

# Data management

In neuroimaging

## General



NeuroView

### The next generation of neuroscientists needs to learn how to code, and we need new ways to teach them

Ashley L. Juavinett<sup>1,\*</sup>

<sup>1</sup>Division of Biological Sciences, Neurobiology Section, 9500 Gilman Drive, La Jolla, CA 92093, USA

\*Correspondence: [ajuavine@ucsd.edu](mailto:ajuavine@ucsd.edu)

<https://doi.org/10.1016/j.neuron.2021.12.001>

Neuroscience education is at an impasse—we need to teach students coding, but many institutions do not have the resources to do so. Here, I outline three major barriers, as well as solutions, to bringing programming education into our undergraduate and graduate programs.

As many researchers are keenly aware, neuroscience data are growing in both size and complexity. This growth has sent need for more instructors who can teach coding skills to their students. However, there is a mismatch between how much code is taught and how much code is actually used in research. Further, an unmet need in individuals

(Juavinett 2022)

## State in 2022

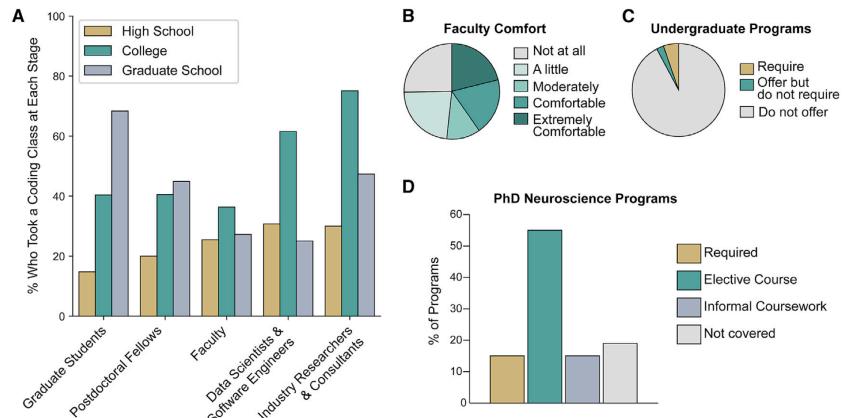


Image credit:  
(Juavinett 2022)

## BIDS

The Brain Imaging Data Structure

About BIDS Blog Standards Getting started Tools Datasets FAQ Collaboration Extensions Impact Merch Contact

About BIDS

About BIDS

Neuroimaging experiments result in complex data that can be arranged in many different ways. For a long time, there was no consensus how to organize and share data obtained in neuroimaging experiments. Even two researchers working in the same lab could opt to arrange their data in a different way. Lack of consensus (or a standard) leads to misunderstandings and time wasted on rearranging data or rewriting scripts expecting certain structure. With the Brain Imaging Data Structure (BIDS), we describe a simple and easy to adopt way of organizing neuroimaging and behavioral data.

dicomdir/

- 1208200617178\_22/
- 1208200617178\_22\_8973.dcm
- 1208200617178\_22\_8943.dcm
- 1208200617178\_22\_2973.dcm
- 1208200617178\_22\_8923.dcm
- 1208200617178\_22\_4473.dcm
- 1208200617178\_22\_8783.dcm
- 1208200617178\_22\_7328.dcm
- 1208200617178\_22\_9264.dcm
- 1208200617178\_22\_9967.dcm
- 1208200617178\_22\_3894.dcm
- 1208200617178\_22\_3899.dcm

1208200617178\_23/

1208200617178\_24/

1208200617178\_25/

my\_dataset/

- participants.tsv
- sub-01/
- anat/
- func/
- dwi/
- sub-02/
- sub-03/
- sub-04/

sub-01\_T1w.nii.gz

sub-01\_task-rest\_bold.nii.gz

sub-01\_task-rest\_bold.json

sub-01\_dwi.nii.gz

sub-01\_dwi.json

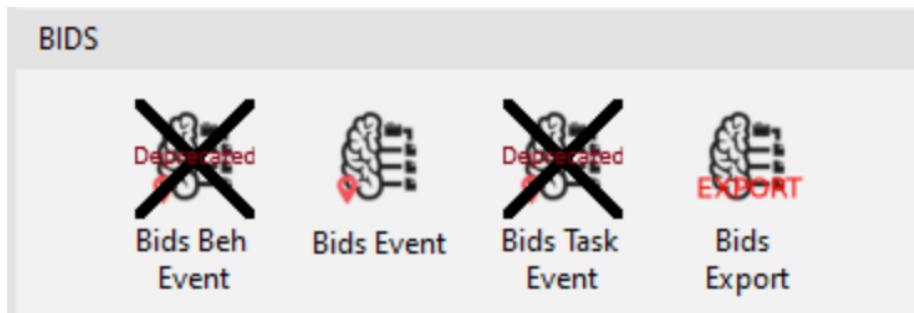
sub-01\_dwi.bval

sub-01\_dwi.bvec

# 1. Psychopy

## BIDS plugin for PsychoPy

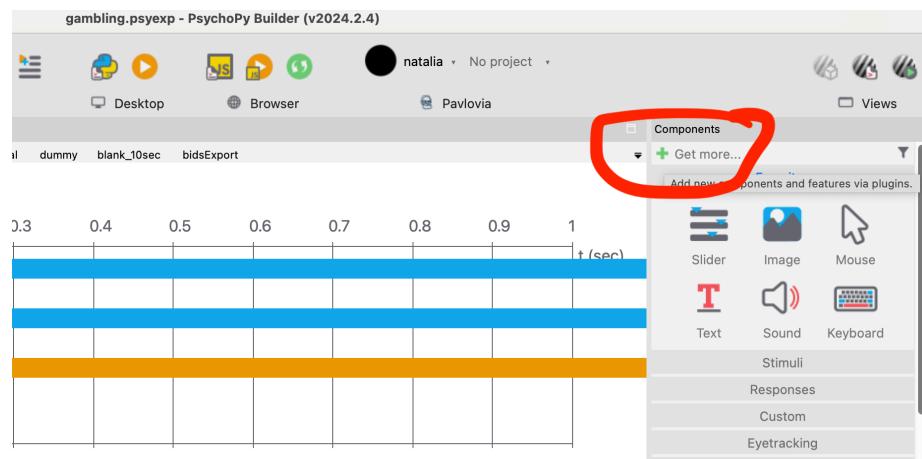
Should be installed and enabled to save events (conditions and button presses) in a comprehensive and standardized manner.



This will also make our data analysis easier

<https://psychopy-bids.readthedocs.io/en/stable/>

## Installation



Open psychopy and press “Get more” on the right side. Alternatively you can open “plugin/packages manager” via “Tools” menue. Psychopy-plugin should be listed among the available plugins.

On the university computers, bids plugin is available through software center.

# Experiment adjustment<sup>1</sup>

Thanks to



**Lukas Wiertz**

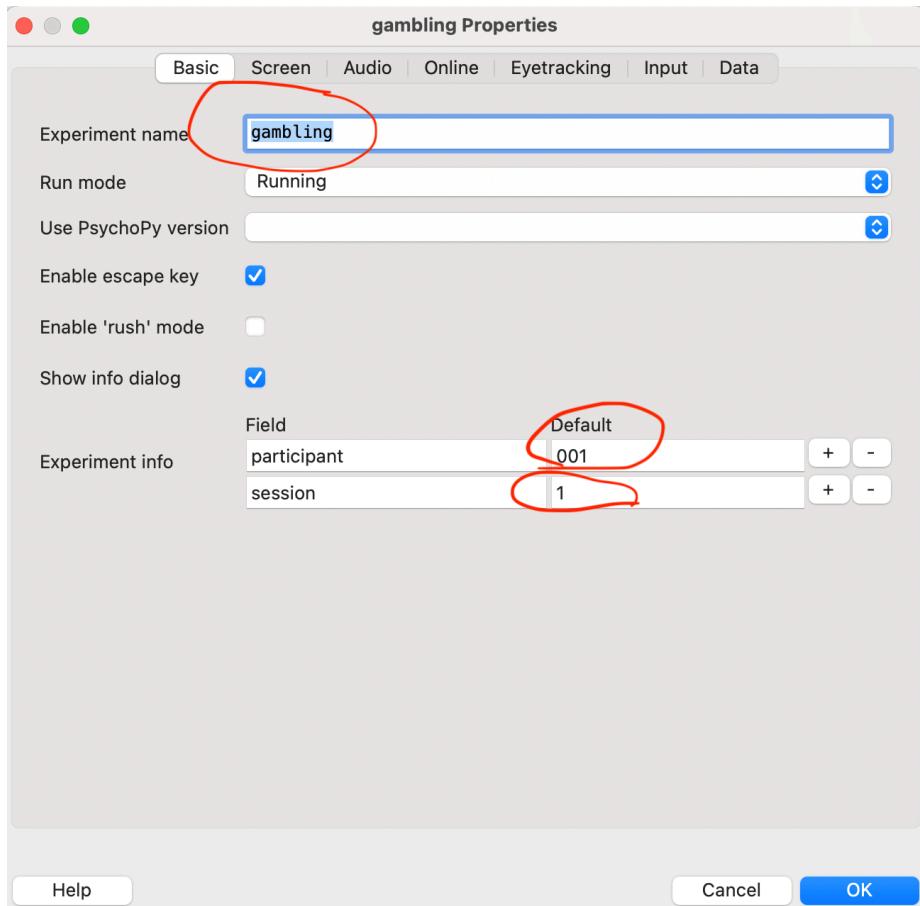
✉ [lukas.wiertz\(at\)uni-graz.at](mailto:lukas.wiertz(at)uni-graz.at)  
📞 +43 316 380 - 3897

Lukas currently took over the BIDS plugin development.

---

<sup>1</sup>valid for PsychoPy v2024.2.4

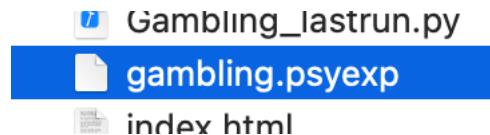
## Properties



1. Experiment name will correspond to the task-[taskname] property in the bids dataset. It is therefore useful to have a short name without capital letters. This did not work on my mac, but maybe this is an OS issue?
2. Participant: switched from random code generator to 001. Since we keep the naming consistent with the MRI dataset, and have multiple runs, it is not useful to generate a new code on every run repetition
3. Session: changed from 001 to 1. This is just our naming convention. Nobody has hundreds of sessions, so we do not expect a number with more than two digits.

## Renaming?

### File



### Project folder



For the task label to work properly in the BIDS plugin, I also tried renaming the experiment file and the folder, but this did not help. Gambling is still spelled out with a capital “G” no matter what I do. Let’s stay tuned for the psychopy-bids updates.

## Research question

### What information do we actually need for analysis?

- **H1:** There are distinct activation patterns for
  - a) winning a bet
  - b) losing a bet
  - c) not taking the bet
- **H2:** The distinct activation pattern for winning and losing will vary depending on the amount of risk/reward

We should keep the research question in mind.

H1 is formulated as a one-way ANOVA with three pairwise post-hoc tests

H2 is formulated as 2x2 two-way ANOVA (at least we will simplify it to that)

## Unique conditions

- small win
- small loss

- large win
- large loss
- pass

## trials.xlsx

	A	B	C	D
1	picture	outcome	win_amount	loss_amount
2	pictures/W2.png	WIN	2	-2
3	pictures/W2.png	WIN	2	-2
4	pictures/W2.png	WIN	2	-2
5	pictures/W2.png	WIN	2	-2
6	pictures/W2.png	WIN	2	-2
7	pictures/W2.png	WIN	2	-2
8	pictures/W2.png	WIN	2	-2
9	pictures/W2.png	WIN	2	-2
10	pictures/W2.png	WIN	2	-2
11	pictures/L2.png	LOSS	2	-2
12	pictures/L2.png	LOSS	2	-2
13	pictures/L2.png	LOSS	2	-2
14	pictures/W6.png	WIN	6	-6
15	pictures/W6.png	WIN	6	-6
16	pictures/W6.png	WIN	6	-6
17	pictures/W6.png	WIN	6	-6
18	pictures/W6.png	WIN	6	-6
19	pictures/W6.png	WIN	6	-6
20	pictures/W6.png	WIN	6	-6
21	pictures/W6.png	WIN	6	-6
22	pictures/L6.png	LOSS	6	-6
23	pictures/L6.png	LOSS	6	-6
24	pictures/L6.png	LOSS	6	-6
25	pictures/L6.png	LOSS	6	-6
26	pictures/W10.png	WIN	10	-10
27	pictures/W10.png	WIN	10	-10
28	pictures/W10.png	WIN	10	-10
29	pictures/W10.png	WIN	10	-10
30	pictures/W10.png	WIN	10	-10
31	pictures/W10.png	WIN	10	-10
32	pictures/L10.png	LOSS	10	-10
33	pictures/L10.png	LOSS	10	-10
34	pictures/L10.png	LOSS	10	-10
35	pictures/L10.png	LOSS	10	-10

For every trial, we should be able to identify whether it is win, loss or pass.

For every win or loss trial, we need to know if it is a win/loss with high stakes or with low stakes.

Note: right now we are not taking the win probability into account, but we know that it is always low for large stakes, and large for low stakes, so it is redundant.

## Side note:

### Ideal paradigm

- Each trial type appears the same number of times
- Carryover counterbalancing - each condition is equally likely to be preceded by each other condition (Brooks 2012)

## Events of interest in a trial

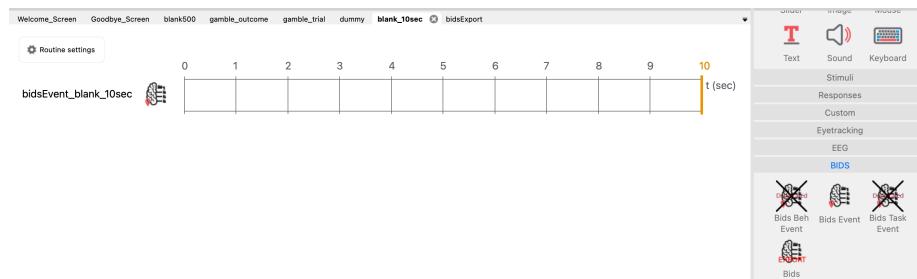
gamble\_trial or gamble\_outcome?



Do we need key presses?

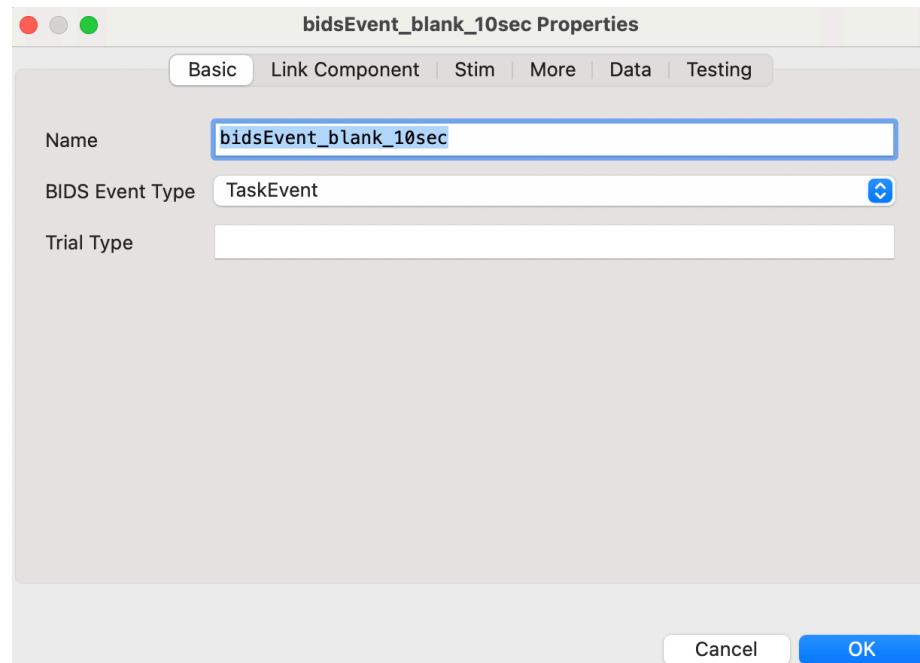
## Adding bids events

### Scanner trigger



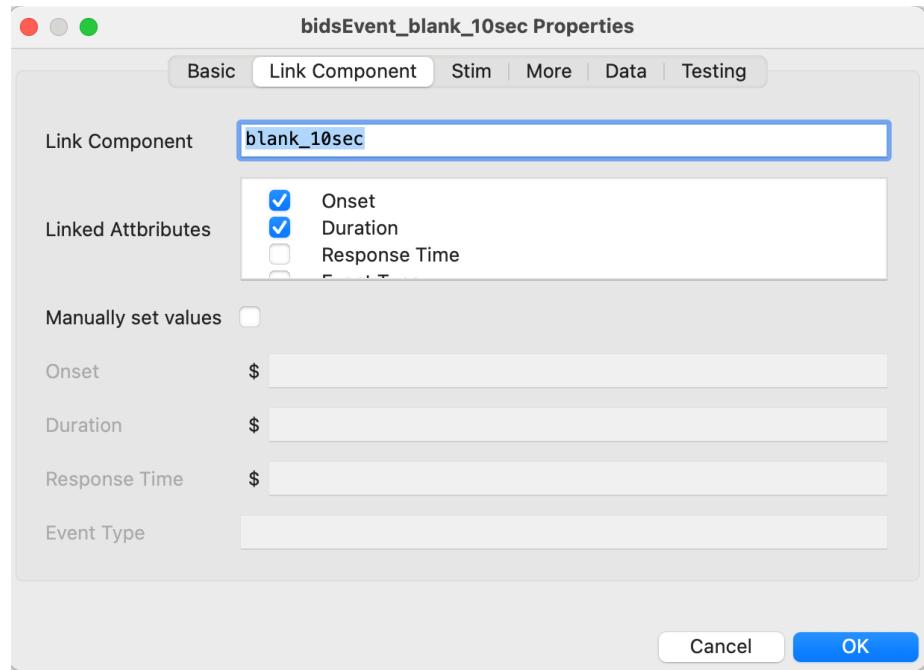
I am not sure if it is needed after all, but I added a bids component to a routine that waits 10 seconds after the scanner trigger and logged its onset.

## Linking to a routine # 1



In the BIDS Event Type field you can choose between TaskEvent and BehEvent. For now the difference does not matter much, let's choose task event.

## Linking to a routine #2

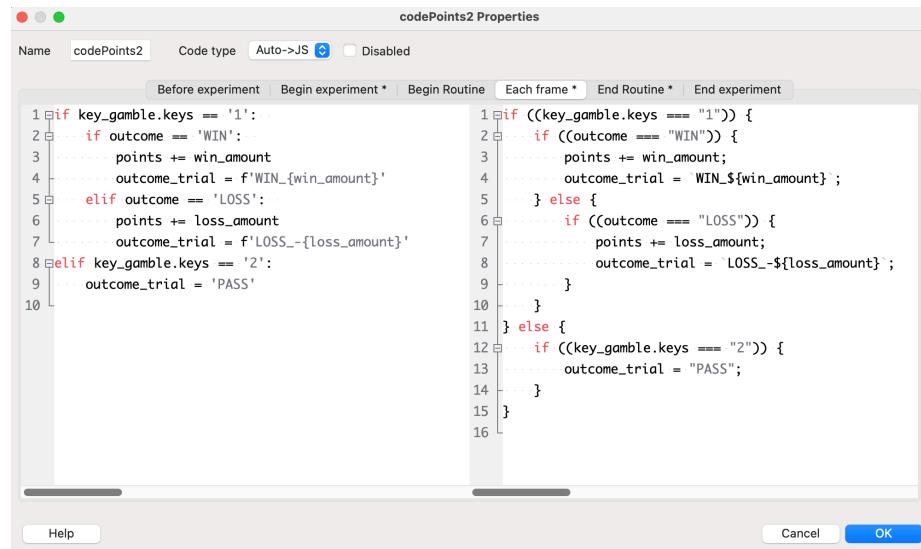


Right now time 0 in our experiment is the start of the welcome screen. We want time 0 to be the scanner trigger. Will take care of it later.

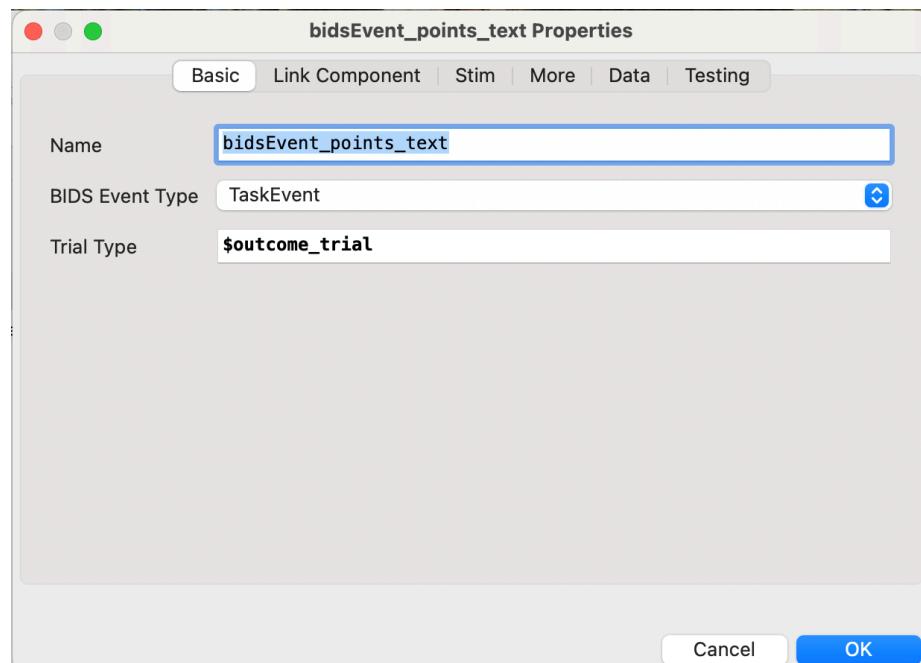
## Adding bids events of interest

### New variable outcome\_trial

We need one variable to take 3 possible outcomes: win, loss or pass. Therefore, I adjusted the code, adding a variable "outcome\_trial" for this purpose.



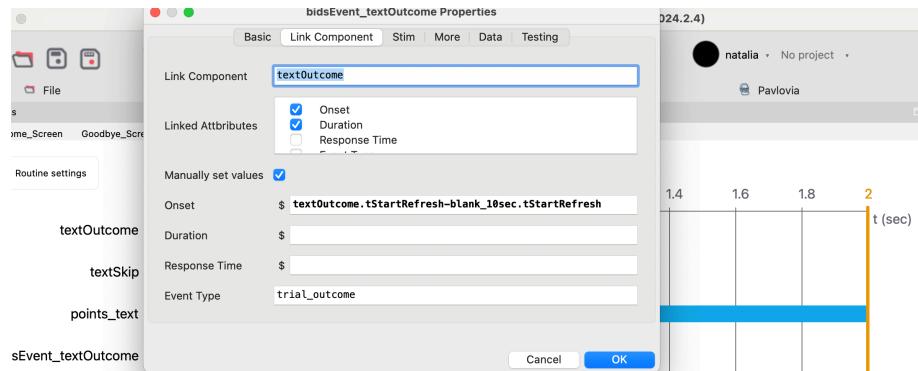
## points\_text



We are creating a bids event and linking it to points\_text within the trial (because this routine appears on every loop iteration).

## Time stamp

### Time 0 - scanning onset



Now we want to adjust the onset time to the scanner trigger arrival. This is done by ticking “Manually set values” and putting the difference between the onset of the current routine and the start of the blank period.

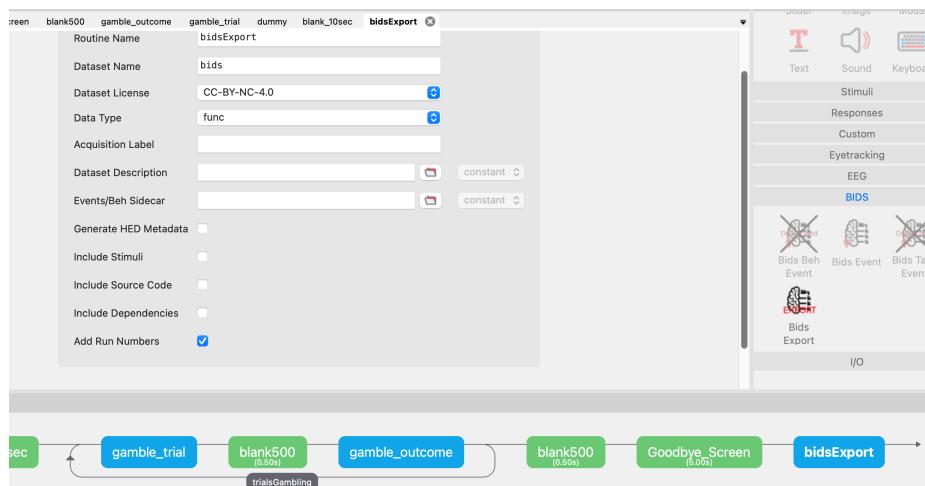
It would have been better and more accurate to subtract the actually key press time, but I ran out of time.

## Other events



Here I added two other bids events that may or may not be useful, just in case:  
 Trial onset and key press, including reaction times.

## BIDS export



It is important to add the bidsExport routine at the end of the experiment.

1. For fMRI, the data type should be “func”
2. Choose your favourite license
3. Tick the box “Add Run Numbers” - this should attach run numbers to the file names in bids format for every new run. I did not get it to work properly on mac (it was always run 1, overwriting the previous file). But supposedly it works on windows and linux.

# Complete BIDS dataset

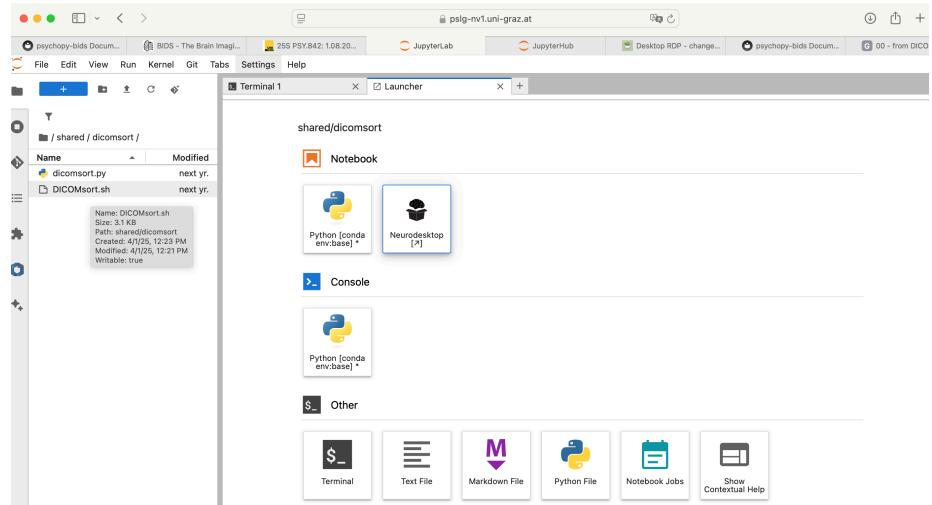
## Example

```
(base) jovyan@neurodesktop-:/storage/bettina/deka/bids/sub-120002/ses-1/func$ ls -l
total 899951
-rw-rw-r-- 1 1024 1030      4387 Feb 12 10:27 sub-120002_ses-1_task-deka_run-1_bold.json
-rw-rw-r-- 1 1024 1030 132122029 Feb 12 10:27 sub-120002_ses-1_task-deka_run-1_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1357 Feb 11 15:04 sub-120002_ses-1_task-deka_run-1_events.tsv
-rw-rw-r-- 1 1024 1030      4387 Feb 12 10:27 sub-120002_ses-1_task-deka_run-2_bold.json
-rw-rw-r-- 1 1024 1030 132025082 Feb 12 10:27 sub-120002_ses-1_task-deka_run-2_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1350 Feb 11 15:04 sub-120002_ses-1_task-deka_run-2_events.tsv
-rw-rw-r-- 1 1024 1030      4349 Feb 12 10:27 sub-120002_ses-1_task-deka_run-2_sbref.json
-rw-rw-r-- 1 1024 1030 573520 Feb 12 10:27 sub-120002_ses-1_task-deka_run-2_sbref.nii.gz
-rw-rw-r-- 1 1024 1030      4389 Feb 12 10:27 sub-120002_ses-1_task-deka_run-3_bold.json
-rw-rw-r-- 1 1024 1030 132241759 Feb 12 10:27 sub-120002_ses-1_task-deka_run-3_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1388 Feb 11 15:04 sub-120002_ses-1_task-deka_run-3_events.tsv
-rw-rw-r-- 1 1024 1030      4349 Feb 12 10:27 sub-120002_ses-1_task-deka_run-3_sbref.json
-rw-rw-r-- 1 1024 1030 573563 Feb 12 10:27 sub-120002_ses-1_task-deka_run-3_sbref.nii.gz
-rw-rw-r-- 1 1024 1030      4389 Feb 12 10:27 sub-120002_ses-1_task-deka_run-4_bold.json
-rw-rw-r-- 1 1024 1030 132224847 Feb 12 10:27 sub-120002_ses-1_task-deka_run-4_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1342 Feb 11 15:04 sub-120002_ses-1_task-deka_run-4_events.tsv
-rw-rw-r-- 1 1024 1030      4351 Feb 12 10:27 sub-120002_ses-1_task-deka_run-4_sbref.json
-rw-rw-r-- 1 1024 1030 574122 Feb 12 10:27 sub-120002_ses-1_task-deka_run-4_sbref.nii.gz
-rw-rw-r-- 1 1024 1030      4382 Feb 12 10:27 sub-120002_ses-1_task-loc_run-1_bold.json
-rw-rw-r-- 1 1024 1030 130534416 Feb 12 10:28 sub-120002_ses-1_task-loc_run-1_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1316 Feb 11 15:03 sub-120002_ses-1_task-loc_run-1_events.tsv
-rw-rw-r-- 1 1024 1030      4382 Feb 12 10:28 sub-120002_ses-1_task-loc_run-2_bold.json
-rw-rw-r-- 1 1024 1030 130077688 Feb 12 10:28 sub-120002_ses-1_task-loc_run-2_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1295 Feb 11 15:03 sub-120002_ses-1_task-loc_run-2_events.tsv
-rw-rw-r-- 1 1024 1030      4384 Feb 12 10:28 sub-120002_ses-1_task-loc_run-3_bold.json
-rw-rw-r-- 1 1024 1030 129979791 Feb 12 10:28 sub-120002_ses-1_task-loc_run-3_bold.nii.gz
-rw-rw-r-- 1 1024 1030      1304 Feb 11 15:03 sub-120002_ses-1_task-loc_run-3_events.tsv
```

This is how a complete dataset for fmri analysis should look like. You should have functional `_bold.nii.gz` files and `_events.tsv` files in one `func/` folder.

## 2. DICOM to BIDS

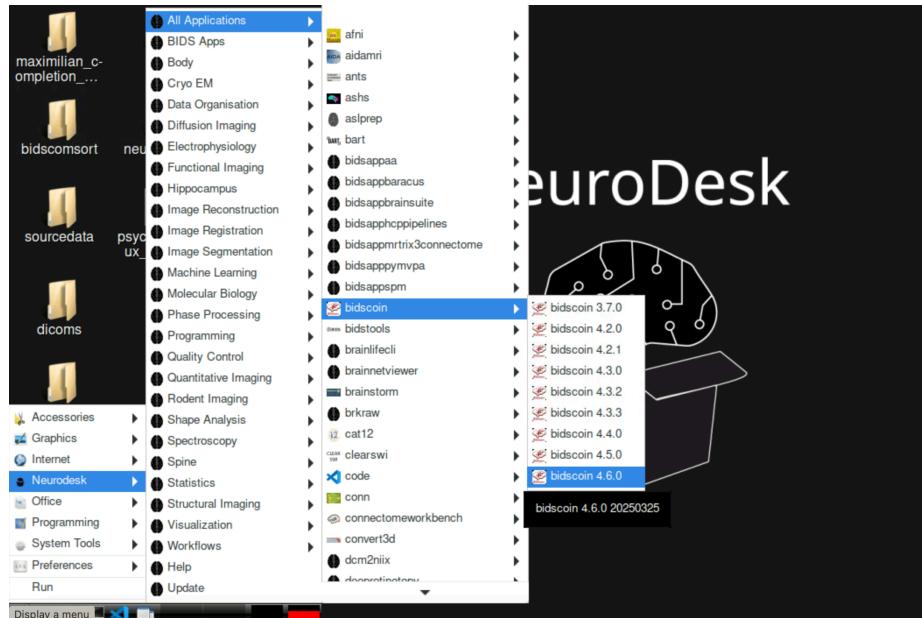
### Neurodesk



### MRI data formats

- DICOM - medical image format
  - .dcm .ima or nothing at all
- NIFTI - neuroimaging format we work with
  - .nii or .nii.gz

## bidscoin



## Shell commands

```
bash
File Edit Tabs Help

Getting started:
bidscoin -h

Convert DICOM to BIDS:
bidsmapper inputfolder bidsoutputfolder
bidscoiner inputfolder bidsoutputfolder
...

More documentation can be found here:
https://bidscoin.readthedocs.io/

Citation:
...
Zwiers MP, Moia S, Oostenveld R. BIDScoin: A User-Friendly Application to Convert Source Data to Brain Imaging Data Structure. Front Neuroinform. 2022 Jan 13;15:770608. doi: 10.3389/fninf.2021.770608. PMID: 35095452; PMCID: PMC8792932.
...

To run container outside of this environment: ml bidscoin/4.6.0

-----
bidscoin-4.6.0:~$ bidscoin-4.6.0:~$
```

```

#!/bin/bash

# you can copypaste the following commands into the terminal window that is opened by BIDSCC

# define directories
sourceFolder=/home/jovyan/completion2/dcm
sortedFolder=/home/jovyan/completion2/dcm_sorted

# check the output of the source folder
ls -l $sourceFolder

# make the target directory with subdirectories
mkdir -p $sortedFolder

# change to the script folder
cd /home/jovyan/dicomsort/

# run the script
./DICOMsort.sh $sourceFolder $sortedFolder

# check the output of the target folder
ls -l $sortedFolder

# run bidsmapper
bidsMapper $sortedFolder /home/jovyan/completion2/bids
# note: in the intendedFor of the fieldmap, add func/sub-001_ses-1_xxx_bold.*

# once you are happy with the mapping, run bidscoiner
bidscoiner $sortedFolder /home/jovyan/completion2/bids

```

## bismapper



Currently, to discover the data we need to delete the content of the “subprefix” field and then save the bids map in the default location. After this, run the same bidsmapper command again.

## Subject and session

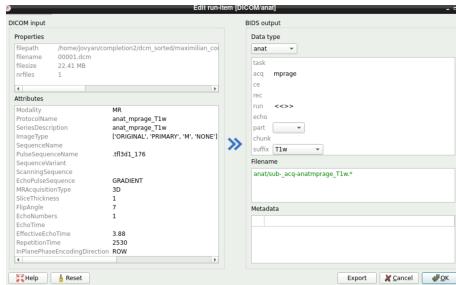
DICOM mappings			
Participant data		BIDS output	Action
participant_id	1		Edit
session_id	1		Edit
age	<<PatientAge>>		Edit
sex	<<PatientSex>>		Edit
height	<<PatientSize>>		Edit
weight	<<PatientWeight>>		Edit
Representative sample			
-> DICOM input			
001	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser01_bold*	Edit
002	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser01_bold*	Edit
003	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser01_bold*	Edit
004	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser02_bold*	Edit
005	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser02_bold*	Edit
006	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser03_bold*	Edit
007	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser03_bold*	Edit
008	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser04_bold*	Edit
009	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser04_bold*	Edit
010	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser05_bold*	Edit
011	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser05_bold*	Edit
012	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser06_bold*	Edit
013	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser06_bold*	Edit
014	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser07_bold*	Edit
015	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser07_bold*	Edit
016	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser08_bold*	Edit
017	00001.dcm	funcsub-999_ses-1_task-functaskcompletionacnouser08_bold*	Edit

Now the bidsmapper discovered everything, but the files are not named correctly.

Edit subject and session first.

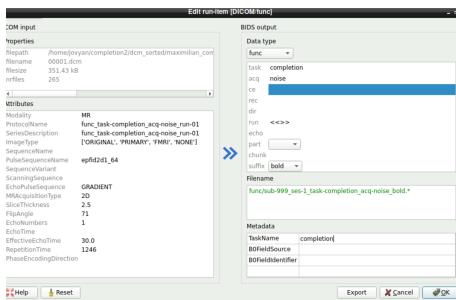
Now we need to rename the files by clicking “Edit” next to every file.

## Editing file names: anatomical



anatmprage -> mprage in the “acq” field

## Editing file names: functional



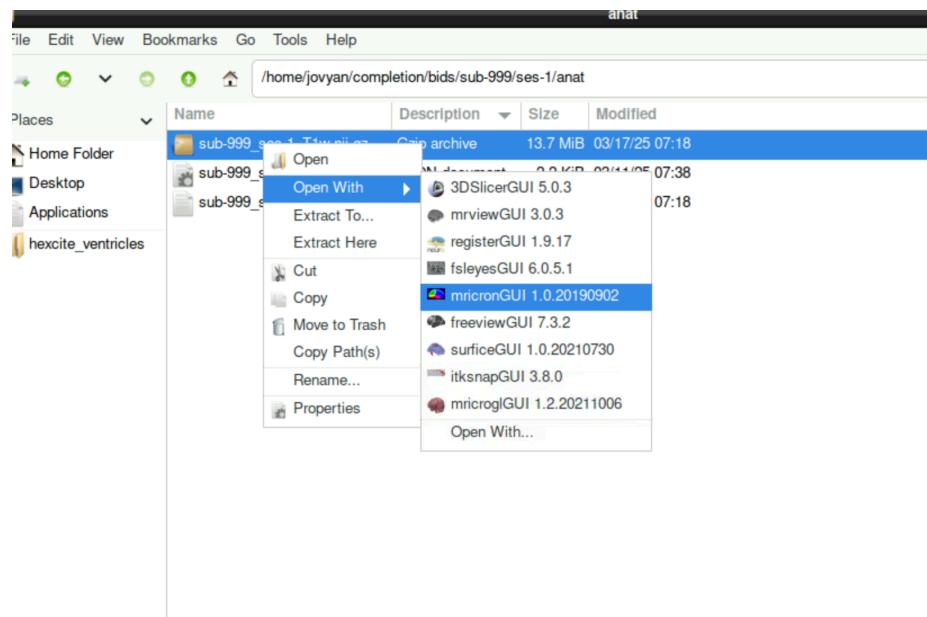
Edit the task name and acq field

Do this for every functional file

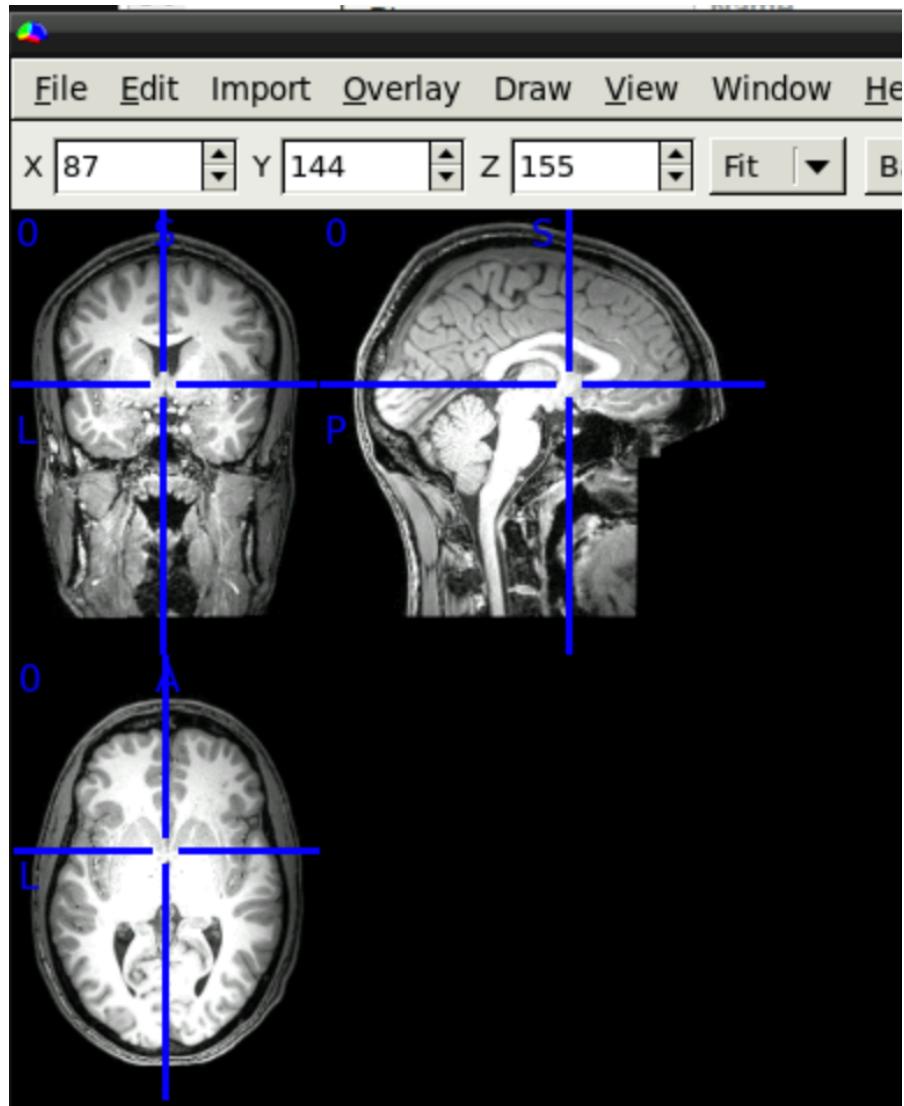
In the intendedFor of the fieldmap, add “func/sub-999\_ses-1\_task-completion\_acq-noise\_bold.\*” with a wildcard. This will select all respective functional runs

Once you are happy, run bidscoiner to do the actual conversion

## View files



## MRICron

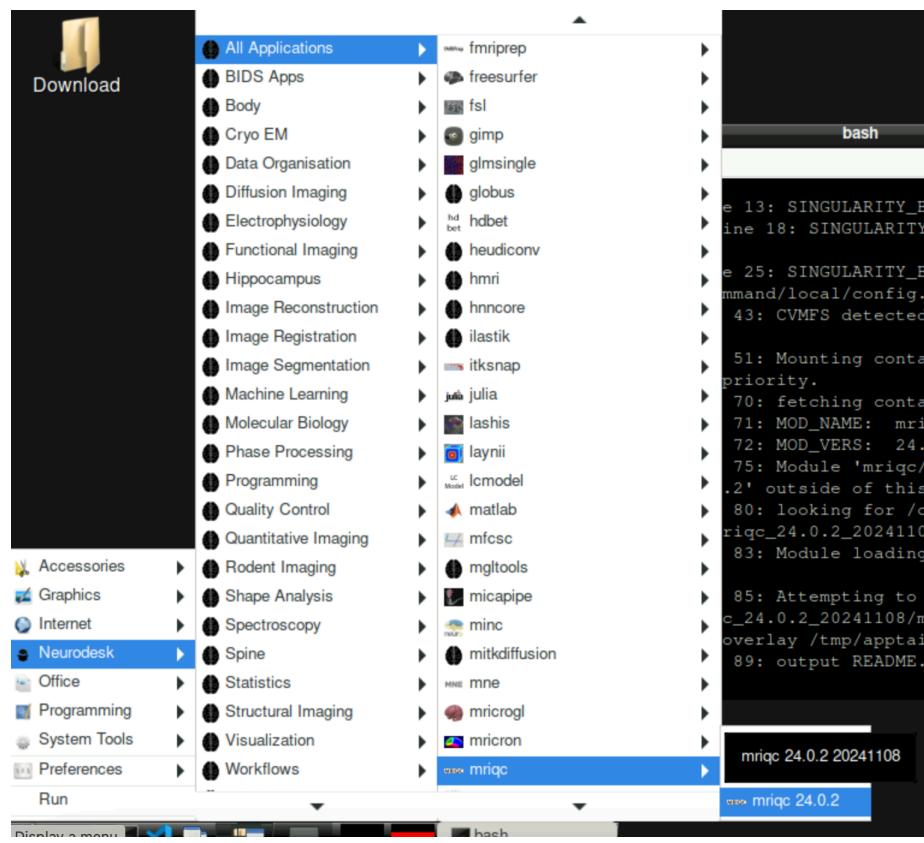


Now you can view the nifti files with any suitable software

### 3. Quality assurance

#### MRIQC

MRIQC is a tool used for quality assurance of bids-valid datasets



# Run

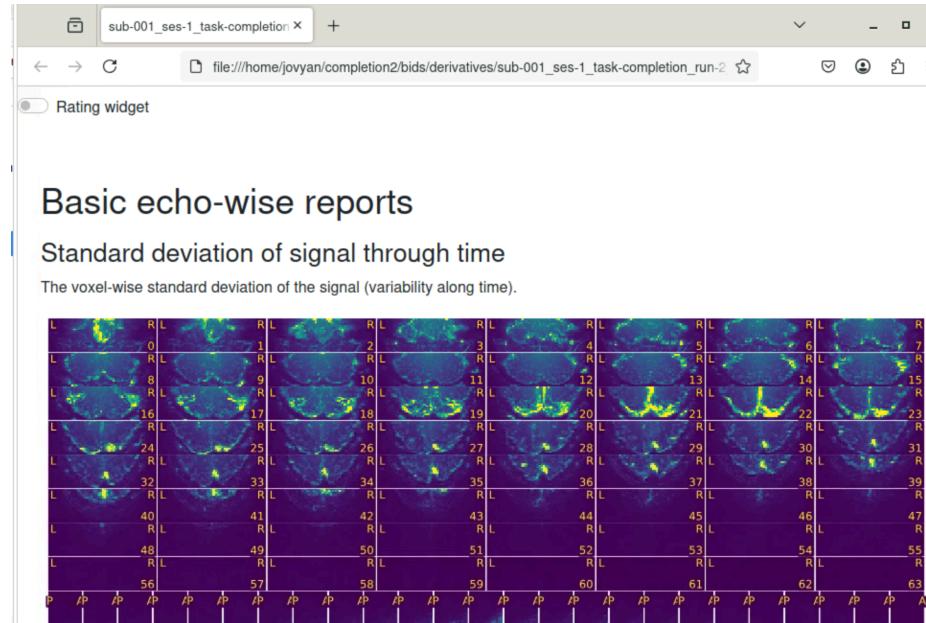
## Command line

```
# define relevant paths
bids_folder=/home/jovyan/gambling/bids/
mriqc_folder=/home/jovyan/gambling/bids/derivatives/mriqc/

# create mriqc directory
mkdir -p $mriqc_doler

# run mriqc
mriqc $bids_folder $mriqc_folder participant --participant-label 999
```

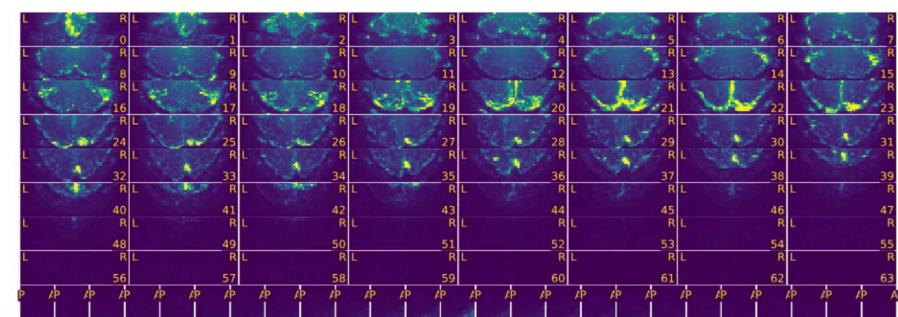
## Output



## Basic echo-wise reports

### Standard deviation of signal through time

The voxel-wise standard deviation of the signal (variability along time).



## References

- Brooks, Joseph L. 2012. “Counterbalancing for Serial Order Carryover Effects in Experimental Condition Orders.” *Psychological Methods* 17 (4): 600–614. <https://doi.org/10.1037/a0029310>.

Juavinett, Ashley L. 2022. "The Next Generation of Neuroscientists Needs to Learn How to Code, and We Need New Ways to Teach Them." *Neuron* 110 (4): 576–78. <https://doi.org/10.1016/j.neuron.2021.12.001>.