Functional Magnetic Resonance Imaging

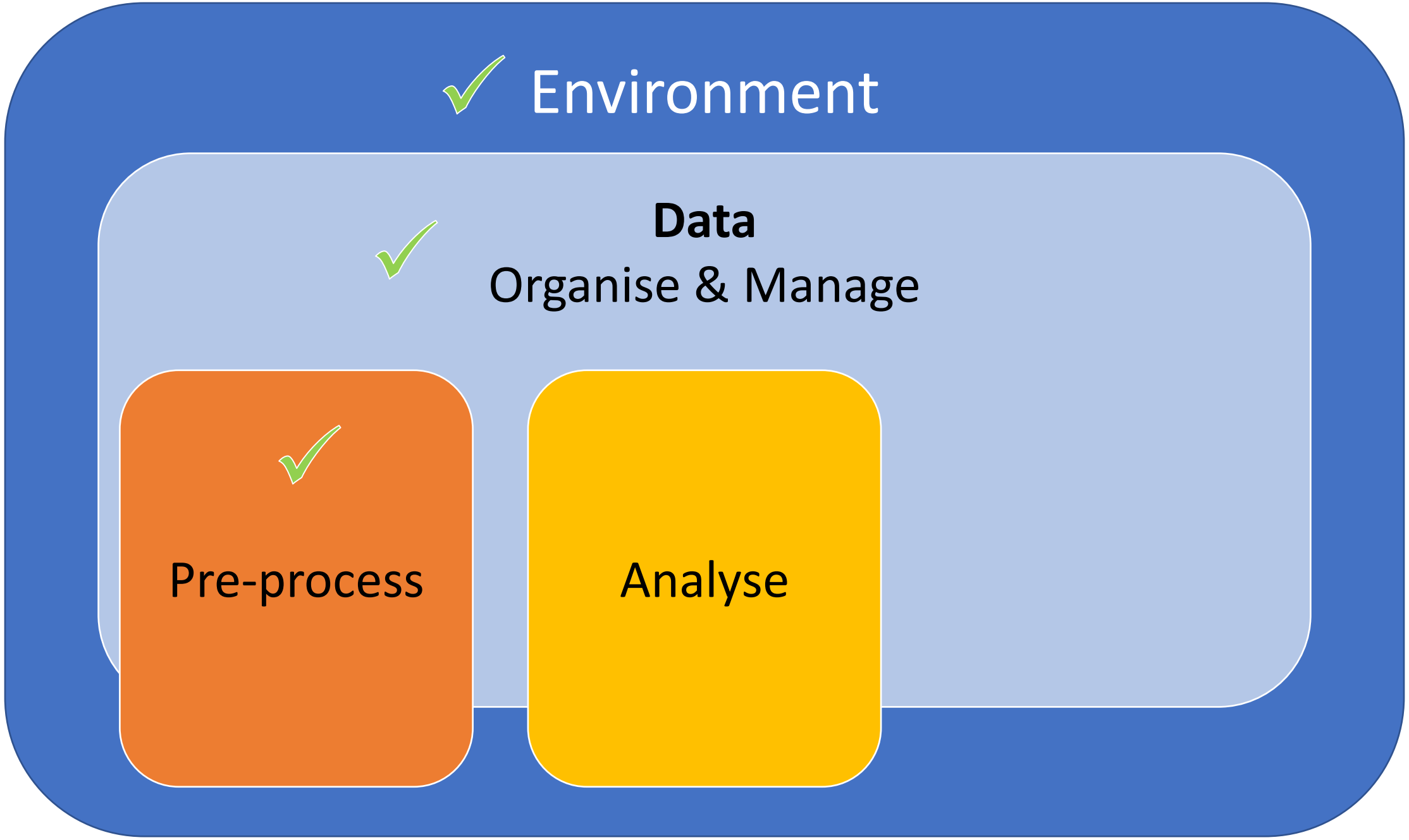
https://github.com/dcdace/fMRI_training

Dace [datza] Apšvalka

February 2026

Outline

- Introduction
- Experimental design
- Data management
- Pre-processing
- **Statistical analysis**
- Practical demo


Environment
Data
Organise & Manage
Pre-process
Analyse

Faces vs **Scrambled** images are processed differently in the brain

Design an experiment

Data

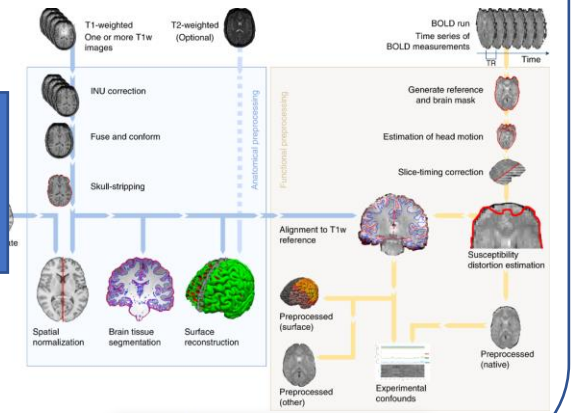
Stimuli
Timing

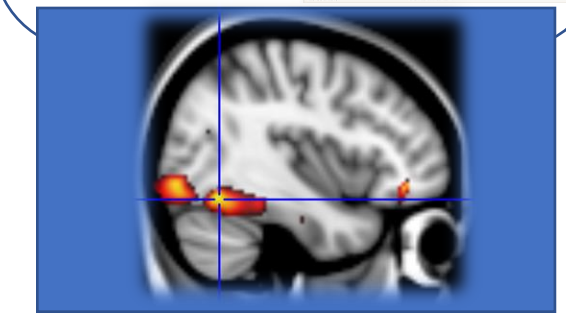
Collect the MRI data

Data

Anatomical image
Functional images
Event details

Pre-process & Analyse





fMRI Aanalysis

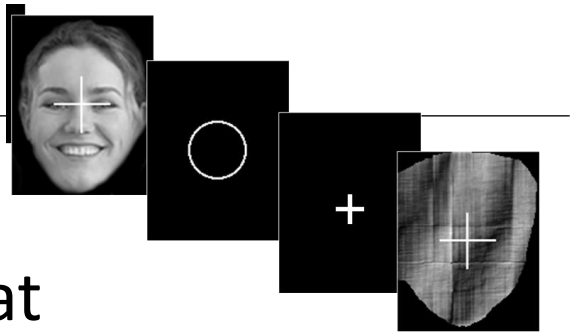
The General Linear Model (GLM)

Largely based on Idan Blank's materials

https://cbmm.mit.edu/videos?field_video_grouping_tid%5B%5D=770

Finding a Face area in the brain

Which brain regions are engaged when people look at faces

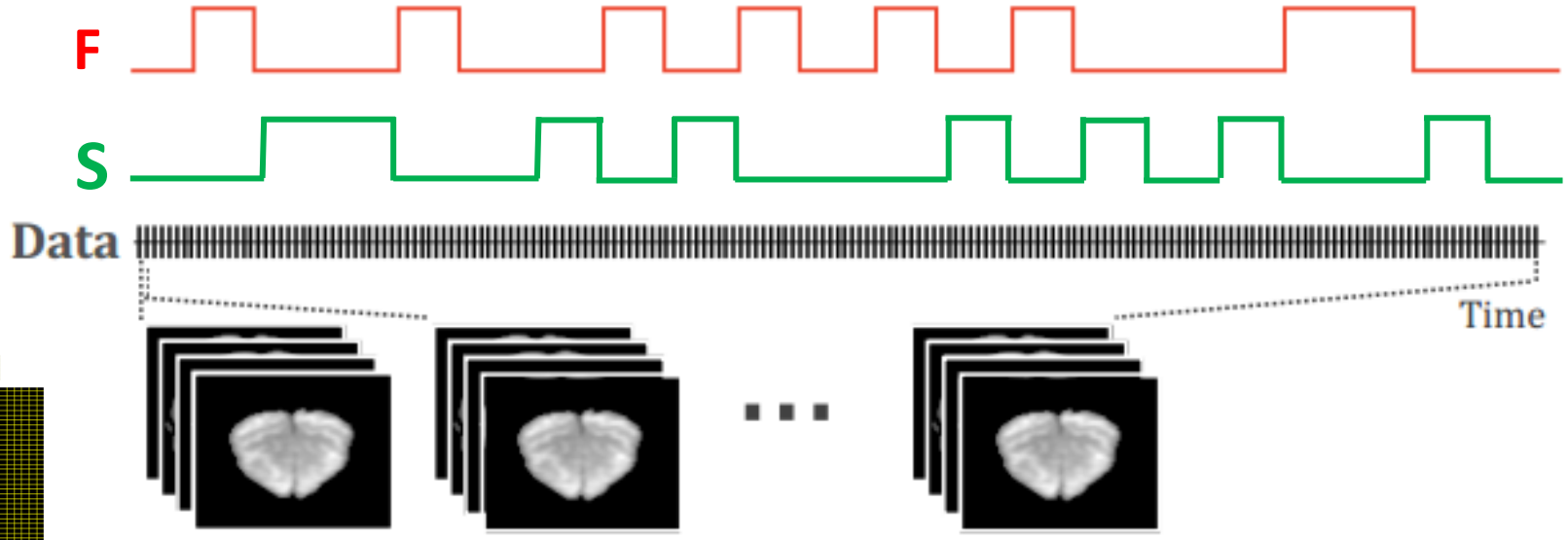


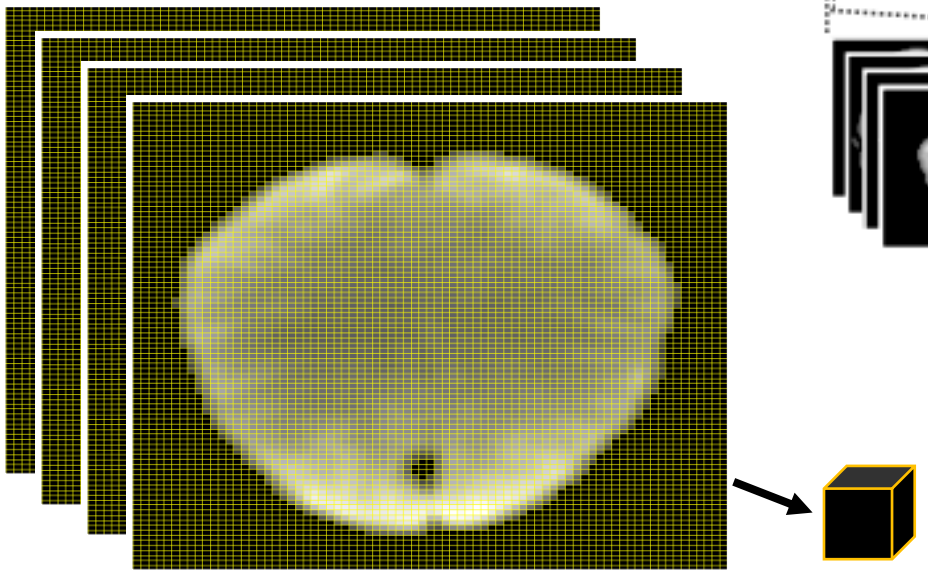
- With fMRI the meaningful questions are questions that **compare two conditions**
  - We need some sort of control condition –> **Scrambled** condition

- **Which brain regions respond more to looking at face images than scrambled images**
  - The control question hopefully helps to **wash out all the regions we are not interested in**. Because regions that we are NOT interested in should activate both conditions to the same extend (e.g. visual areas)

The structure of our data

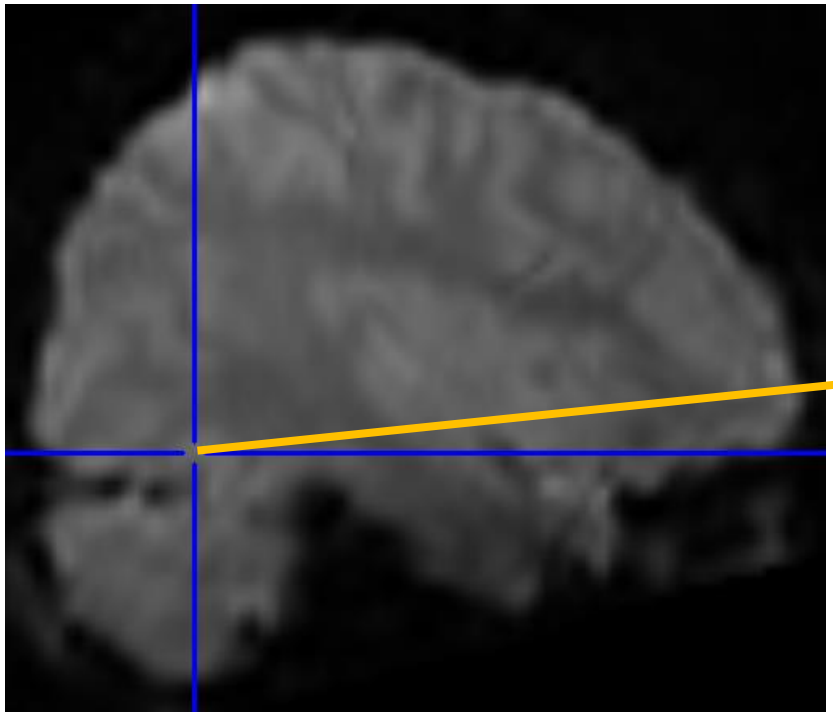
- **Question:** Which brain regions respond more to face images than to scrambled images
- **Experiment**

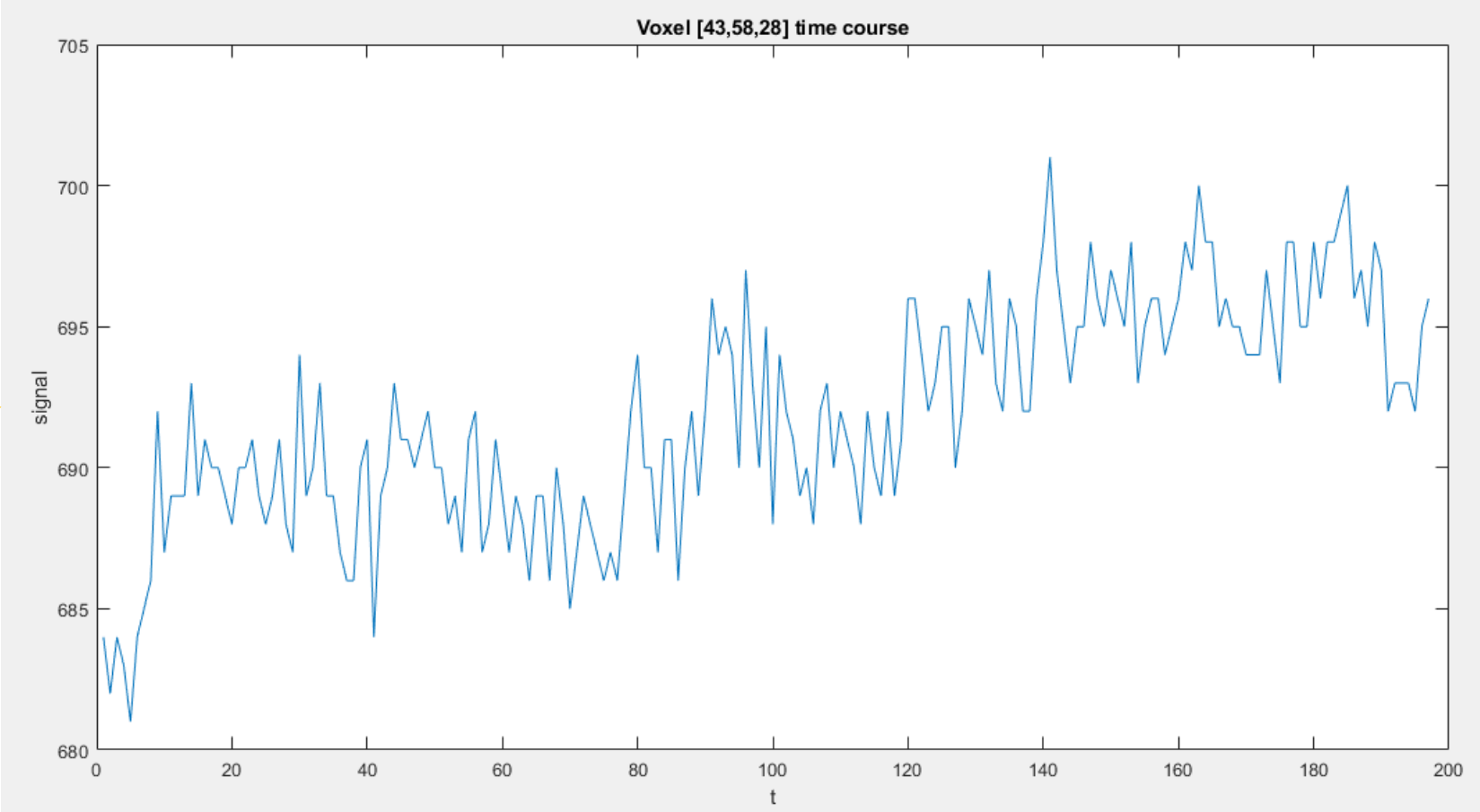




The brain data that we collect are a stack of 2D slices that form a 3D volume -> a large 3D cube with it's smallest unit –> a small 3D cube -> **a voxel**

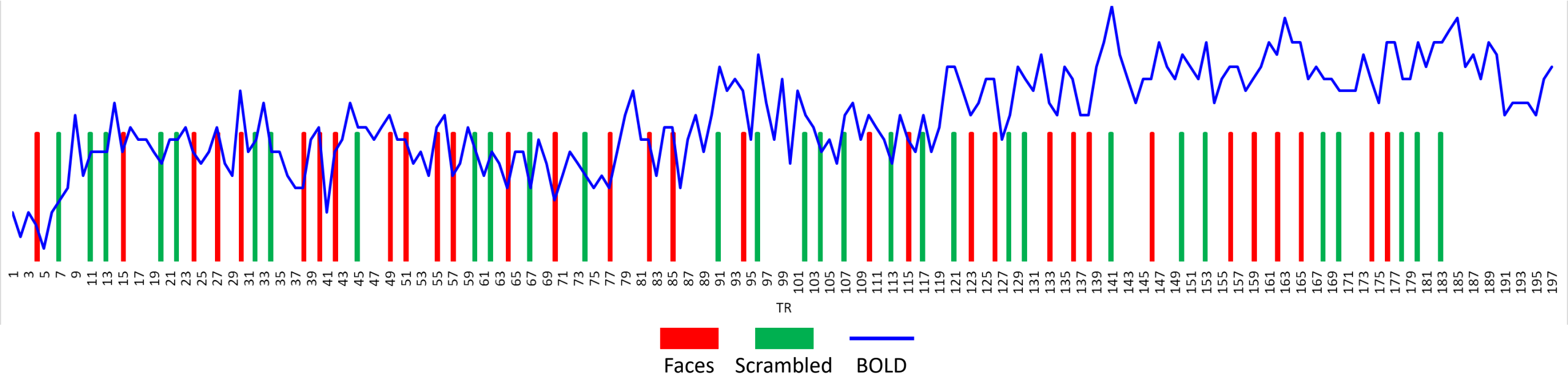
Voxel time course



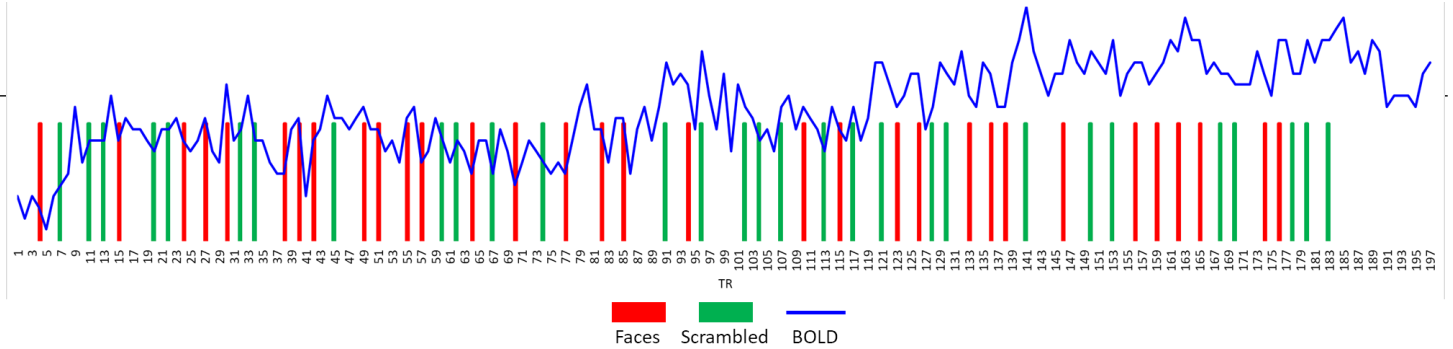


Question: Which brain regions respond more to face images than to scrambled images

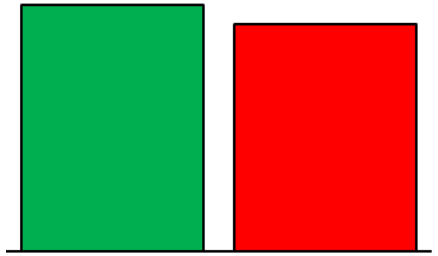
- Event onsets and BOLD signal in a sample voxel
- **How do we analyse this?**



Question: Which brain regions respond more to face images than to scrambled images



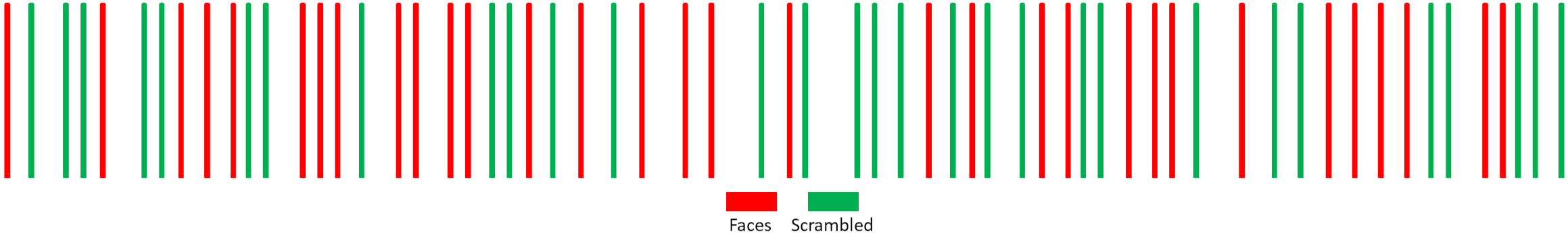
- A raw BOLD signal is noisy
- Let's start with an intuitive way, and that's how people started to analyse fMRI data
- **Analysis:** an intuitive approach
  1. For each voxel, look at its signal time-series (activity across time)
  2. Average the signal across volumes that were collected while participants were looking at Faces
     - Mean Faces = 691.92, Mean Scrambled = 692.35, F-S = -0.43



Why does the averaging method does not work?

Question: Which brain regions respond more to face images than to scrambled images

- **Stimuli**



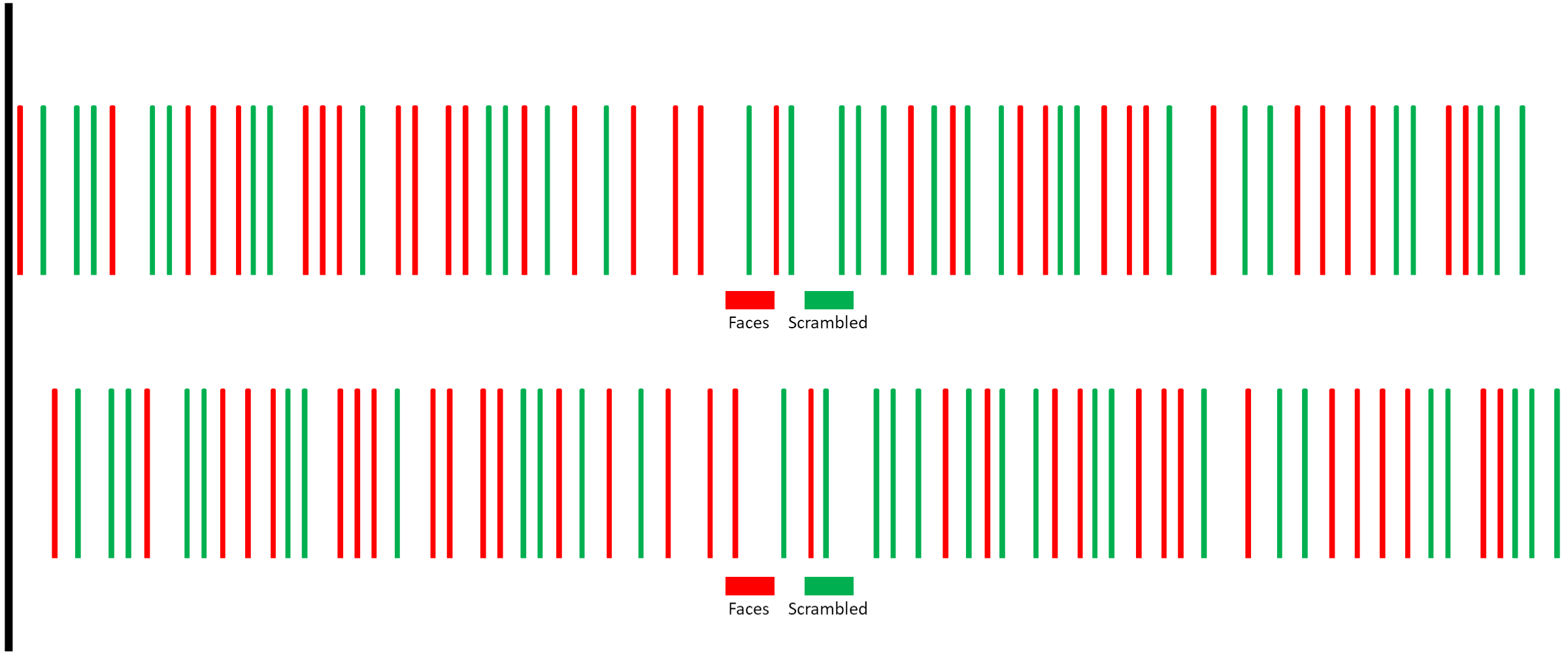
- **Neural activity**
  - Starts briefly after the onset of the stimulus
- **BOLD signal** (not a direct measure of neural activity)
  - When neurons are active, the fMRI BOLD signal will rise _____

A. Immediately
B. 1-3 seconds after the neural activity
C. 6-12 seconds after the neural activity
D. 25-30 seconds after the neural activity

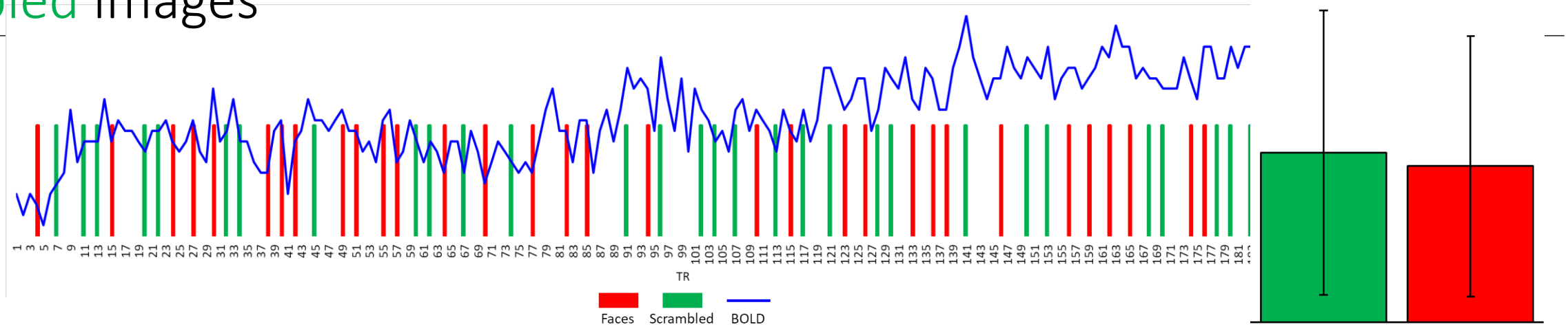
We were averaging the wrong points! The points in time when the 'Face area' would respond are actually little later than the onset of the stimulus; about 6-12 s later then the onset.

Question: Which brain regions respond more to face images than to scrambled images

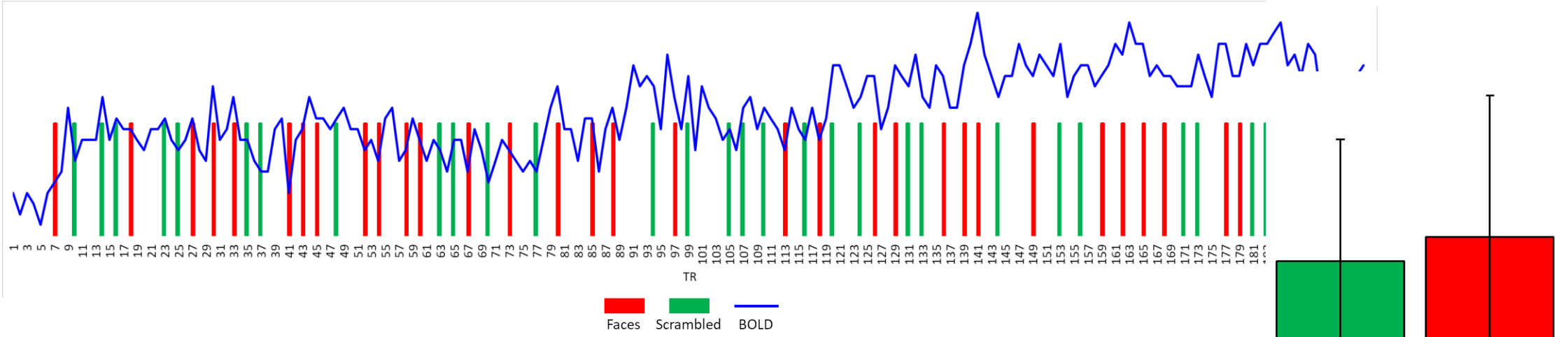
The points that we are looking at should be shifted in time

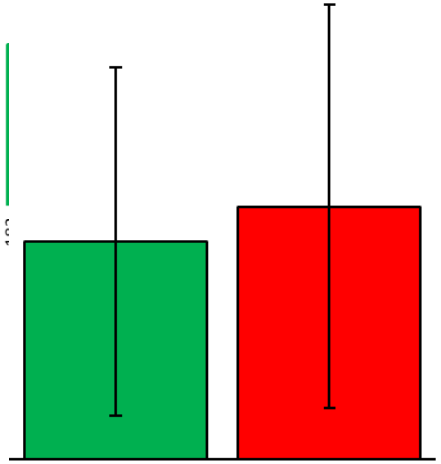


Question: Which brain regions respond more to face images than to scrambled images



+6s shift



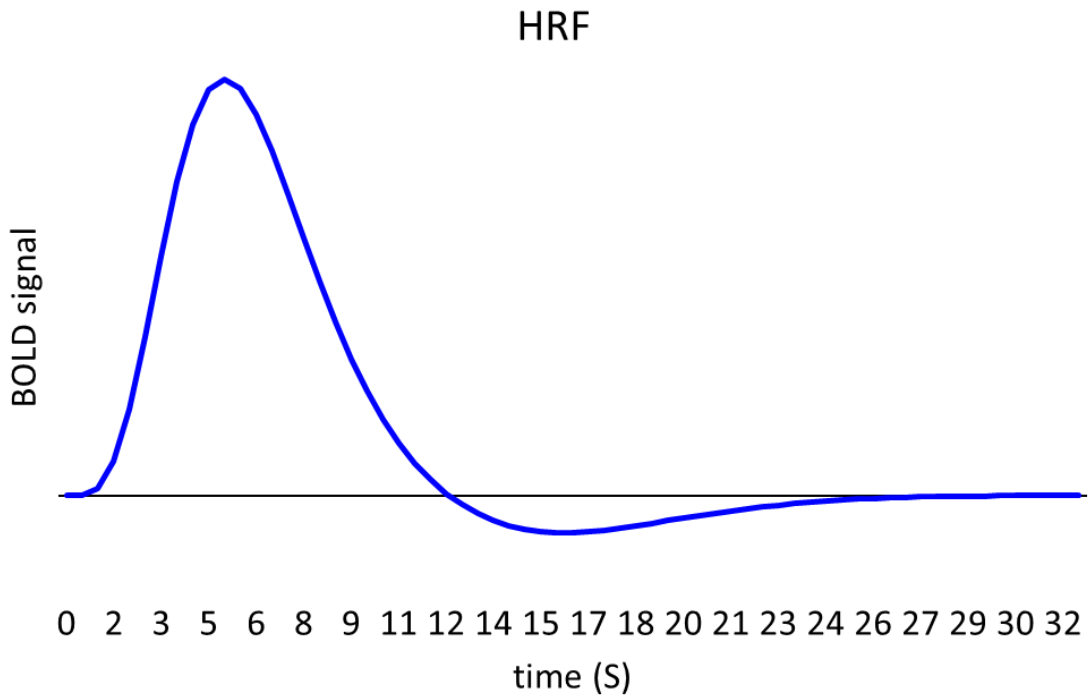


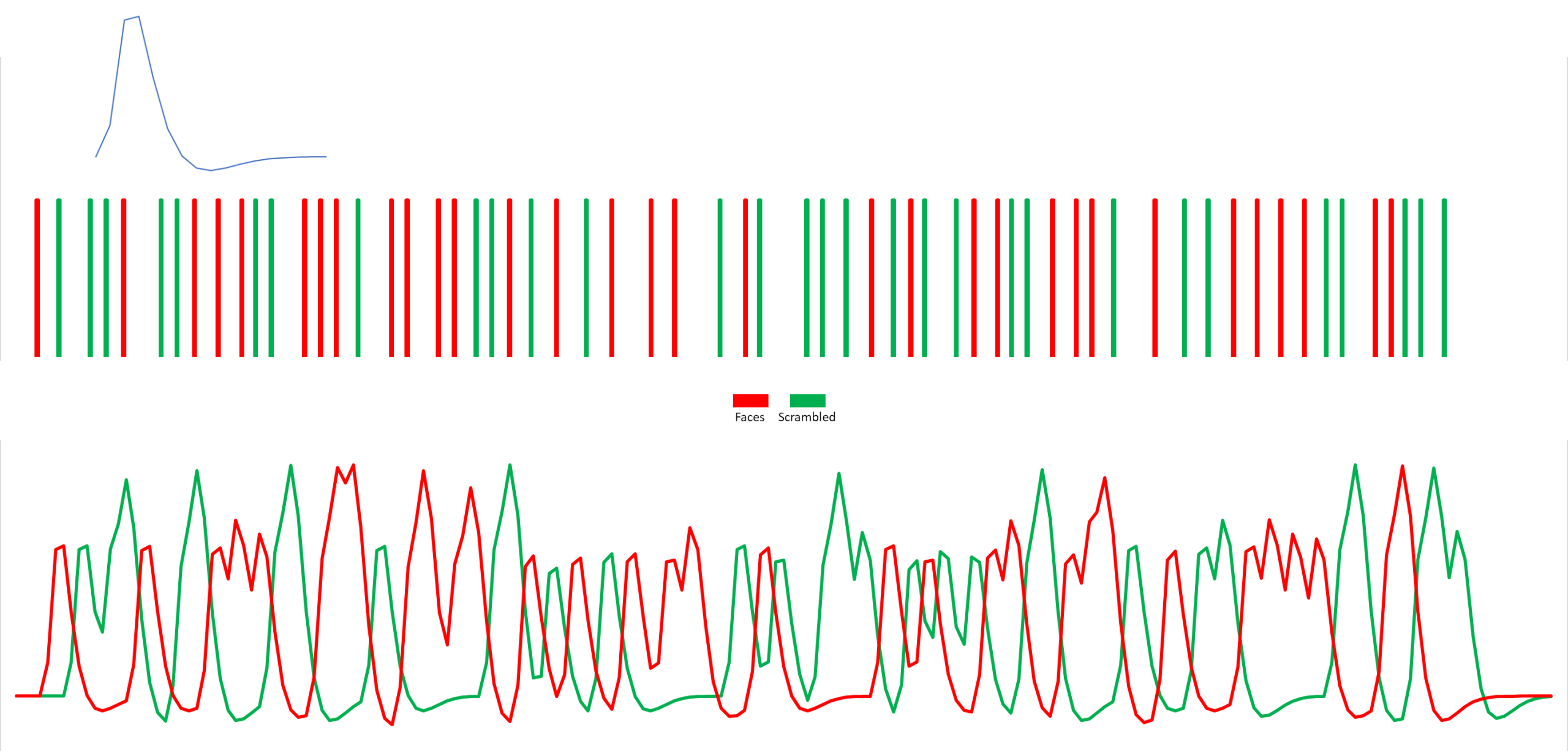
Looks slightly better, but still not working!

Bold signal does not change from 0 to 1 (on or off)!

Question: Which brain regions respond more to face images than to scrambled images

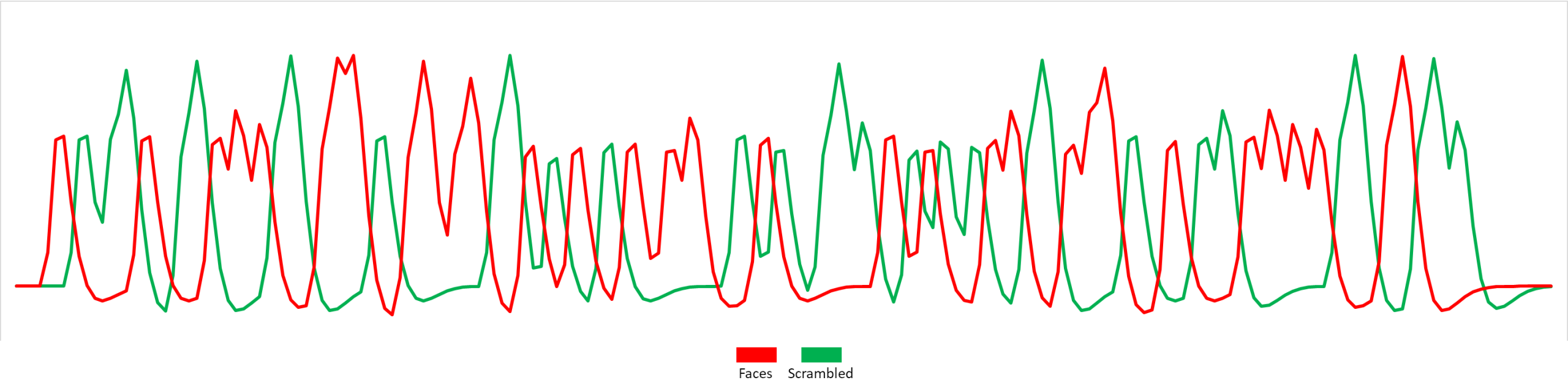
- The haemodynamic response function looks like this
- It is not binary, but smooth
- We need to make our prediction to look more like this
- How do we do that? This is done by mathematical process called **convolution**
- It is filtering the signal to look more like this shape



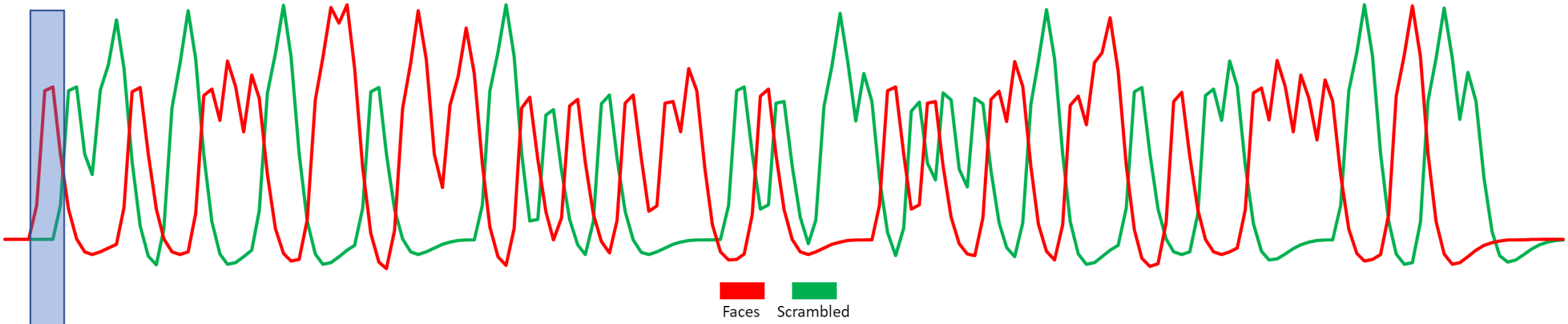


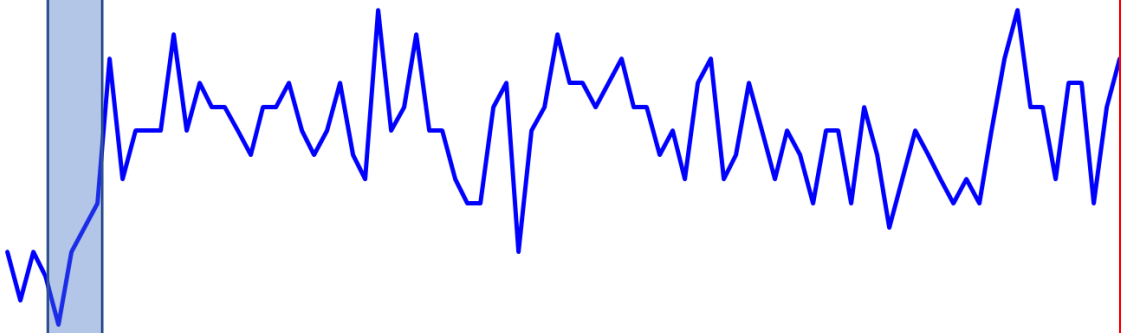
These are our BOLD predictions. **This is how we should expect our signal to look like!**

Question: Which brain regions respond more to face images than to scrambled images



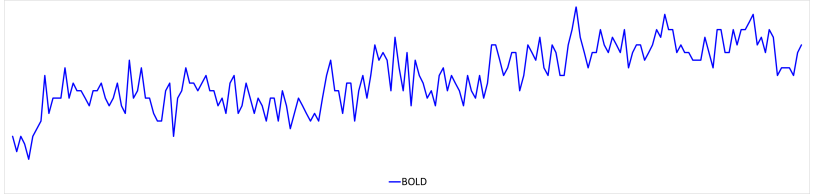
The time series of a voxel that responds more to Faces than to Scrambled should look like a combination of the two signals above, with Faces having more weight than Scrambled.



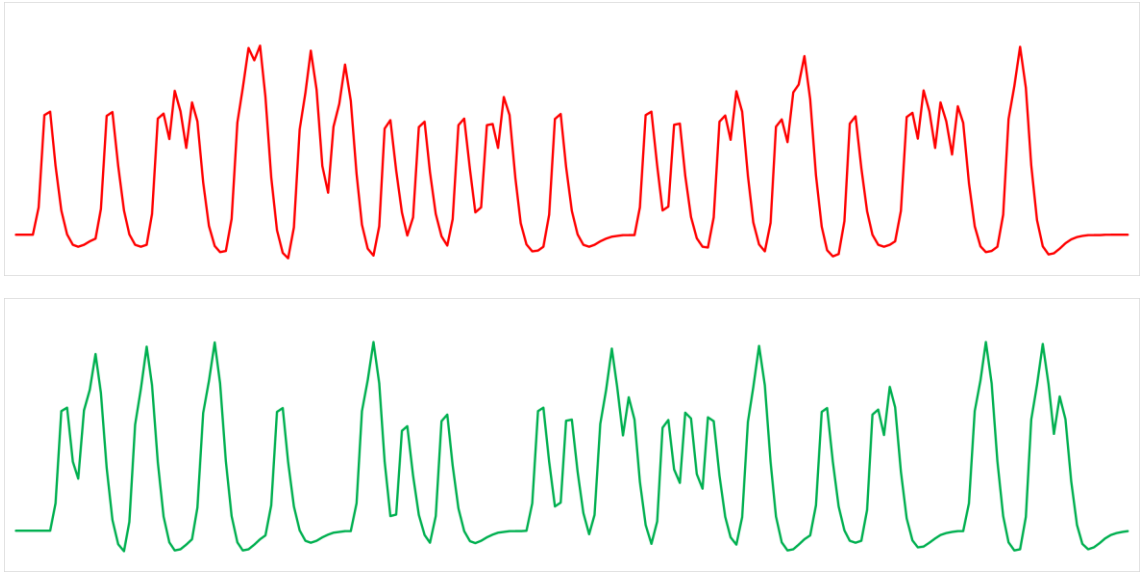


- **Which points should we average?**
- Averaging does not seem to be the best idea anymore.
- We need a more principled solution, not an arbitrary choice of points!
- Voxels in the ‘Face area’ should respond **a lot to** Faces, but very little to Scrambled images.

(de)Constructing the BOLD signal



- We can approximate the signal time-series of a voxel by **combining these 3 signals**:
  - Baseline signal:
    - an average activity in that voxel when you are doing nothing
  - Response to Faces:
  - Response to Scrambled:

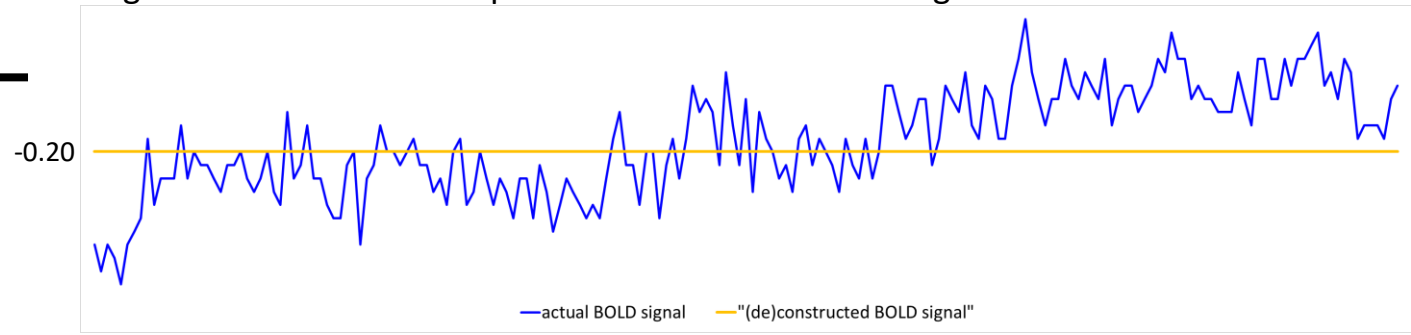


(de)Constructing the BOLD signal

The true BOLD signal
Deconstructed BOLD signal

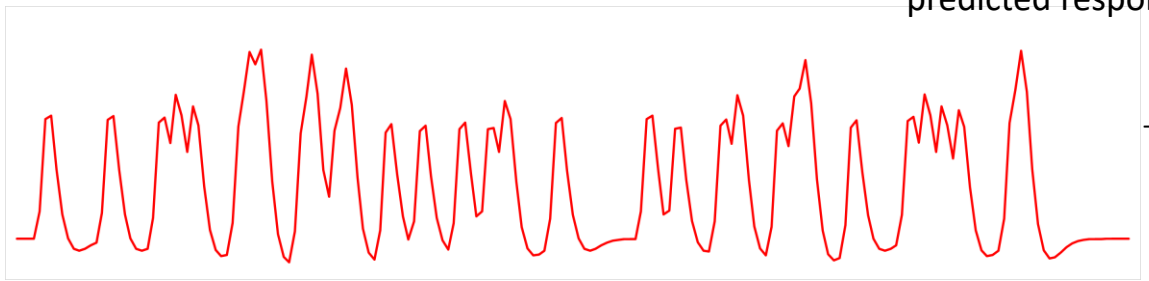
1. Let's say the average signal at baseline was -0.2. We add this baseline measure to our deconstructed BOLD signal and see how it compares with the actual BOLD signal.

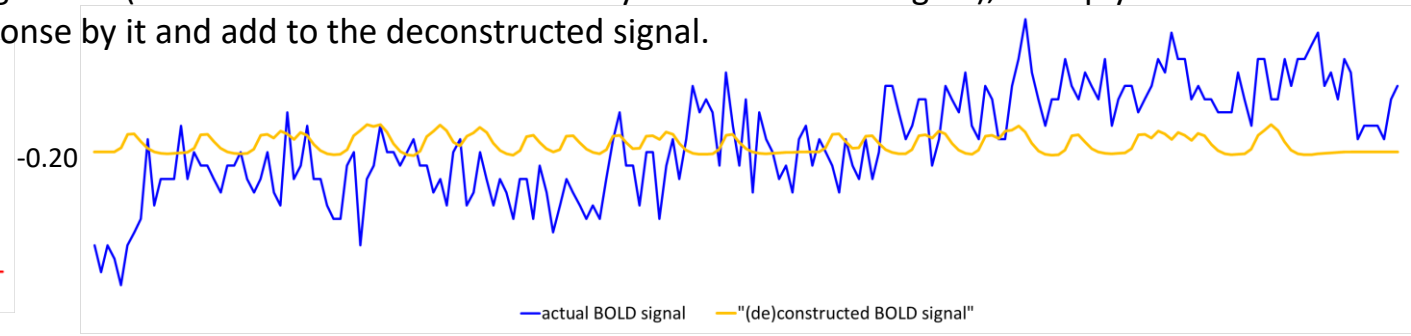
Baseline -0.20



2. We predict that in this voxel Faces condition contributes quite a lot to the true BOLD signal. Let's assign it a weight 0.89 (we will find out later how exactly we find these weights), multiply the predicted response by it and add to the deconstructed signal.

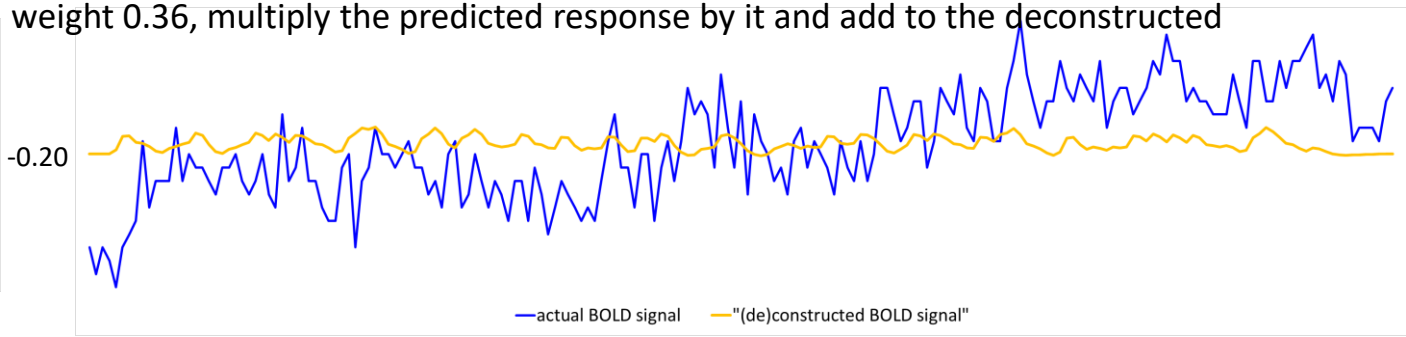
\+ **Faces** 0.89





3. We predict that in this voxel Scrambled condition has little contribution to the true BOLD signal. Let's assign it a weight 0.36, multiply the predicted response by it and add to the deconstructed signal.

\+ **Scrambled** 0.36

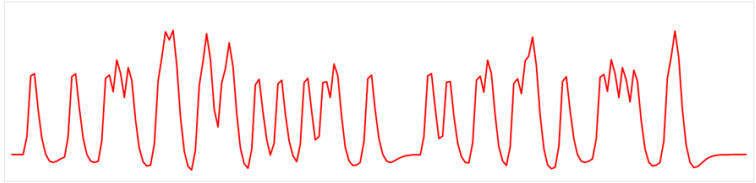


(de)Constructing the BOLD signal

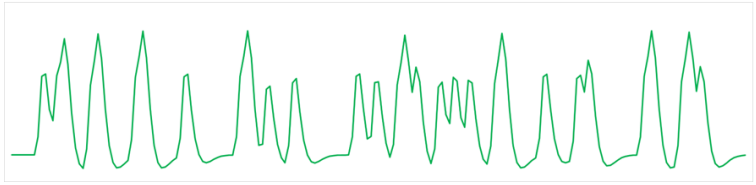
The true BOLD signal
Deconstructed BOLD signal

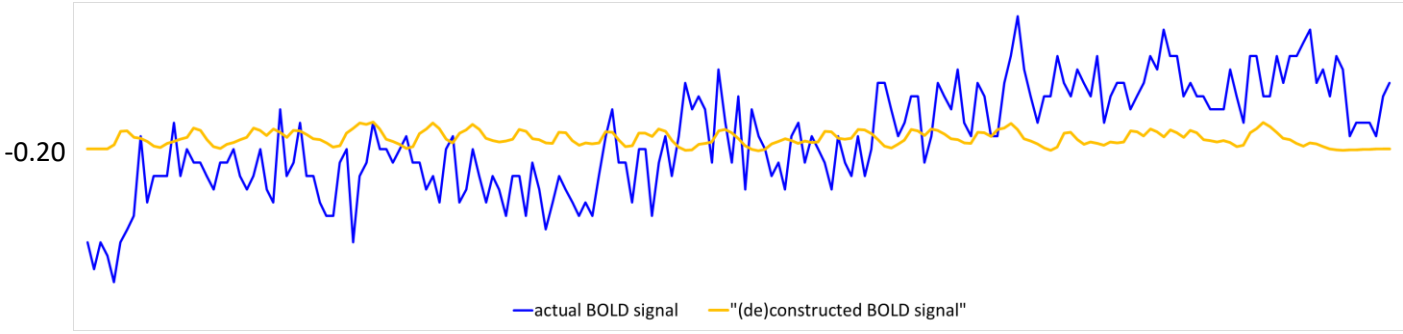
Baseline * -0.20

+ **Faces** * 0.89



+ **Scrambled** * 0.36

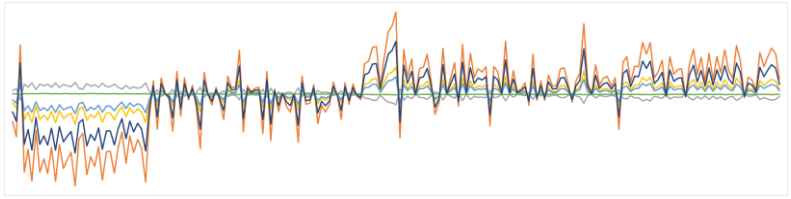




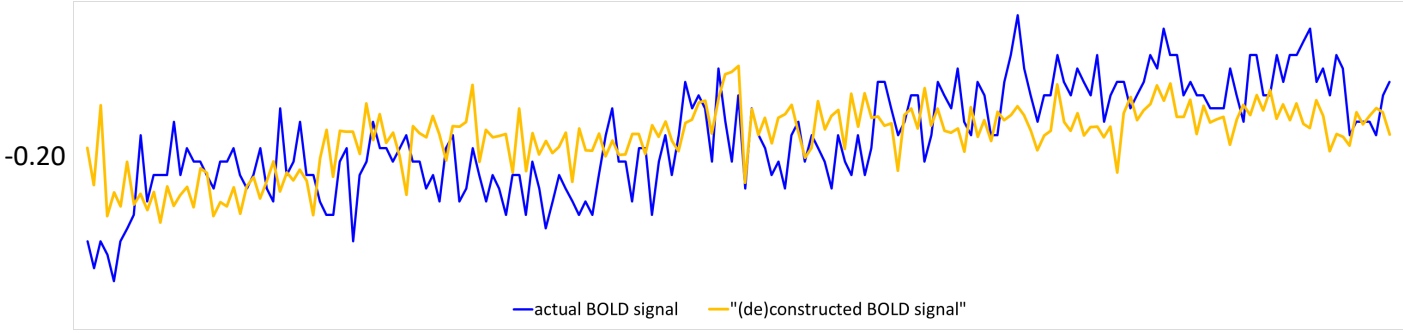
Let's assign some weights and add the **6 movement parameters** (mp) to the deconstructed signal.

+

Participant's movement parameters

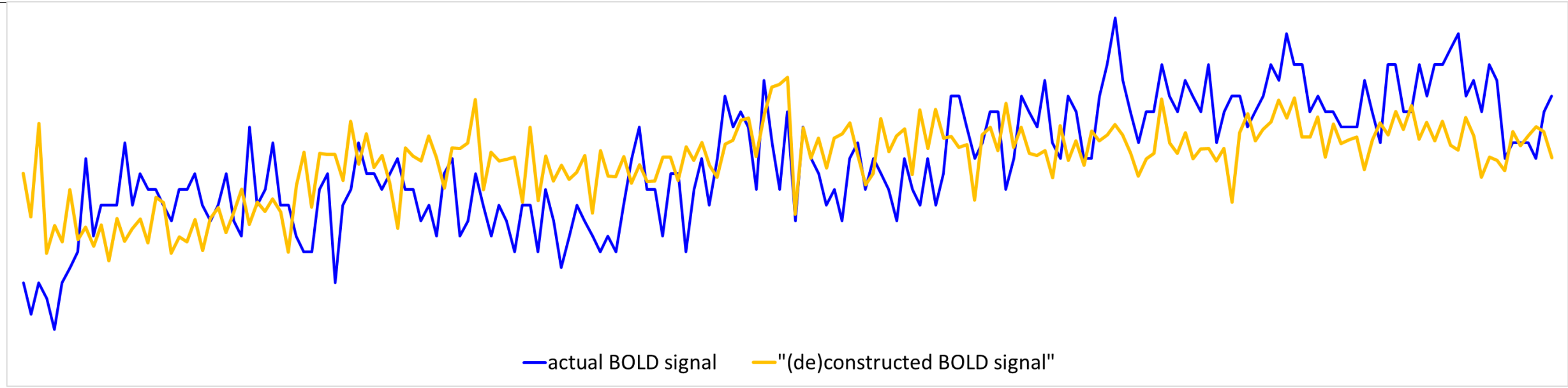


mp1-0.30 +
mp2-0.13 +
mp3 0.33 +
mp4 0.21 +
mp5-0.01 +
mp6 0.64 +



The deconstructed signal matches the true signal much better now!

(de)Constructing the BOLD signal



- These signals that we use to construct the approximation are called **predictors**
- One predictor predicts a **constant response**, just **baseline**
- Another predictor predicts how you should **respond to the task**

- Each predictor is associated with a weight called a **beta-weight**

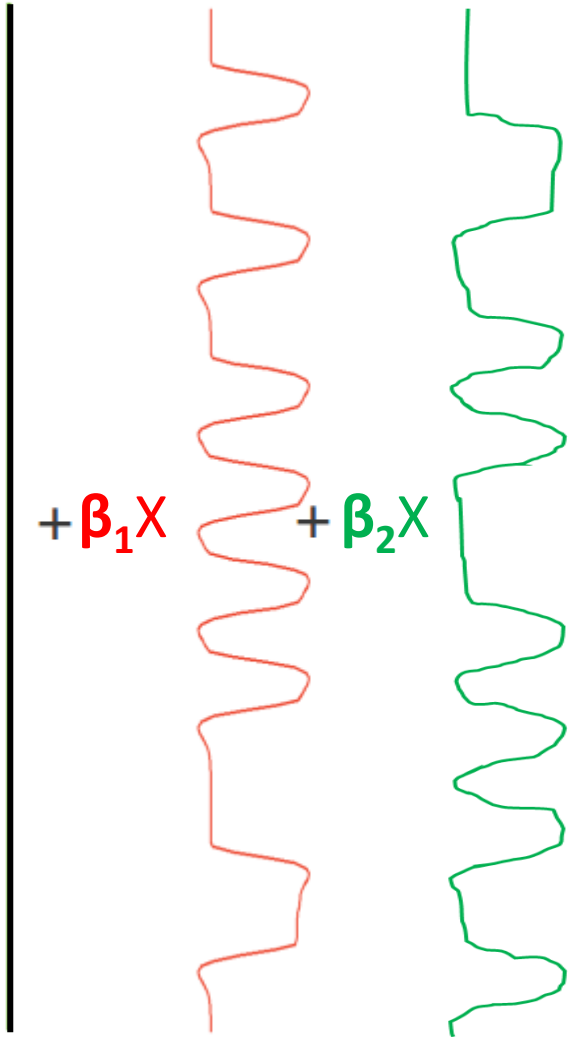
- To create a **linear combination** of predictors, which **approximates a true signal**, we multiply each predictor by its beta-weight and then sum the results.

(de)Constructing the BOLD signal

These signals are just columns (arrays) of numbers stored in a matrix.

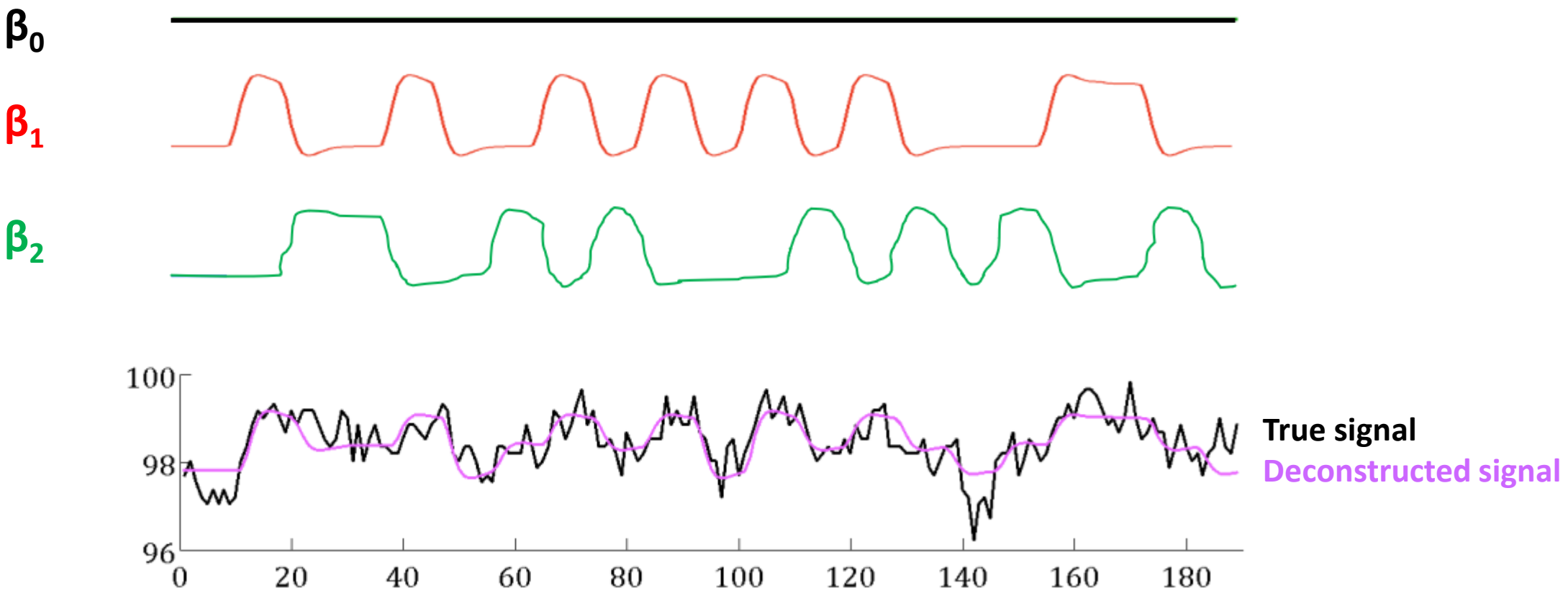
Deconstructed signal

$\beta_0 X$ | $+\beta_1 X$ $+\beta_2 X$ $=$ $\beta_0 X$ $+\beta_1 X$ $+\beta_2 X$ $=$



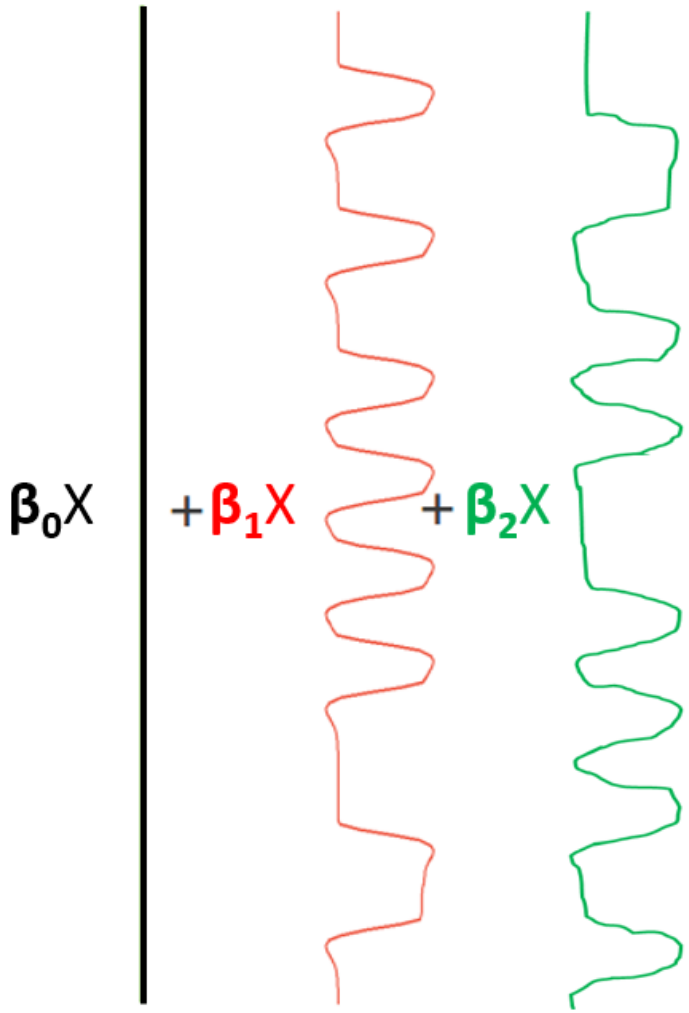
$\beta_0 X$	$\beta_1 X$	$\beta_2 X$
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0.0866	0
1	0.4615	0
1	0.8464	0
1	1.0625	0
1	1.0528	0
1	0.6795	0
1	0.2640	0.0866
1	0.0106	0.4615
1	-0.0971	0.8464
1	-0.1193	1.0625
1	-0.1003	1.1394
1	-0.0688	1.0544
1	0.0460	0.6489
1	0.4403	0.2267
1	0.8365	-0.0203
1	1.0582	-0.1176
1	1.0512	-0.1309
1	0.6790	-0.0196
1	0.2639	0.3900
1	0.0106	0.8047
1	-0.0971	1.0409

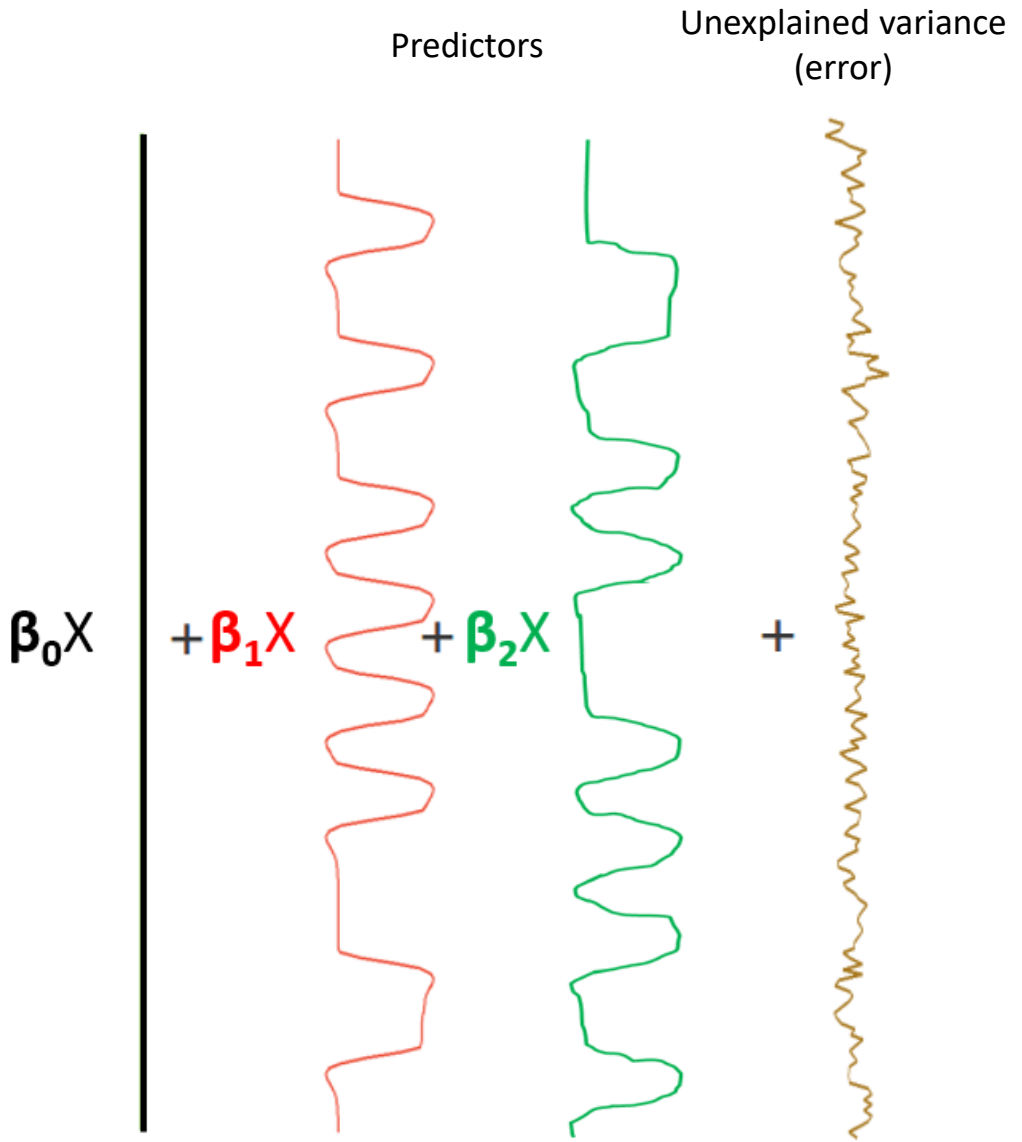
(de)Constructing the BOLD signal

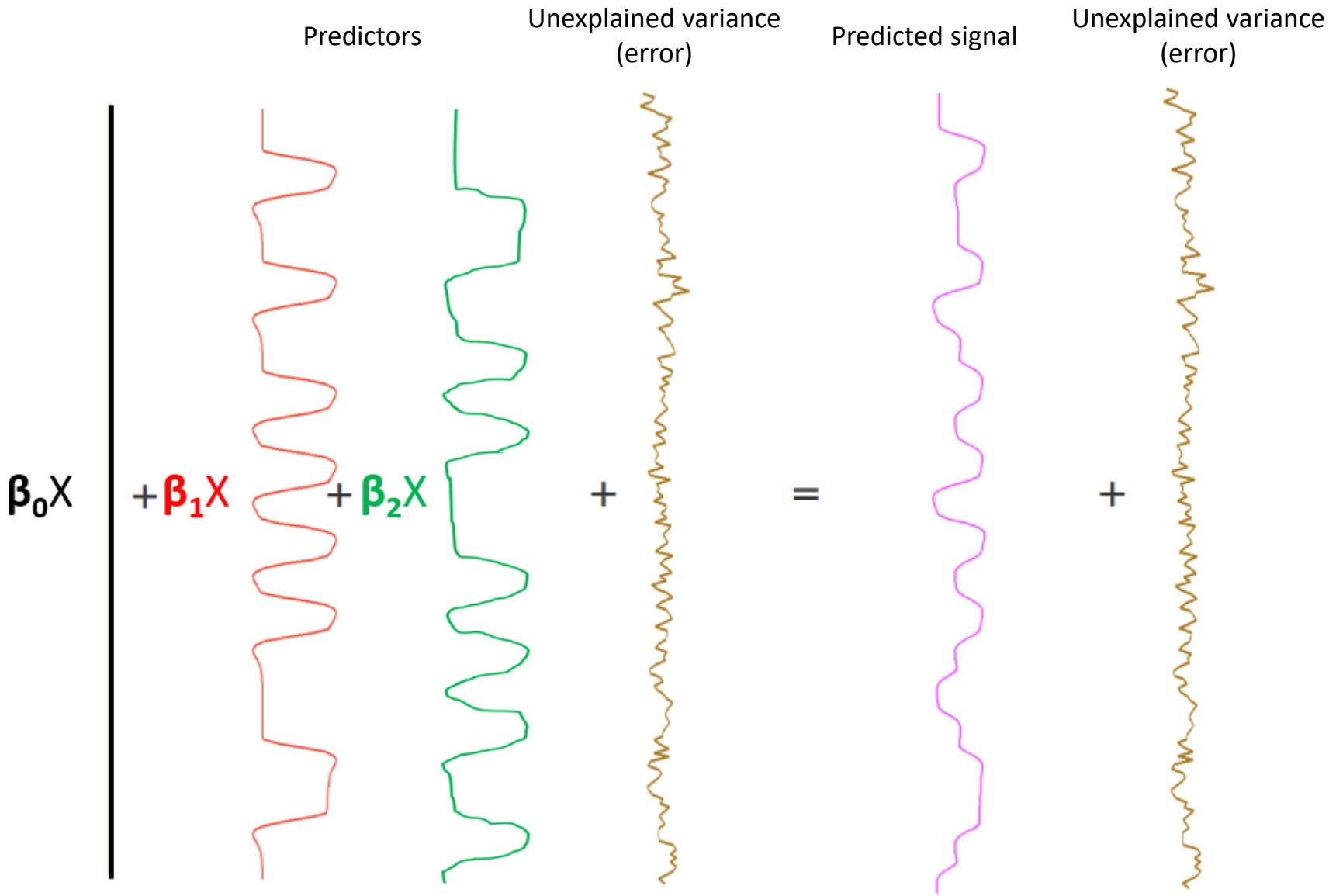


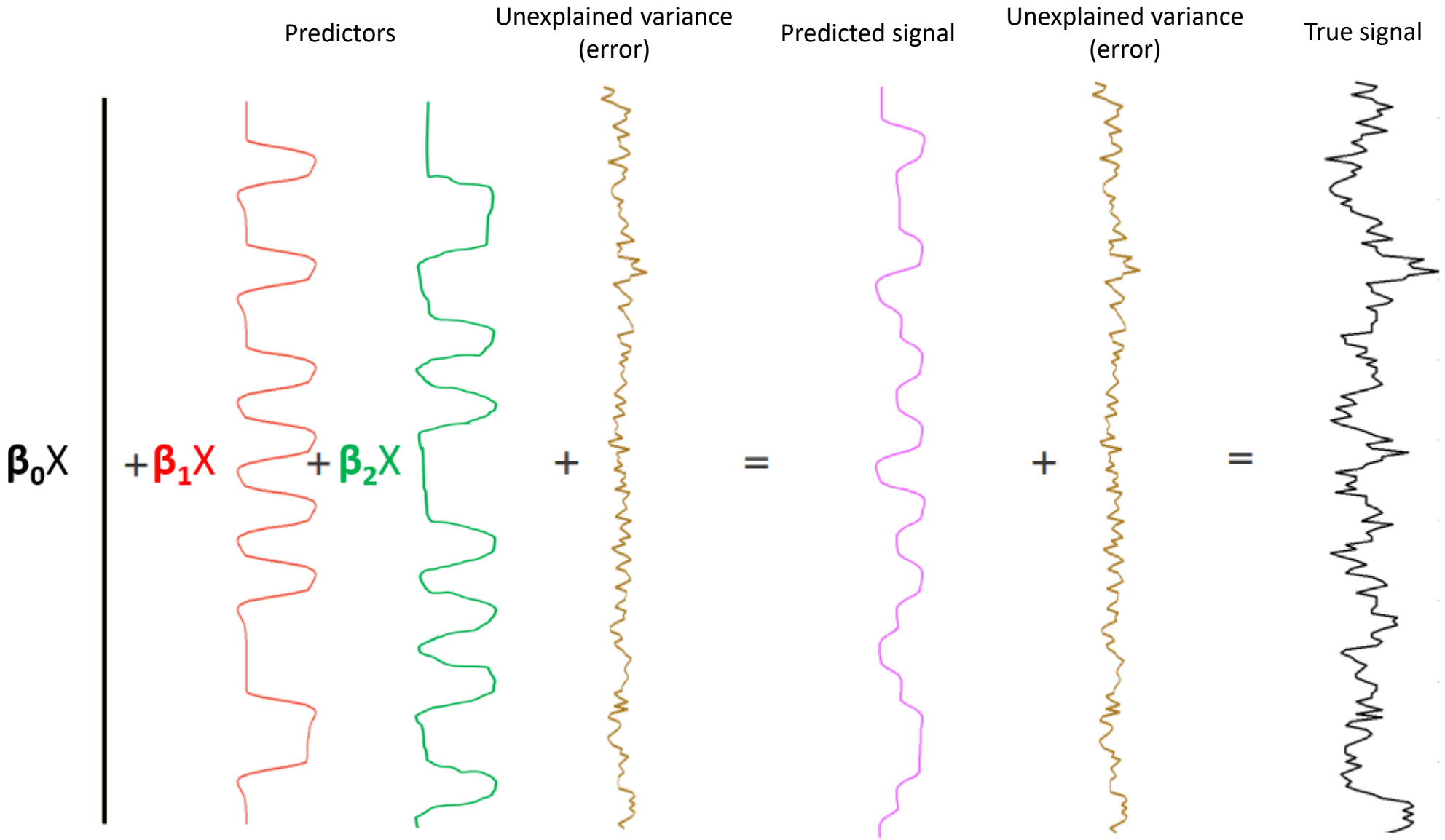
For every time point: **signal(t)** – **prediction(t)** = **error(t)**

Predictors

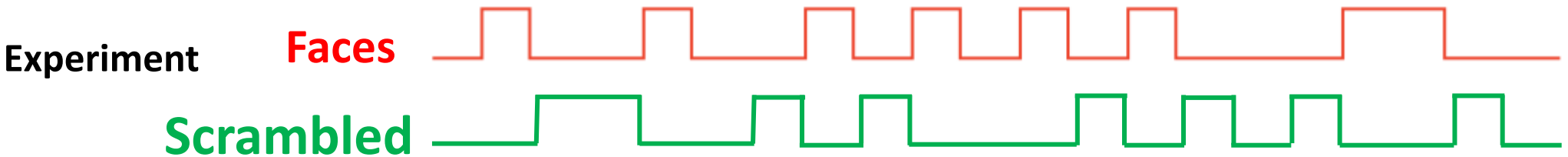



Predictors
Unexplained variance
(error)
$\beta_0 X$
$+\beta_1 X$
$+\beta_2 X$
+


Predictors
Unexplained variance (error)
Predicted signal
Unexplained variance (error)
$\beta_0 X$
$+\beta_1 X$
$+\beta_2 X$
+
=
+


Predictors
Unexplained variance (error)
Predicted signal
Unexplained variance (error)
True signal
$\beta_0 X$
$+\beta_1 X$
$+\beta_2 X$
+
=
+
=

(de)Constructing the BOLD signal



Analysis: the approach that works

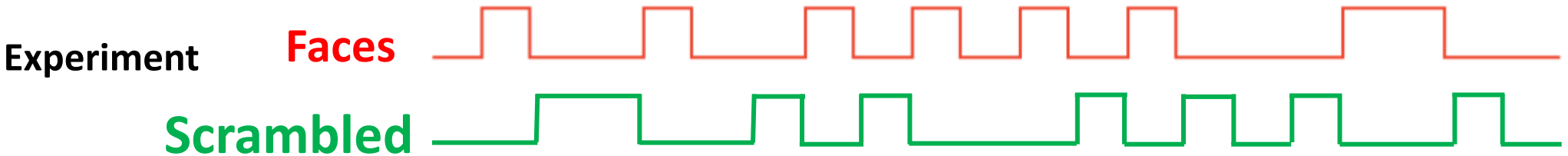
- Find the beta-weights that best approximate the voxel's signal time-series. The best approximation is the one with the least errors
- Compare the beta-weights for estimated response to **Faces** to beta-weights for estimated response to **Scrambled**

BOLD signal = task-related activity changes + noise (other changes)

explained variation — unexplained variation

Linear combination of predictors (deconstructed signal) + errors

(de)Constructing the BOLD signal



Analysis: the approach that works

- Find the beta-weights that best approximate the voxel's signal time-series. The best approximation is the one with the least errors
- Compare the beta-weights for estimated response to **Faces** to beta-weights for estimated response to **Scrambled**

To find beta-weights, we use **General Linear Model (GLM)**

How to find betas: GLM $Y = X\beta + \varepsilon$

BOLD signal = X * b + errors

X * b: explained variation — task-related activity changes

errors: unexplained variation — noise (other changes)

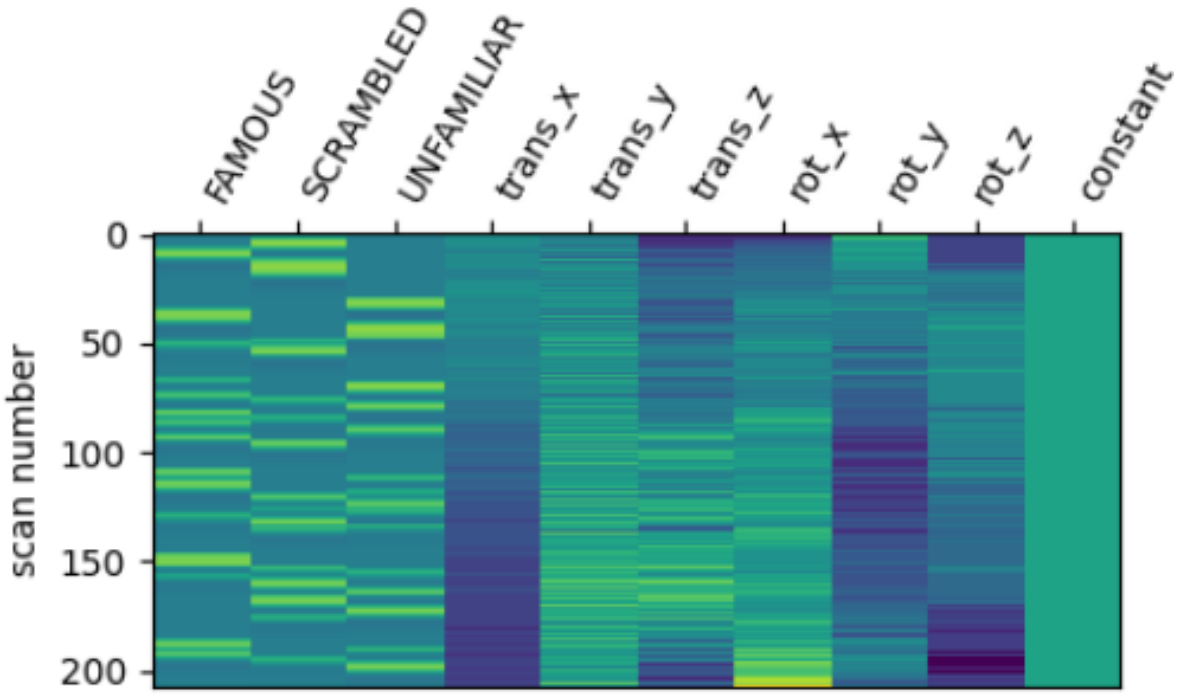
- **What we know?**
  - **BOLD signal**: we collect this from the brain (functional data)
  - X: the design matrix (each column is a predictor that we build ourselves)
- **What we want to find?**
  - b: vector of beta-weights (one weight for predictor in X) that give the best approximation of the BOLD signal
- **How we find it?**
  - By minimising the sum of squared errors. In practice, the **GLM has a formula, which guarantees to find these beta-weights**

How to find betas: GLM $Y = X\beta + \varepsilon$

- Any predictor that can help approximate the BOLD signal will decrease the Sum of Squared Errors
- Therefore, we include additional predictors:
  - The 6 head-motion parameters

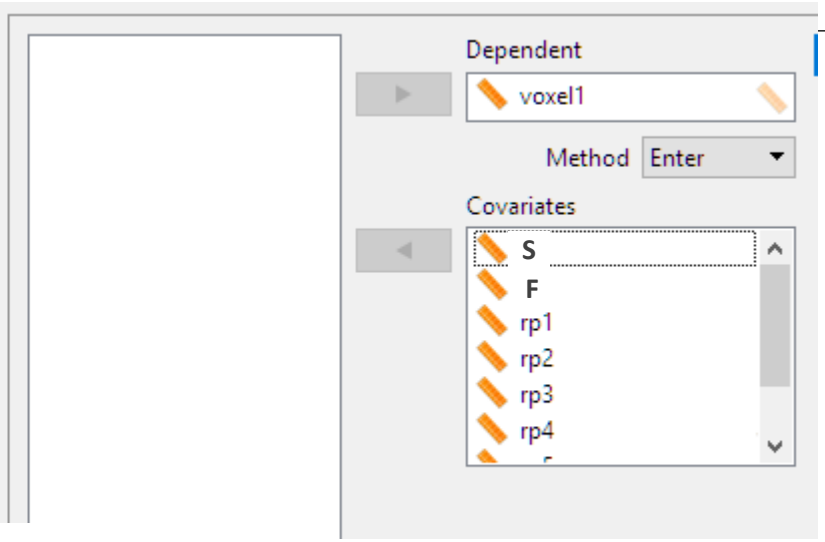
An example design matrix.
Each column is a predictor:

- events
- 6 movement parameters
- constant



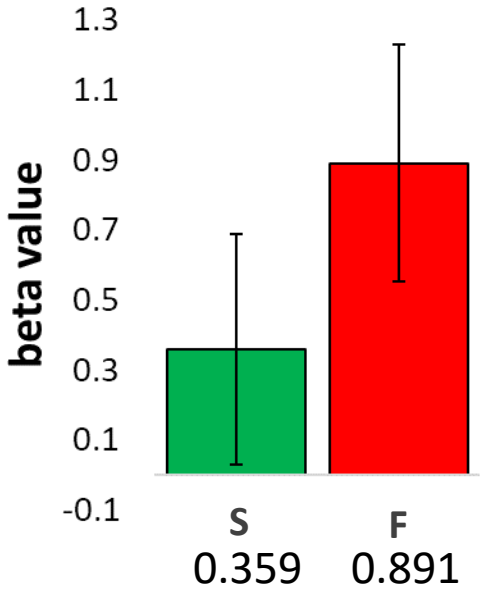
Multiple regression, to find beta-weights

voxel	S	F	rp1	rp2	rp3	rp4	rp5	rp6
-1.98409	0	0	0.744	-0.968	-0.815	-0.263	2.13	-0.0816
-2.49237	0	0	1.15	-0.399	-1.42	-0.365	1.95	0.0903
-1.98409	0	0	-1.33	0.482	-1.81	-0.729	0.666	-0.304
-2.23823	0	0	2.11	0.35	-1.81	-0.748	0.177	-0.157
-2.74651	0	0.0865661	1.49	0.722	-1.99	-0.63	-0.0958	-0.202
-1.98409	0	0.374888	2.35	1.17	-1.98	-0.718	0.0911	-0.509
-1.72996	0	0.384923	0.989	-0.953	-2.12	-1.12	-0.369	-0.378
-1.47582	0.0865661	0.216117	2.11	1.49	-2	-1.41	-0.556	-0.0755



Coefficients

Model		Unstandardized	Standard Error	Standardized	t	p
1	intercept	−0.199	0.106		−1.877	0.062
	S	0.359	0.329	0.070	1.091	0.277
	F	0.891	0.338	0.171	2.637	0.009
	rp1	−0.302	0.077	−0.302	−3.916	< .001
	rp2	−0.129	0.051	−0.129	−2.544	0.012
	rp3	0.326	0.063	0.327	5.223	< .001
	rp4	0.210	0.074	0.210	2.829	0.005
	rp5	−0.008	0.059	−0.008	−0.135	0.893
	rp6	0.643	0.064	0.643	9.978	< .001



A comparison of beta-weights is called a **contrast**.

Formally, a contrast is a vector indicating which beta-weights we are testing. β(F) - β(S): [-1 1 0 0 0 0 0 0] **-0.359 + 0.891 = 0.532**

How to find betas: GLM

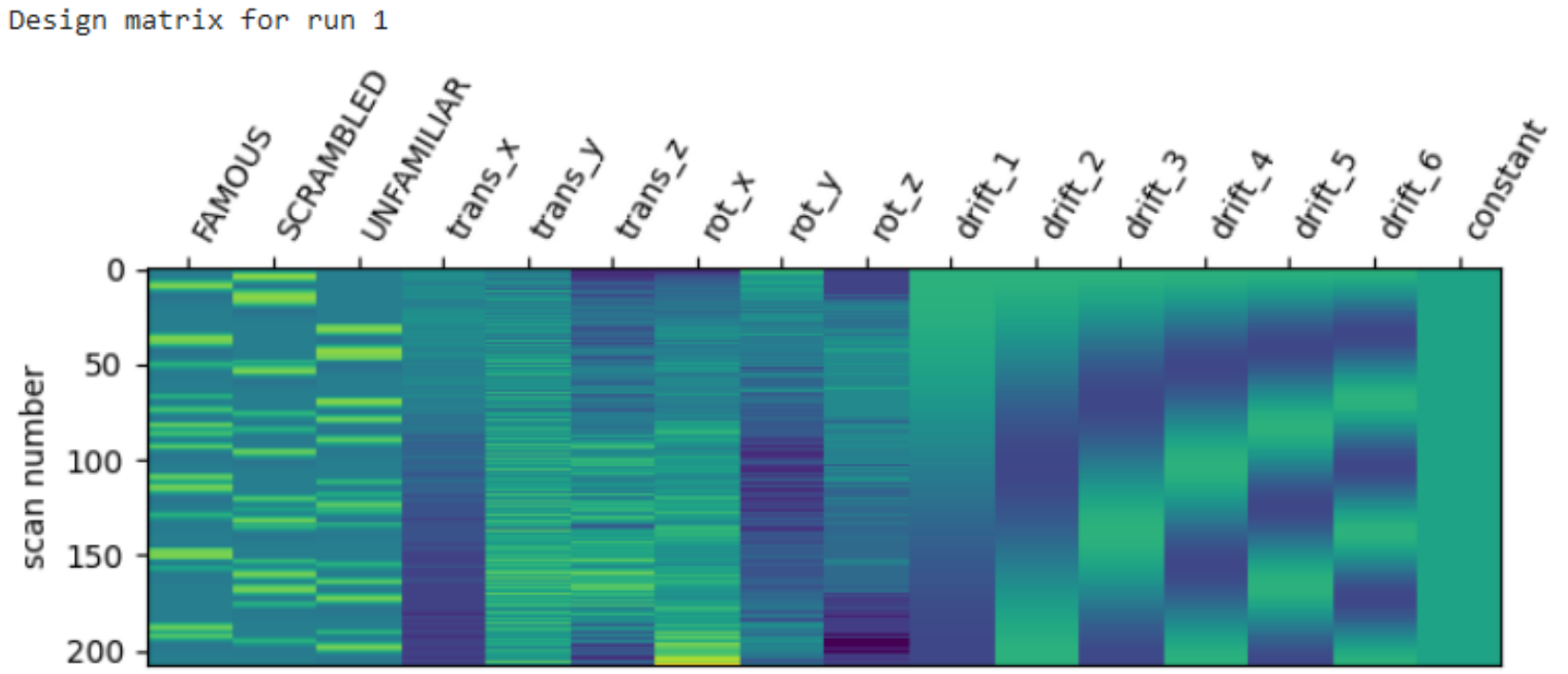
1. **Extract the signal time-series** from a given voxel
2. Create a **design matrix** (predictors)
3. **Run GLM** (the signal and your design matrix are the inputs) to **find beta-weights** that best approximate the true signal
4. Define your **contrast** and test it
5. Repeat for **all voxels**
   - Produces an image file with contrast values for each voxel: **contrast-maps**

First-level analysis: model specification

```
fmri_glm = FirstLevelModel(
    t_r = TR,
    slice_time_ref = 0.5,
    hrf_model = 'spm',
    drift_model = 'Cosine',
    high_pass = 1./128,
    smoothing_fwhm = 6,
    noise_model = 'ar1'
)
```

- **t_r**, Repetition Time (TR)
- **slice_time_ref**: This parameter indicates the time of the reference slice used in the slice timing pre-processing step
- **hrf_model**: defines the HRF model to be used
- **drift_model**: specifies the desired drift model for the design matrices. It can be ‘polynomial’, ‘cosine’ or None. Default = ‘cosine’.
- **high_pass**: specifies the cut frequency of the high-pass filter in Hz for the design matrices. Used only if drift_model is ‘cosine’. Default=0.01 (1/128, as in SPM). Slow signal drifts with a period longer than 128 s will be removed. It is a way to remove possible confounds.
- **smoothing_fwhm**: the full-width at half maximum in millimeters of the spatial smoothing to apply to the signal (smoothing was not done in fMRIPrep!).
- **noise_model**: {‘ar1’, ‘ols’} The temporal variance model. Default=’ar1’. Accounting for serial correlations in fMRI time series due to aliased biorhythms and unmodelled neuronal activity. Accounts for non-independency in the BOLD response.

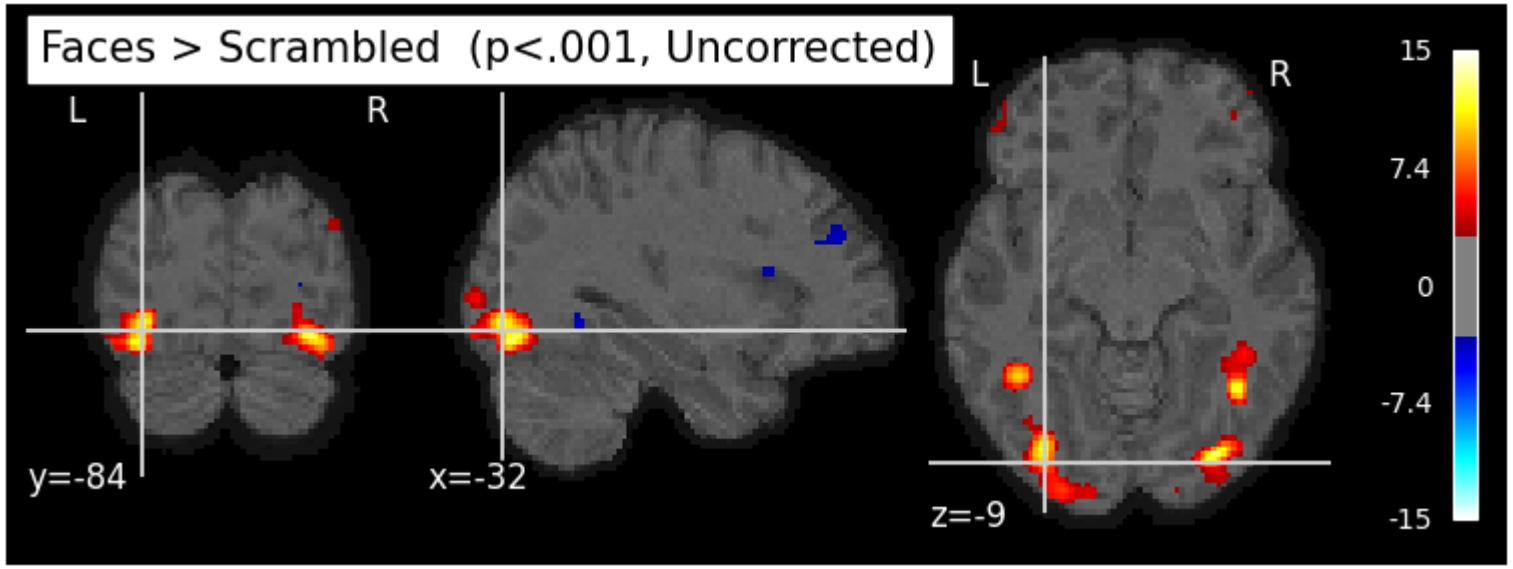
First-level analysis: design matrix and contrasts




```
contrasts = {'Famous_Unfamiliar': pad_vector([1, 0, -1], n_columns),
             'Unfamiliar_Famous': pad_vector([-1, 0, 1], n_columns),
             'Faces_Scrambled': pad_vector([1/2, -1, 1/2], n_columns),
             'Scrambled_Faces': pad_vector([-1/2, 1, -1/2], n_columns),
              'EffectsOfInterest': np.eye(n_columns)[[0,1,2]]}
```

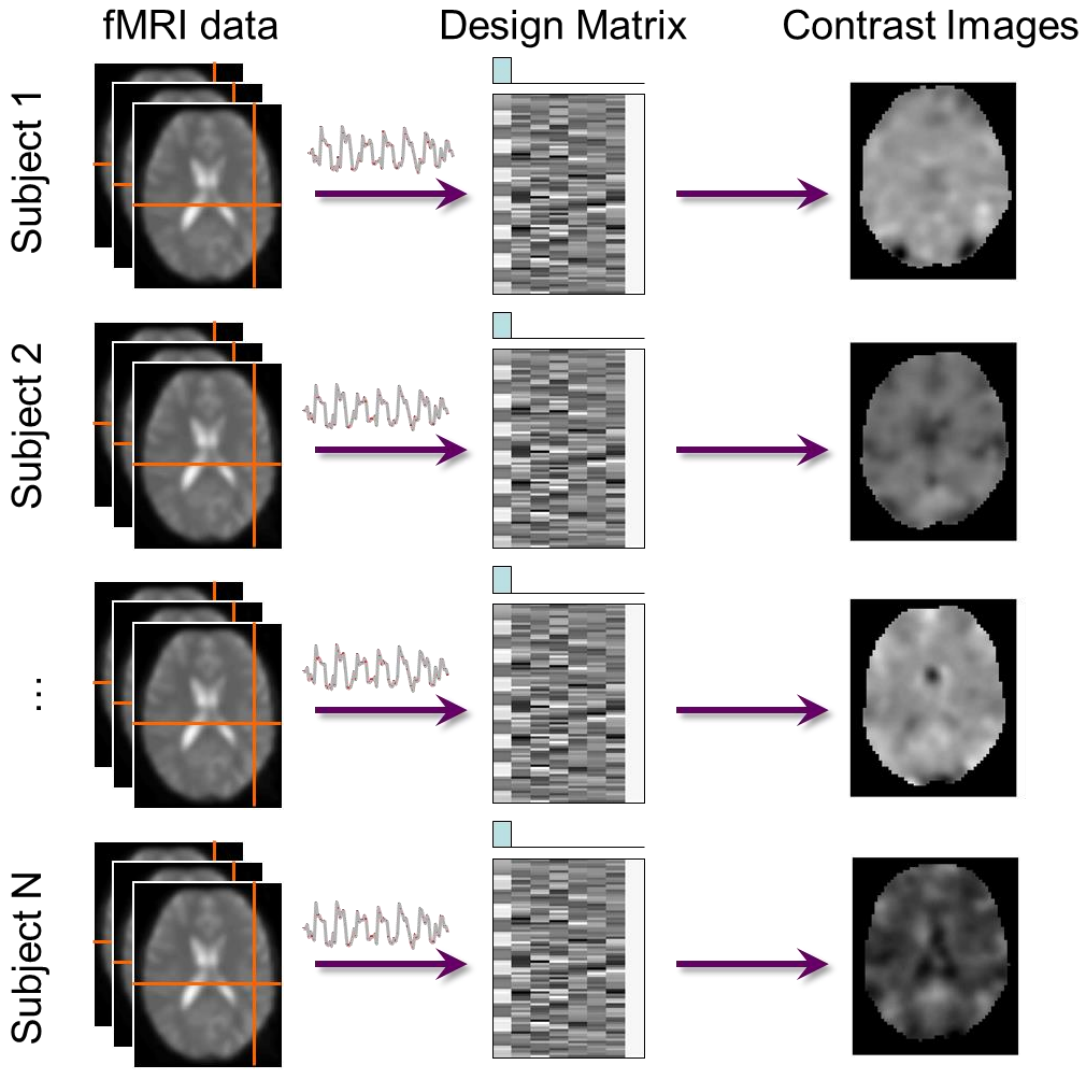
Faces > Scrambled

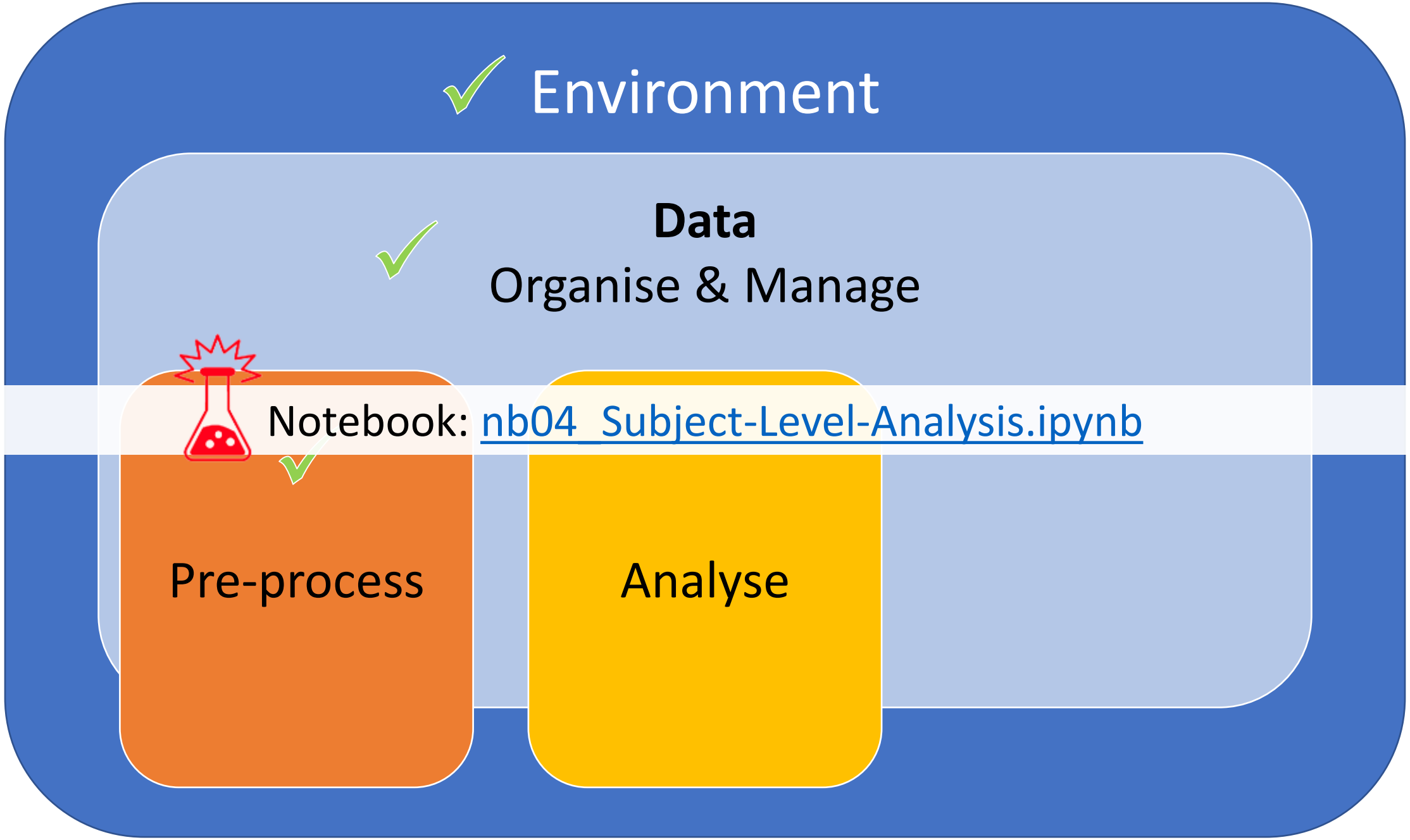
Statistical brain map (z-map) of regions showing significant positive difference between Faces and Scrambled



First-level analysis

- Run the GLM for each subject



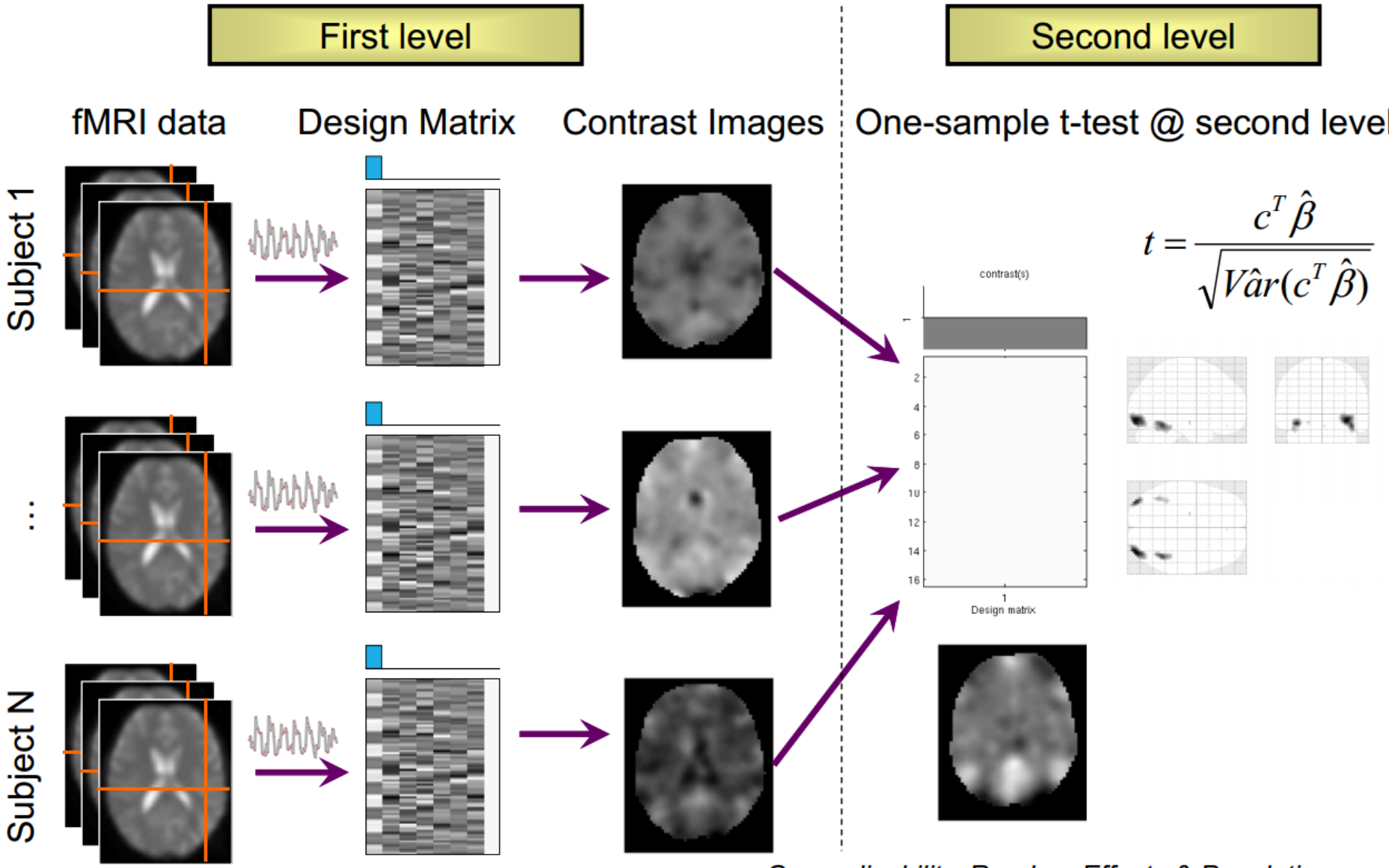

Environment
Data
Organise & Manage
Notebook: nb04_Subject-Level-Analysis.ipynb
Pre-process
Analyse

Group level (2nd level) analysis is across subjects

- Which voxels are showing a significant activation differences between our conditions consistently **within a group**
  - Average contrast effect across sample (e.g., one-sample t-test)

- Importantly, all subject brains need to be in common space, e.g. MNI, to perform voxel-wise group analyses
  - That was achieved by the **Normalisation step in** pre-processing

Summary statistics, Random effects approach

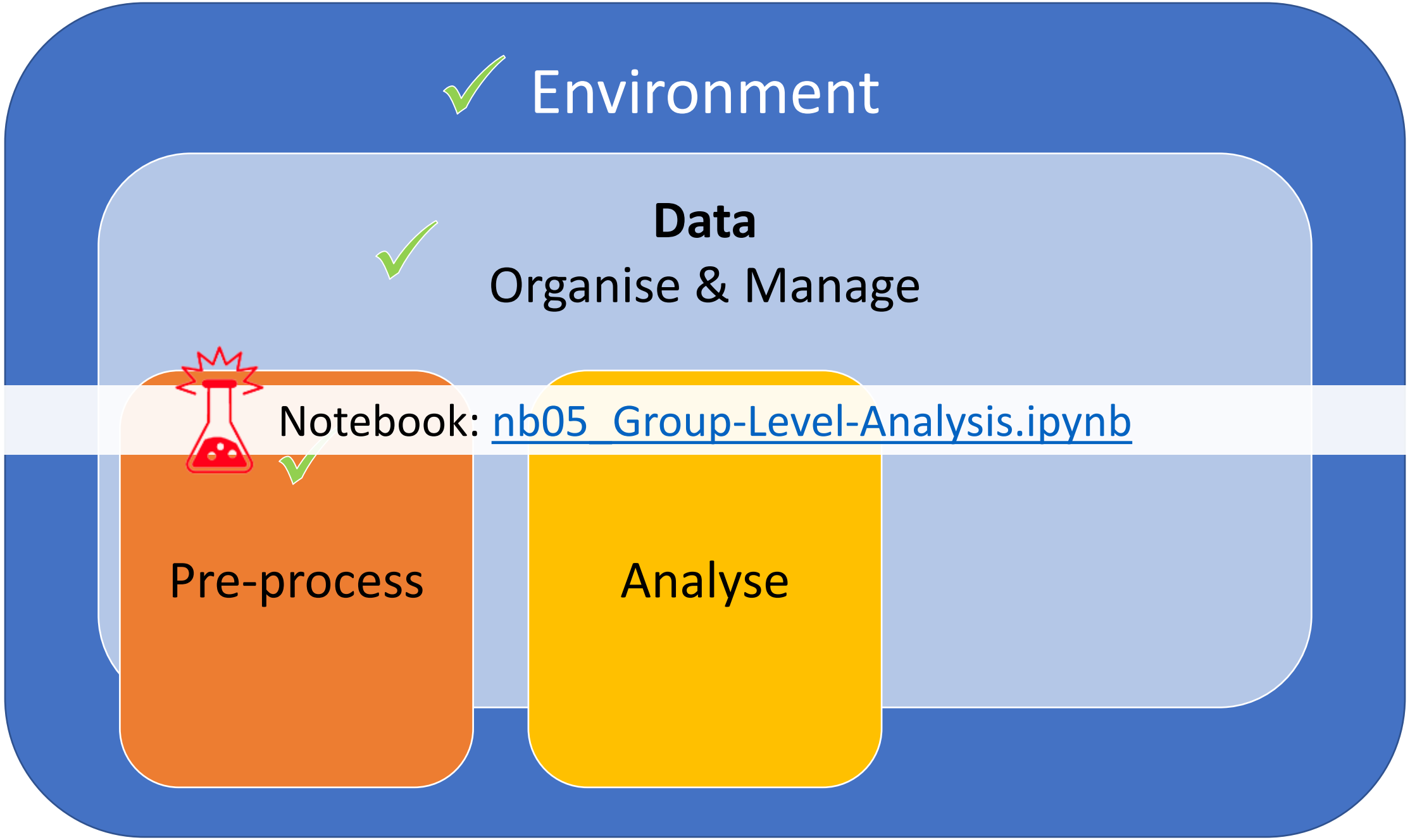


$$t = \frac{c^T \hat{\beta}}{\sqrt{V\hat{a}r(c^T \hat{\beta})}}$$

It is still a GLM model, and we get group level beta estimates!

$Y_{group} = X_{group}\, \beta_{group} + \varepsilon_{group}$

Generalisability, Random Effects & Population Inference. Holmes & Friston, NeuroImage,1998.


Environment
Data
Organise & Manage
Notebook: nb05_Group-Level-Analysis.ipynb
Pre-process
Analyse

Hypothesis

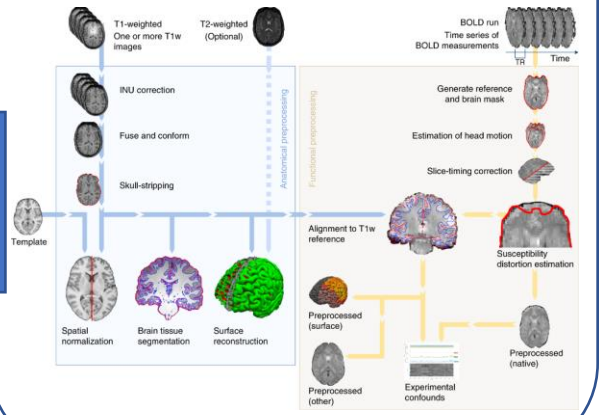
Design an experiment

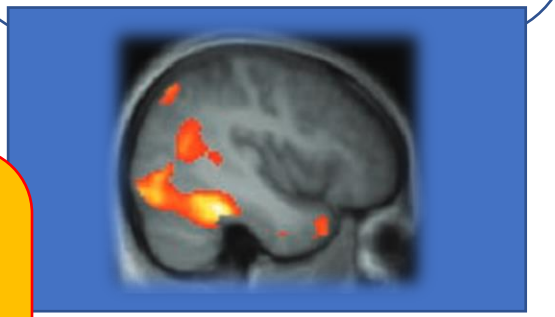
Stimuli
Timing

Collect the data

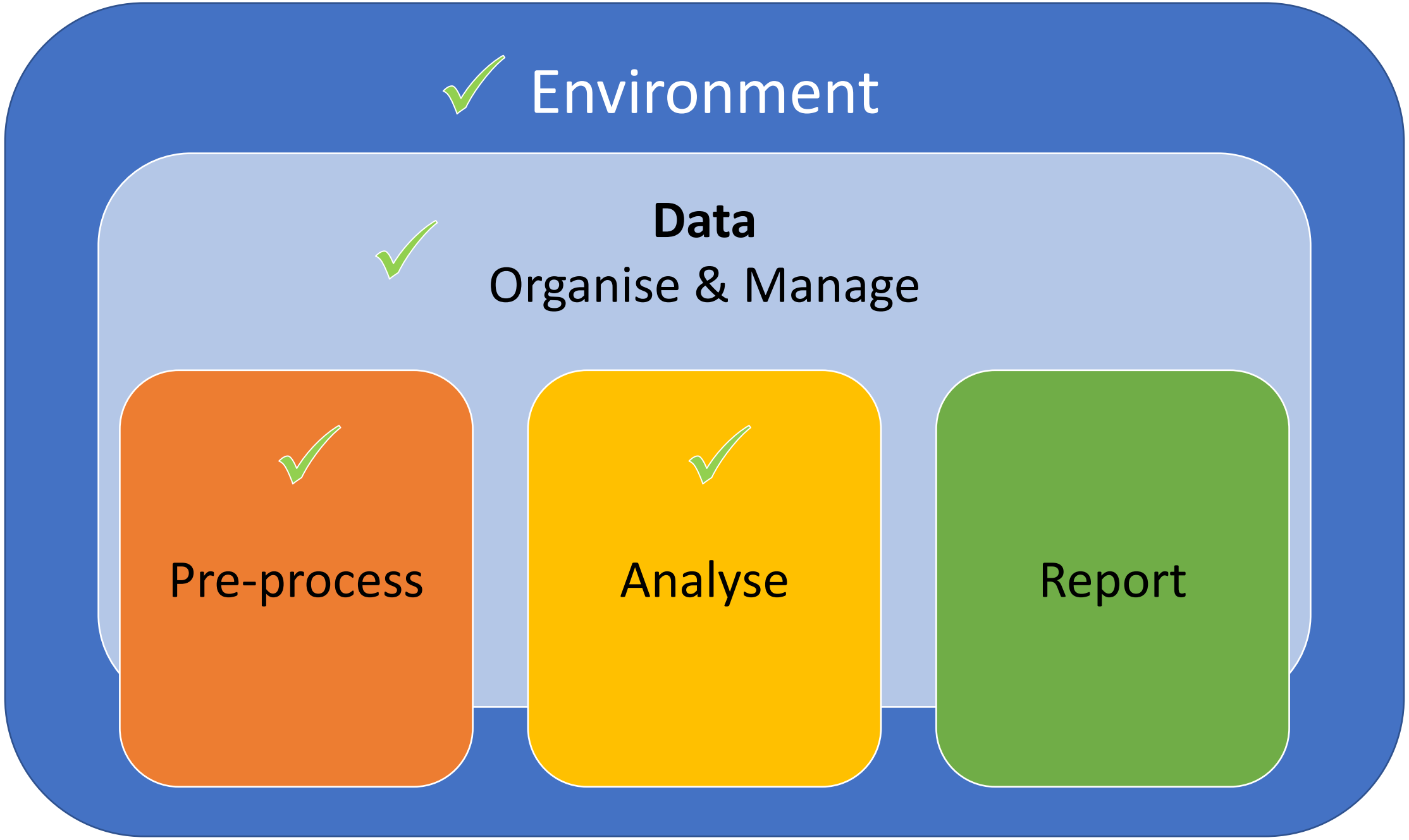
Anatomical image
Functional images
Event details

Pre-process & Analyse





The final push


Environment
Data
Organise & Manage
Pre-process
Analyse
Report

Sharing & Reporting

- Share your **code** and notebooks on GitHub

- Make it **citable** with Zendono
  - https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content

- If you have consent from participants, share the **BIDS data** on OpenNeuro.

- Add your **contrast maps** to NeuroVault

Outline

- Introduction
- Experimental design
- Data management
- Pre-processing
- Statistical analysis
- **Practical demo**