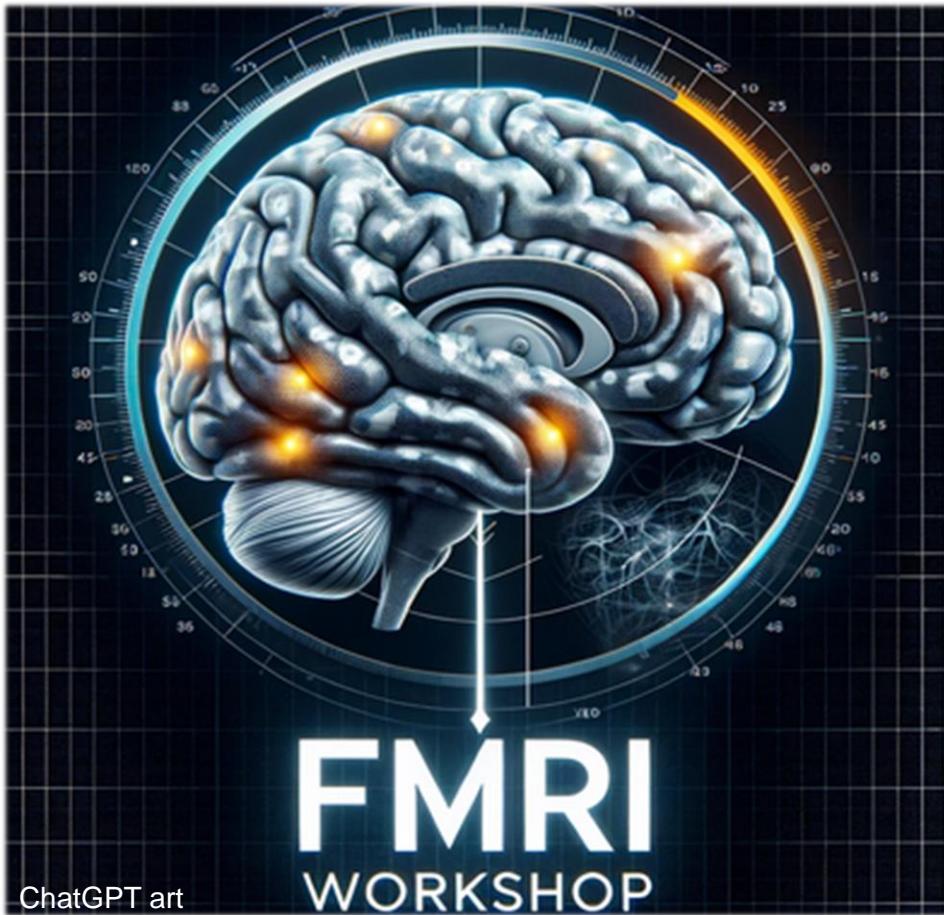




MRC Cognition  
and Brain  
Sciences Unit



UNIVERSITY OF  
CAMBRIDGE



# Functional Magnetic Resonance Imaging



**GitHub** [https://github.com/dcdace/fMRI\\_training](https://github.com/dcdace/fMRI_training)

Dace Apšvalka

Winter, 2024

# Outline

---

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

www.nature.com/scientificdata

# SCIENTIFIC DATA



**OPEN**

**SUBJECT CATEGORIES**

- » Electroencephalography
- EEG
- » Brain imaging
- » Functional magnetic resonance imaging
- » Cognitive neuroscience

Received: 07 April 2014  
Accepted: 05 January 2015  
Published: 20 January 2015

## A multi-subject, multi-modal human neuroimaging dataset

Daniel G. Wakeman<sup>1,2</sup> & Richard N. Henson<sup>2</sup>

We describe data acquired with multiple functional and structural neuroimaging modalities on the same nineteen healthy volunteers. The functional data include Electroencephalography (EEG), Magnetoencephalography (MEG) and functional Magnetic Resonance Imaging (fMRI) data, recorded while the volunteers performed multiple runs of hundreds of trials of a simple perceptual task on pictures of familiar, unfamiliar and scrambled faces during two visits to the laboratory. The structural data include T1-weighted MPRAGE, Multi-Echo FLASH and Diffusion-weighted MR sequences. Though only from a small sample of volunteers, these data can be used to develop methods for integrating multiple modalities from multiple runs on multiple participants, with the aim of increasing the spatial and temporal resolution above that of any one modality alone. They can also be used to integrate measures of functional and structural connectivity, and as a benchmark dataset to compare results across the many neuroimaging analysis packages. The data are freely available from <https://openfmri.org/>.

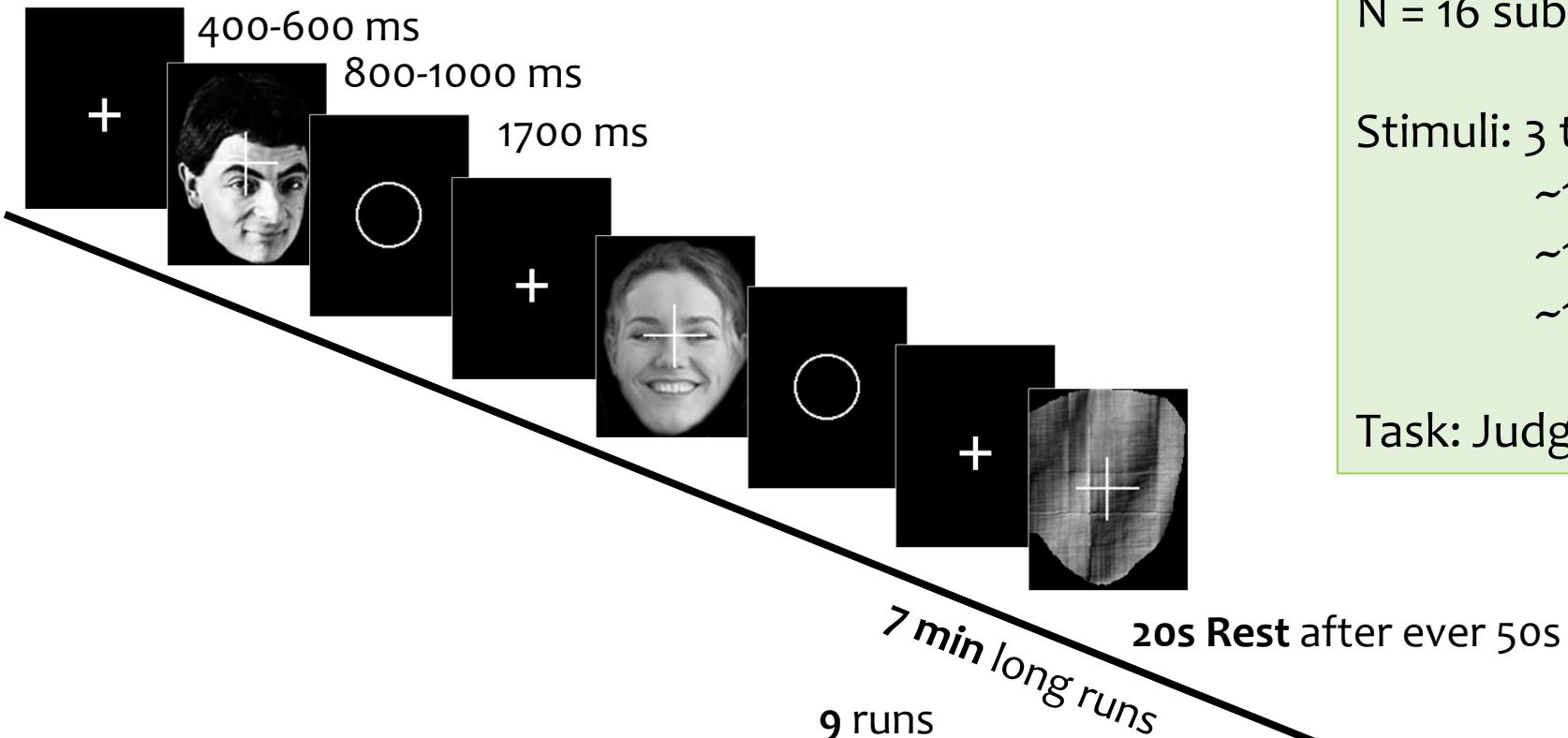
**Wakeman & Henson (2015), Scientific Data, <http://www.nature.com/articles/sdata20151>**

# Example Dataset

<https://openneuro.org/datasets/ds000117/versions/1.0.5>

- Been used in many methods projects and publications, as well as tutorials (e.g. “multimodal” dataset in SPM12 manual)
- Here we will analyse it from the very root – the raw DICOM images

# Experiment: Face Recognition



N = 16 subjects

Stimuli: 3 types of greyscale face images:

~150 x Famous

~150 x Unfamiliar

~150 x Scrambled

Task: Judge face symmetry

# Environment

**Data**

Organise & Manage

Pre-process

Analyse

Report

# Environment

## PROGRAMMING LANGUAGES



### BASH & Shell Scripts

A low-level programming language providing a command line user interface for Unix-like operating systems (e.g., Linux, macOS).

Used to automate repetitive tasks and manage system processes and resources.



A high-level, general-purpose programming language.  
**License-free** – good for reproducible & open code.



A high-level programming language designed for engineers and scientists.

**Requires a license.** Provides loads of useful resources for Neuroimaging analysis.

# Environment

## PROGRAMMING LANGUAGES



### BASH & Shell Scripts

A low-level programming language providing a command line user interface for Unix-like operating systems (e.g., Linux, macOS).

Used to automate repetitive tasks and manage system processes and resources.



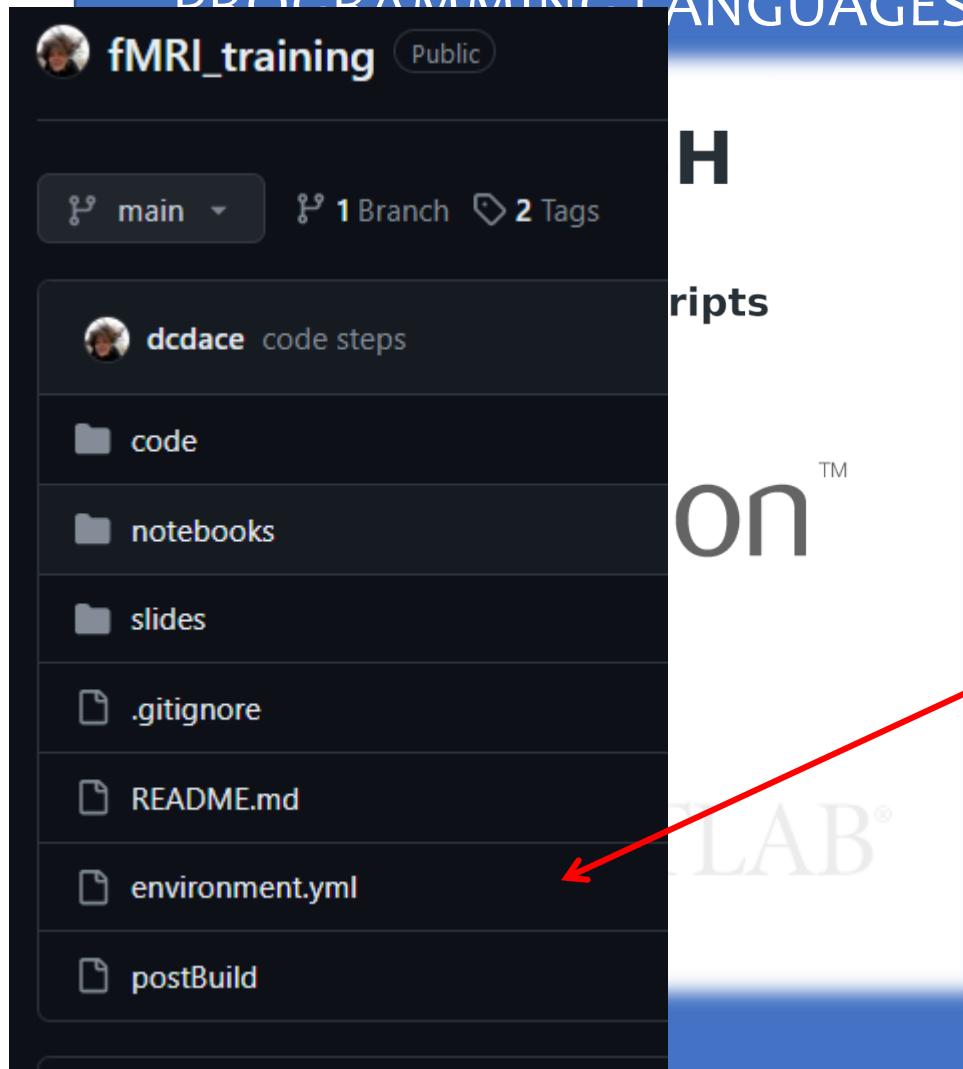
A high-level, general-purpose programming language.  
**License-free** – good for reproducible & open code.



A high-level programming language designed for engineers and scientists.

**Requires a license.** Provides loads of useful resources for Neuroimaging analysis.

# Environment



## PACKAGE MANAGER

**CONDA**

Conda is an open-source, cross-platform, language-agnostic package manager and environment management system.

With conda, you can use environments that have different versions of Python and packages installed in them.

You can, for example, create your **fMRI analysis environment** that includes packages needed for your analysis work.

```
1 name: fmri
2 channels:
3   - conda-forge
4   - defaults
5 dependencies:
6   - dcm2niix
7   - git
8   - python=3.10
9   - ipython
10  - jupyter
11  - matplotlib
12  - numpy
13  - pandas
14  - pip
15  - pytest
16  - scikit-image
17  - scikit-learn
18  - scipy
19  - seaborn
20  - pip:
21    - atlasreader
22    - jupyter_contrib_nbextensions
23    - heudiconv
24    - nibabel
25    - niflow-nipype1-workflows
26    - nilearn
27    - nipy
28    - nipype
29    - nistats
30    - pybids
31    - plotly
32    - requests
```

# Environment

LANGUAGES

H

cripts

ON<sup>TM</sup>

FLAB®

PACKAGE MANAGER



Conda is an open-source, cross-platform, language-agnostic package manager and environment management system.

With conda, you can use environments that have different versions of Python and packages installed in them.

You can, for example, create your **fMRI analysis environment** that includes packages needed for your analysis work.

# Environment

## PROGRAMMING LANGUAGES



**BASH**  
&  
**Shell Scripts**



## PACKAGE MANAGER



## CODE EDITOR

Code editors simplify and speed up typing of source code. They also support running and debugging the code.

The best editors are cross-language, cross-platform and support version control.

# Environment

## PROGRAMMING LANGUAGES



## PACKAGE MANAGER



A screenshot of the Visual Studio Code (VS Code) interface. The interface includes:

- EXPLORER**: Shows a file tree with several notebooks and scripts under "OPEN EDITORS".
- WORKSHOPS (WORKSPACE)**: Shows a list of workshops and their contents.
- CODE**: Shows a list of files including "first\_level\_script.py" and "step08\_first\_level\_analysis.sh".
- TERMINAL**: Shows a command-line interface with a queue of tasks.
- PROBLEMS**: Shows 86 errors or warnings.
- OUTPUT**: Shows logs from the terminal.
- DEBUG CONSOLE**: Shows logs from the debugger.
- PORTS**: Shows network port information.
- JUPYTER**: Shows a Jupyter notebook interface.
- OUTLINE**, **TIMELINE**, and **PROPERTIES** panels at the bottom.

The code editors contain Python and Bash scripts for fMRI analysis. A large blue 'X' is overlaid on the right side of the interface.

```
2024-CBU > code > $ first_level_script.py ...  
1 #!/usr/bin/env python3  
2 # -*- coding: utf-8 -*-  
3 # ======  
4 # Dace Apšvalka (MRC CBU 2024)  
5 # First level fMRI analysis using Nilearn  
6 #  
7 # This script requires step08_first_level_analysis.sh, unless  
8 # and sID here manually  
9 #  
10 # ======  
11 #  
12 # ======  
13 # IMPORT RELEVANT PACKAGES  
14 #  
15 import os  
16 import sys  
17 import pandas as pd  
18 import numpy as np  
19 from bids.layout import BIDSLayout  
20 from nilearn.glm.first_level import FirstLevelModel  
21 import time  
22 import warnings  
23 warnings.filterwarnings("ignore")  
24 # ======  
25 # DEFINE PATHS  
26 # arguments passed from step07_first_level_analysis.sh  
27 #  
28 # ds = sys.argv[1] # dataset location  
29 ds = sys.argv[1].split("-sub-")[1] # subject id  
30 #  
31 #  
32 #  
33 print("Running first-level analysis for subject " + sID)  
34 start_time = time.time()  
35 print("Started at: " + time.strftime("%H:%M:%S", time.localtime()))  
36 #  
37 #  
38 # DEFINE PARAMETERS  
39 #  
40 model_name = 'first-level'  
41 #  
42 bids_path = os.path.join(ds, 'data')  
  
2024-CBU > code > $ step08_first_level_analysis.sh ...  
1 #!/bin/bash  
2 #  
3 # The script will run the first_level_script.py script for each  
4 # subject in the dataset.  
5 # You will need a conda environment that contains the required  
6 # packages.  
7 # Usage:  
8 # Configure the variables below and run the script: ./step08_...  
9 #  
10 #  
11 #  
12 # [FILL IN THE VARIABLES BELOW]  
13 #  
14 #  
15 # Your project's path  
16 PROJECT_PATH="/.../a05/workshops/2024-CBU"  
17 # Location of the first_level_script.py analysis script  
18 ANALYSIS_SCRIPT_PATH="$PROJECT_PATH/first_level_script.py"  
19 # Location of the job logs  
20 LOG_DIR="$PROJECT_PATH/job_logs"  
21 # Location of the BIDS dataset  
22 SUBJECT_LIST="$PROJECT_PATH/subject_list.txt"  
23 #  
24 #  
25 # Get the subject list  
26 SUBJECT_LISTS=$(cat $SUBJECT_LIST)  
27 SUBJECT_LISTS=$(echo $SUBJECT_LIST | sed -e 's/ /,/g')  
28 SUBJECT_LIST="\"$SUBJECT_LISTS\""  
29 #  
30 # You don't have to change anything below this line!  
31 #  
32 #  
33 #  
34 #  
35 #  
36 #  
37 #  
38 #  
39 #  
40 #  
41 # CD to the job logs directory  
42 mkdir -p "$LOG_DIR"  
..
```

Visual Studio Studio Code

Tutorial/Hands-On Notebooks

You can access the hands-on notebooks in two ways:

- python3.1 2024-C...
- bin/tcsh

Ln 28, Col 73 Spaces: 4 UTF-8 LF Python 3.10.7 64-bit

# Environment

## PROGRAMMING LANGUAGES



**BASH**  
&  
**Shell Scripts**



## PACKAGE MANAGER

**CONDA**



Visual Studio Code



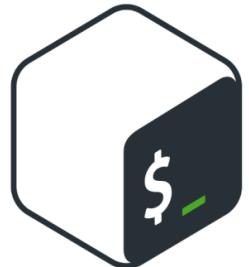
## CODE EDITOR

## VERSION CONTROL

The practice of tracking and managing changes to software code - your analysis code.  
It allows to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, do collaborative coding etc.

# Environment

PROGRAMMING LANGUAGES



**BASH**  
&  
**Shell Scripts**



PACKAGE MANAGER

**CONDA**



Visual Studio Code



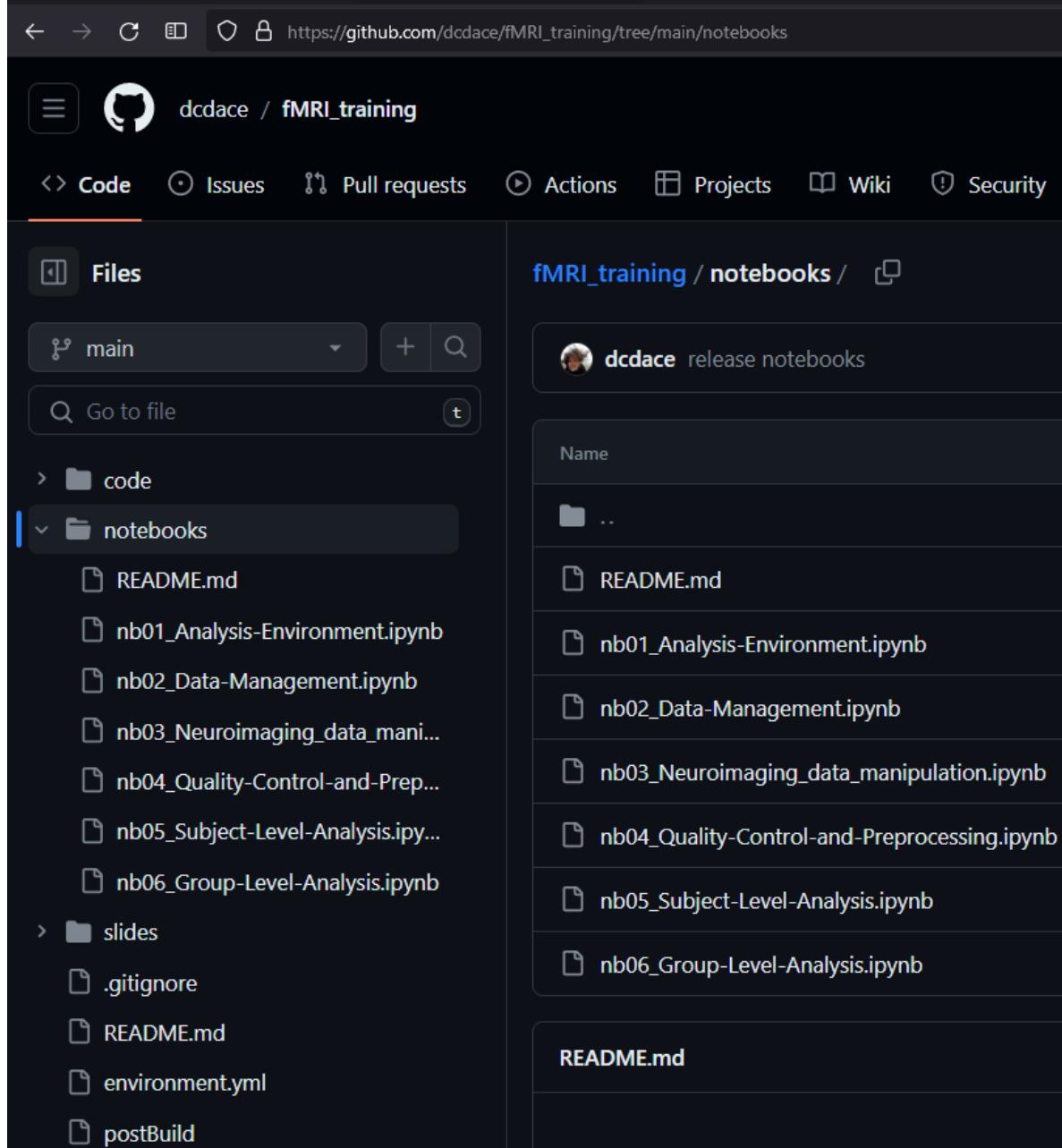
CODE EDITOR

VERSION CONTROL

ANALYSIS NOTEBOOK



# Analysis notebooks



[https://github.com/dcdace/fMRI\\_training](https://github.com/dcdace/fMRI_training)

You can access the notebooks of this workshop in three ways:

1. Interactively through MyBinder  [launch binder](#)
  - Allows you to run Jupyter notebooks directly online
  - Restricted computational environment (1-2GB of RAM)
2. Download the notebooks and run them on your computer
  - The conda environment with the packages required for that to work is in `environment.yml` file
  - The example dataset with 2 subjects can be downloaded from Dropbox:  
<https://dl.dropboxusercontent.com/s/q03ocu844joczm6/FaceRecognition.zip>
3. View static notebooks
  - You can also view the notebooks with all solutions provided just by clicking on them in the `notebooks` directory

# Environment

PROGRAMMING LANGUAGES



**BASH**  
&

Shell Scripts



Notebook: [nbo1\\_Analysis-Environment.ipynb](#)



PACKAGE MANAGER

CODE EDITOR



**CONDA**

VERSION CONTROL



ANALYSIS NOTEBOOK





# Environment

**Data**

Organise & Manage

Pre-process

Analyse

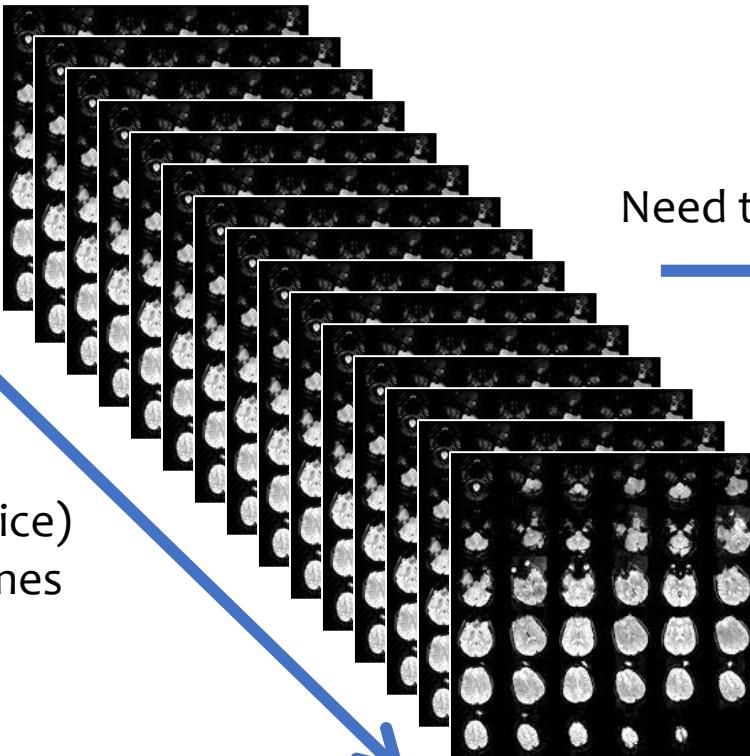
Report

# fMRI file formats

Collect the data

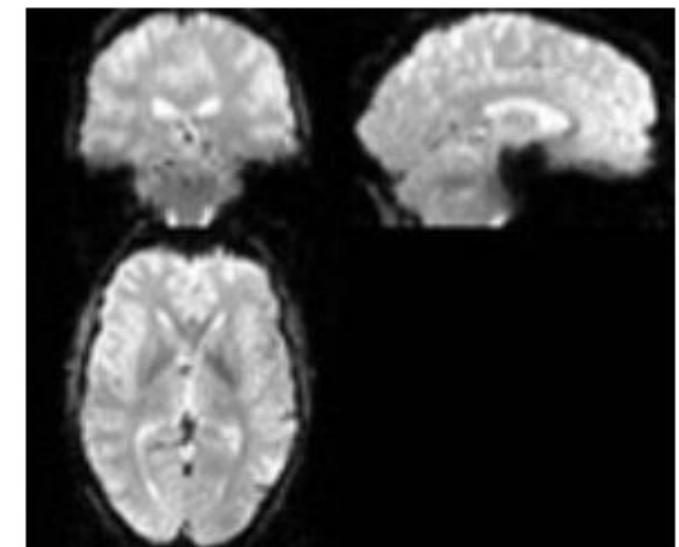


DICOM format



Need to convert to NIfTI

**functional scan**  
A brain volume (slice-by-slice)  
scanned every 2s > 100 times



# DICOM

## Anatomical (T1w) scans

Name
20090826_164150
Series_001_CBU_Localiser
<b>Series_002_CBU_MPRAge</b>
Series_003_CBU_DWEPI_BOLD210
Series_004_CBU_DWEPI_BOLD210
Series_005_CBU_DWEPI_BOLD210
Series_006_CBU_DWEPI_BOLD210
Series_007_CBU_DWEPI_BOLD210
Series_008_CBU_DWEPI_BOLD210
Series_009_CBU_DWEPI_BOLD210
Series_010_CBU_DWEPI_BOLD210
Series_011_CBU_DWEPI_BOLD210
Series_012_CBU_FieldMapping
Series_013_CBU_FieldMapping
DCM File (192)
1.3.12.2.1107.5.2.32.35119.2009082616480248824672575.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480263974172579.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480266173672581.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480282359572587.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480283389772589.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480291722972597.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480294149872599.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480299170772601.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480312524272609.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480314408372615.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480323872972617.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480324494972619.dcm
1.3.12.2.1107.5.2.32.35119.2009082616480334100772625.dcm

## Functional (T2\*/BOLD) scans

Name
20090826_164150
Series_001_CBU_Localiser
Series_002_CBU_MPRAge
<b>Series_003_CBU_DWEPI_BOLD210</b>
Series_004_CBU_DWEPI_BOLD210
Series_005_CBU_DWEPI_BOLD210
Series_006_CBU_DWEPI_BOLD210
Series_007_CBU_DWEPI_BOLD210
Series_008_CBU_DWEPI_BOLD210
Series_009_CBU_DWEPI_BOLD210
Series_010_CBU_DWEPI_BOLD210
Series_011_CBU_DWEPI_BOLD210
Series_012_CBU_FieldMapping
Series_013_CBU_FieldMapping
DCM File (208)
1.3.12.2.1107.5.2.32.35119.200908261649572708873570.dcm
1.3.12.2.1107.5.2.32.35119.200908261649592579973671.dcm
1.3.12.2.1107.5.2.32.35119.200908261650011521873772.dcm
1.3.12.2.1107.5.2.32.35119.200908261650033002573873.dcm
1.3.12.2.1107.5.2.32.35119.200908261650052824473974.dcm
1.3.12.2.1107.5.2.32.35119.200908261650072679074075.dcm
1.3.12.2.1107.5.2.32.35119.200908261650092563574176.dcm
1.3.12.2.1107.5.2.32.35119.20090826165011450474277.dcm
1.3.12.2.1107.5.2.32.35119.200908261650132877074378.dcm
1.3.12.2.1107.5.2.32.35119.200908261650152763374479.dcm
1.3.12.2.1107.5.2.32.35119.200908261650172643174580.dcm
1.3.12.2.1107.5.2.32.35119.200908261650192479574681.dcm
1.3.12.2.1107.5.2.32.35119.200908261650211329574782.dcm

- **DICOM - Digital Imaging and Communications in Medicine (.dcm)**

- Raw data standard for storing and communicating **medical images**
- Contains a **header** (meta data) and the actual **image** itself
- Images are stored in **2D layers**
- A separate file for each **slice** (T1w) or **volume** (BOLD)

# DICOM: Anatomical

DICOM browser

Search...

All patients(Patients: 1, Images: 192)

CBU090938

CBU090938

CBU Neuroimaging(MR: 1 series)

CBU\_MPRAGE

1.3.12.2.1107.5.2.32.35119.2009082609555636896484776.c  
1.3.12.2.1107.5.2.32.35119.2009082609555632008284776.c  
1.3.12.2.1107.5.2.32.35119.2009082609555644050484784.c  
1.3.12.2.1107.5.2.32.35119.2009082609555649109184786.c  
1.3.12.2.1107.5.2.32.35119.2009082609555660191584794.c  
1.3.12.2.1107.5.2.32.35119.20090826095556679284792.c  
1.3.12.2.1107.5.2.32.35119.2009082609555669059484802.c  
1.3.12.2.1107.5.2.32.35119.2009082609555674269184804.c  
1.3.12.2.1107.5.2.32.35119.2009082609555685752984810.c  
1.3.12.2.1107.5.2.32.35119.2009082609555686170384812.c  
1.3.12.2.1107.5.2.32.35119.2009082609555694120284816.c  
1.3.12.2.1107.5.2.32.35119.200908260955569677984818.c  
1.3.12.2.1107.5.2.32.35119.200908260955572374084824.d  
1.3.12.2.1107.5.2.32.35119.200908260955575468484826.d  
1.3.12.2.1107.5.2.32.35119.2009082609555716376884832.c  
1.3.12.2.1107.5.2.32.35119.2009082609555723796484838.c  
1.3.12.2.1107.5.2.32.35119.200908260955572548434840.c  
1.3.12.2.1107.5.2.32.35119.200908260955573303654842.c  
1.3.12.2.1107.5.2.32.35119.2009082609555743189084846.c  
1.3.12.2.1107.5.2.32.35119.200908260955575436474854.c  
1.3.12.2.1107.5.2.32.35119.2009082609555759470484860.c  
1.3.12.2.1107.5.2.32.35119.200908260955576839974866.c  
1.3.12.2.1107.5.2.32.35119.2009082609555770769984870.c  
1.3.12.2.1107.5.2.32.35119.2009082609555790816784884.c  
1.3.12.2.1107.5.2.32.35119.2009082609555789094684880.c  
1.3.12.2.1107.5.2.32.35119.20090826095558444984892.d  
1.3.12.2.1107.5.2.32.35119.200908260955581570984888.d  
1.3.12.2.1107.5.2.32.35119.2009082609555821034484902.c  
1.3.12.2.1107.5.2.32.35119.2009082609555816681284898.c  
1.3.12.2.1107.5.2.32.35119.200908260955582696354908.c  
1.3.12.2.1107.5.2.32.35119.2009082609555832137684912.c  
1.3.12.2.1107.5.2.32.35119.2009082609555844678084918.c  
1.3.12.2.1107.5.2.32.35119.2009082609555842377884916.c  
1.3.12.2.1107.5.2.32.35119.2009082609555851923884924.c  
1.3.12.2.1107.5.2.32.35119.200908260955585706184926.c  
1.3.12.2.1107.5.2.32.35119.2009082609555865957784932.c  
1.3.12.2.1107.5.2.32.35119.2009082609555869132184934.c  
1.3.12.2.1107.5.2.32.35119.2009082609555875856184940.c  
1.3.12.2.1107.5.2.32.35119.2009082609555881020184944.c  
1.3.12.2.1107.5.2.32.35119.2009082609555888987984948.c  
1.3.12.2.1107.5.2.32.35119.20090826095558915829284964.c  
1.3.12.2.1107.5.2.32.35119.2009082609555897357184952.c  
1.3.12.2.1107.5.2.32.35119.200908260955595014584956.d  
1.3.12.2.1107.5.2.32.35119.2009082609555912645334962.c  
1.3.12.2.1107.5.2.32.35119.2009082609555920054584968.c  
1.3.12.2.1107.5.2.32.35119.2009082609555942028984980.c  
1.3.12.2.1107.5.2.32.35119.2009082609555928045984972.c  
1.3.12.2.1107.5.2.32.35119.2009082609555936426884978.c  
1.3.12.2.1107.5.2.32.35119.2009082609555949691084988.c  
1.3.12.2.1107.5.2.32.35119.200908260955596797184998.c  
1.3.12.2.1107.5.2.32.35119.200908260955597421284990.c  
1.3.12.2.1107.5.2.32.35119.200908260955598515485000.c  
1.3.12.2.1107.5.2.32.35119.2009082609555979689485004.c  
1.3.12.2.1107.5.2.32.35119.2009082609555981592785008.c  
1.3.12.2.1107.5.2.32.35119.2009082609555981609235012.c  
1.3.12.2.1107.5.2.32.35119.200908260955599414545016.c  
1.3.12.2.1107.5.2.32.35119.20090826095600206585020.dcr  
1.3.12.2.1107.5.2.32.35119.200908260956002234685022.d  
1.3.12.2.1107.5.2.32.35119.200908260956007930285026.d  
1.3.12.2.1107.5.2.32.35119.2009082609560013313885030.c  
1.3.12.2.1107.5.2.32.35119.2009082609560015539585032.c  
1.3.12.2.1107.5.2.32.35119.2009082609560023177985036.c  
1.3.12.2.1107.5.2.32.35119.2009082609560024350785038.c  
1.3.12.2.1107.5.2.32.35119.2009082609560030415585040.c  
1.3.12.2.1107.5.2.32.35119.2009082609560035837385044.v



26-August-2009 9:51:26

MRC-CBU

TrioTim

DICOM Tags

Search...

Patient Name	CBU090938
PatientID	MRO9029
Patient Birth Date	19830904
Patient Sex	M
Patient Age	25Y
Patient Weight	70.000000
Patient Address	
Study Date	26-August-2009
Study Time	9:43:37
Study ID	1
Study Modality	MR
Study Description	CBU Neuroimaging
Series Date	26-August-2009
Series Time	9:55:56
Series Description	CBU_MPRAGE

Patient information All Tags Favorite

# DICOM: Functional

DICOM browser Search... X

All patients(Patients: 1, Images: 208)

CBU090938

CBU Neuroimaging(MR 1 series)

CBU\_DWEPI\_BOLD210

13.1.2.2.1107.5.2.32.35119.200908260959346867948  
13.1.2.2.1107.5.2.32.35119.200908260959366845383  
13.1.2.2.1107.5.2.32.35119.200908260959386733138  
13.1.2.2.1107.5.2.32.35119.20090826095940683650  
13.1.2.2.1107.5.2.32.35119.200908260959426865646  
13.1.2.2.1107.5.2.32.35119.200908260959446855223  
13.1.2.2.1107.5.2.32.35119.200908260959466836951  
13.1.2.2.1107.5.2.32.35119.200908260959486727751  
13.1.2.2.1107.5.2.32.35119.200908260959506875051  
13.1.2.2.1107.5.2.32.35119.200908260959526863591  
13.1.2.2.1107.5.2.32.35119.200908260959546854651  
13.1.2.2.1107.5.2.32.35119.2009082609595668323861  
13.1.2.2.1107.5.2.32.35119.200908260959586724711  
13.1.2.2.1107.5.2.32.35119.200908261000006532021  
13.1.2.2.1107.5.2.32.35119.20090826100002685521  
13.1.2.2.1107.5.2.32.35119.200908261000046854481  
13.1.2.2.1107.5.2.32.35119.200908261000066829361  
13.1.2.2.1107.5.2.32.35119.200908261000106698061  
13.1.2.2.1107.5.2.32.35119.200908261000126845731  
13.1.2.2.1107.5.2.32.35119.200908261000146845311  
13.1.2.2.1107.5.2.32.35119.200908261000166822831  
13.1.2.2.1107.5.2.32.35119.200908261000186800301  
13.1.2.2.1107.5.2.32.35119.200908261000206692331  
13.1.2.2.1107.5.2.32.35119.200908261000226844571  
13.1.2.2.1107.5.2.32.35119.200908261000246830401  
13.1.2.2.1107.5.2.32.35119.200908261000266809951  
13.1.2.2.1107.5.2.32.35119.200908261000286803781  
13.1.2.2.1107.5.2.32.35119.200908261000306684831  
13.1.2.2.1107.5.2.32.35119.2009082610003264699401  
13.1.2.2.1107.5.2.32.35119.200908261000346828561  
13.1.2.2.1107.5.2.32.35119.200908261000366867901  
13.1.2.2.1107.5.2.32.35119.20090826100038671901  
13.1.2.2.1107.5.2.32.35119.200908261000406778351  
13.1.2.2.1107.5.2.32.35119.200908261000426668311  
13.1.2.2.1107.5.2.32.35119.200908261000446811981  
13.1.2.2.1107.5.2.32.35119.200908261000466800411  
13.1.2.2.1107.5.2.32.35119.200908261000496784121  
13.1.2.2.1107.5.2.32.35119.200908261000506776241  
13.1.2.2.1107.5.2.32.35119.200908261000526662551  
13.1.2.2.1107.5.2.32.35119.200908261000546810231  
13.1.2.2.1107.5.2.32.35119.200908261000566791881  
13.1.2.2.1107.5.2.32.35119.20090826100058677861  
13.1.2.2.1107.5.2.32.35119.200908261001006761571  
13.1.2.2.1107.5.2.32.35119.200908261001026646311  
13.1.2.2.1107.5.2.32.35119.200908261001046454191  
13.1.2.2.1107.5.2.32.35119.200908261001066789611  
13.1.2.2.1107.5.2.32.35119.200908261001086768091  
13.1.2.2.1107.5.2.32.35119.200908261001086770191  
13.1.2.2.1107.5.2.32.35119.200908261001126740631  
13.1.2.2.1107.5.2.32.35119.200908261001146628781  
13.1.2.2.1107.5.2.32.35119.200908261001166764511  
13.1.2.2.1107.5.2.32.35119.200908261001206746451  
13.1.2.2.1107.5.2.32.35119.200908261001236737841  
13.1.2.2.1107.5.2.32.35119.200908261001246633771  
13.1.2.2.1107.5.2.32.35119.200908261001266774791  
13.1.2.2.1107.5.2.32.35119.200908261001286757091  
13.1.2.2.1107.5.2.32.35119.200908261001306741711  
13.1.2.2.1107.5.2.32.35119.200908261001326731881  
13.1.2.2.1107.5.2.32.35119.200908261001346618321  
13.1.2.2.1107.5.2.32.35119.200908261001366425231  
13.1.2.2.1107.5.2.32.35119.200908261001386751891  
13.1.2.2.1107.5.2.32.35119.200908261001406737591  
13.1.2.2.1107.5.2.32.35119.200908261001426712471  
AS R L PI

MRC-CBU  
TrioTim  
26-August-2009 9:59:31

DICOM Tags Search... Patient information All Tags Favorite Ta

ST: 3 SL: -50.170833  
RT: 2000 ET: 30  
FS: 3  
MR  
LittleEndianImplicit  
Images: 1/208  
Series: 3

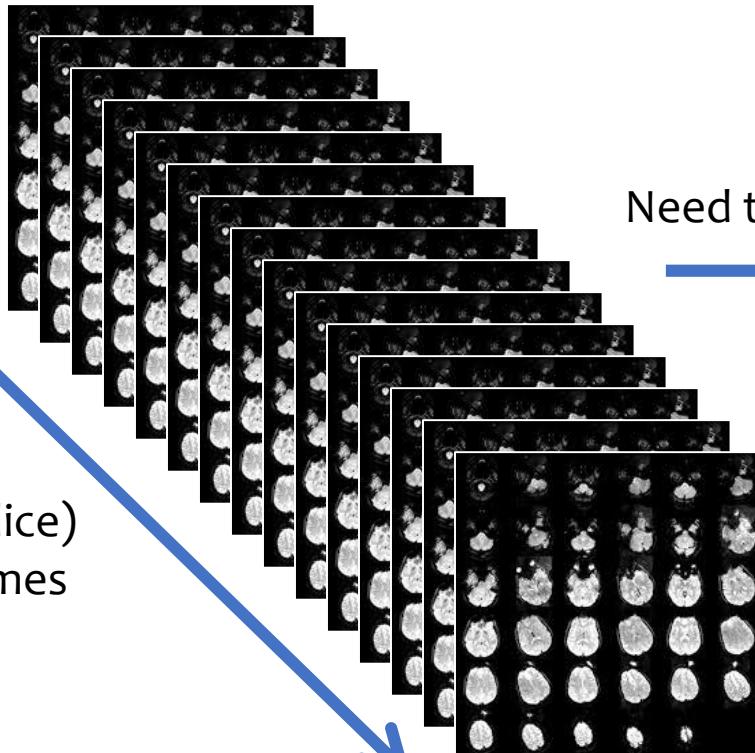
Zoom: 327% WL: 743 WW: 1555

Collect the data



## Anatomical (T1w) image & Functional (T2\*/BOLD) image

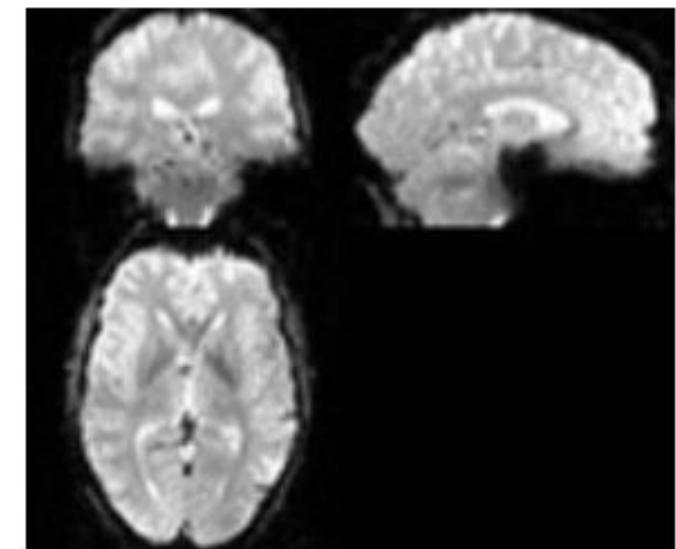
DICOM format



Need to convert to NIfTI

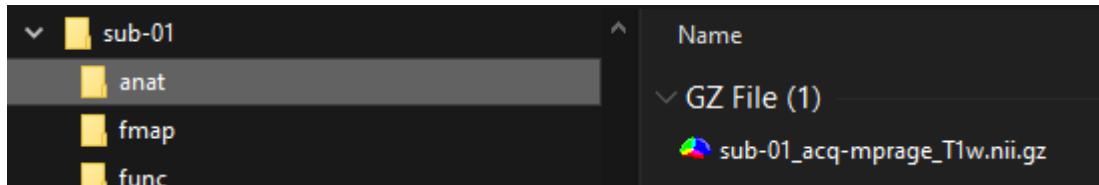
**functional scan**

A brain volume (slice-by-slice)  
scanned every 2s > 100 times



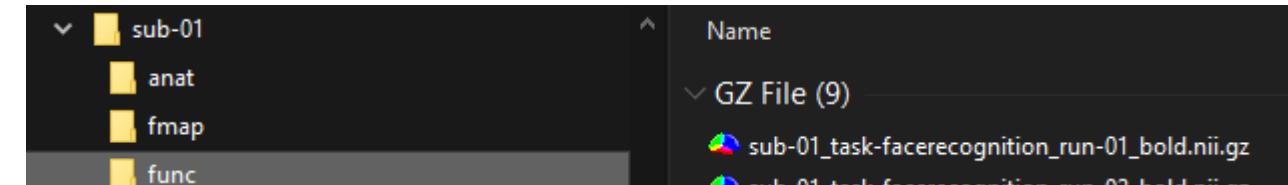
# NIfTI

Anatomical (T1w) scans



3D file

Functional (T2\*/BOLD) scans



4D file

- **NIfTI – Neuroimaging Informatics Technology Initiative (.nii, .nii.gz)**
  - Standardised representation of **brain images**, cross-platform, cross-software
  - Contains **header** and **image**
  - 3D or 4D files (all slices/volumes in a single file)
- **DICOM vs NIfTI**
  - All medical vs brain images
  - 2D layer files vs 3D/4D files
  - More vs less metadata
    - When converting DICOM to NIfTI need to be aware of that and save all metadata that might be needed
- **DICOM to NIfTI**
  - Several tools are available
    - For example, **dcm2niix** Python package

Collect the data



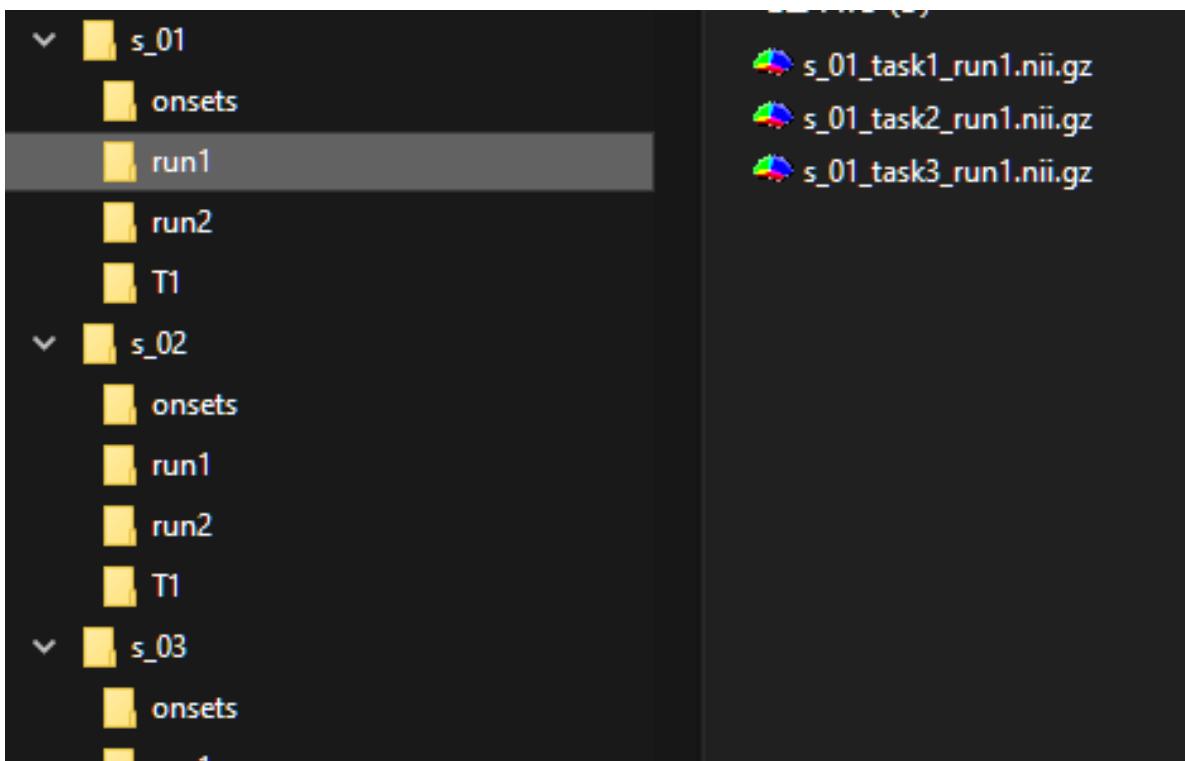
## Anatomical (T1w) image & Functional (T2\*/BOLD) image

- sub-01\_T1w.nii
- sub-01\_run-01\_bold.nii
- sub-01\_run-02\_bold.nii
- sub-02\_T1w.nii
- sub-02\_run-01\_bold.nii
- sub-02\_run-02\_bold.nii
- ...
- sub-100\_T1w.nii
- sub-100\_run-01\_bold.nii
- sub-100\_run-02\_bold.nii
- and even more files (more sessions and tasks)

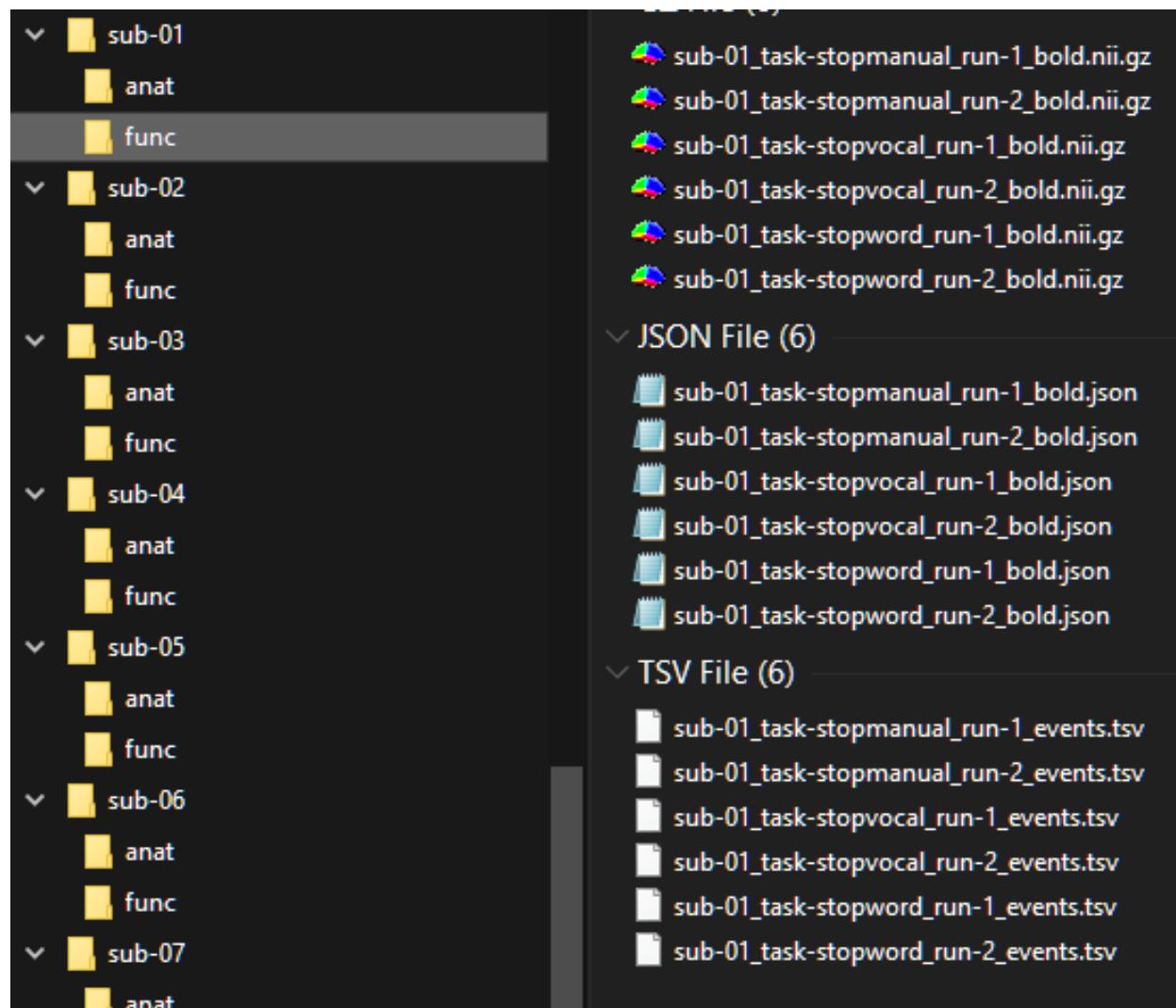
How should we  
organise our files?

# File organisation

## Example 1



## Example 2



# fMRI data management



- **Problems with heterogeneity in data management**

- Difficult for others (and you!) to understand your data and keep track of changes
- Scripts have to be adapted (can't be easily reused)
- Huge effort to automate workflows and no way to automatically validate data sets
- Sharing data becomes a hustle

Wouldn't it be much easier if everybody organised the files in the same way?

# fMRI data management

- A standardised way for organising & describing neuroimaging data



## Brain Imaging Data Structure - **BIDS**



**Stanford** | Center for Reproducible  
Neuroscience

- Documentation: <https://bids-specification.readthedocs.io/en/latest/>

SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES  
» Data publication and  
archiving  
» Research data

The brain imaging data structure,  
a format for organizing and  
describing outputs of neuroimaging  
experiments

Received: 18 December 2015  
Accepted: 19 May 2016  
Published: 21 June 2016

Krzysztof J. Gorgolewski<sup>1</sup>, Tibor Auer<sup>2</sup>, Vince D. Calhoun<sup>3,4</sup>, R. Cameron Craddock<sup>5,6</sup>, Samir Das<sup>7</sup>, Eugene P. Duff<sup>8</sup>, Guillaume Flandin<sup>9</sup>, Satrajit S. Ghosh<sup>10,11</sup>, Tristan Glantz<sup>12</sup>, Yaroslav O. Halchenko<sup>13</sup>, Daniel A. Handwerker<sup>14</sup>, Michael Hanke<sup>15,16</sup>, David Keator<sup>17</sup>, Xiangrui Li<sup>18</sup>, Zachary Michael<sup>19</sup>, Camille Maumet<sup>20</sup>, B. Nolan Nichols<sup>21,22</sup>, Thomas E. Nichols<sup>20,23</sup>, John Pellman<sup>6</sup>, Jean-Baptiste Poline<sup>24</sup>, Ariel Rokem<sup>25</sup>, Gunnar Schaefer<sup>1,26</sup>, Vanessa Sochat<sup>27</sup>, William Triplett<sup>1</sup>, Jessica A. Turner<sup>3,28</sup>, Gaël Varoquaux<sup>29</sup> & Russell A. Poldrack<sup>1</sup>

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods

Krzysztof J. Gorgolewski<sup>1\*</sup>, Fidel Alfaro-Almagro<sup>2</sup>, Tibor Auer<sup>3</sup>, Pierre Bellec<sup>4,5</sup>, Mihai Capota<sup>6</sup>, M. Mallar Chakravarty<sup>7,8</sup>, Nathan W. Churchill<sup>9</sup>, Alexander Li Cohen<sup>10</sup>, R. Cameron Craddock<sup>11,12</sup>, Gabriel A. Devenyi<sup>7,8</sup>, Anders Eklund<sup>13,14,15</sup>, Oscar Esteban<sup>1</sup>, Guillaume Flandin<sup>16</sup>, Satrajit S. Ghosh<sup>17,18</sup>, J. Swaroop Guntupalli<sup>19</sup>, Mark Jenkinson<sup>2</sup>, Anisha Keshavan<sup>20</sup>, Gregory Kiar<sup>21,22</sup>, Franziskus Liem<sup>23</sup>, Pradeep Reddy Raamana<sup>24,25</sup>, David Raffelt<sup>26</sup>, Christopher J. Steele<sup>7,8</sup>, Pierre-Olivier Quirion<sup>15</sup>, Robert E. Smith<sup>26</sup>, Stephen C. Strother<sup>24,25</sup>, Gaël Varoquaux<sup>27</sup>, Yida Wang<sup>6</sup>, Tal Yarkoni<sup>28</sup>, Russell A. Poldrack<sup>1</sup>





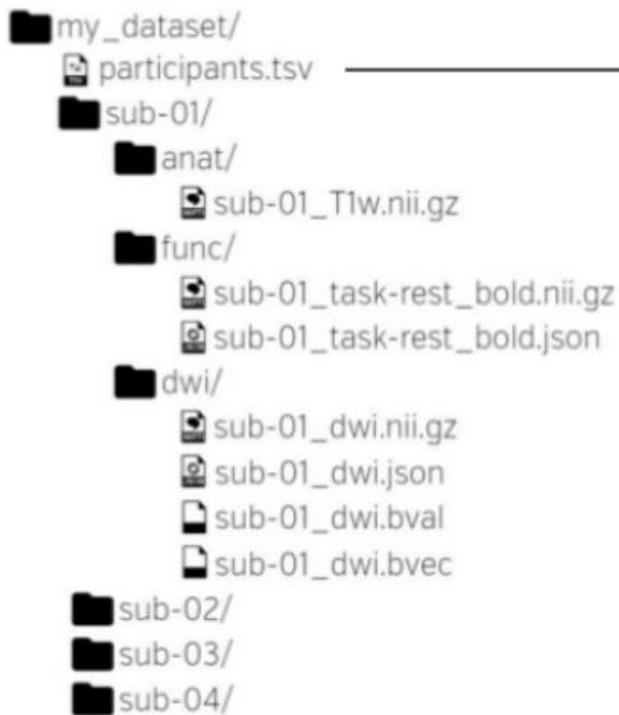
- **Benefits of BIDS**

- Easy for other people to work on your data (for collaborations or contract changes)
- Growing number of data analysis software packages that understand BIDS
- Databases, such as OpenNeuro and LORIS etc., accept and export datasets organised according to BIDS
- Validation tools that can check your dataset integrity and let you easily spot missing values



# BIDS

- Contains participant information

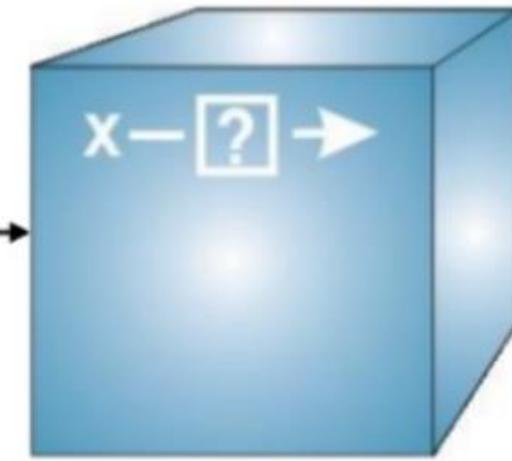
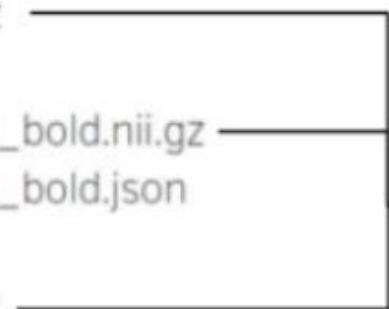


participant_id	age	sex
sub-001	34	M
sub-002	12	F
sub-003	33	F

# BIDS

- Contains data files

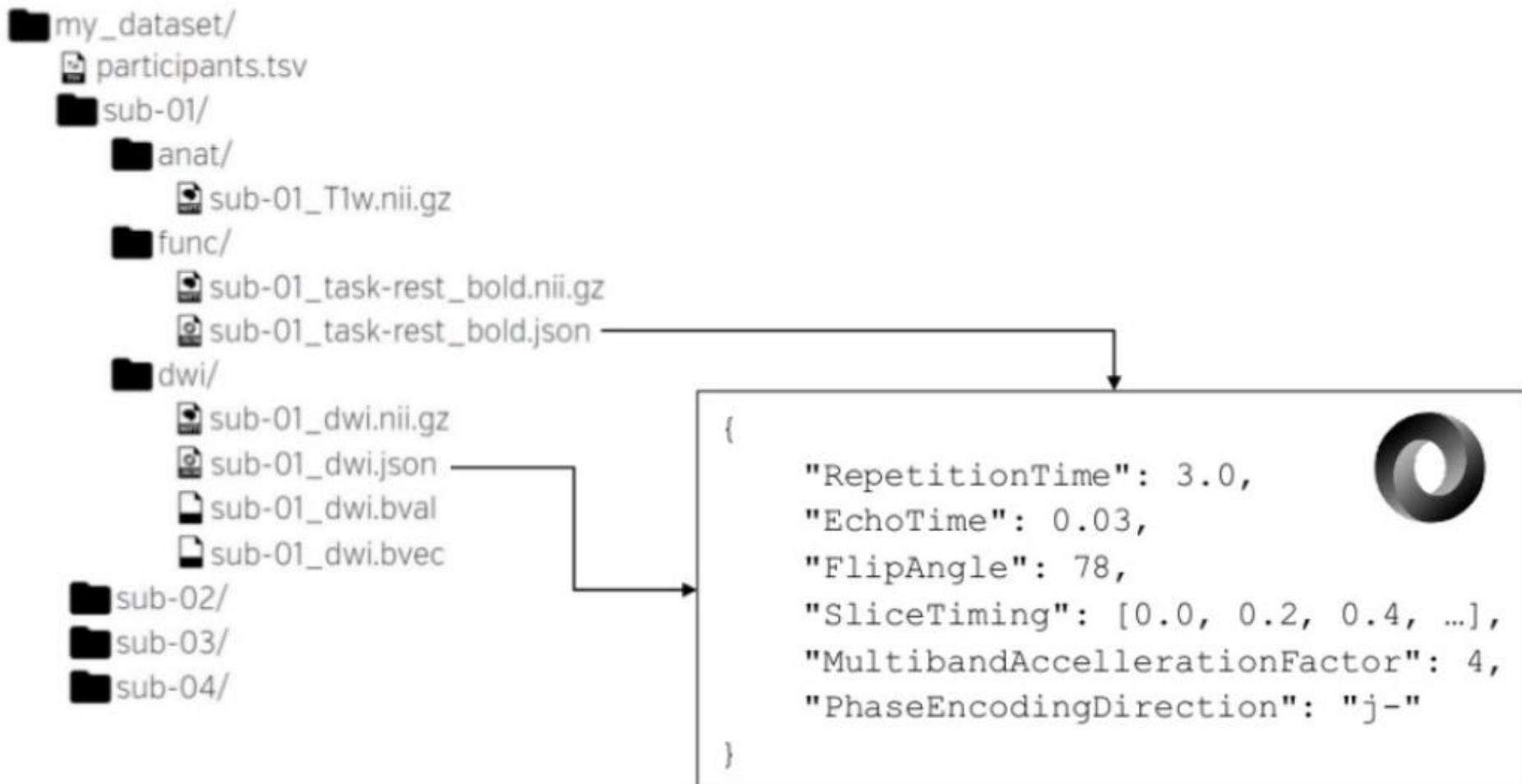
```
my_dataset/
  participants.tsv
  sub-01/
    anat/
      sub-01_T1w.nii.gz
    func/
      sub-01_task-rest_bold.nii.gz
      sub-01_task-rest_bold.json
    dwi/
      sub-01_dwi.nii.gz
      sub-01_dwi.json
      sub-01_dwi.bval
      sub-01_dwi.bvec
  sub-02/
  sub-03/
  sub-04/
```



NIfTI

# BIDS

- Contains study specific JSON (metadata) files



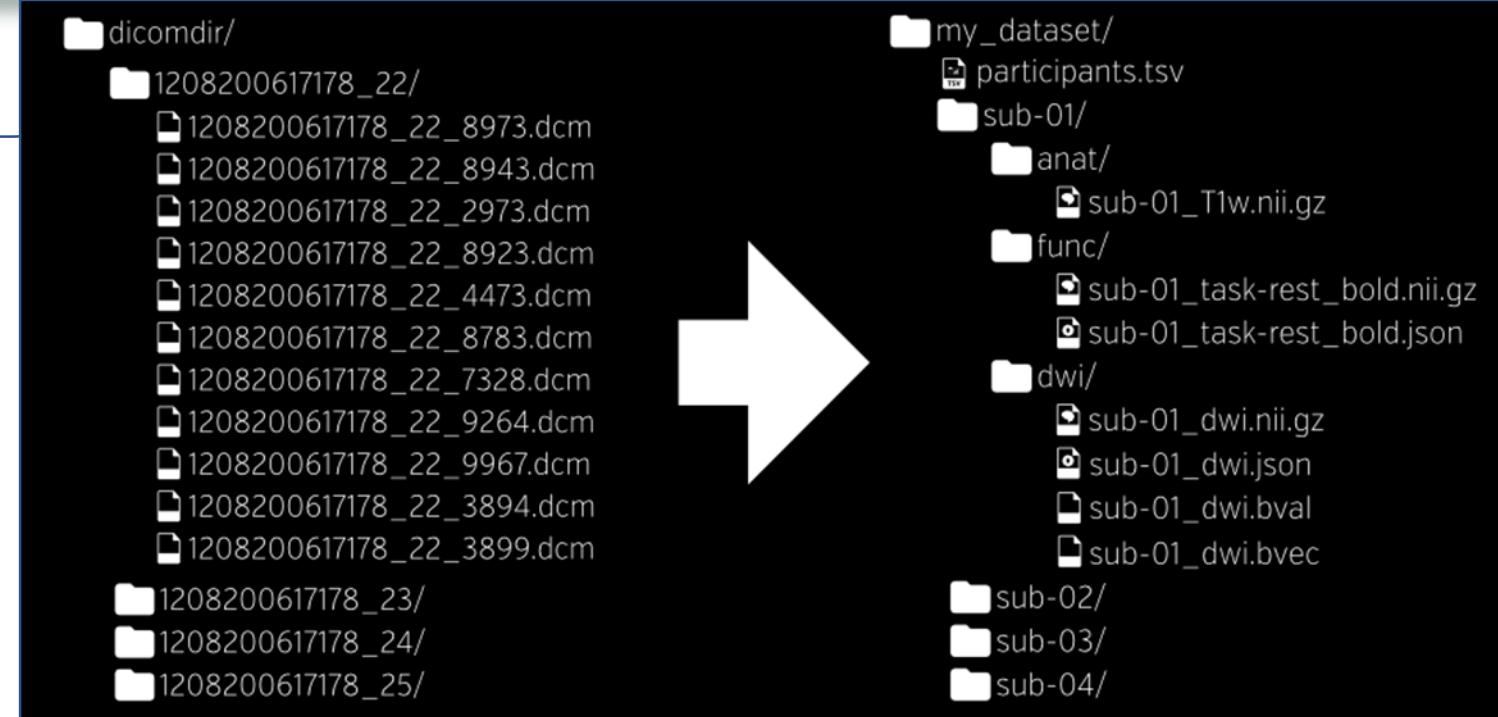
# DICOM → BIDS

Collect the data

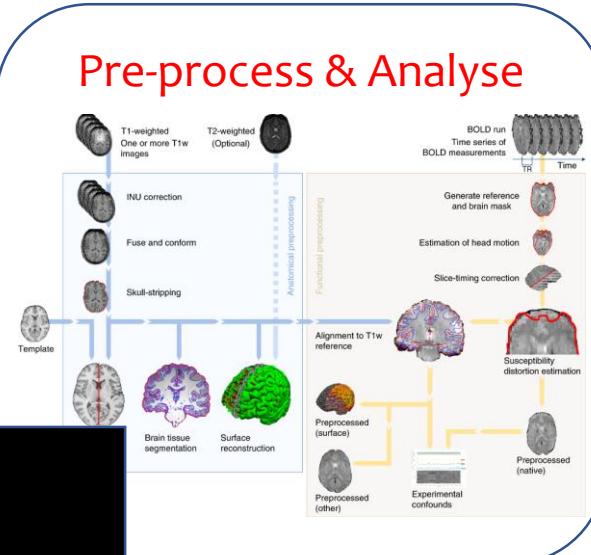


Many BIDS converters available

**HeuDiConv (Heuristic Dicom Conversion)**  
provides sophisticated and flexible creation of BIDS datasets.



Pre-process & Analyse

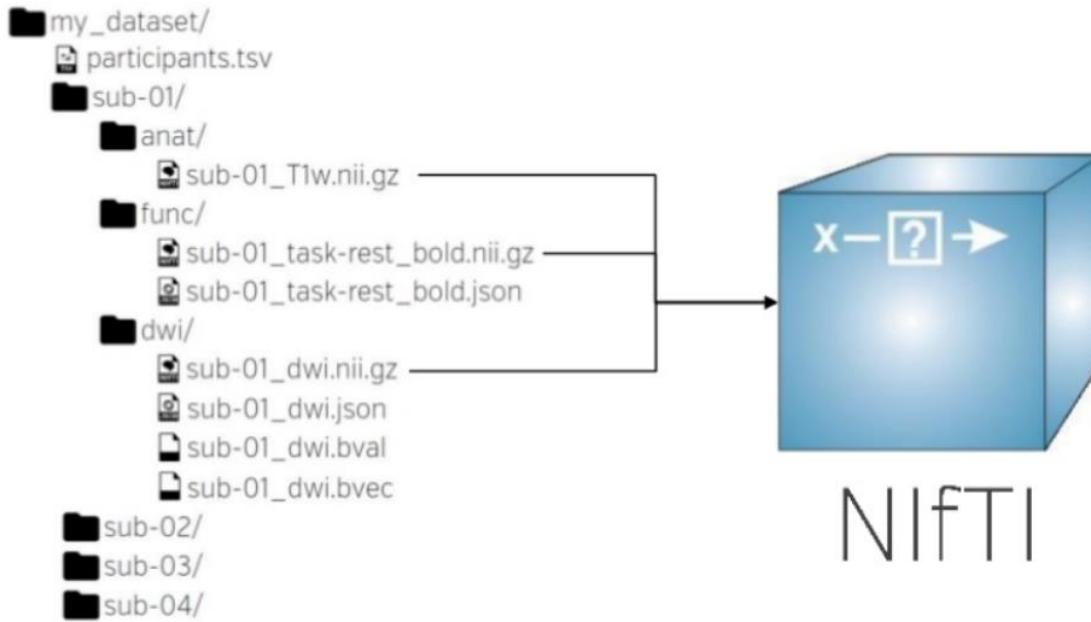


# Environment

Data  
Organise & Manage



Notebook: [nb02\\_Data-Management.ipynb](#)

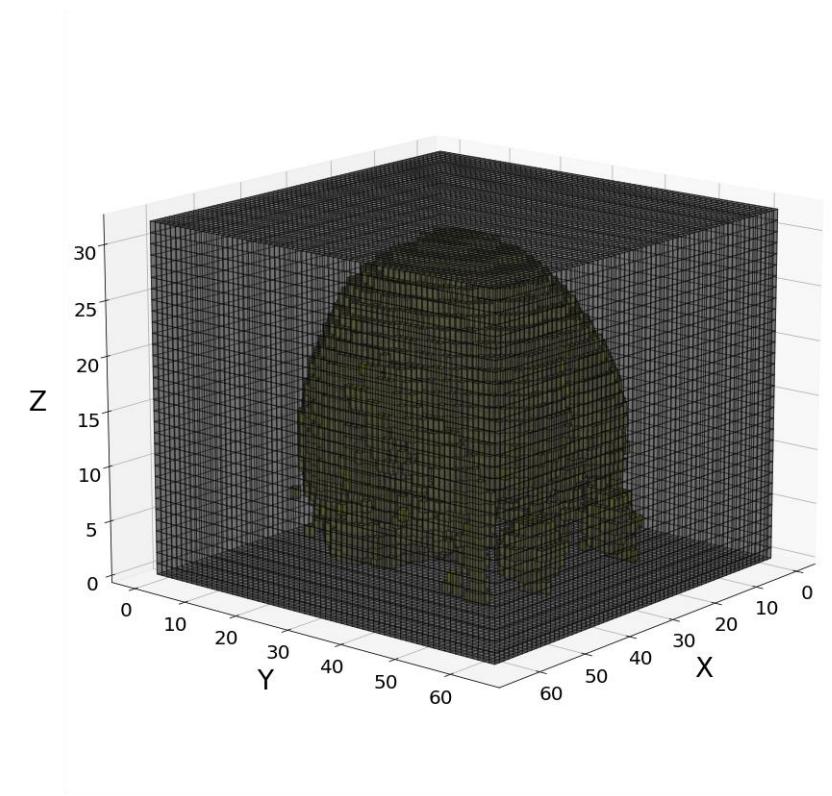
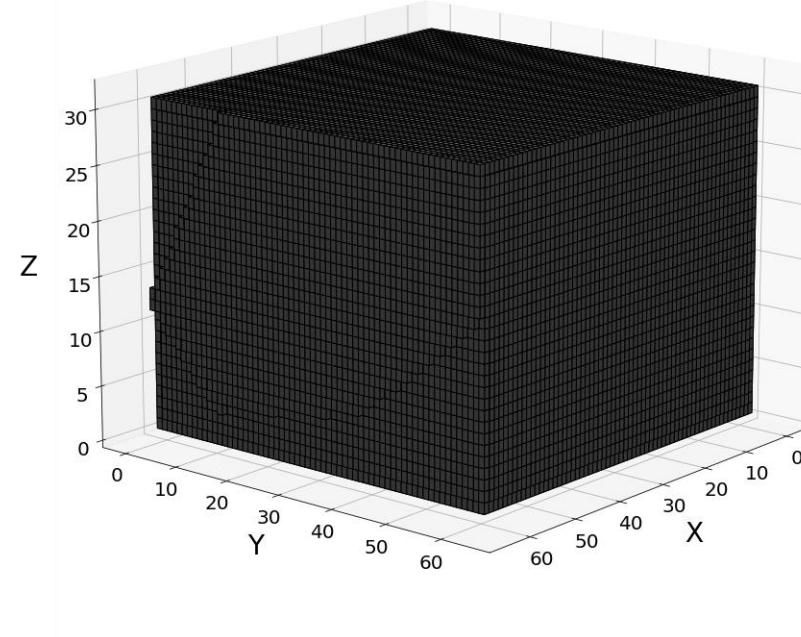


# Imaging data content

# MRI data structure

A 3D or 4D arrays of numbers

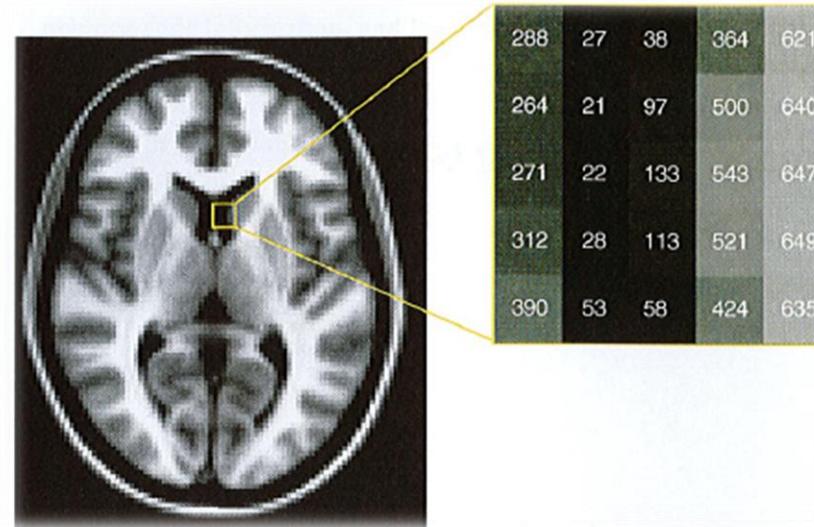
```
([[[ 0., 0., 0., ..., 0., 0., 0.],  
 [ 0., 0., 0., ..., 0., 0., 0.],  
 [ 0., 0., 0., ..., 0., 0., 0.],  
 ...,  
 [ 0., 0., 0., ..., 0., 0., 0.],  
 [ 0., 0., 0., ..., 0., 0., 0.],  
 [ 0., 0., 0., ..., 0., 0., 0.]],  
  
 [[ 0., 0., 0., ..., 0., 0., 0.],  
 [ 0., 25., 23., ..., 23., 32., 0.],  
 [ 0., 28., 21., ..., 25., 25., 0.],  
 ...,  
 [ 0., 26., 24., ..., 40., 20., 0.],  
 [ 0., 44., 28., ..., 30., 21., 0.],  
 [ 0., 0., 0., ..., 0., 0., 0.]],  
  
 [[ 0., 0., 0., ..., 0., 0., 0.],  
 [ 0., 28., 26., ..., 31., 29., 0.],  
 [ 0., 32., 30., ..., 22., 21., 0.],  
 ...,  
 [ 0., 27., 24., ..., 31., 30., 0.],  
 [ 0., 30., 23., ..., 37., 22., 0.],  
 [ 0., 0., 0., ..., 0., 0., 0.]],  
 ...)
```



# MRI data structure

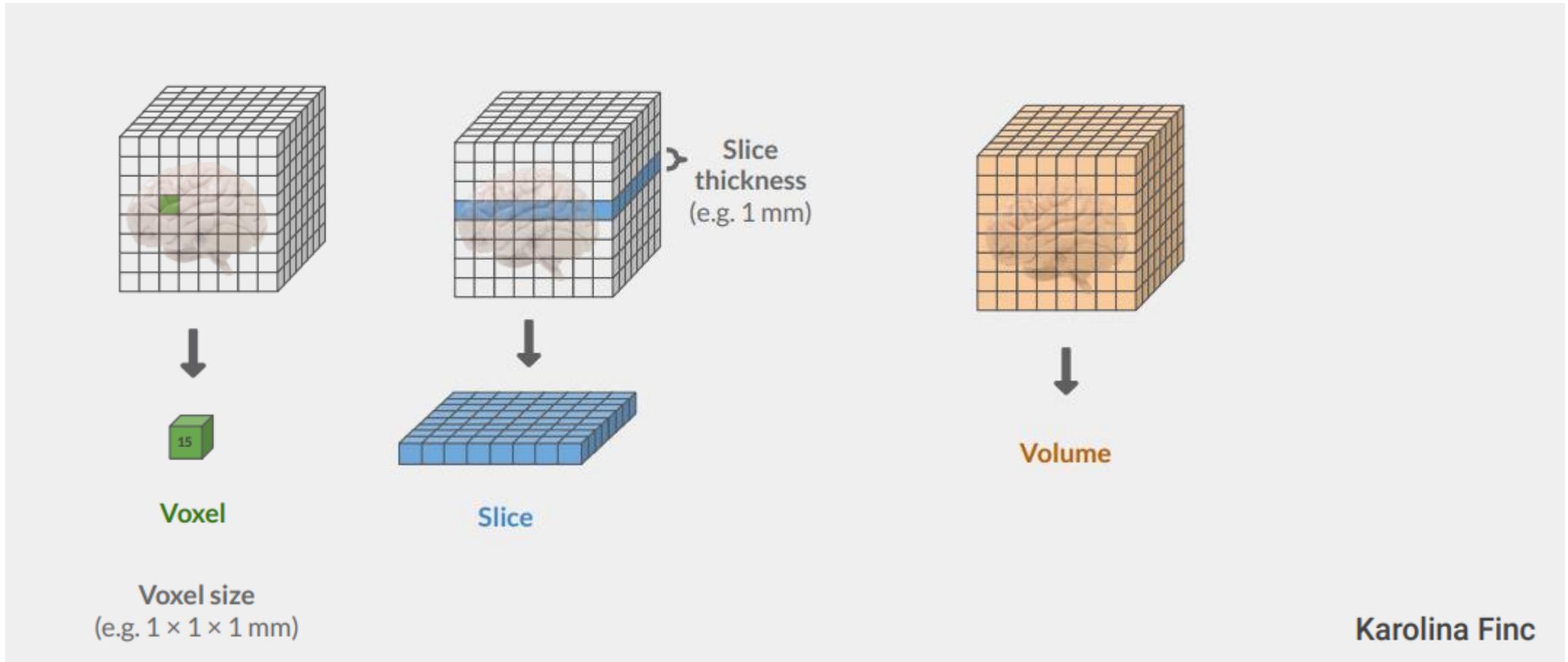
A 3D or 4D arrays of numbers – **intensity values**

```
([[[ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 ...,  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.]],  
  
 [[ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 [ 0.,  25., 23., ..., 23., 32.,  0.],  
 [ 0.,  28., 21., ..., 25., 25.,  0.],  
 ...,  
 [ 0.,  26., 24., ..., 40., 20.,  0.],  
 [ 0.,  44., 28., ..., 30., 21.,  0.],  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.]],  
  
 [[ 0.,  0.,  0., ...,  0.,  0.,  0.],  
 [ 0.,  28., 26., ..., 31., 29.,  0.],  
 [ 0.,  32., 30., ..., 22., 21.,  0.],  
 ...,  
 [ 0.,  27., 24., ..., 31., 30.,  0.],  
 [ 0.,  30., 23., ..., 37., 22.,  0.],  
 [ 0.,  0.,  0., ...,  0.,  0.,  0.]],  
 ...)
```



*Image from Poldrack et al., 2011*

# MRI data structure



# MRI data



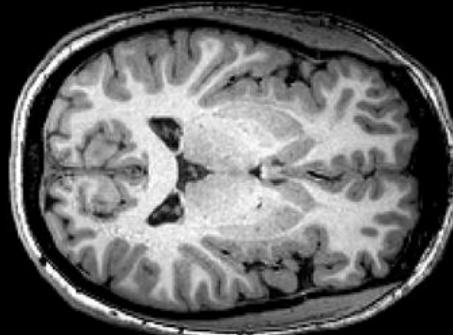
# MRI data

What determines the resolution?

Why can't we acquire the functional images with higher resolution?

high resolution

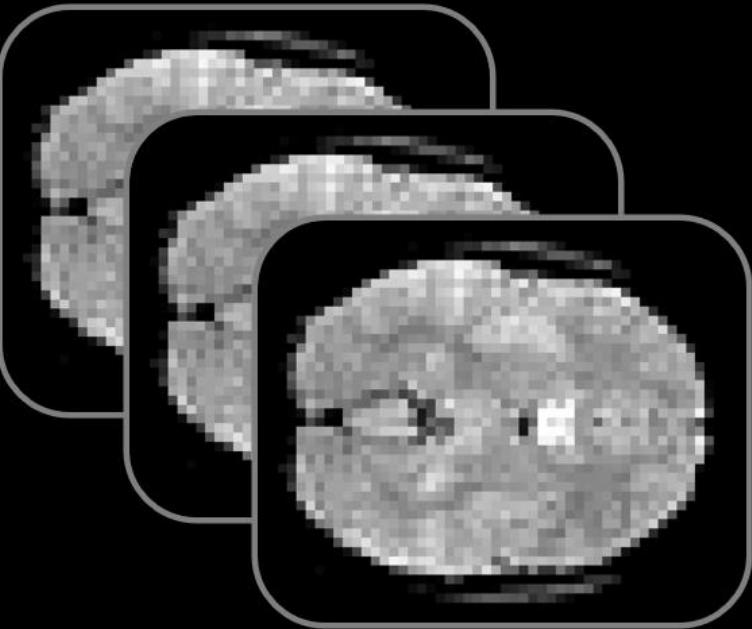
MRI



One 3D volume

fMRI

low resolution

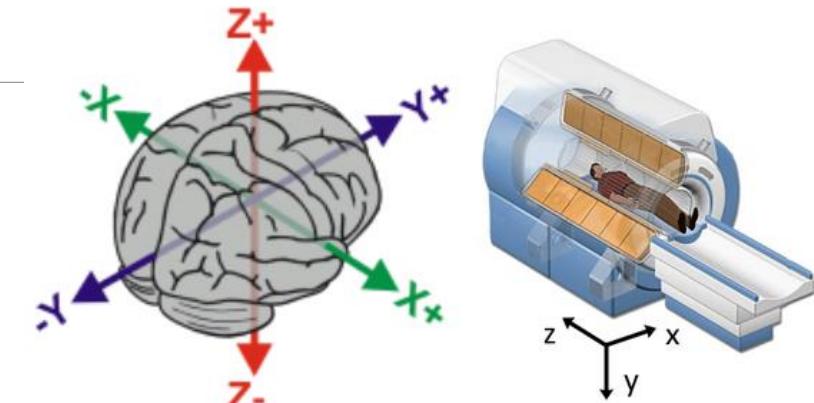
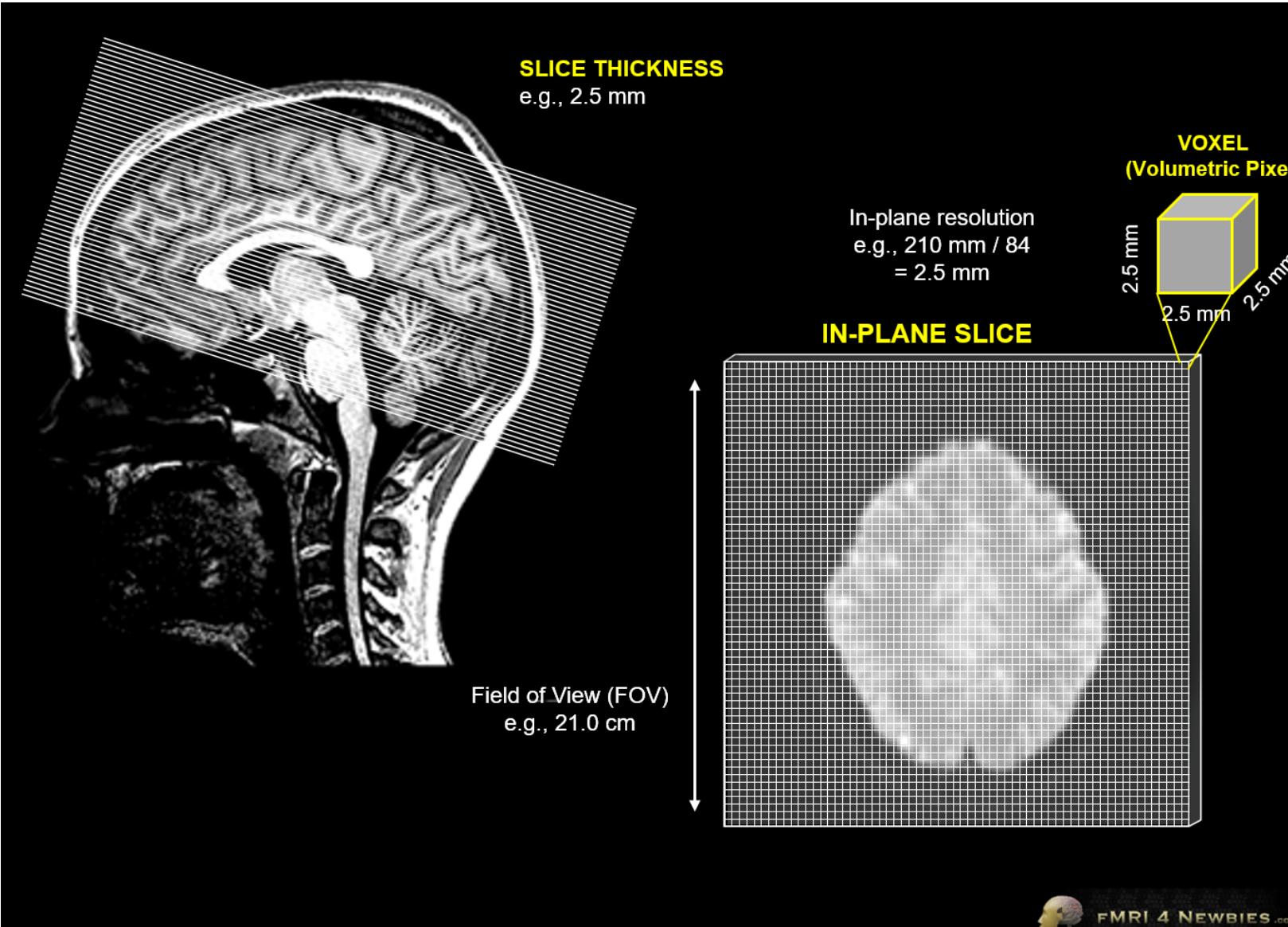


series of 3D volumes (i.e., 4D data)  
(e.g., every 2 sec for 5 mins)



# fMRI data

- Acquired in slices (usually axial; z-axis)

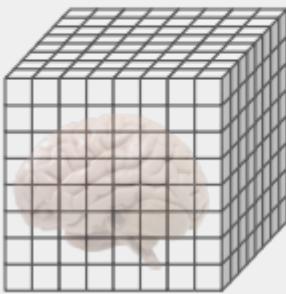


- Temporal resolution (TR), usually 1.5-3s
- Modern sequences allow acquiring multiple slices at the same time (multi-band)
- Typically, 30-50 slices acquired
- More slices = longer TR

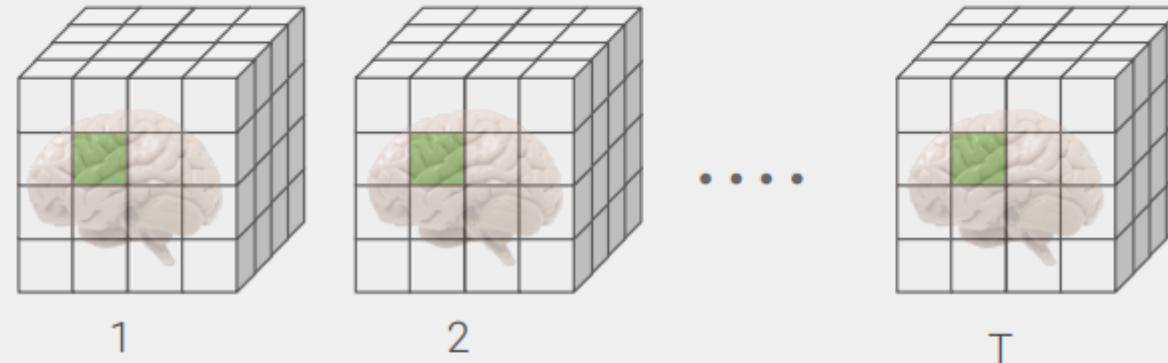


# MRI data structure

Structural data

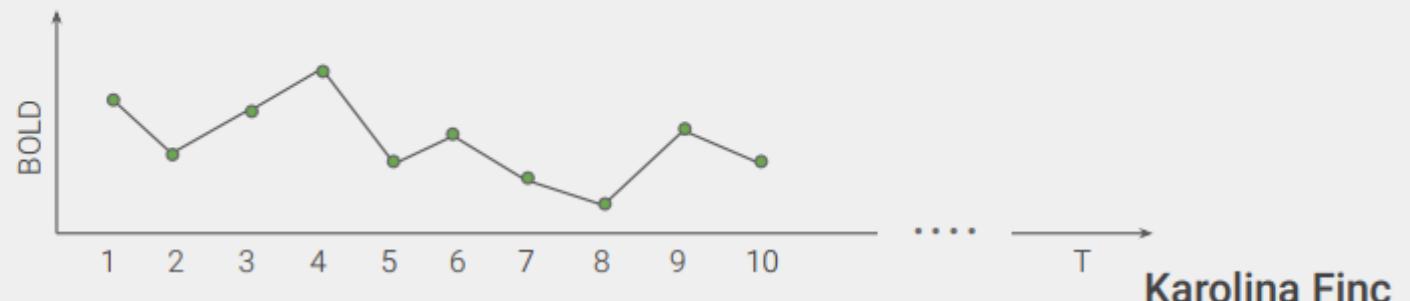


Functional data



**Time series** - is a series of data points listed in time order.

Every voxel has its own time-series.



# Environment

## Data Organise & Manage



Notebook: [nb03\\_Neuroimaging\\_data\\_manipulation.ipynb](#)

✓ Environment

✓ Data  
Organise & Manage

Pre-process

# Outline

---

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