

Data analysis for Neuroimaging - C84DAN

Overview

Denis Schluppeck

What's the plan?

1. Acquire some functional MRI data in a simple, but real experiment
2. Analyze the data with `fsl` ([FMRI B webpage](#))
3. Learn a bit about `UNIX` and version control, in particular `git` and `github`
4. Use `Matlab` to inspect and visualise some data

Timeline

Unit	Topic
1 ★	Introduction, Administtrivia, computers, ...
2	Data acquisition (scanning on 3T)
3	Inspecting & analysing data in FSL
4	Version control (<code>git</code> and <code>github.com</code>)
5	Images in Matlab, display, analyze
6	Timeseries signals in Matlab
7	Reading/writing text, CSV, data files Matlab

What's the assignment?

A short, written report

Summarise the experimental setup, analysis methodology and results. Need to have clearly written abstract (250w), methods, results and discussions.

Submission details

Currently w/ Student Services, *to-be-confirmed*

- turn-it-in submission on moodle page
- deadline: 28 March

What's the assignment (2)

- 250w abstract
- plus a main document (max 1500w)
- references / citations as for standard written work
- max 5 figures¹ illustrating
 - details of the experimental setup
 - analysis methodology
 - results

¹figures can have sub-panels or subplots

! For next time (lab 2)

- sign up for 1 of 4 groups (max 7 people) - moodle
- complete visitor screening form
- we also need 4 volunteers (~30 min in scanner)

Visitor Safety Screening Questionnaire

MR scanning uses strong magnetic fields, for your own safety and the safety of others, it is **very important** that you do not go into the magnet halls with any metal in or on your body or clothing.

NAME	DATE
ORGANIZATION OR ADDRESS	
REASON FOR VISIT	

Please answer the following questions carefully and ask if anything is not clear. All information is held in the strictest confidence.

- Do you have any implants in your body?
e.g. replacement joints, drug pumps Y/N
- Do you have aneurysm clips?
(clips put around blood vessels during surgery) Y/N
- Do you have a pacemaker or artificial heart valve?
(These stop working near MR Scanners) Y/N
- Have you ever had any surgery? Please give brief details*
(*We do not need to know about uncomplicated caesarean delivery, vasectomy or termination of pregnancy) Y/N
- Do you have any foreign bodies in your body (e.g. shrapnel)? Y/N
- Do you wear a hearing aid or cochlear implant? Y/N
- Will you remove all metal including coins, body-piercing jewellery, false-teeth, hearing aids etc before entering the magnet hall? (lockers available by the changing rooms) Y/N

I have read, understood, and answered all questions

Signature:

Verified by:

SPMIC Staff Signature:

Setting up computers, logins

1. Each user (at a particular machine) needs to make sure that `Terminal/shell` is set up correctly by copying a set-up file the first time they use that computer.

```
# copy across new version of .bash_profile
cd ~ # make sure we are in ${HOMEDIR}
cp /Volumes/practicals/ds1/.bash_profile ~/
# restart shell
```


Has setup worked? Reality check.

1. If you see `[ran custom .bash_profile]` in Terminal ✓
2. Also: look at some existing anatomies with `fslview` ✓

```
which fsl # see anything?
```

```
fslview & # File -> Open Standard -> Pick 1st or 2nd
```

Setting up computers, logins

Cheat: Double-click `Set up My Machine` icon



If you have to do this again on another machine, you can use this shortcut. It's located in `/Volumes/practicals/ds1/`

FSL analysis

- get data from sessions `S001` to `S004` into a common folder `data`
- make folders, copy files by "drag & drop"
- point & click version (like some of you have already done)
- digging into the details of how this is implemented
- inspecting analysis output, intermediate files, ...

```
cd ~/data/S001/  # for example  
# run FSL analysis
```

Some UNIX

- only basics are needed for running FSL analysis
- lots of functionality is available through point-and-click
- but command line is helpful for organising (any) research data
- more complex analysis, e.g. `freesurfer` , require some working knowledge

```
# navigate file system  
# cd, ls, pwd, which, ...  
  
# some powerful commands for organising your data  
# cp, rm, touch, mkdir, rmdir  
  
# some stuff to show of how powerful  
# grep, "lists", "wildcards (*, ., ?)"  
# "regular expressions"
```

Version control `git`

- 30min [lecture on principles of version control](#) (`git`)
- start using your (free) `github.com` id by working on a simple project
- make your first modifications to a local copy of code and get it into a repo.

```
mkdir test && cd test # what does this do?
git init
# [[ create, edit a file, say my_first.md ]]
git add my_first.md # add it to "staging area"
git commit # enter commit message
# - OR -
git commit -m 'adds first version of file'
git log
```

Version control (v2.0) 😊

Everyone should sign up for a free `github` account, so we can work together on this from session 4 onwards: <https://github.com/join>

- it's free and useful
- we'll want to play with this in lab #4
- once you have an username, go to our github classroom at <https://classroom.github.com/a/I5Z7qarj>

matlab - reading images (1)

- we'll learn how to read imaging data into matlab (nifti files)
- functions provided by the [mrTools](#) [toolbox](#) for Matlab

```
help mlrImageReadNifti
```

```
% read in some data
```

```
data = mlrImageReadNifti('file_from_scanner.nii');
```

matlab - reading images (2)

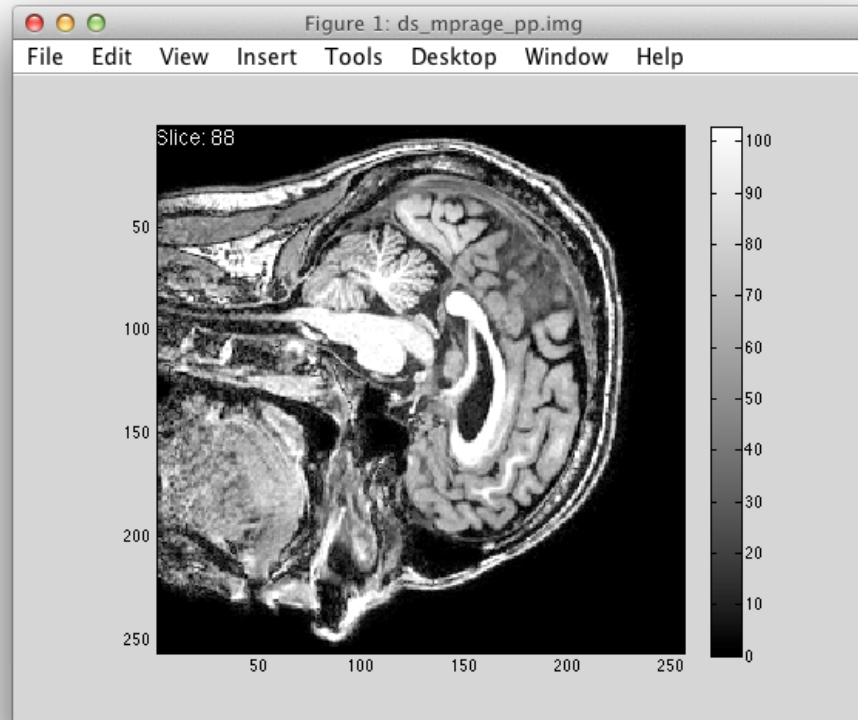
- revisit indexing of arrays, "slicing", etc.

```
data(12, 24, :, 1) % what is this?  
data(:, 24, 24, 1) % ... and this?  
data(34, 44, 12, :) % ... or that?
```

- build a `returnSlice()` function, to complete a simple imageviewer:

```
% function signature  
s = returnSlice(array, sliceNum, orientation);
```

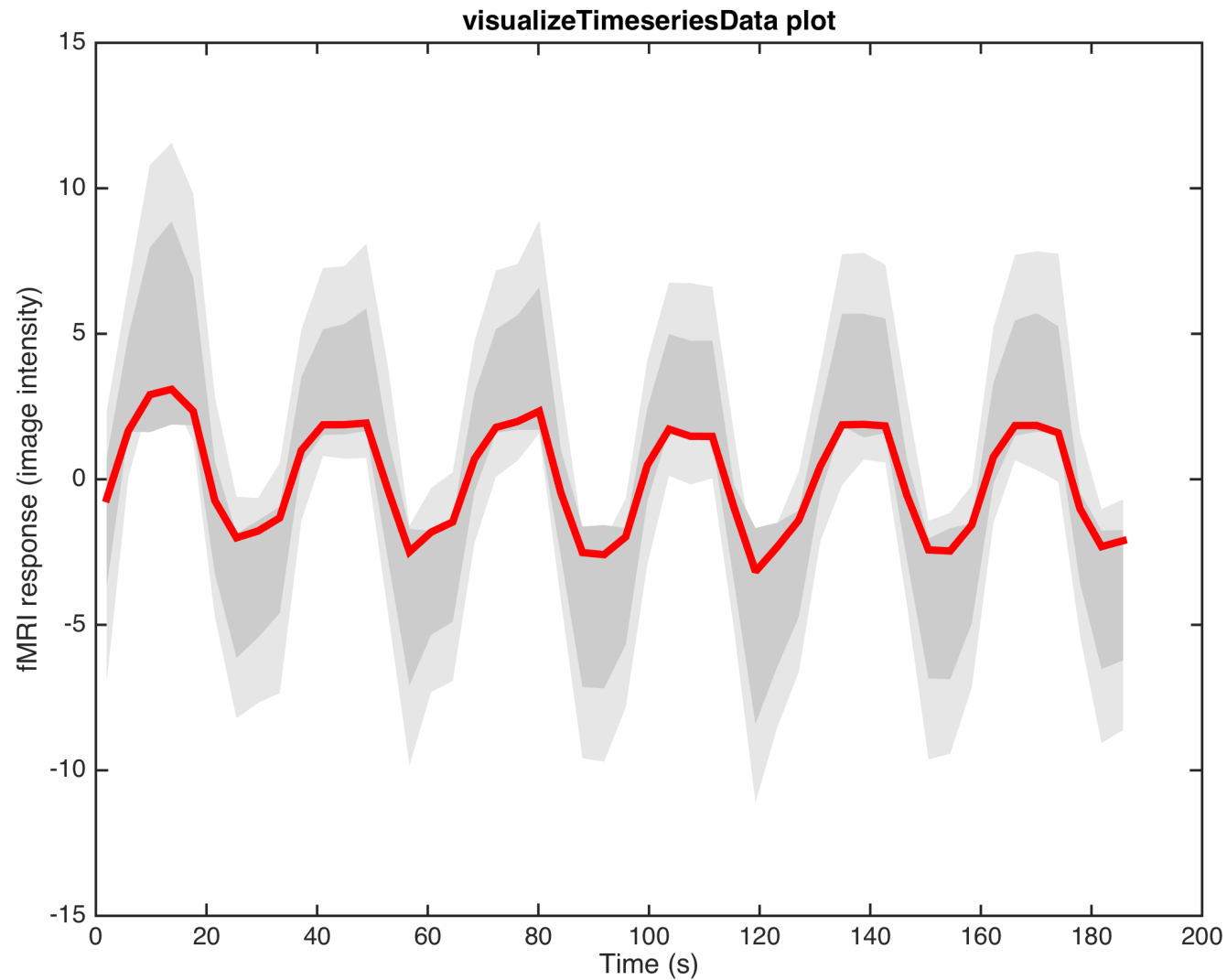

sliceview()



Press the following buttons to:

up/down	change slice
o/O	change orientation
c/C	change cursor
q/	Esc quit

matlab - timeseries and subplots



matlab - text / csv / other data

- think about data formats / interop with other analysis & tools (R , python , ... even UNIX tools). Sometimes a text file is best!

```
% read / write delimited files
```

```
dlmread()
```

```
dlmwrite()
```

```
% basic CSV support (no header lines!)
```

```
csvread()
```

```
csvwrite()
```

```
% read in a simple CSV file, skipping first row (r=0)
```

```
% csvread(file, R, C) % row R, column C (starting at 0!)
```

```
d = csvread('timecourse.csv', 1, 0)
```

Wrap-up (Lab 7)

- recap what have we covered in the last 7 weeks?
- where to go to from here (unleash your inner coding 🐱)
- try to approach each new problem, project with lots of repetition (analysis, writing, coding, ...):
 - there must be a better way!
 - what's the smallest unit that gets repeated all the time?
 - can I use `bash/unix`, `matlab` or another tool to automate?
- just try things out - you'll learn tons in the process

Notes

Small `awk` program for adding a counter `n` and time `t` and turn one column txt file into csv file:

```
awk 'NF      {print NR-1 ", " (NR-1)*1.5 ", " $1}' \  
    timecourse.txt > timecourse.csv
```

- with a headerline (matlab's `csvread()` doesn't like!)

```
awk 'BEGIN {print "n, t, response"}  
    NF      {print NR-1 ", " (NR-1)*1.5 ", " $1}' \  
    timecourse.txt > timecourse.csv
```