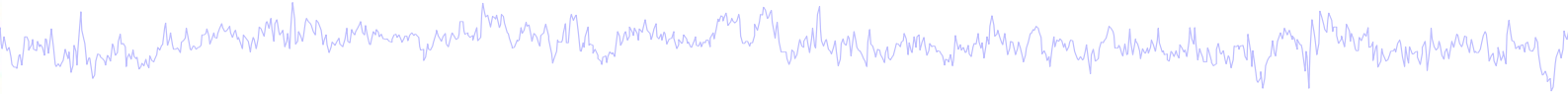


Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Exercise...

Task 2

Eye-blink correction

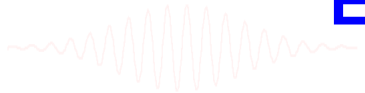
Create a new EEG field

Exercise...

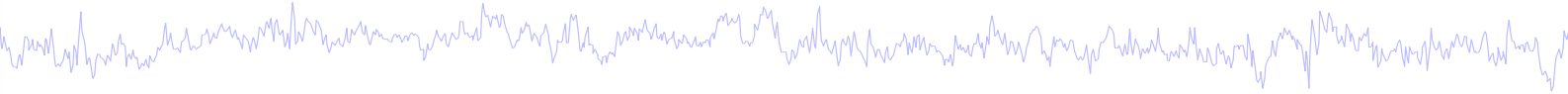
Task 3

Script to process multiple data sets

Exercise...



The example data: faces vs. objects

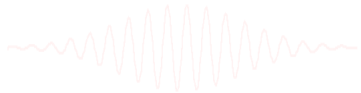


Data

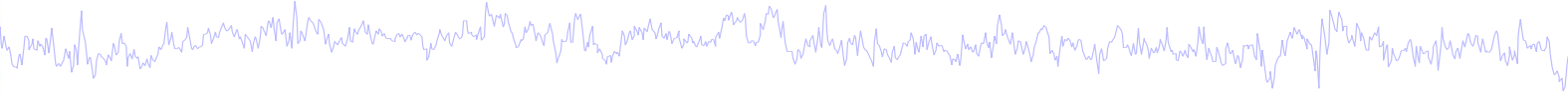
33 channel EEG, nose-tip reference, 250 Hz sampling rate, 0.5-100 Hz, 16 bit, BrainAmps

Task

speeded discrimination between objects and faces, 500 ms presentation duration, ISI 500-1900 ms, 362 trials



Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Exercise...

Task 2

Eye-blink correction

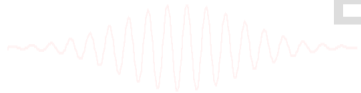
Create a new EEG field

Exercise...

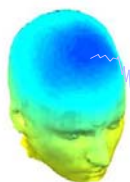
Task 3

Script to process multiple data sets

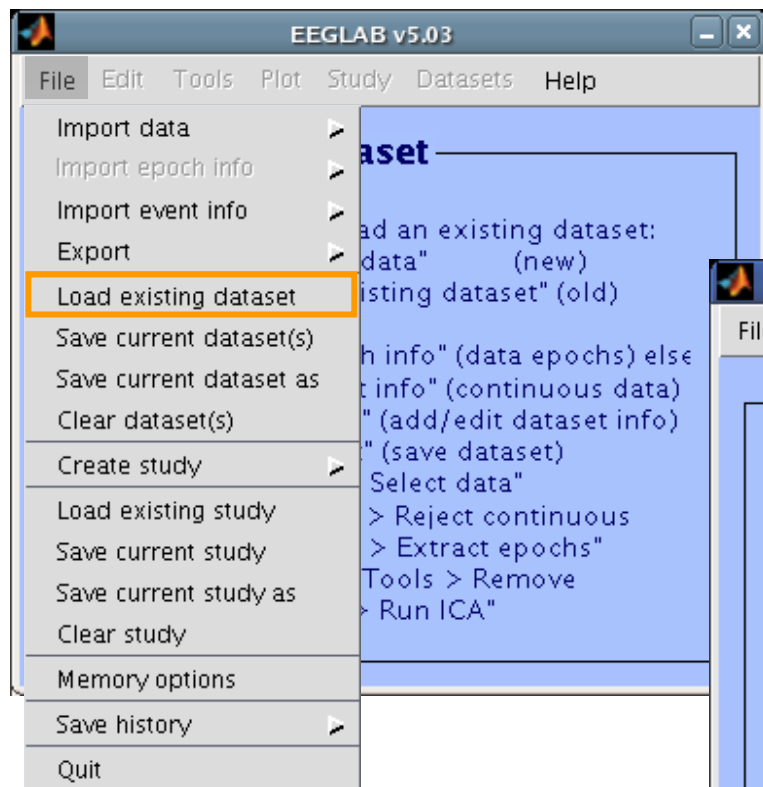
Exercise...



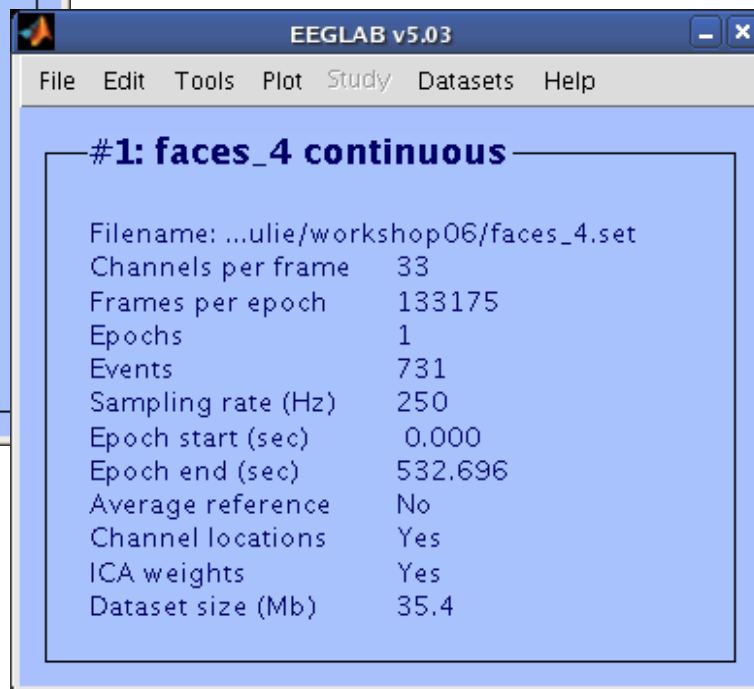
Task 1: Retrieve dataset history



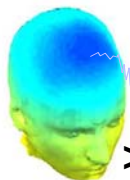
How do I retrieve information about the processing history of a data set?



>> **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'});
```

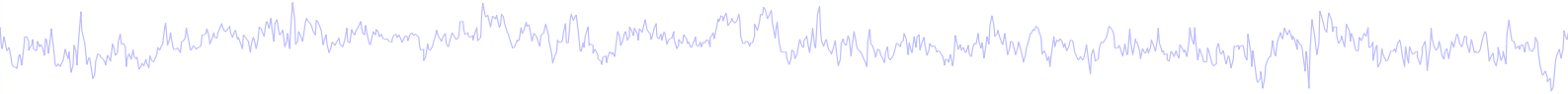
```
pop_topoplot(EEG,0, [1:12] , 'faces_4 continuous',...  
    [3 4] ,0, 'electrodes', 'off', 'masksurf', 'on');
```

```
% 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'

Exercise...

Task 2

Eye-blink correction

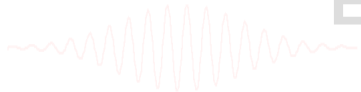
Create a new EEG field

Exercise...

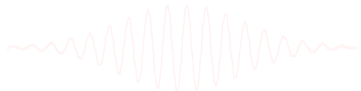
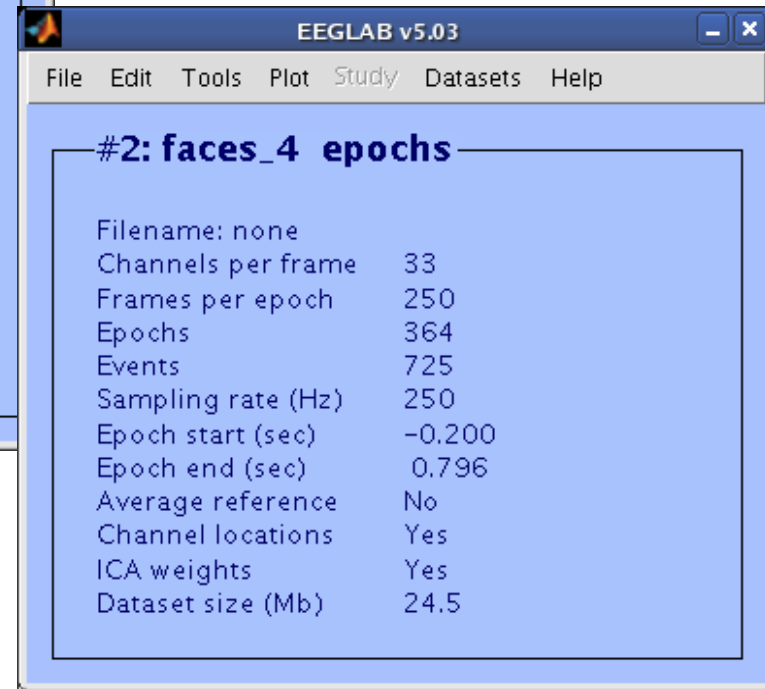
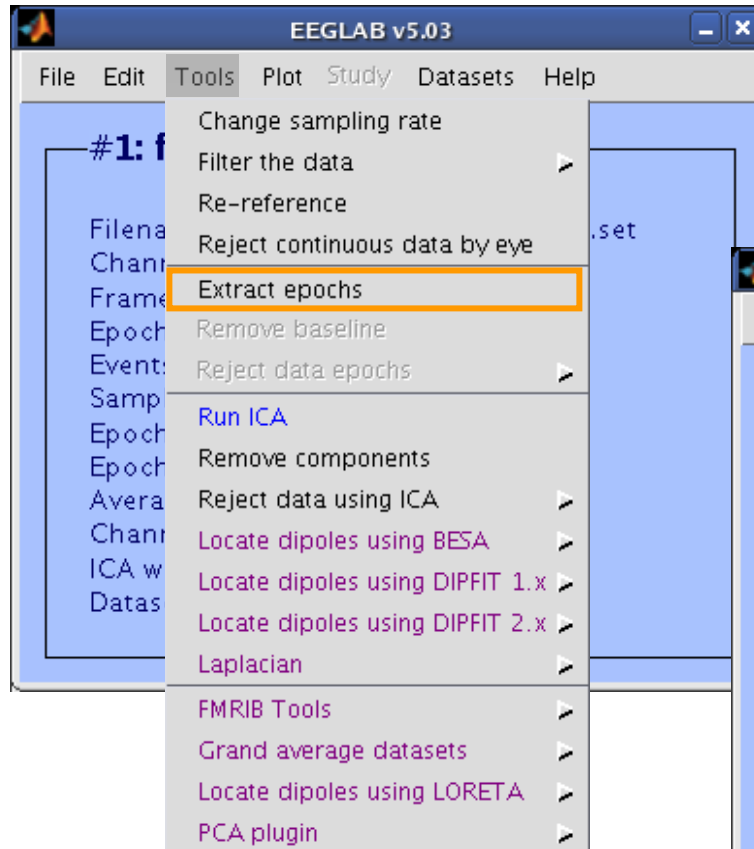
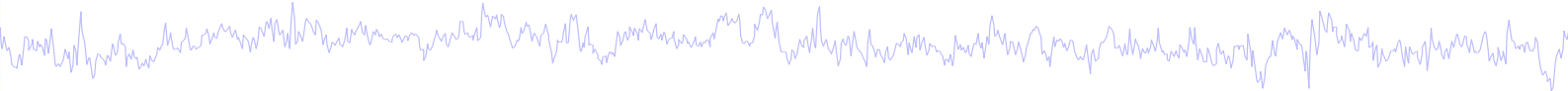
Task 3

Script to process multiple data sets

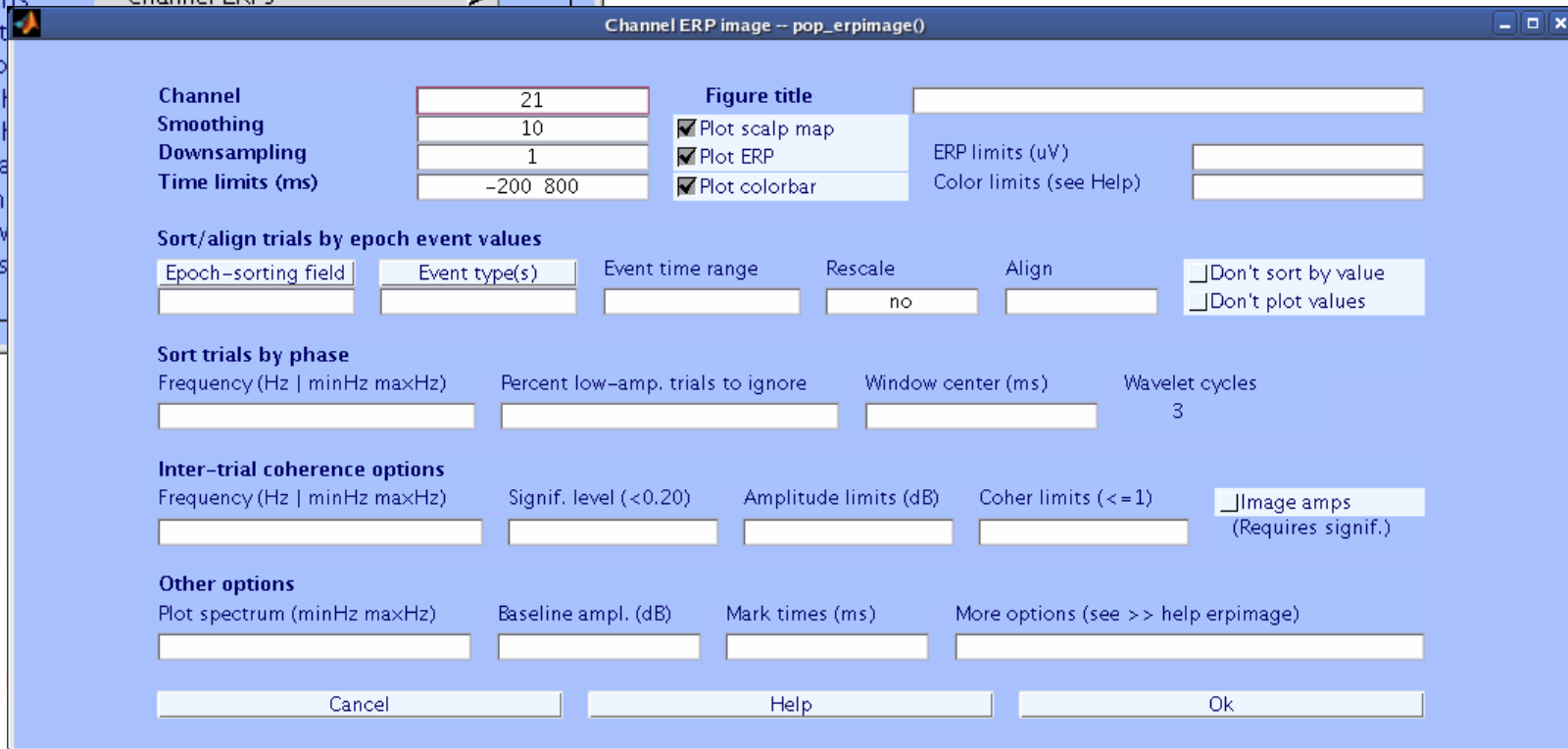
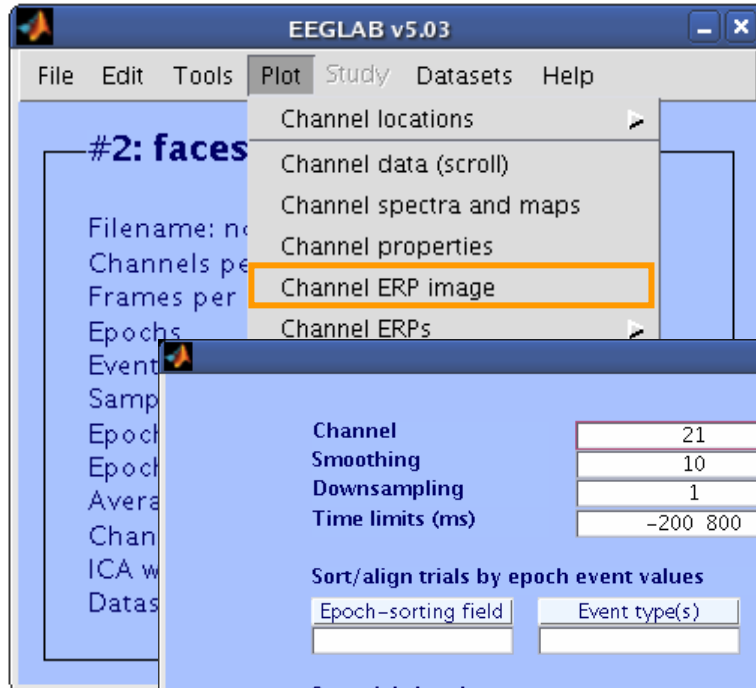
Exercise...



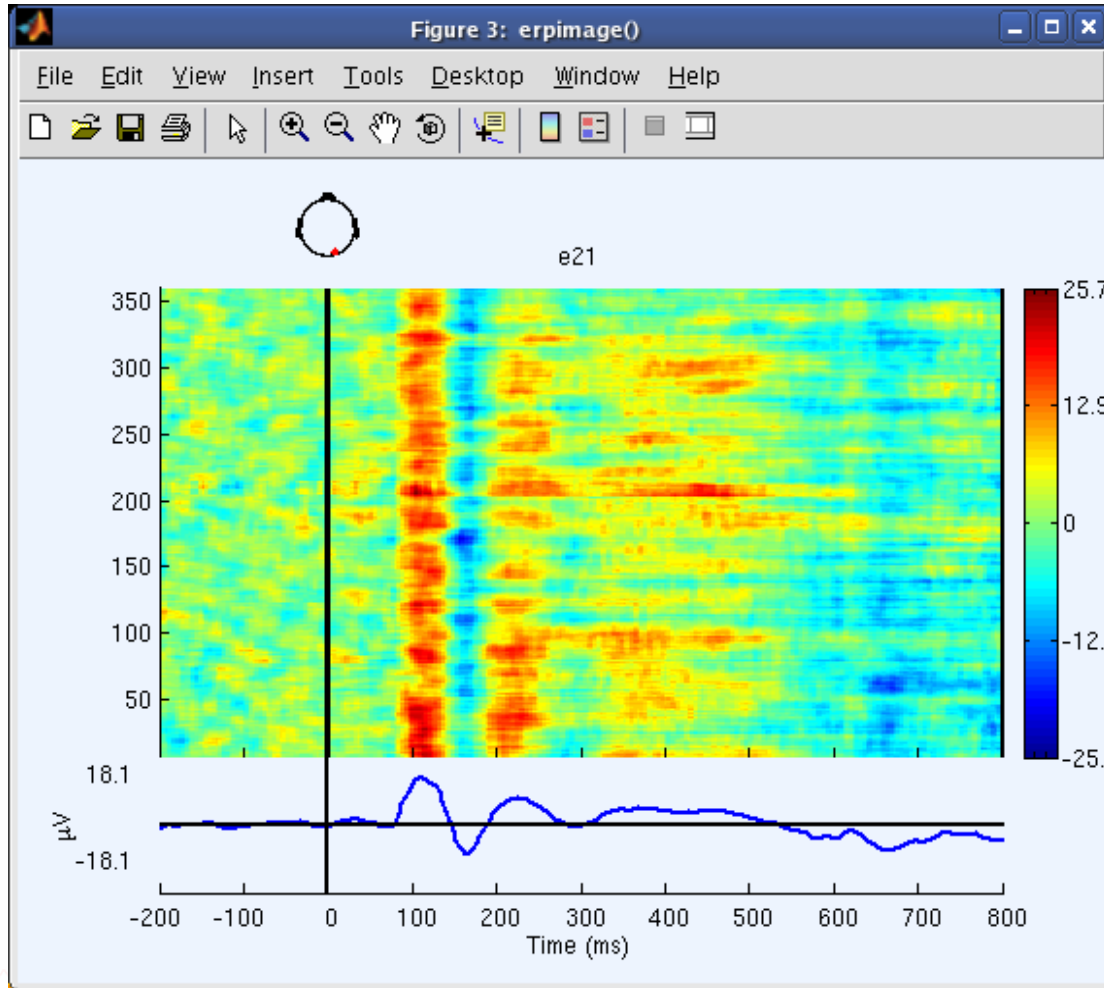
Task 1: plot an ERP image...



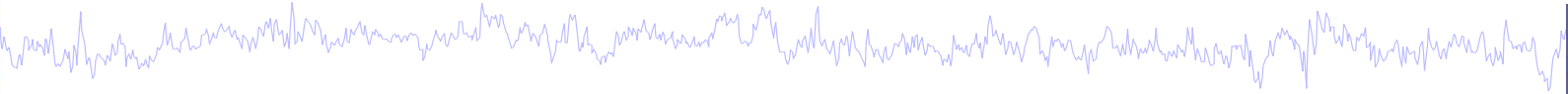
Task 1: Plot an ERP image...



Task 1: Resulting figure

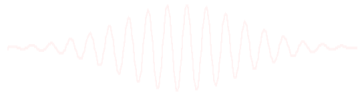


Script task 1 using 'eegh'

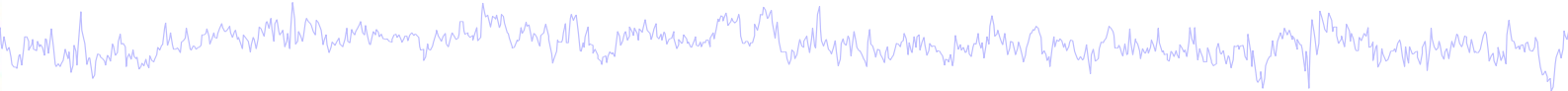


How do I write a script to do this?

```
>> eegh
```



Script task 1 using 'eegh'



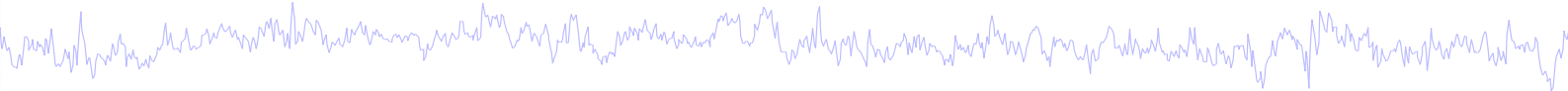
```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

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');
```

Exercise



```
>> eeglab
```

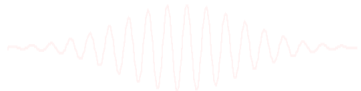
```
% repeat all steps of task 1:  
% load dataset,  
% epoch on 'face' and 'object'  
% plot erpimage for any channel
```

```
>> eegh
```

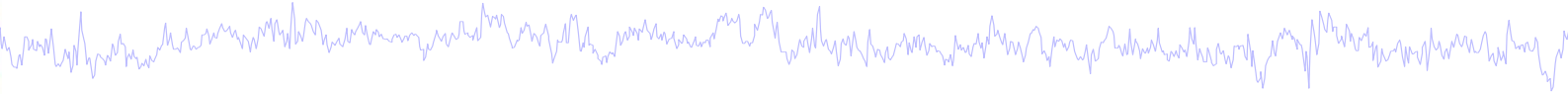
```
% open Matlab editor:  
>> edit
```

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

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



Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Exercise...

Task 2

Eye-blink correction

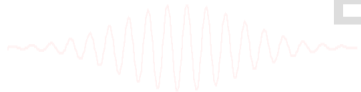
Create a new EEG field

Exercise...

Task 3

Script to process multiple data sets

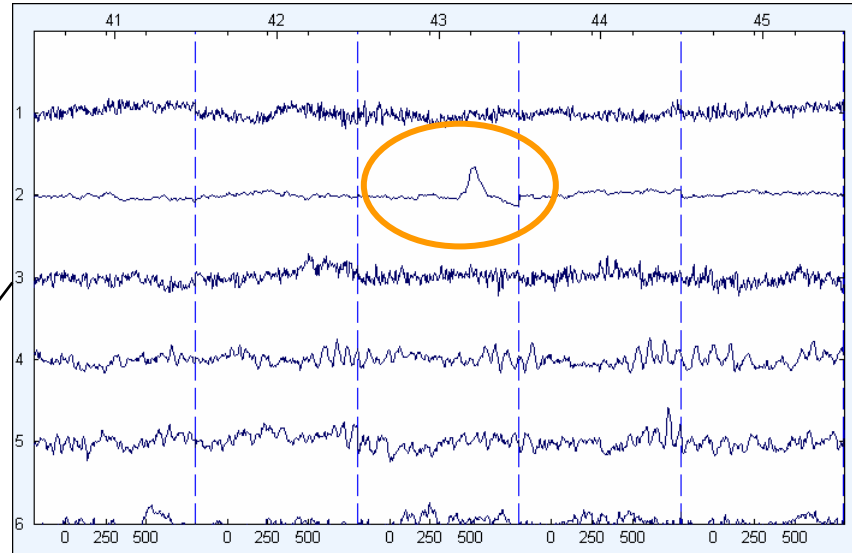
Exercise...



Eye blink correction



How do I identify eye-blink components?



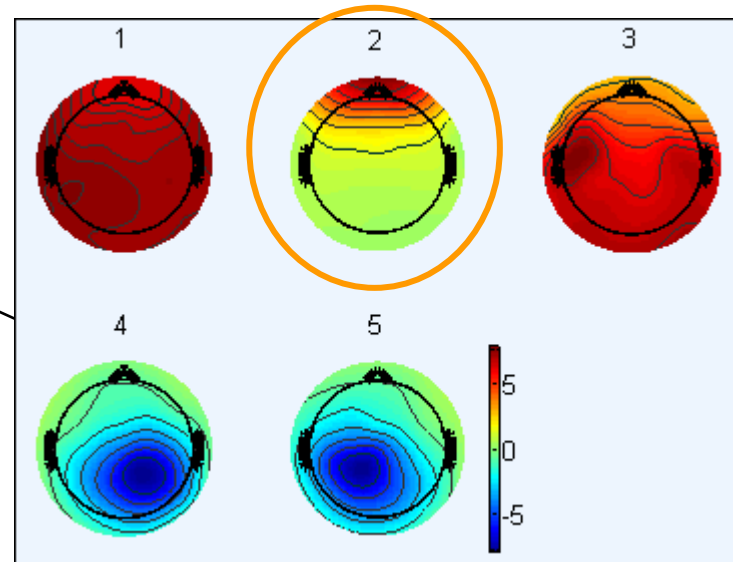
EEGLAB v4.512

File Edit Tools Plot Datasets Help

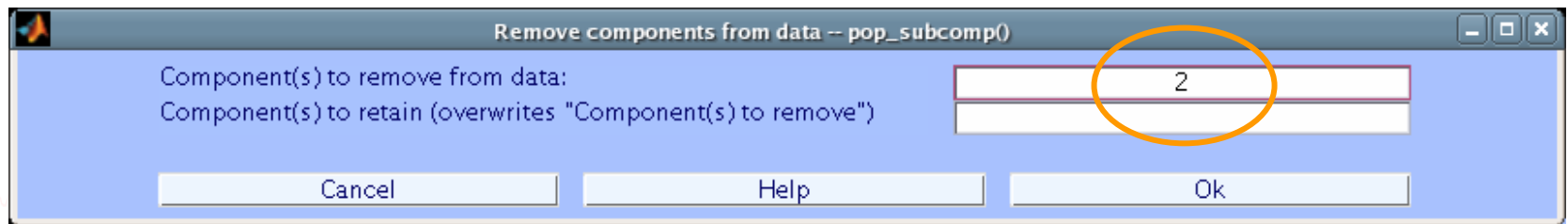
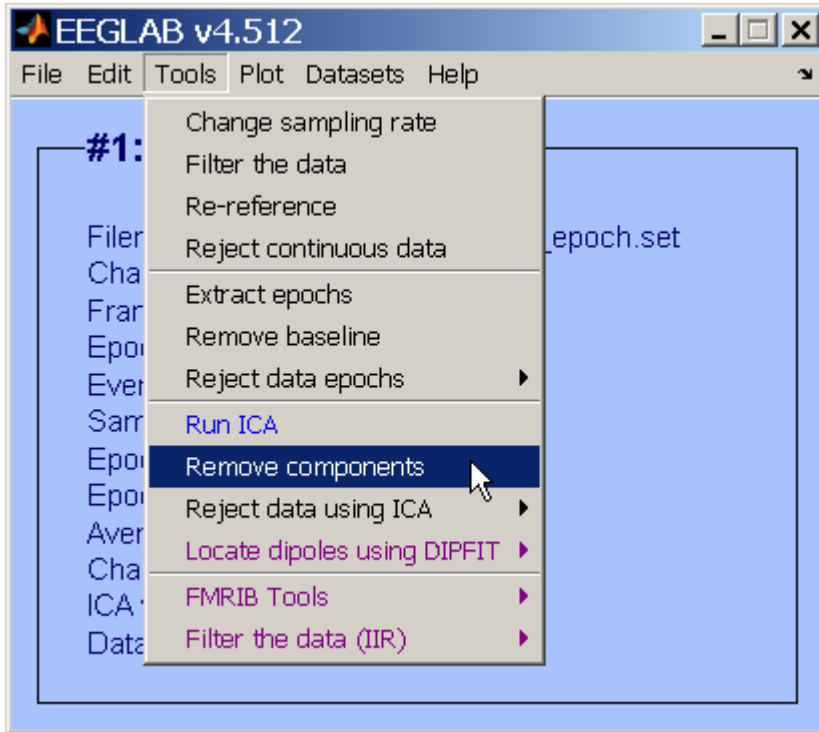
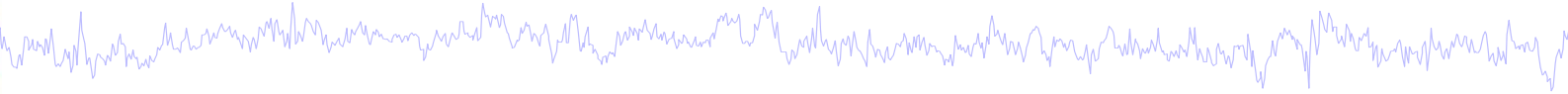
#1: (no)

Filename: .
Channels p
Frames pe
Epochs
Events
Sampling r
Epoch star
Epoch end
Average re
Channel lo
ICA weight
Dataset siz

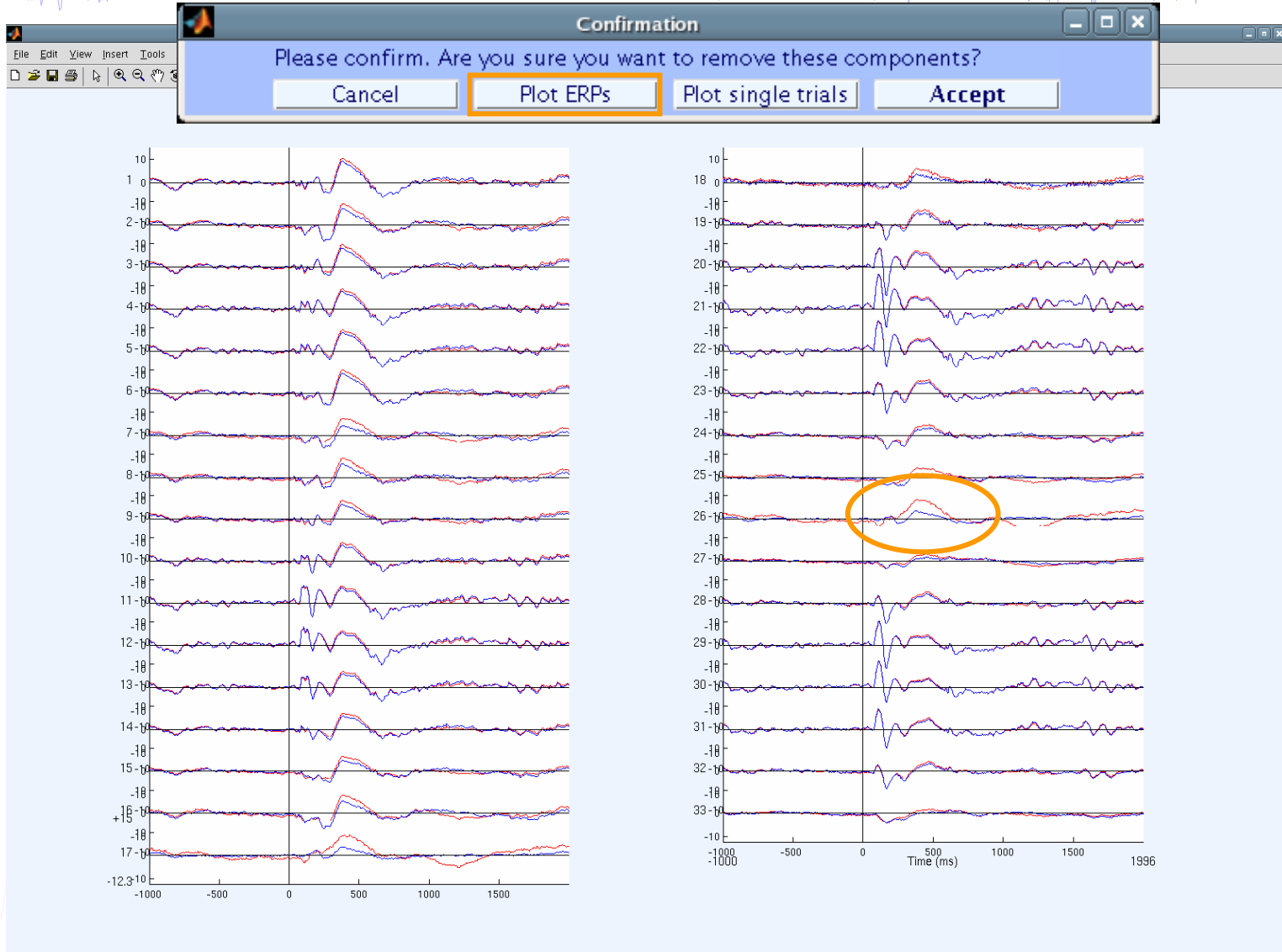
- Channel locations
- Channel data (scroll)
- Channel spectra and maps
- Channel properties
- Channel ERP image
- Channel ERPs
- ERP map series
- Sum/Compare ERPs
- Component activations (scroll)**
- Component spectra and maps
- Component maps**
- Component properties
- Component ERP image
- Component ERPs
- Sum/Compare comp. ERPs
- Data statistics
- Time-frequency transforms



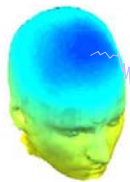
Eye blink correction



Eye blink correction



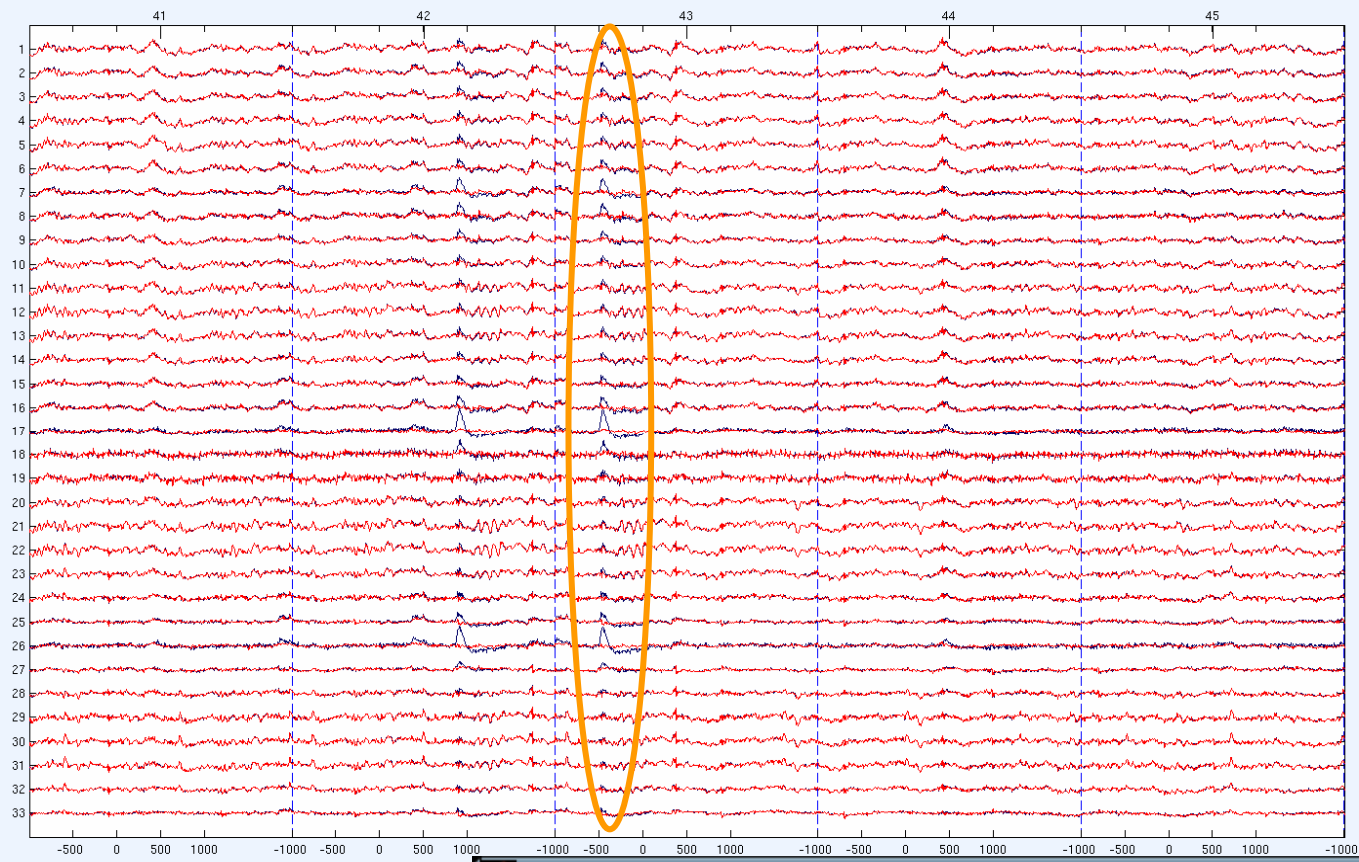
Eye blink correction



Confirmation

Please confirm. Are you sure you want to remove these components?

Cancel Plot ERPs Plot single trials Accept



CLOSE

<< < 41

Confirmation

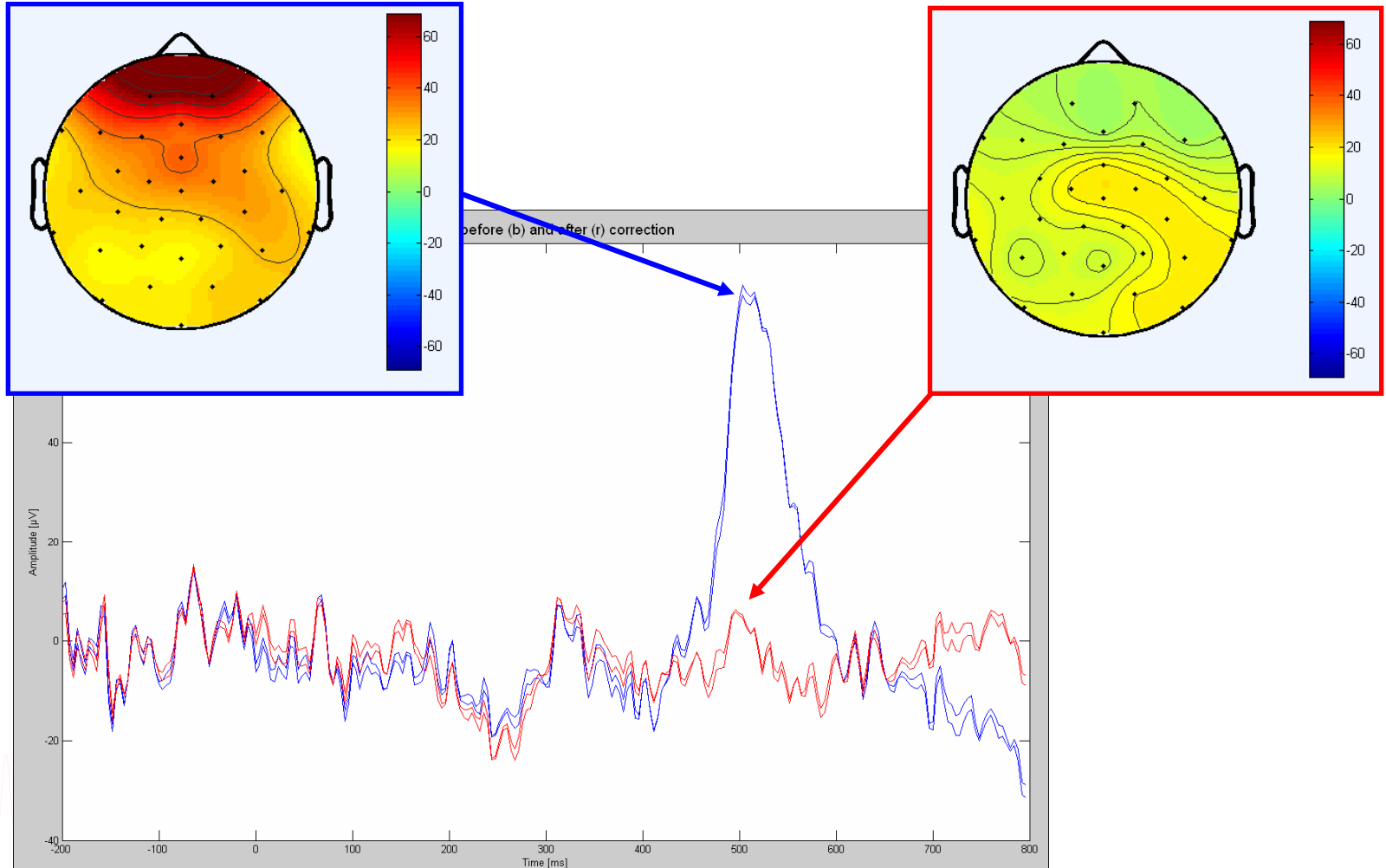
Please confirm. Are you sure you want to remove these components?

Cancel Plot ERPs Plot single trials Accept

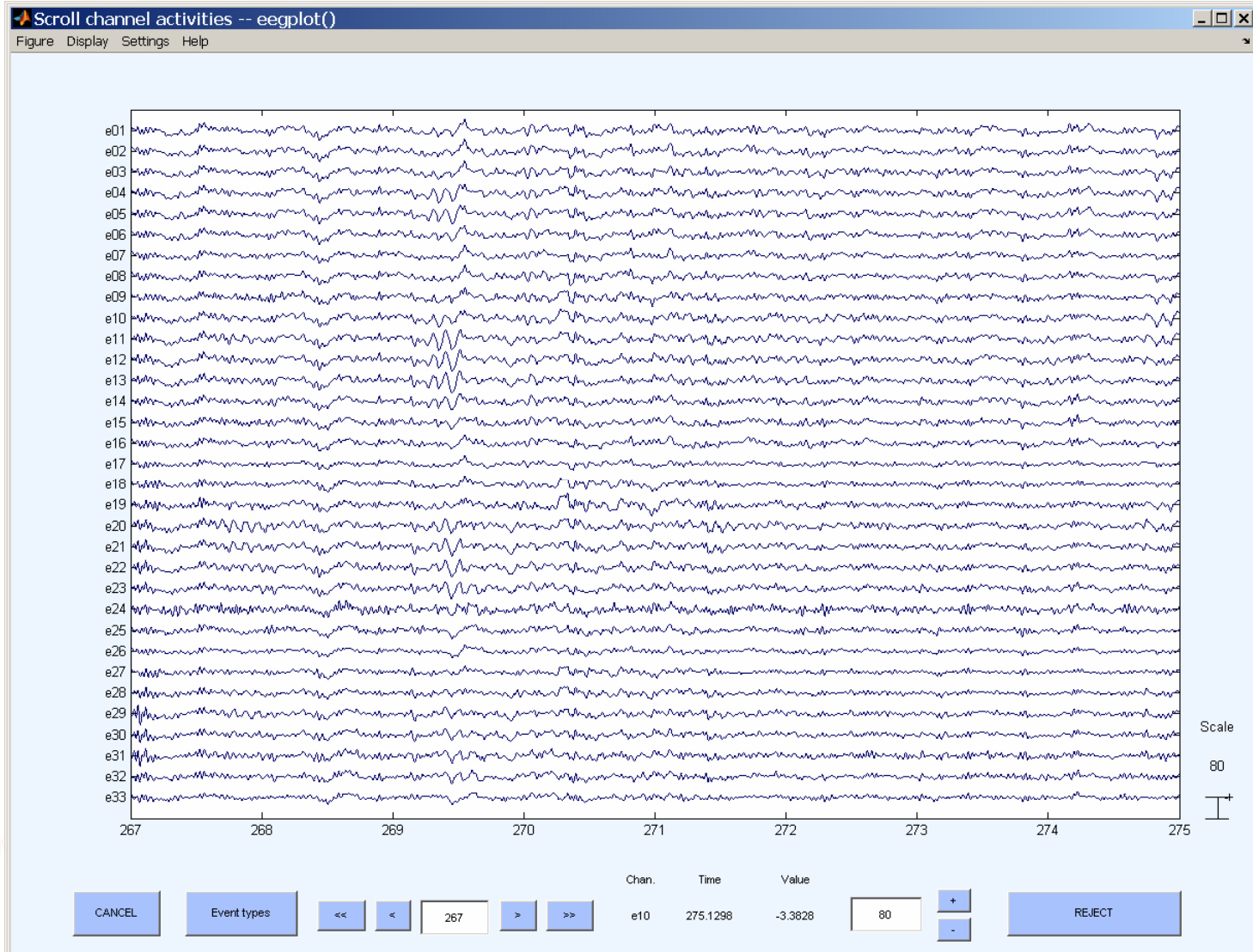
Eye blink correction



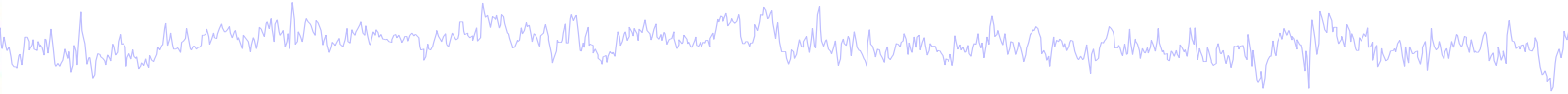
Trial 43: Fp1/2 before (b) and after (r) correction



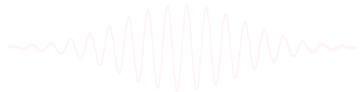
Task 2: Eye blink correction



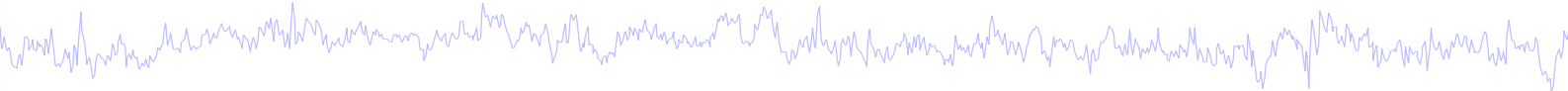
Task 2: Script an addition to EEG structure



- EEG structure can be extended to include new fields
 - Useful procedure to store information that may be accessed in future scripts
- Task: write a semi-automatic script to save eye blink component index as **EEG.blink**



Access functions by pop menus



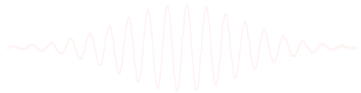
```
>> eeglab
```

```
% 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

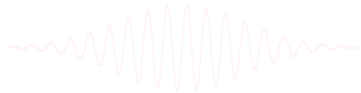


```
% faces2.m
%
infile = 'faces_3.set'; % define input .set file
inpath = 'C:\Documents and Settings\Administrator\...
        EEGLAB workshop\data\'; %input directory

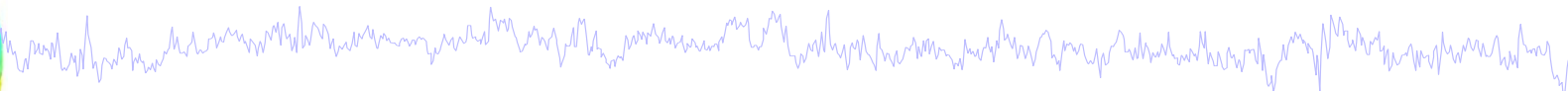
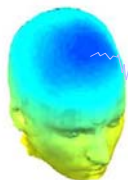
% start eeglab
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

% 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');
```



Task 2: Scripting ICA eye blink correction



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

```
% Enter eye blink indices manually
```

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

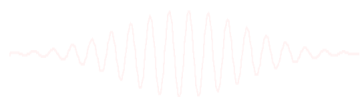
```
% save info in current data set
```

```
EEG = pop_saveset(EEG, infile, inpath);
```

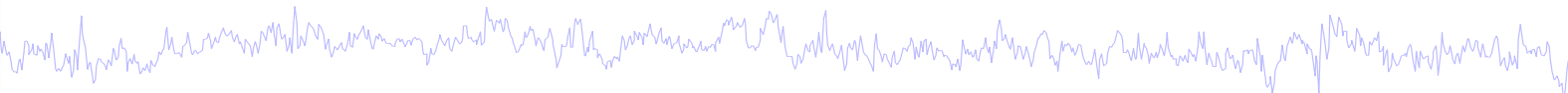
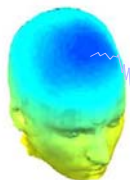
```
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
```

```
% update EEGLAB gui
```

```
eeglab redraw;
```



Exercise: Script it yourself



```
% 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);
```

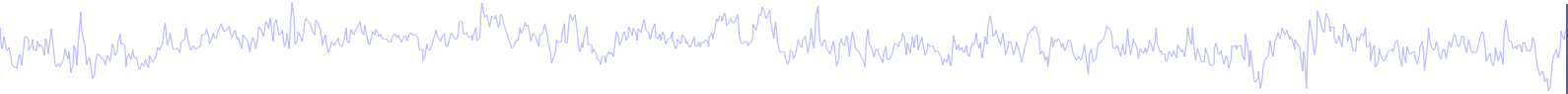
```
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
```

```
eeglab redraw;
```

For reference: example script saved as:

'...\EEGLAB workshop\Scripts\practicum_4.m'

Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Exercise...

Task 2

Eye-blink correction

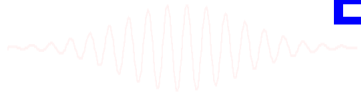
Create a new EEG field

Exercise...

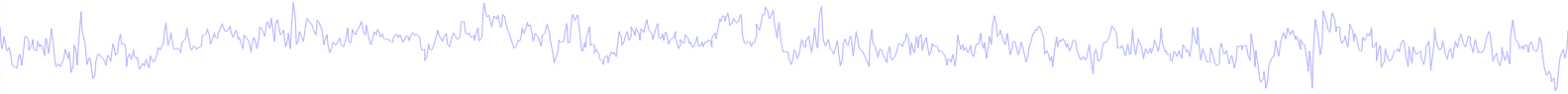
Task 3

Script to process multiple data sets

Exercise...



Task 3: Compare conditions within subject

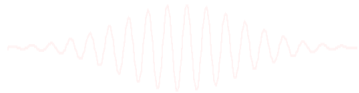


From the gui, perform the following operations:

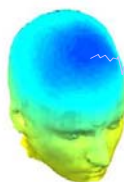
- 1) **Load** the dataset
- 2) **Subtract** ICA eye blink component activity
- 3) **Epoch** data (-200 to 800 ms) on 'object' and 'face' events
- 4) **Compare** the ERPs of both conditions
 - * Use: Plot → Sum/Compare ERPs
- 5) Use '**eegh**' to retrieve commandline functions
 - clear dataset

THEN: Write a script doing all this automatically

- ** Use a **for** loop and **variables** for maximum efficiency



Task 3: GUI functions (review)



EEGLAB v5.03

File Edit Tools Plot Study Datasets Help

- Import data
- Import epoch info
- Import event info
- Export
- Load existing dataset**
- Save current dataset(s)
- Save current dataset as
- Clear dataset(s)
- Create study
- Load existing study
- Save current study
- Save current study as
- Clear study
- Memory options
- Save history
- Quit

EEGLAB v5.03

File Edit Tools Plot Study Datasets Help

- Change sampling rate
- Filter the data
- Re-referencing
- Reject components
- Extract epochs
- Remove baseline
- Reject data epochs
- Run ICA
- Remove components**
- Reject data using ICA
- Locate dipoles using BESA
- Locate dipoles using DIPFIT 1.x
- Locate dipoles using DIPFIT 2.x
- Laplacian

Confirmation

Please confirm. Are you sure you want to remove these components?

Cancel Plot ERPs Plot single trials **Accept**

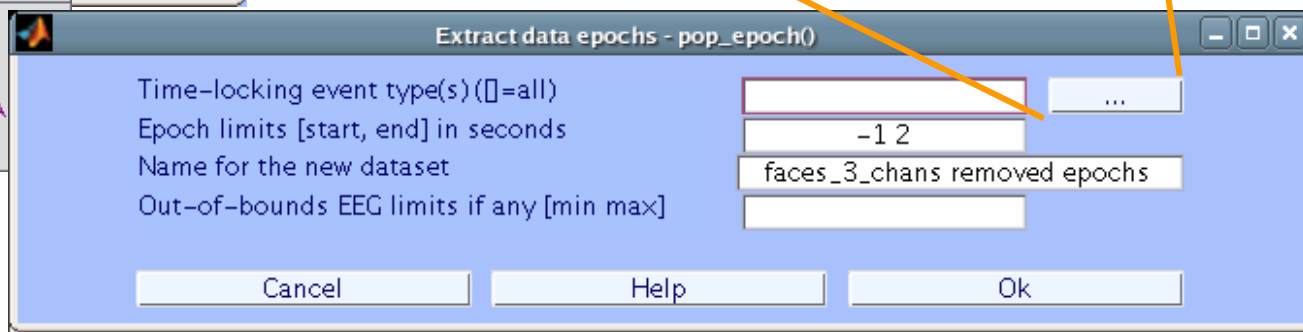
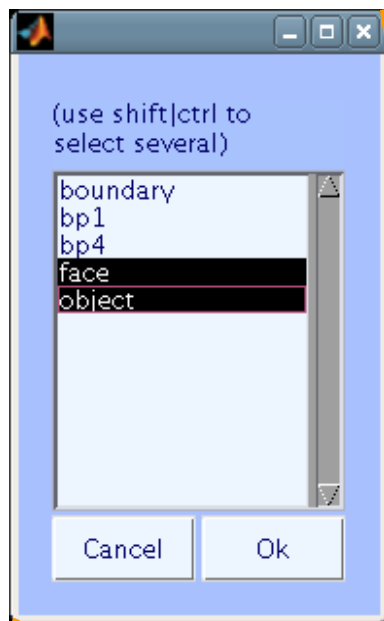
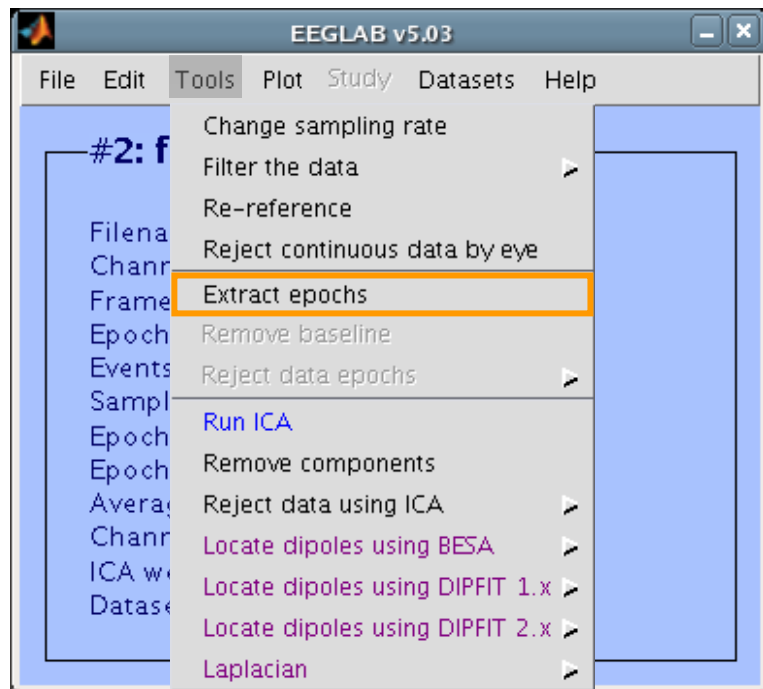
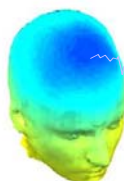
Remove components from data -- pop_subcomp()

Component(s) to remove from data:

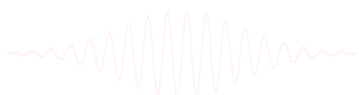
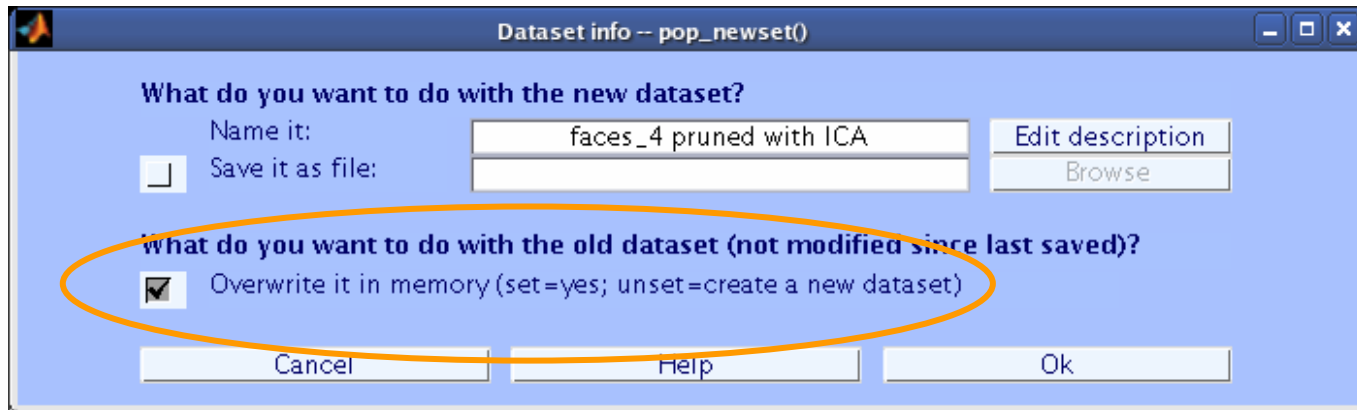
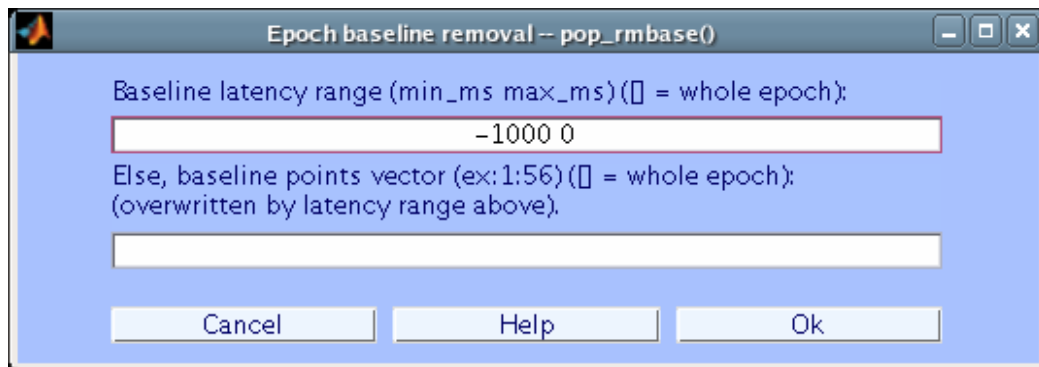
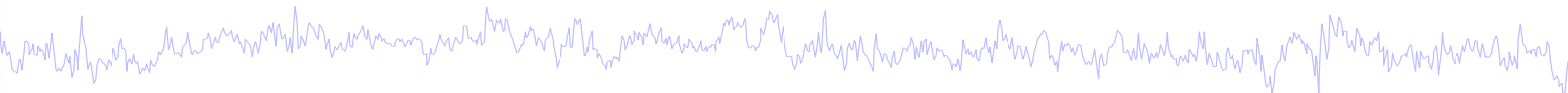
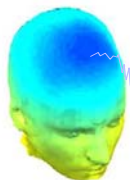
Component(s) to retain (overwrites "Component(s) to remove")

Cancel Help Ok

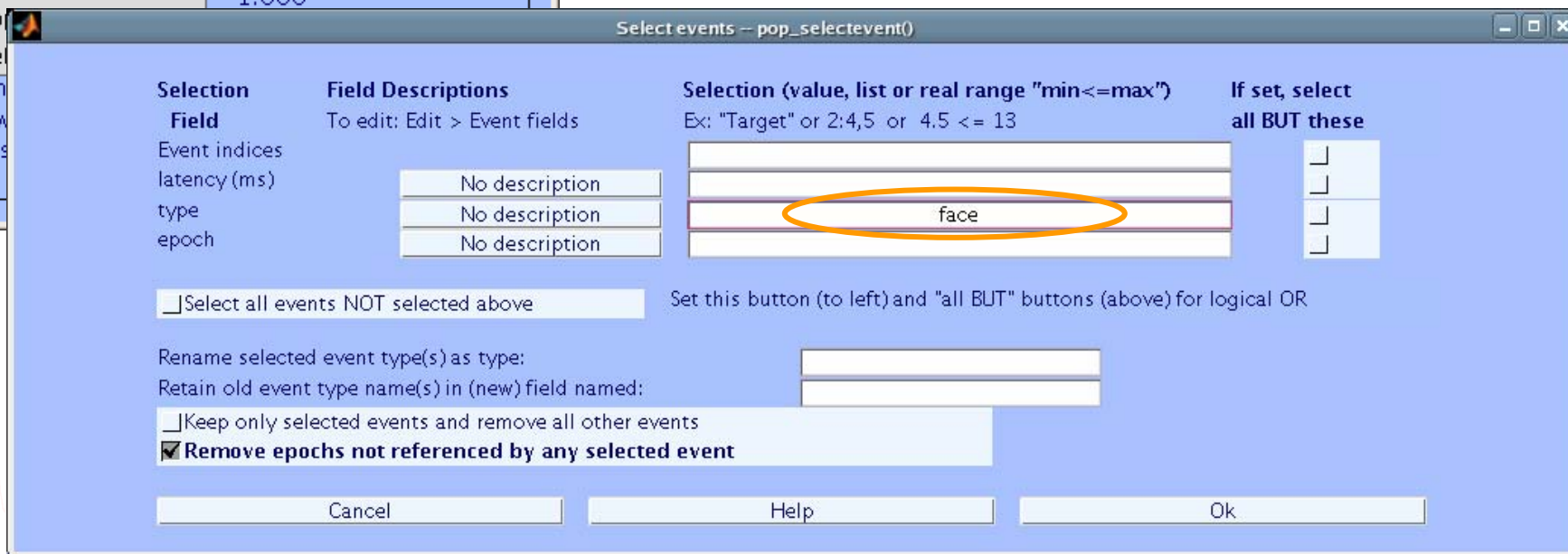
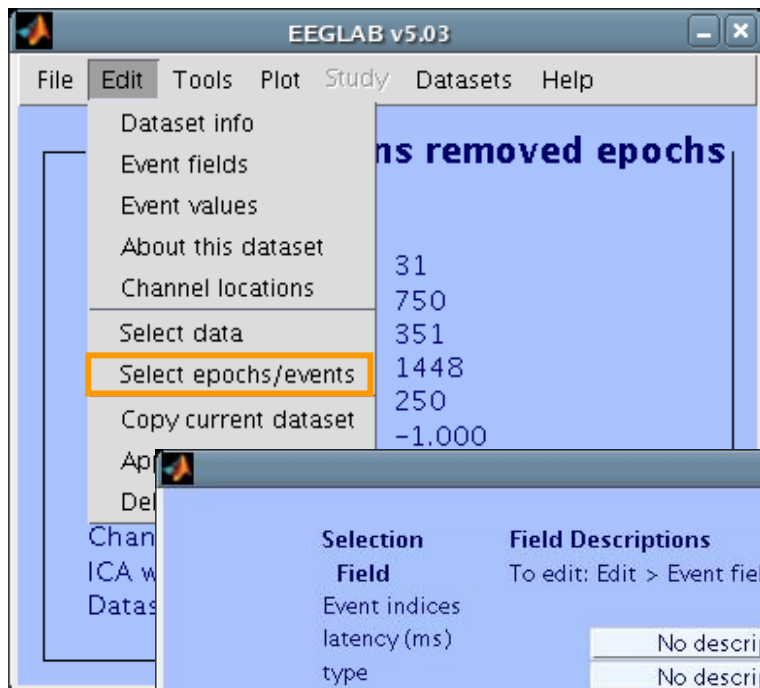
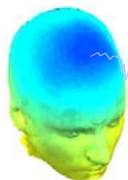
Task 3: GUI functions, cont'd



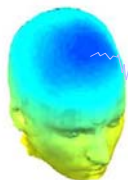
Task 3: GUI functions, cont'd



Task 3: GUI functions, cont'd



Task 3: Matlab window output



```
>> pop_loadset(): loading file ../faces_4.set ...  
eeg_checkset: recomputing the ICA activation matrix ...  
Creating a new ALLEEG dataset 1  
Done.
```

```
Computing projection ....  
Components removed  
eeg_checkset: recomputing the ICA activation matrix ...  
Done.
```

```
pop_epoch():364 epochs selected  
Epoching...  
pop_epoch():364 epochs generated  
eeg_checkset: recomputing the ICA activation matrix ...  
pop_epoch(): checking epochs for data discontinuity  
Done.
```

```
pop_rmbase(): Removing baseline...  
eeg_checkset: recomputing the ICA activation matrix ...  
Done.
```

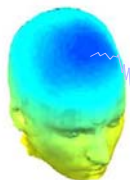
```
Removing 182 trial(s)...  
Pop_select: removing 359 unreferenced events  
Creating a new ALLEEG dataset 2  
Done.
```

```
Removing 182 trial(s)...  
Pop_select: removing 366 unreferenced events  
Creating a new ALLEEG dataset 3  
Done.
```

```
eegh
```



Task 3: eegh history output



```
>> eegh
```

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

```
EEG = pop_loadset( 'filename', 'faces_4.set', 'filepath', '...\data\');
```

```
[ALLEEG, EEG, CURRENTSET] = eeg_store( ALLEEG, EEG, 0 );
```

```
EEG = pop_subcomp( EEG, [4], 0);
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, 'setname',...  
    'faces_4 pruned with ICA', 'overwrite', 'on');
```

```
EEG = pop_epoch( EEG, { 'face' 'object' }, [-1 2], 'newname',...  
    'faces_4 pruned with ICA epochs', 'epochinfo', 'yes');
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, 'overwrite', 'on');
```

```
EEG = pop_rmbase( EEG, [-200 0]);
```

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

```
EEG = pop_selectevent( EEG, 'type',{ 'face' }, 'deleteevents','off',...  
    'deleteepochs', 'on');
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, 'setname', 'faces only epochs');
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 2, 'retrieve',1, 'study',0);
```

```
EEG = pop_selectevent( EEG, 'type',{ 'object' }, 'deleteevents', 'off',...  
    'deleteepochs', 'on');
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, 'setname', 'object only epochs');
```


Task 3: Create a script

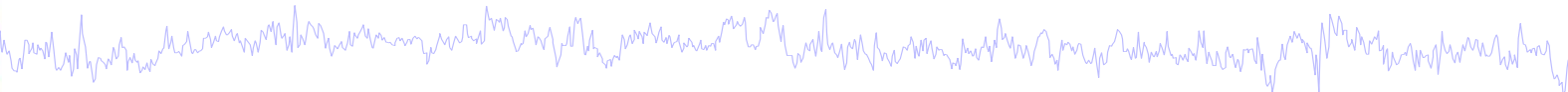
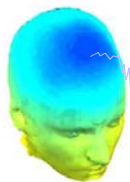


```
indata = 'faces_4.set'; % replace strings
inpath = '...\data\'; % with variables
% load data set
EEG = pop_loadset( indata, inpath);
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, 1);

% remove eye components for ERP plotting
% (replace IC index with your new EEG field):
EEG = pop_subcomp( EEG, EEG.blink , 0);
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, ...
    EEG, 1, 'overwrite', 'on');

% Epoch the data (replace events with a variable)
condition = {'object', 'face'};
EEG = pop_epoch( EEG, condition, [-0.2 0.8], ...
    'epochinfo', 'yes');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, ...
    'overwrite', 'on'); % still in ALLEEG(1)
EEG = pop_rmbase( EEG, [-200 0]);
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);
```

Task 3: Create a script, cont'd



% Create 'face'-only dataset

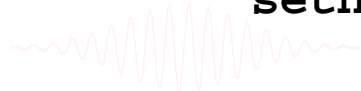
```
EEG = pop_selectevent( EEG, 'type', {condition{1}}, ...  
    'deleteevents', 'off', 'deleteepochs', 'on');  
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, ...  
    'setname', condition{1});
```

% Retrieve original epoched dataset

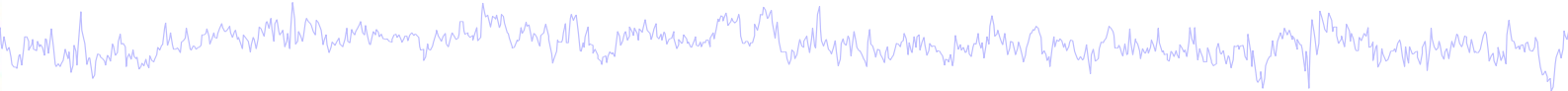
```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 2, ...  
    'retrieve', 1, 'study', 0);
```

% Create 'object'-only dataset

```
EEG = pop_selectevent( EEG, 'type', {condition{2}}, ...  
    'deleteevents', 'off', 'deleteepochs', 'on');  
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, ...  
    'setname', condition{2});
```



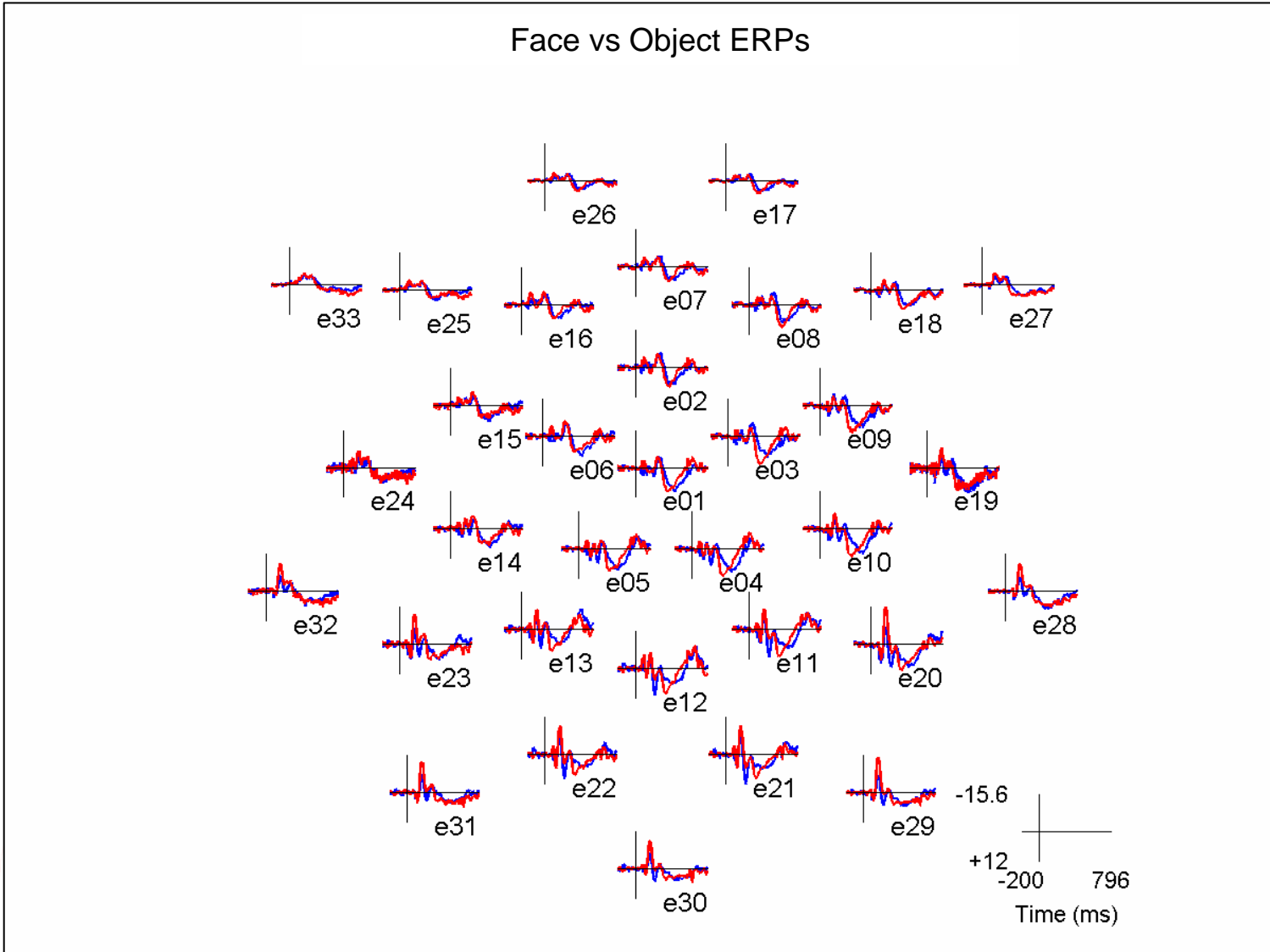
Task 3: Loop across data sets



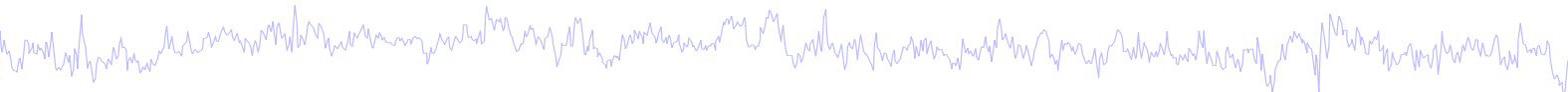
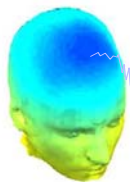
```
% create datasets for each condition
for i = 1:length(condition)
    EEG = pop_selectevent(EEG,'type',{condition{i}}, ...
        'deleteevents','off','deleteepochs','on');
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1,...
        'setname', condition{i});
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, i+1,...
        'retrieve',1, 'study',0); % Go back to ALLEEG(1)
end
% plot ERPs
pop_comperp( ALLEEG, 1, [2 3] ,[],'addavg', 'off',...
    'addstd','off', 'addall', 'on', 'diffavg','off',...
    'diffstd','off', 'tplotopt',{'title',...
    'Object vs Face ERPs', 'ydir', -1});

% save figure;
print('-dtiff', [inpath,'erp.tif']);
```

Task 3: Loop across data sets



Exercise: Script it yourself



```
% replace strings with variables
```

```
indata = 'faces_4.set';
```

```
inpath = '...\data\';
```

```
condition = {'object','face'};
```

```
%-----
```

```
EEG = pop_loadset( indata,inpath);
```

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

```
EEG = pop_subcomp( EEG, EEG.blink , 0);
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG,EEG, 0, 'overwrite', 'on');
```

```
EEG=pop_epoch( EEG,{condition},[-0.2 0.8],...
```

```
    'epochinfo','yes');
```

```
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1,'overwrite','on');
```

```
EEG = pop_rmbase( EEG, [-200 0]);
```

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

```
for i = 1:length(condition)
```

```
    EEG = pop_selectevent(EEG,'type',{condition{i}},'deleteevents','off','deleteepochs','on');
```

```
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1,CURRENTSET, 'setname', condition{i});
```

```
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, i+1,'retrieve',1, 'study',0);
```

```
end
```

```
pop_comperp( ALLEEG, 1, [2 3] ,[],'addavg', 'off',...
```

```
    'addstd','off', 'addall', 'on', 'diffavg','off',...
```

```
    'diffstd','off', 'tplotopt',{ 'title',...
```

```
    'Object vs Face ERPs', 'ydir', -1});
```

```
print('-dtiff', [inpath,'erp.tif']);
```

For reference: example script saved as:

'...\EEGLAB workshop\Scripts\practicum_4.m'