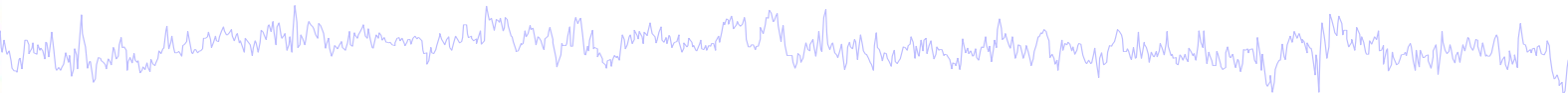
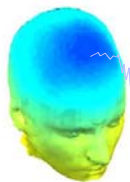


# More EEGLAB Scripting



## Task 1

- Load and epoch a continuous dataset
- Plot an ERP image of a component
- Script a command to 'value' sort ERP image

## Task 2

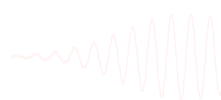
- Use *erpimage()* output to group ERPs

## Task 3

- Use *erpimage()* output to sort a new image

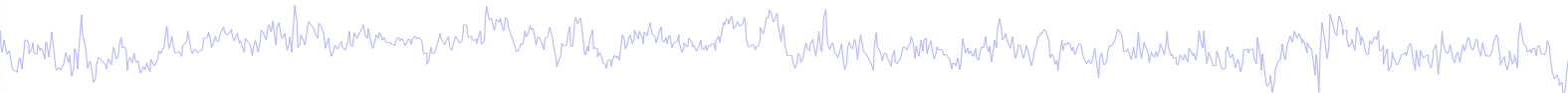
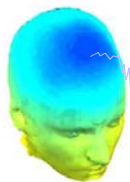
## Task 4

- Plot an ERP image with multiple subjects
- A word on component polarity



**Exercise...**

# More EEGLAB Scripting



## Task 1

- Load and epoch a continuous dataset
- Plot an ERP image of a component
- Script a command to 'value' sort ERP image

## Task 2

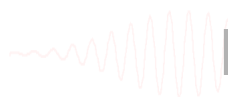
Use *erpimage()* output to group ERPs

## Task 3

Use *erpimage()* output to sort a new image

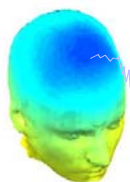
## Task 4

- Plot an ERP image with multiple subjects
- A word on component polarity



## Exercise...

# Task 1: Load dataset and epoch



The screenshot displays three overlapping windows of the EEGLAB v5.03 software. The top-left window shows the 'File' menu with 'Load existing dataset' highlighted in orange. The middle window shows the 'Tools' menu with 'Extract epochs' highlighted in orange. The bottom-right window shows the 'Study' window for a dataset named '#2: faces\_4 epochs' with the following parameters:

Filename:	none
Channels per frame:	33
Frames per epoch:	250
Epochs:	364
Events:	725
Sampling rate (Hz):	250
Epoch start (sec):	-0.200
Epoch end (sec):	0.796
Average reference:	No
Channel locations:	Yes
ICA weights:	Yes
Dataset size (Mb):	24.5

# Task 1: Reject noisy epochs using auto-rejection



Reject trials using component activity statistics - pop\_rejmenu0

Mark trials by appearance  Scroll Comp. Acts. Marked trials 0

**Find abnormal values**

Upper limit(s) (std. dev.)	<input type="text" value="25"/>	Lower limit(s) (std. dev.)	<input type="text" value="-25"/>
Start time(s) (ms)	<input type="text" value="-1000"/>	Ending time(s) (ms)	<input type="text" value="1996"/>
Component(s)	<input type="text" value="1:33"/>	Currently marked trials	<input type="text" value="0"/>

**Find abnormal trends**

Max slope (std. dev./epoch)	<input type="text" value="50"/>	R-squared limit (0 to 1)	<input type="text" value="0.3"/>
Component(s)	<input type="text" value="1:33"/>	Currently marked trials	<input type="text" value="0"/>

**Find improbable data**

Single-comp. limit (std. dev.)	<input type="text" value="5"/>	All comp. limit (std. dev.)	<input type="text" value="3"/>
Component(s)	<input type="text" value="13,14,15,17,18,19,24"/>	Currently marked trials	<input type="text" value="6"/>

**Find abnormal distributions**

Single-comp. limit (std. dev.)	<input type="text" value="20"/>	All comp. limit (std. dev.)	<input type="text" value="5"/>
Component(s)	<input type="text" value="1:33"/>	Currently marked trials	<input type="text" value="0"/>

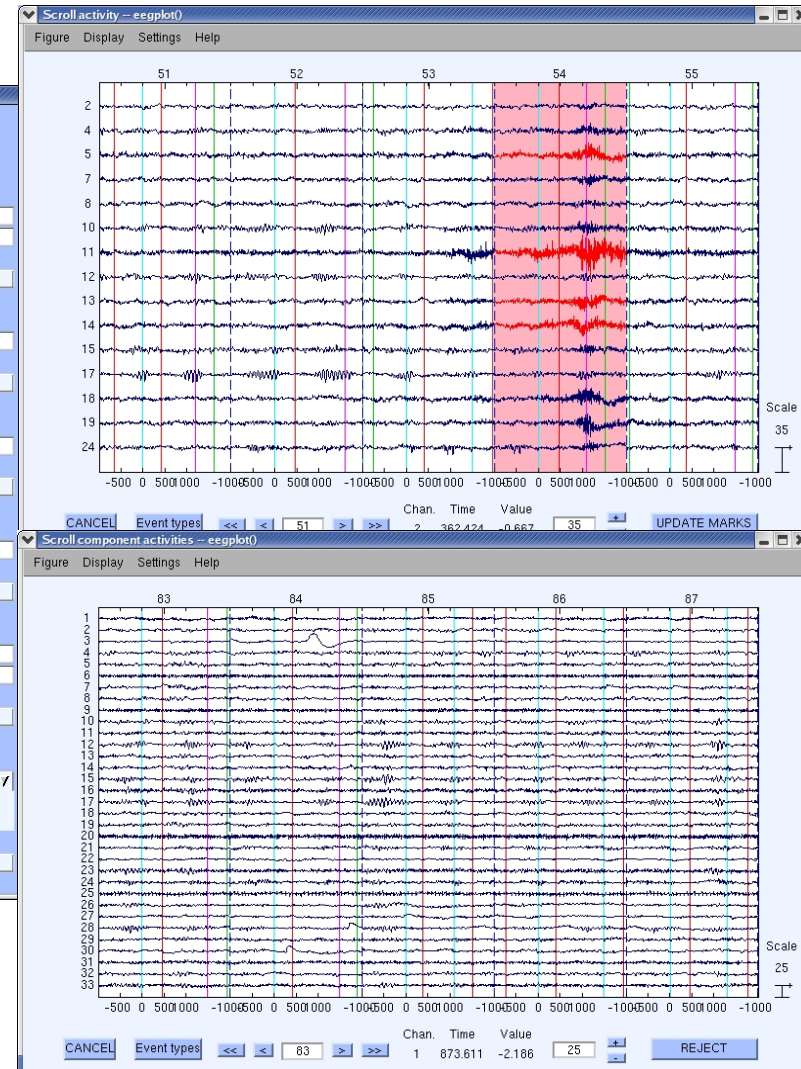
**Find abnormal spectra (efficient but slow)**

Upper limit(s) (dB)	<input type="text" value="25"/>	Lower limit(s) (dB)	<input type="text" value="-25"/>
Low frequency(s) (Hz)	<input type="text" value="0"/>	High frequency(s) (Hz)	<input type="text" value="50"/>
Component(s)	<input type="text" value="1:33"/>	Currently marked trials	<input type="text" value="0"/>

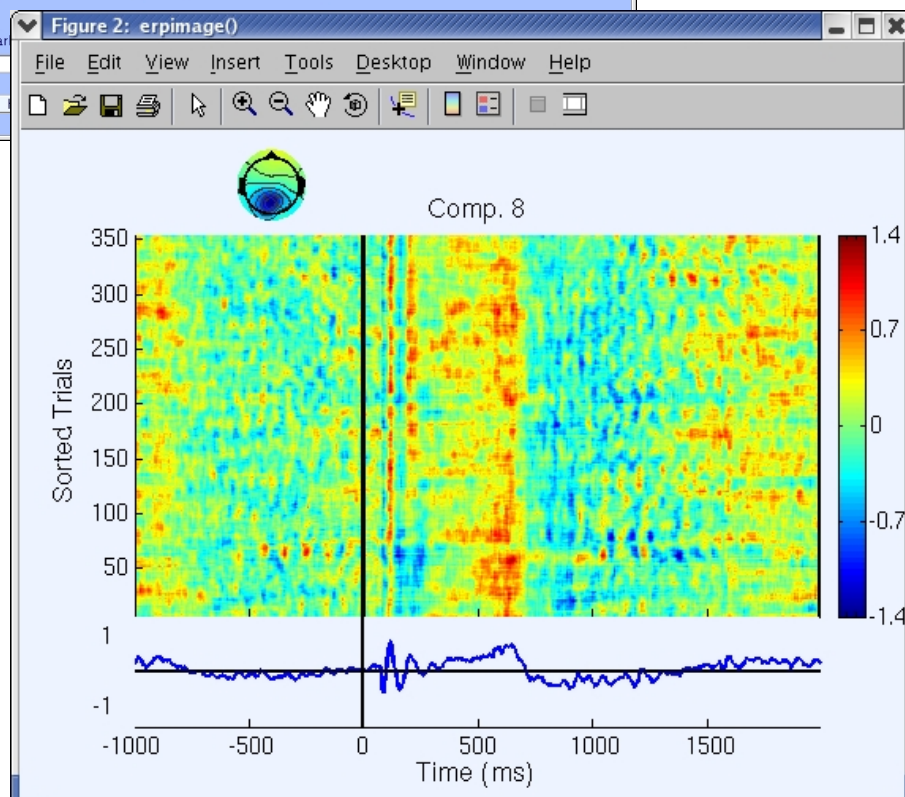
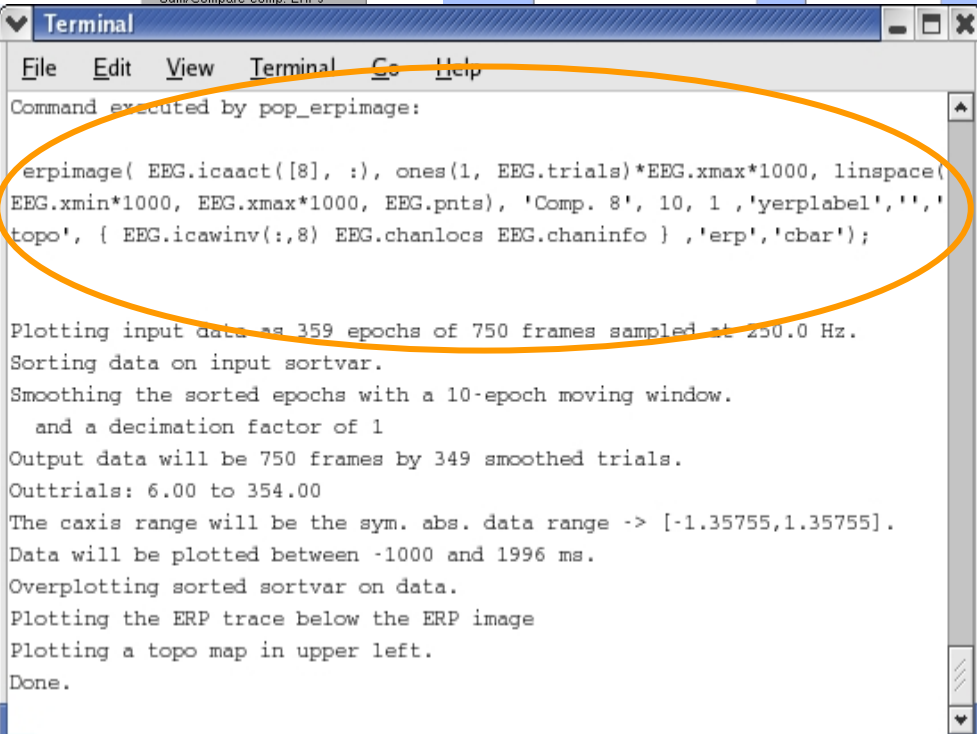
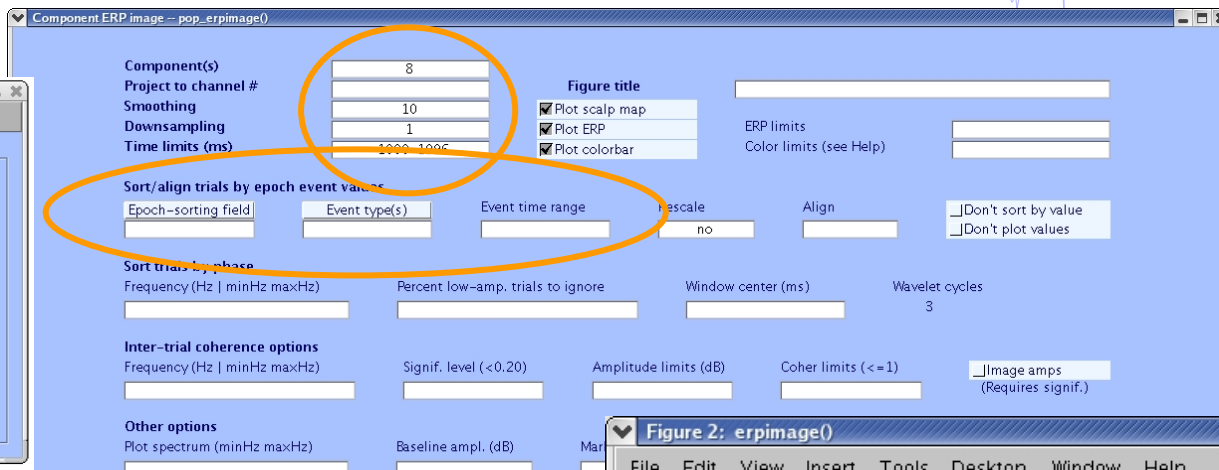
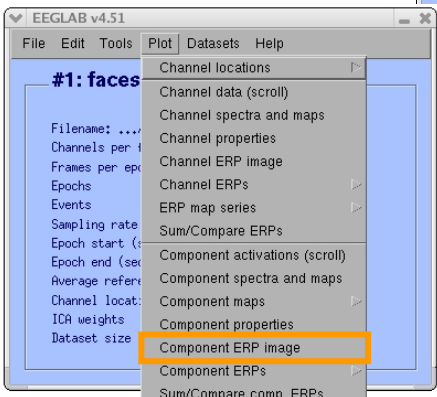
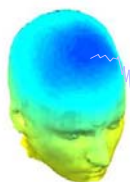
**Plotting options**

Show all trials marked for rejection by the measure selected above or checked below

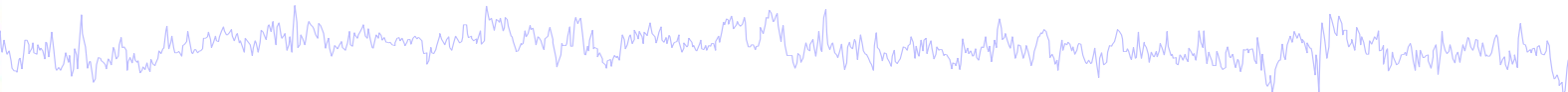
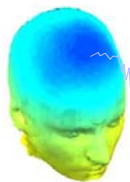
<input checked="" type="checkbox"/> Abnormal appearance	<input checked="" type="checkbox"/> Abnormal values	<input checked="" type="checkbox"/> Abnormal trends
<input checked="" type="checkbox"/> Improbable epochs	<input checked="" type="checkbox"/> Abnormal distributions	<input checked="" type="checkbox"/> Abnormal spectra



# Task 1: Plot ERP image from gui



# Task 1: Retrieve *erpimage()* call



Command executed by `pop_erpimage`:

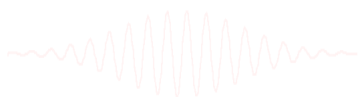
```
erpimage( EEG.icaact([8], :), ones(1, EEG.trials)*EEG.xmax*1000, ...  
         linspace(EEG.xmin*1000, EEG.xmax*1000, EEG.pnts), 'Comp. 8', 10, 1, ...  
         'yerplabel', '', 'topo', { EEG.icawinv(:,8) EEG.chanlocs EEG.chaninfo }, ...  
         'erp', 'cbar' );
```

Plotting options/scalp map

Plot ERP and colorbar

How do I adapt this command to include more *erpimage()* options?

```
>> help erpimage
```





# Task 1: Using help command to build script



```
Terminal
File Edit View Terminal Go Help
>> help erpimage
erpimage() image a collection of single-trial data epochs, optionally sorted on
and/or aligned to an input sorting variable and smoothed across trials
with a moving-average. (To return event-aligned data without plotting,
use eventlock()). Optionally sort trials on value, amplitude or phase
within a specified latency window. Optionally plot the ERP mean and
std. dev. and moving-window spectral amplitude and inter-trial coherence
at a selected or peak frequency. Click on individual figures parts to
examine them separately and zoom (using axcopy()).

Usage:
>> [outdata,outvar,outtrials,limits,axhndls,erp, ...
      amps,cohers,cohsig,ampsig,outamps,phsangls,phsamp,sortidx,ersig] ...
      = erpimage(data,sortvar,times,'title',avewidth,decimate,...
                 flag1,arg1,flag2,arg2,...);

Necessary inputs:
data - [vector or matrix] Single-channel input data to image.
      Formats (1,frames*trials) or (frames,trials)

Optional ordered inputs (with defaults):
sortvar - [vector | []] Variable to sort epochs on (length(sortvar) = nepochs)
          Example: sortvar may be subject response time in each epoch (in ms)
          (default|[]: plot in input order)
times - [vector | []] of latencies (ms) (length(times) = frames)
        Else [startms ntimes srate] Give start latency (ms), time points
        i.e. frames) per epoch, sampling rate (Hz), (default|[]: 0:nframes-1)
'title' - [string] Plot title (default: none)
avewidth - Number of trials to smooth with a moving-average (may be non-integer)
          (default|0->1)
decimate - Factor to decimate ntrials out by (may be non-integer) (default|0->1)
          If this is large ( > sqrt(num. trials)), output this many trials.

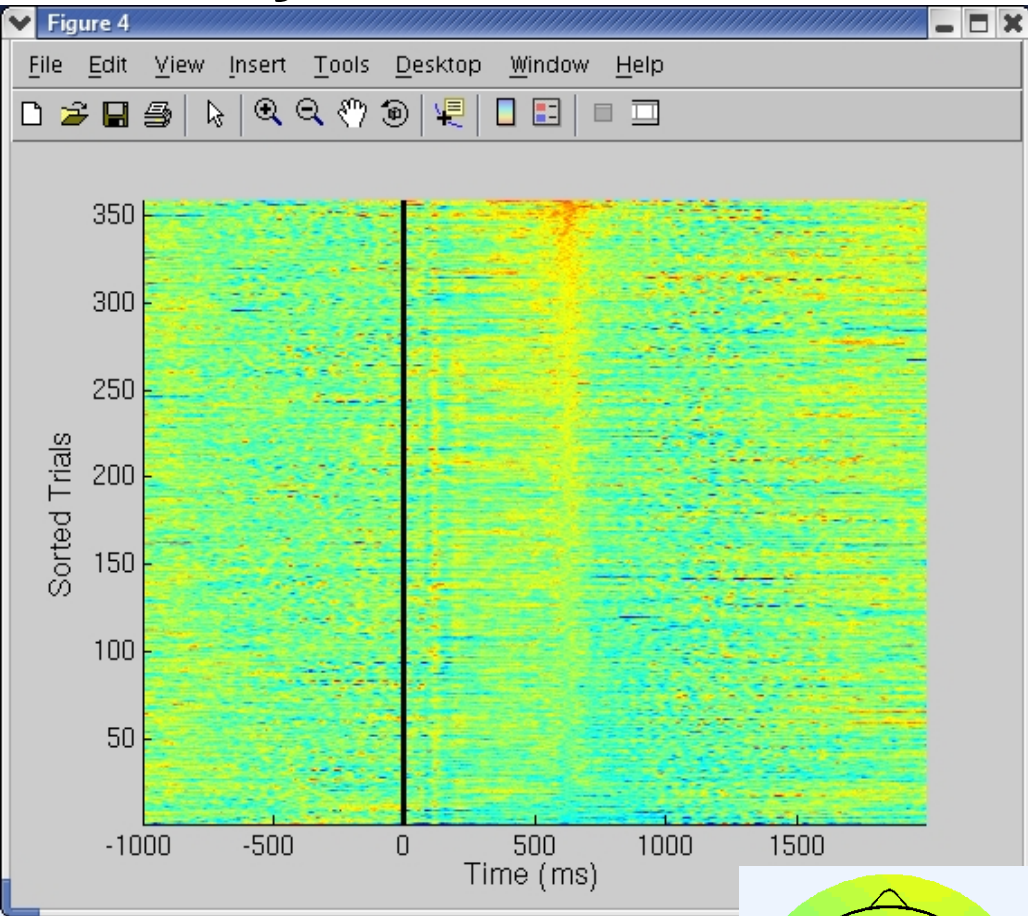
Unordered options ('keyword',argument pairs):

Optionally realign data epochs:
'align' - [latency] Time-lock data to sortvar. Plot sortvar as at latency (ms)
          If latency == Inf, plot at sortvar median (default: no align)
```

# Task 1: ERP image sorted by activation value

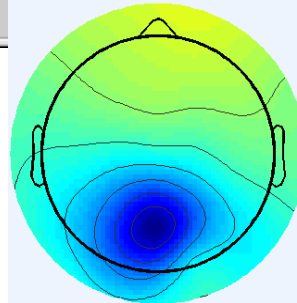
```
[outdata,outvar,outtrials,limits,axhndls,...  
erp,amps,cohers,cohsig,ampsig,outamps,...  
phsangls,phsamp,sortidx,ersig] = ...  
erpimage(data,sortvar,times,'title',...  
avewidth,decimate,flag1,arg1,...);
```

```
%%%%% VARIABLES %%%%%%%%%%%%%%%  
comp1 = 8; % Comp number to plot  
data = squeeze(EEG.icaact(comp1,:,:));  
sortvar = [];  
startms = 580; % ms  
endms = 620; % ms  
smoothby = 1;  
%%%%% PLOT ERPIMAGE %%%%%%%%%%%%%%%  
figure;
```



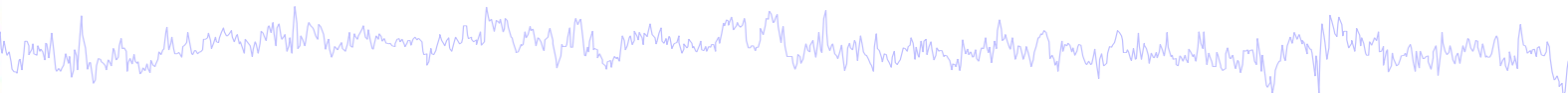
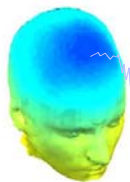
```
[outdata,outvar, outtrials,limits, axhndls, erp, ...  
amps, cohers, cohsig, ampsig, outamps, phsangls, ...  
phsamps, sortidx, ersig] ...
```

```
    = erpimage( data, sortvar, EEG.times,  
               '\', smoothby, 1, 'valsort', [startms endms]);
```





# More EEGLAB Scripting



## Task 1

Load and epoch a continuous dataset  
Plot an ERP image of a component  
Script a command to 'value' sort ERP image

## Task 2

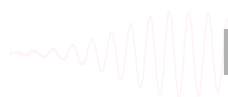
Use *erpimage()* output to group ERPs

## Task 3

Use *erpimage()* output to sort a new image

## Task 4

Plot an ERP image with multiple subjects  
-A word on component polarity



## Exercise...

# Task 2: Separate trials using `erpimage()` output

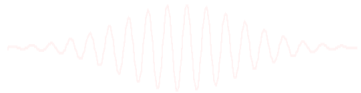


**Objective:** Group and plot trials with low, middle and high amplitude ERPs

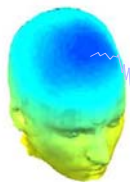
```
lotrials = outdata(:, 1:120);  
medtrials = outdata(:,121:240);  
hitrials = outdata(:,241:end);
```

```
figure;  
subplot( 3, 1, 1)  
plot(EEG.times, mean(lotrials,2), 'b-');  
  
subplot( 3, 1, 2)  
plot(EEG.times, mean(medtrials,2),'g-');  
  
subplot( 3, 1, 3)  
plot(EEG.times, mean(hitrials,2), 'r-');
```

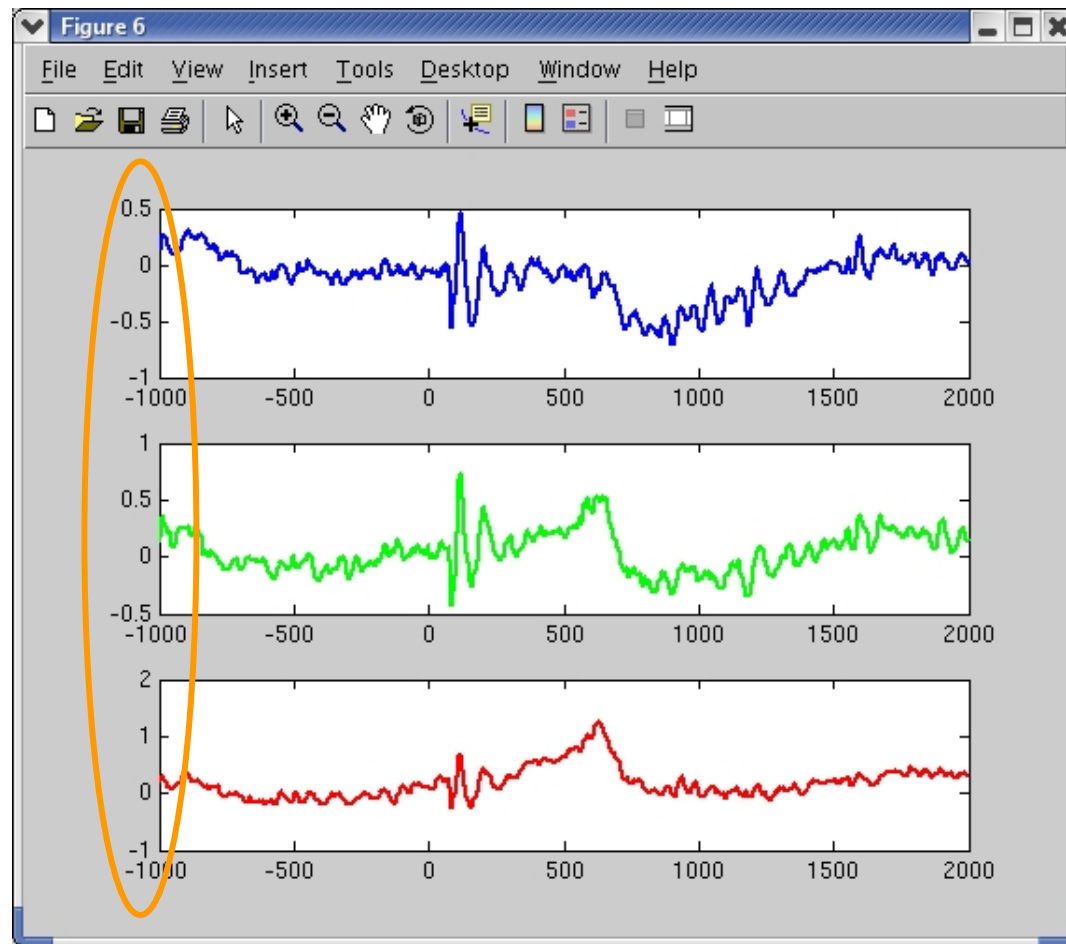
```
Terminal  
File Edit View Terminal Go He  
>>  
>> whos outdata outvar outtrials  
Name          Size  
outdata       750x359  
outtrials     1x359  
outvar        1x359  
Grand total is 269968 elements using  
>>
```



# Task 2: Plotting ERPs in Matlab



```
lotrials = outdata(:, 1:120);  
medtrials = outdata(:,121:240);  
hitrials = outdata(:,241:end);  
  
figure;  
subplot( 3, 1, 1)  
plot(EEG.times, mean(lotrials,2), 'b-');  
  
subplot( 3, 1, 2)  
plot(EEG.times, mean(medtrials,2), 'g-');  
  
subplot( 3, 1, 3)  
plot(EEG.times, mean(hitrials,2), 'r-');
```



Find maximum range

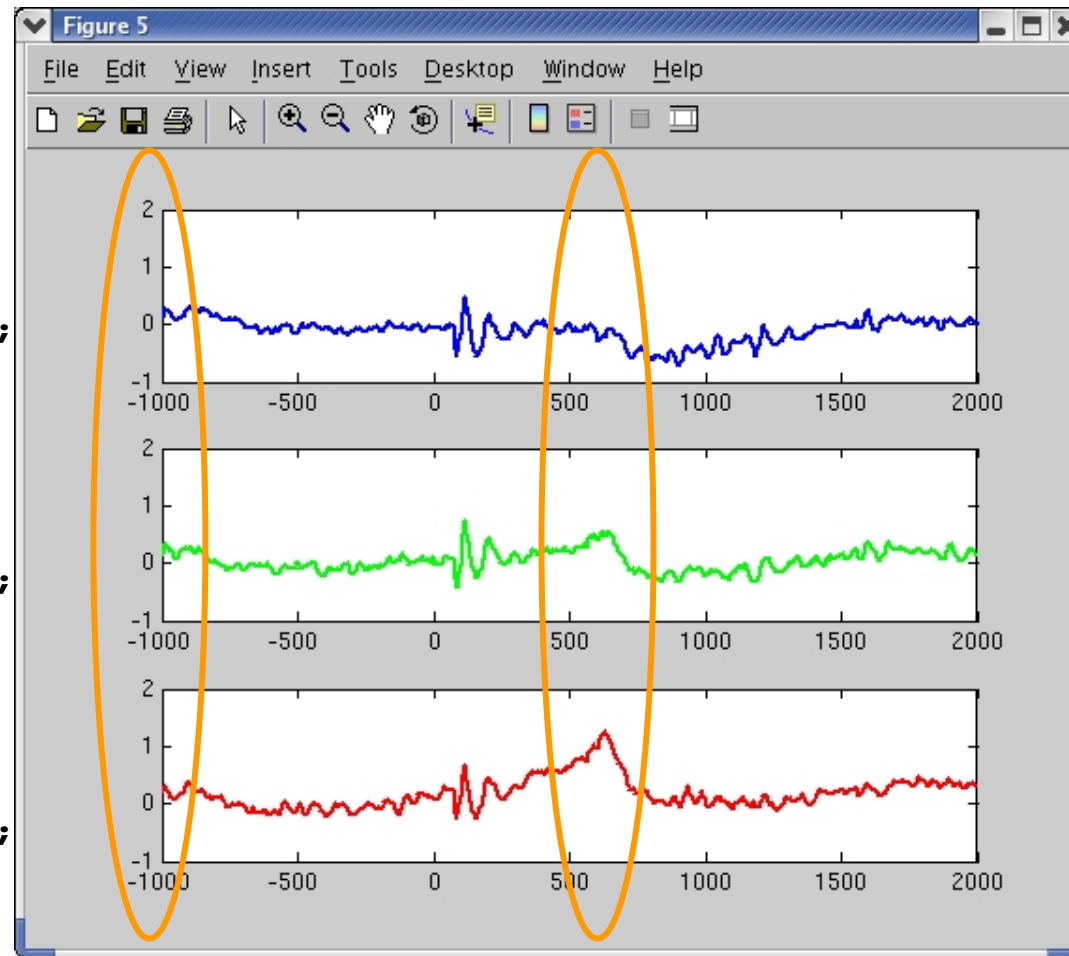
# Task 2: More Matlab plotting...



```
lotrials = outdata(:, 1:120);
medtrials = outdata(:,121:240);
hitrials = outdata(:,241:end);
figure;
subplot( 3, 1, 1)
plot(EEG.times, mean(lotrials,2), 'b-');
set(gca,'ylim',[-1 2]);

subplot( 3, 1, 2)
plot(EEG.times, mean(medtrials,2),'g-');
set(gca,'ylim',[-1 2]);

subplot( 3, 1, 3)
plot(EEG.times, mean(hitrials,2), 'r-');
set(gca,'ylim',[-1 2]);
```



# Task 2: More Matlab plotting...

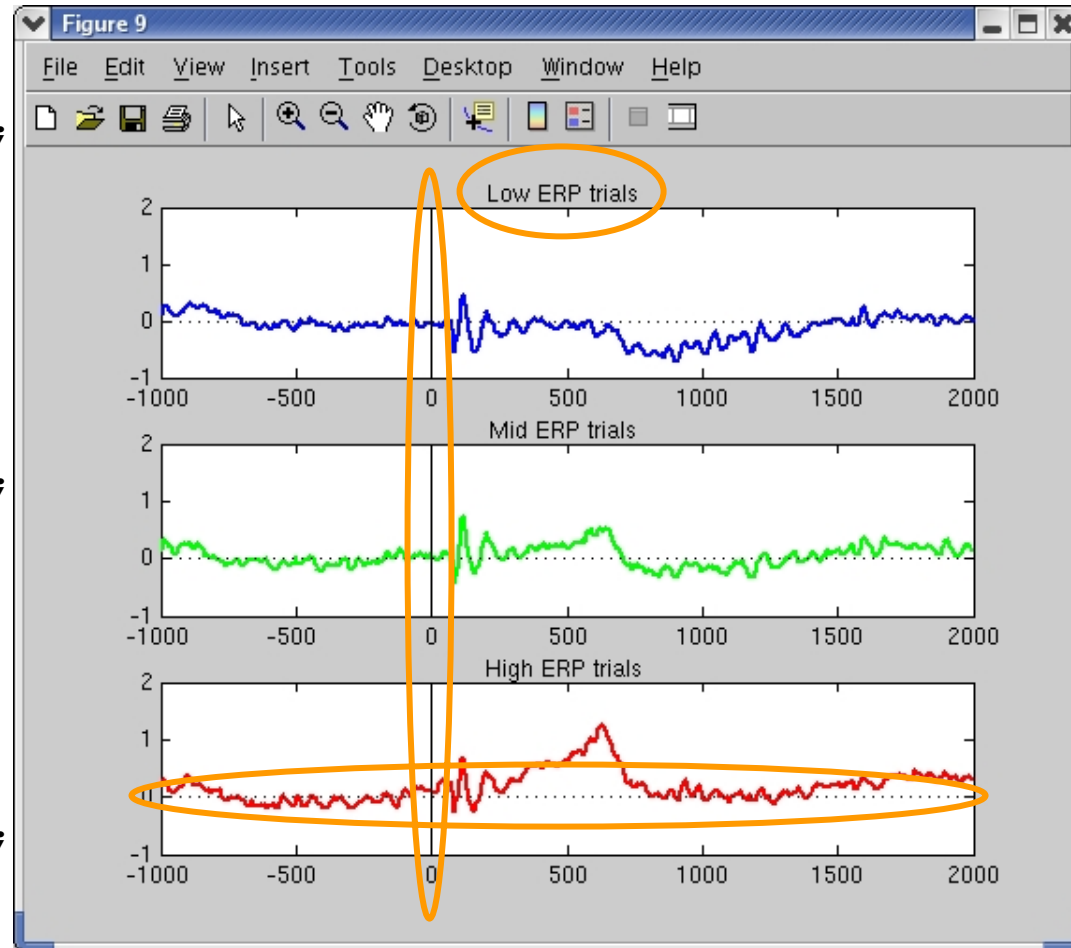
```
lotrials = outdata(:, 1:120);
medtrials = outdata(:,121:240);
hitrials = outdata(:,241:end);

figure;

subplot( 3, 1, 1); hold on;
plot(EEG.times, mean(lotrials,2), 'b-');
set(gca,'ylim',[-1 2]);
plot([-1000 2000],[0 0], 'k:');
plot([0 0],[-1 2],'k-');
title('Low ERP trials');

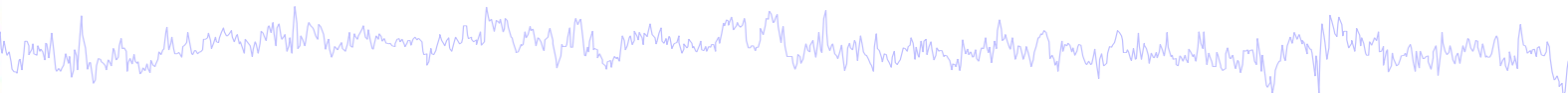
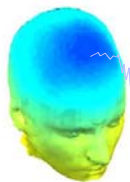
subplot( 3, 1, 2); hold on;
plot(EEG.times, mean(medtrials,2),'g-');
set(gca,'ylim',[-1 2]);
plot([-1000 2000],[0 0], 'k:');
plot([0 0],[-1 2],'k-');
title('Mid ERP trials');

subplot( 3, 1, 3); hold on;
plot(EEG.times, mean(hitrials,2), 'r-');
set(gca,'ylim',[-1 2]);
plot([-1000 2000],[0 0], 'k:');
plot([0 0],[-1 2],'k-');
title('High ERP trials');
```





# More EEGLAB Scripting



## Task 1

Load and epoch a continuous dataset  
Plot an ERP image of a component  
Script a command to 'value' sort ERP image

## Task 2

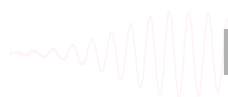
Use *erpimage()* output to group ERPs

## Task 3

Use *erpimage()* output to sort a new image

## Task 4

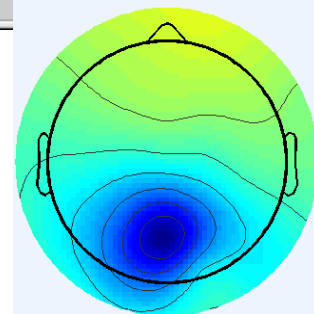
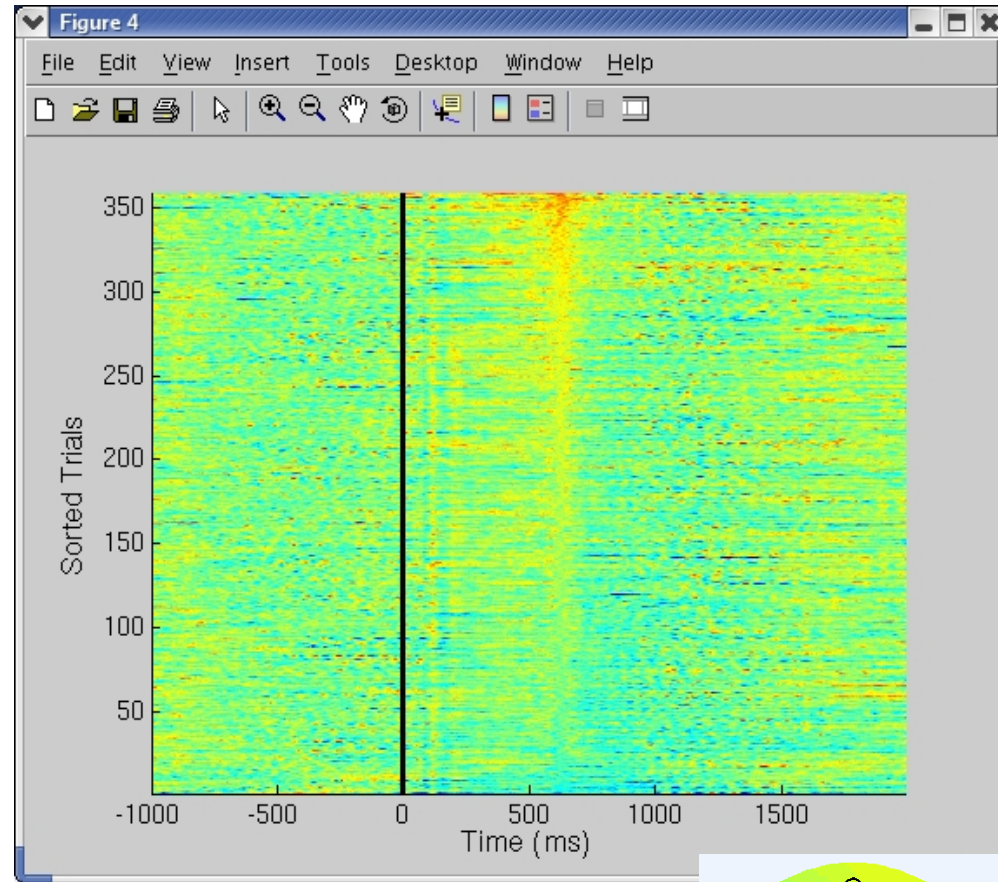
Plot an ERP image with multiple subjects  
-A word on component polarity



## Exercise...

# Task 3: Use sort index to sort a *new* ERP image

```
%%%%% VARIABLES %%%%%%%%%%%%%%%  
compl = 8;  
data = squeeze(EEG.icaact(compl, :, :));  
sortvar = [];  
startms = 580;  
endms = 620;  
smoothby = 1;  
  
%%%%% PLOT ERPIMAGE %%%%%%%%%%%%%%%  
figure;  
[outdata, outvar, outtrials, limits, axhndls, erp, ...  
amps, cohers, cohsig, ampsig, outamps, phsangls, phsamps ...  
sortidx, erpsig] = erpimage( data, sortvar, EEG.times, ...  
'', smoothby, 1, 'valsort', [startms endms]);
```



# Task 3: Index definition



```
>> my_numbers = [101,102,103,104,105,106,107,108,109,110];
```

```
my_numbers =
```

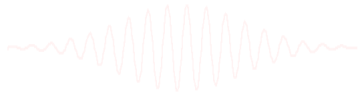
```
101 102 103 104 105 106 107 108 109 110
```

```
>> new_order = [8,2,5,1,10,9,4,6,3,7]; % analogous to sortidx
```

```
>> my_numbers(new_order)
```

```
ans =
```

```
108 102 105 ... 110 109 104 106 103 107
```



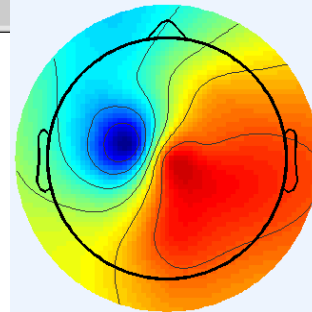
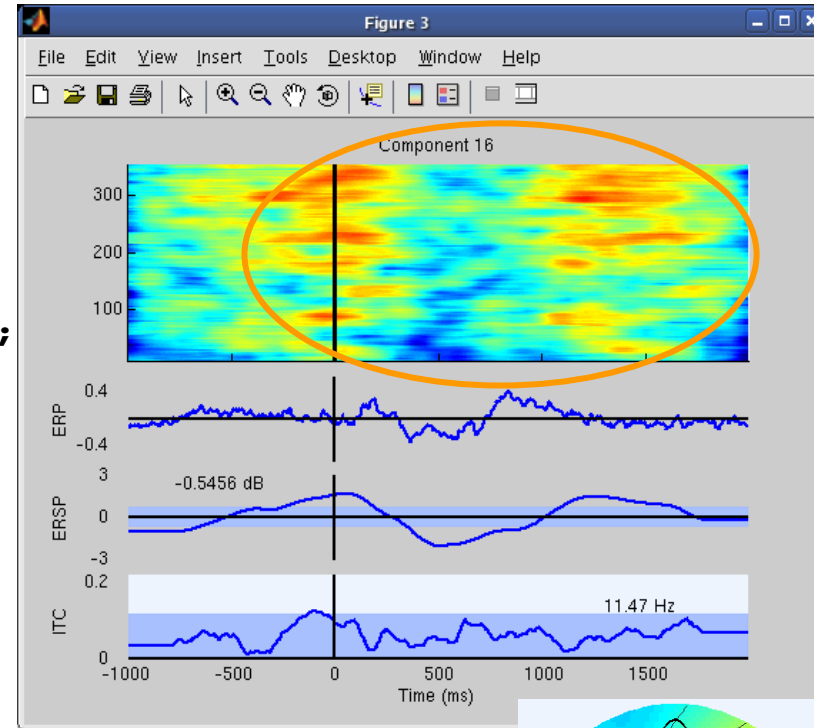
# Task 3: Using sort index to sort a *new* ERP image

**Objective:** Use sort order (*sortidx*) from 'valsort' of *comp1* to create a new ERP image of another component with the same sort order

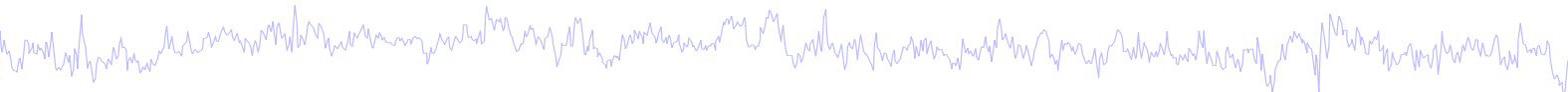
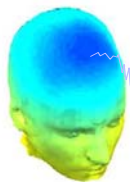
```
%%%%% VARIABLES %%%%%%%%%%%
comp1 = 8;
data = squeeze(EEG.icaact(comp1, :, :));
sortvar = [];
startms = 580;
endms = 620;
smoothby = 1;
%%%%% 1st ERPIMAGE %%%%%%%%%%%
figure;
[outdata, outvar, outtrials, limits, axhndls, erp, ...
amps, cohers, cohsig, ampsig, outamps, phsangls, ...
phsamps, sortidx, erpsig] = erpimage(data, sortvar, ...
EEG.times, '', smoothby, 1, 'valsort', [startms endms]);
```

```
%%%%% 2nd ERPIMAGE %%%%%%%%%%%
%%%%% Sort by previous sort order %%%%%%%%%%%
comp2 = 16;
data2 = squeeze(EEG.icaact(comp2, :, sortidx));
minfrq = 9; % specify freq range for
maxfrq = 12; % amplitude plot
smoothby = 20;
```

```
figure;
[outdata, outvar, outtrials, limits, axhndls, erp, amps, ...
cohers, cohsig, ampsig, outamps, phsangls, phsamps, sortidx2, erpsig] ...
= erpimage(data2, sortvar, EEG.times, ['Component ', int2str(comp2)], ...
smoothby, 1, 'coher', [minfrq maxfrq .01], 'plotamps');
```



# More EEGLAB Scripting



## Task 1

- Load and epoch a continuous dataset
- Plot an ERP image of a component
- Script a command to 'value' sort ERP image

## Task 2

- Use *erpimage()* output to group ERPs

## Task 3

- Use *erpimage()* output to sort a new image

## Task 4

- Plot an ERP image with multiple subjects
- A word on component polarity

 **Exercise...**



# Task 4: Collect and plot data across subjects

```
% Load two epoched datasets, from two different subjects
```

```
%%%%% VARIABLES %%%%%%%%%%%%%%%  
% Create a cell array with comps of interest  
plotcomps{1} = [10]; % subject 1  
plotcomps{2} = [6]; % subject 2  
sortvar = [];  
smoothby = 10;
```

```
%%%%% COLLECT DATA %%%%%%%%%%%%%%
```

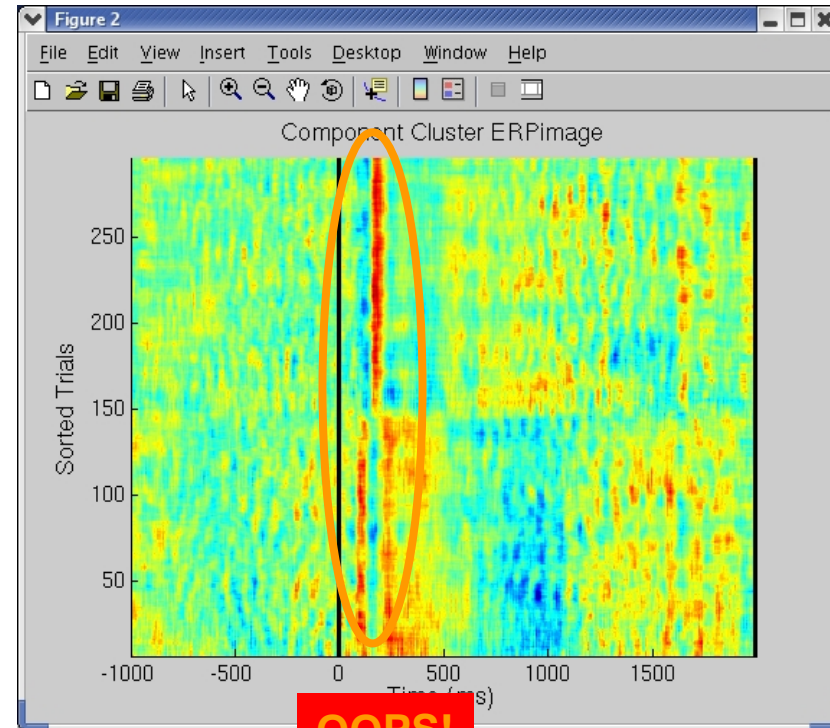
```
data = [];  
for subj = 1:length(plotcomps)  
    [EEG ALLEEG CURRENTSET] = eeg_retrieve(ALLEEG,subj);  
    scalefac = sqrt(mean(EEG.icawinv(:,plotcomps{subj}).^2)); % root mean square (RMS)  
    data = [data squeeze(EEG.icaact(plotcomps{subj},:,:))*scalefac];  
end;
```

```
%%%%% PLOT ERPIMAGE %%%%%%%%%%%%%%
```

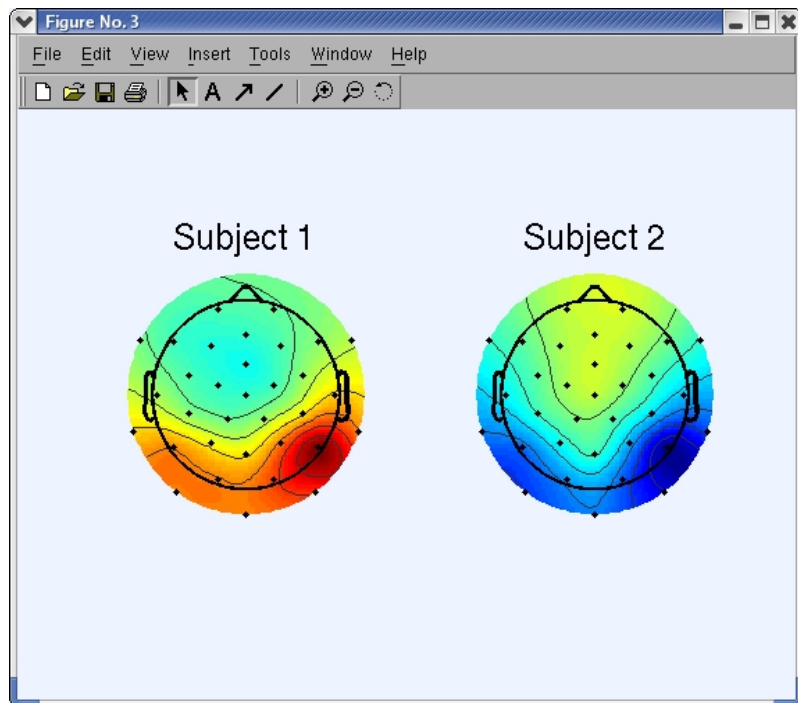
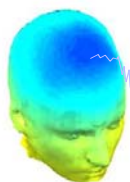
```
figure;
```

```
[outdata,outvar, outtrials,limits, axhndls, erp, amps, cohers, cohsig, ampsig, outamps,  
~ phsangls, phsamps, sortidx, erpsig] ...
```

```
= erpimage( data, sortvar, EEG.times, 'Component Cluster ERPimage', smoothby , 1);
```



# Task 4: Dealing with orientation



## Activation/scalp map polarity is arbitrary (!)

- There IS an absolute orientation for each ELECTRODE when back-projected...
- But you have to determine WHICH scalp location interests you

### Therefore:

- 1) pick one channel's activation orientation
- 2) orient all subjects the same way

```
data = [];  
for subj = 1:length(plotcomps)  
    [EEG ALLEEG CURRENTSET] = eeg_retrieve(ALLEEG,subj);  
    scalefac = sqrt(mean(EEG.icawinv(:,plotcomps{subj}).^2));  
    if subj == 2  
        data = [data squeeze(EEG.icaact(plotcomps{subj},:,:))*scalefac * -1];% reverse  
    else  
        data = [data squeeze(EEG.icaact(plotcomps{subj},:,:))*scalefac];  
    end;  
end;
```

# Task 4: Correctly oriented activations



```
%%% PLOT ERPIMAGE %%%%%%%%%%
```

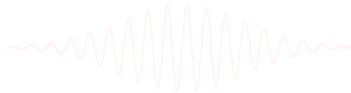
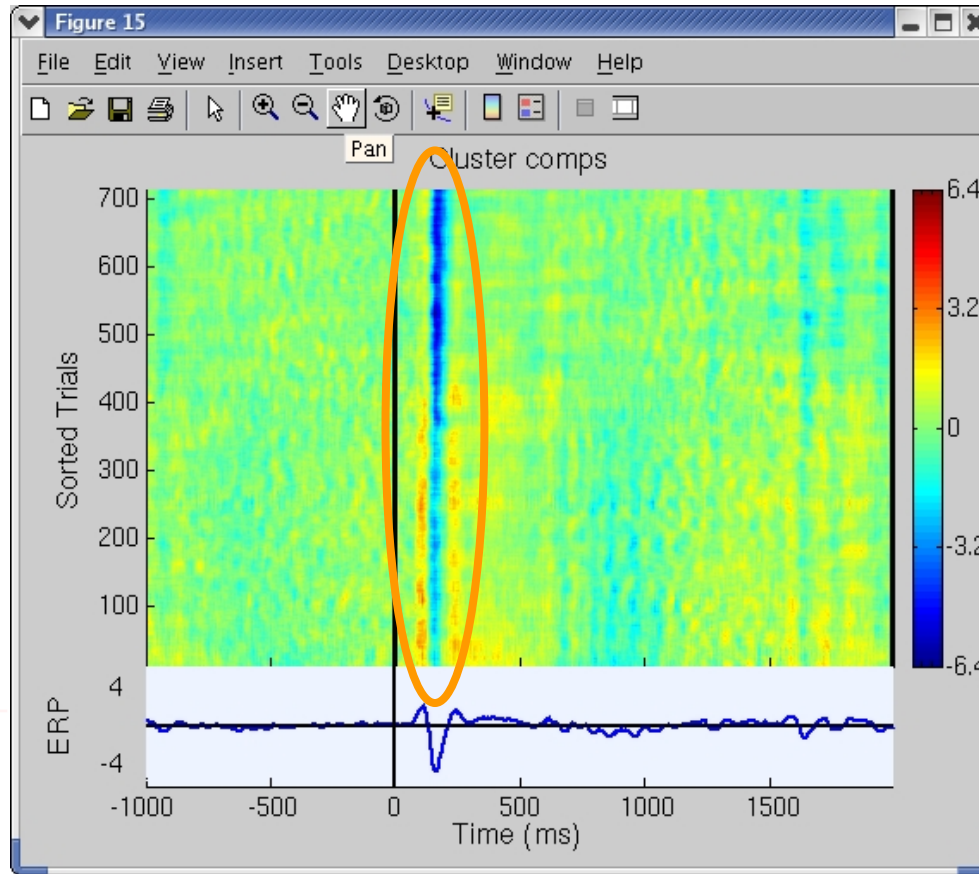
```
figure;
```

```
[outdata,outvar, outtrials,limits, axhndls, erp, ...
```

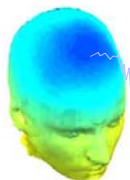
```
amps, cohers, cohsig, ampsig, outamps, phsangls, ...
```

```
phsamps,sortidx, erpsig] = erpimage(data, sortvar,...
```

```
EEG.times, 'Cluster comps', smoothby, 1, 'erp', 'cbar');
```



# Exercise



- Load `'.../data/faces_3.set'`
- Epoch on 'object' and 'face' stimuli
- Remove any noisy epochs
- `'valsort'` the ERP of component **10** at **150 ms**
- **Intermediate:**
  - Collect **sortidx** output
  - Apply sort order to an **activation time course** ERP image of component **4** (try different smoothing factors)
- **Advanced:**
  - Include RTs (use `'eegh'` from gui command) in `'valsort'` plot
  - above and determine **mean/std of RTs** for low, mid and high amplitude trials.
- Is there an effect of ERP size on RT?

