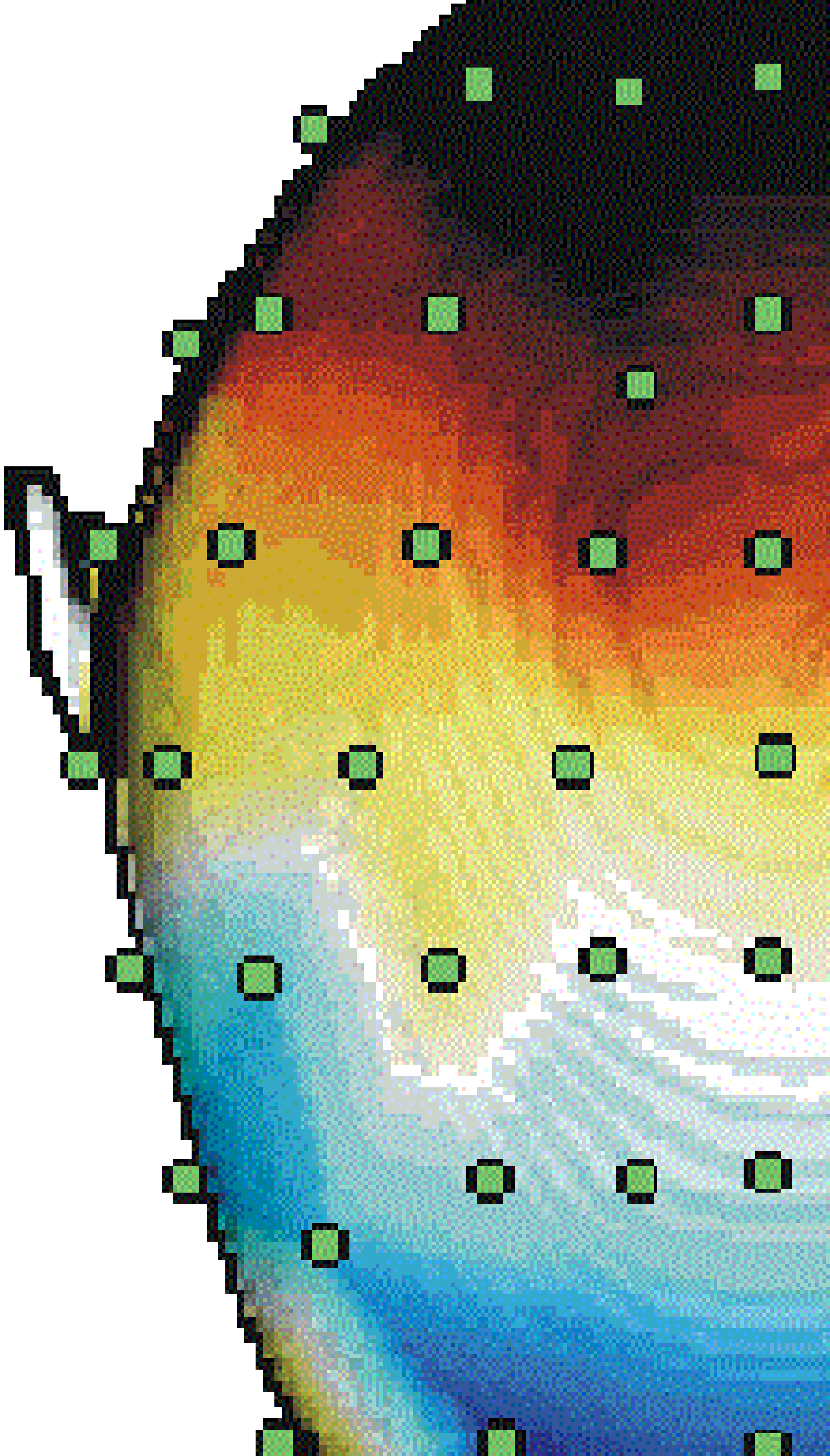
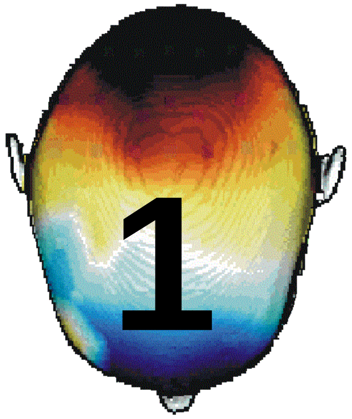
Session 1:

EEGLAB (in MATLAB)



Daniel Bennett
UniSA, March 2016



Introduction

& set-up of computers

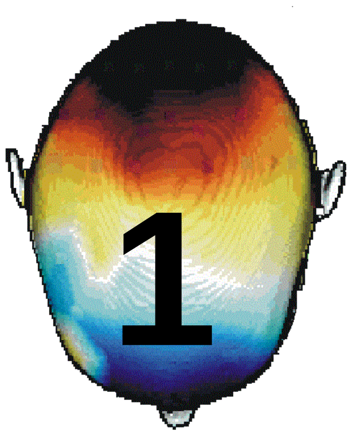
EEGLAB

Program overview

EEGLAB

A whistle-stop tour

Slides available for download at http://bit.ly/21ygstz



Introduction

& set-up of computers

EEGLAB

Program overview

EEGLAB

A whistle-stop tour

Slides available for download at http://bit.ly/21ygstz

Introductions (briefly)

Daniel Bennett
PhD Candidate, University of Melbourne

I research decision making under uncertainty, using a combination of neuroimaging methods (EEG, fMRI) and computational cognitive modelling.

Obligatory embarassing EEG headshot:

Slides available for download at http://bit.ly/21ygstz

Introduction to EEGLAB

EEGLAB is an open-source toolbox for preprocessing and analysis of EEG data.

- MATLAB-based
- Freely available (though MATLAB is not)

This 1-hour session will cover the basics of EEGLAB.

- Features and limitations
- Pointers toward some useful references

No MATLAB knowledge required.

- We will largely use the GUI, not the command line
- Information about command-line access to data will also be provided for those so inclined.

Slides available for download at http://bit.ly/21ygstz

Setting Up Computers

Download the eeglab folder from this link: bit.ly/1WVrClt

Unzip it into your My Documents > MATLAB directory

In MATLAB, click ‘Set Path’ > ‘Add folder and subfolders’

If prompted to save a different path file (‘pathdef.m’), save this to your MATLAB folder too.

Introduction
& set-up of computers

EEGLAB
Program overview

EEGLAB
A whistle-stop tour

Slides available for download at http://bit.ly/21ygstz

Introduction to EEGLAB

- EEGLAB is an open-source toolbox for preprocessing and analysis of EEG data.
  - MATLAB-based
  - Freely available (though MATLAB is not)

- This 1-hour session will cover the basics of EEGLAB.
  - Features and limitations
  - Pointers toward some useful references

- No MATLAB knowledge required.
  - We will largely use the GUI, not the command line
  - Information about command-line access to data will also be provided for those so inclined.

Slides available for download at http://bit.ly/21ygstz

History of EEGLAB

1997-2001: Initially developed as an Independent Components Analysis toolbox at Salk Institute, La Jolla, USA.

2001-present: Maintained by the Schwartz Centre for Computational Neuroscience, UCSD.

Website: http://sccn.ucsd.edu/eeglab/

Citation: Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*,*134*(1), 9-21.

Version: - Current version is v.13 (MATLAB 2014a and later)
- Old version is v. 12 (pre-2014a)

Pros and Cons

Pro:

- Access to single-trial data
- Open-source and non-proprietary code
- Flexible batch scripting
- Extensive documentation
- Surprisingly many inbuilt features
- Ability to create new modules
- Data visualisation

Cons:

- Memory requirements
- Unlovely GUI
- Code stability
- User support
- Data visualisation

For us, the most important considerations were to have control over every step of the analysis, and to have theory-neutral and flexible access to data.

Introduction
& set-up of computers

EEGLAB
Program overview

EEGLAB
A whistle-stop tour

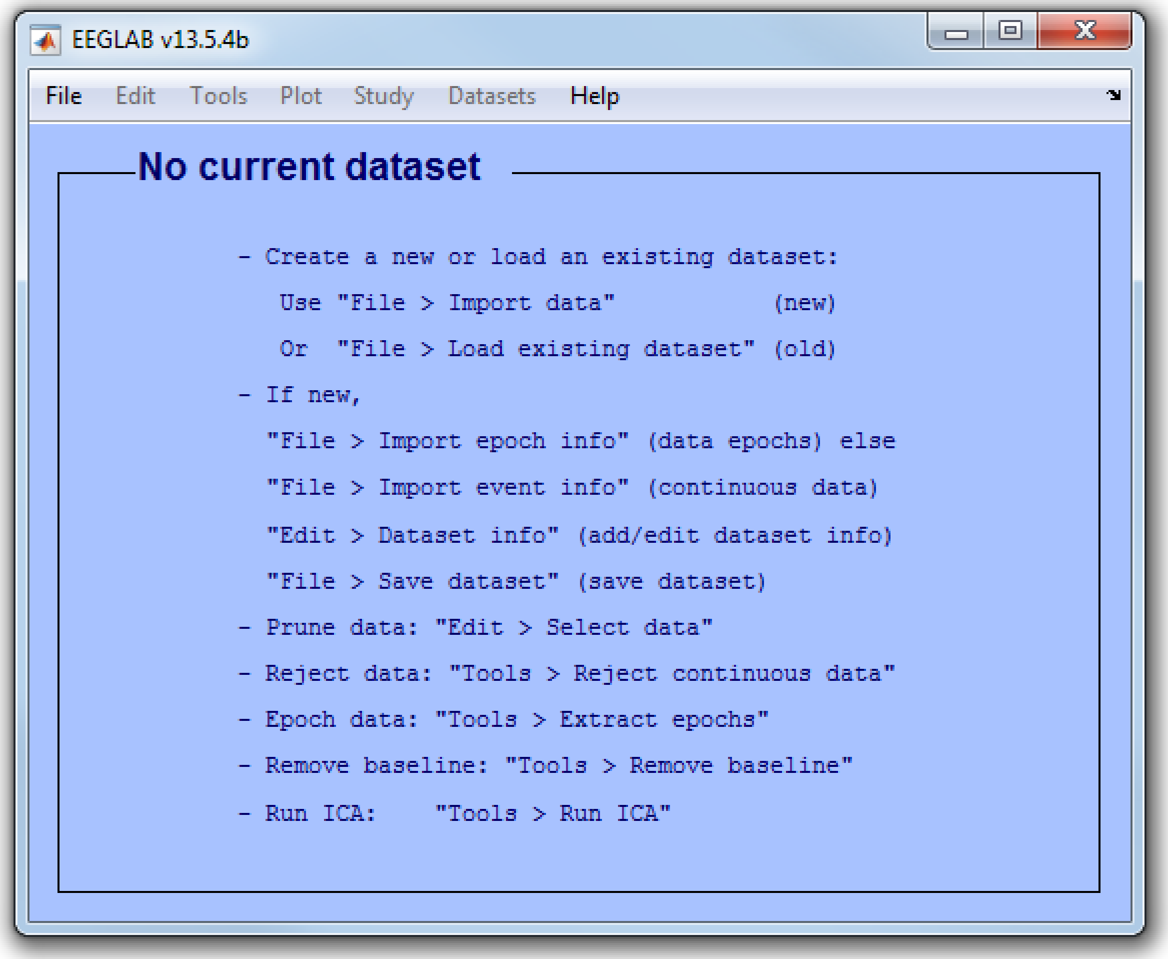
Slides available for download at http://bit.ly/21ygstz

Getting started

1. Open MATLAB

2. Type `eeglab` at the command line

3. Press enter.



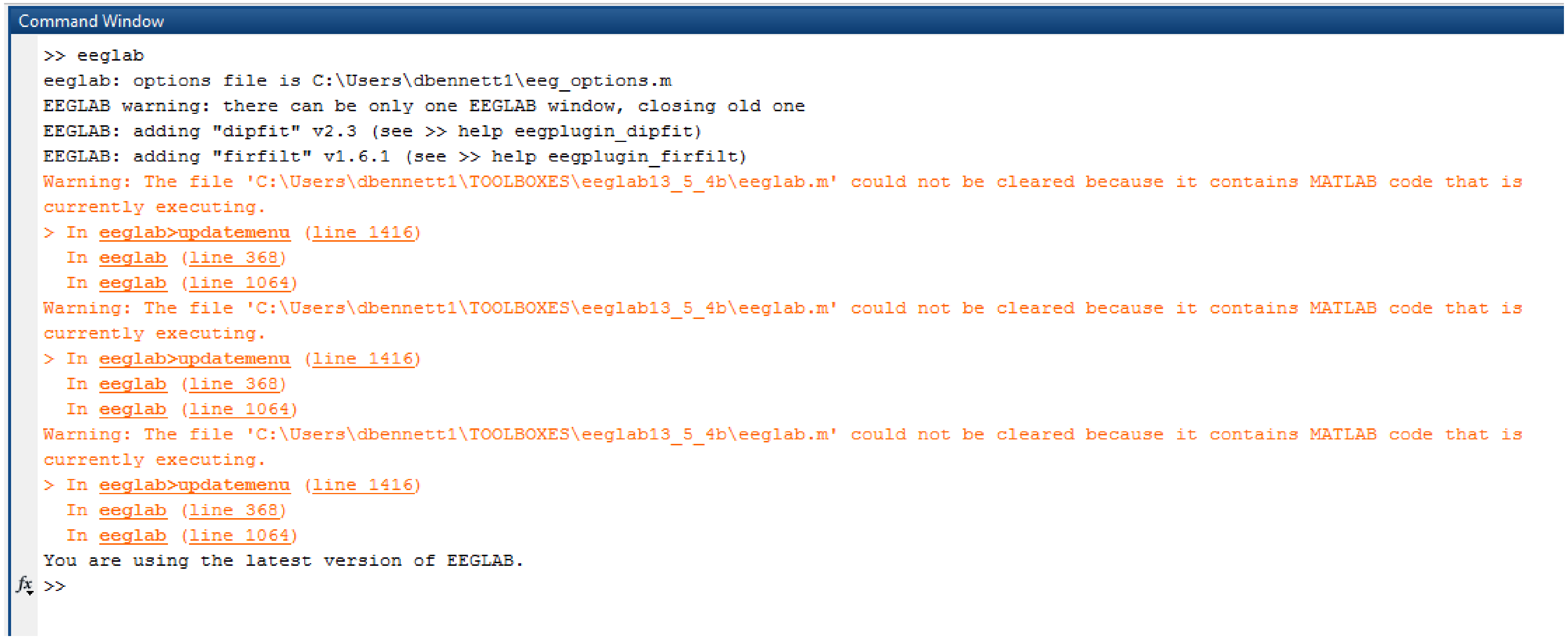
No current dataset

```
- Create a new or load an existing dataset:
  Use "File > Import data"           (new)
  Or  "File > Load existing dataset" (old)
- If new,
  "File > Import epoch info" (data epochs) else
  "File > Import event info" (continuous data)
  "Edit > Dataset info" (add/edit dataset info)
  "File > Save dataset" (save dataset)
- Prune data: "Edit > Select data"
- Reject data: "Tools > Reject continuous data"
- Epoch data: "Tools > Extract epochs"
- Remove baseline: "Tools > Remove baseline"
- Run ICA:    "Tools > Run ICA"
```

Slides available for download at http://bit.ly/21ygstz

Getting started

These warnings are distracting, and kind of ugly…



```
>> eeglab
eeglab: options file is C:\Users\dbennett1\eeg_options.m
EEGLAB warning: there can be only one EEGLAB window, closing old one
EEGLAB: adding "dipfit" v2.3 (see >> help eegplugin_dipfit)
EEGLAB: adding "firfilt" v1.6.1 (see >> help eegplugin_firfilt)
Warning: The file 'C:\Users\dbennett1\TOOLBOXES\eeglab13_5_4b\eeglab.m' could not be cleared because it contains MATLAB code that is
currently executing.
> In eeglab>updatemenu (line 1416)
  In eeglab (line 368)
  In eeglab (line 1064)
Warning: The file 'C:\Users\dbennett1\TOOLBOXES\eeglab13_5_4b\eeglab.m' could not be cleared because it contains MATLAB code that is
currently executing.
> In eeglab>updatemenu (line 1416)
  In eeglab (line 368)
  In eeglab (line 1064)
Warning: The file 'C:\Users\dbennett1\TOOLBOXES\eeglab13_5_4b\eeglab.m' could not be cleared because it contains MATLAB code that is
currently executing.
> In eeglab>updatemenu (line 1416)
  In eeglab (line 368)
  In eeglab (line 1064)
You are using the latest version of EEGLAB.
>>
```

1. Enter the following at the command line: `edit eeglab`

2. Insert a % symbol at the beginning of line 1398: `%clear functions;`

3. Resave the file.

Slides available for download at http://bit.ly/21ygstz

Importing Data

1. File > Load existing dataset > ‘eeglab_data.set’.

NB: The file format for existing eeglab datasets is **.set**. If we had wanted to import directly from the raw datafile, we could have used File > Import data > Using EEGLAB functions and plugins.

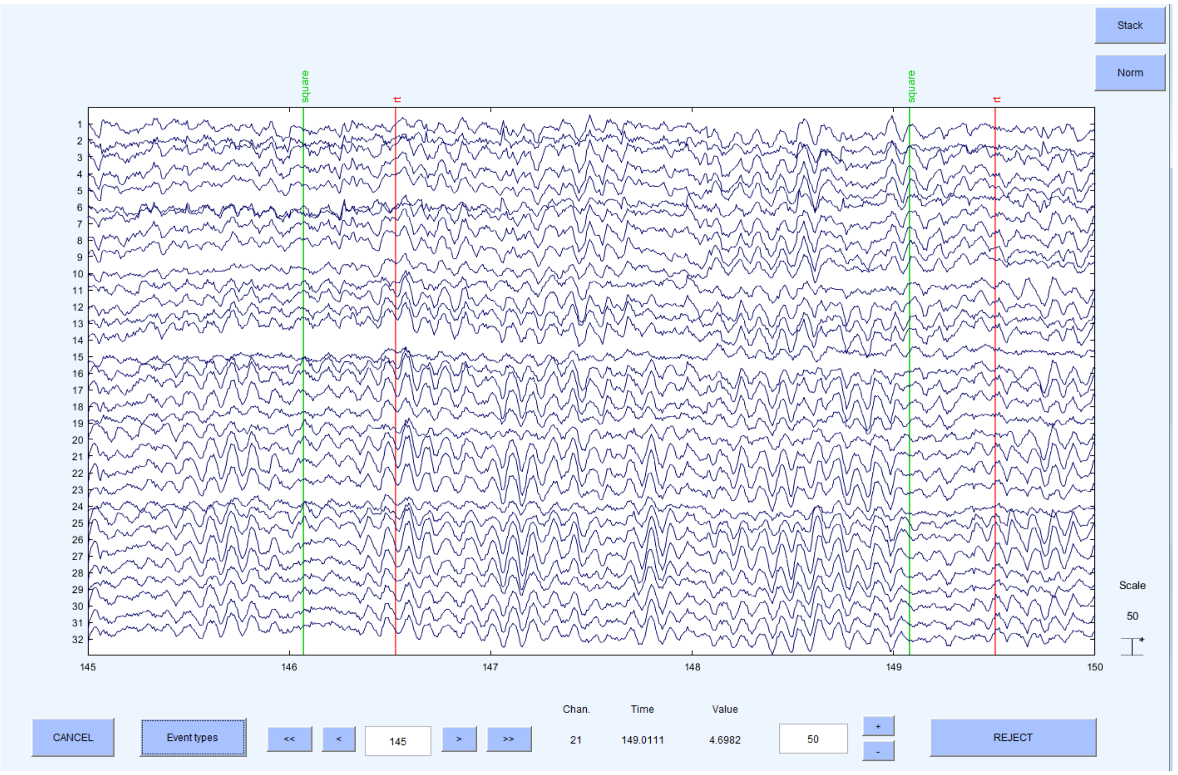
If you look in the data folder, you will notice that there are two datafiles called eeglab_data. The .set file contains metadata, and the .fdt file contains the eeg recordings.

Visualising Data

Plot > Channel data (scroll)

Things to note:

1. Timestamp
2. Vertical scale
3. Events
4. Stacking/unstacking
5. Rejecting data



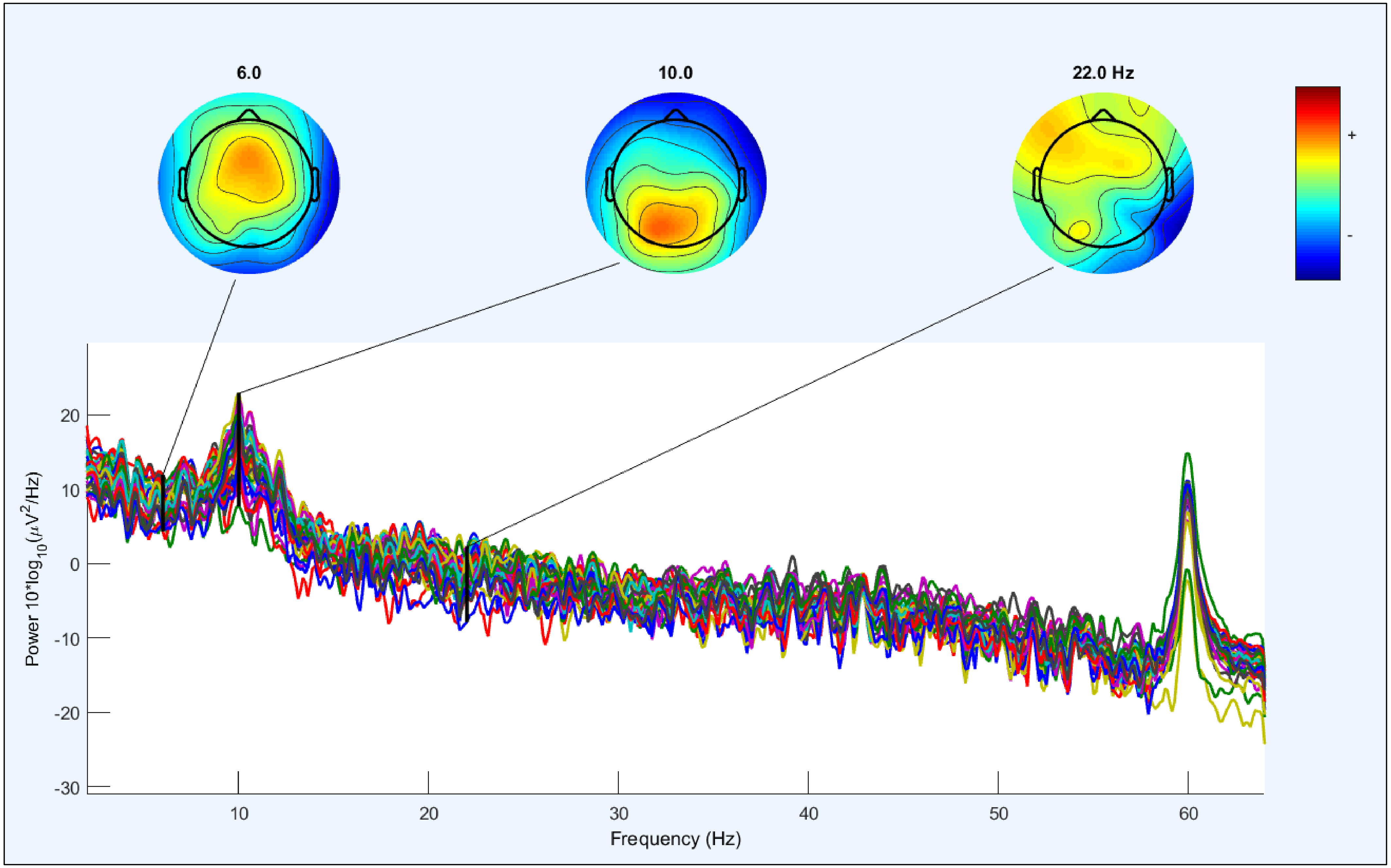
Edit > Channel locations…

Slides available for download at http://bit.ly/21ygstz

Visualising Data

Plot > Channel spectra and maps

Plotting frequency range: [2 64]



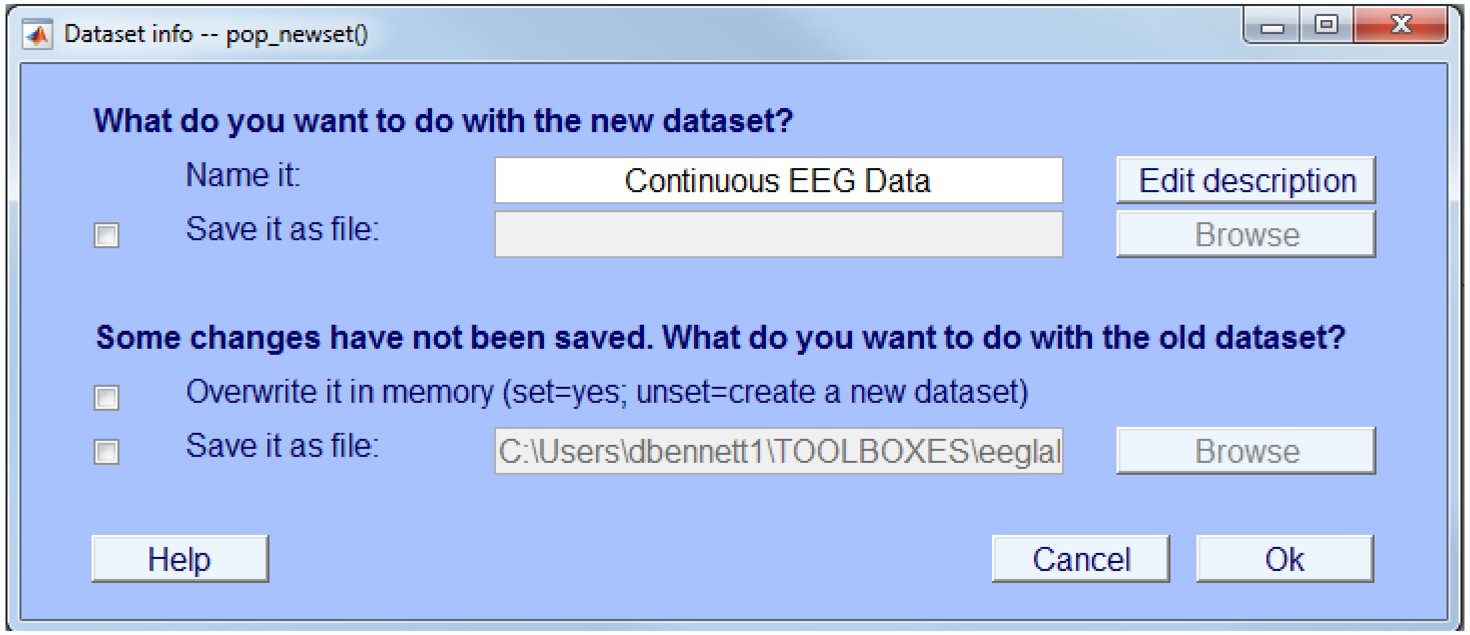
Visualising Data

Tools > Filter the data > Basic FIR filter

Lower edge = 58

Higher edge = 62

‘Notch filter the data…’ = ticked



Dataset info -- pop_newset()

What do you want to do with the new dataset?

Name it: Continuous EEG Data

□ Save it as file:

Some changes have not been saved. What do you want to do with the old dataset?

□ Overwrite it in memory (set=yes; unset=create a new dataset)

□ Save it as file: C:\Users\dbennett1\TOOLBOXES\eeglal

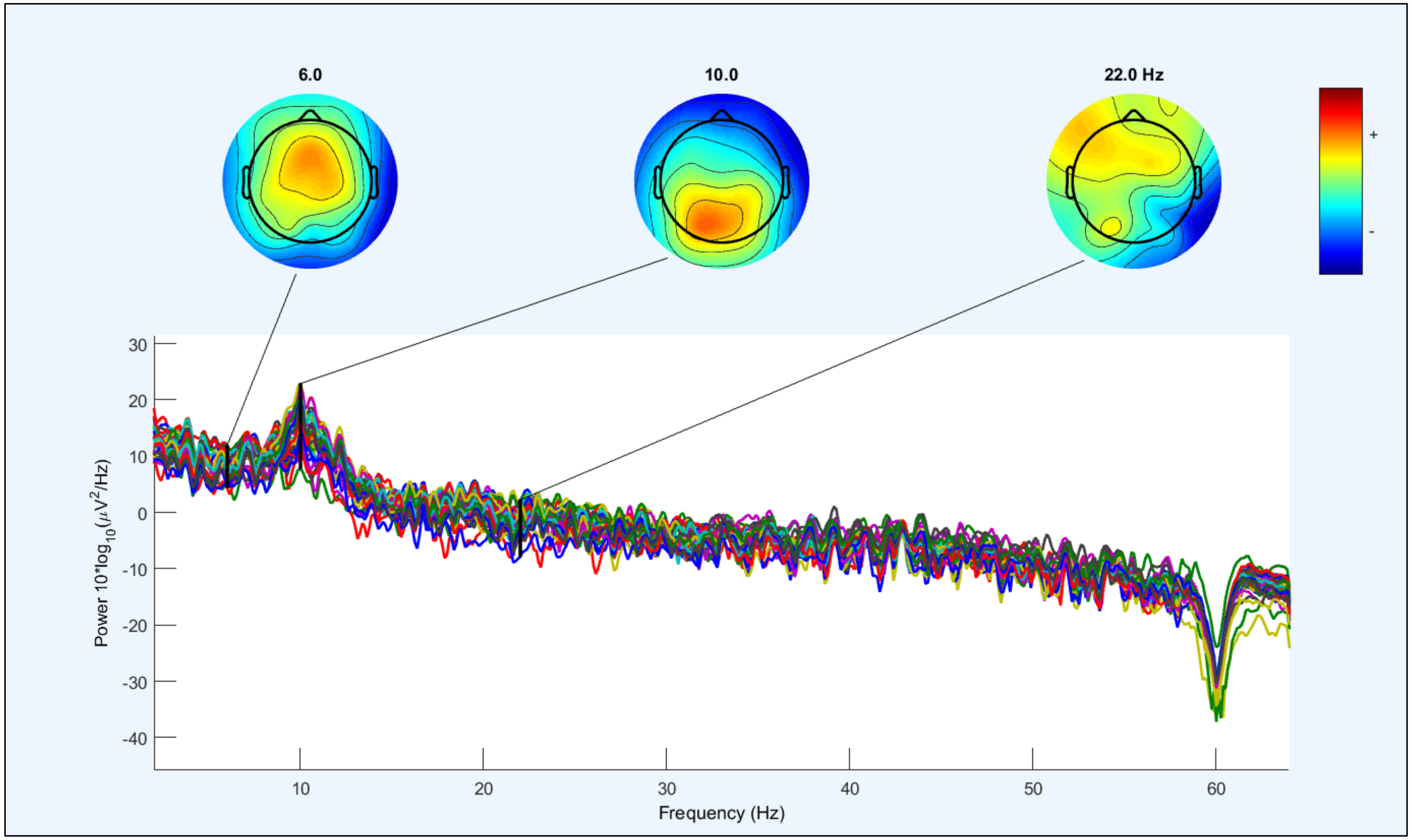
Visualising Data

Tools > Filter the data > Basic FIR filter

Lower edge = 58

Higher edge = 62

‘Notch filter the data…’ = ticked



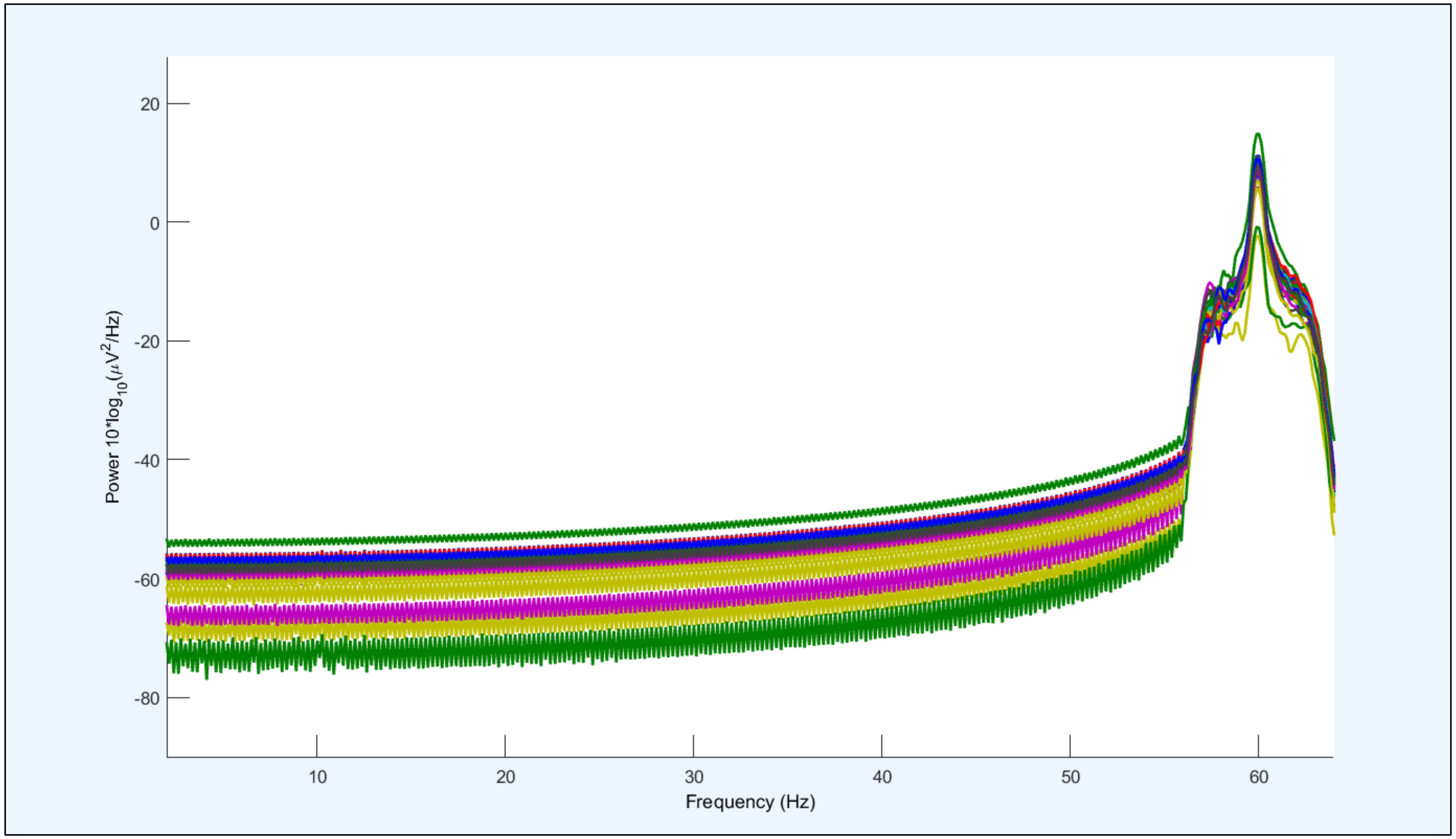
Visualising Data

Tools > Filter the data > Basic FIR filter

Lower edge = 58

Higher edge = 62

‘Notch filter the data…’ = unticked



Accessing Data

Data can be easily accessed from the command line, in the EEG structure:

```
>> EEG
>> EEG.data
```

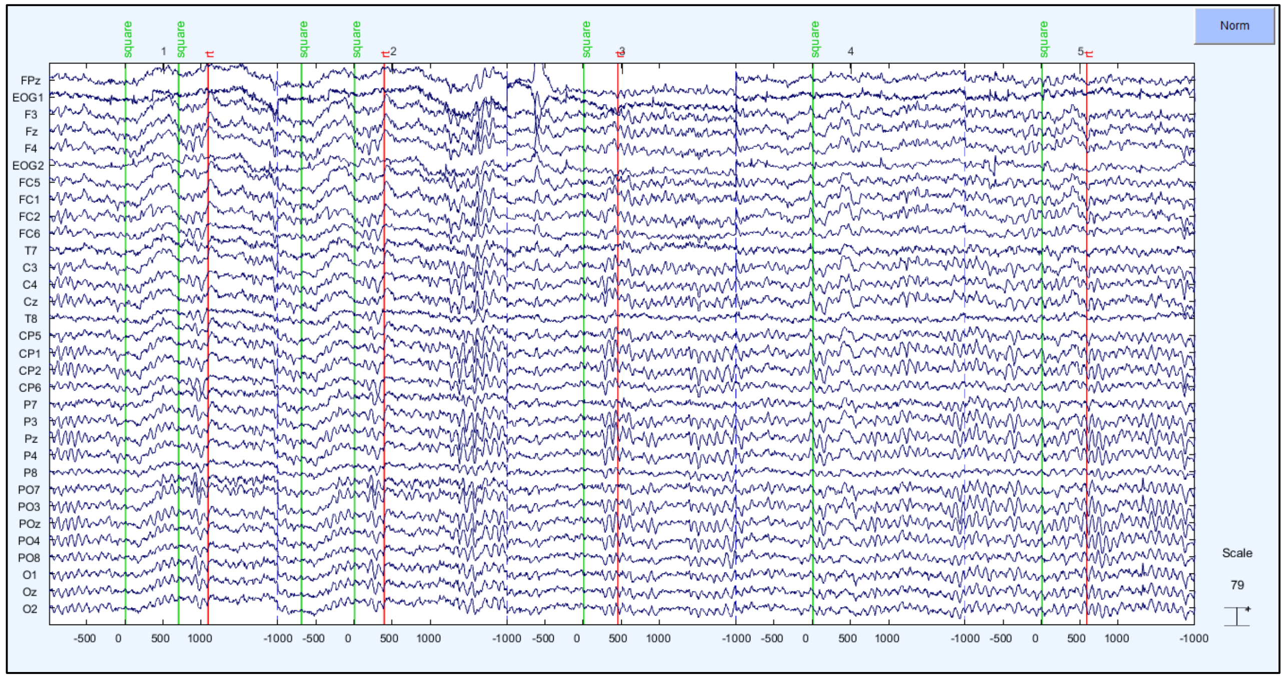
Continuous data like in this file are stored in an A x B matrix, where
A is the number of electrodes,
B is the number of data points

```
>> EEG.data(5,1:128)

>> plot(EEG.data(5,1:128))

>> EEG.data(5,:) = 0
```

Analysing Data

We will now open up a file which has already been ‘epoched’ (divided into segments for analysis)

1. File > Clear dataset(s)
2. File > Load existing dataset > ‘eeglab_data_epochs_ica.set’.
3. Plot > Channel data (scroll)



Note that data is now discrete, not continuous (try clicking on a segment).

Note also that time 0 in a segment always corresponds to a green line.

Analysing Data

Unlike continuous data, epoched data are stored in an A x B x C matrix, where

A is the number of electrodes,
B is the number of data points per epoch
C is the number of epochs

As a result, we can access single-trial data from the command line as follows:

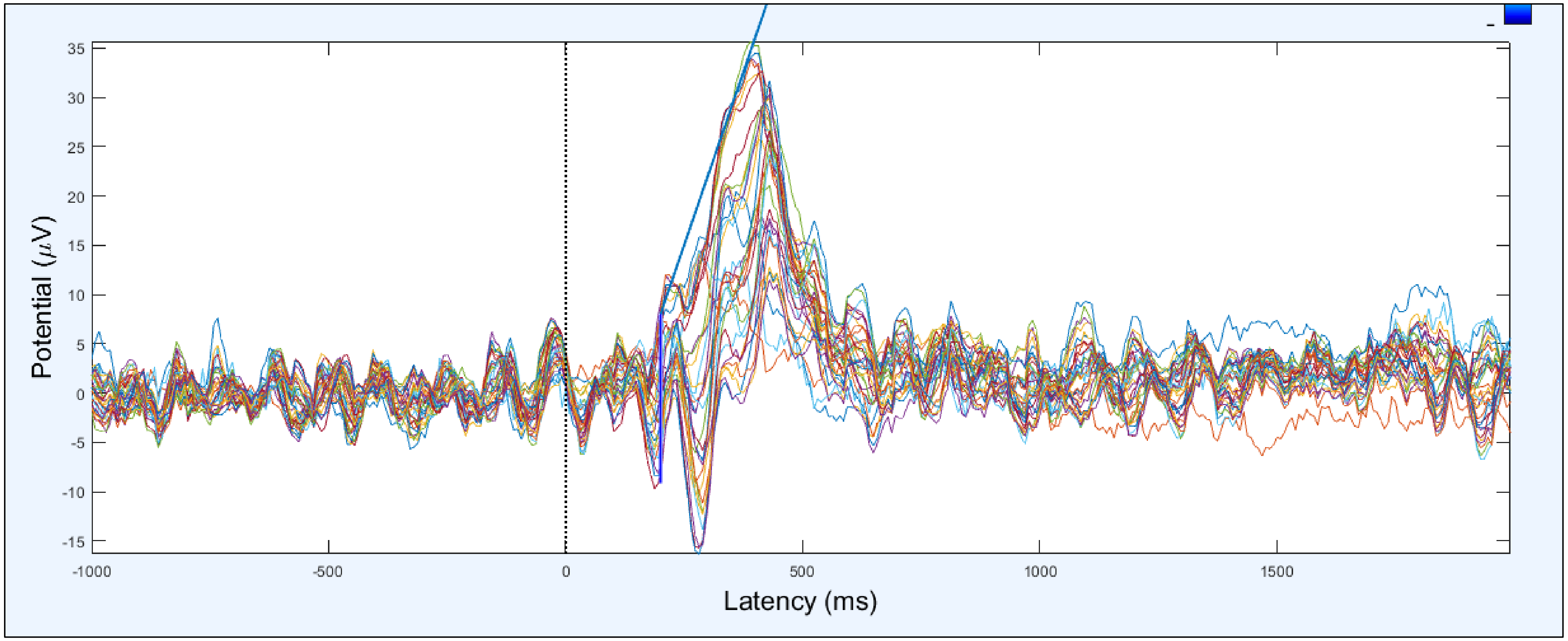
```
>> EEG.data(:,:,1)
```

And, if we so desire, we can access information about all the events that happen within an epoch as follows:

```
>> EEG.epoch(1)
```

Plotting Averaged Data

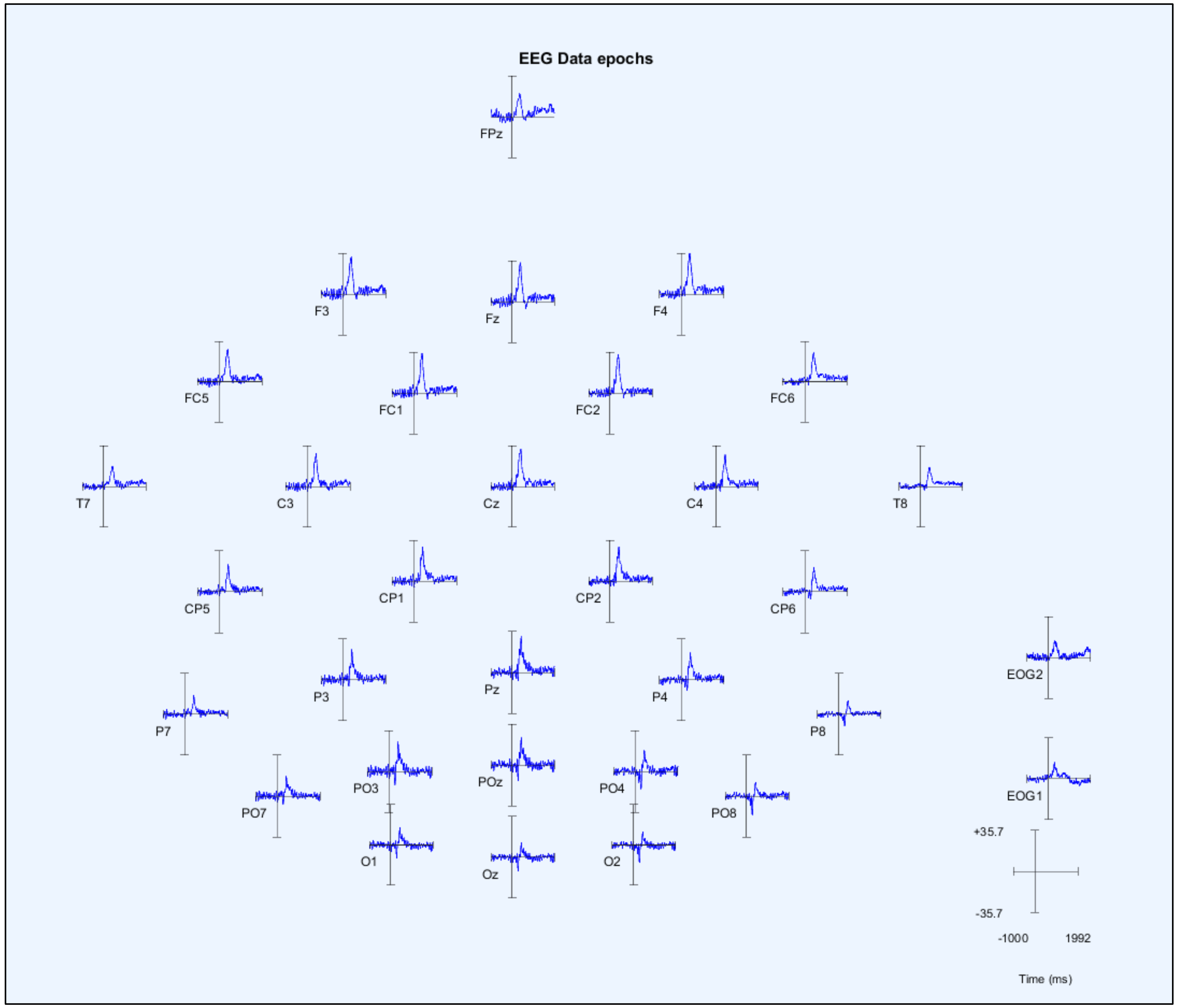
1. Plot > Channel ERPs > With scalp maps



Plotting Averaged Data

1. Plot > Channel ERPs > In scalp/rect array

Click on individual channels to enlarge

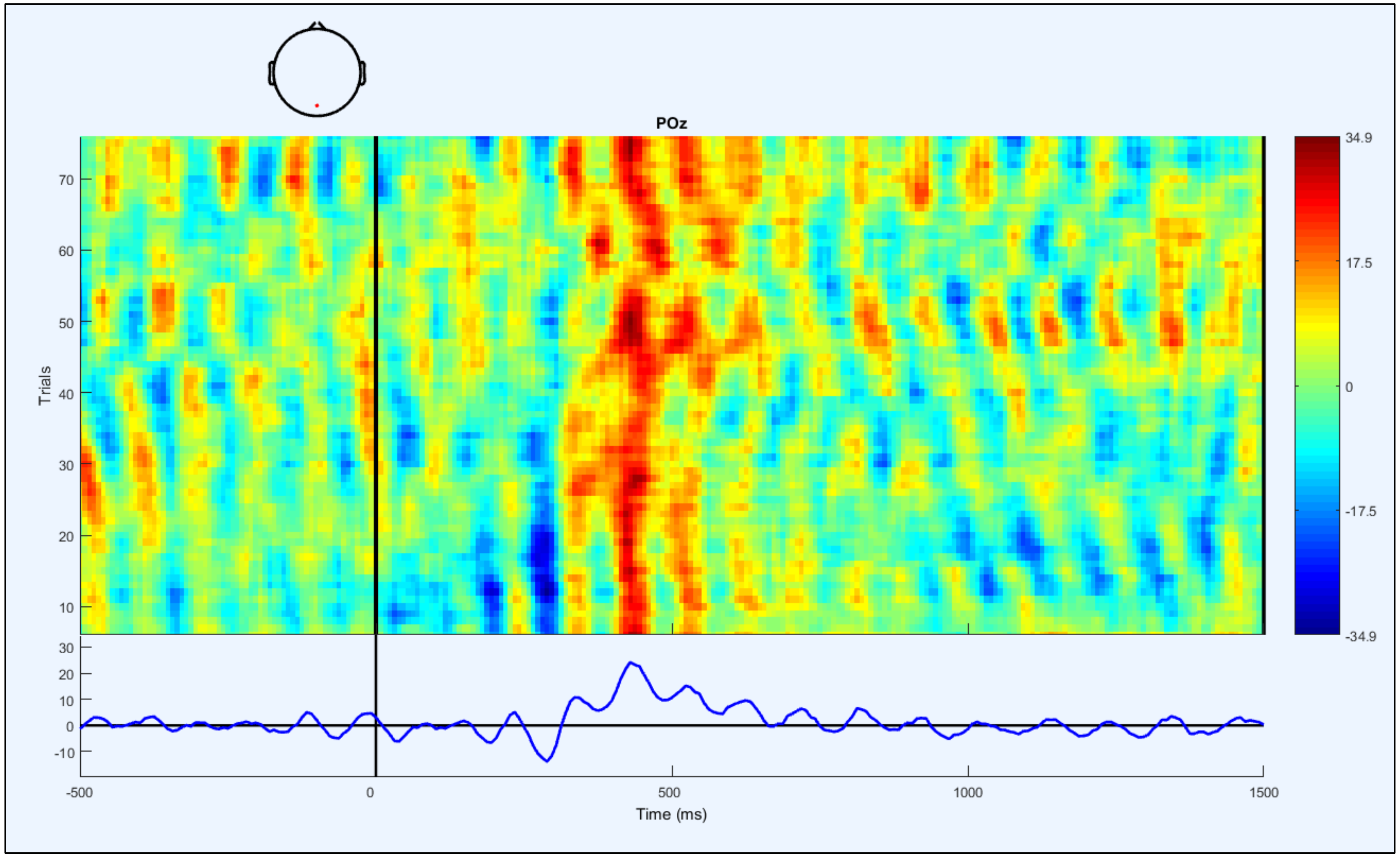


Plotting Averaged Data

1. Plot > Channel ERP image

    Channel = 27

    Time limits (ms) = -500 1500



Slides available for download at http://bit.ly/21ygstz

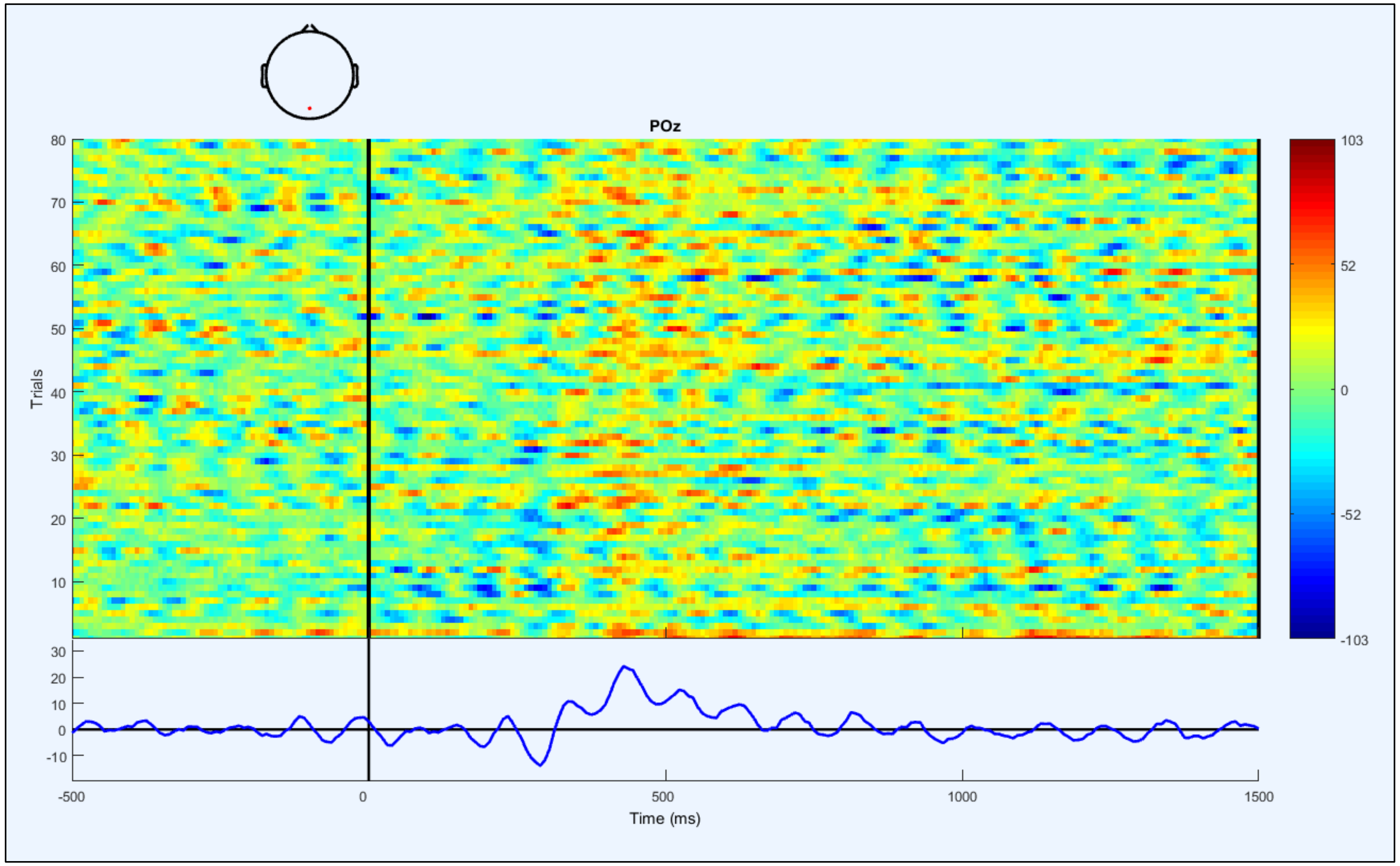
Plotting Averaged Data

1. Plot > Channel ERP image

   Channel = 27

   Time limits (ms) = -500 1500

   Smoothing = 0

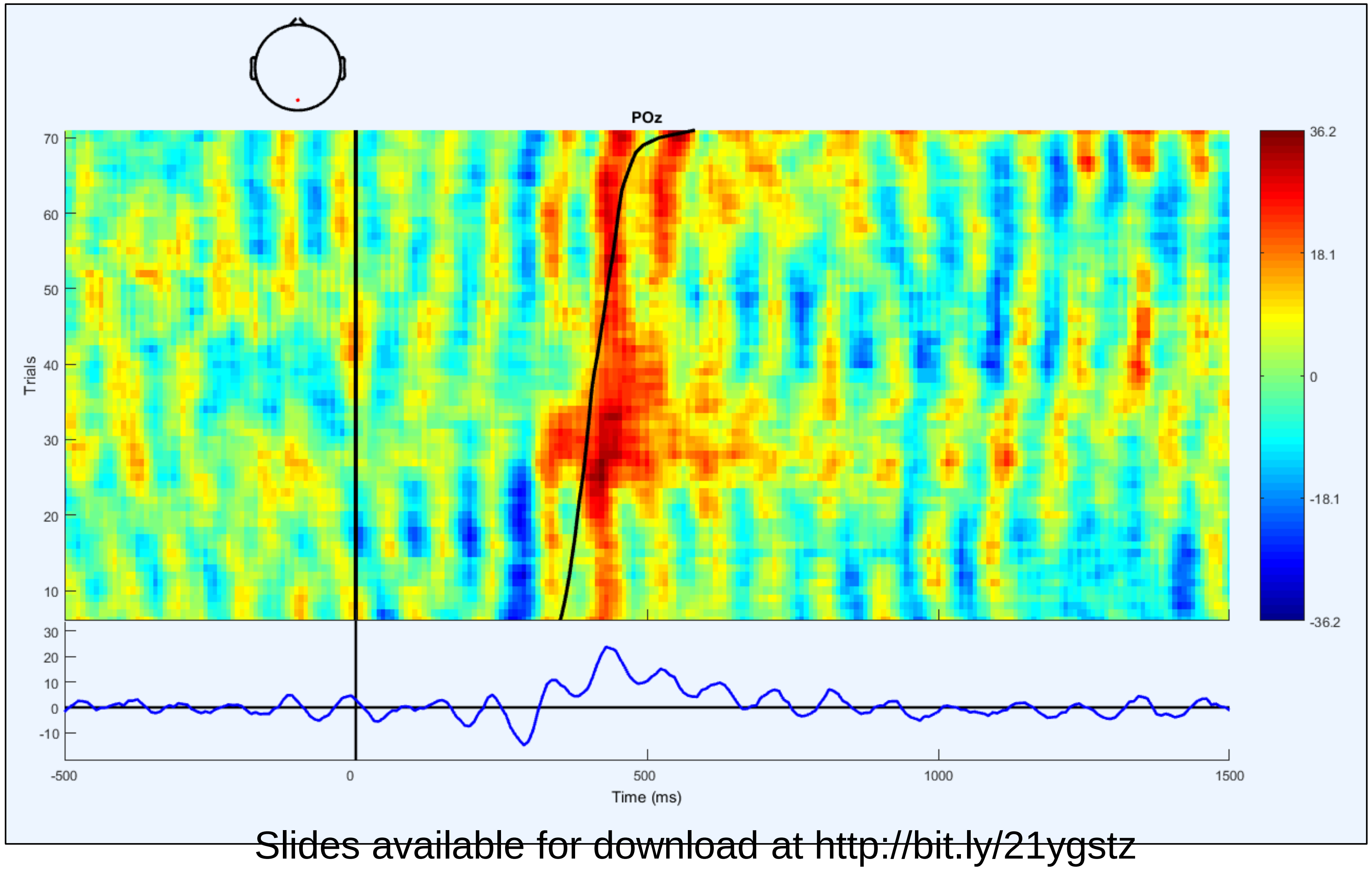


Slides available for download at http://bit.ly/21ygstz

Plotting Averaged Data

1. Plot > Channel ERP image

   Channel = 27
   Time limits (ms) = -500 1500
   Epoch-sorting field = ‘latency’
   Event type(s) = ‘rt’
   Smoothing = 10



Slides available for download at http://bit.ly/21ygstz

Some Final Comments

Moving from point-and-click to batch scripting:

```
>> eegh
```

EEGLAB is tremendously useful for some things, especially for the purposes of MVPA.

It has some limitations in other respects, though.
For instance, no function to detect amplitude of a peak.

In many cases, these limitations are addressed by separate plugins

- ERPLAB (Lopez-Calderon & Luck, 2014)
- sLORETA

Useful resource: http://sccn.ucsd.edu/wiki/EEGLAB_TUTORIAL_OUTLINE