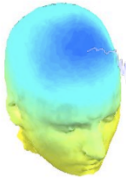


# EEGLAB Data Structures

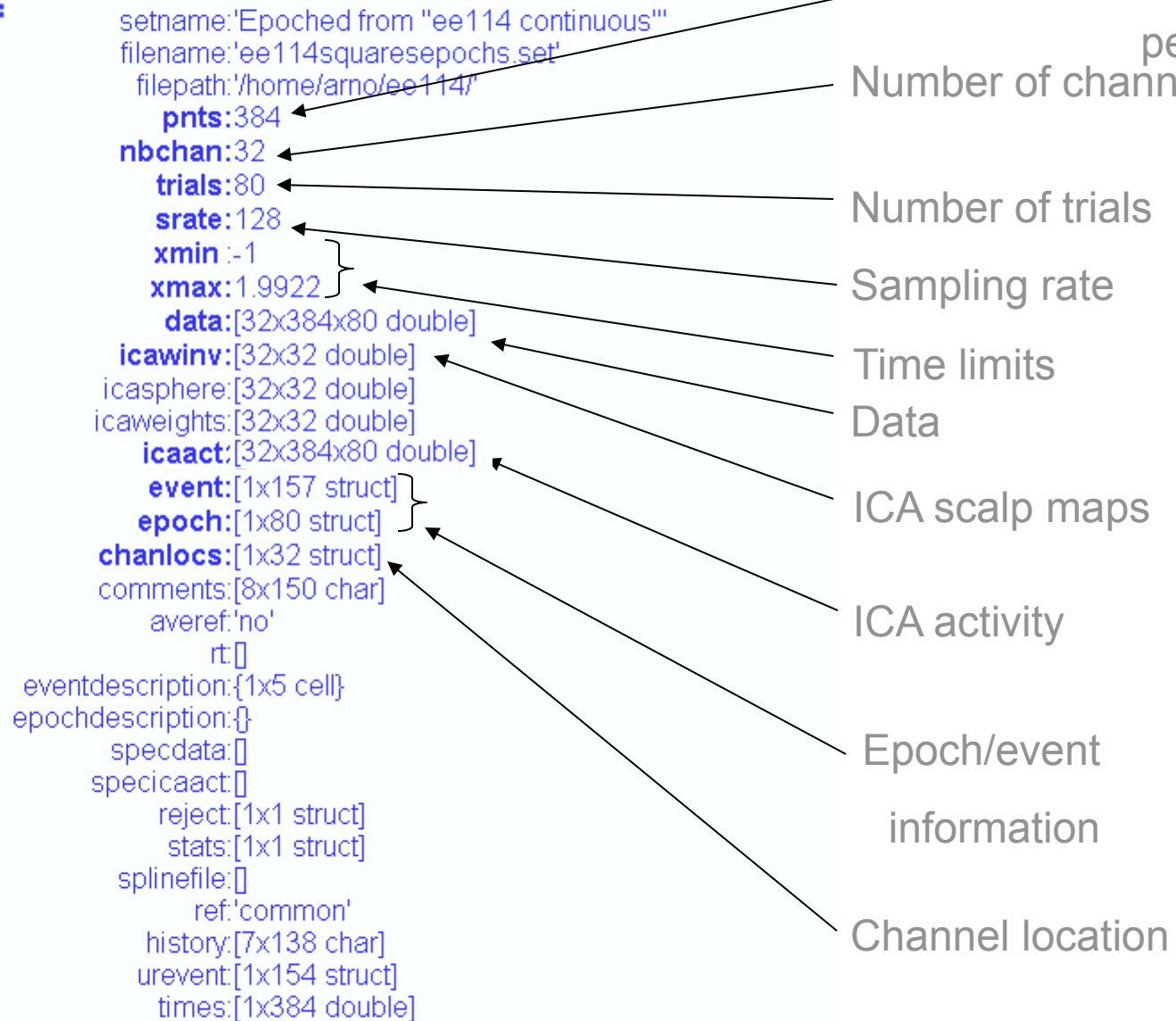


1. EEG
  - root 'dataset' structure
  - .data - the dataset data (2-D, 3-D matrix)
  - .chanlocs - channel locations substructure
  - .event - data events substructure
  - .epoch - data epochs substructure
3. ALLEEG
  - vector of loaded EEG datasets
4. CURRENTSET
  - index in ALLEEG of current EEG dataset
5. STUDY
  - root 'studysel' structure
  - .cluster - component clustering substructure



# EEG structure

EEG =



Number of data points

per trial

Number of channels

Number of trials

Sampling rate

Time limits

Data

ICA scalp maps

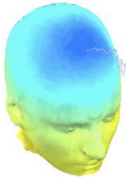
ICA activity

Epoch/event

information

Channel location

# 3 levels of functions



Administrative functions: handle EEG and ALLEEG structures

`eeglab()`, `eeg_checkset()`, `pop_delset()`, ...

Pop functions: interactive functions using EEG structure

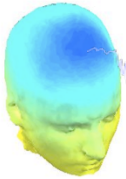
`pop_erpimage()`, `pop_topoplot()`, `pop_envtopo()`, ...

Signal processing functions: perform signal processing

`erpimage()`, `topoplot()`, `envtopo()`, ...



# Command line tools

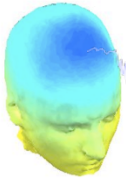


(Menus write both dataset and global history)

- Automated processing on groups of subjects (possibly on several processors).
- Richer options for plotting and processing functions (time-frequency decompositions, ...)
- Selecting data/epoch based on event context
- Custom processing...



# Using EEGLAB history for basic scripting



EEG.history → useful information

## Task 1

Create simple script using 'eegh'

## Task 2

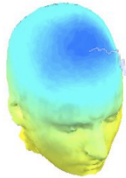
Eye-blink correction

Create a new EEG field

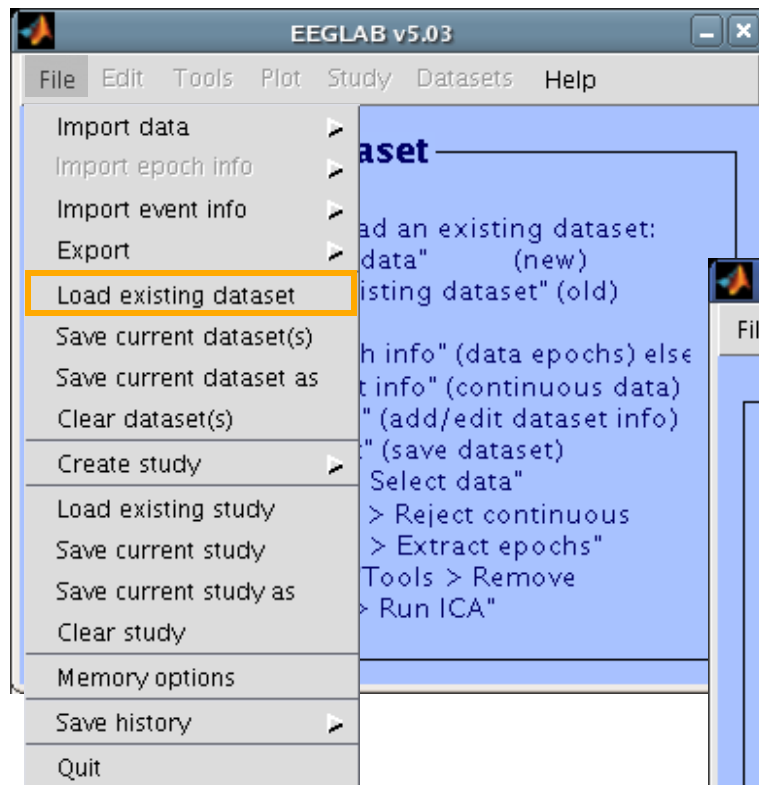
**Exercise...**



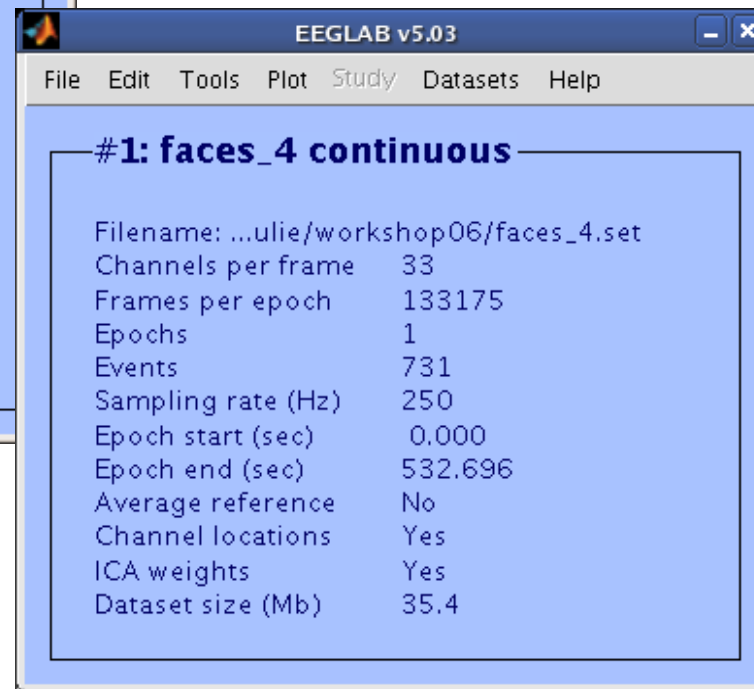
# Task 1: Retrieve dataset history



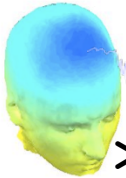
**Retrieve information about the processing history of a data set:**



**>> EEG.history**



# EEG.history



```
>> EEG.history
```

```
ans =
```

```
EEG = pop_loadbv('.../data/rawdata/', 'faces_4.vhdr');
```

```
EEG.setname='faces_4_continuous';
```

```
EEG = eeg_checkset( EEG );
```

```
EEG.chanlocs=pop_chanedit(EEG.chanlocs, 'load',[],...  
    'load',{ '.../wsporto/data/chan_locs.elp', 'filetype',...  
    'besa (elp)' }, 'eval',...  
    'chantmp = pop_chancenter( chantmp, [],[]);');
```

```
EEG = pop_saveset( EEG, 'faces_4.set', '.../workshop/');
```

```
EEG = pop_multifit(EEG, [1:33] , 'settings',{},'threshold',...  
    40, 'plotopt',{ 'normlen', 'on', 'image','fullmri'});
```

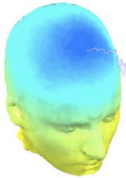
```
pop_topoplot(EEG,0, [1:12] , 'faces_4 continuous',...  
    [3 4] ,0, 'electrodes', 'off', 'masksurf', 'on');
```

```
% no history for manual DIPFIT dipole localization
```

```
EEG = pop_saveset( EEG, 'faces_4.set', '.../workshop/');
```



# Using EEGLAB history for basic scripting



EEG.history → useful information

## **Task 1**

Create simple script using 'eegh'

## **Task 2**

Eye-blink correction

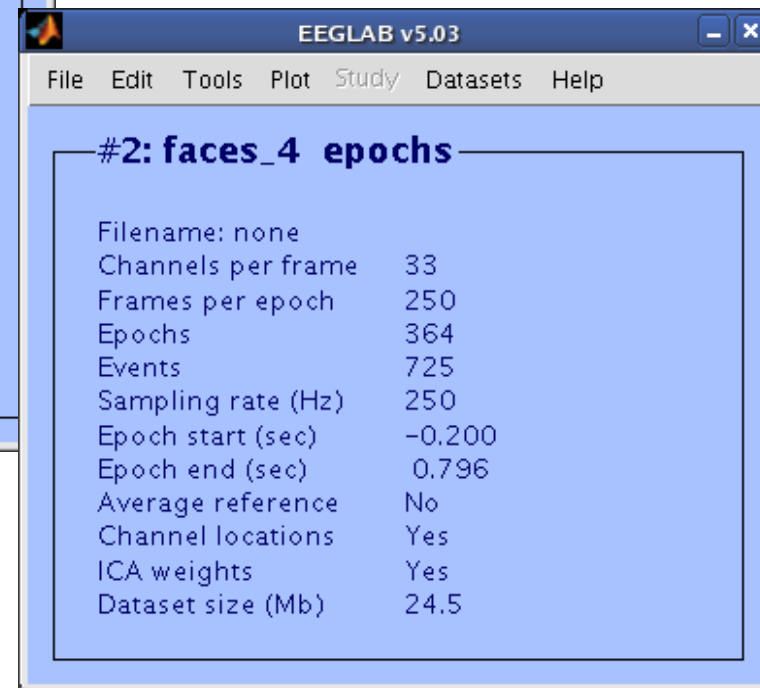
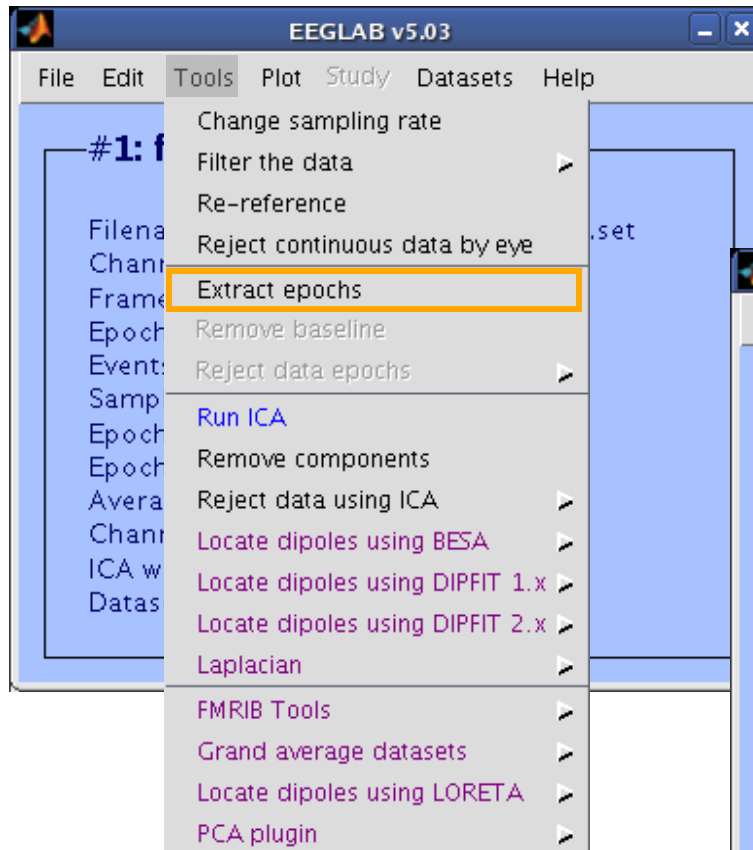
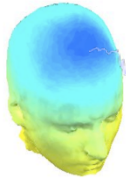
Create a new EEG field

## **Exercise...**

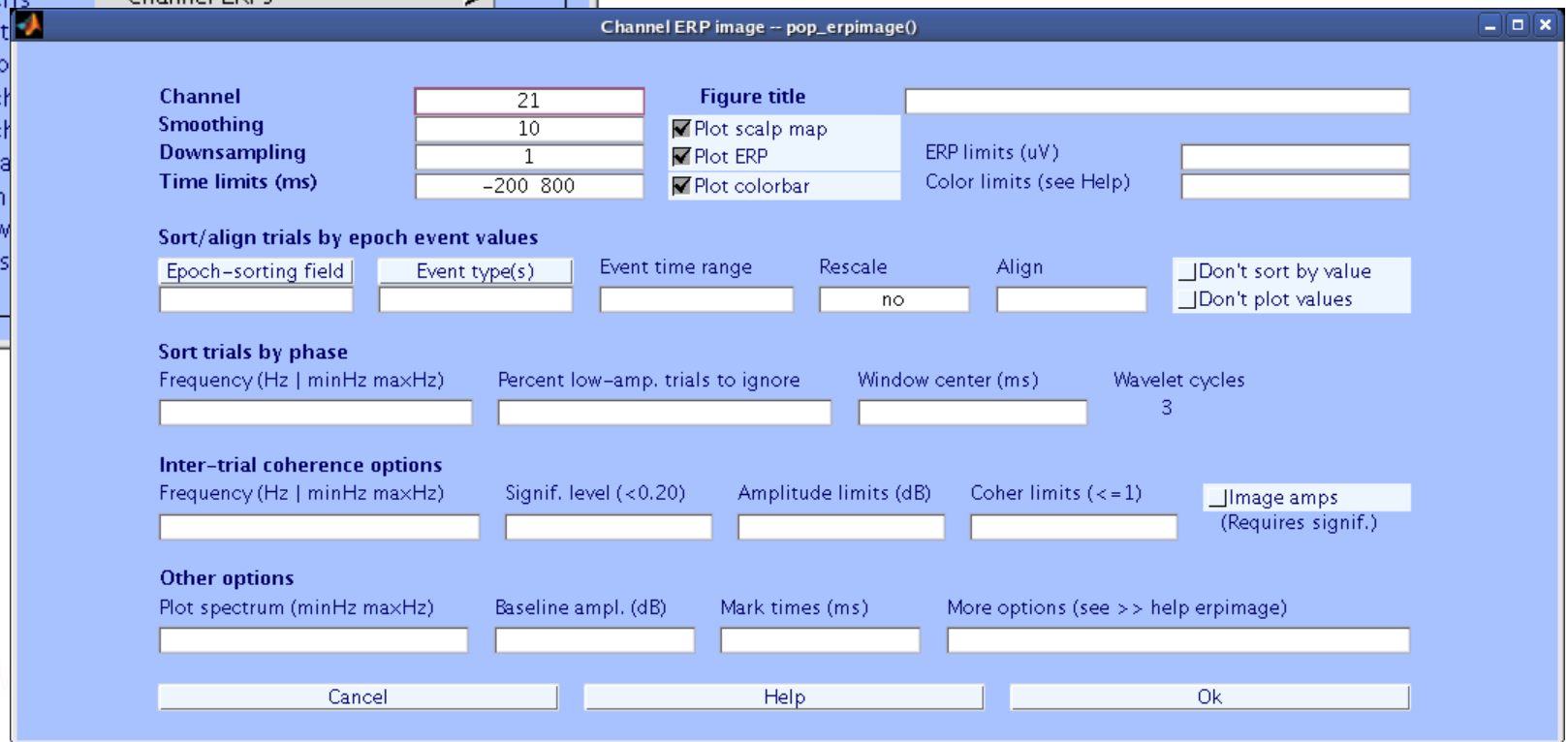
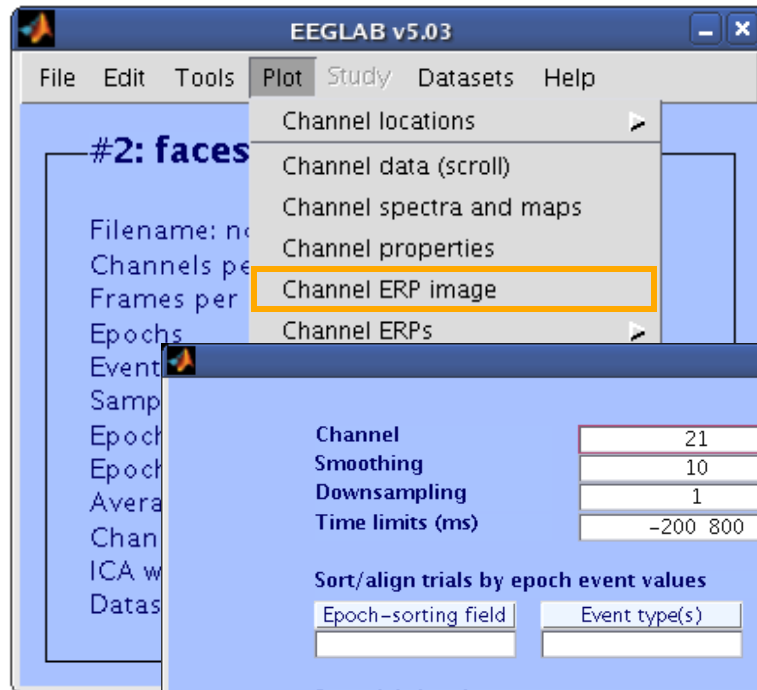
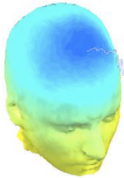




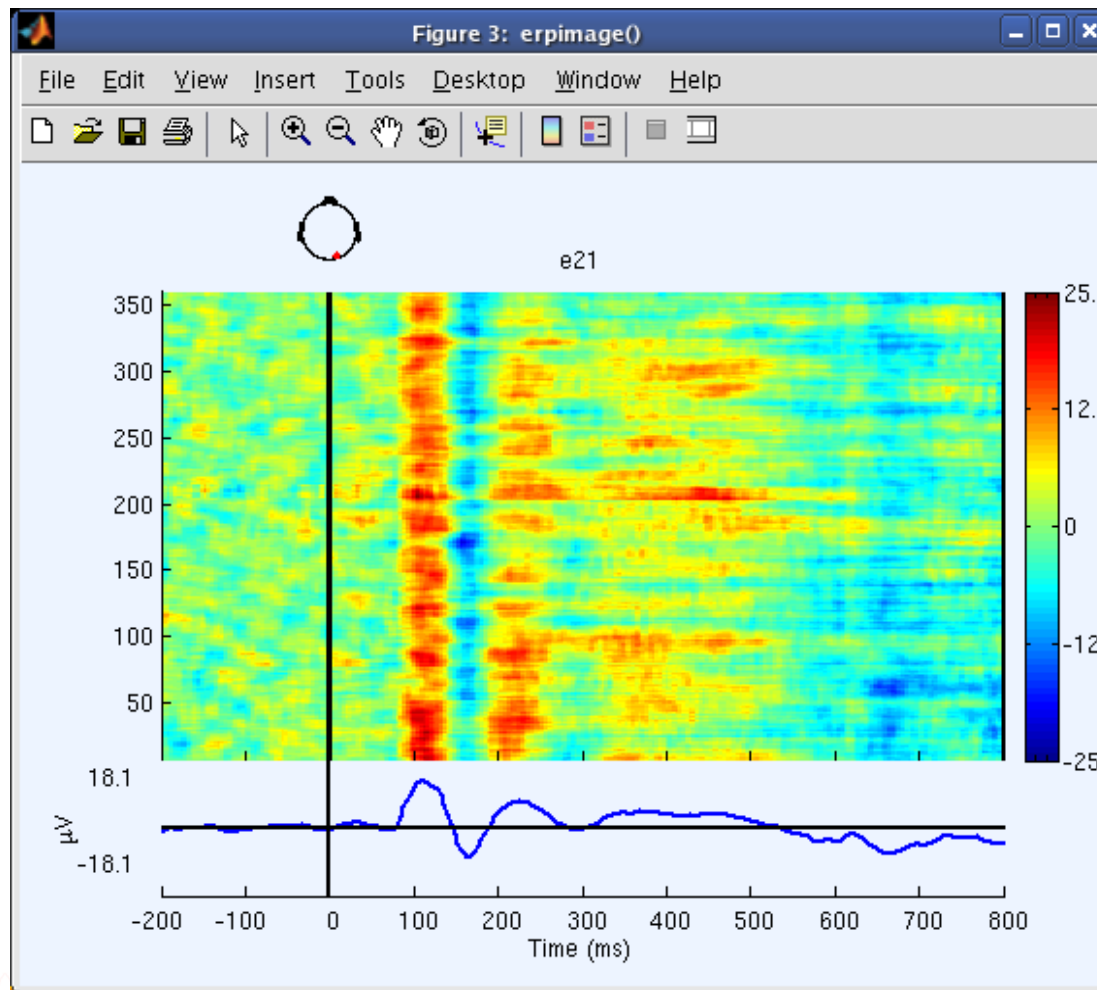
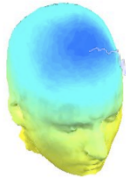
# Task 1: plot an ERP image...



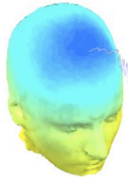
# Task 1: Plot an ERPimage...



# Task 1: Resulting figure



# Script task 1 using 'eegh'

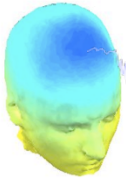


**Write a script to do this:**

```
>> eegh
```



# Script task 1 using 'eegh'



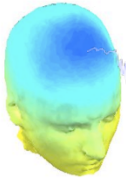
```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

EEG = pop_loadset('filename','faces_4.set','filepath',...
    '..\data\');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 0);

EEG = pop_epoch( EEG, { 'face' 'object' }, [-0.2 0.8],...
    'newname', 'faces_4 epochs', 'epochinfo', 'yes');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1);
EEG = pop_rmbase( EEG, [-200 0]);
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

figure; pop_erpimage(EEG,1, [21], [], 'e21', 10, 1, {}, [],...
    '', 'yerplabel', '\muV', 'topo',...
    { [21] EEG.chanlocs EEG.chaninfo } , 'erp' , 'cbar');
```

# Using EEGLAB history for basic scripting



EEG.history → useful information

## Task 1

Create simple script using 'eegh'

## Task 2

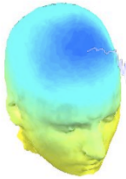
Eye-blink correction

Create a new EEG field

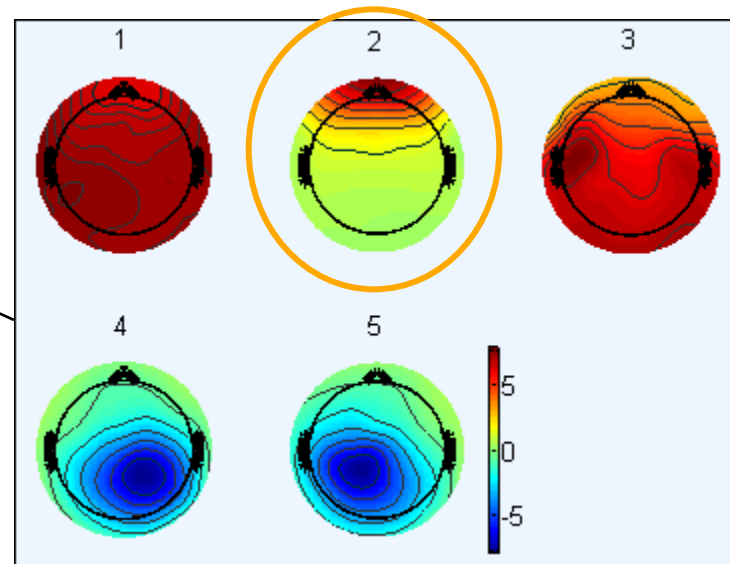
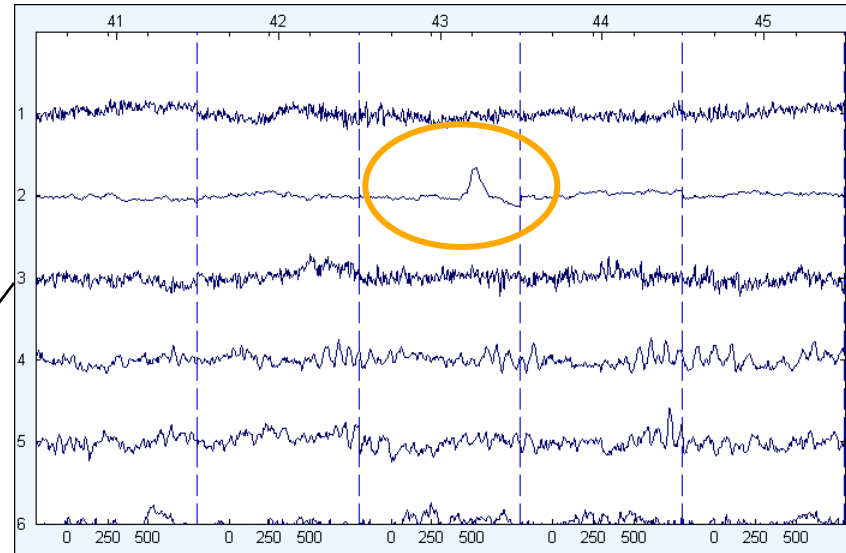
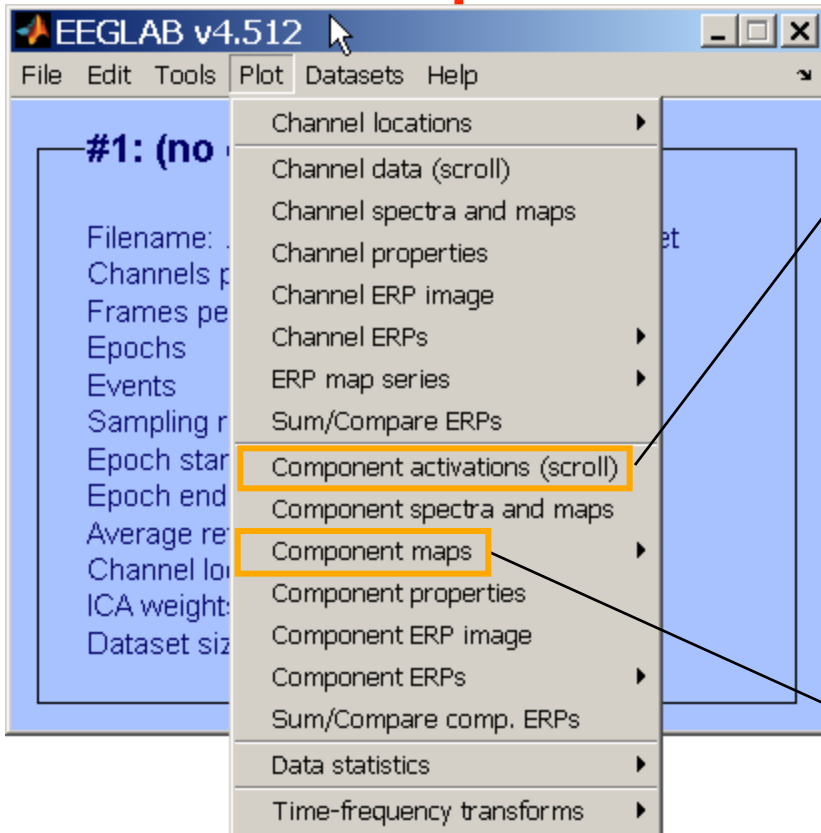
Exercise...



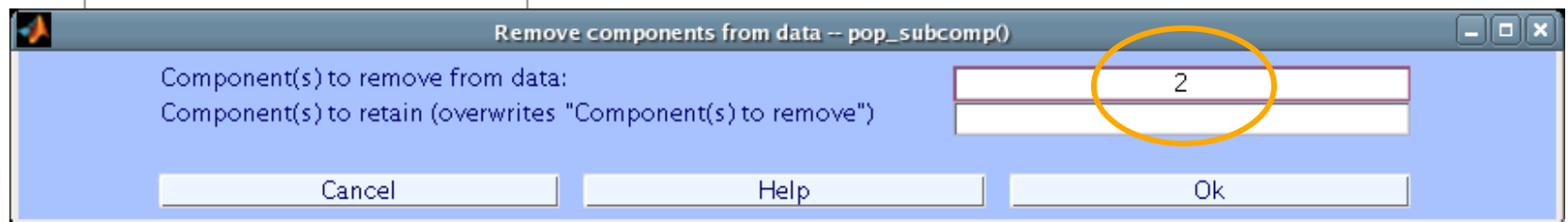
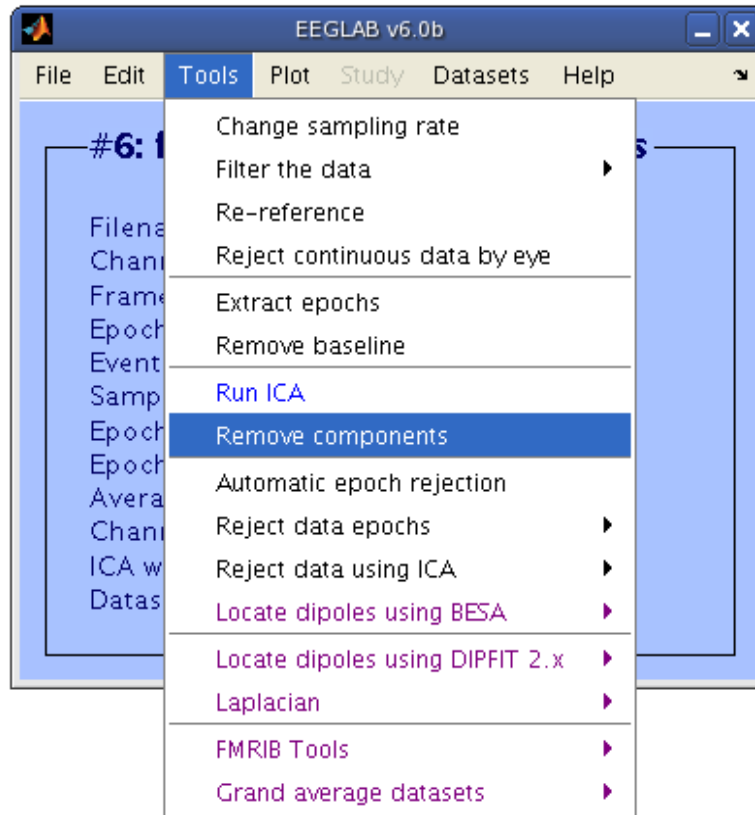
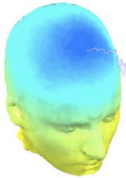
# Eye blink correction



Identify eye-blink components:

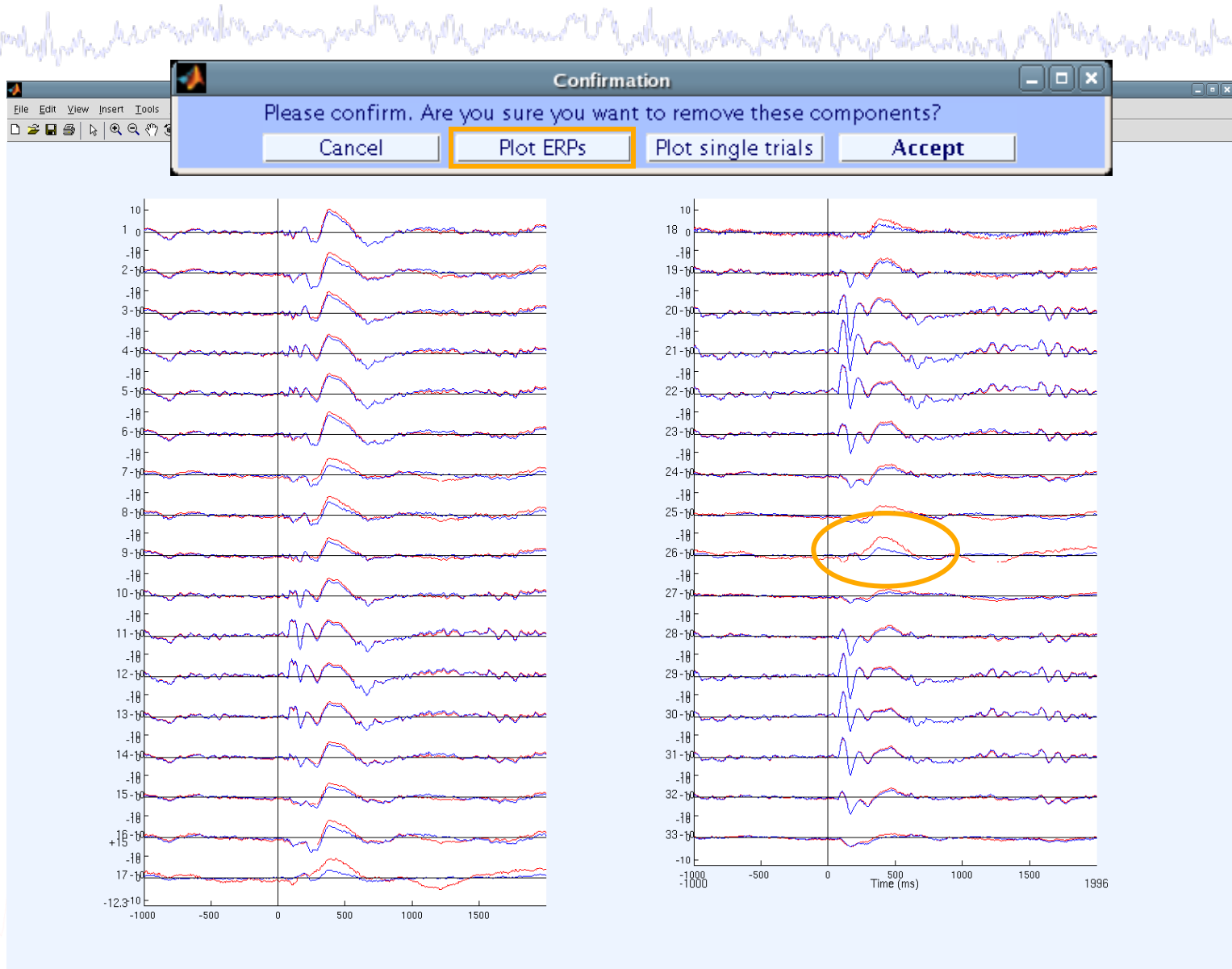
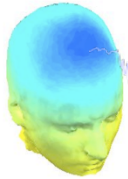


# Eye blink correction

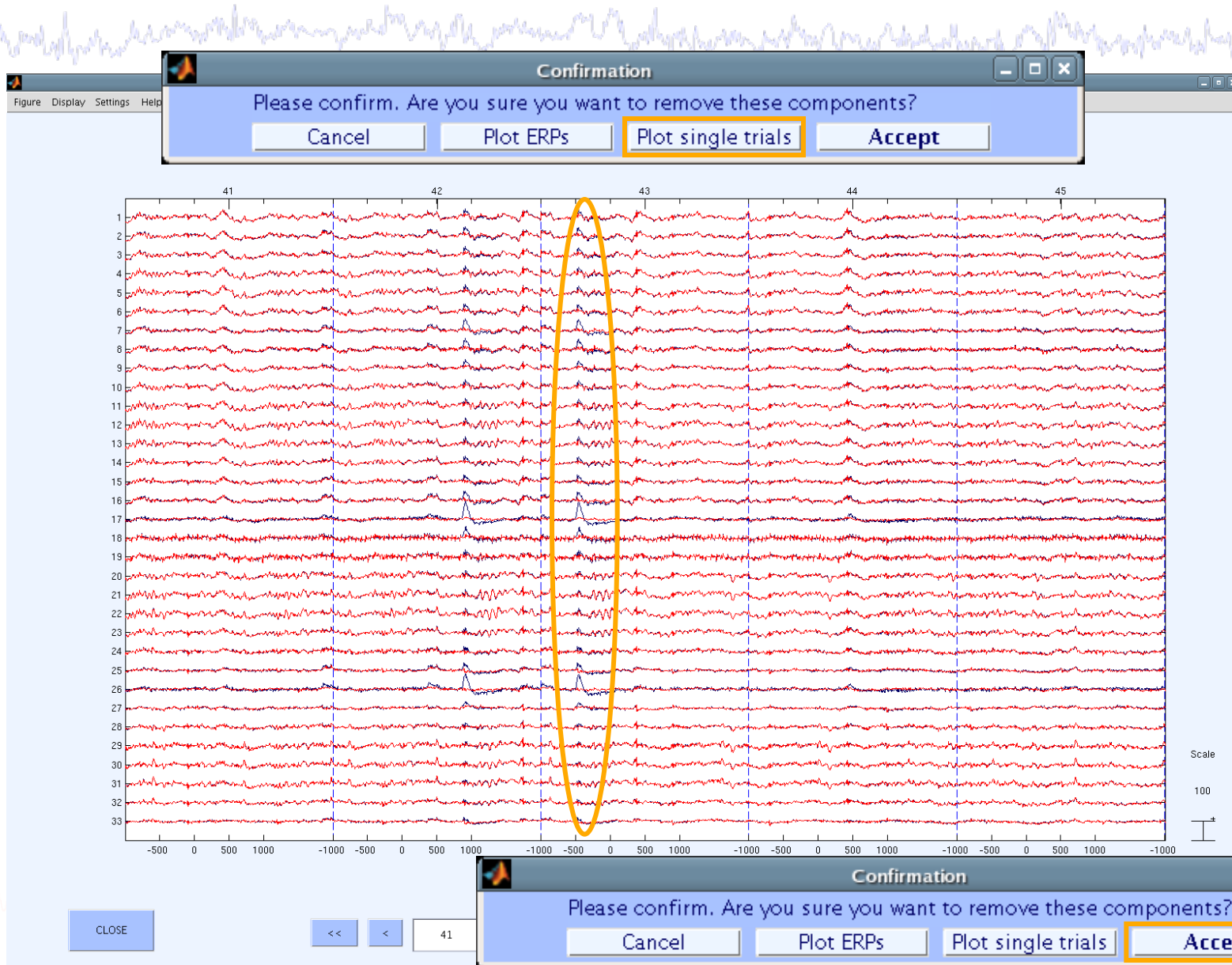
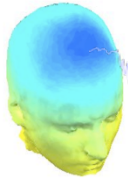




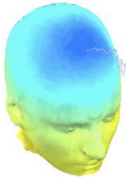
# Eye blink correction



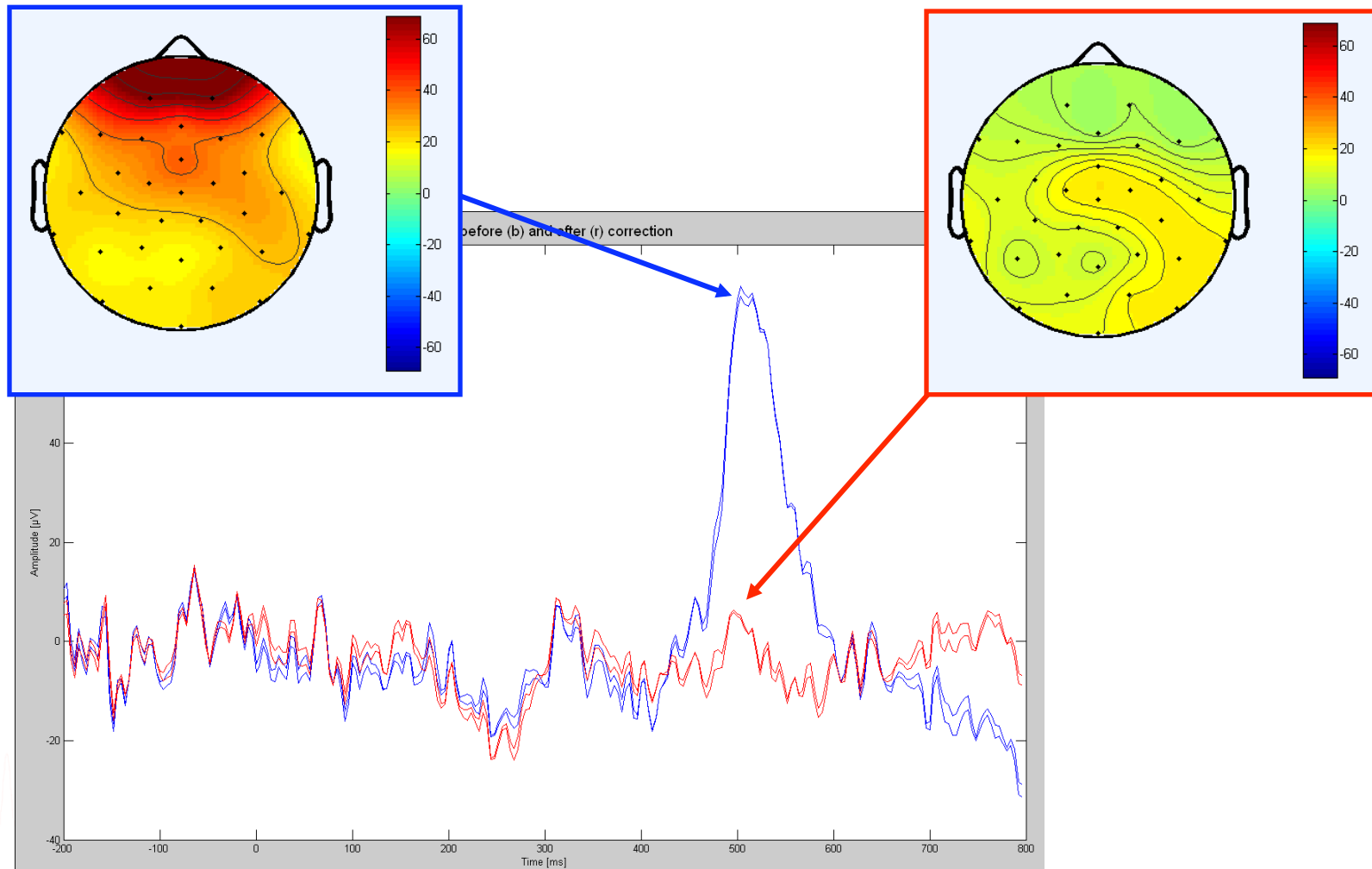
# Eye blink correction



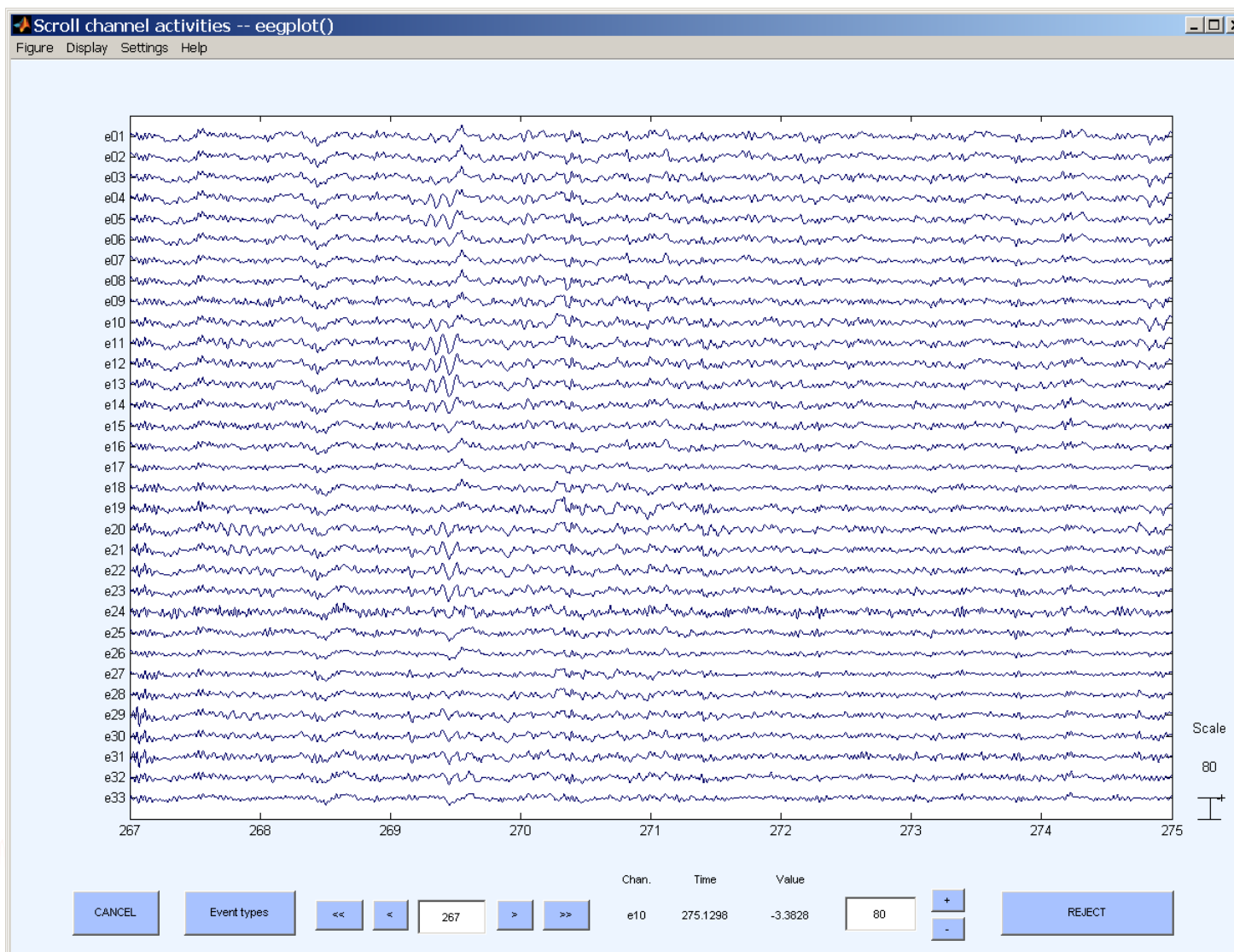
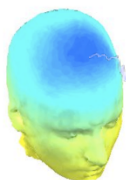
# Eye blink correction



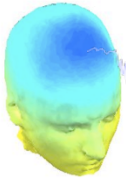
Trial 43: Fp1/2 before (b) and after (r) correction



# Task 2: Eye blink correction



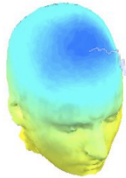
## Task 2: Script an addition to EEG structure



- EEG structure can be extended to include new fields
  - store information for future access
- Task:
  - write a semi-automatic script to save eye blink IC index as 'EEG.blink'



# Create initial script from 'eegh'



```
>> eeglab
```

```
% using GUI:  
% load dataset,  
% plot component maps in 2D  
% save current dataset as... (force a resave)
```

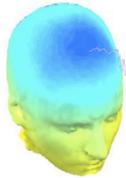
```
>> eegh
```

```
% open Matlab editor:  
>> edit
```

```
% copy & paste eegh results into a new  
% file and save it as faces2.m
```



## Task 2: Modify faces2.m



```
% faces2.m
```

```
infile = 'faces_3.set'; % define input .set file  
datpath = 'C: \...\workshop\data\'; % input directory
```

```
% start eeglab
```

```
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
```

```
% load data set
```

```
EEG = pop_loadset(infile, datpath);
```

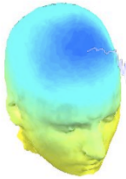
```
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG);
```

```
% plot IC maps
```

```
pop_topoplot(EEG,0, [1:size(EEG.data,1)],infile,...  
    [6 6] ,0,'electrodes', 'off', 'masksurf', 'on');
```



# Task 2: Scripting ICA eye blink correction



```
% faces2.m (cont'd)

% Enter eye blink indices manually
EEG.blink = input('Enter eye blink component indices: ');

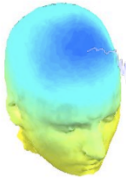
% save info in current data set
EEG = pop_saveset(EEG, infile, datpath);
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

% update EEGLAB gui
eeglab redraw;
```





# Using EEGLAB history for basic scripting



EEG.history → useful information

## Task 1

Create simple script using 'eegh'

## Task 2

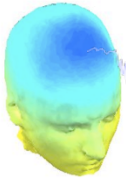
Eye-blink correction

Create a new EEG field

## Exercise...



# Exercise



Script it yourself!

- **Novice**

Script using eegh:

load a dataset, epoch on 'face' and 'object', plot erpimage for any channel or component, copy eegh results to a script file and run your script!

- **Intermediate**

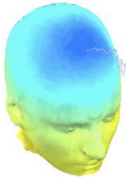
Script a semi-automatic script to add an EEG.blink field into the EEG structure.

\*\* All scripts for exercises can be found in .../  
EEGLAB\_WORKSHOP/scripts/practicum\_4.m

(ALL "DATPATH" VARIABLES MUST BE CHANGED TO POINT  
TO THE DATA ON YOUR COMPUTER!)



# Exercise: NOVICE



```
>> eeglab
```

```
% repeat all steps of task 1:  
% load dataset,  
% epoch on 'face' and 'object'  
% plot erpimage for any channel
```

```
>> eegh
```

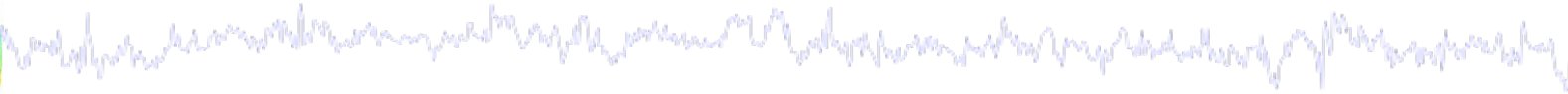
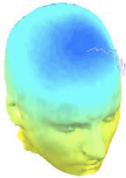
```
% open Matlab editor:  
>> edit
```

```
% copy & paste eegh results into a new  
% file and save it (../faces1.m)
```

```
>> clear  
>> close all  
>> faces1  
>> eeglab redraw
```

For reference: example script  
saved as: '...\scripts\practicum\_4.m'

# Exercise: INTERMEDIATE



```
% faces2.m
infile = 'faces_3.set';
inpath = '...\data\';
% start eeglab
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
% load data set
EEG = pop_loadset(infile, inpath);
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG);
% plot IC maps
pop_topoplot(EEG,0, [1:size(EEG.data,1)],infile,...
    [6 6] ,0,'electrodes', 'off', 'masksurf', 'on');
% Enter eye blink indices manually
EEG.blink = input('Enter eye blink component indices:');
% save info in current data set
EEG = pop_saveset(EEG, infile, inpath);
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
eeglab redraw;
```

For reference: example script  
saved as: '...\scripts\practicum\_4.m'