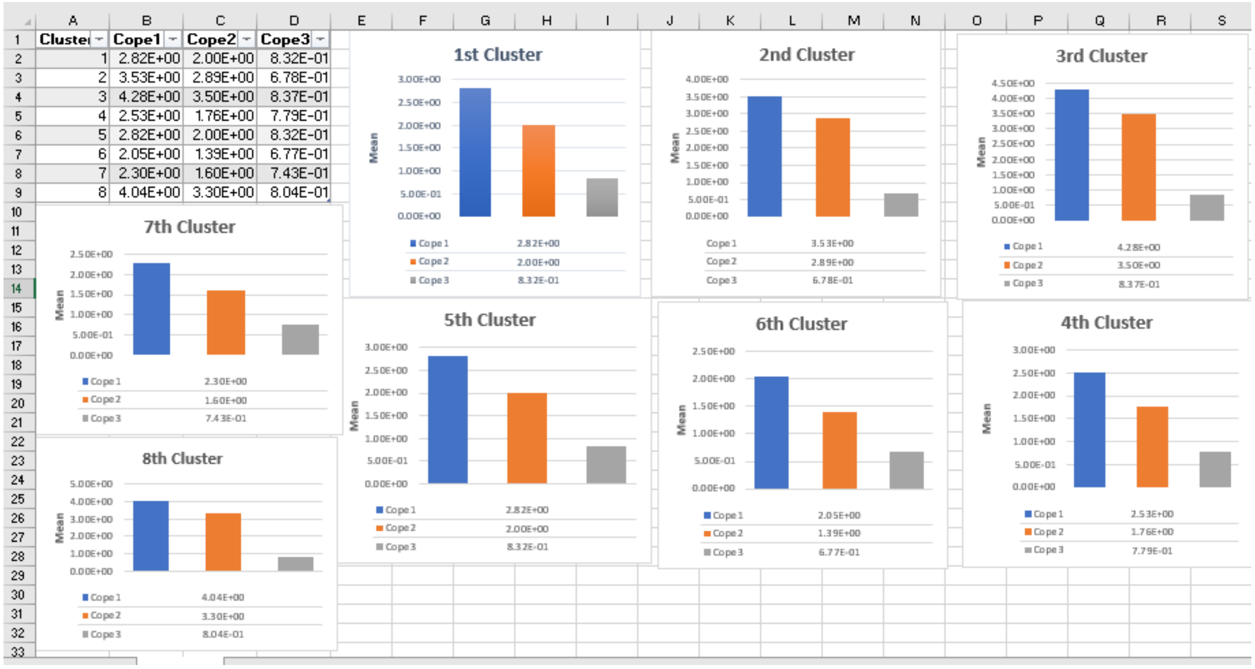
Finally, I used each cluster I got from third level analysis to create a sphere as a mask:



I used each sphere with the three copes (incongruent,congruent, incongruent-congurent) to extract data from it

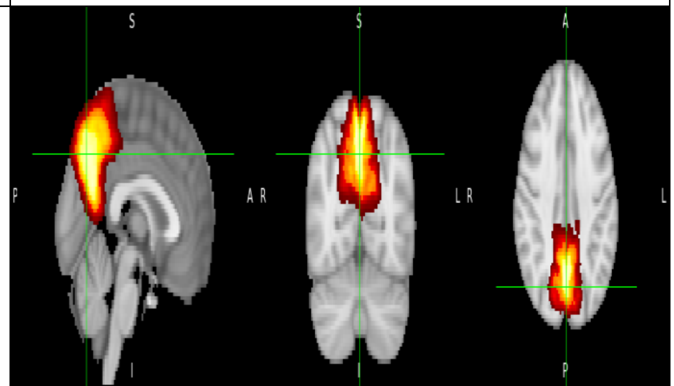
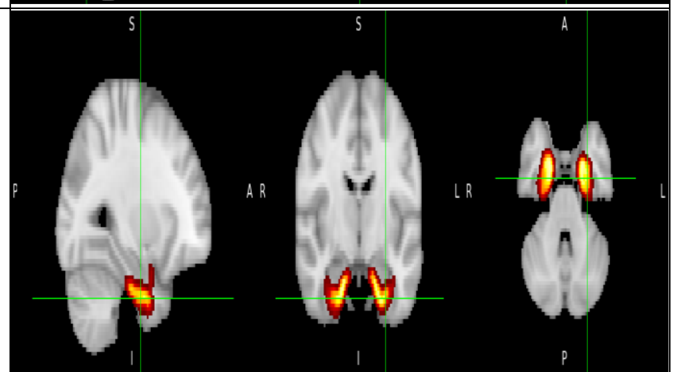
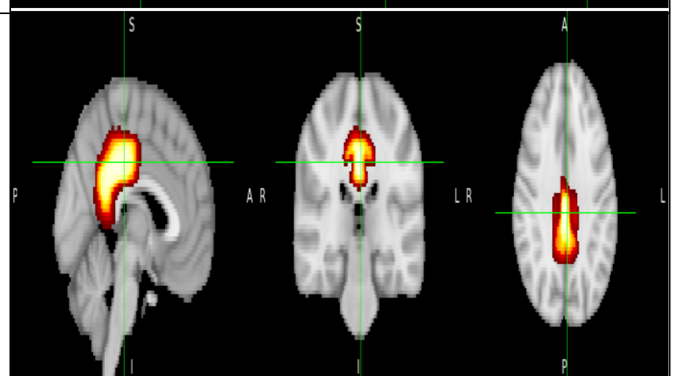
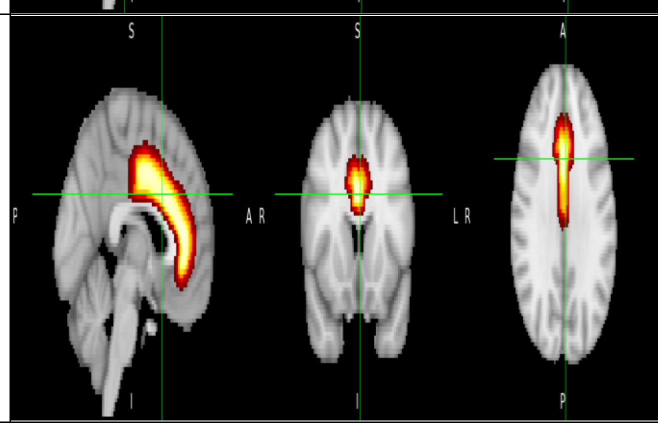
Some of my results:

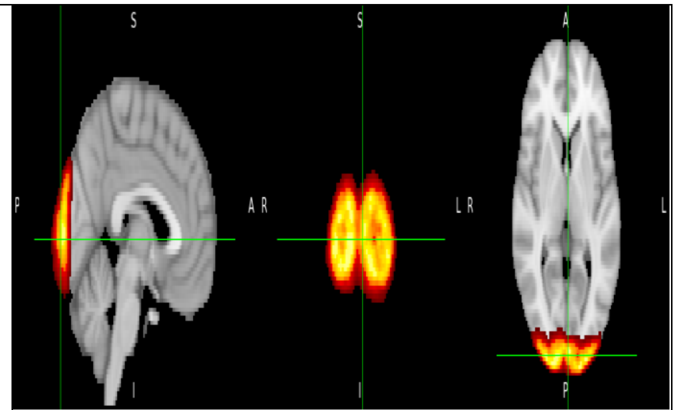
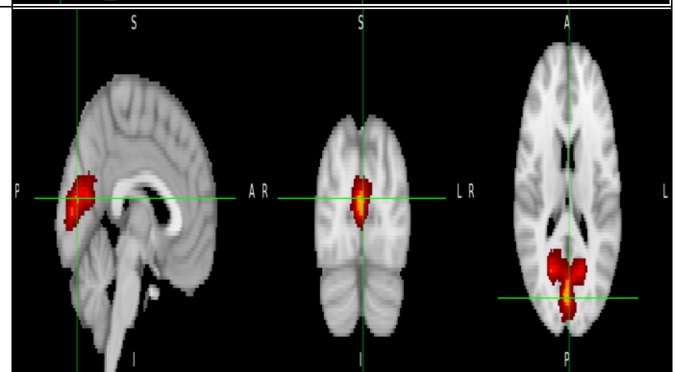
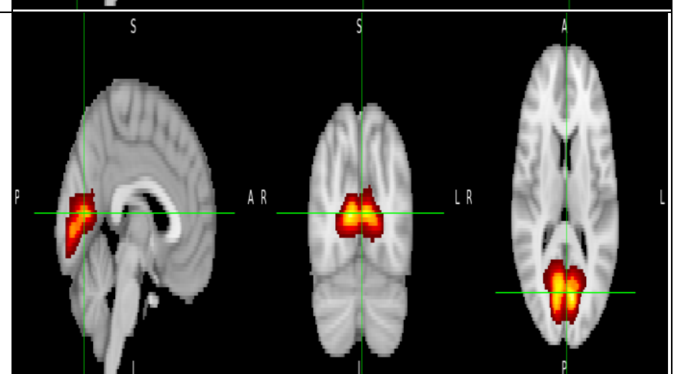
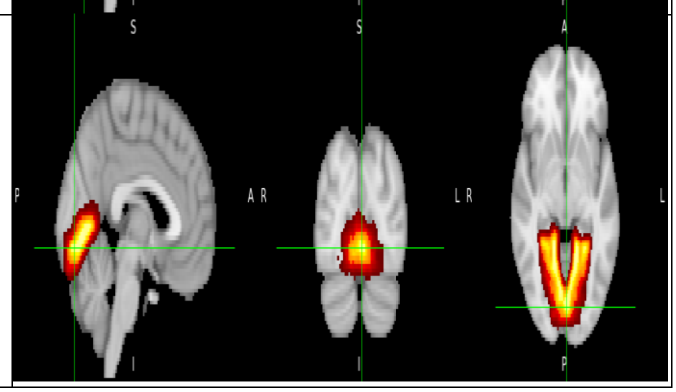
A	B	C	D	E	F	G	H	I	J	K	L
		Voxel Location [32 29 61]				Voxel Location [21 29 29]				Voxel Location [67 27 34]	
	Cope1	Cope2	Cope3		Cope1	Cope2	Cope3		Cope1	Cope2	Cope3
	2.743522	0.720022	1.993961		5.022633	3.252501	1.819283		3.79778	1.893527	2.011291
	2.576381	2.020141	0.607631		3.835656	3.30333	0.071693		4.99928	4.105269	0.50031
	-0.181285	-1.085655	0.314881		2.793671	1.867665	0.611083		0.822954	-0.578298	-0.273205
	3.602095	3.16986	0.437864		4.494892	4.017784	0.050034		3.746135	3.885009	1.140302
	4.42283	3.622796	0.563567		4.680125	3.905519	0.258415		7.606056	6.49975	-0.694439
	3.942885	2.47664	0.006096		4.365519	2.716707	0.543296		3.2595	2.671792	1.05902
	4.150867	1.798032	1.362259		4.05487	2.136414	1.886796		4.538495	3.258952	2.551152
	2.036318	0.549089	1.035328		2.734977	1.755961	0.934872		4.172685	2.414118	0.850895
	1.655613	0.608973	0.703392		3.230273	3.100693	0.391085		4.919298	3.990031	1.036943
	2.411496	1.775565	0.955238		1.890682	1.842753	-0.627518		2.537914	2.004225	0.046877
	2.528525	2.197906	0.241459		2.897791	2.300207	0.187885		4.876155	5.119007	0.480592
	4.705173	4.353986	0.504384		5.900183	5.968736	0.52584		3.400461	2.42655	0.824112
	2.932919	2.427591	-0.052491		2.289871	2.076923	0.655709		4.376484	5.107194	0.651376
	3.904322	3.81835	1.131123		4.27069	3.680854	1.008849		4.843715	3.688849	0.576037
	7.145189	5.918785	1.231526		8.453036	6.830771	0.229981		10.500436	8.368398	0.349804
	2.538755	1.423749	1.419315		2.637276	1.697683	1.422654		5.17402	4.332756	1.468598
	4.046027	3.419372	0.826168		4.766672	4.479745	0.319286		7.292085	6.457552	0.373691
	2.034181	1.161142	-0.020857		1.067549	1.790819	-0.372444		2.539622	2.529367	0.726695
	2.716453	2.422322	-0.402245		1.555777	1.327631	0.477559		5.157873	4.664706	0.43513
	2.077072	2.060886	0.948624		2.258287	1.508268	0.960647		4.378868	3.765974	1.336186
	2.273592	1.212178	0.371906		5.741624	4.897471	0.475848		3.099553	2.571483	-0.078713
	1.849164	0.621487	0.994857		3.677047	3.443127	0.887022		3.874472	3.502445	1.23117
	3.795107	2.270945	1.526757		3.134013	1.61475	1.639243		2.333417	0.740768	0.661301
	2.172485	1.320894	2.260105		2.924369	2.558734	2.024363		2.854494	2.448098	1.44258
	0.883714	0.944185	1.58143		0.383882	0.675751	1.050081		3.320222	2.60588	1.9915
	0.250919	0.666746	1.098844		2.823489	2.486315	0.207689		2.842228	2.498059	1.06649
P_Value	8.29E-10	2.56E-07	6.92E-07		8.87E-11	2.94E-10	2.47E-05		2.46E-11	7.32E-10	3.73E-06

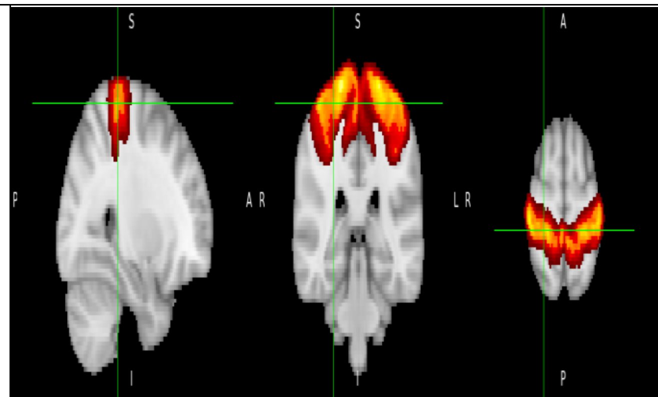
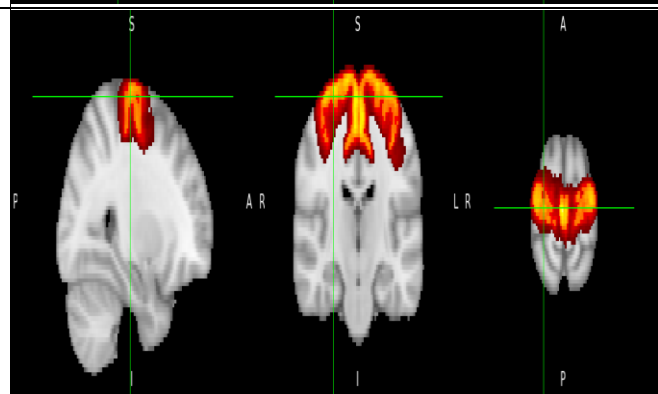
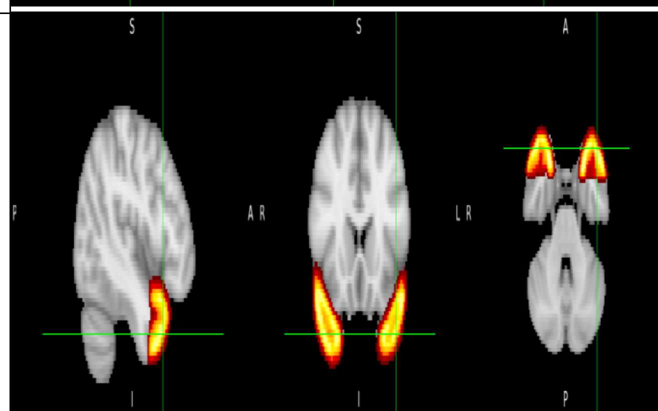
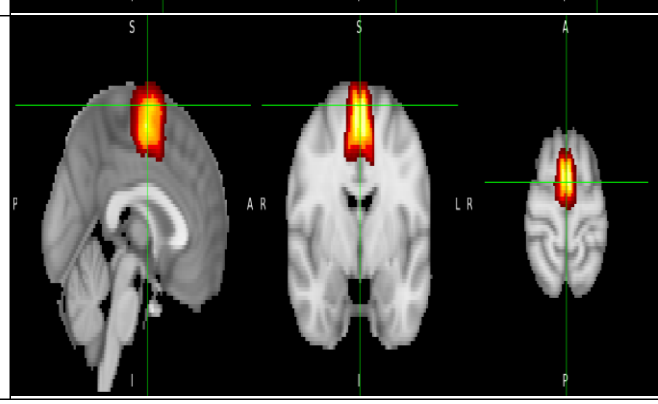


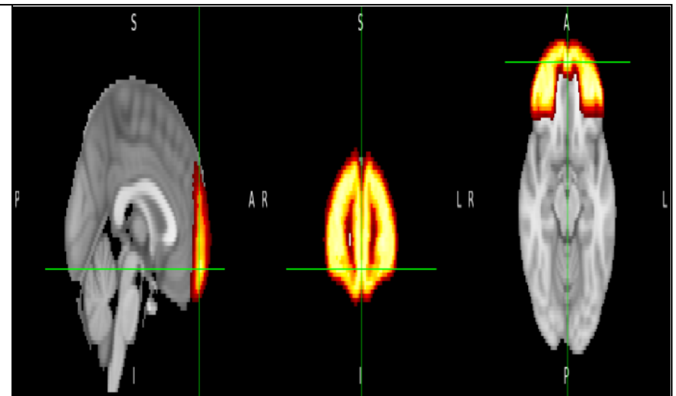
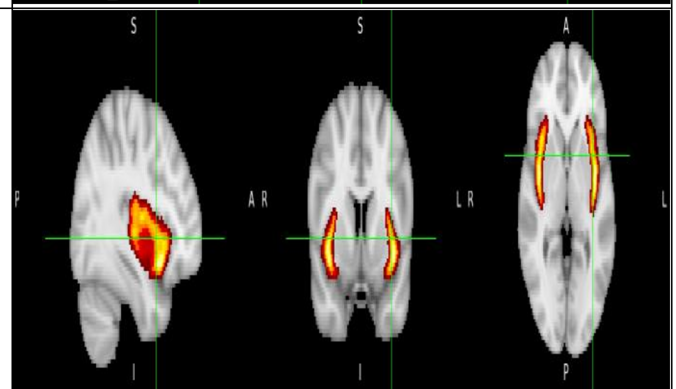
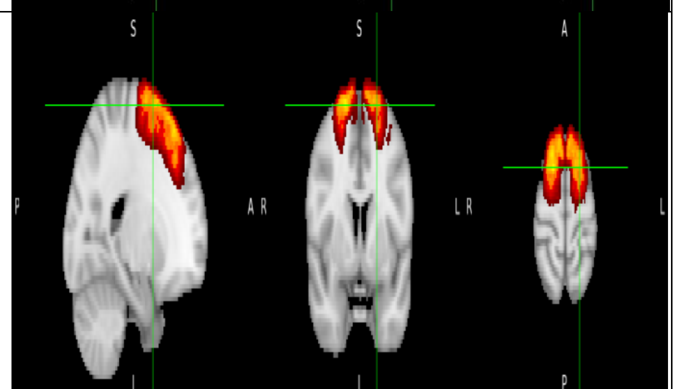
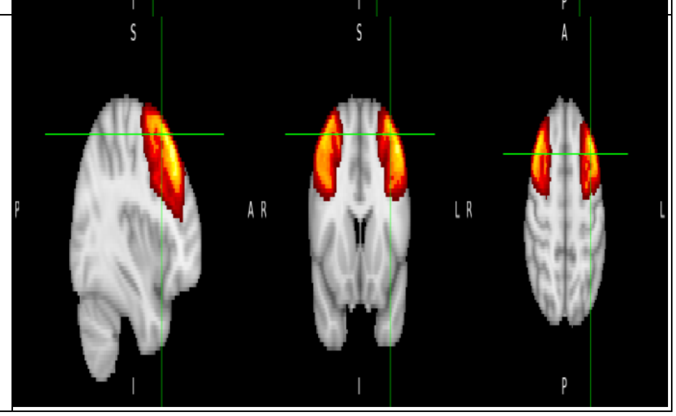
Snippet from used commands:

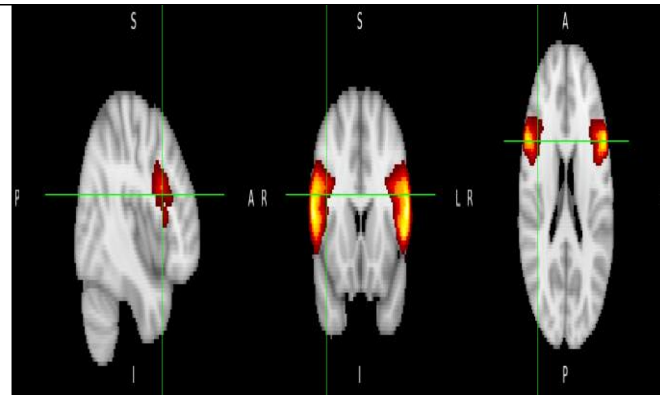
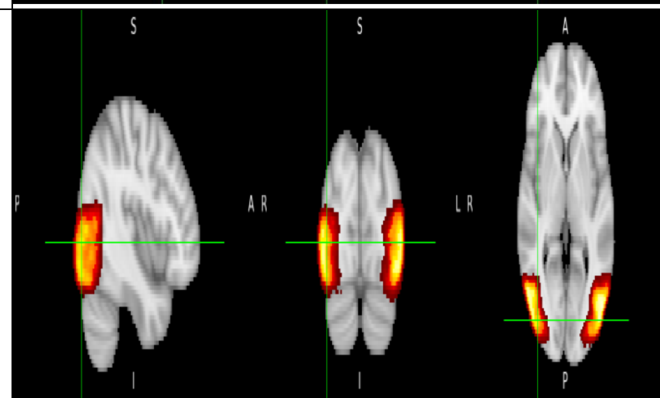
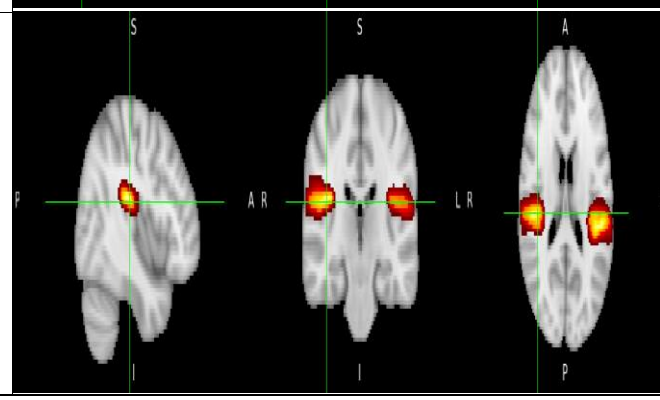
```
cd Flanker_2nd.gfeat/cope2.feat/stats
fslmerge -t allZstats1.nii.gz `ls zstat* | sort -V`
fslmerge -t allZstats2.nii.gz `ls zstat* | sort -V`

fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1 -roi 32 1 29 1 61 1 0 1 ROI_dmPFC_32_29_61.nii.gz -odt float

fslmaths ROI_dmPFC_32_29_61.nii.gz -kernel sphere 5 -fmean Sphere_dmPFC_32_29_61.nii.gz -odt float

fslmaths Sphere_dmPFC_32_29_61.nii.gz -bin Sphere_bin_dmPFC_32_29_61.nii.gz


fslmeants -i allZstats1.nii.gz -m Sphere_bin_dmPFC_32_29_61.nii.gz
fslmeants -i allZstats2.nii.gz -m Sphere_bin_dmPFC_32_29_61.nii.gz
fslmeants -i allZstats.nii.gz -m Sphere_bin_dmPFC_32_29_61.nii.gz

##############

fslmaths $FSLDIR/data/standard/MNI152_T1_2mm.nii.gz -mul 0 -add 1 -roi 21 1 29 1 29 1 0 1 ROI_dmPFC_21_29_29.nii.gz -odt float

fslmaths ROI_dmPFC_21_29_29.nii.gz -kernel sphere 5 -fmean Sphere_dmPFC_21_29_29.nii.gz -odt float

fslmaths Sphere_dmPFC_21_29_29.nii.gz -bin Sphere_bin_dmPFC_21_29_29.nii.gz


fslmeants -i allZstats1.nii.gz -m Sphere_bin_dmPFC_21_29_29.nii.gz
fslmeants -i allZstats2.nii.gz -m Sphere_bin_dmPFC_21_29_29.nii.gz
fslmeants -i allZstats.nii.gz -m Sphere_bin_dmPFC_21_29_29.nii.gz
```

	Some regions in Cortical	
Region	**Function**	**MNI Image**
Precuneous Cortex	1- Memory and recollection: it is involved in memory processes and plays a role in the retrieval of episodic memories. 2- Integration of information and perception: it is involved in integrating sensory information to form a coherent perception of the environment.	
Parahippocampal Gyrus, anterior division	1- Memory encoding and retrieval as it plays a critical role in the formation and retrieval of memories. 2- Spatial navigation as it is involved in spatial navigation, allowing us to navigate and orient ourselves in our environment.	
Cingulate gyrus/posterior division	1- Autobiographical memory recall: especially memories with emotional content. 2- Regulating the balance between internally and externally-focused attention: Its involvement in the default mode network, which is more active during rest or when not engaged in external attention tasks.	
Cingulate gyrus/anterior division	1- it contributes to the regulation of autonomic and endocrine responses, pain perception, and the selection and initiation of motor movements. 2- It is involved in various aspects of cognition ranging from decision-making to the management of social behavior.	

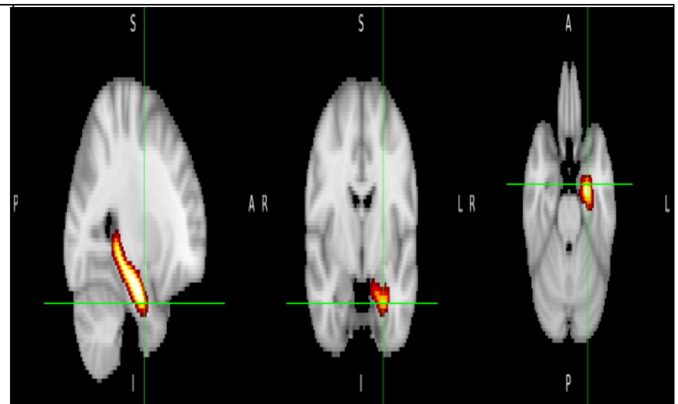
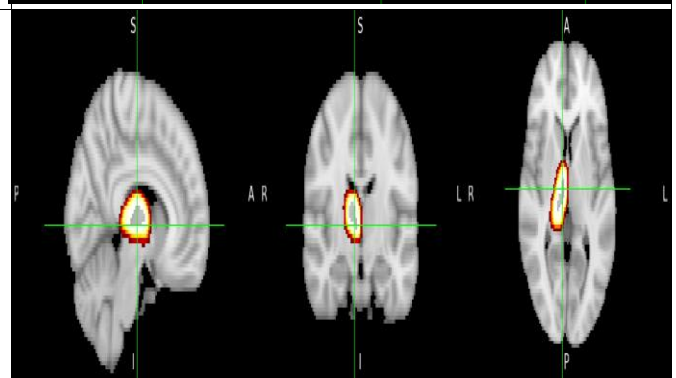
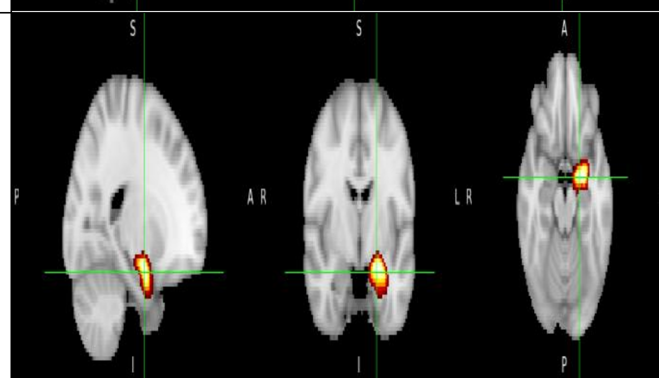
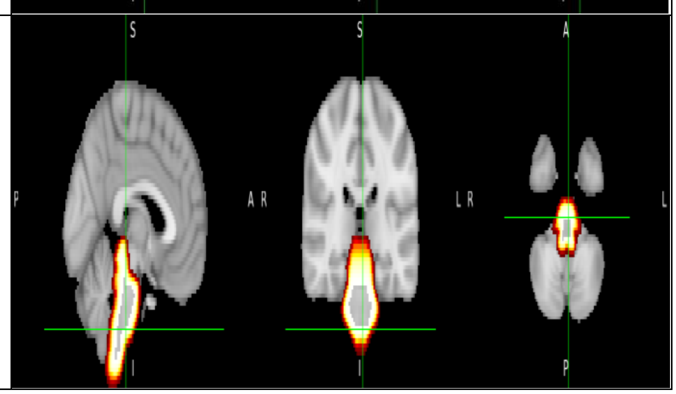
Occipital pole	1- Object and face recognition: This is your brain's ability to recognize things you've seen before, including the faces of people you've seen or met. 2- Spatial processing: This is your brain's decoding of signals from your retinas. It's how you see the shapes, textures and other details of the objects in the world around you.	
Supracalcarine Cortex	1- Integration of visual information: it receives and integrates information from the primary to create more complex and sophisticated visual representations of the environment. 2- Perception of visual stimuli: It helps to identify and distinguish between different objects and their properties, such as shape, size, texture, and movement	
Intracalcarine cortex	1- Analysis of visual properties: it plays a key role in analyzing various visual properties such as orientation, spatial-frequencies, and color of visual stimuli. 2- Projection of visual information: After analyzing visual information,it projects this information to other areas in the occipital lobe for further visual analysis.	
Lingual Gyrus	1- It is involved in the identification and recognition of words. Therefore, it plays a crucial role in language processing. 2- It is also implicated in modulating visual stimuli, particularly letters, indicating it may play a role in visual perception and processing.	

Postcentral Gyrus (primary somatosensory cortex)	1- Somatosensory processing: it receives the majority of the somatic sensory relay information from the thalamus. 2- Sensory integration: involves integrating sensory information from different parts of the body and other modalities such as vision and hearing to create a more complete perception of the environment.	
Precentral Gyrus (primary motor cortex)	1- It is involved in executing voluntary motor movements as It receives input from other parts of the brain and sends out signals to control the body's muscles. 2- it is involved in planning and coordinating movements,It helps to organize motor programs and select the appropriate muscles to use for a given movement	
Temporal lobe	1- Controlling unconscious and apparently automatic reactions, such as appetite, thirst, hunger. 2- Helping the body maintain homeostasis.	
Juxtapositional Lobule Cortex (formerly Supplementary Motor Cortex)	1- Coordination of distal and proximal limb Movement. This involves the integration of sensory information and the initiation of motor commands to produce smooth and coordinated movements. 2- It is more active than the primary motor cortex during complex or challenging motor tasks that require greater coordination and planning.	

Frontal Pole	Voluntary movement control as it is responsible for controlling voluntary movement, which involves the planning, initiation, and execution of movements. Executive function control as it involved in managing higher level executive functions. These functions include the capacity to plan, organize, and control one's responses in order to achieve a goal.	
Insular Cotex	1- Self-awareness as it plays a crucial role in generating our sense of self, by integrating sensory information from the body and linking it to emotional states by connection with other brain regions such as the anterior cingulate cortex and the medial prefrontal cortex. 2- it is involved in decision-making processes	
Superior Frontal Gyrus	1- Working memory: This is the ability to hold and manipulate information in our minds over short periods of time. 2- Cognitive control: This refers to the ability to control our thoughts and actions in order to achieve a goal, involving in cognitive control tasks includes suppressing irrelevant information or inhibiting impulsive responses.	
Middle Frontal Gyrus	1- The dominant (left) middle frontal gyrus is involved in the development of literacy, while the nondominant (right) one is responsible for numeracy. 2- The caudal portion of the middle frontal gyrus, specifically the frontal eye fields is responsible for controlling saccadic eye movements.	

Inferior Frontal Gyrus, pars opercularis	1- Language production as it is involved in the planning and execution of speech movements. 2- Inhibitory control as it is involved in inhibitory control, which is the ability to suppress inappropriate responses or thoughts	
Lateral Occipital Cortex, inferior division	1- It is involved in visual perception and object recognition. 2- it is also involved in processing information related to facial recognition, so it is important in social interactions.	
Parietal Operculum Cortex	1- Plays a role in spatial awareness and the ability to navigate and interact with the environment. 2- it is involved in processing tactile information from the skin, including pressure, temperature, and pain, as well as proprioceptive information about limb and joint position.	

	Some regions in subcortical	
Region	Functions	MNI image

Left Hippocampus	1- Memory formation and retrieval. 2- Spatial navigation in creating cognitive maps of environment.	
Right Thalamus	1- Involved in processing and integration of sensory information that is used in decision making. 2- plays a role in integrating and modulating the sensory information before it reaches the cortex.	
Left Amygdala	1- its role in emotional processing is complex and involves a range of different functions, as encoding of emotional memories, and the regulation of emotional responses. 2- It plays a role in modulating memory storage	
Brain Stem	1- Regulation of vital functions of the body, including breathing, heart rate, blood pressure. 2-Relay center for sensory and motor information, as It contains several nuclei that receive and transmit information to and from different parts of the brain and spinal cord	

Link for material I used:

https://drive.google.com/drive/folders/1p2sYP83XwJIH9VHr-dsD8EMfGlSsFMxM?usp=drive_link